**NetBurst**

**x86**

Almost all modern consumer PCs are built with x86 CPUs. This nomenclature comes from the family of processors of which all modern x86 CPUs were born. Initially there was the Intel 8086, a 16-bit microprocessor that debut in 1978. The 8086 is like the big brother of the infamous 8088, the CPU used into the first IBM PC. The first descendant, the 80186 enjoyed most of its lifetime as an embedded processor, and was not widely used in PCs outside of the notable Tandy 2000. The 80286 is also considered as a descendant of the 8086. It was the first processor to prompt the x86 naming scheme, often simply referred to as the 286. The 286 provided possibly the large performance jump between generations of any single CPU in the x86 lineage, yielding performance numbers greater than twice that of the 8086. The 286 also increased the address bus from 20-bit to 24-bit, allowing it to address up to 16MB of RAM, as compared its predecessor’s limit of 1MB. Some of the advancements leading to this enormous performance include a dedicated module for effective address calculation, and improved ALU performance. The 286 was the first CPU to feature protected mode, which allowed the new memory limit to be addressed by the x86 architecture while still maintaining backwards compatibility with the 8086. Despite its advantages, the 80286 implementation of protected mode was not widely used.

It was not until the 32-bit 80386 was released in 1985 that protected mode became viable. With the addition of technologies such as paging, virtual real mode, and the ability to dynamically switch back into real mode without resetting the CPU, protected mode became a powerful advantage of x86 processors, prompting several third-parties to develop clones of Intel’s increasingly popular architecture. This trend, along with those of performance increase, continues to grow in large chunks through the different models of 386 and 486 processors.

With the i486 in 1989, Intel set many milestones, including the first tightly-pipelined x86 CPU as well the first x86 model to use more than 1 million transistors. This impressive total is due to the existence of a FPU and cache on-chip, as opposed to the separate-socket configuration of previous designs. The i486 eventually reached 100MHz but increasing memory requirements and memory costs and brought its obsolescence.

For the same reasons that Intel called its 80486 the i486, their fifth generation x86 CPU departed completely from the previous naming scheme. The Pentium was born. Including a 60 MHz version with over 3 million transistors, the Pentium represented yet another enormous jump in performance, offering nearly double the processing power of the i486, clock for clock. At this point, competition had grown fierce and a clear standard had emerged. The previous trends of reverse engineering and clone-making begin to fade away as various companies develop their own solutions to compete with what is becoming a very powerful Intel line-up. Many of these attempts used the same sockets as the Pentiums to become low-cost drop-in alternatives to the more expensive brand name.

As the Pentium architecture aged and third-party offerings became more competitive, Intel released perhaps its most important architecture to date under the name Pentium Pro. Although the sixth-generation x86 CPU still used the Pentium brand, it was a stark contrast from its namesake. One of the biggest changes in the new architecture was the addition of decoders to change instructions into micro-operations (µOps) in order to increase the efficiency of pipelining. This allowed the individual µOps to be ordered in a way so that the individual pipeline stages, specifically the execute stage, to be active on nearly every clock cycle, greatly increasing performance. This, coupled with several of Intel’s various industry advantages further solidified their place as market leader. While competitors may have held advantages over the Pentium Pro in individual areas, no individual CPU manufacturer could topple the giant.

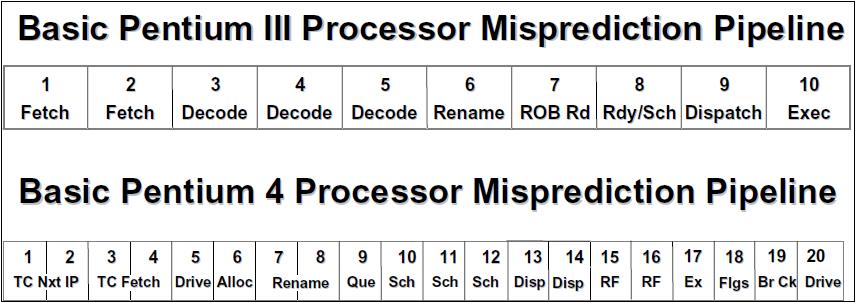
As AMD was developing its K6 CPU, the company acquired NexGen and released the K6 based greatly off the newly-acquired Nx86 design. Releasing this design in April 2007, it was marketed as a Pentium replacement, as it was designed to utilize the same platform as the P6 architecture. This, combined with the Pentium Pro’s underwhelming improvement over the original Pentium in 16-bit applications meant Intel needed a counterpunch. This time, they wouldn’t a new architecture. Intel released 4 new Pentium II cores and 4 new Pentium III cores, all based on the P6 architecture. Each of these cores added additional improvements, from new instruction sets to cost-saving off-chip cache innovations. The culmination of these CPUs was the .13 micron 1.4 GHz Pentium III Tualatin core, with performance rated over 1 GFLOPS.

**NetBurst**

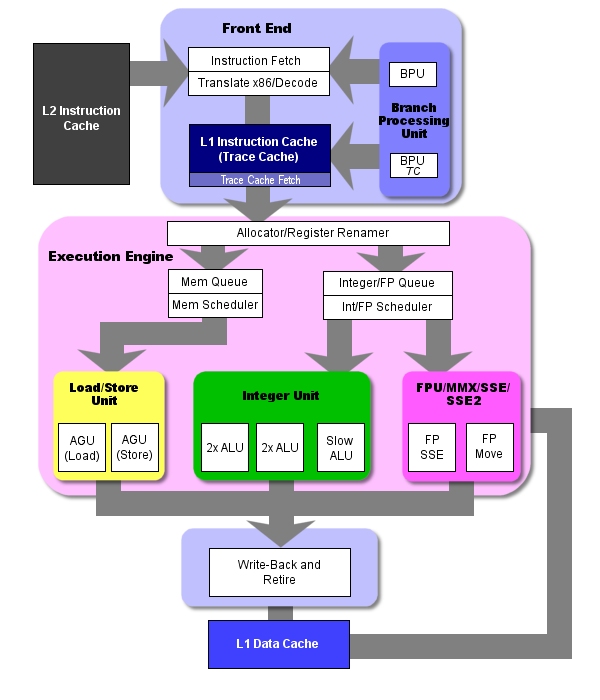
As the AMD Athlon began to significantly out-perform the extremely popular Pentium III, Intel decided it was time for a new platform. Retiring the now 5-year-old P6 architecture, Intel prepared to launch its seventh-generation of x86 CPUs, with several new features. In November 2000, Intel released it next architectural platform, known as NetBurst, with the Pentium 4 CPU as its flagship. With the introduction such technologies as hyper-pipelining, the rapid execution engine, SSE2 and SSE3, and an instruction trace cache, as well the new idea of simultaneous execution of different threads called hyper-threading, Intel was poised to once again revolutionize the x86 CPU market.

The first notable change brought about by the new micro-architecture was marketed under the name hyper-pipelining. This essentially refers to the fact that the Pentium 4 initially doubled, then tripled the number of pipeline stages in their CPU as compared to the Pentium III. While resulting in far fewer IPC--instructions per cycle (as they must pass through many more stages), it allowed the Pentium to reach speeds as high as 4 GHz. Intel even had plans to manufacture NetBurst CPUs clocked as high as 10GHz. These extremely fast clock rates were able to compensate for the reduced number of IPCs.

The NetBurst microarchitecture’s instruction execution can be viewed as three separate parts: the in-order fetch/decode front end, the out-of-order (OOO) execution unit(s), and the in-order retirement order. Notice that both the front and back ends are linear; this implies the highly-scheduled nature of NetBurst CPUs. For further examination of the execution pipeline:



The NetBurst’s Pentium 4 Northwood 20-Stage Pipeline [2]



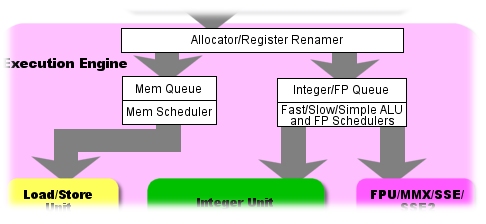
Northwood's Architectural Overview [6]

In this brand-new architectural scheme, the first five pipeline steps are:

1/2. TC Next Instruction—this two-stage process receives the next instruction pointer and feeds this information to the next step.

3/4. TC Fetch—this is the fetch/decode engine, which also takes up to of Northwood’s 20 stages. New to NetBurst, instructions are decoded into µOps and then cached in the Execution Trace Cache (TC). Caching µOps allows the scheduler to parallelize the single stream of data coming from the in-order fetch/decode process. [3] Using the enormous L1 cache (enough to store 12,000 µOps) to store previously decoded µOps as opposed to instructions that may never run allows the Northwood to eliminate the latency that previously existed in the decode process. It also provides new µOps in the event of a branch misprediction.

5. Drive—another first, this stage is simply a time delay to allow data to get where it’s going (also called a wire delay [4]). The extremely high clock speeds of NetBurst CPUs necessitate this step.



Pipeline Stages 6-12 [6]

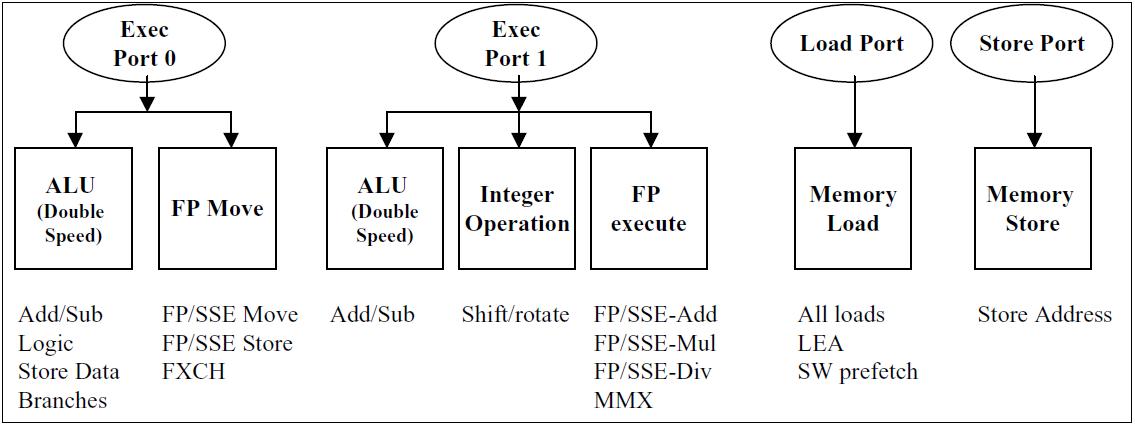
Steps 6 through 8 are the NetBurst implementation of the register renaming process which has become common in x86 as the amount of register available has exceeded the number which are specified in the instruction set. The unique part of this allocator/register renamer is the way it begins the parallelization within the CPU, issuing up to three µOps per cycle to be scheduled to the various execution units.

Step 9 facilitates this scheduling; it is known as the Queue. There are two queues, as you can see—one for memory µOps and the other for arithmetic µOps.

Steps 10 though 12 are where µOps actually get scheduled for execution; Intel explains these stages as follows:

The µOps schedulers determine when a µOps is ready to execute by tracking its input register operands. This is the heart of the out-of-order execution engine. The µOps schedulers are what allow the instructions to be reordered to execute as soon as they are ready, while still maintaining the correct dependencies from the original program. The NetBurst microarchitecture has two sets of structures to aid in µOps scheduling: the µOps queues and the actual µOps schedulers. [2]

There are four schedulers, the memory scheduler that schedules loads and stores for the memory unit, two arithmetic schedulers that feed the Rapid Execution Engine (REE) that will be discussed later, the FPU scheduler that handles SEE and MMX floating-point operations, and the standard ALU/FPU scheduler which schedules any floating point operations and arithmetic operations not handled by the other schedulers (such as shifts and rotates). These schedulers all utilize each of four of what Intel calls execution ports, which are paths to the various execution units, separated discretely as pictured here:



Dispatch ports in the NetBurst Microarchitecture [2]

Steps 13 and 14 represent the actual “dispatching” of µOps from the schedulers, through the dispatch ports and into the execution units. As many as six µOps can be dispatched per clock cycle, giving the Pentium 4 good parallelization through its execution core.

Steps 15 and 16 allow time for µOps to be loaded into register files to be processed.

Step 17 is the Execute stage; µOps get processed here. Remember that there are 7 discrete units working on as many as six µOps during this stage, including the REE.

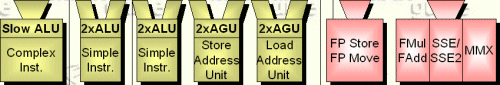
Step 18 is where the CPU sets any flags required by the executed instructions (jumps, etc.)

Step 19 is the Branch Check Stage. This is where the CPU determines whether its branch prediction was correct or not. The fact that Branch checking is done at stage 19 means that if the prediction was wrong, we just wasted 18 stages processes µOps that will be thrown out. This is far more than any other processor to date. This is where the fact that µOps are cache comes in handy—at least they’re already decoded and in L1 cache to be scheduled in the unfortunate effect we must backtrack.

Step 20—Drive. Let the signals catch up; we are running at clock speeds as fast at 4GHz, for crying out loud!

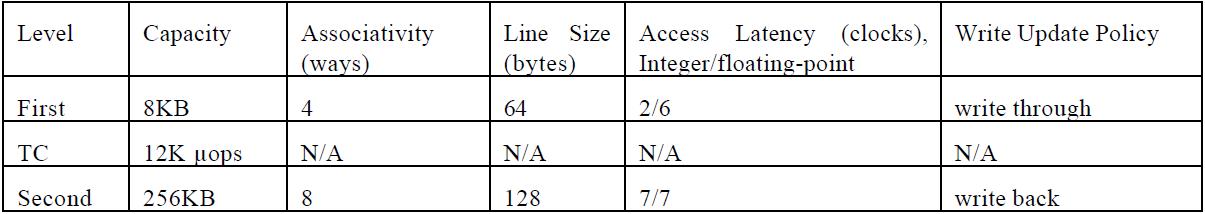
Both the front end and back end of the Pentium 4 can load/retire up to three µOps per cycle. The retirement engine holds responsibility for tracking branches and updating the Branch Target Buffer (BTB in the architecture diagrams), as well as containing the Reorder Buffer, which ensures that instructions are retired in the same order they were loaded, preventing errors and updating IP appropriately. In total, the Pentium 4 can support a massive number of total in-flight µOps, up to 126, with 48 load buffers and 24 store buffers. [2]

The next great innovation in NetBurst CPUs was the Rapid Execution Engine (REE). The REE was another technology Intel used to compensate for the low-IPC count of the Pentium 4. Most of the ALUs in NetBurst-era microprocessors operated at a frequency effectively twice that of the core clock, yielding extremely good integer performance. These “double-pumped” logic units received µOPs on each half-cycle, allowing them to very quickly process arithmetic operations, respective to competitor’s offerings in the same market. However, this led to unbalanced results, as other operations were very slow in relation. While a large number of possible instructions needed to be processed by execution units running synchronously to the CPU, the most commonly used (and simpler) instructions did in fact go through the faster ALUs. This is one of the first “innovations” that caused the Pentium 4 to earn mixed reviews (especially from programmers), as it made cross-platform optimization very difficult. Code that ran exceptionally well on an Athlon may crawl on the P4, and vice versa, respectively. In addition, to take advantage of the Pentium 4’s inclusion of SSE2, and eventually SSE3, Intel reallocated the MMX and floating-point units in a way that prove detrimental to specific performance domains, which resulted in the Pentium III to have better performance than its successor in some applications, mostly those reliant on the x87 FPU instruction set.



The NetBurst Rapid Execution Engine [1]

The Pentium 4 supports two main on-chip cache levels, with the L1 cache having separate portions for instructions and data:

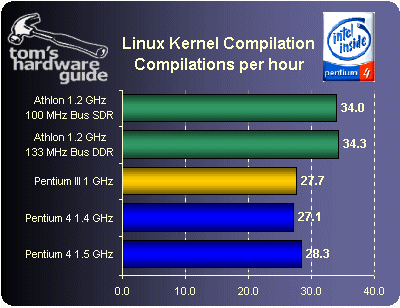
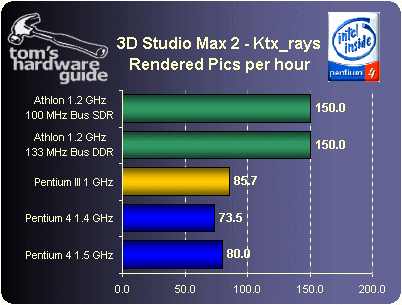
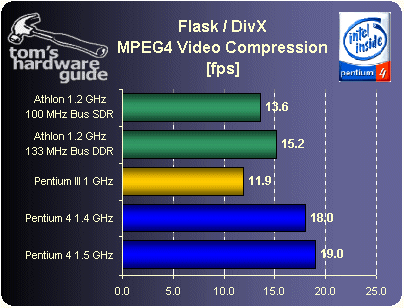
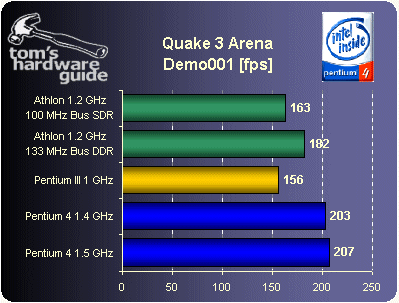


Pentium 4 Cache Parameters [9]

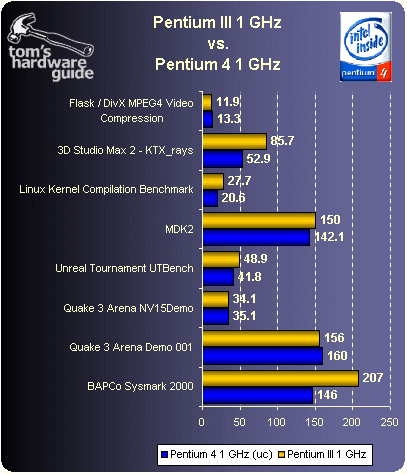
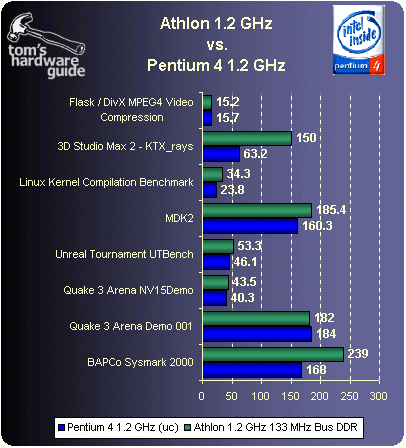
As can easily be seen, the L1 cache updates the next cache level on writes, while the L2 does not. This is mainly due to the extreme penalty paid when accessing the bus interface. Due to the scalar nature of the bus clock, the CPU is often 10-20 times faster than the front side bus, resulting in lost processor cycles of equal number to the processor clock multiplier. For example, a 1.8 GHz Pentium 4 running with a base FSB speed of 100MHz will suffer a penalty of 18 cycles on an L2 cache miss. In conjunction with the initial penalty of 18 cycles before a branch misprediction can be detected, accessing the external memory sub-system can greatly hinder performance.

**Performance Reviews**

In its time, the NetBurst architecture had many ups and downs. This was the period in Intel’s history which it experienced the most market pressure from competitors, to date. As can be observed from its internal structure and design, the NetBurst is very effective when dealing with simple ALU and floating-point operations, like those commonly used in multimedia encoding applications, as well as some specific game engines (Quake3 and Unreal). As the below benchmarks show, the Pentium does very well in tests using large numbers of non-x87 instruction set floating-point operations such as the DivX encoding test, or the Quake 3 performance, but the x87-based 3D Studio Max benchmark shows better performance from the Pentium III than its NetBurst-based successor. The poor performance in the Linux compilation test also demonstrates the impact of Intel’s architectural decision in NetBurst’s design.

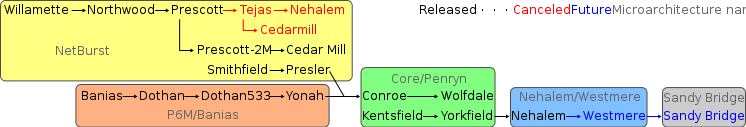


The next two benchmarks compare the Northwood-based Pentium 4 to both a Pentium III and an Athlon clock-for-clock at 1GHz and 1.2 GHz, respectively.



As can easily been seen, the Pentium 4’s design placed it into a specific domain in terms of performance, however, the market presence of Intel cause many software developments to adopt its optimization scheme, specifically those regarding SSE2 and SSE3, which is evident in the fact that later AMD offerings also included these SIMD instruction set optimizations.

**Failure**



Intel Roadmap, showing the NetBurst Failure [9]

As NetBurst CPU speeds climbed higher and higher, Intel run into an unexpected obstacle—heat. Heat and power consumption rose unproportionately to processor speed and prevent Intel from ever releasing any NetBurst CPUs clocked higher than 3.8GHz. This issue was an even bigger problem for mobile CPUs. The later core, Prescott, was far too hot and power-hungry to ever be a viable mobile CPU. In addition, the market shifted to dual-core processors. Intel’s first NetBurst dual-core offering, the Smithfield, was essentially two Prescott cores next to each other in the same die. After a die shrink to 65nm, the Presler (two Cedar Mill cores on the same die), did not yield satisfactory results, NetBurst CPUs began to disappear from Intel development roadmaps. One of the biggest factors in NetBurst’s abandonment was its inability to produce a viable mobile solution. Instead, Intel went back to its P6 core for mobile CPUs, eventually moving their desktop CPUs to this line also, in the form of the Core Architecture, producing the Core Solo, Core Duo, Core 2 Duo, Pentium Dual Core, and Core 2 Quad SKUs.

Although NetBurst had many new and innovative design implementations, in the end, heat, power consumption, and the far-improved performance of AMD CPUs caused the death of this very interesting microarchitecture. The NetBurst era produced six desktop Pentium 4 cores—Williamette (180 nm process, Socket 423/478, RDRAM/SDRAM, 400MHz FSB, 1.3 to 2.0 GHz), Northwood (130nm process, Socket 478, RDRAM/SDRAM/DDRAM, 400/533/800MHz FSB, 1.6 to 3.4 GHz), Gallatin (Extreme Edition, Socket 478, 800/1066MHz FSB, up to 3.73GHz), Prescott (90nm process, DDRAM, Socket 478/775, 533/800/1066MHz FSB, up to 3.8 GHz, Intel64, HTT, SSE3), Prescott 2M (Extreme Edition, Socket 775, 800/1066MHz FSB, up to 3.73GHz, Intel64, HTT, SSE3, EIST, VT), and Cedar Mill (65nm process, DDRAM, Socket 775, 800/1066MHz FSB, up to 3.8 GHz, Intel64, HTT, SSE3, EIST). While these processors always did very well in encoding applications, as soon in the performance section, their overall subpar performance, specifically in gaming and office applications, prompted Intel to rework their P6 into the more energy-efficient and more powerful clock-for-clock Core microarchitecture.**References**

[1] Thomas Pabst, “**Intel's New Pentium 4 Processor**”, tomshardware.co.uk [Online], Available: <http://www.tomshardware.co.uk/intel,review-262-8.html> [Accessed: November 22, 2008]

[2] Glenn Hinton, Dave Sager, Mike Upton, Darrell Boggs, Doug Carmean, Alan Kyker, Patrice Roussel, Desktop Platforms Group, Intel Corp., “The Microarchitecture of the Pentium 4 Processor”, intel.com [Online], Available: <http://download.intel.com/technology/itj/q12001/pdf/art_2.pdf> [Accessed: November 24, 2008]

[3] Jason Waltman, “Intel Pentium 4 and NetBurst Micro-Architecture”, jasonwaltman.com [Online], Available: <http://www.jasonwaltman.com/download/Intel%20Pentium%204%20-%20NetBurst.pdf> [Accessed: November 24, 2008]

[4] Peter N. Glaskowsky, “PENTIUM 4 (PARTIALLY) PREVIEWED Intel Lifts Veil on Hyperpipelined CPU—But Not All the Way”, virginia.edu [Online], Available: <http://www.cs.virginia.edu/~mc2zk/cs451/143501.pdf> [Accessed: November 24, 2008]

[5] Jon Stokes, “The Pentium 4 and the G4e: an Architectural Comparison: Part I”, arstechnica.com [Online], Available: <http://arstechnica.com/articles/paedia/cpu/p4andg4e.ars/6> [Accessed: November 25, 2008]

[6] Dave Altavilla, “Intel Prescott P4 3.2GHz and P4 EE 3.4GHz”, hothardware.com [Online], Available: <http://hothardware.com/Articles/Intel-Prescott-P4-32GHz-and-P4-EE-34GHz/?page=7> [Accessed: December 1, 2008]

[7] Unaccredited, Intel Corporation., “A Detailed Look Inside the Intel® NetBurst™ Micro-Architecture of the Intel Pentium® 4 Processor”, virginia.edu [Online], Available: <http://people.virginia.edu/~zl4j/CS854/netburst_detail.pdf>[Accessed: December 1, 2008]

[8] Imperator3733, WhiteTimberwolf, “IntelProcessorRoadmap.svg“ wikipedia.org [Online] , Available: <http://en.wikipedia.org/wiki/File:IntelProcessorRoadmap.svg> [Accessed: November 20, 2008]