

Intel Pentium 4 and NetBurst Micro-Architecture

Jason Waltman
MS, University of Utah

Advanced Computer Architecture · CS 6810

Al Davis

3 December 2002

(Revised: 5 December 2002)

1 Introduction

Intel Corporation (INTC), founded in 1968, developed the first microprocessor in 1971. Approximately 75% of all computers in use today use Intel processors, making Intel the world's largest microprocessor manufacturer. Intel has 45 offices worldwide, 80,000 employees, and reported \$26.5 billion in revenues in 2001.

Intel's current microprocessor is the Pentium 4. The processor uses a redesigned micro-architecture named 'NetBurst' by the Intel marketing group. NetBurst reflects the end of the P6 micro-architecture era which began with the Pentium Pro in 1995 and was used in all later Pentium versions through the Pentium III until now.

This paper will discuss the architecture of the Pentium 4 (P4). I will give only a brief overview of the IA-32 instruction set architecture (ISA) before discussing the more interesting new NetBurst micro-architecture, including details about the P4 memory and I/O architectures. I will conclude with some analysis of the Pentium 4, specifically, with what I think is good and maybe not so good about the current design and recent Intel design decisions.

2 IA-32 Instruction Set Architecture

The IA-32 (32-bit Intel Architecture) ISA used in the Pentium 4 is the same base ISA that was used by Intel in the first generation 80x86 processor, the 8086, in 1978. (Even though the 8086 was a 16-bit architecture Intel uses the name IA-32 to refer to both 32- and 16-bit architectures since they are compatible.) For compatibility with legacy applications the ISA remains today. This means that any application written for the 8086 over 20 years ago, will run on the Pentium 4 and all 80x86 IA-32 processors in-between.

IA-32 is a CISC-like, general purpose register architecture. The complex instruction set contains over 300 instructions. Beginning in 1995 with the P6 micro-architecture and Pentium Pro, Intel changed to using a more RISC-like core, adding a front-end decoder that would translate IA-32 instructions to simple micro-ops (μ ops) that the execution units would process (an IA-32 instruction is made up of one or more μ ops).

As a result of its humble beginnings, IA-32 has only 6 general-purpose architected registers (8 if you count the reserved stack

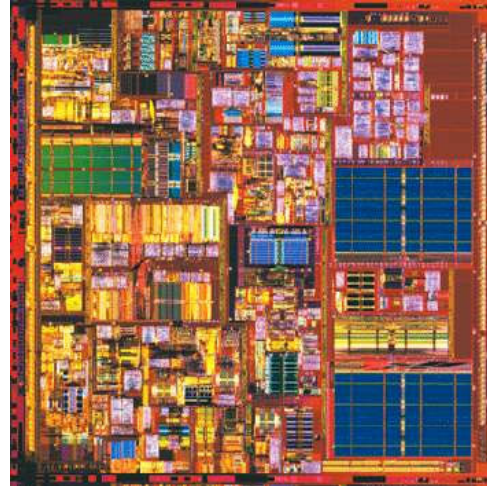


Figure 1: *Intel Pentium 4, 2.4 GHz, 131mm² die.*

and base pointers), 6 segment registers, a status flag register and an instruction pointer. These registers were each 16-bits (one word) in the 8086. Some registers were extended to 32-bits with the 386 processor and later additions to the instruction set added 64- and 128-bit registers for MMX/SSE(2). IA-32 is a byte-addressable, little-endian architecture and can address 64 GB of physical memory. Logical IA-32 addresses are a combination of a 16-bit segment register value and a 32-bit offset. Linear addresses in a segment are further broken down into pages.

IA-32 contains all the common instructions you'd expect in a ISA, including instructions for: data transfer, arithmetic, logical operations, bit shift/rotate, control transfer, string manipulation, et cetera. A few common instructions semantics traditionally vary on the 80x86 from other architectures. For example, IA-32 compare instructions set a flag register that branch instructions look at in determining whether or not to jump. A call instruction saves the current IP on the stack; a return pops the IP from the stack.

Instructions can have 0-2 operands, one may be a memory location, and the others may be constants or registers. Some operands like division and multiplication use two registers for their operands. For example, the dividend of division is two (specific) registers and the quotient and remainder end up in the same two registers.

IA-32 has six addressing modes. For brevity, I won't explain them, but their meaning should be relatively clear from their

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
TC	Nxt IP	TC	Fetch	Drive	Alloc	Rename	Que	Sch	Sch	Sch	Disp	Disp	RF	RF	Ex	Flgs	Br Ck	Drive	

Figure 2: Intel NetBurst 20-stage pipeline.

names: immediate mode, register, direct, register indirect, base+offset indirect, index+offset indirect, and base+index+offset indirect. IA-32 has three execution modes: protected mode (native processor state), system management mode (OS use for context saving and address space switching), and real-address mode (the original 8086 processor environment).

3 Intel NetBurst Micro-Architecture

Details of the NetBurst Micro-Architecture—the main focus of this paper—were released to the public in late 2000. The Pentium 4 and current Xeon processors are the first to use the new micro-architecture. Intel's goals for NetBurst, the successor to the P6, were two-fold: (1) to be able to execute legacy IA-32 and SIMD (executing a single instruction across multiple data) applications and (2) operate at high clock rates that will scale easily in the near future. Intel markets the features of NetBurst that attempt to reach these goals as the following:

- Hyper-Pipelined Technology
- 533 MHz Front Side Bus
- Advanced Dynamic Execution
- Rapid Execution Engine
- Execution Trace Cache
- Advanced Transfer Cache
- Streaming SIMD Extensions 2 (SSE2) instructions
- Hyper-Threading Technology

Rather than divide this section into the above categories, I'll talk about the architecture in terms of processor instruction execution and defer some details of the marketing terminology to later sections.

First, however, the "Hyper-Pipelined Technology" refers to the 20-stage pipeline of the NetBurst micro-architecture (see figure 2). This is twice as long as the pipeline on the P6 micro-architecture and is the primary reason Intel is able to get such fast clock rates (if less work is being done on each clock tick, clock ticks can happen faster). Intel claims that the 20-stage pipeline will allow them to reach 10 GHz clock frequencies in the future without another micro-architecture change.

NetBurst instruction execution is broken into three main parts: an in-order issue front end, an out-of-order (OOO), superscalar execution core, and an in-order retirement. The job of the front end is to feed a continuous stream of μ ops to the execution core. The fetch/decode unit can decode one IA-32 instruction per clock (complex instructions are looked up in a microcode ROM) and passes μ ops to the Execution Trace Cache

(described later). The trace cache passes μ ops to the execution core and can issue 3 μ ops per cycle.

NetBurst's "Advanced Dynamic Execution" is via an OOO, superscalar execution core. Up to 126 instructions may be 'in-flight' at once. Forty-eight of these may be loads; 24 may be stores. The core can dispatch up to 6 μ ops per cycle and uses register renaming (the Pentium 4 has many more than 8 general purpose registers) instead of a reorder buffer which was used in the P6 micro-architecture. In this way, if one instruction is delayed waiting on data or a contended resource, other instructions that have available resources may execute out-of-order. Stores may be forwarded to dependant loads before being written to memory; loads may be speculative (but cannot cause a page fault).

The execution core has seven execution units accessible through one of four issue ports. One or more ports may be issued on each clock cycle (see figure 3) and ports may then dispatch one or two μ ops per clock. The possibility of dispatching two μ ops per clock should come as a surprise. Two integer ALUs on the P4 actually run at twice the core processor frequency (this is what Intel calls the "Rapid Execution Engine") and can execute in half a clock cycle. Two instructions are issued by sending one in the first half of the cycle and the second in the second half (if, of course, the port has access to these "double speed" ALUs.

Retirement in NetBurst is done in-order to insure that system state is left as the programmer intended after execution of an instruction. Exceptions may be raised as instructions are retired and therefore cannot occur speculatively. Up to 3 μ ops can be retired per cycle.

The branch predictor in the Pentium 4 is more advanced than in any other Intel processor. The branch delay on a correctly predicted branch could be as little as zero clock cycles; a mis-predicted branch costs on average 20 cycles (the length of the pipeline). The predictor predicts all near branches including calls,

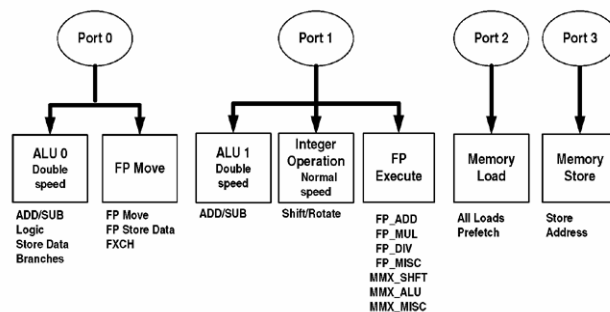


Figure 3: NetBurst Issue Ports and Execution Units.

returns, and indirect branches. The predictor does not predict any far branches. The predictor is dynamic, receiving its information for previous branches from the retirement logic into a 4 KB branch target buffer (this is 8 times larger than the Pentium III's). If no dynamic information is available for a branch, a static predictor which predicts backward branches as taken, is used.

Prefetching in the NetBurst micro-architecture happens automatically by the hardware by bringing data or instruction cache lines into L2 based on prior reference. Basing prefetching decisions on prior reference is a new feature. The hardware can also do normal linear prefetching of instructions. Also, SSE2 (see below) adds instructions that can allow a programmer to request a data prefetch using software for code with irregular access patterns.

NetBurst adds the next level of multimedia extensions to the 80x86 processor line. SSE2 is an additional 144 instructions on top of Intel's MMX and first set of Streaming SIMD Extensions. In part, these instructions add 128-bit SIMD integer and double-precision floating point operations as well as support for new cache and memory management operations.

4 Hyper-Threading Technology

Less than one month ago, on 14 November 2002, Intel released a 3.06 GHz Pentium 4 with "Hyper-Threading Technology" enabled. Hyper-threading (HT)—the ability of a processor to execute more than one thread at a time and appear to the OS as two logical processors—was actually on the P4 die for awhile as the additional logic is less than 5% of the total die size. Had Intel been ready to go public with HT before now we would've seen it in an earlier P4 incarnation. HT works on the theory that usual processor utilization is only about 35% of maximum. By sending two threads into the processor at a time, would-be idle execution units can be used by a second thread. Intel claims that with HT enabled, processor utilization can increase to 50%. The interesting thing about HT is that the execution core and memory

hierarchy don't have to change at all in order for it to work. The trace cache is shared between the two threads; both the trace cache and retirement logic alternate between threads so both logical processors can make forward progress. HT needs OS support in order to work properly. Both Windows XP and Linux 2.4.x support HT. Older OSs will have problems with HT enabled because the system BIOS will report a P4 with HT as two processors—the OS won't be able to distinguish between a logical and a real processor.

5 Pentium 4—NetBurst Memory Details

NetBurst supports caches, TLBs (translation lookaside buffers), and a store buffer for temporary instruction and data storage. The memory architecture is byte addressable, uses segmented-paged addressing (a segment is an independent address space, e.g. to separate code and stack), and three page sizes (normal is 4 KB; "large" are 2 MB or 4 MB).

The Execution Trace Cache (TC) is one of the more interesting additions to the NetBurst micro-architecture. The idea behind the TC is simple. The IA-32 decoder has one of the highest gate counts of all pieces of logic on the P4—which means decoding takes a long time. The traditional decoder has to run every time an instruction is encountered and even worse, also on mis-predicted branches. The TC stores decoded μ ops in program execution order (it can store 12K μ ops) so they don't have to be decoded again. The TC works with the branch prediction hardware to build logical "traces" of μ ops over predicted branches. When instructions are encountered that are in the TC, a trace is sent to the execution core directly, without having to be decoded. The P4 TC is 8-way set associative and can deliver up to 3 μ ops per clock. This replaces a traditional L1 instruction cache.

The P4 L1 data cache is 8 KB (half the size of the Pentium III's), 4-way set associative, and has 64-byte cache lines. The L1 interface has two 64-byte channels and can handle one load and one store per clock. L1 data load latency is 2 cycles.

Depending on how recent the processor (specifically, if it's manufactured with a 0.18- or 0.13-micron process), the P4 has either a 256 or 512 KB unified, on-chip L2 cache. The L2 has a 256-bit interface to the execution core and can transfer data on each cycle. The L2 is non-blocking, full (clock) speed (thus the name "Advanced Transfer Cache"), 8-way set associative, and has 128-byte cache lines. The total load latency from L2 is 7 cycles. Data is received from the northbridge in 64-byte pieces.

The NetBurst memory hierarchy levels are non-inclusive, that is, a line in an upper level does not imply a line in a lower level. Replacement is pseudo-LRU (least recently used). L1 is write-through (buffered); L2 is write-back.

TLBs store most recently used page-directory and page-table entries. The P4 instruction TLB has 128 entries and is 4-way set associative; the data TLB has 64 entries and is fully associative.

The P4 store buffer allows writes to be delayed for more efficient use of memory-access bus cycles. The store buffer has 24 entries.

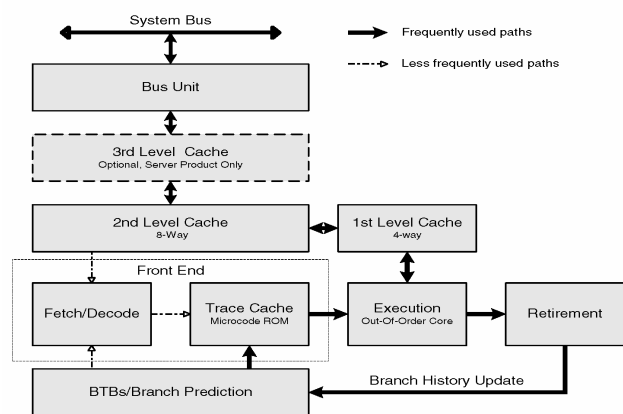


Figure 4: NetBurst Memory Hierarchy.

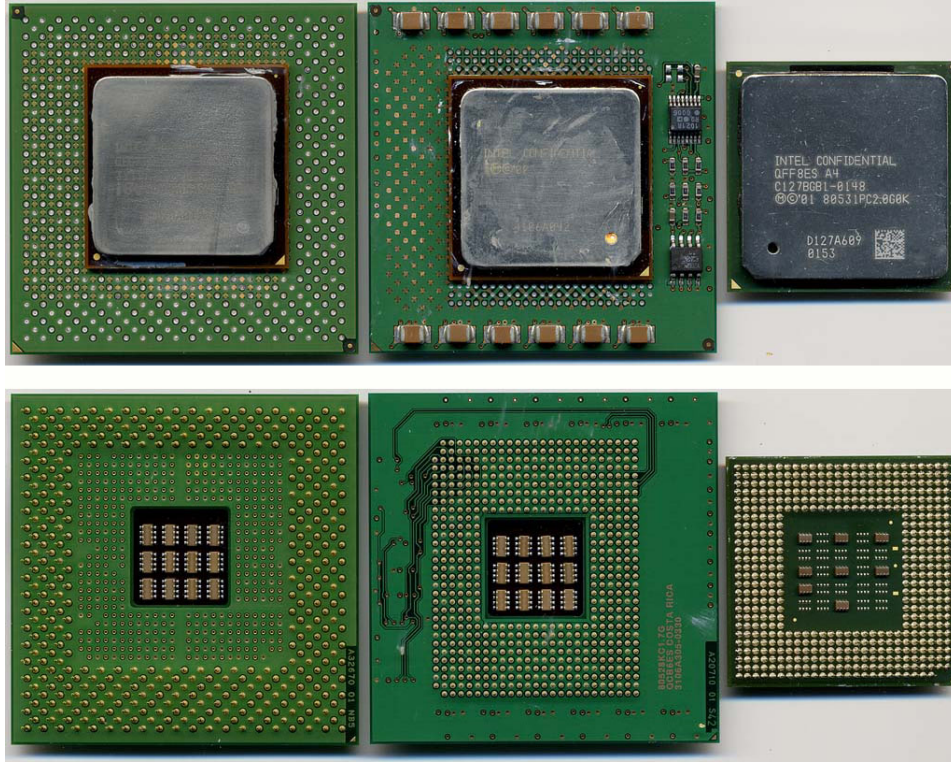


Figure 5: From left to right: Socket-423 Pentium 4, Socket-603 Xeon, Socket-478 Pentium 4.

6 Pentium 4—NetBurst I/O Details

The advertised front side bus (FSB) on the P4 is 400 or 533 MHz (depending on the age of the processor). This is the effective speed of the bus, but actually, the FSB is “quad pumped” and running at only 100 or 133 MHz. A quad pumped bus means that data is transferred four times per clock—twice on the rising edge, twice on the falling edge. At 100 MHz there is only 10ns between clock cycles; detecting 4 different voltages in this short amount of time is quite a feat. Shortening this time to 7.5ns (with 133 MHz bus) is even more incredible and translates to incredible bandwidth (4.2 GB/s). A buffering scheme is used to sustain 533 MHz data transfers.

Intel released the P4 with their i850 chipset (which is similar to the Pentium III’s i840). The i850 supports 4X AGP, Ultra ATA/100, dual channel RDRAM, 24 MB/s USB bandwidth (2 controllers over 4 ports), 6 channel audio, and a LAN Connect Interface. In addition, the i850 was designed to support the P4’s quad pumped FSB. The choice to support only RDRAM with the i850 caused quite a stir in the microprocessor following community. The faster (due to dual channel support—requiring RIMMs in pairs of two), more expensive RDRAM Intel chose nonetheless to pair with their incredibly fast FSB. Since the processor’s introduction in 2000, 3rd party chipset designers as well as Intel have released controllers that work with various types of SDRAM, including the popular DDR-SDRAM.

The P4 has either a 423 or 478 pin interface to the motherboard. The 478 pin interface was introduced with the 2.0 GHz version of the processor with a more densely packed arrangement of pins called a micro Pin Grid Array making the package physically smaller with a smaller motherboard footprint (see figure 5).

7 Pentium 4 Implementation

The first Pentium 4 processors were released in November 2000 at 1.4 and 1.5 GHz using a 0.18-micron fabrication process and 256 KB L2 cache. Since that time, the 20 stage pipeline of the NetBurst micro-architecture, and the move to a 0.13-micron process has allowed Intel the speed scaling they had hoped for. The first 0.13-micron process (see figure 6) P4 was released in January 2002 at 2.0 GHz (and only available with the 478 micro pin interface). The smaller transistors allowed room on die for a bigger L2 cache, which Intel doubled from 256 KB to 512 KB. In April 2002, Intel made the switch from 200mm wafers to 300mm wafers (see figure 7) with their new plant opening in Hillsboro, Oregon. In May 2002 Intel released the 533 MHz FSB and a new i850E chipset to support the new bus speed increase. At launch two years ago, the Pentium 4 had 42 million transistors, a 217mm² die, thermal dissipation of 52W and VCore voltage of 1.75V. More recent versions of the processor have 55 million transistors (due to the L2 increase), a die size of 131mm², thermal dissipation of about 68W and VCore voltage at 1.5V.

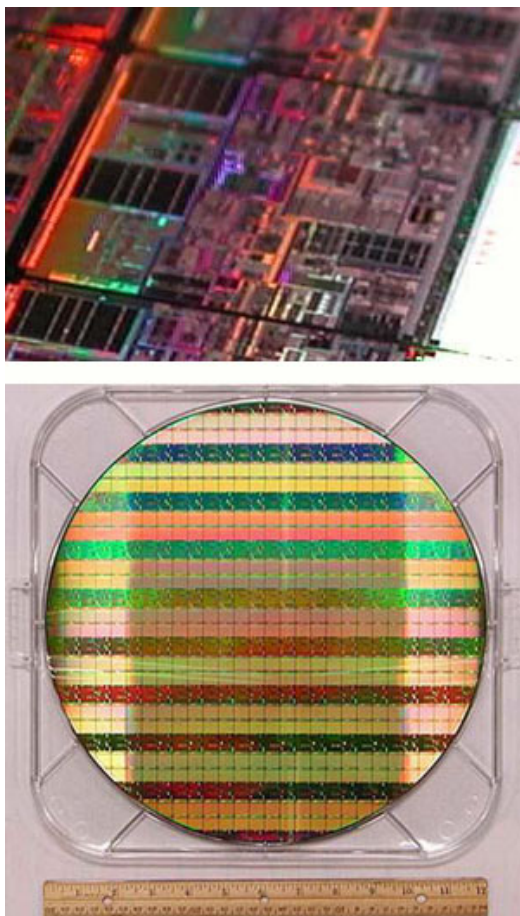


Figure 6 (top): Close up of 0.13-micron Pentium 4 die.

Figure 7 (bottom): A 300mm Pentium 4 wafer.

The most recent Pentium 4 release by Intel was on 14 November 2002, the 3.06 GHz processor with hyper-threading enabled.

8 Analysis

Personally, I think that any current Intel processor is an amazing device. The fact that they've been able to build a microprocessor on top of the 8086 ISA—a design that was never meant to be used the way it is today—is simply amazing. Luckily for Intel, the 8086 ISA was general enough to be able to morph it into today's high performance processors. The continued popularity of the 80x86 processors is due to two facts that consumers have grown to expect: compatibility with older applications and a continuous increase in speed.

8.1 Performance

When first released, the Pentium 4—in certain situations—was actually slower than the then-fastest Pentium III; it was consistently slower than AMD's then-best processor as well.

While sort of an embarrassment for Intel in 2000 I believe that they understood the ramifications of what they were doing and what the NetBurst micro-architecture would be able to excel at in the near future. Computer geeks and hardcore computing consumers like controversy. They want some company to hate, they want the underdog to win sometimes; they don't like things to remain static. Intel was overdue for a new micro-architecture and the computer geeks knew it. AMD's results kept getting better and Intel needed to get the spot light off of them. One could argue that delivering a new 'slow' processor might shine the wrong color light on Intel, but I don't believe so. Intel's goal was to just get the processor out and get a few people to use it. If the marketing hype was good enough, consumers wouldn't ever realize the speed issue.

NetBurst was designed, not to deliver a killer processor in 2000 but, as a platform on which to develop the killer processor for the future. It took a few years, but in May 2002, after switching to the 300mm wafers, 0.13-micron process, 512 KB L2, 533 MHz FSB, and bumping the core clock to 2.5 GHz, the P4 was finally consistently getting better performance than AMD in all benchmarks.

AMD will respond, and soon, but Intel's innovation with NetBurst, I'm assuming, is going to keep them in the lead for a while. Intel's trace cache is one of the most interesting pieces of NetBurst—and something that has proved to be a big performance booster. The SSE2 extensions and recently released hyper-threading technology also have extraordinary potential. As programmers (especially game programmers) begin programming for these new technologies, code on a new Intel processor will have a great advantage over an AMD chip (until AMD can re-implement SSE2 and hyper-threading).

8.2 Execution

The 20-stage pipeline of the NetBurst micro-architecture was an interesting move. To the untrained eye, Intel's high clock speed looks much better than AMD's, however Intel's not fooling any computer geek. Going to the 20-stage pipeline reduced the Pentium 4's IPC (instructions per clock), increased the penalty for mis-predicts, and forces more record keeping (recall up to 126 OOO instructions can be in-flight at once!). Intel was not unaware of these issues and they designed around them. The longer pipeline was needed to allow future scalability (Intel estimates that NetBurst can reach clock frequencies of 10 GHz)—they simply designed around the problems the larger pipeline would cause.

One of the surprising 'work-arounds' was the "Rapid Execution Engine", that is, the double speed integer ALUs. This decision (or at least something similar) was actually necessary in order to keep the initial P4s on par with the Pentium IIIs. It's as good a solution as any as far as I'm concerned. Integer code is usually more unpredictable than floating-point code, which means that branch mis-predicts are going to happen more often with integer code. If you can double the speed of the integer code, it should hide at least some of the latency. What will be interesting to see is how far Intel will be able to keep up with the double

speed ALUs. If the NetBurst architecture is supposed to get to 10 GHz, the integer ALUs are going to need to fly at 20 GHz.

Another current decision that should shine in the future is the quad pumped FSB. The Pentium III only got 1.06 GB/s bandwidth from its FSB. In comparison, the P4 gets 4.2 GB/s—the most of any current processor by a large margin. With new memory technologies that can utilize this bandwidth currently in the works, and new multimedia applications that want to use it, Intel is sitting pretty as far as bandwidth is concerned.

The new hardware automatic prefetcher that looks at prior reference patterns is also ingenious. It's exactly the kind of enhancement programmers like to see as it's going to improve code we've already written.

8.3 Caches

As I've mentioned before, booting the traditional L1 instruction cache in favor of an enhanced trace cache is probably one of the best things that could've been done to the IA-32 aging ISA. Intel followed the Amdahl-law rule of optimization, that is, optimize the thing that's taking the most amount of time, and the decoder was definitely taking a lot of time. The TC also is a very natural progression to an L1 instruction cache. If the processor is executing μ ops, why cache IA-32 instructions?

Of note is the fact that the branch predictor was enhanced right along with the TC enhancement. I would assume that this was absolutely necessary (and it looks that way too, since the branch target buffer is 8 times larger than that from the Pentium III) due to the fact that the TC stores decoded μ ops in execution order across (predicted) branches. Intel claims that in practice, the branch predictor used in the Pentium 4 is so good that mis-penalties due to an incorrect trace cache execution order are quite small.

The Pentium 4 L1 data cache is small (half the size of the Pentium III). Probably this decision has to do with the small latency of a small cache. It would be interesting to see an analysis of a larger L1 with higher latency and see how that effects performance. I don't doubt Intel's decision, I'm sure they did their own analysis. The latency concern is more viable after the switch to a 0.13-micron process in which Intel increased the L2 size, but not L1.

8.4 RAM

When Intel released the Pentium 4, they released the i850 chipset which only supported RDRAM. The reason I believe this was done, in part, was to promote Intel's own investment in RDRAM technology. The RDRAM advantage is two independent channels connecting the memory subsystem to the northbridge—that is twice the bandwidth of a similar technology in a single-channel design. With the new faster FSB (that could handle the increased bandwidth from RDRAM) and the fear of the P4 not performing as well as consumers would like to see, Intel had good reason to use RDRAM.

However, whenever the computer geeks aren't given a choice, they like to yell about it. And when their only choice is an



Figure 8: *The Intel Pentium 4, 2.0 GHz, 478-pin Processor.*

expensive one, they really yell. RDRAM is expensive and thus the problem and one reason RDRAM started showing up on lists of Intel blunders along with the Pentium division error.

I don't believe that Intel was simply trying to force the community to use their investment—I believe that they actually thought that RDRAM was a better technology, that prices would drop, and the Pentium 4 would be better off (why else would they give RAMBUS so much money?). The problem was that the prices didn't fall and the performance wasn't that much better. Intel ended up punting and releasing a chipset to support SDRAM, as other chipset makers did the same. Now dual channel DDR-SDRAM is competing with RDRAM and Intel has publicly committed to supporting DDR. I'm sure that this decision comes only after realizing that dual channel DDR will provide bandwidth similar to RDRAM—which is really what Intel was looking for in RDRAM in the first place.

8.5 Hyper-threading

Hyper-threading marks the beginning of Intel's move away from mining ILP (instruction level parallelism) in favor of the untapped TLP (thread level parallelism) on the desktop. Again, as was the trace cache, hyper-threading seems like a natural enhancement. If the processor is only being used 35% of the time, why not take a page out of the supercomputer manuals and get the execution units to do something else? Execution units become idle—usually—waiting on memory or disk. Having two threads in execution can allow another thread a shot at the execution units while one thread is waiting. If implemented correctly HT won't hurt, it can only help multi-threaded applications and multi-tasking environments (based on preliminary tests turning HT on in the new 3.06 GHz processor doesn't degrade performance in standard—single threaded—benchmarks). As programmers begin

coding for HT we'll see a more significant improvement in high-performance applications.

What's interesting is why Intel waited until now to unveil HT when it's been on the chip for some time. Intel had a problem with HT on a previous Xeon processor (degraded performance). I'm guessing Intel just wanted to make sure the same thing wasn't going to happen with the Pentium 4. Given the somewhat poor processor results at release in 2000, if when released, HT failed (again) it wouldn't be very good publicity for Intel.

9 A Note on Windows XP

Since the release of Windows 2000, Microsoft is no longer supporting MIPS and Alpha processors. [If you're reading this and are surprised that Windows OSs before 2000 ran on processors other than 80x86 architectures, you're not alone.] The decision was for performance reasons, so the OS could be tailored specifically for the IA-32 architecture. Windows XP in fact, was designed specifically for the Pentium 4 and Intel and Microsoft aren't shy about advertising this fact (sort of a you scratch my back I'll scratch yours thing). Particularly, XP uses the new SSE2 SIMD instructions and the fast FSB to its advantage. Windows, like the rest of the computing society, is becoming more 'multimedia-centric'. Microsoft's new DirectX 8 uses SSE2 and Direct3D even has a SSE2 specific math library. XP moved its GUI to GDI+ to make direct use of DirectX's enchantments and SSE2's 128-bit integer operation performance is exploited in XP's Encrypting File System. With further collaborations between Microsoft and Intel, the Windows+80x86 combination should prove to continue to be a more and more reliable solution for consumers.

10 Summary

Intel's Pentium 4 and NetBurst micro-architecture may have had a rocky start, however, NetBurst was designed for scalability and now, two years after its incarnation, that fact is proving to be true. New innovations such as the Execution Trace Cache, a quad pumped FSB, improved branch prediction, and hyper-threading illustrate that Intel won't be slowing down in the processor race anytime soon. Competition from AMD and consumer reluctance to overlook details will keep future advancement alive.

References

INTEL CORP. 2002. *IA-32 Intel architecture software developer's manual, Volume 1: basic architecture*. Available 3 December 2002: <http://developer.intel.com/design/Pentium4/manuals/>.

INTEL CORP. 2002. *IA-32 Intel architecture software developer's manual, Volume 2: instruction set*. Available 3 December 2002: <http://developer.intel.com/design/Pentium4/manuals/>.

INTEL CORP. 2002. *IA-32 Intel architecture software developer's manual, Volume 3: system programming guide*. Available 3 December 2002: <http://developer.intel.com/design/Pentium4/manuals/>.

INTEL CORP. 2002. *Intel Pentium 4 and Intel Xeon processor optimization reference manual*. Available 3 December 2002: <http://developer.intel.com/design/Pentium4/papers/>.

INTEL CORP. 2002. *Intel Pentium 4 processor with 512-KB L2 cache on 0.13 micron process datasheet*. Available 3 December 2002: <http://developer.intel.com/design/Pentium4/datashts/>.

INTEL CORP. 2002. *Intel Pentium 4 processor website*. Available 3 December 2002: <http://www.intel.com>.

RIST, O. 2001. *Windows XP on the Pentium 4 processor*. Available 3 December 2002: <http://cedar.intel.com/cgi-bin/ids.dll/topic.jsp?catCode=CVD>.

SHELBURNE, B. 1998. *A brief introduction to Intel 80x86 assembler programming*. COMP 255 Computer Organization Class Notes, Wittenberg University.

SHIMPI, A. L. 2000. *Intel Pentium 4 1.4GHz & 1.5GHz*. Available 3 December 2002: <http://www2.anandtech.com/showdoc.html?i=1360>.

SHIMPI, A. L. 2000. *Intel's NetBurst architecture – the Pentium 4's innards get a name*. Available 3 December 2002: <http://www.anandtech.com/cpu/showdoc.html?i=1301&p=1>.

SHIMPI, A. L. 2001. *Intel Pentium 4 2.0GHz: the clock strikes two*. Available 3 December 2002: <http://www2.anandtech.com/showdoc.html?i=1524&rndr=11182002094337>.

SHIMPI, A. L. 2002. *AMD's Athlon XP 2000+ vs Intel's 0.13-micron Northwood*. Available 3 December 2002: <http://www.anandtech.com/cpu/showdoc.html?i=1574&p=1>.

SHIMPI, A. L. 2002. *Intel introduces 533MHz FSB CPUs – Pentium 4 2.53GHz*. Available 3 December 2002: <http://www2.anandtech.com/cpu/showdoc.html?i=1615&p=1>.

SHIMPI, A. L. 2002. *Intel's Pentium 4 2.4GHz: taking the lead*. Available 3 December 2002: <http://www.anandtech.com/cpu/showdoc.html?i=1605&p=1>.

SHIMPI, A. L. 2002. *Intel's Pentium 4 3.06GHz: hyper-threading on desktops*. Available 3 December 2002: <http://www2.anandtech.com/cpu/showdoc.html?i=1746>.

STOKES, J. 2002. *Dual-channel DDR comes to the P4*. Available 3 December 2002: <http://www.arstechnica.com/wankerdesk/3q02/dual-ddr.html>.

VISIONARY. 2002. *Intel Pentium 4 2.4GHz review*. Available 3 December 2002: <http://www.vr-zone.com/reviews/Intel/P42400/>.

VOLKE, F., TOPELT, B., AND SCHEFFEL, U. 2002. *Single CPU in dual operation: P4 3.06 GHz with hyper-threading technology*. Available 3 December 2002: <http://www17.tomshardware.com/cpu/02q4/021114/index.html>