Second International Symposium on High-Performance Computer Architecture, San Jose, CA, February 1996.

PERFORMANCE CHARACTERIZATION OF THE ALPHA 21164 MICROPROCESSOR USING TP AND SPEC WORKLOADS

Zarka Cvetanovic and Dileep Bhandarkar[+]
Digital Equipment Corporation

Abstract

This paper compares the performance characteristics of the Alpha 21164 to the previous-generation 21064 microprocessor. Measurements on the 21164-based AlphaServer 8200 system are compared to the 21064-based DEC 7000 server using several commercial and technical workloads. The data analyzed includes cycles per instruction, multiple-issued instructions, branch predictions, stall components, cache misses, and instruction frequencies. The AlphaServer 8200 provides 2 to 3 times the performance of the DEC 7000 server based on the faster clock, larger on-chip cache, expanded multiple-issuing, and lower cache/memory latencies and higher bandwidth.

1. INTRODUCTION

This paper analyzes the performance characteristics of major architectural improvements in the second generation Alpha 21164 microprocessor using several commercial and technical workloads. The workloads include transaction-processing (TP1) and commercial/multi-user UNIX workloads (AIM-III), as well as technical/scientific workloads (SPEC92 and Sparse Linpack). The measurements were done on the high-end AlphaServer 8400/8200 system (the AlphaServer 8200 and 8400 have similar performance characteristics in uniprocessor configurations).

The performance characteristics were obtained by using several profiling tools based on built-in non-intrusive hardware monitors: CPU performance counters and memory-interconnect performance counters. These monitors collect various events including the number and type of instructions issued, multiple-issues vs. single-issues, branch mispredicts, stall components, cache misses, memory read/write latencies, data sharing in a multiprocessor system, etc. The monitors are a useful tool for analyzing system behavior under various workloads. The results of this analysis can be used by computer architects to drive hardware design tradeoffs in future system designs.

The 21164-based AlphaServer 8200 provides 2 to 3 times the performance of the previous-generation 21064-based DEC 7000 server. The clock speed improvement (300 MHz vs. 200 MHz) provides 50% gain, the remaining factor of 1.3 to 2 times comes from several micro-architectural improvements. The most significant factors that contribute to this gain include: the second-level on-chip cache (96 KB), lower cache latency, and quad instruction issue. Lower memory latency and higher bandwidth in the AlphaServer 8200 platform also contribute to higher performance.

A simple model based on the data collected by using hardware monitors is proposed for estimating the performance effect of various stall components. The model is useful for predicting the future performance trends as the processor and memory hierarchy are further enhanced.

2. WORKLOAD DESCRIPTION

The commercial workloads analyzed include the TP1 benchmark (called Debit-Credit in [7]) and AIM-III [12] benchmark. The TP1 benchmark (a basis for TPC-A benchmark) was defined in 1985 by a group of 25 database experts to measure the transaction processing performance. The TP1 benchmark is representative of on-line transaction processing and collects the activities of a banking network including database reads and writes, terminal I/O, and transaction commits. The AIM-III benchmark is representative of time-sharing multi-user commercial workloads.

The technical/scientific workloads analyzed here include SPEC92 integer and floating-point suite (SPEC) and Sparse Linpack. SPEC CINT92 is an integer suite containing six benchmarks written in C and represents circuit analysis,

[+] Currently with Intel Corporation.

LISP interpreter, logic design, text compression, spreadsheet, and software development applications. SPEC CFP92 consists of fourteen floating-point benchmarks, two written in C and the rest in FORTRAN, and it represents application areas in the circuit design, Monte Carlo simulation, quantum chemistry, optics, robotics, quantum physics, astrophysics, weather prediction, and other scientific and engineering problems. Sparse Linpack is representative of large scientific applications that do not fit in the board-level cache.

3. SYSTEM DESCRIPTION

The 300 MHz Alpha 21164 superscalar pipelined processor fabricated in a 0.5 micron CMOS technology is the central processor in the AlphaServer 8200. The chip has three on-chip caches: 8 KB instruction cache (Icache), 8 KB data cache (Dcache), and 96 KB second-level data/instruction cache (Scache). The primary caches (Icache and Dcache) are direct-mapped and write-through. The second-level cache (Scache) is 3-way set-associative and write-back. The 21164 has two 64-bit integer pipelines, a floating-point add pipeline, and a floating-point multiply pipeline. The chip also has a 48-entry, fully associative instruction translation buffer (ITB) and a 64-entry fully associative data translation buffer (DTB). The cache latencies and bandwidth of two generations Alpha servers are compared in Table 1. More detailed descriptions of the 21164 and 21064, and systems based on those chips can be found in [1][2][4][5][6][10][11].

In the AlphaServer 8200, the 4 MB board-level backup cache (Bcache) uses a write-back, conditional update, write allocate cache coherency protocol. The writes to shared blocks are broadcast on the bus, and all the processors invalidate their copy of the shared block. The memory interconnect bus is non-pended, pipelined, with 64-byte memory transfers and distributed arbitration [6]. The main advantages over the DEC 7000 bus are: separate address and data busses, data bus width increased from 128 bits to 256 bits, synchronous design (bus cycle is multiple of a CPU cycle, compared to a fixed bus cycle in DEC 7000), reduced bus cycle (13.3 ns vs. 20 ns in DEC 7000), multiple outstanding transactions per node, and early bus arbitration. As a result, AlphaServer 8200 achieves lower memory latency and higher bandwidth.

The CPU and memory interconnect include special hardware counters that allow monitoring of various events [10][11]. The events that can be monitored on the CPU chip include: cycles, issues, non-issues, pipeline stall cycles, PAL (Privileged Architecture Library) cycles, cache misses, branch mispredictions, and instruction types (load, store, branch, integer, and floating-point instructions). The events monitored on the memory interconnect include second-level cache misses (read and write), bus transactions, stalls, latencies, etc.

Table 1.
Cache/memory comparison

	AlphaServer	DEC
	8200	7000
CPU	21164	21064
Clock Frequency	300 MHz	200 MHz
On-chip Cache		
1st-level Dcache		
size	8 KB	8 KB
latency	6.6 ns (2 cy)	15 ns (3 cy)
bandwidth	4.8 GB/s	1.6 GB/s
1st-level Icache	8 KB	8 KB
2nd-level unified cache		
size	96 KB, 3-way	none
latency	20 ns (6 cy)	-
bandwidth	4.8 GB/s	-
Off-chip Cache		
size	4 MB	4 MB
latency		
reads	20 ns (6 cy)	25 ns (5 cy)
writes	16.7 ns (5 cy)	20 ns (4 cy)
bandwidth	970 MB/s	640 MB/s
Memory		
latency	253 ns	340 ns
	76 cycles	68 cycles
bandwidth	1.6 GB/s	800 MB/s

The event monitoring is non-intrusive because it is built into the hardware and does not require any special setup. The data collection is based on interrupt handling upon counter overflow (CPU monitors) or periodic counter polling (memory interconnect monitors), and the overhead is negligible. In addition to collecting event counts, the performance monitoring tools can be used to collect program counter and process status samples in a file, that can be later processed to obtain distribution of samples across code sections, routines, processes, modes (system or user), cache locations, etc. Hardware monitors allow all aspects of program execution, including system functions, to be monitored. The information collected provides insights into system behavior and makes these tools valuable for computer architects and engineers as well as application programmers.

The benchmark performance of the AlphaServer 8200 and the DEC 7000 is shown in Table 2. The AlphaServer 8200 achieves about 2.5 times the performance of the DEC 7000 on the SPEC benchmarks, and almost double on transaction processing (TP) workloads.

Table 2.
Benchmark performance comparison

	AlphaServer 8200	DEC 7000
SPECint92	341	133
SPECfp92	513	201
Linpack (MFLOPS)		
100x100	140	43
1000x1000	411	156
Livermore Loops		
(Geom. Mean MFLOPS)	69	28
McCalpin Copy BW MB/sec	900	310
AIM-III (jobs/min)	16168	5054
Transactions Per Second	600 TPS	320 TPS

The performance improvement of AlphaServer 8200 compared to DEC 7000 comes from several factors. The micro-architecture of the 21164 is the biggest factor, but it is complemented by the low cache/memory latency of the system platform and the ability of the compiler to generate optimized code for the quad-issue 21164:

Larger on-chip cache (96 KB, 3-way set-associative, instruction/data): the Scache reduces the number of off-chip misses by a factor of 3 to 6 times in SPEC92 and 1.7 times in TP workloads compared to 21064. The Scache misses are still high in the commercial workloads, indicating that the commercial workloads could benefit from larger caches. The primary I/D caches show a high miss rate. However, the advantage of primary caches is very low latency (2 or 3 cycles).

Lower cache/memory latencies, higher bandwidth: the AlphaServer 8200 improves memory latency/bandwidth compared to DEC 7000 (see McCalpin bandwidth in Table 2). This had no effect on SPEC92 performance (fits in the Bcache), but helped commercial workloads. The improvement in the Dcache latency (Table 1) was beneficial for all workloads.

Expanded multi-issue: Alpha 21164 doubles the number of floating-point and integer pipelines compared to DECchip 21064. This improves the multi-issuing time in Alpha 21164 2-3x compared to DECchip 21064. In the commercial workloads Alpha 21164 spends only 7% of the time multi-issuing because of higher stall time. The integer and commercial workloads do not benefit from triple/quad issuing (2 integer pipelines), while SPECfp92 spends on average 7% of the time in triple/quad issuing.

Reduction in the stall time: the time CPU is stalled is 20-30% lower in Alpha 21164 than in DECchip 21064. This is mainly a result of larger on-chip cache and lower latencies. The Alpha 21164 stall time is higher in the commercial workloads.

Less significant benefit in AlphaServer 8200 compared to DEC 7000 came from:

Miss Address File merging (load miss merging) of the 21164 provided 7-8% improvement in several SPECfp92 workloads (no effect in integer/commercial workloads).

Reduction in TB misses with the larger on-chip 48-entry ITB, 64-entry DTB reduced PAL time in SPEC92 (from 3-4% to 1%) and in the commercial workloads (from 15% to 14%).

The Alpha 21164 reduces the number of mispredicted branches compared to DECchip 21064 by using a 2-bit entry instead of a single history bit. However, the number of cycles spent on branch mispredictions did not change much in Alpha 21164 compared to DECchip 21064.

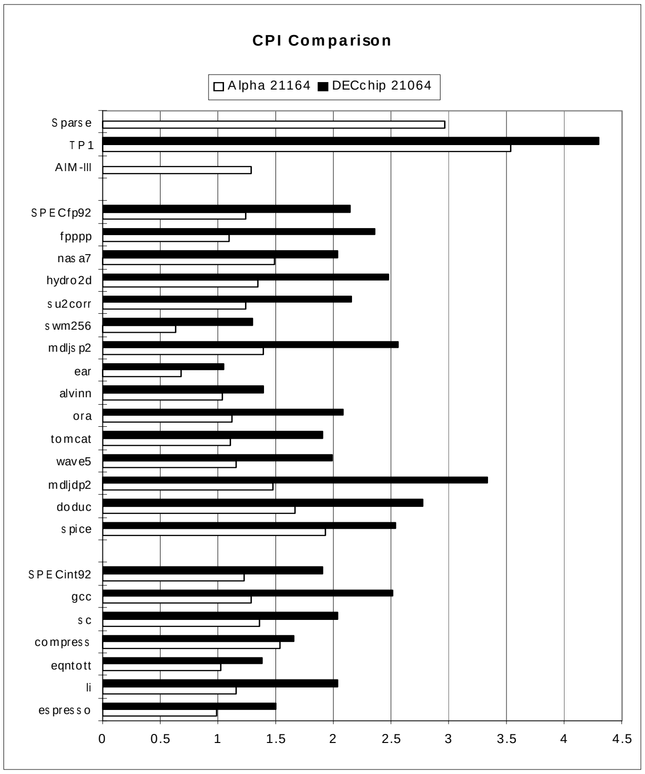

Figure 1. CPI comparison.

4. CPI

Figure 1 compares the cycles per instruction (CPI) for the AlphaServer 8200 and the DEC 7000 systems. The DEC 7000 measurements were done in 1994 with code scheduling optimized for the 21064 [3]. The AlphaServer 8200 results were obtained with code scheduled for the 21164 and also benefited from other generic compiler enhancements not included in the DEC 7000 code.

The Alpha 21164 achieves consistently lower CPI than the 21064 in spite of running at a 50% faster clock rate. The quad issue, lower latency pipelines, two-level cache architecture, and greater overlapping of memory accesses contribute to the lower CPI.

5. COMPILER EFFECTS

The Figure 2 shows the SPEC92 performance difference between the AlphaServer 8200 (based on the Alpha 21164) and the DEC 7000 (based on DECchip 21064). The figure also shows the software performance improvement in SPEC92 from the initial measurements to the final performance results. The major contributing factor was enhancements to the GEM compiler [2][8].

Code scheduling for Alpha 21164 targets for quad issue, dual FP pipelines, lower latencies, aligning instructions for better match of issuing and slotting rules. This provided up to 10% improvement in alvinn and ear (2% improvement in SPECfp92). Code scheduling had a much higher effect on the performance of vectorizable FP benchmarks than on integer benchmark performance.

Software pipelining (starting the loads for the next iteration during the current iteration) improved SPECfp92 performance by 6%. It had the most effect in the vectorizable floating-point benchmarks. The 21164 also implements cache prefetching. Loads to R31 and F31 can be used to prefetch data into the Dcache and Scache (R31 and F31 always read as zero). Prefetching was beneficial for some floating-point benchmarks, with majority of the benchmarks benefiting from scattering prefetches around the loop. Alternating loads and stores was beneficial for most of the floating-point benchmarks. Scheduling loads for the second level cache improved the performance of some benchmarks, but degraded some others (alvinn and swm256 degraded 28%). These software pipelining techniques provided 4% improvement in SPECfp92, they had much less effect on SPECint92 performance.

Speculative execution: starting the instructions that take a long time (loads) speculatively before the outcome of a branch is known. The most significant improvements were achieved in mdljsp2 (36%), mdljdp2 (23%), and li (6%).

Profile-based optimizations: gathering information about the run-time behavior of a program and providing such a profile to the compiler for the decisions on branch outcome, inlining, etc. This provided additional 3% improvement in SPECfp92.

Link time optimizations: re-arranging the routines for better cache usage (avoiding displacing the routines that map to the same cache block), re-scheduling instructions based on global address resolution [9]. The link-time optimizations improved the performance of some SPEC92 benchmarks by 4% to 14%.

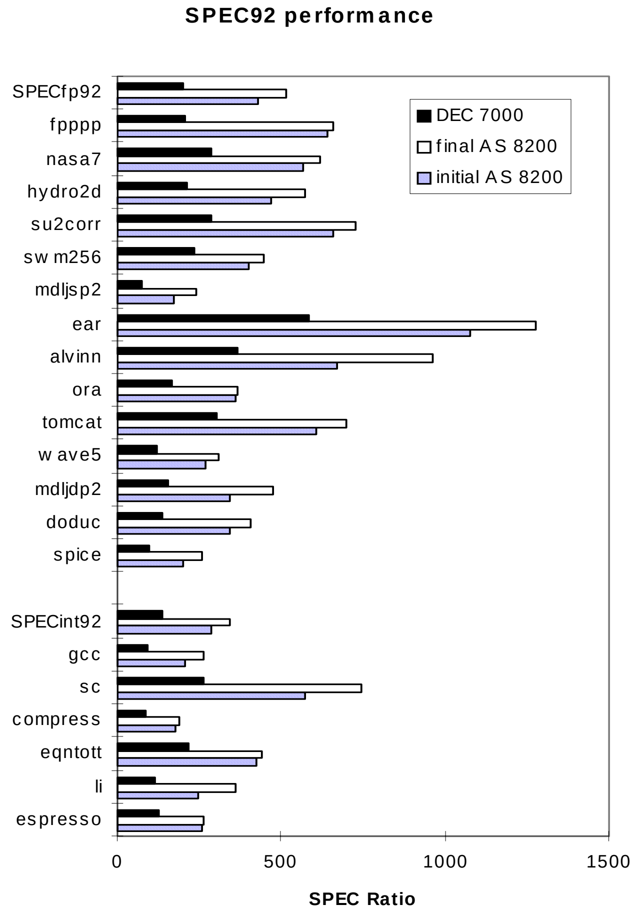

Figure 2. SPEC92 performance comparison and compiler/software improvements.

6. MULTIPLE ISSUE

The Alpha 21164 has 2 integer and 2 floating-point pipelines, and is capable of issuing up to 4 instructions simultaneously. The integer pipeline 0 executes arithmetic, logical, load/store, and shift operations. The integer pipeline 1 executes arithmetic, logical, load, branch/jump operations. The FP pipeline 0 executes add, subtract, compare, and FP branch instructions. The FP pipeline 1 executes multiply instructions.

The number of multiple-issued instructions relative to the total instructions increased in Alpha 21164 compared to DECchip 21064:

- SPECfp92: from 41% to 70%
- SPECint92: from 31% to 62%
- commercial: from 26% to 54%

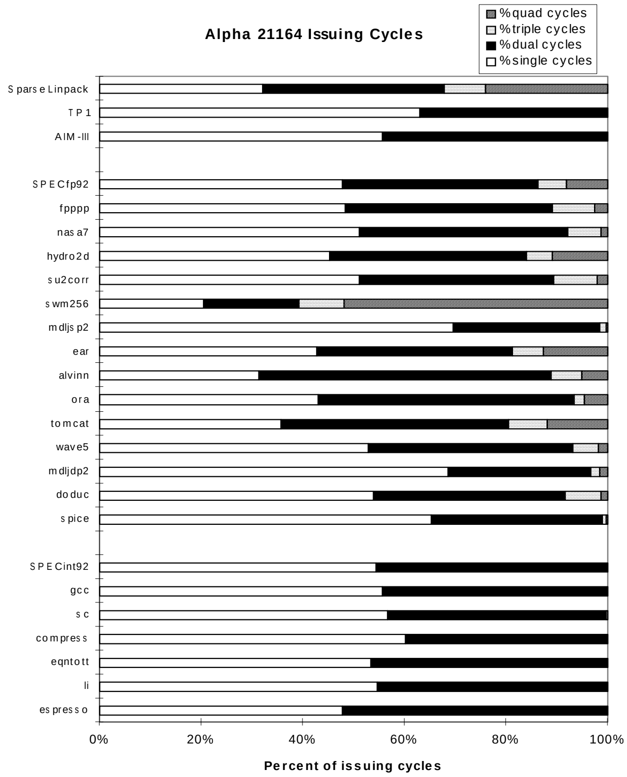

Figure 3. Distribution of issue cycles in Alpha 21164.

Figure 3 shows the percentage of issue cycles that are single, dual, triple, and quad issued on the 21164. A small number of floating point benchmarks take advantage of the ability to issue two integer instructions, and a floating multiply and add in the same cycle to achieve significant quad issue cycles. Dual issue accounts for almost half the issue cycles in many integer benchmarks. Figure 4 shows the percentage of dual and single issuing cycles in DECchip 21064.

The SPEC integer and commercial workloads (no FP operations) do not benefit from quad and triple issuing (2 integer pipelines). In floating-point workloads, 23% of all instructions are triple/quad issued.

The percentage of total time that the Alpha 21164 spends multiple issuing varies from 7% in commercial to 26% in SPEC92 as shown in Figure 5. The multiple-issuing time improves 2 to 3 times in Alpha 21164 compared to DECchip 21064:

- SPECint92: from 9% to 26%
- SPECfp92: from 11% to 26%
- commercial: from 4% to 8%

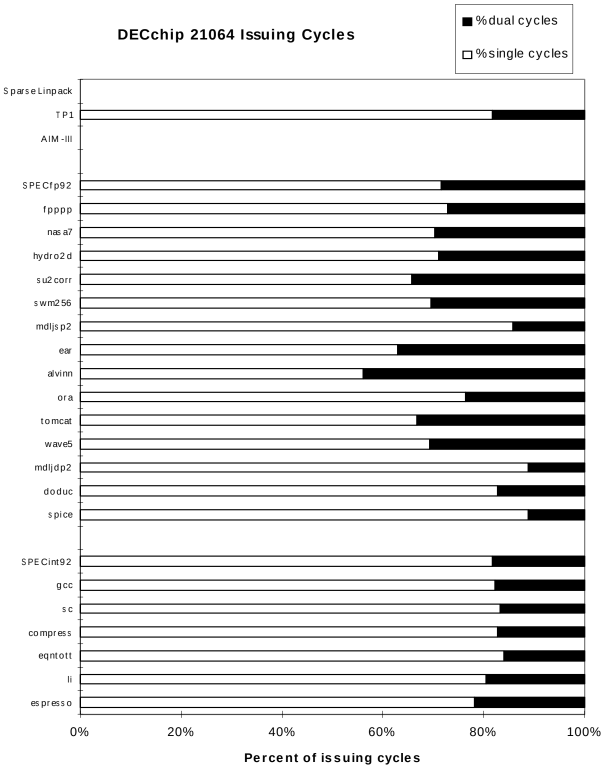

Figure 4. Distribution of issue cycles in DECchip 21064

The reason that multiple-issuing time is low in the commercial workloads is the high stall time (70% - 82% of the time Alpha 21164 is stalled in the commercial workloads).

The time spent on triple and quad issues is less than 5% in SPECfp92 (with the exception of ear and swm256), and none in SPECint92/commercial workloads.

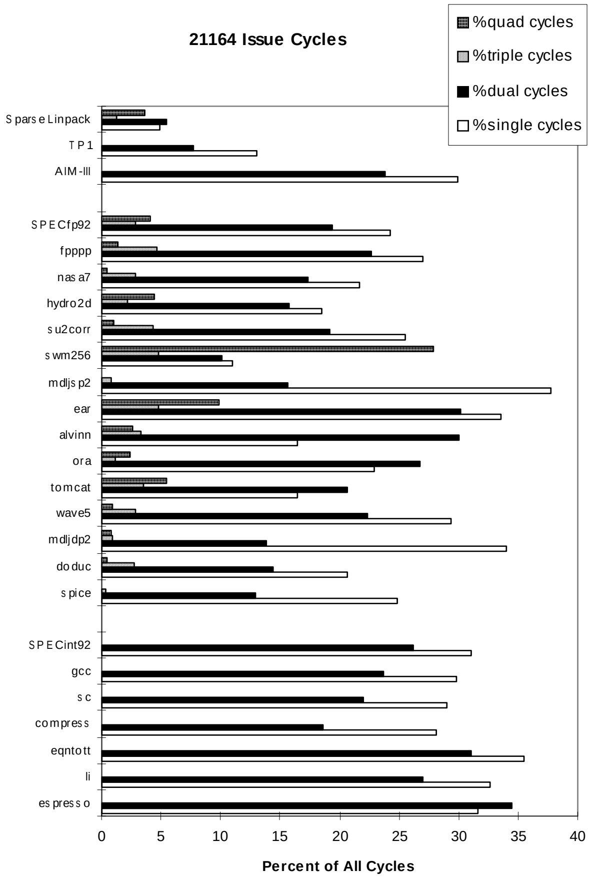

Figure 5. Issue cycles as a percentage of all cycles.

7. BRANCH PREDICTION

The Alpha 21164 keeps the outcome of branch instructions in a 2-bit history state (compared to 1-bit in DECchip 21064) for each Icache location, and uses it to predict execution of the next branch instruction.

Figure 6 shows the branch instructions that are mispredicted as a percentage of all instructions in Alpha 21164 and DECchip 21064. The number of mispredicted branches is reduced by close to a factor of 2 on Alpha 21164 compared to DECchip 21064. Most of the branches are conditional branches. The branch mispredicts are higher in integer workloads than in FP benchmarks.

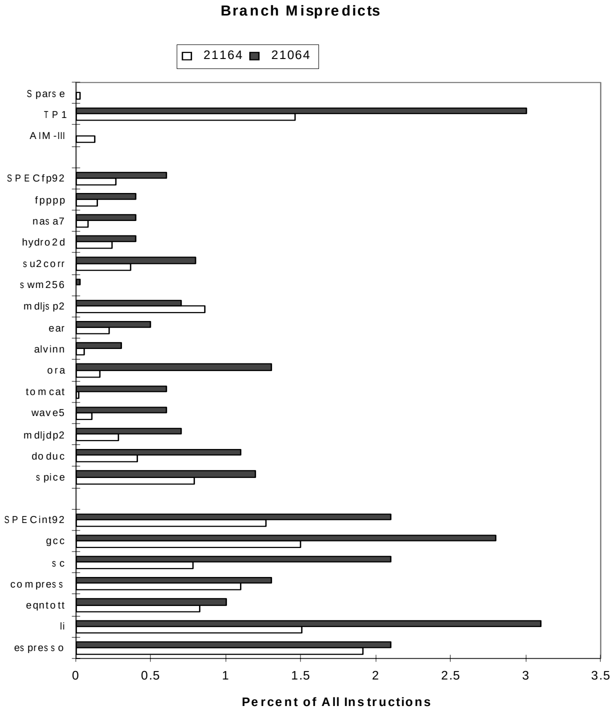

Figure 6. Mispredicted branches - 21064 vs. 21164.

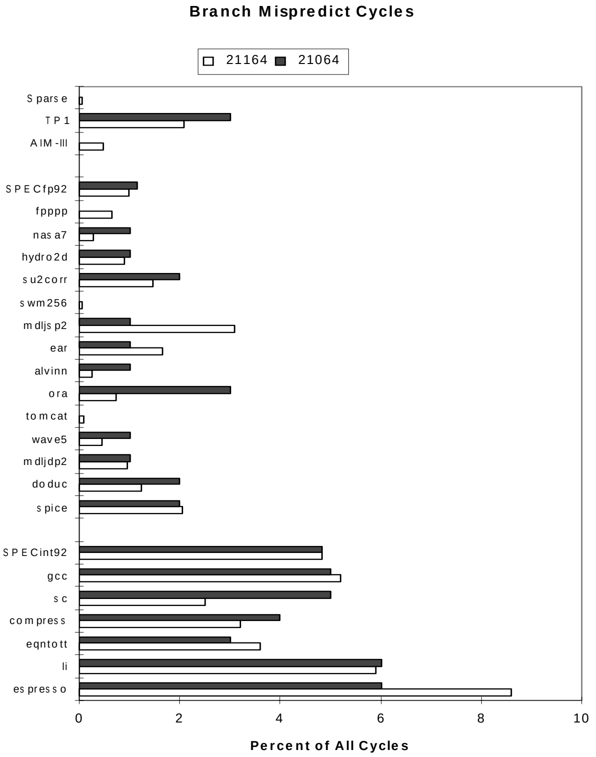

Figure 7. Branch mispredict cycles - 21164 vs. 21064.

The data in Figure 7 shows that the 21164 stall time due to branch mispredictions is higher in SPECint92 (4.8%) than in SPECfp92 (1%) and commercial workloads (2%). Even in integer and commercial workloads, branch misprediction stalls are insignificant compared to the other stall components.

The number of cycles spent on branch mispredictions is similar in DECchip 21064 and Alpha 21164. Although the number of branches mispredicted is lower in Alpha 21164 (Figure 6), the performance effect of other Alpha 21164 improvements (cache size and multiple issuing) is more dominant.

8. STALLS

The percentage of time that the CPU is not issuing instructions (stalled) varies significantly between technical and commercial workloads. Figures 8 and 9 show the measured components of the stalls as a percentage of the total execution time on the 21164 and 21064 respectively.

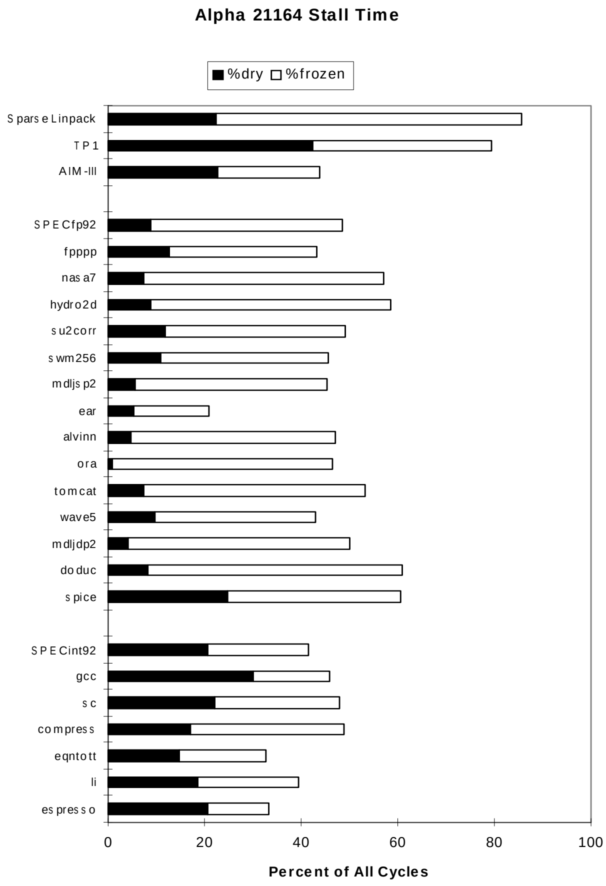

Figure 8. Alpha 21164 Stall time.

Dry stalls include I-stream stalls caused by the branch mispredicts, PC mispredicts, Replay traps, Istream misses and exception drain. Frozen stalls include D-stream stalls caused by the Dcache/Scache/Bcache misses as well as stalls caused by register conflicts and unit busy.

The Alpha 21164 reduces the performance penalty due to cache misses by implementing a large 96 KB instruction/data cache on chip. This cache (Scache) is 3-way set-associative and contains both instructions and data.

The Ibox contains a 4-entry prefetch buffer. The buffer allows prefetching of the next 4 consecutive cache blocks on an Icache miss (DECchip 21064 had 1-entry prefetch buffer). This reduces the penalty for Istream stalls.

The 6-entry Miss Address File (MAF) merges loads in the same 32-byte block. A 6-entry write buffer (compared to 4 entries in DECchip 21064) is used to reduce the store bus traffic and aggregate stores into 32-byte blocks [1][5][11].

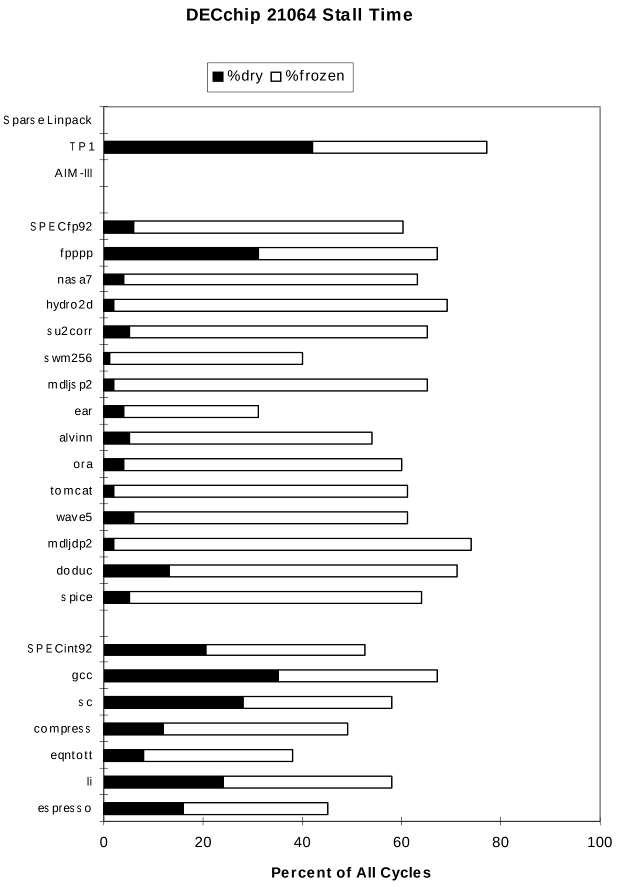

Figure 9. DECchip 21064 Stall time.

The processor is stalled much more (78% of the time) in the commercial and large scientific workloads than in

SPECint92 (41%) and SPECfp92 (49%). The pipeline dry and frozen time is comparable in SPECint92 (21% each) and commercial workloads (40% each). The pipeline frozen time (40%) is 4x higher than the dry time (10%) in SPECfp92.

Compared to DECchip 21064, Alpha 21164 reduces stall time from 52% to 41% in SPECint92, and from 60% to 49% in SPECfp92. The improvement is lower in commercial workloads.

9. CACHE MISSES

The off-chip misses (per 1000 instructions) on Alpha 21164 and DECchip 21064 are compared in Figure 10. The 96KB on-chip cache in 21164 significantly reduces the number of misses compared to 21064: 5x in SPECfp92 and 4.4x in SPECint92. The improvement is lower in the commercial workloads (1.3x), indicating that those applications can take advantage of larger caches. The reduction in off-chip cache misses is one of the major contributors to the performance improvements on 21164 vs. 21064.

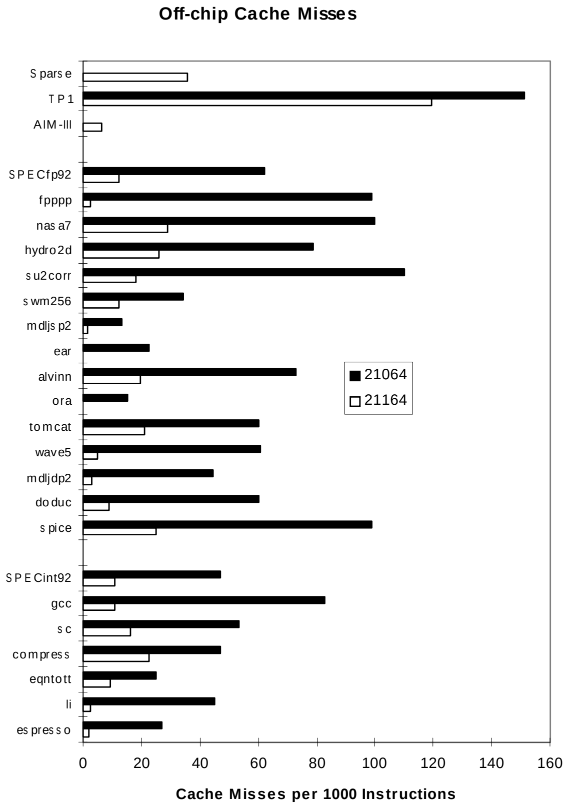

Figure 10. Off-chip cache miss comparison.

The number of the cache misses (per thousand Alpha instructions) in various levels of the cache hierarchy of the AlphaServer 8200 is shown in Figure 11.

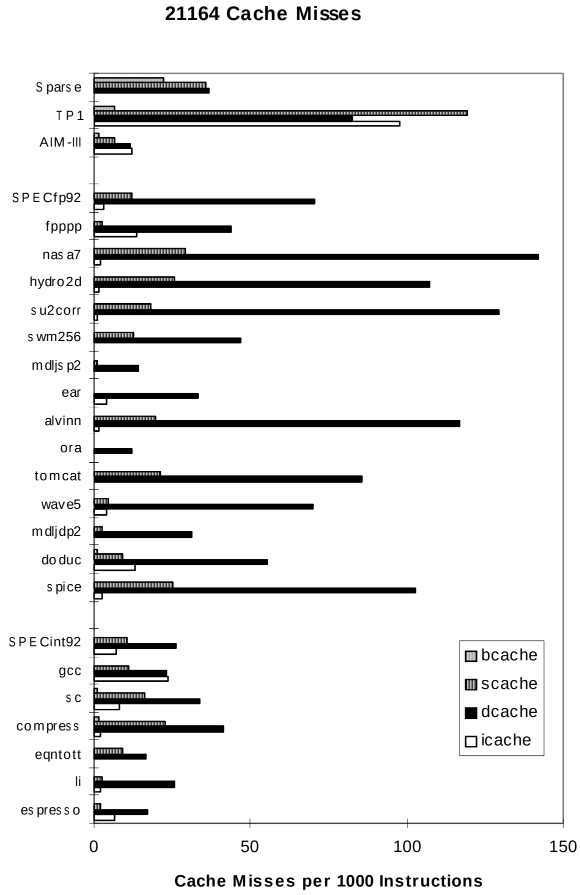

Figure 11. Cache misses in 21164.

The number of Icache misses is much higher in the TP workloads than in the SPEC benchmarks. The I-stream misses in commercial workloads are caused by frequent branches, process context switches, PAL traps, as well as subroutine, Run Time Library, and system calls. It is less likely that the more aggressive prefetching of the next consecutive Istream block will be beneficial for reducing Istream miss rate. Instead, providing high-bandwidth and low latency cache/memory interconnects will provide more benefits in future designs.

The Icache misses are very low in SPEC floating-point benchmarks (with the exception of Fpppp and Doduc). These workloads mostly fit in the Icache.

The Dcache misses (Figure 11) are high in the TP workloads as well as in SPEC CFP92 benchmarks (50-

100 per 1000 instructions). Most of the workloads would benefit from larger Dcache.

Note that Scache is not as beneficial in large technical and commercial applications as in SPEC92 benchmarks. Several floating point benchmarks incur 60 to 140 Dcache misses per thousand instructions. A large majority of these Dcache misses hit in the Scache, as indicated by the significantly lower Scache miss rates. The TP workloads show higher Scache misses than Dcache and Icache misses. These workloads have poor locality and result in high Icache and Dcache misses. The Scache receives both the Dcache and Icache misses and it is likely that Istream and Dstream traffic victimizes each other. The data indicates that about half the misses are served by the Scache.

The number of Bcache misses is negligible in SPEC92, very high in Sparse Linpack: (1 in every 45 instructions), and high in commercial workloads (1 in every 75 instructions).

Commercial applications do not fit well even in large 4 MB caches. Furthermore, they also exhibit poor locality of access. Instead, high-bandwidth and low latency cache/memory interconnects may provide better performance for commercial applications in future designs.

10. TIME ALLOCATION MODEL

A simple model that can be used to analyze the effect of the stall components is presented below. The total execution time is divided into two components: the compute component (where the CPU is issuing instructions) and the stall component (where the CPU is stalled). The stall component is further divided into the Dry and Frozen stalls:

time = compute + stall
stall = Dry + Frozen
Dry = Branch Mispredict + PC Mispredict + Replay Traps
+ Istream Miss + Exception Drain Stalls
Frozen = Dcache miss + Scache Miss + Bcache Miss
+ Register Conflicts and Unit Busy

Figure 12 shows the estimated percentage of the time in various stall components according to the model above. The remaining time (up to 100%) is the component where CPU is issuing instructions (not stalled). The branch and PC mispredicts affect the performance of SPEC integer workloads, and have little effect on the performance of commercial and SPEC FP workloads.

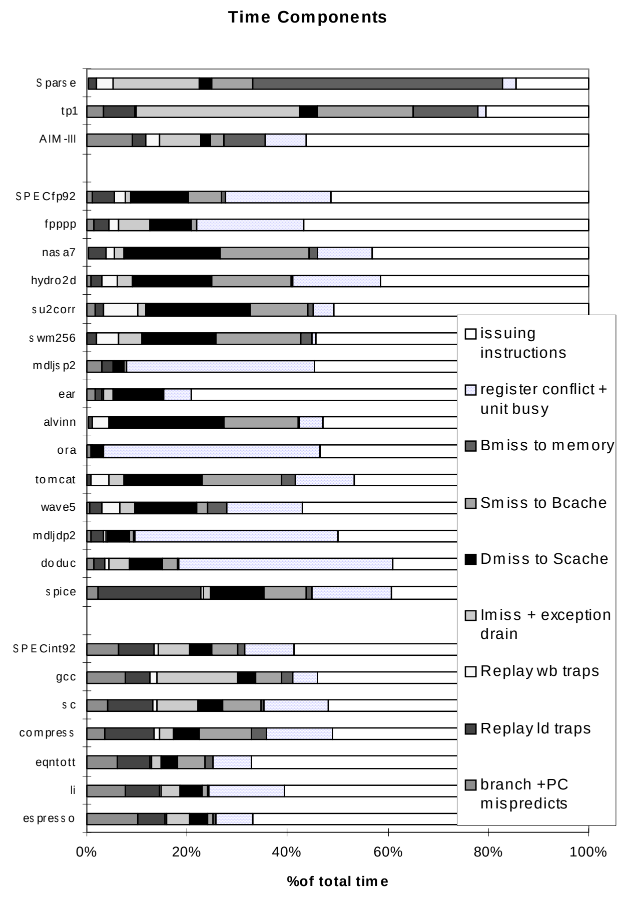

Figure 12. Time components in 21164.

The Replay traps stalls are caused mostly by (1) write-buffer traps: a full write buffer when a store instruction is executed or a full MAF when a load instruction is executed, (2) load traps: speculative execution of an instruction that depends on a load instruction that misses in the Dcache [11]. All workloads are affected by the Replay traps stalls (up to 20% in Spice). The integer SPEC and commercial workloads are affected mostly by the load traps, while FP SPEC benchmarks are affected by both load and write-buffer traps. Note that the time spent on a load Replay trap is overlapped with the load-miss time.

The cache (Scache and Bcache) stalls are high in commercial workloads, where the stall time is dominated by cache latencies. Several SPECfp92 benchmarks that do not fit in the Scache are affected by Bcache stalls (nasa7, hydro2d, su2cor, swm256, alvinn, tomcatv). The Bcache stalls have little effect on ora, mdljsp2, mdljdp2, li and espresso (fit in the on-chip caches).

Note a high stall time waiting for data from memory (40%) in Sparse Linpack (representative of large scientific applications). The memory component is high in commercial workloads (20%), and negligible in SPEC92 benchmarks (fit in the on-board cache).

This model, based on detailed measurements, can be an effective tool for evaluating the performance impact of various components on the overall system design. System architects can vary parameters like cache or memory access time, or cache size, and adjust the appropriate stall component to predict performance of alternative designs without going into detailed and often time-consuming architectural simulations.

11. PAL TIME

The Alpha 21164 significantly reduces the total time in PAL compared to DECchip 21064, as shown in Figure 13. Most of the PAL time is spent on TB misses. The Alpha 21164 benefits significantly from its 64-entry DTB and 48-entry ITB, compared to the 32-entry DTB and 8-entry ITB on the DECchip 21064.

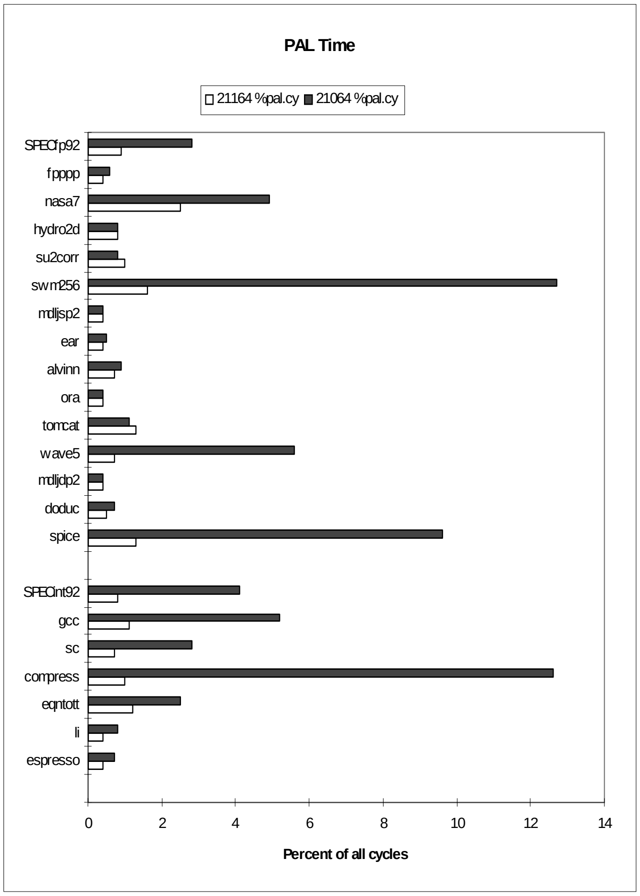

Figure 13. Percentage of time in PAL.

12. INSTRUCTION SET USAGE

Figure 14 shows the instruction mix for the new code that is not only optimized for the Alpha 21164 but also includes other generic compiler enhancements. The Alpha instructions are grouped into the following classes: Load (both floating-point (FP) and integer), Store (both FP and integer), Integer (all integer instructions excluding ones with only R31 or literal as operands), Branch (all branch instructions including unconditional), and FP (except FP loads and stores). Figure 14 shows the percentage of instructions in each class relative to the total number of instructions executed.

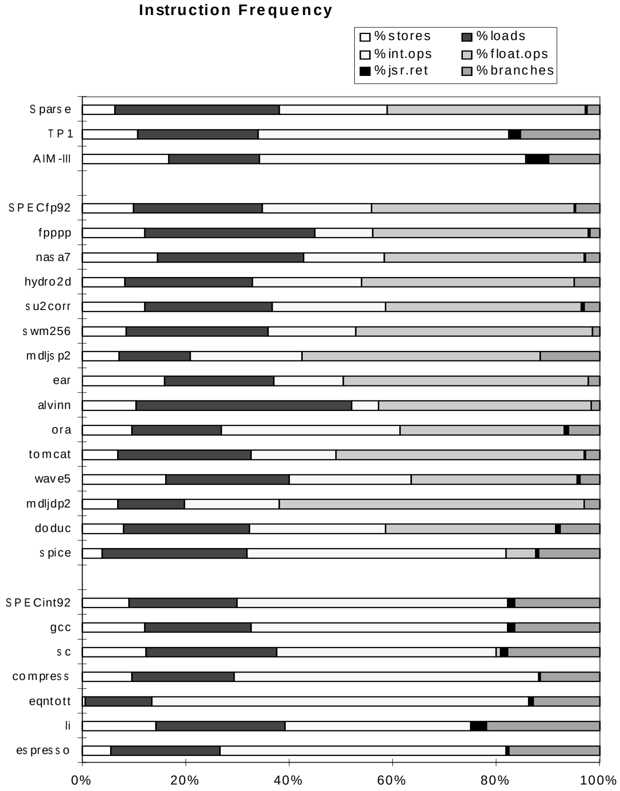

Figure 14. Instruction set usage distribution.

Both technical and commercial workloads make heavy use of memory operations: Load/Store instructions account for 20% - 50% of all instructions (the highest are Alvinn, Li, and Fpppp with close to 50%). Commercial and integer SPEC workloads have higher percentage of Branch instructions than the floating-point SPEC workloads. The number of floating-point operations in commercial and integer SPEC workloads is negligible. The integer instructions represent most of the instructions in the commercial and integer SPEC workloads, but are also present in the floating-point SPEC workloads. Although the number of floating-point instructions is higher than the number of integer instructions in floating-point workloads, there are several cases where they are comparable (Doduc, Wave5, Ora). Spice has very few (6%) floating-point instructions.

13. CONCLUSIONS

Cache and memory system design, as well as compiler techniques that can manage the memory access patterns were recognized as major performance factors in the first implementation of the Alpha architecture [3]. The new implementation of the Alpha architecture addressed these issues and provided 2 to 3 times the performance of the previous generation. Since the design addressed stalls caused by cache misses, quad issuing provided additional benefit in the floating-point SPEC92 benchmarks. Quad issuing had little effect on the commercial performance, because these workloads do not contain floating point operations. However, commercial performance did benefit from being able to issue two integer instructions in the same cycle.

The AlphaServer 8200 complements the micro-architectural enhancements of the 21164 with lower cache and memory latencies and higher bandwidth. Improving the cache and memory bandwidth/latency further will provide the most benefit in future designs. Future processors can also benefit from increased overlap of outstanding memory requests with instruction execution.

In this paper, only a sample of commercial and technical workloads was selected for the analysis. The SPEC benchmarks do not generate a lot of operating system activity and fit in megabyte-sized caches. Real technical applications will have different characteristics. Therefore, further study is needed on a broader range of workloads and applications. The performance characteristics of multiprocessor systems for both commercial and technical workloads represents another interesting area for investigation.

Acknowledgments
The authors would like to thank to George Chrysos and Paula DeAngelis for the data collection and Brian Wibecan for the performance monitoring tools. Special thanks to David Fenwick from the AlphaServer 8200 team, Paul Rubinfeld, Pete Bannon, and John Edmondson from the Alpha 21164 team, and Rich Grove, Steve Hobbs, Bill Noyce, Kent Glossop, and Lucy Hamnett from the GEM compiler team. Dick Sites and Darrel Donaldson provided numerous comments and suggestions. The authors are grateful to Bob Supnik, Masood Heydari and Pauline Nist for supporting this work.

REFERENCES

[1] P. Bannon, and J. Keller, "Internal Architecture of Alpha 21164 Microprocessor", *Proceedings of COMPCON Spring 1995*, pp. 79-87.

[2] D. Bhandarkar, "Alpha Implementations and Architecture: Complete Reference and Guide" ISBN1-55558-130-7, *Digital Press*, 1995, Newton MA.

[3] Z. Cvetanovic and D. Bhandarkar, "Characterization of Alpha AXP Performance Using TP and SPEC Workloads", *The 21st Annual International Symposium on Computer Architecture*, April 1994, pp. 60 - 70.

[4] D. Dobberpuhl, *et. al*, "A 200-MHz 64-bit Dual-issues Microprocessor", *IEEE Journal of Solid-State Circuits*, Vol. 27, No. 11, November 1992, pp. 1555-1567.

[5] J. Edmondson, P. Rubinfeld, V. Rajagopalan, "Superscalar Instruction Execution in the 21164 Alpha Microprocessor", *IEEE Micro*, Vol. 15, No. 2, April 1995.

[6] D. M. Fenwick, D. J. Foley, S. R. VanDoren, "Enterprise AlphaServer System", *COMPCON*, March 1995, pp.95-100.

[7] J. Gray (ed.), "The Handbook for Database and Transaction Processing Systems", *Morgan Kauffman*, San Mateo, 1991.

[8] R. B. Grove, D. S. Blickstein, K. D. Glossop, W. B. Noyce, "GEM Optimizing Compilers for Alpha AXP Systems", *COMPCON*, March 1993, pp. 465-473.

[9] A. Srivastava and D. Wall, "Link-Time Optimization of Address Calculation on a 64-bit Architecture", *SIGPLAN'94 Conference on Programming Language Design and Implementation*, June 1994, pp. 49-60.

[10] "DECchip 21064-AA Microprocessor Reference Manual", *Digital Equipment Corporation*, Order No. EC-N0079-72, October 1992.

[11] "Alpha 21164 Microprocessor Hardware Reference Manual", *Digital Equipment Corporation*, Order No. EC-QAEQA-TE, September 1994.

[12] "UNIX System Price Performance Guide", *AIM Technology*, Fall 1995.