Maximizing AGP Performance

April 23, 1998
Rev. 2.1

Jim Chu Frank Hady

Platform Architecture Labs
Intel Corporation

1.0 Introduction

Graphics intensive applications continue to demand more from the PC Platform. The Accelerated Graphics Port (AGP) has been introduced to address the increasing demands of graphics controllers for main memory bandwidth. High AGP bandwidth is one of a number of key ingredients needed for excellent graphics performance. This paper presents usage guidelines for maximizing AGP bandwidth.

The paper is organized into three main sections: Data Movement, AGP Performance, and Guidelines. The first section relates platform data movement to the three different modes used to move data over AGP. The next section describes the features of each mode, and explains how these features influence the mode's performance. The last section presents guidelines for each mode along with data justifying the guidelines and examples showing the importance of following the guidelines. This paper focuses on AGP specification 1.0 performance.

2.0 Data Movement

Execution of 3D graphics applications requires moving textures, geometry, and commands across AGP to the graphics controller. There are a variety of paths and AGP modes from which to choose. This section describes the path and the AGP modes that may be used.

Figure 1 shows four ways to get data to and from the AGP graphics controller:

- CPU – GC (Frame Mode Only).
- GC – MMF (AGP Frame Mode).
- GC – MMS (AGP Sideband Addressing Mode).
- GC – MMP (AGP Pipelining Mode).

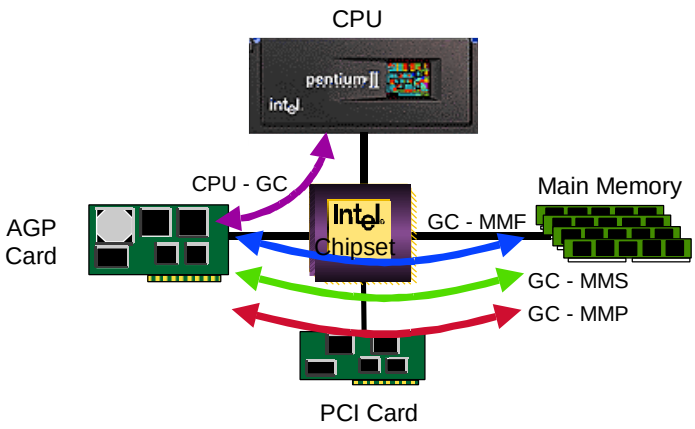


Figure 1: Data Movement Paths and Corresponding AGP Modes.

Data written/read directly to/from the graphics controller is represented by the CPU – GC path of Figure 1. This path is typically used for CPU writes of geometry data and command lists to a memory mapped region on the graphics card. By the AGP 1.0 Specification, data moved along this path, must cross AGP using Frame Mode (PCI protocol).

Data written/read to/from main memory (represented by the arrows marked GC – MMF, GC – MMP, or GC – MMS) is typically texture data, geometry data, or command lists stored in main memory. This path has three choices of modes Frame (F), Pipelining (P) and Sideband Addressing (S). Mode cost and performance and functionality must be considered by the designer when determining the best mode(s) to implement in an AGP graphics controller.

3.0 AGP Performance

Each mode introduced above offers a unique feature set and level of AGP performance. AGP Frame Mode offers baseline performance using the PCI protocol. AGP Pipelining offers higher performance and AGP Sideband Addressing offers the highest level of performance. This section explains the features, performance benefits and performance drawbacks of each mode.

3.1 CPU – GC (Frame Mode Only)

The AGP 1.0 Specification allows only one master for AGP Pipelining and AGP Sideband Addressing. Since this master is the graphics controller, only Frame is available for moving data directly from the CPU to the graphics controller. Guidelines 1, 2, and 3 presented in section 4.1 show how to maximize AGP bandwidth for this path.

3.2 GC – MMF (AGP Frame Mode)

AGP Frame Mode uses the PCI protocol. This mode provides baseline AGP performance for main memory reads and writes from the graphics controller. Baseline AGP performance offers a substantial bandwidth increase over the previous high volume bus for graphics controllers, shared 33MHz PCI. The higher bus frequency allows twice the throughput (264 MB/Sec).

Unlike standard PCI graphics controllers, AGP graphics controllers have a dedicated chipset port, eliminating the possibility of losing bandwidth to competing non-graphics bus traffic. Since Frame Mode uses the PCI protocol some of the performance limitations inherent in PCI graphics controllers are also present on AGP Frame mode graphics controllers. Read performance from main memory is still limited by the need to wait for data to be returned for the current request before another request can be made. This round trip delay time reduces main memory read efficiency, reducing throughput. To maximize bandwidth using AGP Frame Mode follow guidelines 4 and 5 presented later in section 4.2.

Frame Mode is useful when the CPU and the graphics controller communicate via a shared main memory location, since Frame Mode enforces coherency through automatic snooping of the processor's caches. According to the AGP 1.0 spec, Pipelining and Sideband Addressing accesses to main memory may or may not provide coherency by snooping the processor's cache (chipset designer's choice). When the CPU and the graphics card wish to communicate via a shared main memory location using AGP Pipelining or AGP Sideband Addressing Mode, other means (such as the use of uncacheable memory or careful software handshaking) must be employed. The guarantee of coherency inherent in Frame Mode clearly simplifies this type of communication.

3.3 GC – MMP (AGP Pipelining Mode)

AGP Pipelining offers higher AGP performance than Frame Mode. Pipelining uses the PIPE# signal of the AGP protocol to create multiple outstanding transactions, eliminating the bandwidth degradation from round trip time to main memory. Also, Pipelining transfers are not coherent with the CPU's cache. This means that reads or writes are not delayed while the CPU's caches are snooped to see if they contain the most recent copy of the data being read or written. In a system with lots of AGP traffic, these non-coherent transfers avoid heavily loading the system bus and allow data to be returned-to/taken-from the graphics controller more quickly.

Like Frame Mode, AGP Pipelining address and data lines are shared (multiplexed). A multiplexed bus requires turn around cycles to be inserted between most address-to-data and data-to-address transitions. Bus overhead from turn around cycles and address cycles reduce the maximum transfer rate of Pipelining transactions. To maximize bandwidth for AGP Pipelining Mode follow the Guidelines in section 4.3.

3.4 GC – MMS (AGP Sideband Addressing Mode)

AGP Sideband Addressing offers the highest level of AGP performance. Like AGP Pipelining, AGP Sideband Addressing provides multiple outstanding transactions and non-coherent accesses to main memory. In addition, Sideband Addressing introduces a separate address/command bus, the Sideband Address Port (SBA). Because the SBA and data buses are not multiplexed, a graphics controller can use the SBA to initiate data requests without interrupting the data bus. Support for this extra port requires additional graphics controller ASIC pins, possibly increasing graphics controller cost. Feedback from card designers however, indicates that AGP Sideband Addressing is easier to implement than AGP Pipelining because the lack of arbitration for a single shared port. To maximize bandwidth for AGP Sideband Addressing Mode follow the Guidelines in section 4.4.

Both Pipelining and Sideband Addressing offer two data rates 1X (66MHz data rate) and 2X (133MHz data rate). AGP Frame Mode can only use the 1X data rate. At the cost of more high-speed board signals, 2X mode almost doubles the available bandwidth.

4.0 Guidelines

Designers will choose from AGP Frame Mode, Pipelining and Sideband Addressing based on feature, cost and performance considerations. This section presents guidelines to maximize AGP bandwidth. These guidelines are organized according to the paths pictured in Figure 1. Note that the rules that apply for AGP Frame Mode do not apply to AGP Pipelining or AGP Sideband Addressing. Guidelines 6, 7, and 8 are common to both AGP Pipelining and AGP Sideband Addressing. Guideline 9 applies only to AGP Pipelining. The guidelines are as follows:

- CPU – GC Frame Mode Guidelines:
  1. Use Write Combining Memory for CPU to graphics controller writes.
  2. Graphics controller should never refuse data transfers.
  3. Use memory commands, not I/O commands.
- GC – MM AGP Frame Mode Guidelines:
  4. Use long bursts.
  5. Use Memory Read Multiple (MRM) for reads of data spanning more than 1 cacheline (32 bytes).
- GC – MM AGP Sideband Addressing Guidelines:
  6. Use 32 byte (1 cacheline) bursts for reads and writes.
  7. Allow at least 8 simultaneous outstanding transactions.
  8. Minimize the use of High Priority Commands.
- GC – MM AGP Pipelining Guidelines:
  6. Use 32 byte (1 cacheline) bursts for reads and writes.
  7. Allow at least 8 simultaneous outstanding transactions.
  8. Minimize the use of High Priority Commands.
  9. Enqueue at least 4 requests during a single Pipelining transaction.

4.1 CPU – GC Frame Mode Guidelines:

CPU writes to a memory region on the graphcs card must be AGP Frame Mode. Frame Mode performance is sensitive to both burst size, and the memory type . Guideline 1 explains why Write Combining Memory should be used to increase bandwidth. Guideline 2 explains how the graphics controller's ability to accept data from the CPU impacts system performance. The difference between memory and I/O commands is explained in Guideline 3.

1. **Use Write Combining Memory for CPU to graphics controller writes.**
   Un-Cacheable (UC) writes from the CPU to the graphics controller appear on the CPU bus as 4 byte transfers. Each transfer must be handled as a separate transaction. This fills the transaction queue slots of the CPU increasing CPU utilization. Depending on the timing of these writes into the chipset, they may or may not be combined before reaching AGP. Eight UC writes would have to be combined to reach the one cache line recommended size for Memory Writes (MW) commands (see Guideline 4). So it is likely the UC writes to the graphics controller will appear as small AGP Frame Mode transactions, lowering AGP efficiency, and increasing AGP utilization. UC writes from the CPU to the graphics controller will result in throughputs of only 60 to 80 Megabytes/second.

   Write Combining (WC) Memory[1] writes are combined into cacheline sized transfers in the CPU. These writes appear as 32 byte bursts on the CPU bus making much more efficient use of both the CPU and it's bus. These transfers result in efficient 32 byte or longer MW bursts on AGP. WC writes from the CPU to the graphics controller result in throughputs of greater than 200 Mbytes/sec., much greater than the UC bandwidth.

[1] For more detailed information on WC memory and other memory types see Chapter 11.3 of Volume 3 in the Pentium ® Pro Family Developer's Manual set.

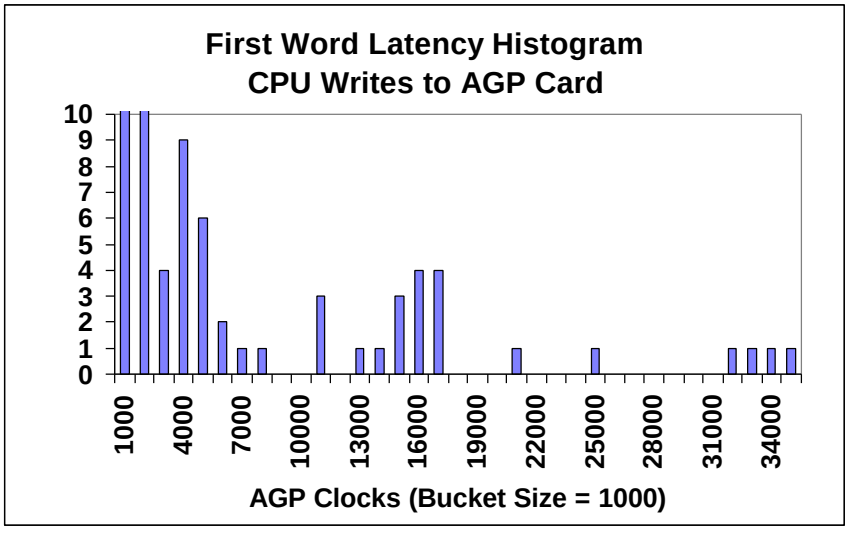
Using WC memory for writes from the CPU to the graphics controller will decreases CPU utilization while increasing throughput.

2. Graphics controller should never refuse data transfers.

Data should only be sent from the CPU to the graphics controller when the graphics controller is known to be ready to accept the data. When data is sent to a full graphics controller it remains posted in the chipset while a series of retries occurs on AGP. The histogram in Figure 2 shows first word latency times for an actual AGP card. The *x*-axis is the number of clocks from the initial read request until the first Dword of data is returned. This time includes all retries between the initial request and the first data returned. The *y*-axis provides a count of the number of reads during which *x* time passed before data was returned. The figure shows that some writes took over 500 microseconds to complete! This is a result of the numerous retries on AGP.

When there are multiple successive CPU to GC writes and a single write requires so long to complete, later writes must be buffered (posted) in the chipset. The chipset diagram in Figure 2 shows an example chipset data queue implementation. Posted writes to a full graphics controller will fill all of the CPU to AGP queues. With these queues full, no Frame mode data may be transferred to AGP. Such a temporary halting of Frame mode service may break multifunction AGP cards (example: A graphics controller also hosting video capture).

Unfortunately, writing to a full graphics controller has become a common practice in many PCI based designs. Since the histogram in Figure 2 is data from an AGP card, it is apparent that some AGP graphics controllers are also refusing to accept data. While the practice may result in slightly higher benchmark scores, it has been found to cause failures in applications featuring Isochronous[2] data delivery (examples include USB, video conferencing and video capture). PCI 2.2 will likely enforce a 10 usec limit on such retries. Intel is working to make such behavior visible through the PCI IPEAK (formerly SPAK) tool available from Hewlett Packard (2925A, 2972), and through a new PCI SIG Compliance Test Card to be supplied with all renewals and new memberships starting in April '98.



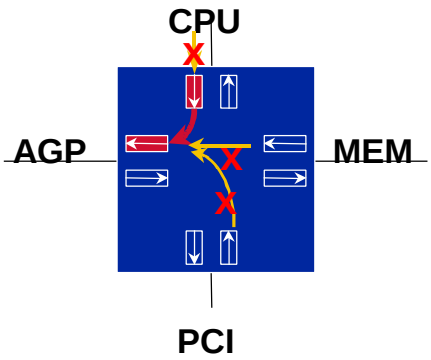


Figure 2: Measured AGP Graphics Card Write Latency & Example Chipset Implementation

3. **Use memory commands, not I/O commands.**
   I/O commands (like those initiated on the CPU with `in` and `out`) are considered to be serializing operations. To ensure this synchronization the chipset may flush all buffers before completing the I/O command, destroying concurrency. I/O commands issued by the CPU also partially serialize CPU execution. I/O commands should be used only when such serialization is desired**.**

4.2 GC – MM AGP Frame Mode Guidelines:

When the graphics controller is serving as an AGP master, baseline AGP performance is provided with AGP Frame Mode. Frame Mode provides guaranteed coherent access to main memory, snooping the processors cache, unlike AGP Pipelining and AGP Sideband Modes. Guidelines 4 and 5 presented below are repeated from "Efficient Use of PCI"[3]. Following these guidelines increases AGP Frame Mode bandwidth. For more information on PCI see the PCI 2.1 Specification[4] and "Efficient Use of PCI"[3]

4. **Use long bursts.**
   - For reads, bursts of at least 64 Dwords (256 bytes) are needed for high bandwidth on some platforms[5].
   - For writes, bursts should be at least as big as one or two cache lines (32 bytes or 64 bytes).

   Figure 3 shows Frame Mode throughput as a function of burst size for a graphics controller acting as an AGP master. Each line represents a different PCI command. Bandwidth increases for MR as burst size increases until 50 Mbytes/sec is reached at 32 bytes. For higher read throughputs use of both MRM and long bursts is required. MRM bursts of 256 bytes provide excellent bandwidth. Longer MRM bursts provide even better bandwidth. While it is not evident from the chart, MR (not MRM) should be used for short reads to decrease the load on main memory (by decreasing fetched but

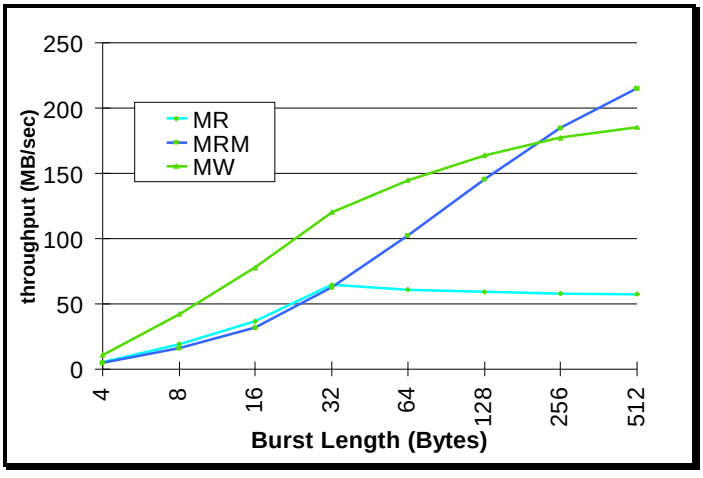


Figure 3: AGP Frame Mode Measurements

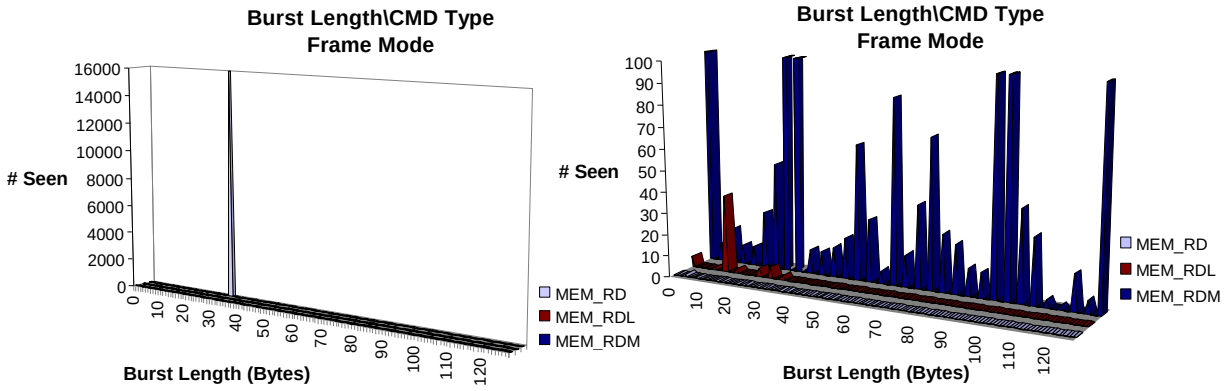
[2] A stream of data that is guaranteed a minimum amount of bandwidth over a time quanta is Isochronous.
[3] "Efficient Use of PCI is at http://support.intel.com/oem_developer/chipsets/pci/general/PCI001.HTM.
[4] For ordering information on the PCI 2.1 Specification contact the PCI Special Interest Group (SIG) at http://www.pcisig.com/specs.html.

unused data). Memory Write (MW) bandwidth also increases with burst size. Memory writes can be posted in the chipset (i.e. the chipset completes the PCI transaction on the initiating bus, PCI, and takes responsibility for ensuring the accepted data is written to main memory across the destination bus, the main memory bus). Since write posting avoids the round trip delay experienced by reads, bandwidth increases more rapidly. 32 byte and 64 byte transfers provide good bandwidth; longer bursts provide even higher bandwidth.

5. Use Memory Read Multiple (MRM) for reads of data spanning more than one (1) cacheline

The MRM command signals to the chipset that a multiple cache line read has been initiated. Most chipsets respond by fetching multiple cache lines from main memory, making longer read transfers possible. In contrast a Memory Read (MR) command causes most chipsets to fetch only a single cache line. Since it has no more data, the chipset terminates a MR at the first cache line boundary. Longer read transfers increase throughput by amortizing the initial round trip time to main memory over more data. On Pentium® Processors, Pentium Pro Processors and Pentium II Processors a cache line is 32 bytes (8 Dwords).

Figure 4 shows a histogram of burst length and command pairs observed for two different AGP graphics cards. Both cards were running the same workload on the same platform. While card A uses 32 Byte MR commands, Card B uses large bursts and advanced commands such as MRM. Both cards are functionally correct on AGP, but Card B wastes fewer AGP cycles while completing the needed transfers. Fewer wasted cycles means either higher maximum bandwidth for the Frame Mode traffic pictured (see Figure 3) or more time to move other necessary data across AGP.



Card A Card B

Figure 4: Two different AGP cards on the same platform running the same workload use different burst lengths and commands.

[5] Chipset implementation differences result in different bandwidth vs. burst size vs. command type plots (more in "Efficient Use of PCI"[3]).

4.3 AGP Sideband Addressing Guidelines:

The rules that apply for AGP Frame Mode do not apply to AGP Pipelining or AGP Sideband Addressing. These two modes offer multiple outstanding requests and the elimination of CPU snoops. Sideband Addressing also offers separate address and data paths.

6. Use 32 byte (1 cacheline) bursts for reads and writes:

Figure 5 shows the motivation for this guideline, AGP Sideband Addressing Mode throughput measurements for varying burst lengths. In all tests enough transactions were outstanding to keep the memory busy.

With 2X AGP Sideband Addressing, extremely high bandwidths are possible. Sideband Addressing bandwidth exceeds 500 Mbytes/second for 32 and 64 byte reads. While 32 byte reads approach maximum throughput or link rate, Figure 5 also shows that 2X Sideband Addressing read throughput for long bursts (256 bytes) or short bursts (8 bytes) are 40% below link rate. Sideband Addressing write bandwidth is greater than 200 Mbytes/sec and concurrent tests show AGP write throughput is higher for (1) cacheline accesses. The rule of thumb for Sideband Addressing transfers, is to use one (1) cacheline sized transfers (32 bytes).

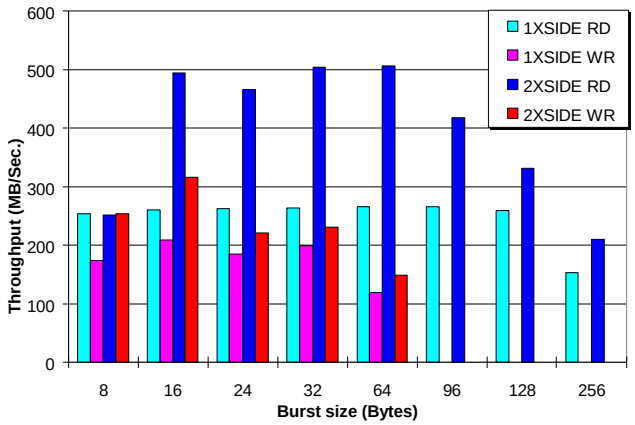


Figure 5: AGP Sideband Addressing Measured Throughput

7. Allow at least 8 simultaneous outstanding transactions.

The ability to create multiple simultaneous outstanding transactions is a key performance feature of AGP. Figure 6 shows three example timelines: one outstanding transaction, eight outstanding transactions for Pipelining and eight outstanding transactions for Sideband Addressing. Each timeline is started on an idle bus with no outstanding transactions. Transactions are the issued as rapidly as possible up to 8 outstanding transactions. Neither the amount of data or number of requests is constant across the three time lines.

With only a single outstanding transaction (top case), much of the AGP port time is wasted (white boxes) waiting for data to return or waiting for transitions between data and command/address. As a result throughput is low. Pipelining with eight outstanding transactions (middle case) eliminates most of this waste by allowing data from previous transactions to return while new transactions are being serviced. In this case, the AGP master initially sends eight commands. After data from the first four commands is returned, four more commands are sent, again resulting in eight outstanding transactions. Multiple outstanding transactions allow the AGP master to keep main memory busy, increasing throughput greatly.

The bottom case, Sideband Addressing with eight outstanding requests, results in even higher throughput since data and address no longer share the same bus. Here the AGP master sends a new command each time data from a previously sent command is returned. With dual paths, data may be returned on every clock. Eight outstanding requests allow an AGP master to maximize AGP bandwidth. More outstanding requests will increase the AGP master's timing flexibility for asserting new requests.

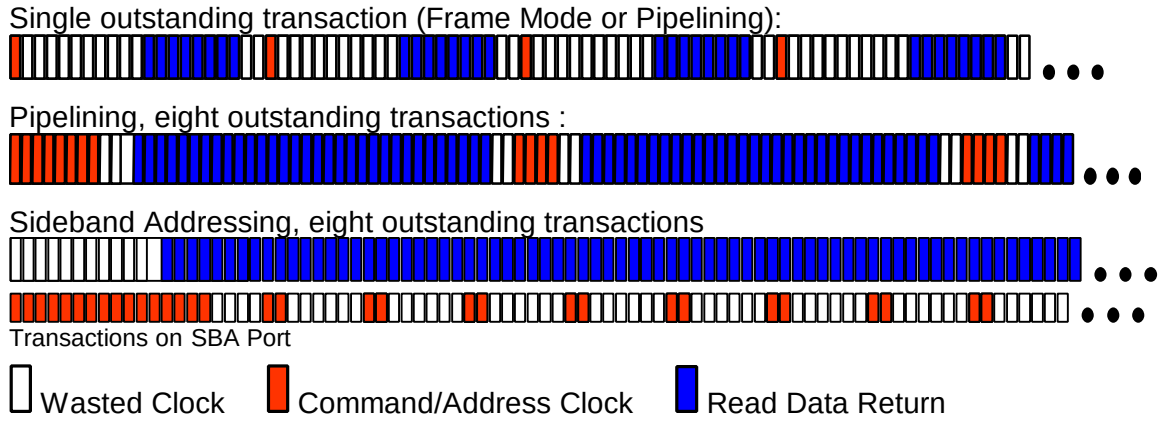


Figure 6: Multiple Outstanding Transactions

8. Minimize the use of High Priority Commands.

High Priority Commands circumvent standard arbitration schemes to deliver low latency access to main memory. Substantial use of high priority commands can cause starvation of other consumers of main memory bandwidth, such as PCI, USB, or even the CPU. High priority commands should be avoided or used only rarely.

4.4 AGP Pipelining Guidelines:

AGP Pipelining Mode offers higher performance than AGP Frame Mode. For smaller burst sizes, Pipelining performance is lower than AGP Sideband Addressing performance since Pipelining employs a multiplexed address and data bus.

6. **Use 32 byte (1 Cacheline) bursts for reads and writes.**
   Figure 7 shows AGP Pipelining Mode throughput measurements for varying burst lengths. In all tests, enough transactions were outstanding to keep the memory busy.

   Pipelining throughputs are substantially higher than Frame mode throughputs since Pipelining allows for multiple outstanding transactions. At the same time, Pipelining throughputs are lower than Sideband Addressing. Sharing of AGP between address and data causes sustained Pipelining throughputs to be less. The drop is especially visible for smaller burst sizes where the address burden is proportionately higher. Pipelining writes provide bandwidths of greater than 200 Mbytes/sec. The rule of thumb for Pipelining transfers is to use single cacheline sized transfers (32 bytes). While not providing the maximum bandwidth, use of this burst size provides excellent bandwidth and eases the transition to AGP Sideband Addressing. Also since chipsets are optimized for cacheline sized access (the CPU has historically been the principle bandwidth consumer), concurrency is handled well for 32 byte AGP accesses.

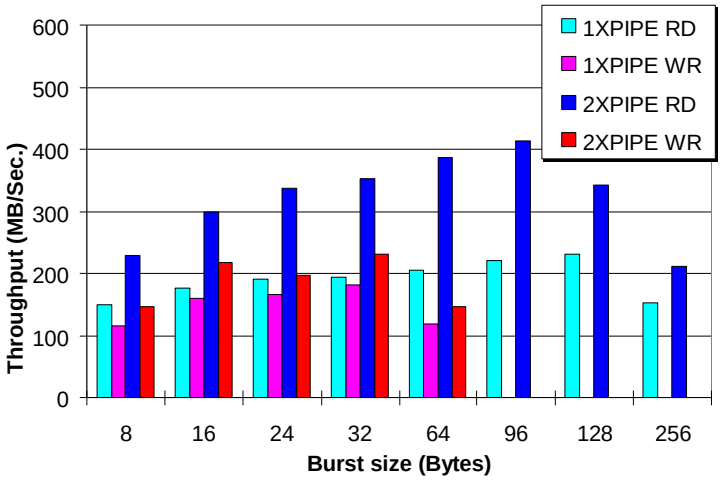


Figure 7: AGP Pipelining Measured Throughput

7. **Allow at least 8 simultaneous outstanding transactions.**
   This guideline is the same as Guideline 7 presented in the section on AGP Sideband Addressing, but here an example using AGP Pipelining Mode is presented.

   The Multiple outstanding transactions allowed by Pipelining Mode keep memory access latency from degrading memory access throughput only when the pipeline of requests is kept full. When Pipelining Mode is used with only single outstanding transaction, Pipelining throughput approaches that of Frame Mode. The histogram in Figure 8 shows commands per AGP PIPE# request. The *x*-axis represents the number of requests en-queued during each active PIPE# cycle. The *y*-axis denotes the number of times each *x*-axis event was observed. The AGP graphics accelerator measured most often issues one AGP command per PIPE# assertion. This particular trace also showed a large delay between PIPE# assertions. Taken together these two facts indicate that this card is likely not utilizing enough simultaneous outstanding transactions and is therefore not achieving high bandwidths on AGP.

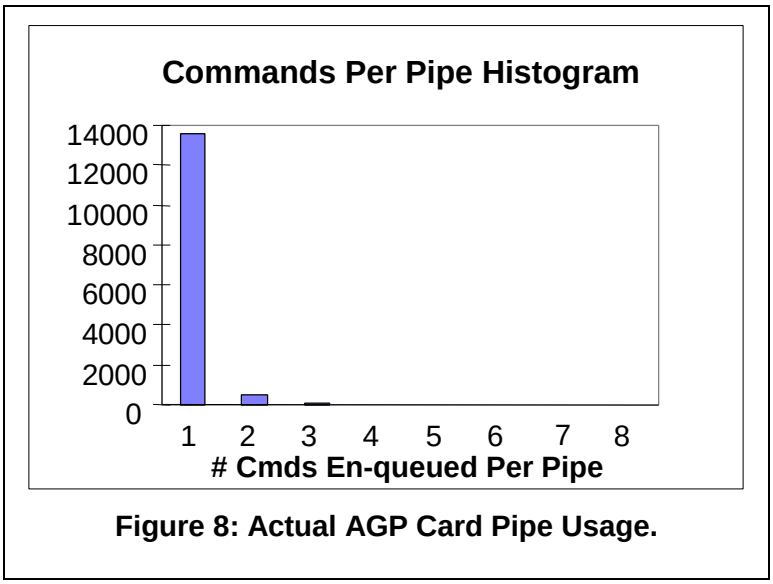


Figure 8: Actual AGP Card Pipe Usage.

8. **Minimize the use of High Priority Commands.** Same as above.

9. Enqueue at least 4 requests during a single pipeline transaction.

As shown in Figure 6, Pipelining requires commands/address and data to share the same signal lines. Transitions from sending data to sending commands/addresses (activation of the PIPE# signal) requires one to two turnaround clocks. Sending multiple commands per PIPE# activation (four were sent in Figure 6) minimizes the resulting throughput degradation. Figure 9 charts theoretical AGP Pipelining throughput versus the number of 32 byte read requests en-queued during a single PIPE# assertion. For 2x, 4 commands per PIPE# assertion provides excellent bandwidth.

Figure 8 shows a graphics controller issuing only a single request per PIPE# assertion. This behavior decreases the efficiency of the bus and reduces the maximum throughput available to the card.

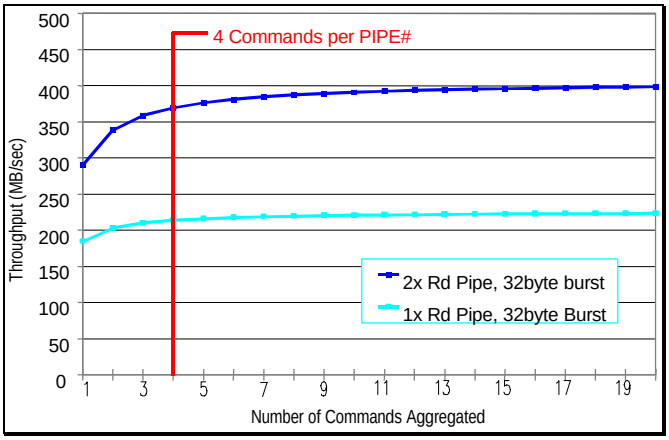


Figure 9: Benefits of enqueueing multiple commands per PIPE#

5.0 Conclusion

AGP is one of the key enabling technologies for high performance 3D graphics applications on the volume desktop PC. Graphics controller vendors should move from the PCI bus to the AGP port. To maximize AGP performance, vendors should move from Frame Mode to Pipelining and to Sideband Addressing wherever possible. Graphics controller designers should assure that their designs follow the guidelines presented here to maximize AGP performance. System integrators should use the guidelines presented here to grade the AGP performance of cards they consider including.