

Application Performance and Loaded Memory Latency

White Paper
(WP)



Revision: 6.0

Original Date: May, 2000

*Copyright© 2000 Intel Corporation
All Rights Reserved*

*Other names and brands are property of their respective owners

Table of Contents

1. Introduction.....3

2. Platform Efficiency and System Performance4

 2.1 Application Performance and Idle System Latency5

 2.2 Loaded Latency7

 2.3 Loaded Latency Measurements and Prediction9

3. Summary13

4. Appendix A: Configurations14

5. Appendix B: Benchmark Descriptions15

List of Figures

Figure 1. Intel® 820/840 Chipset's Performance Comparison with RDRAM vs. Intel® 440BX AGPset & SDRAM 3

Figure 2. Real Memory Latency Output vs. Bus Throughput 8

Figure 3. Lower Latency Equals Better Application Performance 9

Figure 4. Latency vs. Bandwidth for Alternative Memory Configurations 11

List of Tables

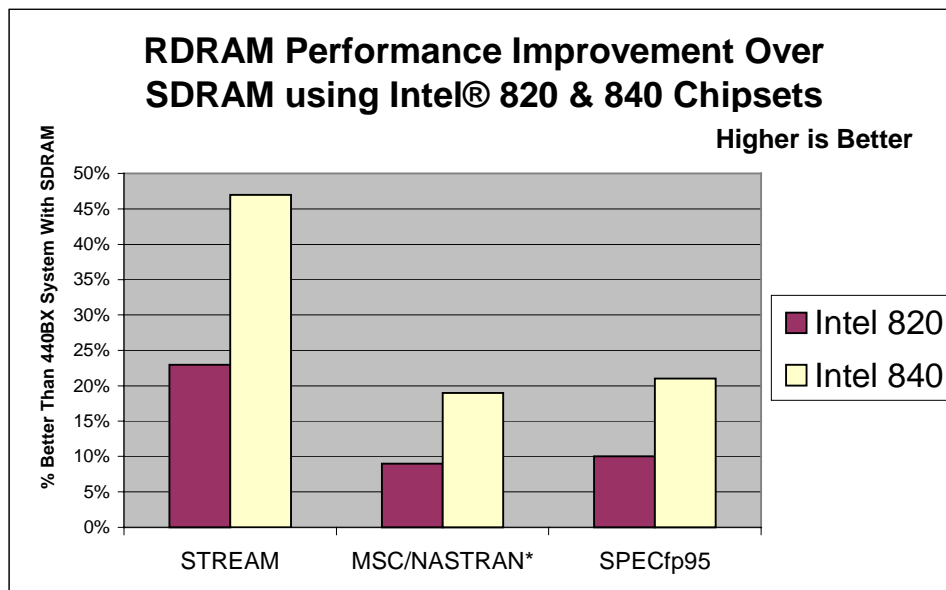
Table 1. Intel® 840 Chipset Benchmark Results Showing Relative Performance..... 6

Table 2. Basic System Configurations..... 11

1. Introduction

Application execution times are influenced by many factors, among them processor frequency and platform efficiency. For most desktop, workstation, and server applications, the memory latency or delay seen by the processor core while running actual workloads (i.e., loaded latency) dominates platform efficiency. Main memory latency is often expressed as constant time or clock count. However, loaded latency is a complex function that is influenced by memory technology, request rate, available memory bandwidth, memory access patterns, and chipset scheduling of memory commands.

The superior data and command bandwidth of RDRAM, faster system bus, and the advanced scheduling abilities of the Intel® 820 and 840 chipsets allows delivery of data and instructions to the processor with extremely low latencies under heavy loads. This enables superior application performance. Figure 1 demonstrates the measured platform advantages of the Intel® 820/840 chipsets over the Intel® 440BX AGPset.



Source: Intel Corporation

Note: The system configurations for this benchmark test are described in Appendix A, systems 1, 2, and 3.

Figure 1. Intel® 820/840 Chipset's Performance Comparison with RDRAM vs. Intel® 440BX AGPset & SDRAM

SPECfp95 benchmark tests reflect the performance of the microprocessor, memory architecture and compiler of a computer system on compute-intensive, 32-bit applications. SPEC benchmark tests results for Intel® microprocessors are determined using particular, well-configured systems. These results may or may not reflect the relative performance of Intel microprocessor in systems with different hardware or software designs or configurations (including compilers). Buyers should consult other sources of information, including system benchmarks, to evaluate the performance of systems they are considering purchasing. For more information about SPEC95, including a description of the systems used to obtain these test results, and other information about microprocessor and system performance and benchmarks, visit Intel's World Wide Web site at www.intel.com or call 1-800-628-8686.

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel® products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, reference <http://www.intel.com/procs/perf/resources/spectrum.htm>

2. Platform Efficiency and System Performance

The performance of a typical desktop or workstation application depends upon the performance of both the processor and platform that surrounds it. Upper limits of application performance are set by the rate at which the processor can execute instructions. The platform efficiency determines how close to this “upper limit” an application executes. The maximum execution rate is directly related to the time it takes for the system processor to execute a single instruction and is often expressed in terms of processor frequency. However, as advancing semiconductor technology pushes frequency levels to new heights, platform efficiency becomes increasingly important.

Processor designers utilize a variety of methods and design enhancements to minimize the effects of system latency and push application performance as close as possible to the upper performance limits. Parallelism, pipelining, and transaction reordering allows sequential instructions to execute concurrently. Large caches and carefully designed cache line replacement policies are two of the methods used to minimize the processor efficiency losses caused by the interaction with slower system memory.

The efficiency of a processor and system as applied to instruction execution may be reduced to a simple equation, in which efficiency (or lack of efficiency) is expressed by the term clocks per instruction (CPI):

$$\frac{\text{InstructionsExecuted}}{\text{Second}} = \frac{\left(\frac{\text{clocks}}{\text{Second}} \right)}{\left(\frac{\text{clocks}}{\text{InstructionsExecuted}} \right)}$$

OR

$$\frac{\text{InstructionsExecuted}}{\text{Second}} = \frac{\text{ClockFrequency}}{\text{CPI}}$$

Clearly, a system with fewer clocks per instruction provides higher performance, since more instructions are executed for a given timeframe. Consequently, the focus of this paper is mainly on platform efficiency. The assumption is to hold the processor frequency constant and focus on approximating *CPI* as a key measure for system performance.

The number of clocks required for a given instruction depends on whether the operands are in the Level 1 cache (L1), Level 2 cache (L2) or must be fetched from system memory. Therefore, the system CPI, as seen by a given processor, may be approximated by a fairly straightforward equation:

$$CPI = CPI_0 + (L2 \text{ Access Time} * P_{L2}) + (Average \text{ system latency} * P_{Sys})$$

Where:

- CPI_0 is the baseline CPI, achievable if all required data/instructions are available in the L1 caches
- P_{L2} is the probability that an instruction requires access to the L2 cache
- P_{Sys} is the probability that an instruction requires access to main memory

For purposes of evaluating platform performance, CPI_0 , all of the probability components, the L2 access time, and any hidden efficiency implemented by the processor designers in order to hide system latency, may also be considered constant.

Based on the previous model, the memory latency is the key platform-level factor contributing to an application's CPI. All of the other terms of the equation relate purely to processor design.

2.1 Application Performance and Idle System Latency

The key component that interfaces with the processor to system memory is the chipset. In 1998 Intel introduced the Intel® 440BX AGPset supporting the PC100 SDRAM memory. In 1999 Intel advanced the computing platform with the introduction of the first chipset supporting RDRAM memory. The Intel® 820 chipset was introduced for the performance desktop market, and the Intel® 840 chipset with RDRAM was introduced for the workstation market. The primary memory performance difference between the Intel® 820 chipset and the Intel® 840 chipset, is the number of RDRAM based memory channels. The Intel® 820 chipset supports a single RDRAM memory channel while the Intel® 840 chipset supports dual memory channels.

Assuming latency is directly translated into application performance, average system latency can be inferred from measured differences in application performance. Therefore, a processor-to-memory intensive benchmark that performs better on System A than on System B implies that System A has lower effective latency than System B.

The following table illustrates various benchmark results that may be indicative of latency differences.

Table 1. Intel® 840 Chipset Benchmark Results Showing Relative Performance (Higher Is Better)

	440BX²	Intel® 820¹	Intel® 840¹
STREAM	1.00	1.23	1.47
Nastran*	1.00	1.09	1.19
Spec FP95	1.00	1.10	1.21

1. Intel® 820 and Intel® 840 chipset results used 733 MHz processor /133 MHz bus.

2. 440BX results used 750 MHz processor /100 MHz bus.

SPECfp95 benchmark tests reflect the performance of the microprocessor, memory architecture and compiler of a computer system on compute-intensive, 32-bit applications. SPEC benchmark tests results for Intel® microprocessors are determined using particular, well-configured systems. These results may or may not reflect the relative performance of Intel microprocessor in systems with different hardware or software designs or configurations (including compilers). Buyers should consult other sources of information, including system benchmarks, to evaluate the performance of systems they are considering purchasing. For more information about SPEC95, including a description of the systems used to obtain these test results, and other information about microprocessor and system performance and benchmarks, visit Intel's World Wide Web site at www.intel.com or call 1-800-628-8686.

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel® products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, reference: <http://www.intel.com/procs/perf/resources/spectrum.htm>

Based on the initial results shown in the previous table, the assumption is that the Intel® 840 chipset has a lower effective latency than the Intel® 440BX AGPset. The results further show that STREAM, MSC/NASTRAN, and SPECfp95 all demonstrate higher performance on the Intel® 840 chipset than on the Intel® 440BX AGPset. The implication, then, is that RDRAM has lower latency than SDRAM. However, it must be noted that the performance gain of the Intel® 840 chipset differs across the various benchmarks. Since latency and application performances are meant to be directly correlated, the relative performance of different applications on a given set of systems must be consistent. However, a direct correlation is not valid if latency has a dynamic component that is somehow influenced by the interaction between the processor and chipset when running a particular application.

To summarize, this data shows that a constant value representing idle system latency is not a good predictor of real-world platform performance. Rather, a more thorough analysis is required.

2.2 Loaded Latency

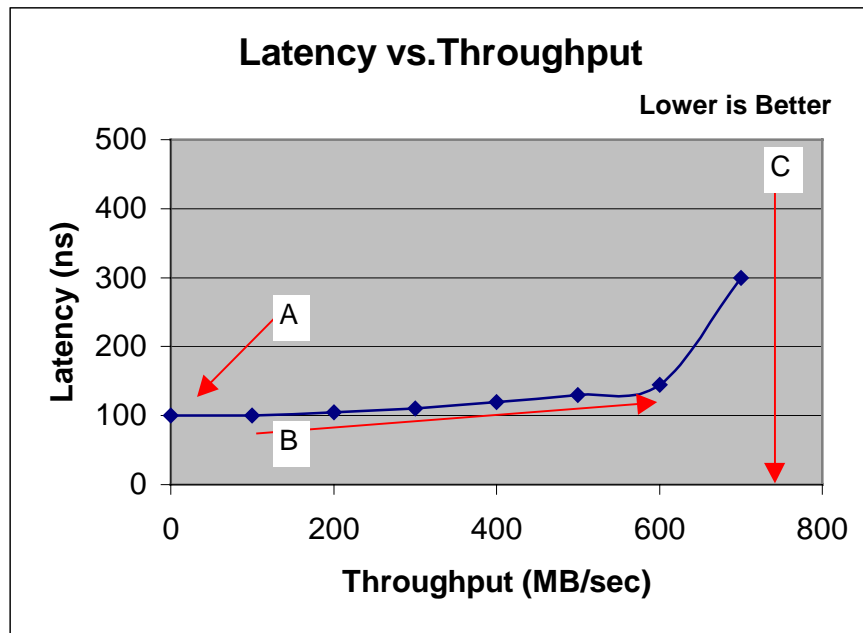
Often, the numbers quoted for chipset-memory latency are naïve and simplistic indicators, which frequently misrepresent relative application performance. The Intel® 440BX AGPset using SDRAM, for example, is said to deliver an 8-clock latency to memory. This narrow metric of chipset performance obscures both a great deal of design subtlety and inherent SDRAM protocol inefficiency. For example, the Intel® 440BX AGPset is only able to achieve this 8-clock latency when using non-registered DIMMs in a system with very few SDRAM devices connected to the memory controller. If one were to populate registered DIMMs or a pair of DIMMs with fine-grain memory devices, the number then becomes 9. Both populated together can lead to a “leadoff” of 10-clocks. This latency has the potential to balloon to 11-clocks if the most cost-effective DRAM devices are used.

Further, this 8/9/10/11-clock leadoff only describes the memory subsystem’s page hit behavior. A transaction is said to be a page hit when the memory controller has previously opened the page in an appropriate DRAM bank, in order to service an earlier transaction request. If the page is not initially opened, then the transaction becomes either a page empty or a page miss, which can be said to take from 2-to-4 or from 3-to-6 additional cycles, respectively, depending upon the speed of the DRAM device. The ratio of page hits to page empties and page misses depends upon an application’s locality, which is a term that describes how close together an application’s consecutive memory operations are located. Applications with high-locality can cause high-page hit rates and are conducive to chipset “prefetching” techniques. Conversely, applications with low-locality may cause low-page hit rates and therefore experience higher leadoffs.

However, even this degree of subtlety does not provide a robust model for application performance prediction. This is due largely to the fact that all of the metrics and conditions previously described still deal only with the chipset-memory system’s read behavior under idle conditions. Both intuition and empirical evidence show that this idle condition does not dominate application behavior. In fact, real applications (running on Pentium® III processors) tend to exhibit a “bursty” behavior, meaning they request several transactions in a row and then pause for a short time before requesting further memory operations. In addition, real applications always cause the processor to write into the memory system. Both of these application characteristics add substantial amounts of time to the latency seen by memory transactions, because the P6 processor bus is incapable of delivering data as quickly as it can accept requests. The electrical turnarounds required by adjacent read-write or write-read operations only exacerbate the situation.

Taken together, these characteristics are sometimes called queuing effects and demonstrate that even simple system loading often adds latency to the ideal numbers quoted in a product datasheet. This loading is reflected in bus throughput. As a result, there is a powerful correlation between the throughput required for an application and the average memory latency seen by that application.

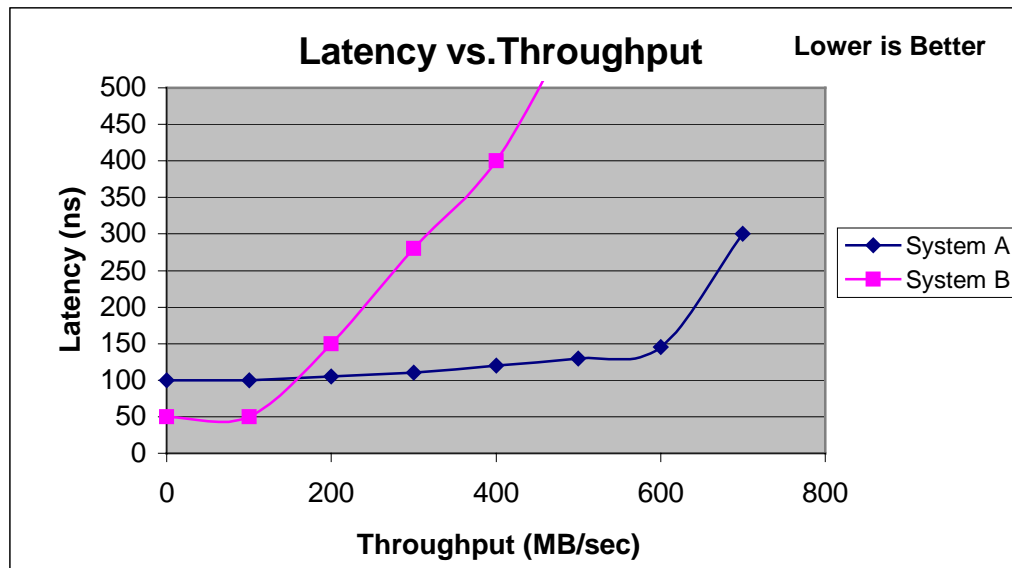
Figure 2 illustrates the effects of bus throughput on real memory latency. Point A is an indication of the latency to be expected in an idle system, where the requested bus throughput is near 0 MB/sec. Slope B is indicative of the latency penalty paid for system loading while the system is still able to reasonably respond to processor requests. Line C represents a throughput asymptote, which a chipset-memory implementation is unable to exceed under a set of loading conditions. Both the slope and the asymptote are dependent upon platform design and application request types.



Source: Intel Corporation

Figure 2. Real Memory Latency Output vs. Bus Throughput

Different memory systems and memory controller implementations exhibit different characteristic curves. These characteristic curves capture a great deal of memory subsystem implementation detail and can be used to explain the benchmark differences shown previously. Take, for example, a hypothetical memory technology or controller that has a very low initial latency and very poor pipelining characteristics for a given request pattern. As seen in Figure 4, when the technology's characteristic curve (System B) is superimposed on the sample graph (System A), it becomes apparent that the latency prediction depend heavily upon the application's requested throughput. If the application causes the processor to request 100 MB/sec of throughput from the bus and the memory system, then the new "low latency" technology System B delivers data with low latency. If the processor request stream is closer to 300 MB/sec, however, then the other (System A) technologies deliver substantially lower latency and therefore, better application performance.



Source: Intel Corporation

Figure 3. Lower Latency Equals Better Application Performance

2.3 Loaded Latency Measurements and Prediction

The power of using loaded latencies and “latency vs. throughput” charts to characterize platform performance lies in the fact that the charts can be readily generated through simulation and easily captured using standard lab techniques. The later approach has a tremendous advantage, since simulation of real applications over interesting time intervals is fairly impractical. Once several system implementations and an application are characterized, the intersection of the systems’ characteristic curves and the application’s average requested throughput are used to estimate relative system performance. Given that processor speed improvements typically compress the time taken for application execution, and increase the throughput requested over a given time period, application performance relative to future processors also may be characterized.

To study the latency vs. bandwidth behavior of alternative memory configurations, a modified version of the STREAM benchmark has been used to generate memory traffic using a specific mix of read and write transactions. In its original form, STREAM is a synthetic benchmark used to measure maximum sustained system bus bandwidth. It uses four- (4) long vector kernels to generate various mixes of processor-to-memory read and write transactions. The sizes each vector is defined much larger than available cache size to eliminate the possibility of data reuse.

This STREAM Triad kernel, frequently used in many algorithms from audio processing to math libraries, was used to generate various loading levels on the system bus. This was accomplished by adding a delay loop to adjust the rate at which system bus transactions are used by the processor.

```
temp = 0.0;
for (i = 0; i < Vector_Length; i++)
{
    a[i] = k * b[i] + c[i];           /* k is a constant */
    for (j = 0; j < Delay_Length; j++) /* delay loop */
        temp = temp + 1.0;
}
```

The delay loop is executed entirely within the processor without requiring memory access. Adjusting the value of the Delay_Length constant can control the amount of time spent in this loop. Three- (3) difference values were used to generate the loading levels corresponding to the first, three- (3) latency/bandwidth measurements for each configuration. The fourth measure was obtained by reverting to the original STREAM Triad operation without the delay loop. Finally, the fifth measurement was obtained by increasing the rate at which memory transactions are issued by stepping through cache lines instead of individual vector elements. This can be accomplished by incrementing the loop variable *i* by 4 for each iteration of the loop. (The individual vector element are 8 byte doubles, thus there are four- [4] elements per 32 byte cache line).

Bandwidth and latency measurements were performed using EMON*; a performance monitoring tool that utilizes the hardware counters available on the processor. Bandwidth consumption was derived from a counter that monitors all processor clocks during which data is being transferred over the system bus. Read latency, on the other hand, was derived by dividing the number of outstanding read transactions by the number of completed read transactions (burst RD transactions excluding RFO and IFETCH).



NOTE

The number of outstanding transactions incorporates a built-in measure of latency, because each transaction is essentially counted as many times as the number of cycles that it remains outstanding.

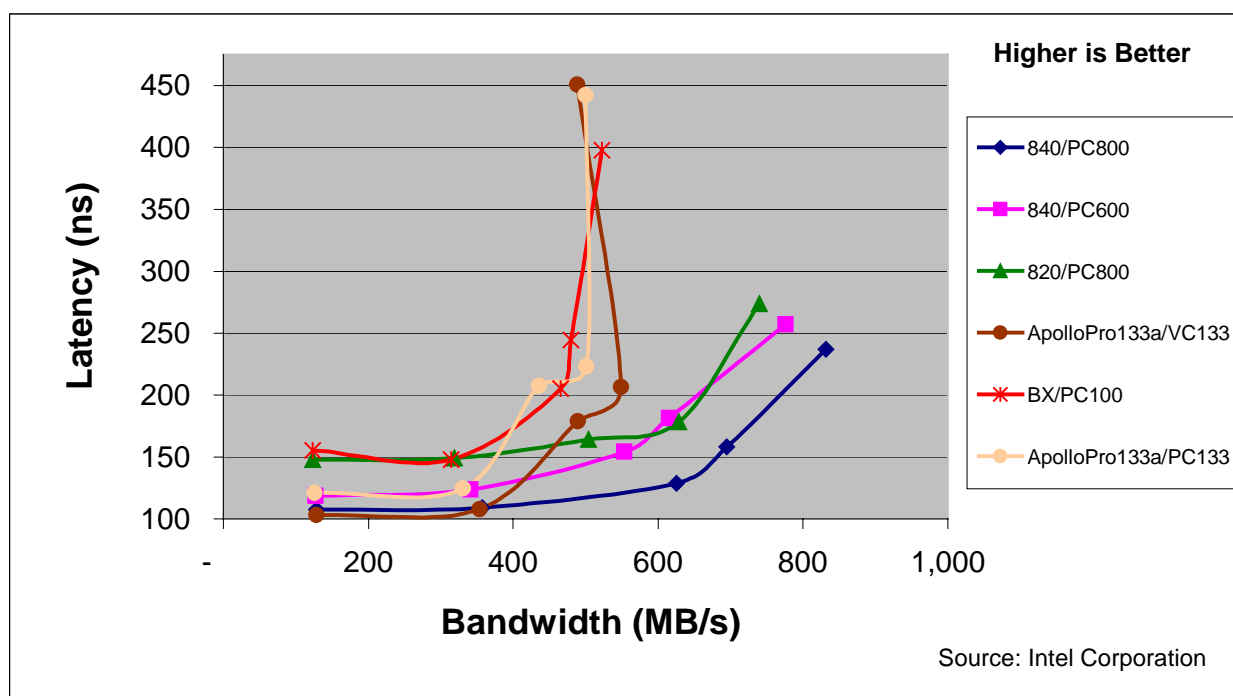
Latency measured in this fashion reflects the time from discovering an L2 cache miss to the time that the last chunk of data arrives at the processor. However, the measurements have been adjusted to discount the time needed to bring in the last three- (3) chunks of data.

The transfer of a 32 byte cache line over the 8 byte-wide system bus requires four- [4] bus cycles.

The characteristic curves shown in Figure 4 were obtained by applying this methodology to a number of alternative SDRAM and RDRAM-based system configurations, all using virtually identical Pentium® III processors (only the Intel® 440BX AGPset-based platform uses a 750 MHz processor; all other configurations use 733 MHz).

Two- (2) curves have been included for the ApolloPro133a* configuration using PC133 memory as well as PC133 memory with virtual channel capabilities referred to as VC133.

The curves show that the “Via Apollo PC133a” implementation with VC133 delivers the lowest latency under light loading conditions. This implies that for low-throughput applications, the Apollo is likely to deliver the best application performance. However, for more stressful applications, platforms based on the Apollo demonstrate extreme system inefficiencies. In fact, applications that cause the processor to demand more than 600 MB/sec of data from main memory find themselves “choked,” as they waste processor cycles waiting for main memory access.



Note: The system configurations for this benchmark test are described in Appendix A. Refer back to Table 2 for the basic system configurations.

Figure 4. Latency vs. Bandwidth for Alternative Memory Configurations

Table 2. Basic System Configurations

Processor Speed	Chipset	System Bus Speed	Memory Type
733 MHz	Intel® 840 chipset	133 MHz	RDRAM-PC800 & PC600
733 MHz	Intel® 820 chipset	133 MHz	RDRAM-PC800
733 MHz	Via Apollo Pro133a*	133 MHz	SDRAM-PC133 & VC133
750 MHz	Intel® 440BX AGPset	100 MHz	SDRAM-PC100

*Other names and brands are the property of their respective owners.

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel® products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, reference <http://www.intel.com/procs/perf/resources/spectrum.htm>.

It is important to note that the performance advantages currently visible in Intel® 820 and Intel® 840 chipsets are likely to become more pronounced as processor speeds increase. If, for example, an application currently requires 500 MB/sec with a 733 MHz Pentium® III processor, it is easy to extrapolate that the same application attempts to consume 682 MB/sec with a hypothetical 1 GHz Pentium® III processor. Under this scenario, the average latency difference between the Intel® 820 chipset with RDRAM implementation and the Intel® 840 chipset with RDRAM implementation grows from approximately 45 ns to around 80 ns. The magnitude of application performance differences seen clearly depend upon many system factors; however, it is likely to change a ~5% difference in application performance into a ~ 8% difference. Ignoring the absolute magnitudes for the moment, the delta works to widen the performance gap between the Intel® 820 and Intel® 840 chipsets. At the same time, the SDRAM technologies used by the Intel® 440BX AGPset and Apollo* chipsets are then extremely inadequate.

3. Summary

Application execution times are influenced by many factors, among them processor frequency and system efficiency. System efficiency may be encapsulated in a single parameter, CPI, which reflects the number of clocks required for the processor to execute a single instruction. CPI is dominated by the latency seen by the processor core when it attempts to access main memory.

Main memory latency, while often expressed as a simple number, is actually a complex function that is influenced by memory technology, requested and available memory bandwidth, memory access patterns, and the efficiency of the chipset in its scheduling of memory commands. This latency is best expressed as a loaded latency or “Throughput vs. Latency” graph that plots the latency seen by the processor when requesting read data from memory against the amount of data being requested by the processor per second. These charts then help to illustrate the differences in performance between chipsets and memory technology and permit extrapolation of chipset application performance with hypothetical future processor speeds.

4. Appendix A: Configurations

The following configurations have been used for benchmarking and latency analysis:

1. Intel® OR840 platform with 733 MHz Pentium® III processor, 133 MHz system bus, Intel® 840 chipset, 2 x 256 MB of PC800 RDRAM without MRH-R, memory ECC on, E&S Tornado 3000* graphics, Adaptec* 2940U2W SCSI controller with two Seagate Cheetah* 10K RPM hard drives, Windows® NT 4.0* Build 1381.
2. Intel® VC820 platform with 733 MHz Pentium® III processor, 133 MHz system bus, Intel® 820 chipset, 2 x 256 MB of PC800 RDRAM without MRH-R, memory ECC on, E&S Tornado 3000* graphics, Adaptec* 2940U2W SCSI controller with two Seagate Cheetah* 10K RPM hard drives, Windows® NT 4.0* Build 1381.
3. Asus* P2B-F platform with 750 MHz Pentium® III processor, 100 MHz system bus, Intel® 440BX chipset, 4 x 128 MB of PC100 SDRAM with 2-2-2 timing, memory ECC on, E&S Tornado 3000* graphics, Adaptec* 2940U2W SCSI controller with two Seagate Cheetah* 10K RPM hard drives, and Windows® NT 4.0* Build 1381.
4. Tyan* Trinity400 platform with 733 MHz Pentium® III processor, 133 MHz system bus, ApolloPro133a chipset, 3 x 128 MB¹ of PC133/VC133 SDRAM with CL=2, E&S Tornado 3000* graphics, Adaptec* 2940U2W SCSI controller with two Seagate Cheetah* 10K RPM hard drives, Windows® NT 4.0* Build 1381.

*Other names and brands are the property of their respective owners.

¹ The Tyan* platform has only 3 DIMM sockets; however, the Stream benchmark used for latency testing requires no more than 128 MB of memory.

5. Appendix B: Benchmark Descriptions

STREAM is a synthetic benchmark, which measures the maximum sustainable memory bandwidth. It uses four- (4) long vector operations to generate four- (4) different mixes of processor-to-memory read and write transactions. The size of each vector is defined larger than the cache size so that data re-use (either inside the cache or in registers) is eliminated. Bandwidth measurements in MB/s are obtained by simply dividing the size of all data transferred over the bus during a particular vector operation by the amount of time it takes to carry out that operation.

STREAM is not designed to model a real workstation application, but rather to evaluate a particular architectural feature of the machine. However, no workstation benchmark exercises the memory system as much as STREAM does. Improvement in STREAM performance does not automatically mean that a similar improvement will be measured for workstation applications in general. This is because memory bandwidth requirements vary substantially from one workstation benchmark to another. In addition, workstation applications, which rely heavily on access to main memory, can be much more sensitive to memory latency than they are to memory bandwidth.

MSC/NASTRAN* is a leading Finite Element Analysis (FEA) program from MacNeal-Schwendler Corporation. It offers a wide variety of analysis types, including linear statics, normal modes, buckling, heat transfer, dynamics, frequency response, transient response, random response, response spectrum analysis, and aeroelasticity. MSC's* solutions have played a key role in the design of virtually every major automobile, aircraft, and space vehicle developed in the past decade. MSC* products have helped ensure the product performance, safety, and reliability of a broad spectrum of products. Like other memory-intensive applications, MSC/NASTRAN* is sensitive not only to processor speed but also to memory throughput and access times. MSC/NASTRAN* version 70.5.5* has been optimized to use Streaming SIMD Extensions for improved memory throughput when running on Pentium® III Xeon™ processors.

The BCLL benchmark represents the use of MSC/NASTRAN* to analyze a cube of solids with a plate on each surface of each solid, measuring overall processing time. The lgamd benchmark represents work typically done by engineers working in automotive design. It models the behavior of the transmission part of an automobile engine, analyzing its behavior when subjected to environmental vibration. bd03fix is a small tutorial model of an angle bracket that represents a “textbook” example of normal mode analysis. The benchmark used in this white paper is an average of these data sets.

SPECfp95* is a set of floating-point benchmarks produced by the Standard Performance Evaluation Corporation. SPEC95* provides a standard for comparing compute-intensive workloads on different computer systems, consisting of measuring and comparing compute-intensive floating-point performance. This provides component-level benchmarks that measure the performance of the computer's processor, memory architecture and compiler.

*Other names and brands are the property of their respective owners.