A Comparison of Pentium® and Pentium® Pro Architectures

Martin Glynn

Christopher Hammond

CS237 Comparative Computer Architecture Professor Cherrice Traver Nov 15, 1996

Table Of Contents

Index to Figures, Tables, Charts

1. Abstract

The Intel Pentium and Pentium Pro processors are the latest in the *x*86 family of processors. How does the Pentium Pro achieve its performance improvements over the Pentium?

The Pentium features Intel's first superscalar architecture, of degree 2. It uses a 5-stage integer pipeline, and a separate floating-point pipeline and execution unit. A 256 entry branch target buffer is used with a dynamic history algorithm to achieve branch prediction accuracy of 80%.

The Pentium Pro is superscalar (degree 3) and superpipelined (14 stages). It features Dynamic execution, an on-board L2 cache, and a transactional external bus. A 512 entry branch target buffer and improvements to the dynamic history algorithm achieve branch prediction accuracy greater than 90%. Floating point stack latencies are eliminated.

Intel claims that the Pentium Pro runs twice as fast as the Pentium for 32-bit code running on a 32-bit operating system. We found that this was indeed possible with some applications, assuming processors using the same fabrication technology. For 32-bit applications with a 32-bit operating system and same speed clocks, the performance improvement was typically in the range or 35%-45%. 16-bit code severely impacted performance of the Pentium Pro. For some 16-bit applications with a operating system with 16-bit code, it was possible to find that the Pentium outperformed the Pentium Pro.

1. Introduction

In June, 1978, Intel introduced the 8086 processor. It had 16-bit registers, a 16-bit data bus, and used 29,000 transistors. It was the first of a series of Intel processors that have implemented basically the same instruction set (referred to as *x*86 throughout this document) in increasingly more sophisticated and higher performing hardware designs. The 8086 was followed by the 80286 processor, introduced in February, 1982, then the 80386 in October, 1985, and then the 486DX in April 1989. Most recently, Intel introduced the first of the Pentium processors in March, 1993, and the Pentium Pro in February, 1996. During this time period, the operating systems marketed by Microsoft Corp. have come to dominate the personal computer market, and this series of Intel processors has met with tremendous success, largely by supplying the processors for that same market. Microsoft's Windows operating system and Intel's *x*86 processors have become so closely tied in the public's mind that they have collectively been referred to as "Wintel".

This same time period has seen a vigorous development of many kinds of microarchitectures, and many theoretical investigations in hardware design. Intel has attempted to incorporate the very latest in performance enhancing features into their latest processor offerings, while maintaining the ability to support existing *x*86 based applications. We have investigated what ideas Intel has implemented, and with what success, by comparing their latest two processors, the Pentium, and the Pentium Pro.

1. How this report was developed

Our objective was to see how Intel, the manufacturer of the dominant line of processors in the PC market, was modifying their microarchitecture to improve the processors' performance. This report examines the latest two products, the Pentium and Pentium Pro processors. We planned to describe the features of each processor, than compare the differences, particularly looking at performance. Our preconception was to discuss each processor in the areas of datapath, control, and memory system.

Our first step was to accumulate what publicly available information we could find on the architectures and performance comparisons of the Pentium and Pentium Pro processors. This turned out to be strictly periodicals and the internet. We studied these materials and developed, independently, descriptions of one of the two processors, each. This involved coordinating and combining the various sources we had discovered, in some cases summarizing and in others expanding on that information. These became our first drafts of the sections **Pentium** and **Pentium Pro**. We discovered that the technical developments in each of the processors was primarily in the datapath, and therefore the majority of the processor descriptions are concerned with developments in the datapath. There was no information in the area of control on the Pentium Pro that we were able to find (see Appendix 3 for an E-mail response from Intel technical support about control information on the Pentium Pro). We also found that the types of technologies used to improve the product for each of the two processors was different. We therefore decided to not require that the organization of the two sections conform. Instead we set each in the form that we felt most clearly presented the critical developments for each processor.

From our processor description sections, and we were able to develop a summary of the differences in design. To this we added explanations, and performance test results, and summaries. This formed the **Comparison** section. The **Conclusion** represents our own ideas on how successful Intel was with the Pentium Pro. These two sections, and the remainder of the report, were developed through a series of revised drafts that we worked on together; sometimes collaboratively, and sometimes independently.

The report was developed using Microsoft Word version 7 (using a Pentium !!). The pictures and charts included here come directly from Intel's web site. While we don't have explicit permission from them, this report is not being used for profit, will not divert any potential revenue from Intel. We feel, therefore, that it is reasonable to use these images for this limited purpose. There is mention in the body of the document for each such image that it from Intel's web site. The other figures were developed by us in Microsoft Powerpoint version 7; tables were developed directly in Word. The ideas for the figures and tables are definitely derived from sources cited in the bibliography, but none is an exact duplicate of any one figure or table found there (tables 3 and 5 are extremely similar, however[8,9],and references are cited there).

1. Pentium®

1. *Overview*

The term "Pentium Processor" (aside from being a marketing term) refers to a family of micro processors that share a common architecture and instruction set. In naming the Pentium processor, Intel departed from the "x86" naming convention that was previously used. Since Intel could not trademark the numbers "586" (the natural progression from 386, 486) they decided to use the Greek word for 5 (pente) concatenated with a common ending for elements in the periodic table (ium). The first Pentium processors (called the P5) were introduced in 1993. The processor originally used a 5.0 volt power supply and used 0.8 micron bipolar complementary metal oxide semiconductor (BiCMOS) technology. Due to the power consumption, speed, and heat considerations, this design was abandoned for the P54C design. This design, introduced in 1994, (yes technology does change rapidly) uses a 3.3 volt power supply and a 0.6 micron BiCMOS technology. This design can run at speeds up to 133 MHz.

The processor has a System Management Mode (SMM) for advanced power management. SMM involves putting the processor into a low power state during non-processor intensive tasks. The processor also can be put into a very low power state when the computer is not in use (sleep mode). This mode provides intelligence such that various components can be slowed, suspended, or shut down to maximize energy savings.

1. ***The Pentium®*** Processor Data Path and Control

The Pentium Architecture is a pipelined architecture. This means that the architecture executes integer instructions in the following five stages : prefetch, decode 1, decode 2, execute, and writeback. This permits several instructions to be in various stage of execution. The five cycles can be described as follows:

1. Prefetch - During this cycle the CPU prefetches code from the instruction cache and aligns it, in an alignment buffer, to the initial byte of the instruction to be decoded. Two cache lines and an alignment buffer are needed because the variable length *x*86 instructions could start at any point in the first cache line and could overlap into the next line.
2. Decode 1 - The CPU decodes the instruction to generate a control word. A single control word executes an instruction directly. More complex instructions require microcode.
3. Decode 2 - The CPU decodes the control word generated in Decode 1 for use in the execution cycle. Additionally, the CPU generates addresses for data memory references.
4. Execute - The CPU either accesses the data cache or calculates results in the ALU, barrel shifter, or other functional units in the data path.
5. Writeback - The CPU updates the registers and flags with the instruction results. All exceptions must be resolved prior to performing this stage.

The previous processor (Intel486) was able to run many of its instructions in one clock cycle. As such, the pipelined architecture of the Pentium provides a natural progression from previous versions of the Intel architecture. Figure1 shows a block diagram of the Pentium processor. The core functionality of the processor is handled by the two integer pipelines and a floating-point pipeline with a dedicated multiplier, adder, and divider. Also note the separate data and instruction caches in conjunction with the branch prediction buffer. These caches are used to increase throughput of the Pentium by providing fast access to instructions and data with the branch prediction buffer augmenting the instruction cache for fast access to branch target addresses.

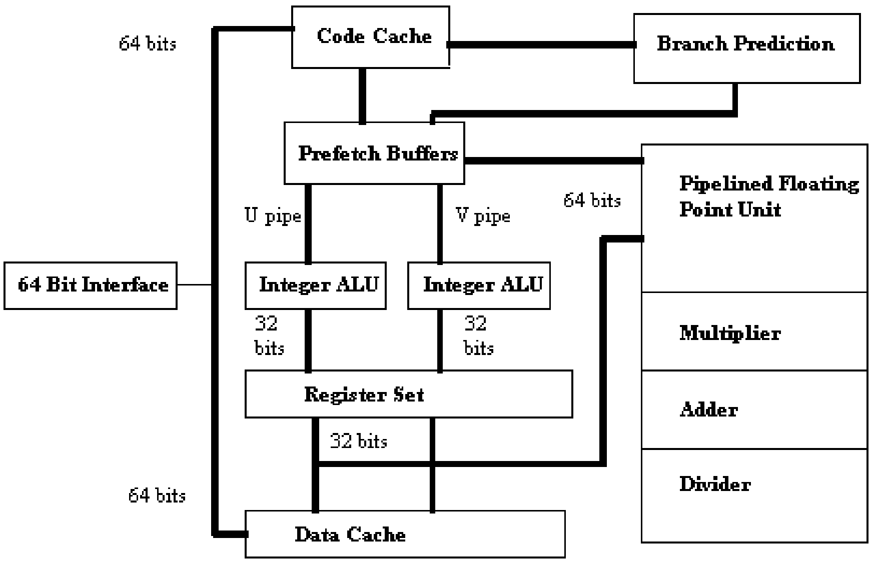


Figure 1. Pentium® Conceptual Block Diagram

The Pentium Architecture is described as being "superscalar". What this means is that the processor consists of more than one execution unit. Two pipelined execution units as well as a pipelined floating point unit are used to increase performance. This means that the architecture can execute up to two instructions simultaneously as well as perform pipelined floating point calculations. The Pentium also uses hard-wired instructions to replace many of the microcoded instructions. Hard-wired instructions are simple, easy to use, and can be executed by the processor without the use of micro code. A performance gain is also realized by the enhanced micro code that uses the facilities of the dual-pipelined architecture.

1. *Integer Pipelines*

The Pentium data path contains two integer pipelines. These pipelines are (designated U and V) are fed by the prefetch and decode 1 stages of the data path. These stages can fetch and decode two simple instructions and submit them in parallel to the U and V pipelines. In addition, the decode 1 stage can generate microcode sequences to that control the U and V pipelines. Figure 2 shows the "superscalar" execution of the U and V pipes in block diagram form.

Since instructions are executing in parallel, a method must be provided to avoid simultaneously writing to the same register. To eliminate these interdependencies, the following algorithm is used (the instruction issue algorithm):

Decode two consecutive instructions (I1 and I2)

---

if (I1 is a simple instruction &&

I2 is a simple instruction &&

I1 is not a jump instruction &&

Destination of I1 != source of I2 && **Destination of I1 != destination of I2) then**
{
issue I1 to U pipe and I2 to V pipe **}**
else
{ **issue I1 to U pipe**
}

Table 1. Pentium® instruction issue algorithm

Note that the above algorithm applies to a class of instructions called "simple" instructions. Simple instructions are a set of instructions that have no resource dependencies. That is, no two simple instructions require a single functional unit or control or data path. The instructions must be directly executed (that is they do not require micro code to execute). As it turns out, over 90 percent of instructions in the integer SPEC benchmark are simple instructions. The U and V pipes execute simple instructions in such a way as to eliminate data dependencies. Data dependencies occur when one instruction writes a register that is written to or read by another instruction (called WAR or Write After Read and RAW or Read After Write dependencies).

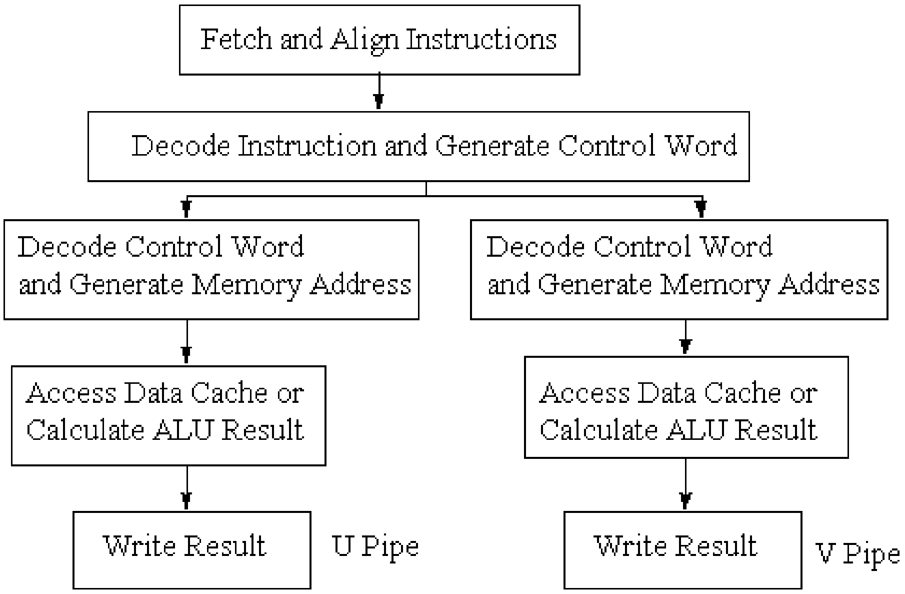


Figure 2. Pentium® Instruction Flow

1. ***Control Word***

Figure 2 shows the sequence of events in the superscalar execution of instructions. Note that the D1 stage decodes the fetched instruction and generates the control word. The control word is used to send the control signals to the data path through the control unit. Note that the truth tables are encoded in the control

ROM. Figure 3 is a closer view of the Pentium processor architecture. In this view, a connection can be seen between the Instruction Decode block and the Control ROM block. The control logic is encoded into the Control ROM. During instruction decode, the control logic output is decoded and fed directly to the control unit. The input to the control unit become the address lines for the control ROM which implements the control logic. A single control word executes instructions directly. More complex instructions require microcoded control sequences in the decode cycle.

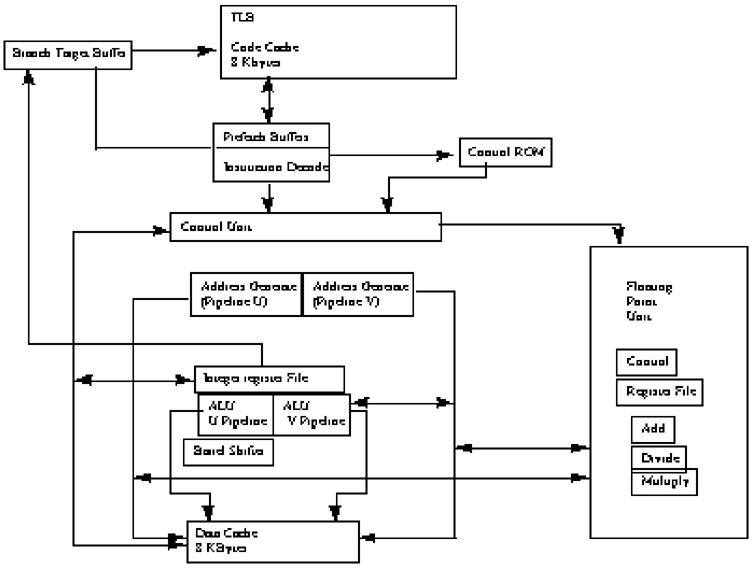


Figure 3. Pentium® Architecture Block Diagram

1. ***Floating Point Pipeline***

A pipelined floating point unit is also an integral part of the Pentium architecture. Although an instruction may require many clock cycles, the floating point unit's pipeline architecture allows the Pentium to effectively perform one floating point instruction per one or two clock cycles. The floating point unit performs one add/subtract per clock cycle or one multiply/divide per two clock cycles. Data dependencies exist between two floating point instructions when one instruction uses the result of a previous instruction. The actual computation takes place during the Execute 1, Execute 2, and Write Float cycles described below. Consequently, the latency of a floating point instruction is three clock cycles. In the older Intel architectures, floating point instructions required many clock cycles. As a result, the floating point pipeline kept relatively full. Dual execution pipelines in combination with a floating point pipeline allow the Pentium to perform integer instructions while simultaneously performing floating point calculation. They key here is that the number of integer instructions be aligned with the number of clock cycles required for the floating point calculation.

The Pentium's floating point pipeline consists of eight stages. As in the integer pipelines, the first three stages are processed by the common integer pipeline resources of pre-fetch, decode 1, and decode 2. The eight stages of the floating point pipeline are

1. Prefetch - same as integer pipeline.
2. Decode 1 - same as integer pipeline.
3. Decode 2 - same as integer pipeline.
4. Operand Fetch - Fetch the operands necessary from the data cache and the floating point register file.
5. First Execute - Execute the first stages of the floating point calculation. When data is read from the data cache, it is written to the floating point register file.
6. Second Execute - Continue to execute the floating point computation.
7. Write Float - Complete the computation and write it into the floating point register file.
8. Error Reporting - report internal errors and update the status word.

1. ***Branch Prediction***

Branches represents approximately 15 to 20 percent of instructions executed on the Pentium architecture. As such, they represent an area that provides an opportunity for performance improvement. A considerable gain is realized through the use of the Branch Target Buffer.

The BTB is a 256 entry four-way associative memory used to store branch target and history information. A specialized algorithm is used to predict the location of the branch target. When a branch is taken, memory is allocated in the BTB and the branch instructions address is associated with the destination address. At this point, the history buffer is initialized for the branch prediction algorithm. As instructions are decoded in the Decode 1 stage, the BTB is searched to determine if the branch target is stored there. When there is a hit, the CPU uses the history information to determine where to branch to. If the algorithm determines the address, instructions are fetched from that location. This process is performed early in the Write Back cycle. If the prediction of the target address is incorrect, the pipeline is flushed and resumes fetching along the correct path. If the prediction was correct, the Pentium performs branches with no delay.

1. ***Cache Memory***

The Pentium processor provides two 8-Kb caches, one for data and one for instructions. Both of the caches are two-way set associative and the cache line is 32 bits wide. The data cache provides dual access by the U and V pipes to provide additional bandwidth. Each cache has its own 64 entry Translation Look-aside Buffer (TLB) for virtual memory access. The Pentium provides a 4-Mb page size. Since these pages are so large, as a page of memory is accessed the TLB needs to be loaded only once for each page. No further TLB updates need to occur as the program accesses memory locations within that 4-Mb space. Memory accesses are greatly reduced due to reduced TLB updates. Performance gain is realized since there are fewer stalls while the CPU is waiting for data or instructions.

The data cache is organized into eight banks. The first four bytes of each cache line are fetched from bank 1, next four from bank 2, and so on. Both pipes can access the cache simultaneously as long as they are accessing different banks. The Pentium processor is designed to work with multiple processors. As a result, the cache access is designed to work using the MESI cache coherency protocol. This protocol allows the Pentium to maintain coherency between the cache and system I/O activities of different processors.

1. *Performance*

In the October issue of IEEE MICRO, Michael Bekerman and Avi Mendelson discussed their findings in "A Performance Analysis of Pentium Processor Systems"[1]. Their findings are summarized here.

Performance of the Pentium processor is largely affected by processor stalling. Processor stalling occurs when the processor must wait for some other system component of piece of data to be available before it can proceed. The three dominant causes of processor stalling were found to be memory access time, branch misprediction, and address generation interlock. Various system activities were determined to penalize the performance of the Pentium.

1. A second level cache hit was found to have a penalty of 3 cycles.
2. Main memory access was found to have a 13 cycle penalty
3. BTB misprediction caused a 3.5 cycle penalty.
4. TLB processing penalty is 28 or 41 cycles. This is the penalty to replace dirty words in the TLB.

CPI was also measured using these basic components

1. Basic CPI - this is the CPI assuming everything runs correctly and both pipes are used as intended by the instructions. (100% cache hits, no missed branches, etc.)
2. V pipe use - The V pipes contribution to performance.
3. Memory access penalties - Average system stall waiting for memory access.
4. BTB misprediction - Number of times the processor pipe had to be flushed due to missed branches.
5. Address Generation Interlock (AGI) - Next instruction needs data that has not been calculated.

The paper compared various applications using the above described criteria. Applications from the SPEC92 and SYSmark93 benchmarks. SPEC92 applications consisted of applications such as gcc and compress while the SYSmark93 suite consisted of standard Windows applications such as Lotus, Word, and Excel. Analysis data for gcc and Word are shown in the table below.

Performance Parameter	**gcc**	**Word**
Data Cache read hit rate (%)	95.69	93.79
Data Cache reads per instr	0.33	0.45
Data Cache write hit rate (%)	78.32	83.74
Data Cache writes per instruction	0.19	.24
Data TLB hit rate (%)	99.85	98.8
Instruction Cache hit rate (%)	90.09	81.6
Instruction cache reads per instr	0.38	0.36
Instruction TLB hit rate (%)	99.81	98.59
BTB Correct Prediction (%)	70.94	63.55
Instructions executed in V pipe	30.96	17.98
Overall CPI	2.15	3.49

®

Table 2. Pentium® performance running gcc, Word

In general, applications in the SYSmark93 suite performed more poorly than those is the SPEC92 suite. A noticeable difference was seen in the number of instructions executed in the V pipe for the two applications. Overall CPI could be greatly enhanced by not using as many complex microcoded and memory to memory instructions. The authors felt that the constraint rules for selecting instructions for dual pipeline use could be reduced. Also, the SYSmark93 applications tended to have a very high memory reference rate relative to the SPEC92 applications. Weak spatial and temporal locality also contributed to a much lower cache hit rate for the SYSmark93 applications. Finally, the Windows applications were not compiled with a "Pentium Aware" compiler. The use of such a compiler would take advantage of the Pentium architecture and reduce stalls.

1. **Pentium®** Pro

1. ***Overview***

The Pentium Pro processor is a completely new design from the ground up. This is in spite of being the latest in a line of *x*86 compatible processors from Intel, beginning with the 8086 processor in 1978. This new architecture incorporates many performance enhancing features, while still being able to execute existing *x*86 code. One way to examine the design of the Pentium Pro is in three increasingly detailed layers.

1. ***The First Layer: the view from the top***

1. **The on-board L2 cache**

The most obvious feature of the Pentium Pro is that it looks like this:

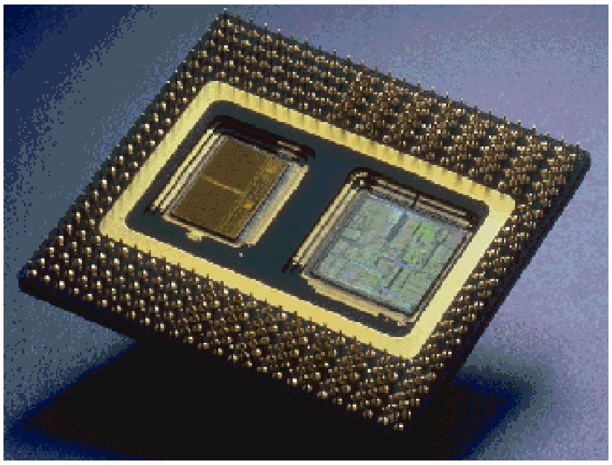


Photo from //www.intel.com

Intel calls this a dual-cavity package. One cavity contains the actual processor, the other contains the level two (L2) cache, which is offered in 256Kb and 512Kb sizes. Of the 21 million transistors in the Pentium Pro package, 5.5 million are in the CPU; 15.5 million are in the [256Kb] cache. (It is interesting that the die size for the cache is smaller that for the processor; SRAM can be packaged very densely). This on-board L2 cache is a specific feature of the Pentium Pro, and is connected to the processor by a 64-bit data bus, which includes eight bits for error correcting code (ECC) (72 bits total). The cache operates at full processor speed, unlike off-chip caches. In addition, the CPU itself maintains the cache tags. Together this means that the Pentium Pro can access data from the L2 cache and impose only a single wait state on the processor. The cache is also non-blocking (as is the L1 cache, see below). This means that when a cache miss occurs, the processor can continue to process other instructions while initiating a bus transaction to satisfy the miss.

1. **The transactional external bus**

The last point implies two other features of the Pentium Pro, namely out-of-order execution, which will be discussed below, and a transaction oriented external bus. A double cache miss (i.e. data item not found in the L1 or L2 cache) can be quite expensive as memory access is much, much slower than accessing cache. The Pentium Pro minimizes this by allowing the *backside* bus (to the L2 cache) and the *frontside* bus (the external bus) to operate concurrently. The external bus is unusual in other ways as well.

The external bus is transactional. Transactional means that it supports multiple outstanding bus requests, i.e. while a reply to one bus request is pending, additional requests may be made. Devices that can issue bus requests are called bus agents. The bus can handle up to eight transactions. The processor can track up to four transactions, and can queue up to sixteen additional transactions in the bus-interface unit (part of the processor, see below). Outstanding transactions will be resolved in the order that they were issued, for a particular bus agent. The bus has 64 data bits (plus 8 bits ECC), 36 address bits on a second bus, uses MESI (modified, exclusive, shared, invalid) protocol, and runs at around 66 MHz (i.e. $^1/_2$, $^1/_3$, $^1/_4$ clock).

The Pentium Pro "intrinsically supports glueless four-way symmetric multiprocessing" [6]. This means that the hardware is designed to allow up to four Pentium Pro's to be connected to the same external bus, i.e. it has snooping support built in. It also has a mechanism that allows it to defer a bus transaction for a relatively long time (say, several minutes). It is claimed that this should allow the designers to extend the multiprocessing ability beyond four-way.

The external bus also uses an electrical signaling scheme called GTL+ (an extension of Gunning Transceiver Logic) instead of TTL (transistor-transistor logic). This is to allow multiple devices to access the bus simultaneously, allow versions of the processor to use different voltages, and to allow higher frequency operations. "GTL+ also has better noise immunity and longer signal-propagation characteristics" [6].

1. Superscalar, superpipelined

The Pentium Pro processor is a superscalar, superpipelined architecture. Superscalar means that it can process more than one instruction per clock cycle. Superpipelined means that it uses significantly more pipeline stages than the typical 5 or 6; in fact, it uses 14. The smaller individual pipeline stages helped increase the clock speed. Intel avoided the downside of using such deep pipelines, very high stall penalties, by avoiding most stall situations. This was primarily achieved by separating the pipeline stages into three largely independent parts, and allowing out-of-order (see below) instruction execution

1. Instruction set changes

There is one change to the instruction set for the Pentium Pro. A instruction called "conditional move", or CMOV, has been added. This has no immediate effect since existing programs don't use it, and any program that did would not be able to run on an earlier *x*86 family processor, so it is not expected to see much use in the near future. It will not be discussed further here.

1. *The Second Layer: the pipeline*

1. The three units of the CPU

The fundamental strategy of the Pentium Pro design is two-fold. Firstly, translate the *x*86 instructions to RISC-like counterparts called micro-operations, or micro-ops. These micro-ops imitate a load-store architecture with triadic instructions (having three operands, two input, one destination). Secondly, buffer these converted instructions and allow them to be executed out of order. This implies that there will be some reordering of the results, called retirement in this design. These three stages, Fetch/Decode, Dispatch/Execute, and Retire, are completely decoupled, and communicate through an instruction pool called the reservation station, a 20-entry buffer that contains micro-ops that are either ready for execution or ready for retirement.

Figure 4 shows the relationship between these stages of the processor. These stages are conceptual and do not exactly correspond to physical reality, but they do accurately describe the functions performed by the processor. Figure 6 gives a more detailed view of the processor.

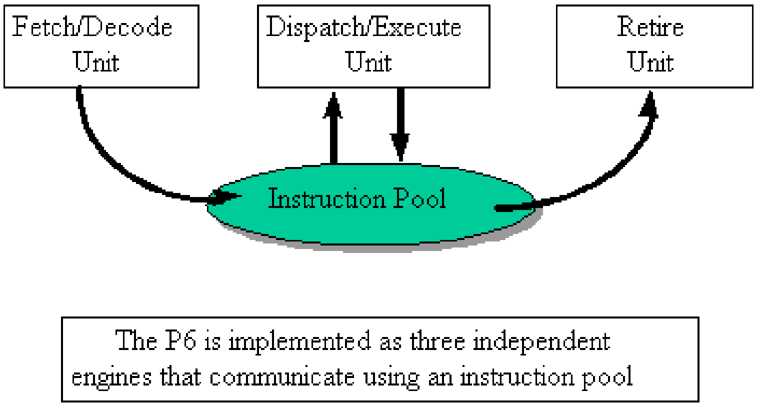


Figure 4. Pentium Pro® Instruction Flow

These stages can be correlated to the fourteen stages of the pipeline. Figure 5 shows this correlation. In a pipelined, architecture instruction interpretation is begun before previous instructions have completed, i.e. instructions are executed speculatively. This means that when a branch instruction is encountered, the instructions that the processor fetches next may, or may not be the ones that should be executed next. Also instructions may begin execution that require data from prior instructions that have not completed yet. The first section of the pipeline, which corresponds to the Fetch/Decode unit, is shown as "in-order", because instructions are processed in the order that they are fetched, i.e. in the order of the stored program. Data, i.e. program results, must be stored in memory, and in the architectural registers, in the order specified by the original program, also. This is why the last section of the pipeline, which corresponds to the Retire unit is also "in-order". Only the second section, corresponding to the Dispatch/Execution unit, can be run out-of-order. Nonetheless, this is a very powerful concept. By decoupling the three sections of the pipeline, buffering the interfaces, and allowing out-of-order execution, the Dispatch /Execution unit can kept busy almost all the time, if the other two units can keep up.

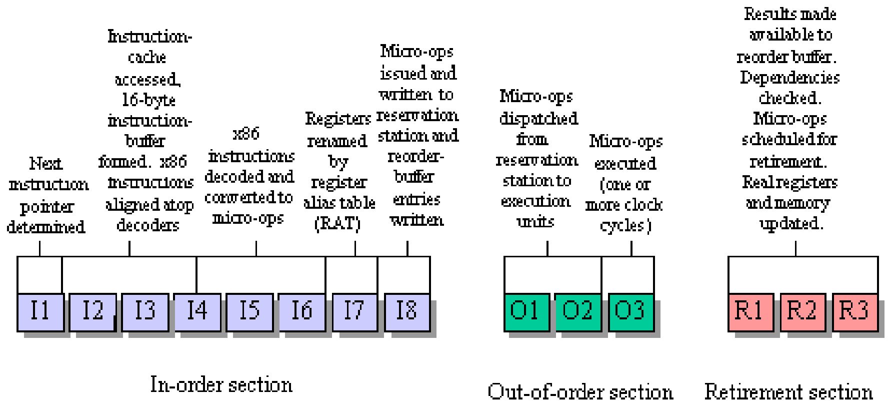


Figure 5. Pentium® Pro Pipeline Design

1. The Fetch/Decode Unit

The first part of the pipeline is the in-order Fetch/Decode unit, and occupies the first eight stages of the pipeline. *x*86 instructions are fetched from the level 1 instruction cache (L1 Icache) and are decoded into micro-ops. Up to three *x*86 instructions are read each clock cycle, and up to six micro-ops can be generated each clock cycle. More typically, three micro-ops are produced each clock cycle. Generally these three micro-ops correspond to slightly less than three *x*86 instructions. The micro-ops are each 118 bits long.

Branch prediction is also done here. This is important if the pipeline is to be kept full. The Pentium Pro can predict branches at greater than 90% accuracy. The algorithm is adaptive (or dynamic), i.e. it keeps a history of previous predictions and results to improve the prediction process.

The *x*86 standard defines eight general purpose registers (GPR), called architectural registers. The Pentium Pro has an additional 40 GPR's. Having only eight registers not only creates false dependencies, i.e. where registers are reused only because there are no more free registers, but would greatly limit the out-of-order execution designed for the CPU core. This limitation is relieved through a process called register renaming, where at any time any of the additional registers can be used to represent any of the logical registers that the *x*86 programs expects to see. The additional GPR's are wide enough to handle floating point values, and status flags.

1. The Dispatch/Execute Unit

This is the out-of-order core of the Pentium Pro processor. At this point we are dealing strictly with micro-ops, as all of the original *x*86 instructions have been translated. This unit is typically able to look 20 to 30 instructions ahead to find a micro-op that is ready to execute (theoretically there is no limit). Every clock cycle as many as five micro-ops can be dispatched to the execution units, although an average of three is more typical since some units will be busy. This process of finding which micro-ops have all their operands ready is called data-flow analysis.

This section is represented in the pipeline diagram in figure five as three stages. This is a little misleading. First of all, the diagram assumes integer operations, which take one clock cycle to execute. Floating point operations would add many additional cycles. Secondly, the first and last stages of this section almost always overlap with the adjacent stages in the adjacent sections (Intel actually claims that the pipeline length is less than fourteen because of this overlapping). Thirdly, memory access is left out of the pipeline diagram completely.

1. The Retire Unit

It is the job of the retire unit to ensure that stores to memory and changes to the architectural registers occurs in the original order of the program. Up until now every part of every instruction that has been executed has been speculative. This means that the results could be flushed entirely without affecting program correctness. This unit is where the results are committed.

Each micro-op that has been produced by the Fetch/Decode is kept in a queue (FIFO). If it is has been executed, and if all preceding instructions have been retired, then it can be retired. The retirement process involves updating the architectural registers and committing results to memory. The maximum retirement rate is three micro-ops per clock cycle.

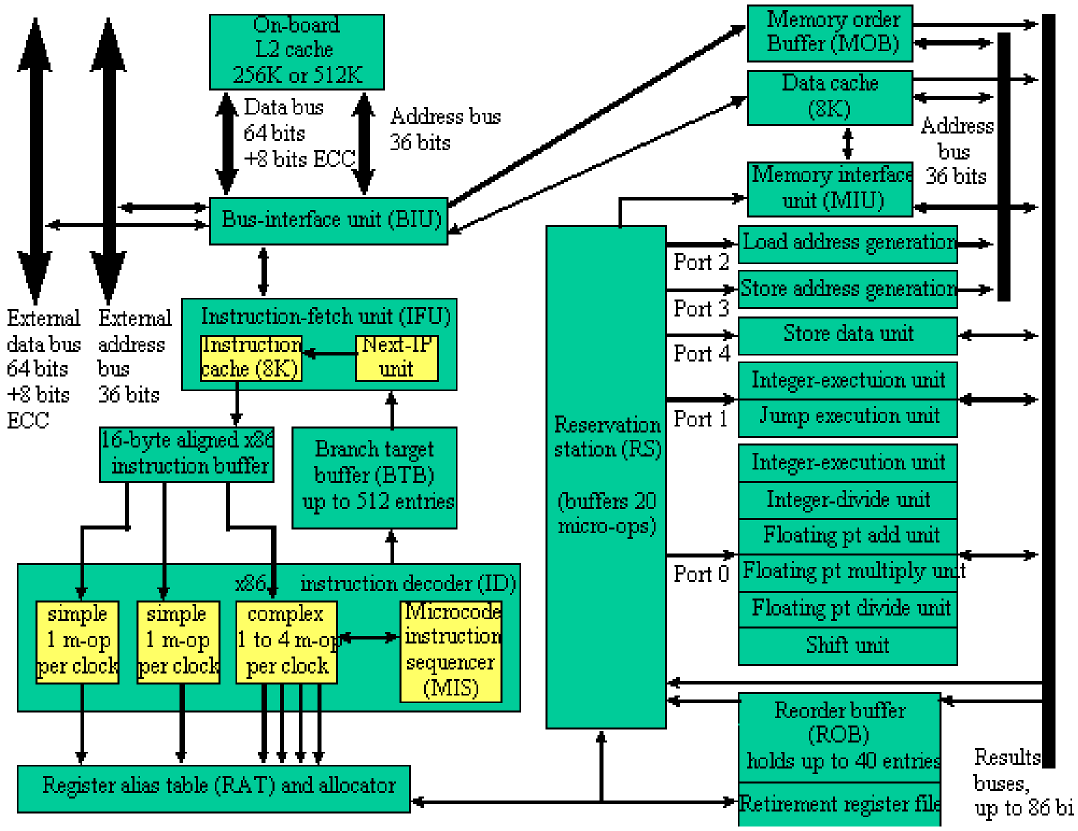


Figure 6 A View Inside the Pentium Pro® Processor

1. ***The Third Layer: the real nitty gritty***

1. **The Pentium®** Pro components

Figure 6 is a simplified block diagram of the Pentium Pro processor. It shows the contents of the three units diagrammed in figure 4 (the "instruction pool" corresponds roughly to the reservation station), as well as the L2 cache, the BIU, and the interface to the external bus.

The Fetch/Decode unit is composed of the instruction fetch unit (IFU) (which in turn is composed of the next instruction pointer and the L1Icache), the instruction buffer, the branch target buffer (BTB), the instruction decoder (ID) (which in turn is composed of two simple decoders, one complex decoder, and the microcode instruction sequencer or MIS), and the register alias table (RAT).

The Dispatch/Execute unit is composed of the reservation station (RS), the various execute units (EU’s), the memory order buffer (MOB), the data cache (DCU or L1Dcache), and the memory interface unit (MIU).

The Retire unit is composed of the reorder buffer (ROB), and the retirement register file (RRS). The MOB, DCU, and MIU can be thought of as residing in this unit as well, as it is a retirement activity to schedule memory updates.

1. **From x86 instructions to micro-ops**

Each clock cycle the IFU supplies two cache lines to be decoded, based on the output of the Next-IP unit, which takes into account inputs from the BTB, trap/interrupt status, and branch misprediction information from the execution section. Each cache line is 32 bytes long. It is necessary to supply this much data because the variable length *x*86 instruction format means that an instruction could begin at any point in a cache line, and could overlap to the next. The boundaries of the instruction are marked and the instruction is aligned in an instruction buffer for the instruction decoder (ID).

The ID actually contains three decoders that operate in parallel. They are not all identical. Two are; they are called simple decoders and are able to handle those *x*86 instructions that translate directly into a single micro-op. The other, the complex decoder, handles those *x*86 instructions that translate into as many as four micro-ops. This is why the maximum output of the ID is six micro-ops per clock cycle. Some *x*86 instructions are too complex for this decoder, or involve prefixes that modify the following instruction. These are passed to the MIS, which correspond to the microcode ROM in earlier *x*86 chips. The ID typically produces three micro-ops per clock cycle.

The BTB reads the instructions from the ID, looks for branches, and predicts the results. The BTB has 512 entries, and uses a 4-bit dynamic history algorithm to keep track of whether the branch was predicted correctly or not. This algorithm is smart enough to recognize patterns such as alternating pairs of taken/not-taken branches. Branch instructions carry with them information about the branch prediction. When the branch instruction executes and the actual branch result is known, that information is carried back to the BTB. The BTB correct prediction rate exceeds 90%, and can typically predict 10 to 15 nested branches. There is also special logic to handle subroutine CALL/RETURN pairs. It is difficult to predict where the instruction after a RETURN should be fetched from. The next instruction after a CALL is pushed onto a *return stack* which then can supply the proper fetch address to be used after a RETURN.

The last stage in preparation for the out-of-order core of the processor is to eliminate the dependence on the limited set of architectural registers. This is accomplished through the RAT, which assigned one of the 40 additional physical registers in the Pentium Pro processor to be used in place of each of the original architectural registers. The results of the RAT are placed in a six entry buffer. From there they will be passed to the RS and the ROB. That's right, they're sent to two places. The RS is where the micro-ops will be executed from, but the ROB is where they are retired from. This is necessary so that the original order can be restored later.

1. Dynamic execution

This is the out-of-order core of the Pentium Pro processor, and is considered a true data-flow engine. That means that it determines its execution sequence based on data availability and requirements, and the sequence in which the instructions are presented. The RS is capable of storing up to 20 micro-ops. It has five execution ports, and is capable of issuing micro-ops to all five ports simultaneously, if appropriate instructions are ready, and if the EU's are available. The typical rate is about three. The look-ahead capability is effectively greater than 20 because of micro-ops that are already in the execution stages or ready to be retired. Each EU has the results of the other EU's available directly without intermediate register updates via an interconnection network, similar to a crossbar switch.

A quick look at figure 6 will show an interesting set of execution units associated with each of the five ports. In particular, port 0 has a lot units available to it. The main reason is that the floating point units need a wider path. The intermediate data form of a floating point value is 86 bits. The triadic form of the micro-ops tells us that there will be three such operands. Only one port is made capable of such a wide path. Since floating point operations have such long latencies (i.e. take many clock cycles), they very rarely block access to the integer units attached to the same port. The Pentium Pro tries to maximize opportunities for parallelism by how the various execution units are distributed across the five ports. The two integer execution units do not share the same port. One port is dedicated to loads, so a load and a store operation can be dispatched simultaneously. A store is broken into two micro-ops, one to generate the data, and one to generate the address, and these can be dispatched in parallel to the two ports that are dedicated to them.

It is the responsibility of the Jump Execution Unit (JEU) to handle mispredicted branches. It changes the status of all the micro-ops that were incorrectly speculatively executed in the RS and the ROB, and provides the BTB with the correct address from which to restart the pipeline. If 20% of all micro-ops are branches, and the prediction accuracy is 90%, then 1.8% of all micro-ops are mispredictions. Every stage in the decoding portion of the pipeline pays the price for these mispredictions. The potential penalty here is anywhere from four to fifteen clock cycles[6]. The severity is less than it could be due to the out-of-order execution core which may allow unaffected micro-ops to continue to be executed.

1. Restoring order: retirement

The ROB queue holds up to 40 micro-ops that have been decoded and not yet retired. A micro-op can be retired if its results are complete, and all preceding original instructions are ready to retire. Also, all the micro-ops generated by a given *x*86 instruction must be retired at the same time. The ROB entries come directly from the RAT, so they are in the order of the original program. They are updated from the Dispatch/Execution unit in an out-of-order fashion, and it is the job of the Retire unit to impose the original program order again. Memory updates are stored in the MOB, and the

MIU is notified to issue the actual memory update when the corresponding micro-ops are retired. The RRF contains the original architectural registers, and it is updated when the micro-ops are retired, as well.

It is important that micro-ops be retired in order so that at any point in time that memory and the architectural register file reflects a true and consistent program state. The Retire unit must also watch for interrupts, traps, faults, breakpoints, and misdirected branches. When a micro-op is retired, that may affect the status of other micro-ops in that are in the ROB. The Retire unit normally retires micro-ops at the rate of three per clock cycle

1. Floating Point

The Pentium Pro uses the same basic floating point unit as the Pentium, but implements a number of improvements to the overall floating point model. The Pentium Pro's floating-point unit is fully pipelined, which will greatly improve the execution of back-to-back floating point operations. The effects of the long latencies of floating-point operations is less noticeable due to the dynamic execution features of the Pentium Pro. There is also special FXCH (floating-point exchange) hardware to allow access to any value in the floating-point stack. The *x*86 instruction set operates on a stack, and programs must do frequent FXCH operations to access values that are not at the top of the stack. The Pentium Pro will experience no additional latencies when accessing any item in the floating-point stack.

1. Frequency constraints on hardware

During the design and development of the Pentium Pro, there were many trade-offs and decisions that had to be made. In "Tuning the Pentium Pro Microarchitecture" [4], David B. Papworth discussed some of the ways that the design evolved. One interesting point was the design pressure that the high clock speed put on the allowable complexity of each pipeline stage, and the overall die size. The decision to dedicate separate execution ports to loads and to stores is one choice that was made largely to eliminate hardware that might have limited the clock speed. The original planned size of the BTB was 1024, which was cut back to 512. The original design also called for a 4-2-2-2 template for the ID (i.e. four decoders, one capable of handling up to four micro-ops *x*86 instructions, and three capable of handling up to two micro-ops *x*86 instructions) which was cut back to 4-1-1.

1. Physical Address and Paging Size Extensions

Notice that the results address buses in figure 6 are 36 bits wide. The Pentium Pro supports a 36 bit, or 64-Gb address space. It is an optional feature that will probably not see a lot of use immediately as systems will have to be developed to take advantage of it. This feature is known as Physical Address Extensions (PAE) and supports two page sizes, 4-Kb and 2-Mb. When PAE is enabled, control register 3 (CR3) points to a 4-entry Page Directory Pointers Table (PDPT), which in turn points to a page directory. If 4-Kb pages are being used, the page directory points to a page table, which in turn points to the page frame; if 2-Mb pages are being used, the page directory point directly to the page frame. When PAE is not enabled, CR3 points directly to the single page directory. Use of the PDPT does not degrade performance as all four entries will always be cached. Even when PAE is enabled, only a 32 bit (4-Gb) address space is available to the processor at any given time. Most software written to support the previous paging implementation will be incompatible with PAE.

The 32-bit linear address is interpreted as follows with PAE:

- bits 31-30: 2 bit index to the PDPT
- bits 29-21: 9 bit index to the page directory
- bits 20-12, 4-Kb pages : 9 bit index to page table
- bits 11-0, 4-Kb pages; bits 20-0, 2-Mb pages : page offset

The Pentium supports an alternate page size feature known as Page Size Extensions (PSE), which allowed 4-Mb pages in addition to 4-Kb pages. These feature is present in the Pentium Pro as well, but PSA and PSE should be considered mutually exclusive. The following table[9] shows the possible combinations of the PSA flag, the PSE flag, and the page size (PS) flag in the page directory:

PAE	PSE	PS	page size	# address bits
0	0	0	4 Kb	32
0	0	1	4 Kb	32
0	1	0	4 Kb	32
0	1	1	4 Mb	32
1	0	0	4 Kb	36
1	0	1	2 Mb	36
1	1	0	4 Kb	36
1	0	1	2 Mb	36

Table 3. PSA and PSE effects on page, address size

1. Comparison

1. *Overview*

Although the Pentium Pro is the successor processor to the Pentium, and although they are both constrained to execute the 16 year old *x*86 instruction set, the architectures have many differences. There are differences of design, size and construction, and performance, i.e. execution time for applications. The two processors are not pin compatible, and require different system environments. The following table gives an overall comparison of some of the features:

	Pentium Pro	Pentium
Clock speed	133Mhz	100Mhz
Performance (SPECInt92)	200	112.7
Superscalar degree	3	2
Execution units	5	3
Branch prediction	Dynamic, 512 entries, > 90% accuracy	Dynamic, 256 entries, 80% accuracy
Speculative execution	Yes	No
Out of order execution	Yes	No
Register renaming	Yes	No
Return stack	Yes	No
Register size	32	32
Data-bus size	64 bits plus 8 bits ECC	64 bits
Address-bus size	36 bits	32bits
Transistors	5.5 Million	3.3 Million
Level 1 cache	8 Kb instruction, 8Kb data 2-way set associative	8 Kb instruction, 8Kb data 2-way set associative
Level 2 cache	256 Kb, or 512Kb 4-way set associative	External
Nonblocking cache	Yes	No
Transactional bus	Yes	No
Processor technology	0.6 micron BiCMOS	0.6 micron BiCMOS
Die size	306 mm square	163 mm square
Price	$1200-$1600	$935, plus $100 for 256Kb SRAM

Notes: A P54C 100 MHz Pentium was used

The Pentium Pro transistor count does not include the L2 cache

Table 4. Pentium® vrs Pentium® Pro features

1. ***Design***

In the preceding table a 100MHz Pentium was compared to a 133MHz Pentium Pro. On the surface it might seem more appropriate to compare two processors with clocks at the same frequencies. Actually, that is not true. For a given technology (of microchip fabrication) it is always possible to construct a Pentium Pro with a faster clock than a Pentium. The Pentium has a 5-stage pipeline design; the Pentium Pro uses a 14-stage pipeline design and reduces the time needed for a given pipeline stage by 33%, relative to the Pentium. This design allows the Pentium Pro to run at 33% faster clock speeds than the Pentium. In a comparison of equal processor speeds (133Mhz vrs 133Mhz) it was found that the Pentium performed at 150 SPECInt92, which is a 33% improvement in performance.

The Pentium was the first superscalar processor for the Intel *x*86 architecture. It was capable of processing two instructions per clock cycle through its two pipelines, and is therefore considered superscalar of degree 2. The Pentium does not get the full use of both of its pipelines. Only certain instructions can be handled in parallel, and to the degree other instructions are present in a given application, the number of instructions per clock is reduced to less than two. This design is quite simple relative to the Pentium Pro. The Pentium Pro is a considered superscalar of degree 3, but that summarizes many details. The Pentium Pro pipeline is really three independent units. The first unit can read up to three *x*86 instructions per clock cycle and translate them to typically three but up to six micro-ops per clock cycle. The execution unit typically handles three micro-ops per clock cycle, but can dispatch up to five. The retire unit typically handles three micro-ops per clock cycle. Altogether, the Pentium Pro executes at a rate that is typically equivalent to a little less than three *x*86 instructions per clock cycle.

The Pentium Pro experiences fewer stalls than the Pentium. Stalls occur when a particular instruction cannot proceed through the pipeline because data or a resource that it needs is not yet available. The entire Pentium pipeline is in-order; if any instruction cannot proceed at any point, then all instructions that follow must wait as well. The Fetch/Decode unit of the Pentium Pro will only stall if the buffers that it writes to are full; the Dispatch/Execute unit only stalls if none of the 20 micro-ops in the RS are able to be executed; the Retire unit only stalls if none of the 40 micro-ops in the ROB can be retired. The Pentium Pro simply has more opportunities to continue. There is a caveat to this, however. Certain instructions cannot be executed out-of-order by the Pentium Pro (segment loads, I/O operations, control-register operations). These instructions cause a compete flush of the pipeline. Similarly, partial register updates can stall the Pentium Pro pipeline. An instruction in stage I7 accessing a register that an instruction at a later pipeline stage is updating a subset of will stall the Fetch/Decode pipeline. These types of instructions are common in 16-bit code, and the Pentium may actually out-perform the Pentium Pro for certain 16-bit applications because of these kinds of stalls.

Other differences between the Pentium and Pentium Pro include the following:

- Branch prediction, a key element for any pipelined architecture. The Pentium Pro has twice as deep a BTB, can predict multiple branches, utilizes a return stack for subroutine linkages, and has other improvements to the prediction logic as well. As a result the prediction accuracy of the Pentium Pro is over 90%, while the Pentium is around 80%.
- The Pentium Pro eliminates false data dependencies through register renaming and an additional 40 non-architectural general purpose registers.
- The Pentium Pro floating point unit (FPU) has improved pipelining, and latencies associated with the floating point stack have been eliminated. The FPU is capable of sustaining 200 MFLOPS at 200 MHz.
- The Pentium Pro will typically have faster access to memory due to the on-board L2 cache, accessed through a dedicated 64-bit data bus, running at full CPU core speed. Access to the L2 cache imposes only a single wait state; the Pentium will require at least two. The non-blocking nature of both on-board caches, and the transactional external bus (see section 5.2), will mean that cache misses will not prevent the out-of-order core processor from continuing execution.
- The Pentium Pro intrinsically supports up to four-way symmetric multiprocessing.
- ECC is implemented for the Pentium Pro, allowing for correction of single bit data errors (and detection of 2-bit errors).
- The Pentium Pro contains additional built-in diagnostics; more than 100 events and variables can be counted or output, e.g. cache misses, register contents, occurrences of self-modifying code.

1. *Performance*

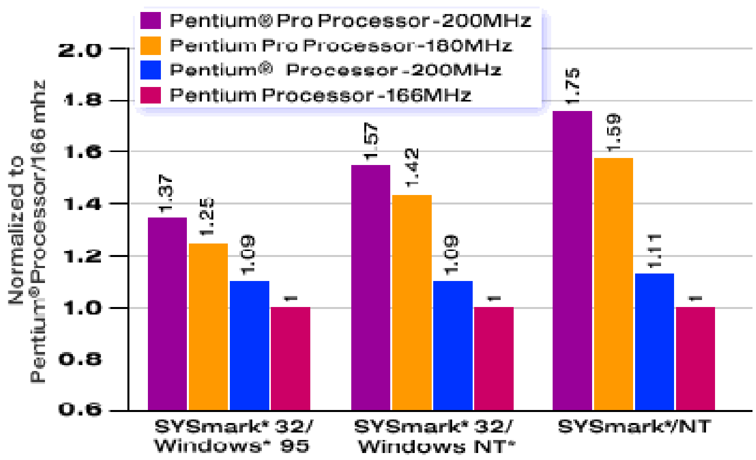
The bottom line measure of the success of the changes made for the Pentium Pro is how well it can run applications. The following table[8] contains results of a BYTE magazine study of several applications running on a variety of Pentium and Pentium Pro systems, all with 200 MHz clocks, executing under WindowsNT 3.51.

	ZD Winstone 96	AutoCAD	C Compiler	Photoshop Gaussian Blur	Photoshop Lighting Effects
	score	min:sec	min:sec	min:sec	min:sec
Pentium Pro/200 Systems					
NEC Powermate Pro 2200	157	3:05	1:46	0:43	1:09
SAF STA Pro 200 Desktop	169	2:55	**1:43**	0:40	1:03
Zenon Optima Pro Station	**174**	**2:51**	2:08	**0:33**	**0:57**
Pentium/200 Systems					
Dell Dimension XPS P200s	124	4:36	2:46	1:00	1:24
Gateway p5-200XL	121	4:43	2:53	0:58	1:06
IBM PC 350	122	4:53	2:56	0:57	1:08
Micron Millennia P200	101	4:44	**2:45**	0:59	1:11
Sys Performance P200T2	**129**	**4:35**	2:49	**0:55**	**1:04**

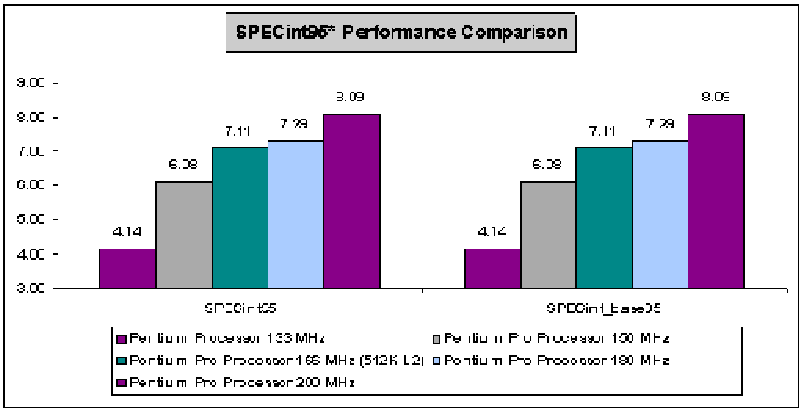
Table 5. Pentium® vrs Pentium® Pro NT application comparison

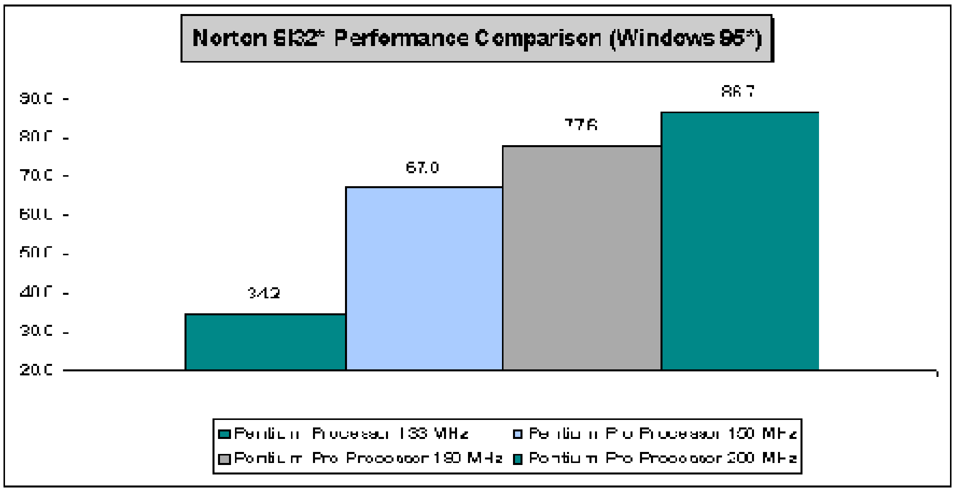
Using the best results (in bold) the Pentium Pro outperformed the Pentium by 35%, 38%, 38%, 40%, and 11% in each category, respectively.

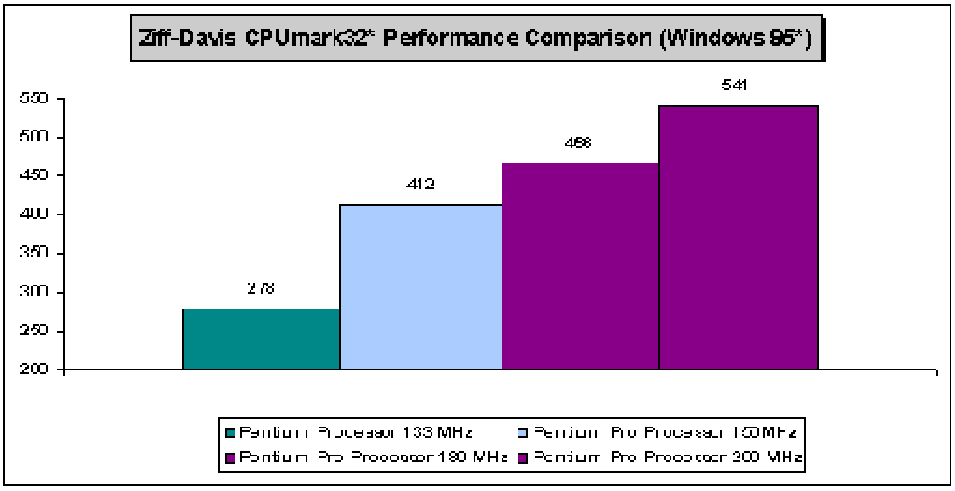
Intel provides the following two charts from their web site on the internet[16]. Comparing the two 200MHz machines, under WindowsNT, as in the above example, we find performance improvements of 44% and 58% for the two test cases. Using the same applications as the 44% result, but with Windows95 (which contains some 16-bit code) the improvement drops to 26%. When we choose a set of 16-bit applications, but with WindowsNT, the improvement drops to 6%. A set of 16-bit applications, this time with Windows95 produced a performance degradation of 4%. (Configurations and benchmark information for these results can be found in Appendix 1[16]).

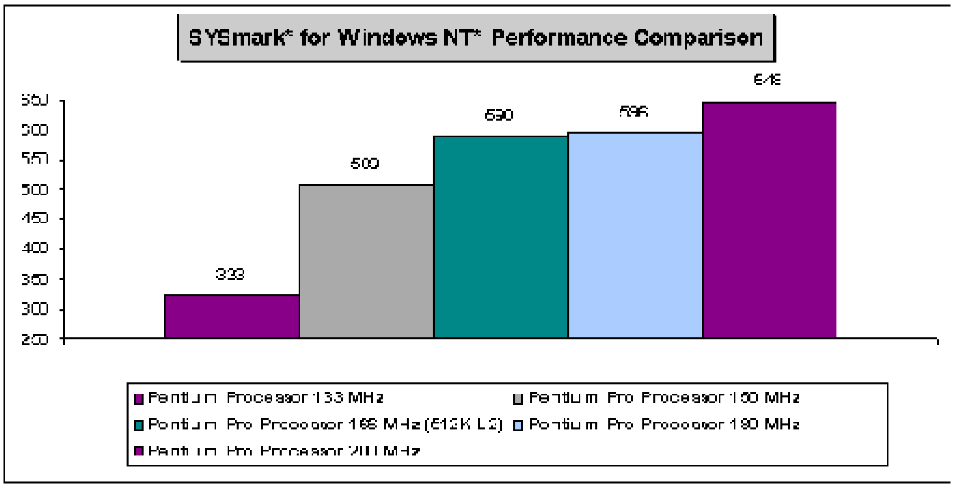


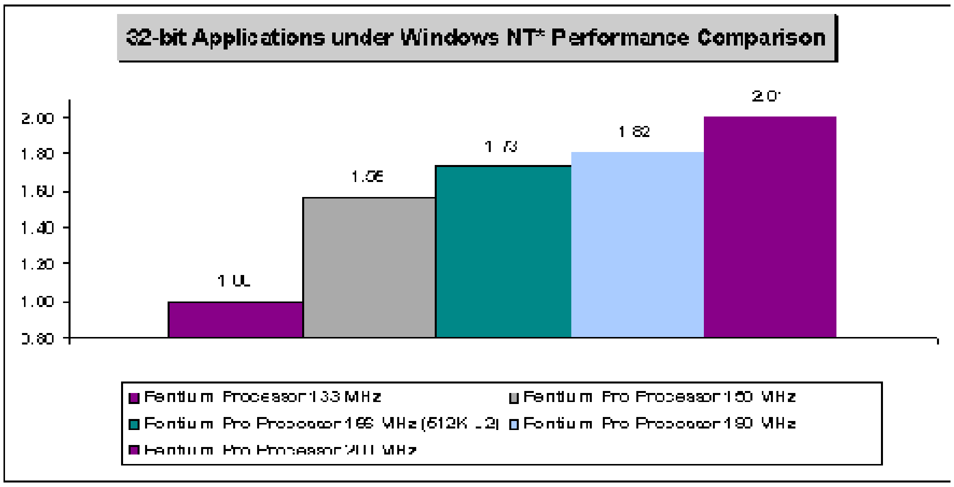
Intel provides the following 7 charts from the support section of their web site[14]. As was pointed out earlier, it is not entirely accurate to compare a Pentium and a Pentium Pro at the same clock speed. The following results includes a 133 MHz Pentium, and a 180 MHz Pentium Pro, a roughly 33% faster clock. A table summarizing the performance differences of these two machines from these charts follows the charts. (Configurations and benchmark information for these results can be found in Appendix 2[14]).











Benchmark	Operating System	Pentium Pro 180MHz/ Pentium 133MHz (% improvement)
SPECint95	UNIX	76%
SPECint_base95	UNIX	76%
SPECfp95	UNIX	95%
SPECfp_base95	UNIX	118%
Norton SI32	Windows95	127%
Ziff-Davis CPUmark32	Windows95	68%
SYSmark	WindowsNT	85%
32-bit applications	WindowsNT	82%
32-bit applications	Windows95	35%

Table 6. Pentium® vrs Pentium® Pro benchmark summary

While it is no surprise that the lowest performance improvement numbers were for cases that were run on the Windows95 operating system, which contains some 16-bit code, the very highest result was also run on Windows95. In the one case where the same applications were run on WindowsNT and Windows95, we see the expected drop in performance improvement, from 82% to 35%.

One industry analyst[5] stated the following expectations:

- 32 bit applications, with a 32 bit operating system, the Pentium Pro should be 40-60% faster
- 32 bit applications running on Windows95, the Pentium Pro should be 20-30% faster
- 16 bit applications running on Window 3.11, the Pentium Pro will likely be slower

One thing is for sure; Intel has gone to great pains both to ensure and to reassure the public that there will not be a repeat of the floating-point-divide (fdiv) debacle. Their web site is full of information about how thoroughly the Pentium Pro was tested; and they promise to tell everyone about it this time if is something should go wrong anyway. An interesting anecdote is that the way the Intel people confirmed the existence of the Pentium fdiv bug was when they discovered that the Pentium Pro results agreed with the 486, but not with the Pentium.

1. *The Future*

This investigation led us to be curious about where Intel might be going from here. Some of the sources we found contained some information on this, although it appeared that the authors were uncertain about it. Therefore the following may not be 100% accurate, and is certainly subject to change:

- a lower cost processor, codenamed Klamath, will be available in 1997; it will be the Pentium Pro CPU without the L2 cache, plus MMX multimedia extensions, with 16-bit enhancements (perhaps a segment-descriptor cache, a Pentium feature omitted from the Pentium Pro); maybe more L1 cache (32-Kb?), maybe an interface to special high-speed external SRAM's.
- a "beefed-up" version of the Pentium Pro, known only as P68 right know, planned for late 1997, that may add units to the ID, and entries to the RS; maybe add an EU
- the P7, or Merced, a joint project with HP Corp., will implement a true 64-bit architecture, probably using VLIW technology; planned for late 1998
- something known only as the P8, being developed by HP Corp. concurrently with the other projects mentioned above.

Intel declined to comment on the veracity of these predictions (see appendix 3 for a copy of the E-mail response from Intel).

1. Conclusion

The primary issue that the Pentium Pro design team attempted to address was the underutilization of the CPU core. When the execution units of the CPU are idle, it is usually due to memory access, unavailable resources, or, for pipelined architectures, data hazards and branch hazards. Waiting for memory access is a particularly onerous problem; over the last 10 years core CPU speeds have increased by a factor of ten, while memory access speeds have increased only 60%. This creates an ever-growing issue of memory latency.

There were evolutionary improvements in the Pentium Pro, like improved branch prediction, FXCH hardware, and an increased number of execution units. More revolutionary, and more directly to the objective, was the introduction of Dynamic execution, coupled with the superpipelined design, and the modifications to the memory management, i.e. the on-board L2 cache, the non-blocking nature of the on-board caches, and the transactional external bus. These features work in tandem to address the design issues. Dynamic execution allows more execution units to be used effectively, allowing a higher degree of superscalar behavior. The superpipeline's short stages allowed a 33% higher clock speed. The transactional bus allows multiple memory requests, which supports the multiple execution units in the out-of-order core of the processor, and the non-blocking caches allow other (out-of-order) instructions to continue to execute.

The Pentium Pro largely achieved its objectives. Under targeted conditions, the Pentium Pro can deliver double the performance of the Pentium. Looking to the future, the Pentium Pro seems to be a superior platform for the applications that being developed and designed today.

There are some very strong exceptions and conditions to that success. While many causes of pipeline stalls have been eliminated, or ameliorated, by the Pentium Pro's innovative design, the very deep superpipeline leaves it exposed to severe performance penalties by those stall conditions that remain. This is exacerbated by the specific design choice to omit certain hardware features of the Pentium that could provide some assistance, presumably out of concern that the clock speed might be impacted. That these conditions are common in 16-bit code, and uncommon in 32-bit code means that the Pentium Pro will become increasingly attractive as time goes by.

The Pentium Pro is not an upgrade option for Pentium systems. The dual-cavity package will not fit in the space of a Pentium chip, and the memory system requirements are completely different. The bundling of the L2 cache and the processor takes away some systems design options for computer manufacturers; memory design has been a major differentiating factor for Pentium systems. It is unclear if the future holds a Pentium Pro processor that does not include this on-board cache, or how Intel will make the more expensive Pentium Pro attractive without the performance benefit of the on-board L2 cache.

Overall it seems that Intel has a winner with the Pentium Pro, but it may take some time to realize it. Many heavy users of 16-bit code may wait quite a while before upgrading. Those who find the Pentium insufficient for their needs, a very small percentage of the total marketplace, have a solid, viable solution available now.

1. Bibliography

1. Michael Bekerman, Avi Mendelson, "A Performance Analysis of Pentium Processor Systems", IEEE MICRO, page 72, October 1995.
2. Donald Alpert, Dror Avnon, "Architecture of the Pentium Processor", IEEE MICRO, page 11, June 1993.
3. Ramesh Subramaniam, Kiran Kundargi, "Programming the Pentium Processor", Dr. Dobbs's Journal, page 34, June 1993.
4. David B. Papworth: "Tuning the Pentium® Pro Microarchitecture", IEEE Micro, page 8, April 1996
5. John Clyman, Nick Stam, Sebastian Rupley: "P6: The Next Step", PC Magazine, page 104, Sept 12 1995
6. Nick Stam, John Clyman: "Inside the P6", PC Magazine, page118, Sept 12 1995
7. Tom R Halfhill: "Intel's P6", BYTE, page 42, April 1995
8. "Pentium Beats Classic at 32 Bits", PC magazine, page 155, Sept 24 1996
9. Robert R. Collins: "Paging Extensions for the Pentium Pro Processor", Dr. Dobb's Journal, page 107, July 1996
10. "A Tour of the Pentium® Pro Processor Microarchitecture", http://www.intel.com/procs/p6/p6white
11. "The Intel Pentium® Pro Processor, Frequently Asked Questions: Technical", http://www.intel.com/procs/ppro/faq/p6techq.htm
12. Robert P Colwell, Randy L Steck: "A Technical Tour of the Intel Pentium® Pro Processor", http://www.intel.com/procs/info/isscc
13. "The Pentium Processor", http://www.intel.com/procs/pentium/pptech
14. "Intel Pentium® Pro Processor Performance Brief", http://support.intel.com/oem_developer/microprocessors/pentium_pro/7583.HTM
15. "Pentium® Pro Processor Product Highlights", http://channel.intel.com/business/europe/sales/pprobrf.htm
16. "Business Desktop Compu>

Transfer interrupted!