Performance Characterization of the Pentium® Pro Processor

Dileep Bhandarkar and Jason Ding
Intel Corporation
Santa Clara, California, USA

Abstract

In this paper, we characterize the performance of several business and technical benchmarks on a Pentium® Pro processor based system. Various architectural data are collected using a performance monitoring counter tool. Results show that the Pentium Pro processor achieves significantly lower cycles per instruction than the Pentium processor due to its out of order and speculative execution, and non-blocking cache and memory system. Its higher clock frequency also contributes to even higher performance.

Keywords: Pentium® Pro processor, computer architecture, performance evaluation, workload characterization, out of order execution, speculative execution, SPEC CPU95, SYSmark/NT.

1. Introduction

The Intel Pentium® Pro processor was disclosed in February 1995 at ISSCC [1] and began shipping later that year. The micro-architecture implements several new features that are not found in previous implementations of the Intel Architecture. This paper analyzes the major performance characteristics of several business and technical benchmarks on a Pentium Pro processor based system. Measurements were performed using the built-in performance counters of the processor. Results are presented for cycles per instruction, cache miss statistics, branch prediction statistics, speculative execution, stall cycles, and other micro-architecture features.

Current literature contains numerous papers that present simulations of various machine structures. Often these simulations do not model the entire machine accurately or only use traces of parts of popular benchmarks. We present measured characteristics of a recent microprocessor to allow researchers to calibrate their theoretical results. The paper presents a lot of raw data and some analysis wherever possible. In a modern superscalar out-of-order processor, it is not always possible to derive precise cause-effect relationships.

Some of the results presented here are consistent with the behavior of SPEC benchmarks on other architectures, e.g., the FP benchmarks have lower Icache misses and higher Dcache misses than the integer benchmarks. Other measurements (branch mispredicts, micro-op statistics, and speculative execution) provide insight into the inner workings of the Pentium Pro processor.

2. Architectural Features of the Pentium® Pro Processor

The Intel Pentium Pro processor implements dynamic execution using an out-of-order, speculative execution engine, with register renaming of integer, floating point and flags variables, multiprocessing bus support, and carefully controlled memory access reordering. The flow of Intel IA-32 Architecture instructions is predicted and these instructions are decoded into micro-operations (uops), or series of uops. These uops are register-renamed, placed into an out-of-order speculative pool of pending operations, executed in dataflow order (when operands are ready), and retired to permanent machine state in source program order. This is accomplished with one general mechanism to handle unexpected asynchronous events such as mispredicted branches, instruction faults and traps, and external interrupts. Dynamic execution, or the combination of branch prediction, speculation and micro-dataflow, is the key to its high performance.

Figure 1 shows a block diagram of the processor. The basic operation of the microarchitecture is as described in the ISSCC paper [1]:

1. The 512 entry Branch Target Buffer (BTB) helps the Instruction Fetch Unit (IFU) choose an instruction cache line for the next instruction fetch. Icache line fetches are pipelined with a new instruction line fetch commencing on every CPU clock cycle.

2. Three parallel decoders (ID) convert multiple Intel Architecture instructions into multiple sets of uops each clock cycle. Instructions that require more than 4 uops are handled by the microinstruction sequencer.

3. The sources and destinations of up to 3 uops are renamed every cycle to a set of 40 physical registers by the Register Alias Table (RAT), which eliminates register re-use artifacts, and are forwarded to the 20-entry Reservation Station (RS) and to the 40-entry ReOrder Buffer (ROB).

4. The renamed uops are queued in the RS where they wait for their source data - this can come from several places, including immediates, data bypassed from just-executed uops, data present in a ROB entry, and data residing in architectural registers (such as EAX).

5. The queued uops are dynamically executed according to their true data dependencies and execution unit availability (integer, FP, address generation, etc.). The order in which uops execute in time has no particular relationship to the order implied by the source program.

6. Memory operations are dispatched from the RS to the Address Generation Unit (AGU) and to the Memory Ordering Buffer (MOB). The MOB ensures that the proper memory access ordering rules are observed.

7. Once a uop has executed, and its destination data has been produced, that result data is forwarded to subsequent uops that need it, and the uop becomes a candidate for "retirement".

8. Retirement hardware in the ROB uses uop timestamps to reimpose the original program order on the uops as their results are committed to permanent architectural machine state in the Retirement Register File (RRF). This retirement process must observe not only the original program order, it must correctly handle interrupts and faults, and flush all or part of its state on detection of a mispredicted branch. When a uop is retired, the ROB writes that uop's result into the appropriate RRF entry and notifies the RAT of that retirement so that subsequent register renaming can be activated. Up to 3 uops can be retired per clock cycle.

The Pentium Pro processor implements a 14-stage pipeline capable of decoding 3 instructions per clock cycle. The in-order front end has 8 stages. The out-of-order core has 3 stages, and the in-order retirement logic has 3 stages. For an integer op, say a register-to-register add, the execute phase is just one cycle. Floating point adds have a latency of 3 cycles, and a throughput of 1 per cycle. FP multiply has a latency of 5 cycles and a repetition rate of 1 every 2 cycles. Integer multiply has a latency of 4 cycles and a throughput of 1 every cycle. Loads have a latency of 3 cycles on a Dcache hit. FDIV is not pipelined; it takes 17 cycles for single, 32 cycles for double, and 37 cycles for extended precision. The processor includes separate data and instruction L1 caches (each of which is 8KB). The instruction cache is 4-way set associative, and the data cache is dual ported, non-blocking, 2-way set associative supporting one load and one store operation per cycle. Both instruction and data cache line sizes are 32 byte wide. More details of the microarchitecture can be found elsewhere [2].

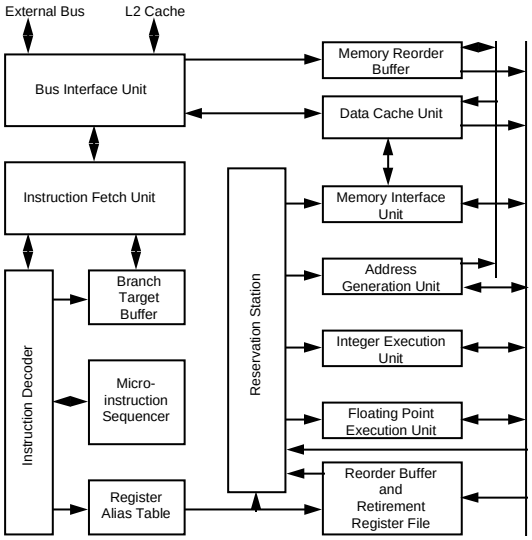

Figure 1 Pentium® Pro Processor Block Diagram

The secondary cache (L2 cache), which can be either 256KB or 512KB in size, is located on a separate die (but within the same package). The L2 cache is 4-way set associative unified non-blocking cache for storage of both instructions and data. It is closely coupled with a dedicated 64-bit full clock-speed backside cache bus. The L2 cache line is also 32 bytes wide. The L2 cache fills the L1 cache in a full frequency 4-1-1-1 cycle transfer burst transaction. The processor connects to I/O and memory via a separate 64-bit bus that operates at either 60 or 66 MHz. The bus implements a pipelined demultiplexed design with up to 8 outstanding bus transactions.

3. Performance Monitoring Facility

The Pentium® Pro processor implements two performance counters[3]. Each performance counter has an associated event select register that controls what is counted. The counters are accessed via the RDMSR and WRMSR instructions. Table 1 shows a partial list of performance metrics that can be measured by selecting the two events to be monitored.

Table 1. Pentium® Pro Processor Counter based Performance Metrics

Performance Metric	Numerator Event	Denominator Event
Data references per instruction	DATA_MEM_REFS	INST_RETIRED
L1 Dcache misses per instruction	DCU_LINES_IN	INST_RETIRED
L1 Icache misses per instruction	L2_IFETCH	INST_RETIRED
ITLB misses per instruction	ITLB_MISS	INST_RETIRED
Istalls cycles per instruction	IFU_MEM_STALL	INST_RETIRED
L1 cache misses per instruction	L2_RQSTS	INST_RETIRED
L2 cache misses per instruction	L2_LINES_IN	INST_RETIRED
L2 Miss ratio	L2_LINES_IN	L2_RQSTS
Memory transactions per instruction	BUS_TRAN_MEM	INST_RETIRED
FLOPS per instruction	FLOPS	INST_RETIRED
UOPS per instruction	UOPS_RETIRED	INST_RETIRED
Speculative execution factor	INST_DECODED	INST_RETIRED
Branch frequency	BR_INST_RETIRED	INST_RETIRED
Branch mispredict ratio	BR_MISS_PRED_RETIRED	BR_INST_RETIRED
Branch taken ratio	BR_TAKEN_RETIRED	BR_INST_RETIRED
BTB miss ratio	BTB_MISSES	BR_INST_DECODED
Branch Speculation factor	BR_INST_DECODED	BR_INST_RETIRED
Resource stalls per instruction	RESOURCE_STALLS	INST_RETIRED
Cycles per instruction	CPU_CLK_UNHALTED	INST_RETIRED

Table 2. Basic Characteristics of Systems

Processor	Intel Pentium® Pro Processor	Intel Pentium® Processor
CPU Core Frequency	150 MHz	120 MHz
Bus Frequency	60 MHz	60 MHz
Data bus	64-bit	64-bit
Address bus	36-bit	32-bit
On-chip L1 cache	8 KB data, 8 KB instruction	8 KB data, 8 KB instruction
Off-chip L2 cache	4-way 256 KB	512 KB (Dell), 256 KB (Gateway)
L2 cache timing	4-1-1-1 @ 150 MHz CPU freq.	3-1-1-1 @ 60 MHz bus frequency
System Chip Set	82450GX/KX	82430FX
Memory timing	14-1-1-1 (4-way interleaving)	13-3-3-3 (Fast Page Mode DRAM)
(bus cycles)	14-2-2-2 (2-way interleaving)	13-2-2-2 (EDO DRAM)
	14-4-4-4 (no interleaving)	
Basic Pipeline	14 stages	5 stages
Superscalar	3-way	2-way
Execution units	5	3
Branch prediction	4-way 512 entry BTB,	4-way 256 entry BTB,
	4-bit history, 2 level adaptive	2-bit history
Execution model	Out of order	In order
Speculative Execution	Yes	No
McCalpin Streams	140 MB/sec (4-way interleaving)	82 MB/sec (Gateway 2000 P120)
Memory Bandwidth	128 MB/sec (2-way interleaving)	
	97 MB/sec (no interleaving)	
SYSmark/NT rating	497 (Digital Celebris* XL6150)	294 (Gateway 2000 P120)
SPECint95	6.08 (Intel Alder System)	3.53 (Dell Dimension XPS P120)
SPECfp95	5.42 (Intel Alder System)	2.92 (Dell Dimension XPS P120)

4. Comparing the Pentium® and Pentium® Pro Processors

This section compares the basic performance characteristics of the Pentium [4] and Pentium Pro processors. Table 2 compares the basic characteristics of these two processors. We chose the 120 MHz Pentium and the 150 MHz Pentium Pro processors because both are fabricated in the same 0.6µ technology and use a 60 MHz external bus. For the SPEC benchmarks, the Pentium system was a Dell Dimension XPS P120 with a 512KB pipelined burst L2 cache, and the Pentium Pro system was an Intel Alder system with a 150MHz Pentium Pro CPU with 256KB L2 cache and a 4-way interleaved memory.

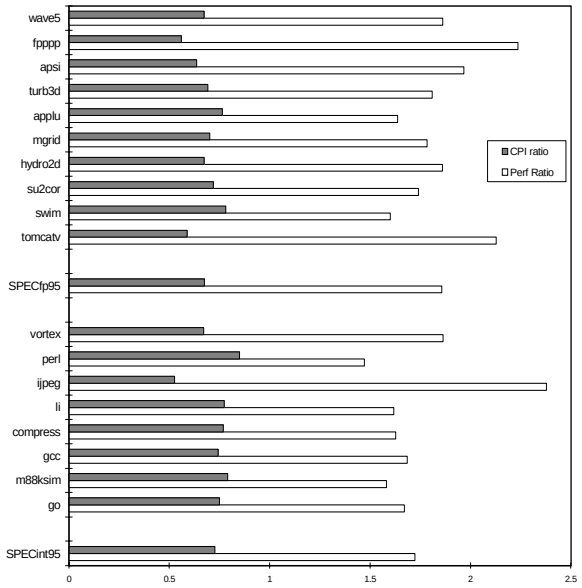

Figure 2 Performance Comparison of Pentium® and Pentium® Pro Processors on SPEC95

Figure 2 shows the SPECratios and the cycles per instruction of the Pentium Pro processor relative to the Pentium processor for the SPEC95 benchmark suite for the two systems. The SPEC* results were obtained using Intel Reference Compiler 2.3 Beta on UnixWare* v2.0 on an Intel Alder system. The Pentium Pro processor achieves CPIs 15% to 50% lower than the Pentium processor, in spite of the fact that it uses a design style that emphasizes a fast clock frequency. Designs that emphasize clock frequency generally result in deeper pipelines and longer CPI. The Pentium Pro processor design attempts to increase frequency while reducing CPI, without being overly focused on optimal CPI or fastest clock [5].

The Pentium Pro processor runs at 1.6 to 2.4 times the performance of the Pentium processor on the SPEC95 suite[6], achieving 70% higher SPECint95 and 85% higher SPECfp95. This performance comes from a 25% faster clock frequency and a 15 to 50% reduction in CPI compared to the Pentium processor. The Pentium Pro processor can issue up to 3 instructions every clock cycle, while the Pentium processor can issue only two. The out of order execution model of the Pentium Pro processor also allows useful work to proceed while prior operations are stalled, thereby lowering the CPI.

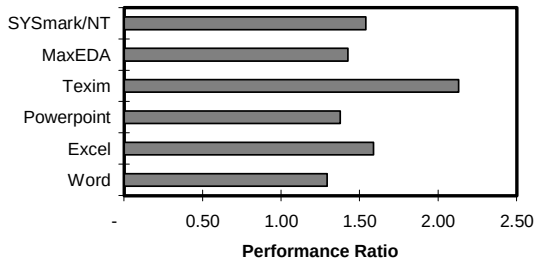

Figure 3 Performance Comparison of Pentium® and Pentium® Pro Processors on SYSmark/NT

Figure 3 shows the performance of a 150 MHz Pentium Pro processor based Digital Celebris 6150 compared to a 120 MHz Pentium processor based Gateway 2000 P120 system on the SYSmark* for Windows NT suite from BAPCO[7], which contains project management software (Welcom Software Technology Texim* Project 2.0e), computer-aided PCB design tool (Orcad MaxEDA* 6.0) and Microsoft Office* applications for word processing (Word* 6.0), presentation graphics (PowerPoint* 4.0), and spreadsheets (Excel 5.0). Both system had a 256KB L2 cache, but the Pentium Pro processor had a faster L2 cache (4-1-1-1 timing at full CPU clock frequency) on its dedicated L2 cache bus. The Pentium Pro processor runs 29% to 113% faster than the Pentium processor, with an overall 54% higher SYSmark score.

These results are slightly lower than the SPEC95 results because the desktop applications in the SYSmark benchmark perform some I/O operations that include wait times that do not scale with CPU performance. The SPEC benchmarks used compilers that generate binaries that are optimized for each target machine. The SYSmark/NT benchmarks use old binaries that are not optimized for the Pentium Pro processor. These benchmarks have large working set sizes for code and data and also contain many context switches. The SYSmark/NT benchmarks also result in higher L2 cache misses as shown in a later section.

5. Detailed Characterization of SPEC CPU95 Benchmarks

This section presents a detailed characterization of Pentium® Pro processor running the SPEC CPU95 suite. The performance counter measurements presented in the rest of this paper were done on a Digital Celebris XL6200 running Microsoft Windows* NT Workstation Version 3.51. The central processor in the Digital Celebris XL6200 is a 200MHz Pentium Pro processor with 256KB L2 cache. The Celebris XL6200 system that we used in our test was configured with 128MB DRAM with 2-way interleaving and 14-2-2-2 memory timing at 66 MHz bus frequency. The SPEC benchmarks were compiled with Intel FORTRAN and C Reference Compilers Version 2.3.

5.1 Cycles per Instruction

Figure 4 shows the cycles per instruction (CPI) for the SPEC95 benchmark suite. Several integer benchmarks achieve less than one cycle per instruction. The CPIs are remarkably low for a processor that implements a 14-stage pipeline. The low CPI is due to the overlapped out-of-order execution that mitigates the effect of the latency of individual operations, fast L2 cache, and adaptive two level branch prediction scheme. The FP benchmarks have higher CPI due to longer execution latencies and higher L2 cache misses.

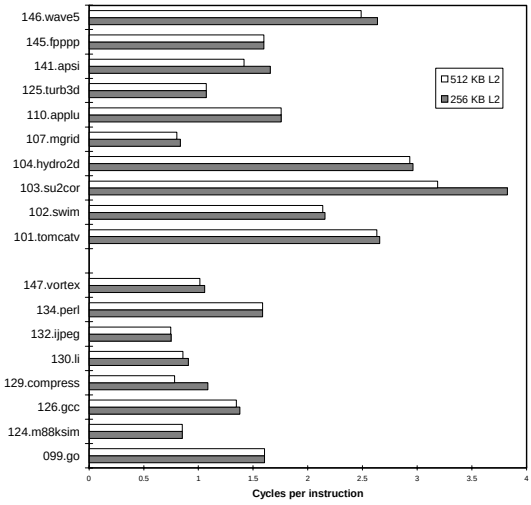

Figure 4 Cycles per Instruction

We measured the CPI for the processor with 512 KB L2 cache too, by replacing the CPU in the Digital Celebris system. Only compress (28%), li (5%) su2cor (17%), apsi (15%), and wave5 (6%) showed more than 5% improvement in CPI with the larger cache. The L2 miss ratio reduction was 85%, 89%, 40%, 48%, and 19% respectively. Simulations show that the CPI would decrease further for su2cor (25%), apsi (14%), and wave5 (5%) if the 4-way L2 cache is doubled again to 1 MB.

5.2 Instruction Decode

The Pentium Pro processor has 3 decoders that can handle up to 3 instructions every cycle (one instruction with up to 4 uops, and two single uop instructions)[5]. The decoder has a 6 uop queue at its output. Only 3 uops can be renamed per cycle, so the decoder has to stall if the queue is too full. Figure 5 shows the percentage of cycles in which 0, 1, 2, or 3 instructions were decoded. Benchmarks with high Icache or L2 misses show many cycles (35% to 51% for integer, 67% to 83% for FP) in which no instructions are decoded. During L2 misses, the CPU can run out of other machine resources causing back pressure on earlier pipe stages. On the integer benchmarks 33% to 54% of the instructions are decoded in cycles in which 3 instructions are decoded; 25% to 64% for FP benchmarks.

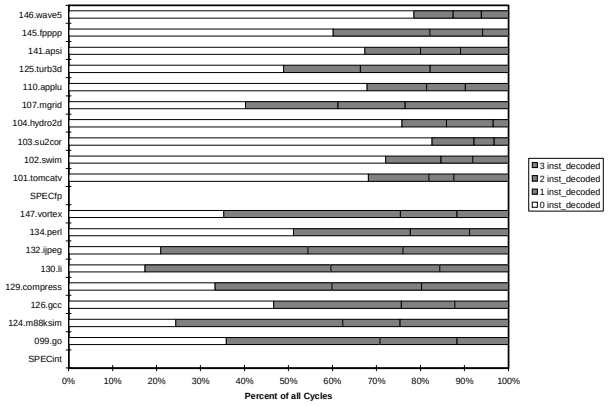

Figure 5 Instruction Decode Profile

5.3 Cache Misses

The L1 data cache can accept a new load or store every cycle and has a latency of three cycles for loads. It can handle as many as four simultaneously outstanding misses. Figure 6 shows the L1 data and instruction cache misses, and L2 cache misses. Except for gcc and m88ksim, the L1 data misses are always higher than the L1 instruction misses. In most cases the L1 instruction misses are so small that they don't even show on the scale used in Figure 6. The integer benchmarks, in general, show much lower L1 data cache and L2 misses than the floating point ones (larger data sets); but higher L1 instruction cache misses (larger code size and fewer loops). The benchmark (wave5) with the highest L1 misses does not have the highest L2 misses. Figure 7 shows a fairly strong

correlation between L2 misses and CPI, indicating that the L2 miss latency (about 50 CPU cycles) is not completely overlapped.

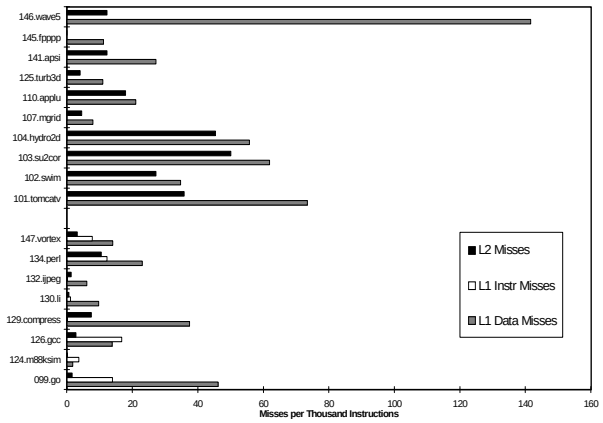

Figure 6 Cache Misses per Thousand Instructions

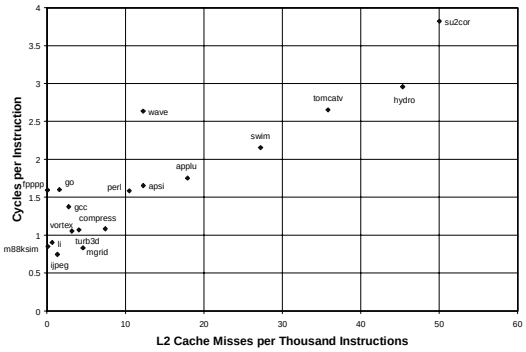

Figure 7 CPI versus L2 Cache Misses

5.4 TLB Misses

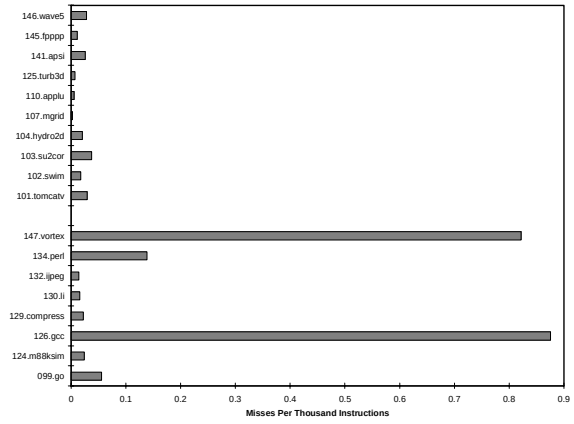

Figure 8 ITLB Statistics

The Pentium® Pro processor has separate TLBs for instructions and data. The processor also has separate TLBs for 4-Kbyte and 4-Mbyte page sizes. The ITLB for 4KB pages has 32 entries. The DTLB for 4KB pages has 64 entries. Both are 4 way set associative. The ITLB for large pages has 4 entries, while the DTLB has 8 entries; both are 4-way set associative. As shown in Figure 8, the ITLB misses are well below 0.1 per thousand instructions, except for a couple of integer benchmarks. The DTLB misses are generally higher than ITLB misses, but they could not be measured accurately. TLB misses do not contribute much to the CPI, as shown later.

5.5 Memory References

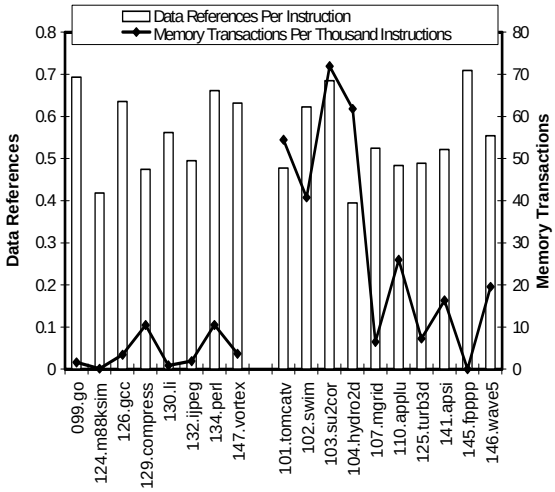

Figure 9 Memory Reference Statistics

Figure 9 shows the number of data references per instruction and the number of memory transactions per thousand instructions. On the average, both the integer and FP benchmarks generate about 1 data reference every two instructions. The IA-32 architecture results in more data references than most RISC architectures because it has fewer registers (8 vs. 32). As might be expected, there is a strong correlation between L2 cache misses and memory transactions. The memory transactions per instruction are higher for the FP benchmarks due to a higher L2 cache miss rate. Note that there can be more than one memory transaction per L2 cache miss if a dirty cache block has to be written back to memory.

5.6 Branch Prediction

The Pentium® Pro processor implements a novel branch prediction scheme, derived from the two-level adaptive scheme of Yeh and Patt[8]. The branch target buffer (BTB) retains both branch history information and the

predicted branch target address. The BTB contains 512 entries. If a branch is not found in the BTB, a static prediction (backwards taken, forward not taken) is used. There is no penalty for correctly predicted not-taken branches. Correctly predicted taken branches incur a 1 cycle penalty. Mispredicted branches incur a penalty of about 10-15 cycles, plus additional cycles required to retire the mispredicted branch.

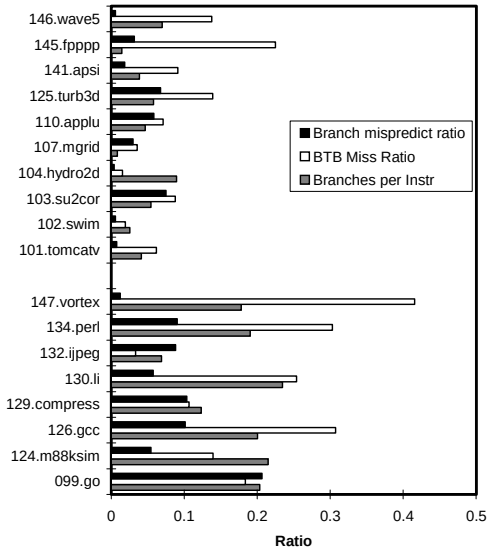

Figure 10 Branch Statistics

Figure 10 shows the frequency of branches, fraction of branches that hit in the BTB, and the accuracy of branch prediction. Even though the BTB miss ratio is fairly high, the branch mispredict ratio is less than 10% for all but one benchmark. The BTB miss ratio is high partly due to the fact that unconditional branches are not stored in the BTB, but are included in the total branch instruction count. As might be expected, the integer benchmarks contain more branches than the FP benchmarks, and they incur a higher branch mispredict ratio (fewer loop branches). The number of mispredicted branches range from about 2 to 40 per thousand instructions for the integer benchmarks, and about 0.1 to 4 for the FP benchmarks. For most of the benchmarks, branch mispredict stalls are not a major contributor to overall CPI.

5.7 Speculative Execution

The Pentium® Pro processor fetches instructions along the predicted path and executes them until the branch is resolved. If a branch is incorrectly predicted, the speculated instructions down the mispredicted path are flushed. Note that there can be other mispredicted branches down a mispredicted branch. Figure 11 shows the average number of instructions issued per retired instruction for the SPEC benchmarks. There are about 13 to 37 speculated instructions per mispredicted branch. Mispredicted branches are not recognized for about 10 to 15 cycles, and the processor can issue up to 3 instructions per cycle. Benchmarks with higher mispredicted branches per instruction have higher speculated instructions.

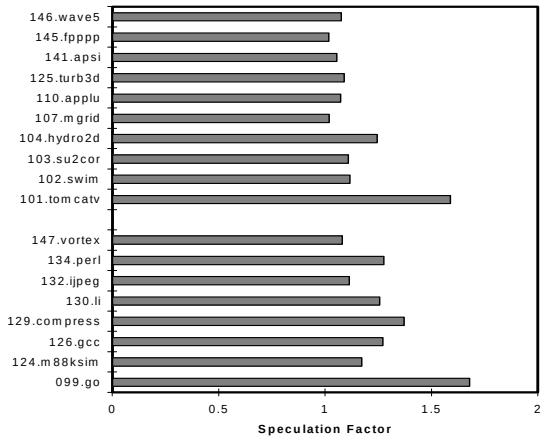

Figure 11 Speculation Factor

5.8 Resource Stalls

Figure 12 shows the I-stream stalls and resource stalls, measured in terms of the cycles in which the stall conditions occur. I-stream stalls are caused mainly by I-cache misses and ITLB misses. Resource stalls show the number of cycles in which resources like register renaming or reorder buffer entries, memory buffer entries, and execution units are full; but these stalls may be overlapped with the execution latency of previously executing instructions. The FP benchmarks, except for fpppp (long basic blocks), incur negligible I-stream stalls. They do incur significantly more resource stalls than integer benchmarks, probably due to long dependency chains.

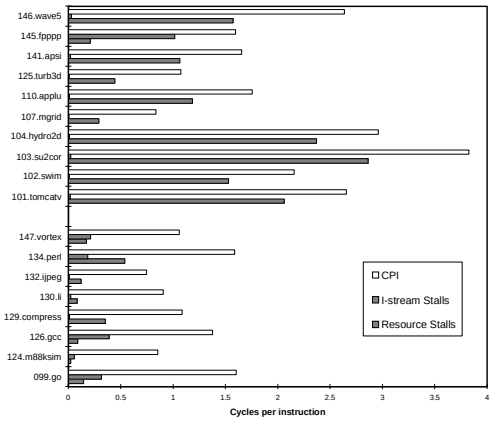

Figure 12 Stall Cycles

5.9 Micro-Operations

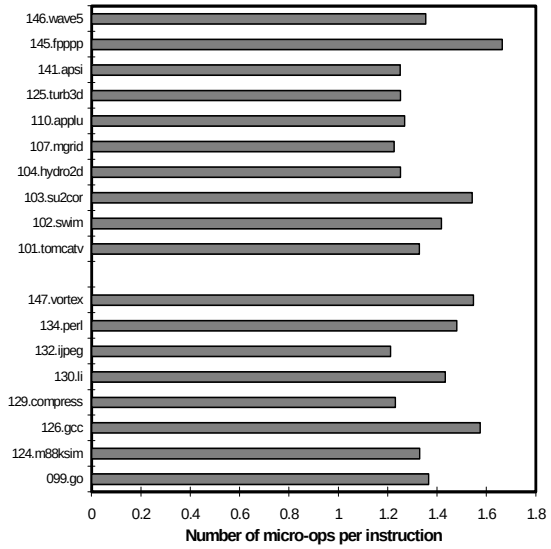

Figure 13 Micro-operations per Instruction

The instruction fetch unit fetches 16 bytes every clock cycle from the I-cache and delivers them to the instruction decoder. Three parallel decoders decode this stream of bytes and convert them into triadic uops. Most instructions are converted directly into single uops, some are decoded into one-to-four uops, and the complex instructions require microcode (sequence of uops). Up to 5 uops can be issued every clock cycle to the various execution units, and up to 3 uops can be retired every cycle. Figure 13 shows the average number of uops executed per instruction for each of the SPEC95 benchmarks. The range is from 1.2 to 1.7, with an average around 1.35.

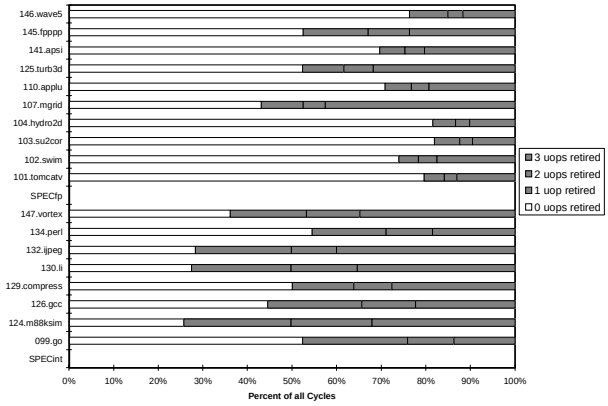

Figure 14 Micro-operations retirement profile

Figure 14 shows that no uops are retired in 25% to 55% of the cycles on the integer benchmarks, and 43% to 82% of the cycles for FP. Benchmarks with low CPI have fewer cycles with no uops retired. Furthermore, about 65% and 80% of the uops are retired in cycles in which 3 uops are retired for the average integer and FP benchmark respectively. This indicates that executed uops often have to wait for uops from previous instructions to be ready for retirement, thereby confirming the value of out of order execution. These younger uops build up more for FP benchmarks because of higher cache misses and longer latencies of FP operations.

5.10 Adding Up the Cycles

Accounting for cycles in an out-of-order machine like the Pentium® Pro processor is difficult due to all the overlapped execution. It is still useful to examine the various components of execution and stalls and compare them to the actual cycles per instruction as shown in Figure 15. The CPI is about 20 to 50% lower than the individual components due to overlapped execution. The figure also shows resource stall cycles in which some resource such as execution unit or buffer entry is not available. Execution can proceed during a resource stall cycle in some other part of the machine. Since more than one uop can be dispatched in a cycle, the figure does not account for execution parallelism. Micro-ops seem to dominate in most integer benchmarks. Resource stalls, and L2 misses contribute the most to the CPI in the FP benchmarks. Branch mispredicts are not a major factor.

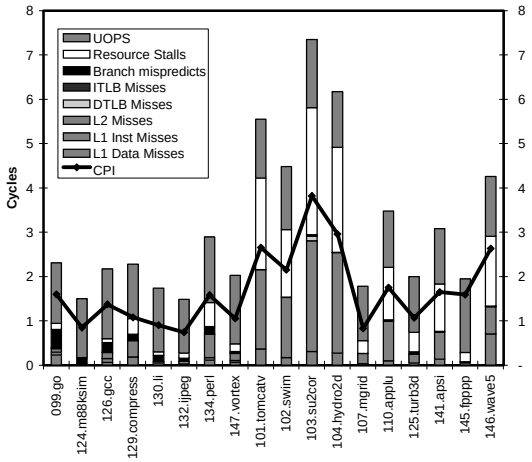

Figure 15 CPI vs. Latency Components

6. Characteristics Across Different Workloads

SPEC95 is a popular CPU intensive benchmark suite. It is widely used to characterize CPU performance. However,

the behavior of other workloads can be quite different. This section presents the characteristics of desktop applications running on a Pentium® Pro processor. In particular, we present results for the SYSmark/NT benchmark. These benchmarks are not floating point intensive; Excel contains about 9% FP instructions, MaxEDA has 4%, and the rest less than 0.5%. While the SPEC95 benchmarks were optimized for the Pentium Pro processor using the latest compilers, the SYSmark/NT benchmarks are based on old binaries that have been shipping for many years and were probably not generated with all optimizations turned on.

In this section, we compare the SYSmark/NT benchmark statistics with the minimum, median, or maximum for the SPECint95 and SPECfp95 suites. The data presented here shows that the SPEC integer benchmarks should not be used to predict the performance of real business applications

Figure 16 shows the CPI across different workloads. The business applications (using old binaries with non-optimal code) incur higher CPIs than the median for the SPECint95 benchmarks. Two of the five SYSmark/NT benchmarks incur higher CPI than the median observed for the SPECfp95 suite. The CPI is higher due to higher L2 miss rates, Istream stalls, and resource stalls.

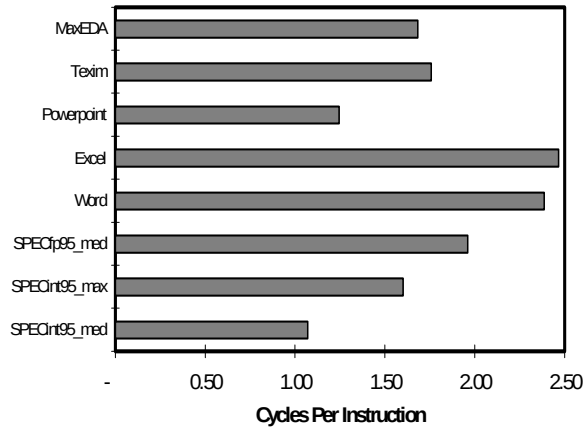

Figure 16 CPI for SYSmark/NT

Figure 17 shows the L2 cache misses. The three Microsoft Office benchmarks incur much higher L2 misses than the SPECint95 median, but well below the SPECfp95 median. The code and data sizes of these business applications are much larger than the SPEC integer benchmarks. Once again, there is fairly strong correlation between L2 misses and CPI. Overall, Word and Excel exhibit the highest L2 misses and stall cycles among the SYSmark/NT benchmarks.

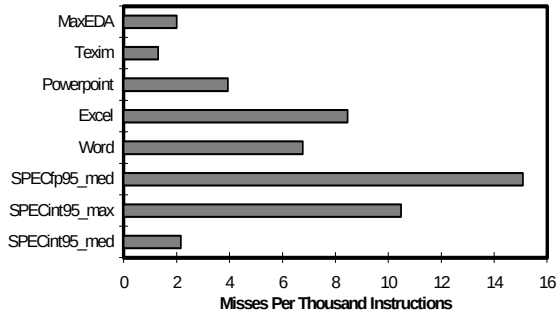

Figure 17 SYSmark/NT L2 Cache Misses

Figure 18 shows the resource stalls. The SYSmark/NT benchmarks incur higher resource stalls than the SPECint95 median, but are well below the SPECfp95 median. The higher resource stalls can be attributed to higher L2 misses during which the internal resources can be consumed by instructions waiting to be retired.

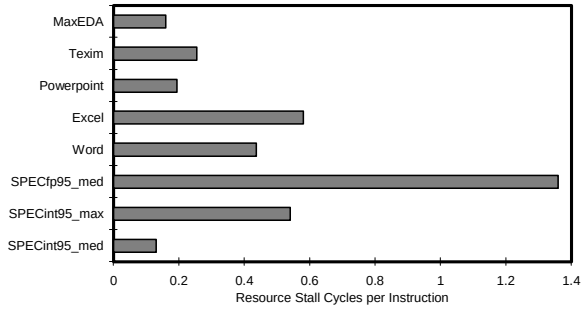

Figure 18 SYSmark/NT Resource Stalls

Figure 19 shows the instruction stalls. They are higher than the SPECint95 median. Once again, this is due to higher occurrence of string instructions in Word and Excel that invoke the microsequencer and require the decoders to stall. These workloads also have high context switch activity resulting in ITLB flushes and Icache misses.

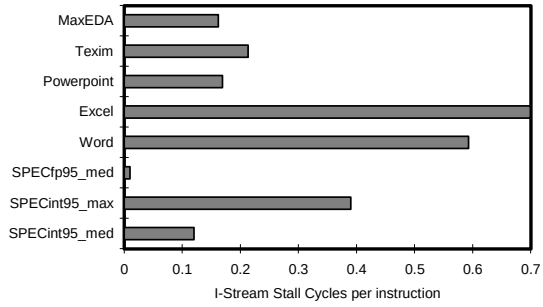

Figure 19 SYSmark/NT Instruction Stalls

Figure 20 shows the uops per instruction. All the SYSmark/NT benchmarks execute more uops than most

of the SPEC95 benchmarks. This is probably due to the higher use of character string instructions. There is a strong correlation between uops/instruction and CPI, a trend not observed in the SPEC95 suite.

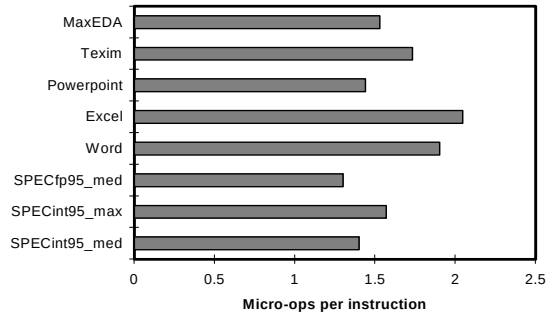

Figure 20 SYSmark/NT Micro-ops Per Instruction

Figure 21 shows the speculation factor. It is in the bottom half of the distribution for SPECint95. The speculation factor is lower because there are fewer mispredicted branches.

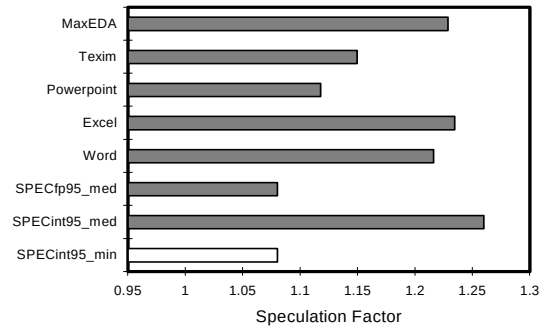

Figure 21 SYSmark/NT Speculation Factor

7. Concluding Remarks

The Pentium® Pro processor was designed to achieve significantly higher performance than the Pentium processor in the same process technology. It achieves this performance through a superpipelined design that yields a 25% faster clock, and with an out of order dynamic execution engine that reduces the CPI. The data presented here shows that the Pentium Pro processor achieves a 15 to 45% reduction in CPI compared to the previous generation design (Pentium processor) in the same process technology, while running at a 25% faster clock frequency. The processor's out-of-order, speculative execution engine does manage to overlap useful work with pending memory accesses to reduce the impact of cache misses. The impact of resource stalls is also reduced by out of order execution. The branch prediction scheme reduces branch mispredictions so as not to make them a significant performance limiter. It performs well even on old binaries that were not optimized for its microarchitecture. Performance counter based measurements show that the overall CPI achieved by the Pentium Pro processor is about 20 to 50% lower than the individual latency components due to overlapped execution.

A detailed comparison of the Pentium Pro processor and Digital's Alpha 21164 RISC processor is reported in another study [9].

8. Acknowledgments

The authors would like to thank the Pentium® Pro processor designers for producing such an interesting microprocessor and for incorporating the performance counting mechanisms that enabled this study. Special thanks to Bob Colwell for his extensive review and valuable comments.

9. References

[1] Robert P. Colwell and Randy L. Steck, "A 0.6um BiCMOS Processor with Dynamic Execution", ISSCC Proceedings, February 1995, pp. 176-177.

[2] Linley Gwennap, "Intel's P6 Uses Decoupled Superscalar Design", Microprocessor Report, Vol. 9, No. 2, 16 February 1995, pp. 9-15.

[3] Intel Corporation, "Pentium Pro Family Developer's Manual, Volume 3: Operating System Writer's Manual", Intel Corporation, Order Number 242692, 1996.

[4] Donald Alpert and Dror Avnon, "Architecture of the Pentium Microprocessor," IEEE Micro, June 1993, pp. 11-21.

[5] David Papworth, "Tuning The Pentium Pro Microarchitecture," IEEE Micro, April 1996, pp. 8-15.

[6] Jeff Reilly, "A Brief Introduction to the SPEC CPU95 Benchmark," IEEE-CS TCCA Newsletter, June 1996. Also, see http://www.specbench.org/osg/cpu95/.

[7] http://www.bapco.com/nt1.htm

[8] Tse-Yu Yeh and Yale Patt, "Two-Level Adaptive Training Branch Prediction," Proc. IEEE Micro-24, Nov 1991, pp. 51-61.

[9] Dileep Bhandarkar, "RISC versus CISC: A Tale of Two Chips," submitted for publication.
