# The End of an Era in Processor Evolution


## Abstract

Beginning with second generation superscalars, the continuous 10-fold-per-decade increase of processor efficiency leveled off for reasons discussed in the Introduction section. Designers responded by two fundamentally different approaches. The main road of processor evolution was marked by massively rising clock frequencies at up to a 100-fold-per-decade rate in order to sustain an approximately 100-fold-per-decade performance increase. The other approach strove to significantly increase processor efficiency by capitalizing on the EPIC style of computing. In Sections 2-4 of our paper we discuss both main approaches and point out the progress achieved as well as the constraints evoked.


## 1 INTRODUCTION

Let us first focus on the *performance of computer systems*[1] running *general purpose applications,* such as compilers, operating systems, office applications etc. In these environments, performance (P) can be characterized by the average number of instructions processed per second, which can be expressed as

$$P = f_c * IPC * \eta \qquad (1)$$

with:

- $f_c$: clock frequency,
- IPC: average number of instructions issued per cycle,
- $\eta$: efficiency of speculative execution, equaling the ratio of the total number of successfully executed (retired) instructions/total number of issued instructions.

In expression (1) we interpret *IPC * η* as the *effective IPC* ($IPC_{eff}$) or, in other words, the *effective width* or simply the *efficiency of the processor:*

$$IPC_{eff} = IPC * \eta \qquad (2)$$

[1] Although computer system performance depends on the features of a number of components, such as the processor, memory, hard disk, operating system and the compiler, it is typically the processor that has the largest impact to the resulting system performance in general purpose applications. Considering this, we use the terms "performance", "processor performance", "computer system performance" in our paper—admittedly somewhat imprecisely—as synonyms.

Then *performance* (P) yields

$$P = f_c * IPC_{eff}. \quad (3)$$

It is insightful to investigate the *integer performance growth* (reflecting actual performance in general purpose applications) computer manufacturers achieved in their major processor lines.

Intel managed to increase the integer performance of its x86 line immensely for more than two decades, *by about 100-fold each 10 years*, i.e. about two orders of magnitude per decade as shown in Figure 1. Other manufacturers achieved similar performance growth rates in their lines [1], [2]. However, such an impressive acceleration could not be sustained for a long time, and obvious signs of leveling off became apparent in the last few years (see Figure 1). Next we will discuss the sources and constraints of this tremendous performance increase.

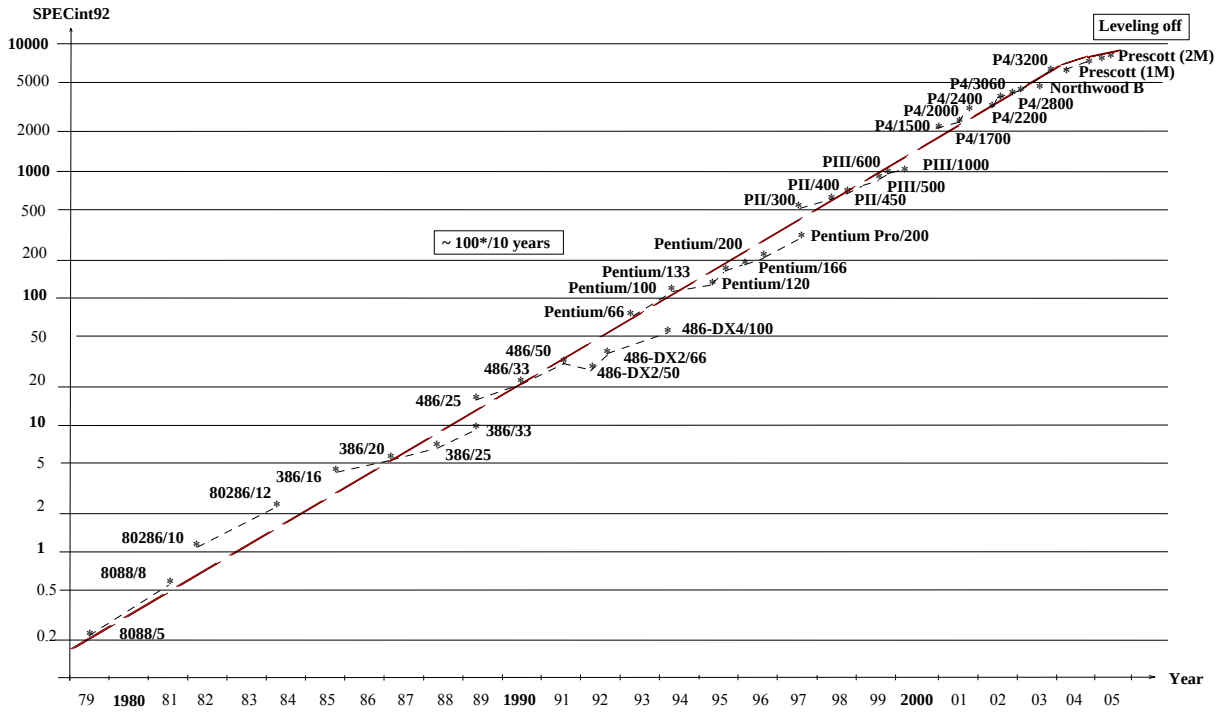



Figure 1: Historical growth of integer performance in Intel's x86 line of processors [3], [4]

According to expression (3), processor performance can be increased either by raising the *clock frequency* ($f_c$) or enhancing *processor efficiency* ($IPC_{eff}$). First let us discuss how processor efficiency contributed to increasing performance in general purpose applications. Again, we initially focus on Intel's widespread x86 line, followed by lines of other manufacturers.

As far as Intel's early processors are concerned, Figure 2 shows an *approximately 10-fold* increase in processor efficiency *each 10 years*. We note that in the figure processor efficiency is expressed in terms of SPECint_base2000/fc, where fc is given in MHz.

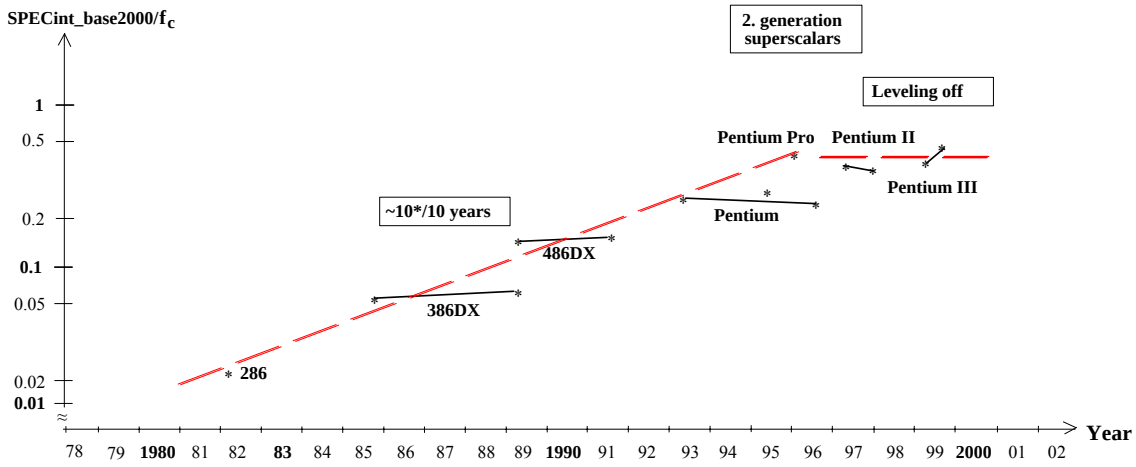



Figure 2: Processor efficiency growth in Intel's early processors [3], [4]

As far as the *sources* of increasing processor efficiency are concerned, we point out that Intel raised the efficiency of its early x86-processors basically in two dimensions, along the main road of processor evolution [5]: (a) by increasing the word length; and (b) by raising the extent of utilized instruction level parallelism (ILP). On the one hand, when upgrading its line from the 286 to the 386 DX processor, Intel extended the word length from 16 to 32 bits. On the other, the company made use of increasingly more ILP in four subsequent steps, in a fashion similar to other processor manufacturers, as follows:

(i) First, Intel introduced temporal parallelism in their first generation pipelined 386 DX;

(ii) Subsequently, they eliminated the bottlenecks inherent in pipelining due to increased memory bandwidth requirements and the inefficient processing of branches by means of introducing caches and branch prediction in their second generation pipelined 486DX;

(iii) After exhausting temporal parallelism, in a third step, the company made use of issue parallelism as well in their first generation superscalar Pentium; and

(iv) Finally, they removed the issue bottleneck of first generation superscalars by applying shelving (dynamic instruction issue) and capitalizing on the elimination of the issue

bottleneck by upgrading CISC cores from 2-wide to 3-wide models through introducing a number of advanced techniques, such as register renaming, reorder buffer (ROB), and a direct coupled L2 cache in their second generation superscalar Pentium Pro [5].

Here we point out with reference to Figure 1 and expression (3) that up to the appearance of the second generation superscalar Pentium Pro, efficiency and clock speed contributed nearly equally (roughly 10-fold per decade) to the approx. 100-fold-per-decade performance increase of the x86 family.

As far as *other processor families* are concerned, investigations reveal a *similar*, approximately 10-fold-per-decade processor efficiency growth rate [2]. This is quite evident as subsequent models of other manufacturers followed basically the same evolution path (first and second generation pipelined processors succeeded by first and second generation superscalars) as the x86 line [5]. The referenced data also indicate that up until the arrival of second generation superscalars, processor efficiency and clock rate contributed nearly equally to the performance increase in more or less all processor families [2].

The introduction of the second generation superscalar Pentium Pro, however, clearly marked *the advent of a new scenario* in the evolution of the x86 line, for two reasons. First, this 3-wide CISC processor—roughly equivalent to a four-wide RISC counterpart — already utilized nearly all instruction level parallelism available in general purpose applications, since Wall's cardinal investigations in the beginning of the 90's revealed that the available parallelism in general purpose applications usually does not exceed more than 4-8 RISC instructions per cycle [6]. Accordingly, a further widening of x86 cores would only have brought a diminishing return in efficiency for general purpose applications.

Secondly, beginning with the second generation superscalar Pentium Pro, an ever widening speed gap opened up between the processor on the one hand and the memory and I/O-subsystem on the other. For instance, the main memory was usually attached via a PCI 2.0 compliant FSB (Front Side Bus) to early first generation superscalar 60/66 MHz Pentium processors, clocked at the same frequency as the processor. The clock rate was increased already to 150-200 MHz in second generation superscalar Pentium Pro's, whereas their FSB's were still clocked typically at 60/66 MHz—that is, the same speed as the FSB's of early Pentium processors. Thus, beginning with the second generation superscalar Pentium Pro and fueled by rapidly increasing clock rates,

the inherently slow memory subsystem began to lag more and more behind the processor, increasingly impeding processor efficiency, as detailed later in Section 3.1. Accordingly, due to the cumulative effect of both reasons mentioned, the *efficiency* of the x86 line of processors *leveled off* beginning with the second generation superscalar Pentium Pro, as denoted in Figure 3 and discussed in more detail in Section 3.1.

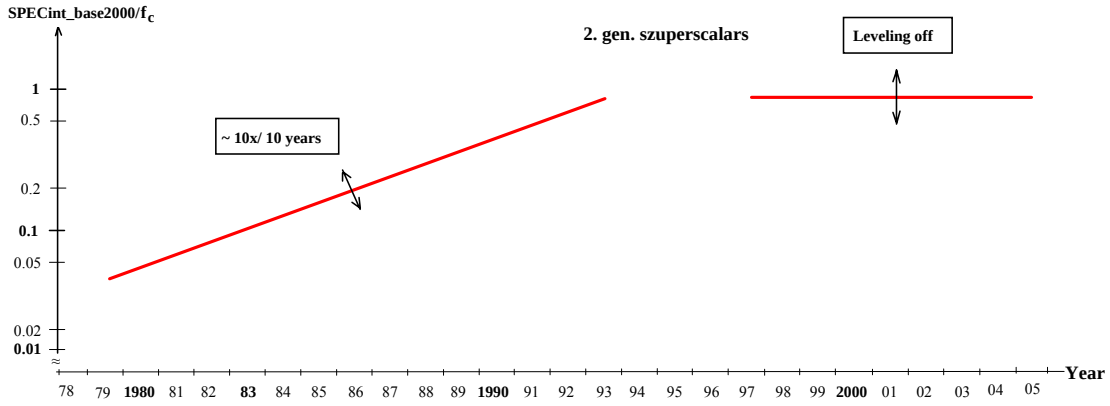



Figure 3: Historical growth of processor efficiency (in general)

Both of the above mentioned reasons for the leveling off in efficiency with Intel's x86 line, starting with its second generation superscalar Pentium Pro, are essentially valid for other families (RISC families in particular), too, beginning with second generation superscalar models, due to two key reasons: (a) second generation superscalar RISCs have usually 4 instructions/cycle wide cores (that is, cores roughly as wide as the 3-CISC-instructions/cycle-wide Pentium Pro); and (b) they typically operate at similar or higher clock frequencies as the Pentium Pro. So, *from second generation superscalars onwards*—whether of RISC or CISC style—microprocessor evolution entered a new era around the mid-90's, where *core efficiency* could no longer be increased further at the impressive previous rate of one order of magnitude per decade, but the growth rate *leveled off*, as indicated in Figure 3.

Designers addressed this crucial challenge using two approaches that result from expression (3): (a) *aggressively raising clock frequency* (fc); or (b) *essentially widening the processor core* (IPCeff) through introducing EPIC-style computing.

Our paper addresses these main routes of processor evolution and the consequences they brought about. The rest of the paper is structured as follows. Sections 2 outlines the main road of evolution, that is, aggressively raising clock frequency from second generation superscalars

onward in order to counter the leveling off in processor efficiency. Section 3 discusses the limits of this approach. Section 4 is devoted to the second approach to address the leveling off in processor efficiency, based on the introduction of EPIC-style computing, while Section 5 summarizes the conclusions drawn.

## 2 THE MAIN ROAD TO COUNTER THE LEVELING OFF IN PROCESSOR EFFICIENCY: AGGRESSIVELY RAISING CLOCK FREQUENCY

Obviously, the leveling off in processor efficiency can be compensated *by more intensely increasing* the other component of processor performance, i.e. the *clock frequency* than before, as expression (3) indicates. Intel spearheaded this approach, both by means of actively enhancing their fabrication technology and introducing their Netburst architecture, an architecture style conceived to facilitate high clock frequencies even beyond 10 GHz [7] and to serve as a basis for its Pentium 4 line of processors.

Basically, clock frequency can be raised either by *reducing* the *feature size* via scaling down the fabrication technology, or by *reducing* the *critical "length"* of the pipeline stages, i.e. the number of subsequent logic levels in a stage [8], [9] primarily by using longer pipelines. In order to achieve such a massive clock frequency growth, Intel made use of both options; not only did the company scale down the feature size by a factor of about 0.7 in every two years, but they also extensively lengthened the basic pipelines in subsequent cores from 10 in the Pentium III to about 20 in the Pentium 4 Willamette to approximately 30 in the Pentium 4 Prescott [10], [11]. As a result, Intel raised the clock frequency beginning with their second generation superscalar Pentium Pro until recently by an outstanding rate of about two orders of magnitude per decade, as shown in Figure 4.

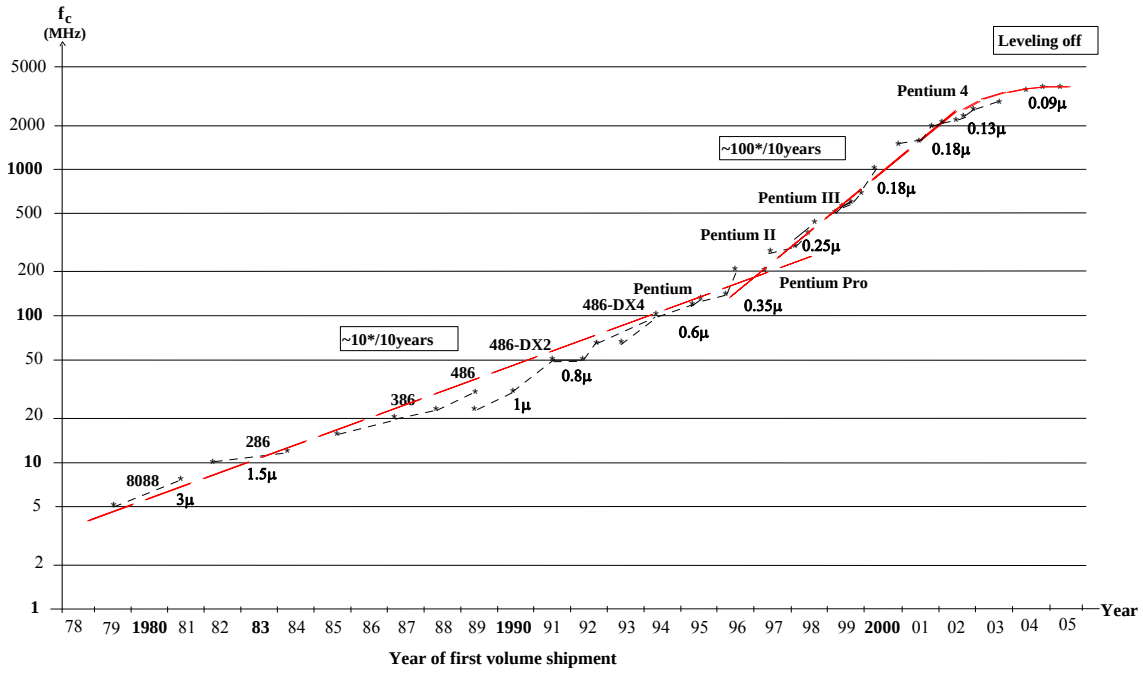



Figure 4: Historical growth of the clock frequency in Intel's x86 line of processors [4]

As Figure 4 shows, the *clock frequency* of Intel's x86 line evolved according to *three distinct patterns*. In the first period, roughly until 1995 (that is, until the debut of the second generation superscalar Pentium Pro), Intel raised fc approximately *10-fold per 10 years*. During this period, processor efficiency and clock frequency contributed equally to processor performance, as stated earlier. Subsequently, from the advent of the second generation superscalar Pentium Pro until the last few years, Intel raised fc by a breathtaking rate of about *100-fold per decade* in order to counter the leveling off in processor efficiency. Finally, this steep increase obviously *declined* in the last few years due to various reasons summarized in the next section.

However, the exceptionally hard race of chasing ever higher clock frequencies in the second period could not be followed by all competing processor manufacturers. As compiled data for different processor lines show, *clock rates of RISC processors could not be raised as massively as in Intel's x86 line* [12], [13]. The reason is quite straightforward: RISC cores were designed inherently as simpler but higher clocked ones compared to CISC cores. For instance, DEC's 21164 led the performance race in 1995 with a clock rate of 300 MHz, whereas at the same time Intel's Pentium Pro was clocked only about half this rate [12]. Obviously, it is a much more intricate task to significantly raise (e.g. to double) clock frequencies that are already considerably higher.

As a consequence, a *major shift* ensued *with second generation superscalars* in the performance race *between RISC and CISC processors*. While RISC processors were still leading the integer performance contest even in the mid 90's within just a few years the x86 line of CISC processors took the lead, as a survey of the integer performance growth of RISC and CISC processors covering the years 1995-2000 [12] reveals. For this reason and due to intensive efforts of market leaders Intel and HP to position EPIC processors (see Section 4) as being the potential next step of processor evolution, many processor manufacturers canceled their RISC lines in subsequent years—MIPS canceled their R series, HP their Alpha and PA families and the PowerPC Consortium their PowerPC line. Currently only two RISC families (IBM's Power line and SUN's UltraSPARC line) have survived to compete with Intel's recent Core line and AMD's K8-based families.

However, the aggressive boost of clock frequencies inevitably invoked intricate design problems in the GHz range, such as decreasing core efficiency, overwhelming power dissipation and increasing skew among different bit lines of parallel buses, as detailed in the following section.

## 3 LIMITS OF AGGRESSIVELY RAISING CLOCK FREQUENCIES

### 3.1 The processor efficiency wall

Processor evolution is hindered by the wellknown fact that neither the memory nor the processor bus can be sped up at the same high rate as the processor due to differences in their inherent operating principles. Therefore, processor evolution brought about a *widening speed gap between the core and these subsystems*—the wider the speed gap, the more it impedes processor efficiency. As a consequence, processor performance growth does not follow increasing clock rates linearly [14].

This implication forces designers to enhance the memory subsystem and the processor bus as intensively as possible. Consequently, processor efficiency in fact evolves as a result of two opposite effects: it automatically decreases for higher clock rates, while it improves as soon as the memory subsystem, the processor bus or other system components, such as the disk subsystem, the operating system or the compiler is upgraded. In this section we first present an overview of the sources of speed gaps, then show how the processor efficiency of widespread processor families evolved as a result of both opposite effects mentioned.

As detailed below, there are *three main sources* for a speed gap between the core and the memory subsystem: increasing memory latencies, increasing cache latencies, and lagging memory transfer rates for higher clock rates. In addition, a further speed gap can potentially arise with increasing clock rates between the processor and its relatively slow front-side bus (FSB). Next we will discuss these issues.

First, in Figure 5 we show how rapidly *chip level memory latency* rose with increasing clock frequencies. As Figure 5 indicates, at lower clock rates of a few tens of MHz, the chip level latency of FPM and EDO DRAMs (used e.g. in connection with Intel's early processors, such as the 386 or the 486) still only amounted to a few clock cycles. But with clock frequencies rising beyond the 100 MHz range, latency already became tens of clock cycles. Although designers continually improved memory performance by introducing more and more advanced memory technologies (such as SDRAM, RDRAM, DDR or DDR2) and also raised memory speed at the same time (e.g. from 66 MHz to 533 MHz and beyond), chip level memory latency grew steadily—in the GHz clock range it already reaches a level of well over 100 cycles, as Figure 5 shows.

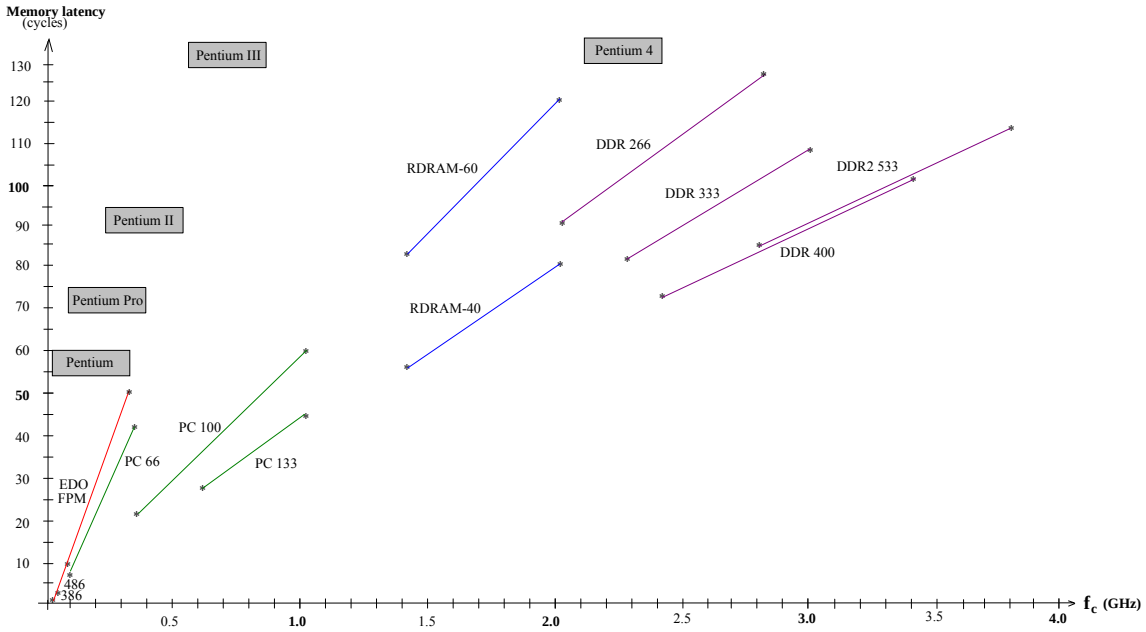



Figure 5: Chip level memory latency

(Latency figures are calculated assuming 150 ns latency for FPM and EDO DRAMs, a 3-3-3 memory timing for SDRAMs and 40 / 60 ns latency for RDRAMs, respectively)

*System level memory latencies* are roughly two to three times higher than chip level latencies in conventionally attached memory systems (where the memory is connected through the processor bus and the memory control hub, also called the "northbridge"). For instance, aggregated system level memory latencies in recent Pentium 4-based systems amount to 200–400 clock cycles [15]. In contrast, systems with on-die memory controllers such as those implemented in K8-based 64-bit AMD systems (Athlon 64, Athlon FX and Opteron) feature approx. 20-30% lower latencies [15].

Similarly, higher clock rates result in longer cache latencies (measured in clock cycles) at all levels (L1, L2 or L3). This problem is aggravated even further if subsequent cores of the same line include larger caches. For instance, cache sizes and relative latencies of Level 2 caches were increased in subsequent Pentium 4 cores as follows [10], [11]:

| | $f_c$ max at intro. (GHz) | L2 size (Kbyte) | L2 latency (clock cycles) |
|---|---|---|---|
| Willamette | 1.5 | 128 | 7 |
| Northwood | 2.0 | 512 | 16 |
| Prescott | 3.4 | 1024 | 23 |

Table 1: Latencies of L2 caches in subsequent cores of Intel's Pentium 4 line [10], [11]

As far as the *relative transfer rates of memories* (memory transfer rates related to the clock frequency) are concerned, Figure 6 indicates a less dramatic progress than that discussed for memory latencies. Nevertheless, despite immense efforts to enhance the memory subsystem through novel memory technologies, higher memory speeds or dual channel memory attachments, the relative transfer rates of memories remained more or less in the 0.2–0.4 range for higher clock rates, as Figure 6 indicates.

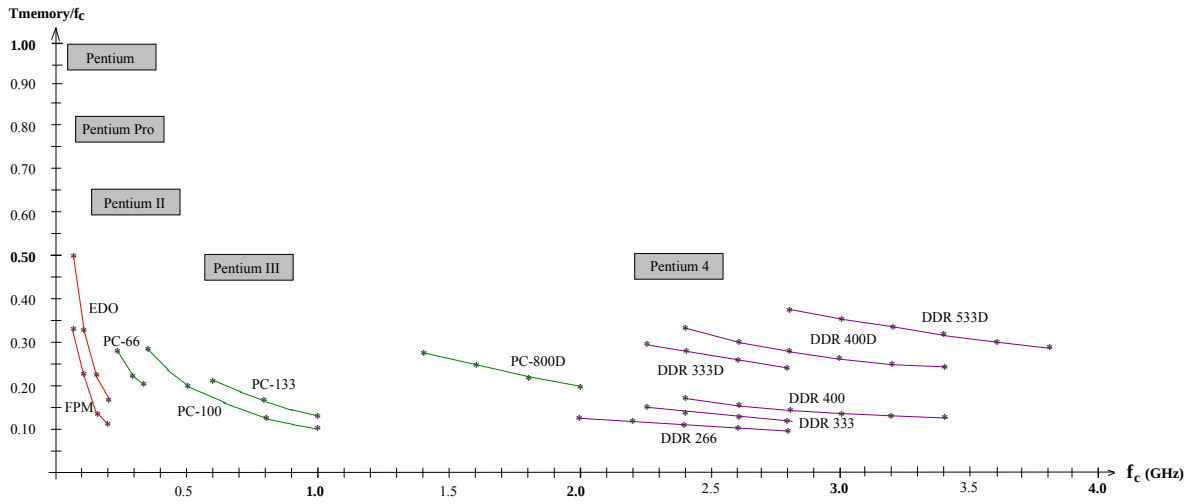



Figure 6: Relative memory transfer rates of subsequent memory implementations in Intel's x86 processor family (D: dual channel)

The *processor bus* (also designated as the front side bus (FSB) in conventional systems) is another subsystem whose speed has fallen clearly behind the core speed in the course of processor evolution. For instance, while both the core and processor bus were typically clocked at the same rate in earlier x86 processors, including the first Pentium models (60/66 MHz) subsequently, more or less along with the second generation superscalar Pentium Pro, the bus frequency has fallen more and more behind the clock frequency, as Figure 7 indicates for the x86 line of processors. Despite intensive efforts to raise the effective transfer rate through double-clocking and even quad-clocking the bus, actual transfer rates of processor buses in recent Pentium 4 cores amount to less than 30% of the clock frequency. The main reason for this huge gap between the clock rate of the core and the effective transfer rate of the bus is the skew occurring between different bit lines of a parallel processor bus, as detailed in Section 3.3.

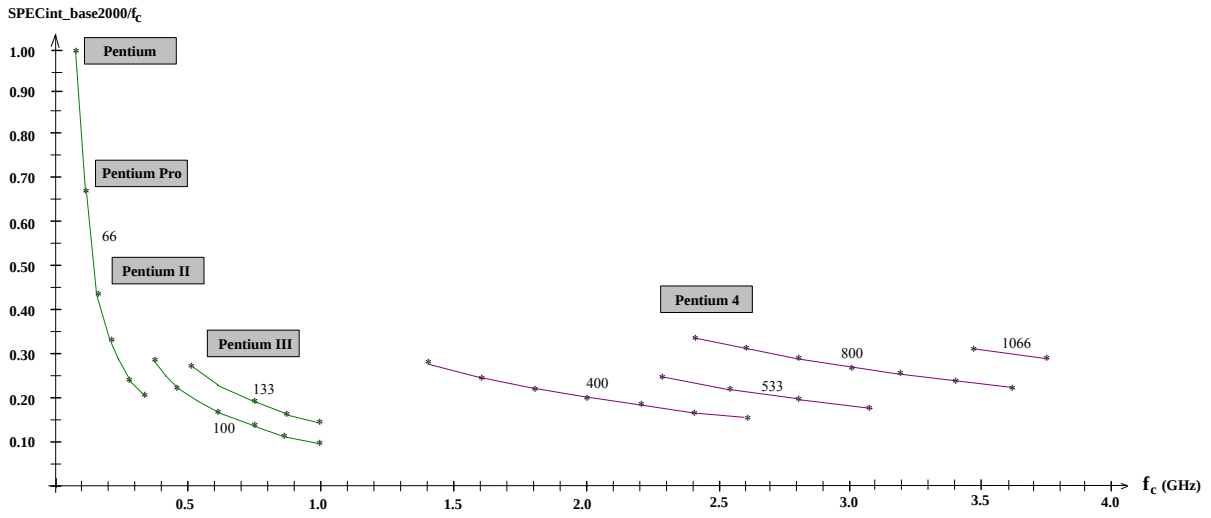



Figure 7: Relative speed of the processor bus in Intel's x86 line [3]

We note that *longer pipelines*, a technique often utilized to increase clock frequency, also impede processor efficiency, since lengthened pipelines give rise to more idle cycles ("bubbles") for mispredicted branches [14]. For instance, while the Pentium III generates only 10 idle cycles for a mispredicted branch, the Willamette and Prescott cores in the Pentium 4 family already cause 20 and 30 "bubbles", respectively [10], [11]. In our discussion of processor performance, the performance reduction caused by mispredicted branches is taken into account in expression (1) by the parameter η. The performance reduction caused by longer pipelines obviously calls for enhanced branch prediction in order to reduce or compensate for the impediments of longer pipelines [11], [14].

Due to the cumulative effect of all factors discussed so far, *processor efficiency* clearly *decreases for higher clock rates* in the GHz range (assuming the same design), as indicated in Figures 8 and 9 for Intel's and AMD's third generation superscalars. These figures also show the extent to which memory and bus subsystem enhancements contribute to higher processor efficiency.

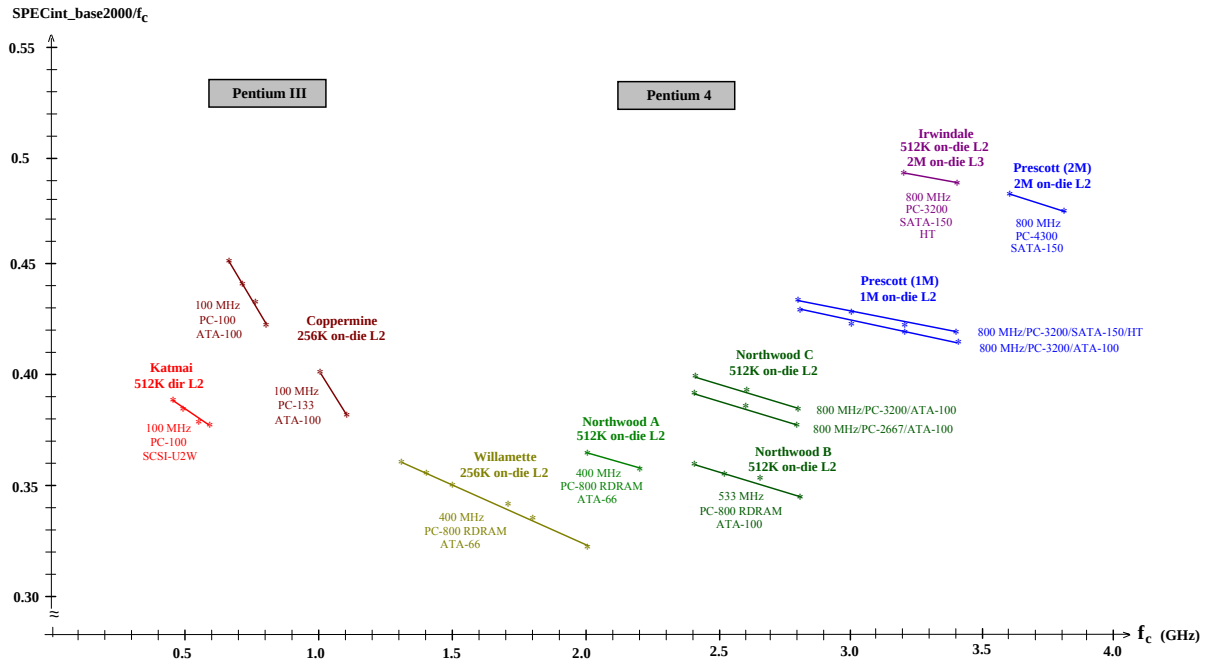



Figure 8: Efficiency of Intel' Pentium III and Pentium 4 cores [3]

Figure 8 reviews the *efficiency of subsequent Pentium III and Pentium 4 cores* as demonstrated by SPECint_base2000 benchmark programs [3]. It also highlights the most critical system parameters impacting processor efficiency, such as the type and size of the L2 cache, the FSB frequency, the type and speed of the memory as well as the type and speed of the hard disk interface used. For details of the software environment we refer to [3]. As seen in the figure, core efficiency clearly decreases for higher clock rates, assuming the same set of parameters. For instance, while the clock frequency of Coppermine cores was raised by nearly 100% (from about 0.6 GHz to 1.1 GHz), their efficiency decreased by almost 20% (from approx. 0.45 to nearly 0.38), despite increasing the memory speed from PC-100 to PC-133. On the other hand, designers attempted to compensate the declining processor efficiency by enhancing the memory subsystem and the processor bus, first of all increasing the size of the L2 cache or speeding up the memory or the processor bus. As Figure 8 shows, these enhancements, first in the form of larger (1 or 2-MByte) L2 caches implemented in the Prescott and Irwindale cores, result in a remarkable efficiency increase, whilst raising the hard disk speed contributes only marginally to higher processor efficiency. Finally, we point out that hyperthreading (HT) contributes to higher core efficiency only slightly (about 10-20%) in general purpose applications, as indicated in Figure 8 for the Prescott (1 MByte) cores and in [16].

As far as *AMD's third generation superscalars* are concerned, Figure 9 basically confirms the same behavior as discussed before for Intel's cores. Concerning this figure, however, we make two comments. First, it is noteworthy that the Thunderbird core has a remarkably low core efficiency despite its 256 KB on-die L2 cache introduced in this model. The reason is a too narrow (only 4 bytes wide) datapath between the L2 cache and the Thunderbird core. After AMD eliminated this bottleneck in its subsequent Palomino core (by increasing the width of the respective L2 datapath from 4 bytes to 16 bytes—the same width as implemented in Intel's Coppermine cores), core efficiency increased markedly, while other parameters were slightly improved as well. Secondly, as Figure 9 indicates, Athlon 64 cores are remarkably efficient. Such a high efficiency could be achieved primarily by two innovations: first by attaching the memory directly to the core via a high-bandwidth (in both directions 3.2 Gbytes/sec) serial bus, called the HyperTransport bus, and secondly by using an on-die memory controller.

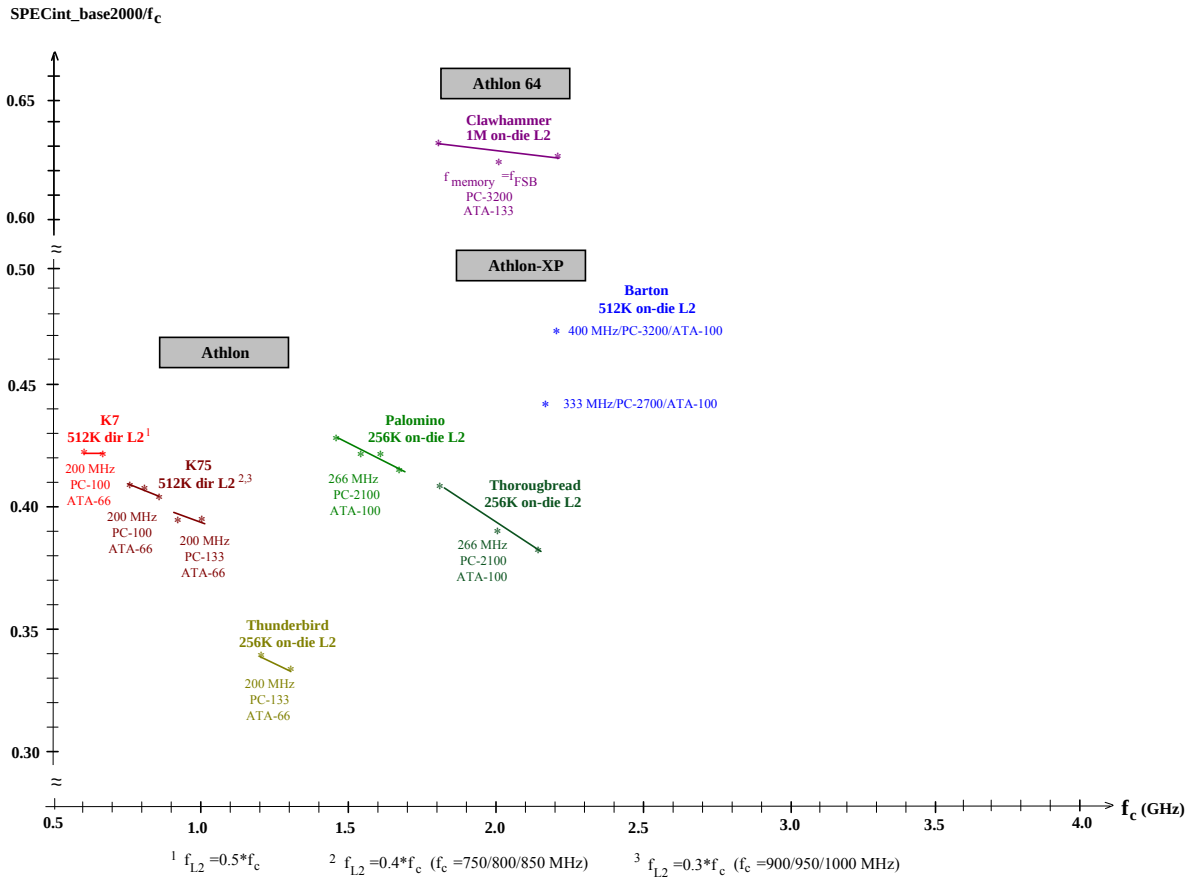



Figure 9: Efficiency of AMD's Athlon, Athlon XP and Athlon 64 cores [3]

At this point it is worth contrasting the *design philosophies* of Intel's Netburst architecture (underlying the Pentium 4 line) and AMD's K8 architecture (underlying the Athlon 64, Athlon FX and Opteron lines) with reference to Figure 10. Broken line segments in the figure indicate similar levels of performance in terms of SPECint_base2000 results. Figure 10 indicates that while the Pentium 4 line (and its underlying Netburst architecture) prefers clock rate over core efficiency, advanced AMD processors and the first incarnations of the Athlon 64 line (together with its underlying K8 architecture) already favor core efficiency over clock rate. For instance, the Athlon 64 achieves approximately the same performance (a SPECint_base2000 value of 1400) at 2.25 GHz as a Pentium 4 Prescott processor at 3.4 GHz. Considering the fact that higher clock rates not only induce decreasing core efficiency, but also give rise to serious additional design problems like increasing power dissipation and skew, as discussed in Sections 3.2 and 3.3, the design philosophy preferring clock rate has no future. This conclusion is confirmed by Intel's recent move to replace its Netburst architecture by the novel Core architecture based on the more conservative, mobile market oriented Pentium M design that favors core efficiency and particularly power efficient performance over clock rate [17] - [20].

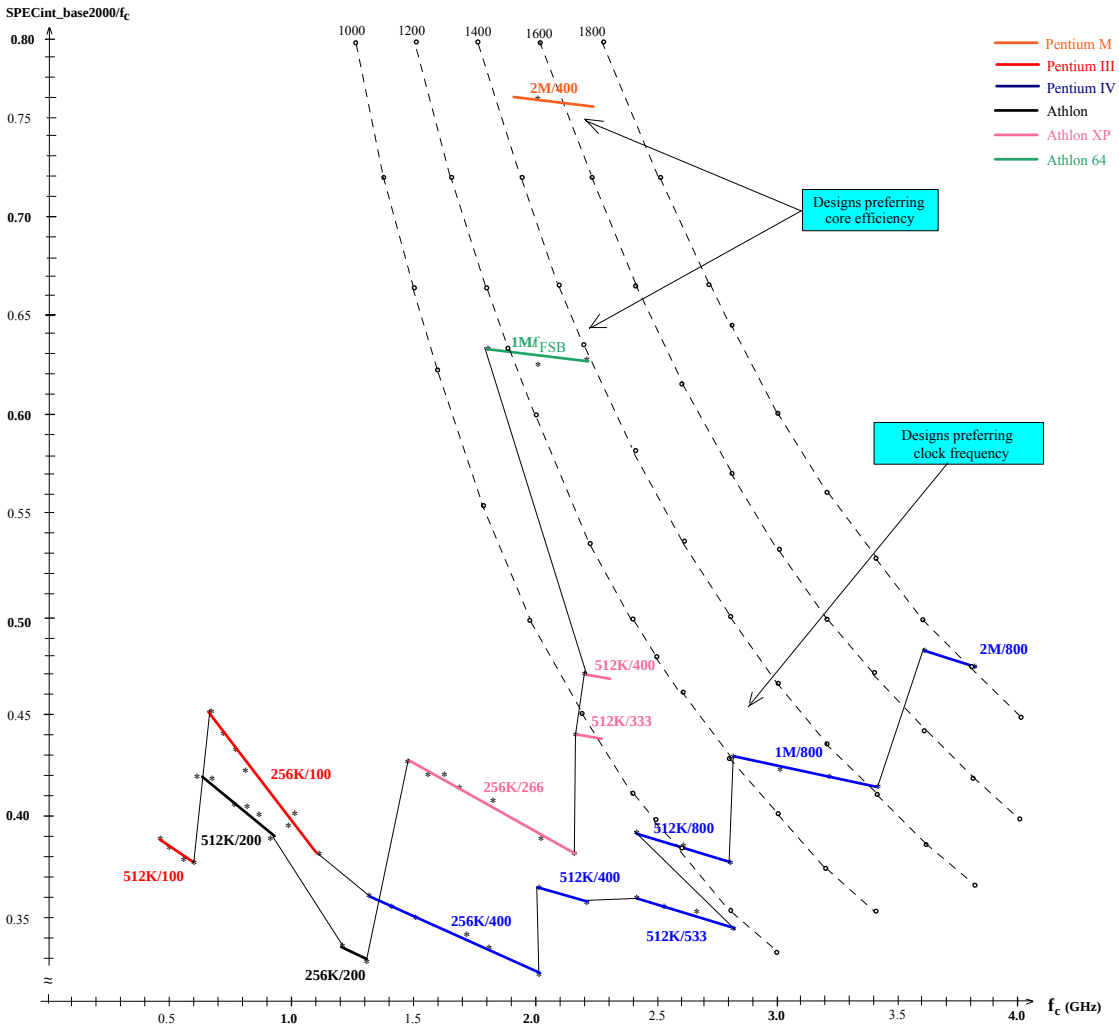



Figure 10: Contrasting Intel's and AMD's processor design philosophies

All in all, irrespective of the design philosophy chosen, higher clock rates lead to decreasing processor efficiency and thus to diminishing returns in performance, assuming the same design. Sooner or later (depending on the design philosophy practiced) a processor efficiency wall emerges with rising clock frequencies, increasingly limiting achievable performance gains.

### 3.2 The thermal wall

The amount of *thermal dissipation* (D) generated during the operation of a processor can be approximated as follows [21], [22]:

$$D = A * C * V^2 * f_c + V * I_{leak} \qquad (4)$$

with A: rate of the active gates,

C: effective capacity of the gates,

V: supply voltage,

$f_c$: clock frequency,

$I_{leak}$: leakage current.

As expression (4) indicates, the generated dissipation consists of two components: a dynamic part, representing the dissipation caused by charging and discharging the effective capacity of all active gates, and a static part, arising from the leakage current of all gates being in the off-state. Clearly, the dynamic part of the dissipation increases linearly with fc.

Despite intensive efforts to reduce power consumption e.g. through decreasing the supply voltage, recent microprocessors in the GHz range dissipate as much as about 100 W/cm$^2$, as indicated for Intel's x86 family in Figure 11 and for other families in [23]. For instance, the relative dissipation of the Pentium 4 Prescott core, announced with a clock frequency of 3.5 GHz, amounts to 100 W/cm$^2$, a very high value already causing intricate cooling problems (assuming air cooling). Thus Intel already approached the *thermal wall* with the Prescott core, as illustrated in Figure 12. As a consequence, Intel's x86 line could no longer sustain the extraordinary high (100-fold per decade) increase of clock frequencies in the 3 GHz range due to the emerging thermal wall. The company canceled the formerly announced successor of the Prescott core, called the Tejas core and its Xeon sibling, dubbed the Jayhawk core in May 2004 [24], and in October 2004 they withdrew from launching 4 GHz processors in the near future as well [25], despite their 2001 expectation to exceed the 10 GHz clock rate mark during the lifecycle of the Pentium 4 family [7]. Subsequently in 2006 the company replaced their Pentium 4 line by the Core family [18], [19], as already mentioned in the previous section. Thus, the era of intensively raising clock frequencies in Intel's Pentium 4 line—and more or less in all superscalar lines—hit a dead end.

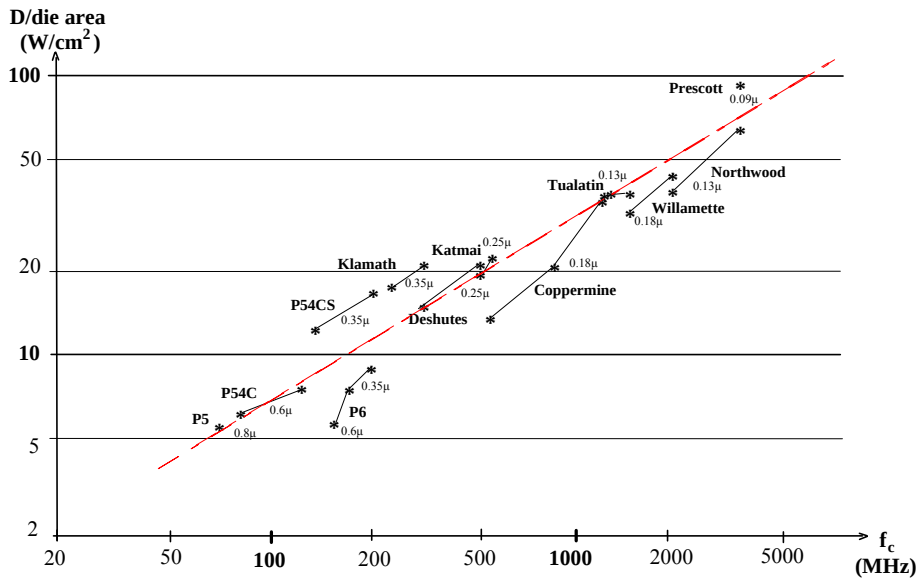



Figure 11: Relative dissipation of Intel's x86 family of processors [4]

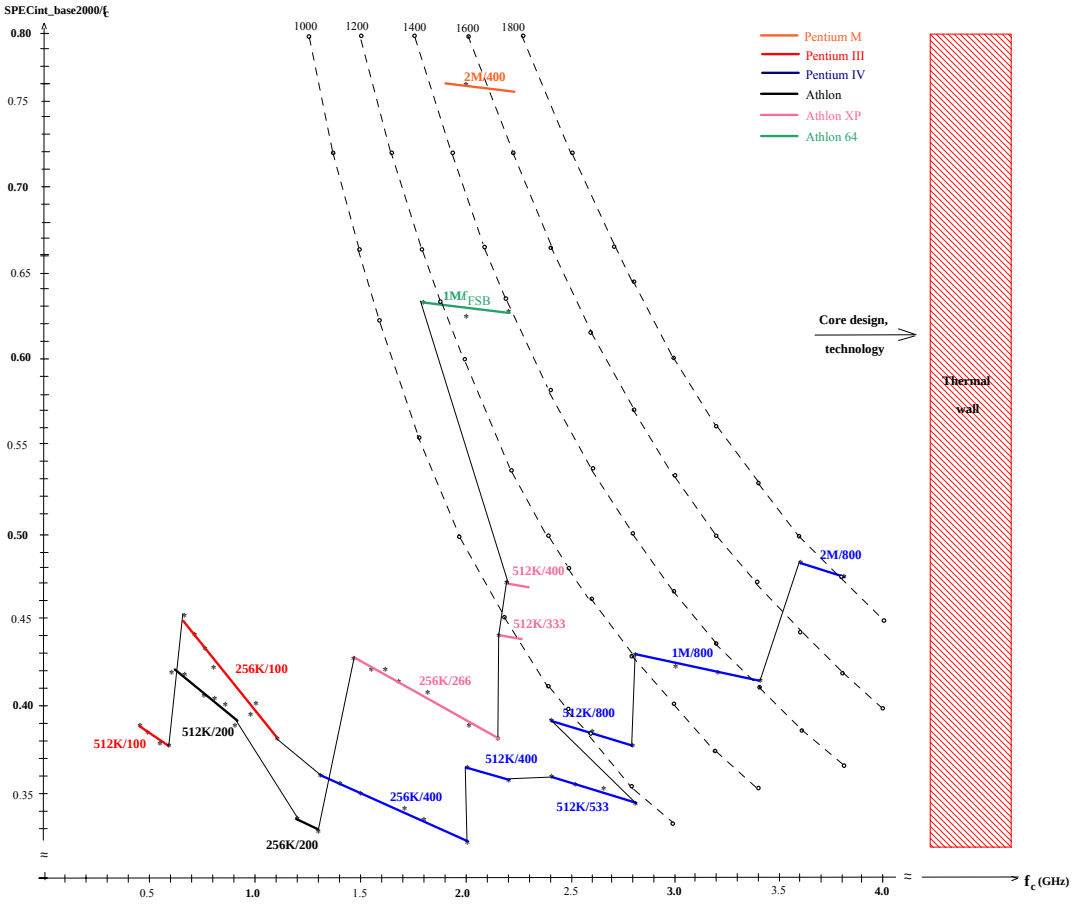



Figure 12: Contrasting the evolution of Intel's and AMD's processor lines with the thermal wall

Increasing dissipation with higher clock rates becomes even more detrimental for dual or multicore processors, representing the next era of processor evolution (see Section 5). All in all, we have witnessed dramatic changes in the last few years as regards the approaches to processor development: instead of focusing on raising clock rates, designers now concentrate more and more on reducing the dissipation of the processors by using power aware techniques, such as switching off inactive parts, resting hot spots, reducing the clock frequency etc. [22], [26], augmented by improved processor cooling techniques utilized on motherboards such as the BTX form factor [27].

### 3.3 The skew wall

It is widely known that signals traveling on different bit lines of parallel buses are distorted by the time they arrive at the receiving end due to *skews* occurring between different bit lines (as illustrated in Figure 13) and *noises*, such as crosstalk between the lines and external interference. *Skews* arise because bit lines occasionally have different lengths and electrical parameters, such as lump capacities. When the bus frequency is raised, pulse width becomes smaller and smaller, thus skews, crosstalk and external interferences become more and more corruptive to the signal transfer.

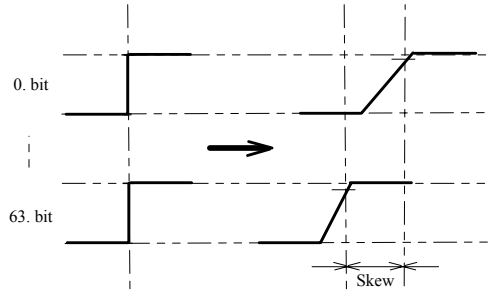



Figure 13: Skew appearing between different bit lines of a parallel bus

Despite the fact that motherboard designers concentrate more and more attention to equalizing bit lines of parallel buses for higher clock frequencies, as illustrated in Figure 14, raising the effective transfer rates of parallel processor buses in the GHz range becomes an increasingly convoluted task. Parallel buses have recently begun to approach their limits, called the *skew wall.* Consequently, in order to raise transfer rates beyond the skew wall, parallel processor buses inevitably have to be substituted by high speed, scalable sequential buses. These buses make use

of two lines per bit and implement a differential low-voltage (a few hundred mV) signal transfer, as indicated in Figure 15.

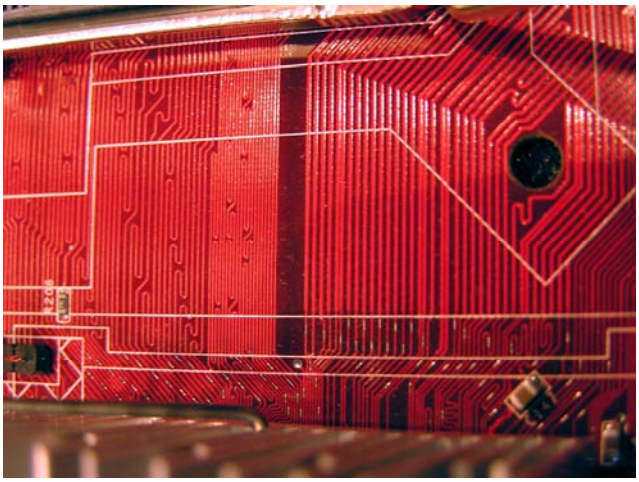


Figure 14: Equalizing skews between different bit lines of the processor bus on the MSI 915G Combo motherboard

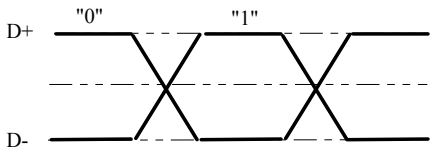



Figure 15: Signal transfer over a sequential bus

AMD pioneered sequential processor buses as the company made use of the HyperTransport bus in its 64-bit processor families (Opteron, Athlon-FX and Athlon-64) in April 2003.

We point out that easily scalable, inherently high speed (in the range of Gbit/s) sequential buses have already found their way into peripheral buses for years (e.g USB, PCI-Express, SATA, SAS etc.), thanks to impressive cost savings achieved through their significantly reduced bus and connector widths.

### 3.4 The end of an era in processor evolution

As pointed out in Section 1, processor efficiency in general purpose applications leveled off with second generation superscalars, which subsequently triggered an aggressive, *nearly 100-fold boost of clock rates* per decade along the main road of processor evolution. However, this fascinating increase of clock frequencies *inevitably came to an end*, as indicated in Figure 16,

due to serious design problems caused by decreasing processor efficiency and increasing dissipation as well as skew, as discussed in previous sections.

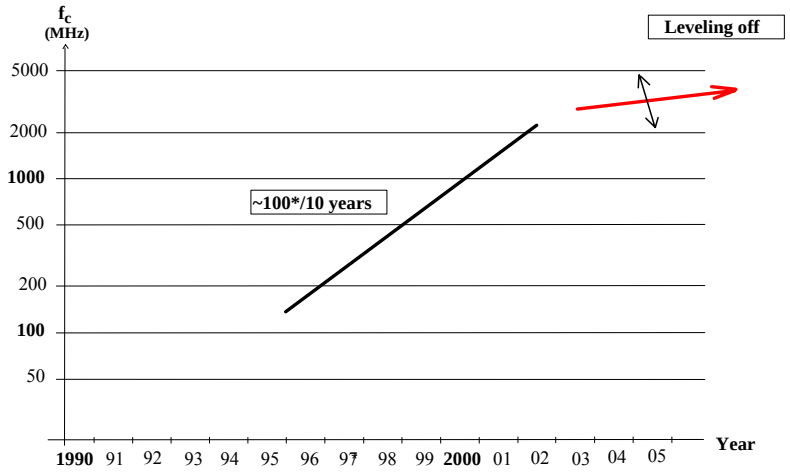



Figure 16: Recent evolution of clock frequency

## 4. THE ALTERNATIVE APPROACH TO COMBATING THE LEVELING OFF IN PROCESSOR EFFICIENCY – INTRODUCING EPIC ARCHITECTURES

As expression (3) indicates, the *leveling off* in processor efficiency can also be countered by striving to *significantly enhance processor efficiency* ($IPC_{eff}$). This option, however, requires a new architectural approach with a potential for processor efficiency that is considerably higher than that provided by superscalar processing. A number of computer manufacturers, most notably HP and IBM, became active at the end of the 1980's in developing a novel, more efficient architecture in a planned time frame of five to ten years —although with different motivation—for their future systems. . Typically oriented to enhance the VLIW style of computing conceived in the 1980's [2], [28] - [32]with features coined mostly for advanced superscalars, these efforts include architecture projects like HP's PlayDoh and PA-Wide Word (PA-WW also termed as the SWS SuperWorkStation) initiatives as well as IBM's DAISY (Dynamically Architected Instruction Set from Yorktown) ISA and its implementation called the BOA (Binary translation Optimized Architecture) project [2], [28] - [32]. From the point of commercial implementation, the most notable activity was a joint development of HP and Intel, announced in June 1994. The first results of this project were presented at the Microprocessor Forum in October 1997, outlining the EPIC (Explicitly Parallel Instruction Computing) style of

computing, the IA-64 ISA and its first implementation, called the Merced core [2], [28]. The *EPIC philosophy* is based on the VLIW style of computing, augmented with advanced features of superscalars, such as instruction bundling, predicated execution, compiler control of cache hierarchies, data and control speculation [28]. Although the first IA-64 processor, the 6-wide Merced core was scheduled to enter the market in 1999, it was introduced with a considerable delay in May 2001 [4]. Concurrently, Intel designated its line of IA-64 processors the Itanium family. However, the Merced processor failed to impress the market, partly due to its longer than planned time to market, since its performance features were set to compete with processors from 1999 rather than from 2001. Based on the completely redesigned and greatly improved McKinley core, the next processor in this line appeared one year later to become the first member of the Itanium 2 processor family. Subsequently a number of enhanced models followed, as shown in Figure 17.

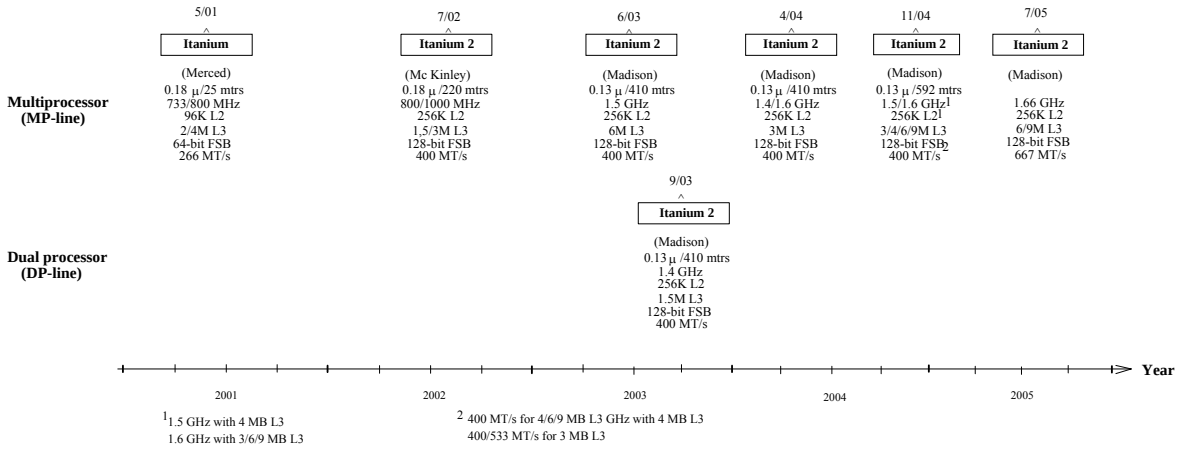



Figure 17: Main features of different models of the Itanium family

The features given in the figure are as follows: date of introduction, core designation, technology, transistor count, clock frequency, the size of the L2 and L3 cache, the width of the FSB (Front Side Bus) and maximum transfer rate of the FSB.

Let us now discuss how much the Itanium line achieved its *efficiency goals* for general purpose applications, with reference to Figure 18. As to the first Itanium core (the Merced), a comparison of efficiency data presented in Figure 8 for Pentium III processors and Figure 18 for Merced cores reveals the disappointing conclusion that Merced cores achieve basically the same efficiency at 800 MHz as Pentium III cores at the same clock rate. By contrast, Itanium 2 cores

with their huge L3 caches of 3–9 MBytes are considerably, roughly twice more efficient than comparable Pentium 4 cores, i.e. designs that prefer clock rate over core efficiency. Nevertheless, their efficiency figures are only insignificantly (about 10 to 20%) higher than those obtained for Intel's Pentium M and AMD's Athlon 64 superscalar cores, i.e. designs already preferring core efficiency over clock frequency, as a comparison of Figures 12 and 18 proves. For many reasons, including unimpressive efficiency figures for general purpose applications (as pointed out above), a lagging application base, the inconvenience of replacing legacy environments as well as the appearance of the upwards compatible 64-bit x86 processors (x86-64 from AMD and EM64T from Intel), former expectations for a rapid proliferation of the IA-64 platform (aka IPF (Itanium Processor Family)) [33] failed. As a consequence, overall revenue figures remained significantly below expectations [34]. Therefore—at least *for general purpose applications*—the alternative approach of introducing *EPIC style computing* in order to address the leveling off by significantly enhancing processor efficiency turned out to be an *unpromising course for the future*. Notwithstanding this conclusion, Itanium processors may still have a future in environments incorporating much more parallelism as general purpose applications offer, such as multimedia or server applications.

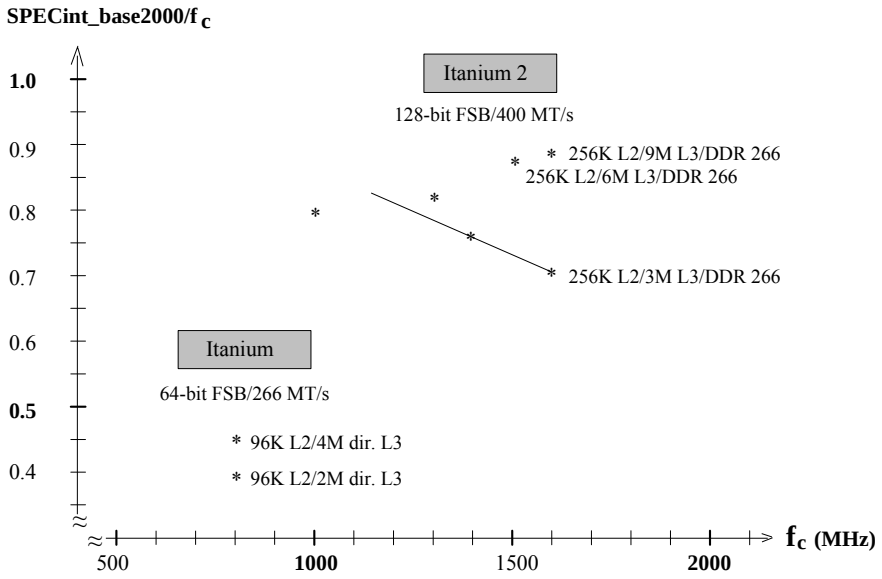



Figure 18: Efficiency of Intel's Itanium cores [3]

## 5 CONCLUSIONS

In the last decade, the main road of processor evolution was marked by an aggressive, approximately 100-fold-per-decade boost in clock frequencies, as discussed in Section 2. Such a rapid progress inevitably encountered its limits due to declining processor efficiency, increasing dissipation and skew in parallel buses, as discussed in Section 3. As a consequence, the course of massively raising clock frequencies came to an end. Also, the alternative approach to significantly raise processor efficiency by introducing EPIC style computing did not fulfill expectations—at least for general purpose applications—as pointed out in Section 4. Thus, a decade long era of processor evolution, heralded by third generation superscalars, ended in the last few years. The new era is characterized by principles such as power and core efficiency, more efficient use of available complexity (whose exponential growth follows further on Moore's Law), multicore and multithreaded designs [35], and introducing fast serial buses.

## REFERENCES


[1] L. Gwennap, "Processor Performance Climbs Steadily," *Microprocessor Report*, vol. 9, no. 1, pp. 17-23, Jan., 1995.

[2] J. Birnbaum, "Architecture at HP: Two Decades of Innovation," Microprocessor Forum, October 14, 1997, San Jose, California, http://www.hpl.hp.com/speeches/mpforum.html.

[3] Standard Performance Education Corporation, "SPEC CPU92, CPU95, CPU2000 results," http://www.spec.org.

[4] Intel Corp., "Microprocessor Quick Reference Guide," http://www.intel.com/pressroom/kits/quickref.htm

[5] D. Sima, "Decisive Aspects in the Evolution of Microprocessors," *Proc. IEEE*, vol. 92, no. 12, pp. 1896-1926, Dec. 2004.

[6] D.W. Wall, "Limits of Instruction Level Parallelism," *Proc. 4th Int'l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS '91),* pp. 176-188, April 1991.

[7] P. Otellini, "Beyond Gigahertz," *Intel Developer UPDATE Magazine*, pp. 1-7, Sept. 2001.

[8] M.S. Hrishikesh D. Burger, N. P. Jouppi, S. W. Keckler, K. I. Farkas and P. Shivakumar, "The optimal logic depth per pipeline stage is 6 to 8 FO4 inverter delays," *Proc. 29th Ann. Int'l Symp. Computer Architecture (ISCA '02),* pp. 14-24, May 2002.

[9] A. Hartenstein and T.R. Puzak, "The optimum pipeline depth for a microprocessor," *Proc. 29th Ann. Int'l Symp. Computer Architecture (ISCA '02)*, pp. 7-13, May 2002.

[10] G. Hinton, D. Sager, M. Upton, D. Boggs, D. Carmean, A. Kyker and P. Roussel, "The Microarchitecture of the Pentium 4 Processor," *Intel Technology Journal*, vol. 5, no. 1, pp. 1-12, Q1, 2001.

[11] D. Boggs A. Bakhta, J. Hawkins, D. T. Marr, J. A. Miller, P. Roussel, R. Singhal, B. Toll and K. S. Venkatraman, "The Microarchitecture of the Intel Pentium 4 Processor on 90nm Technology," *Intel Technology Journal*, vol. 8, no. 1, pp. 1-18, Feb. 2004.

[12] V. Agarwal, M.S. Hrishikesh, S.V. Keckler and D. Burger, "Clock Rate versus IPC: The End of the Road for Conventional Microarchitectures," *Proc. 27th Ann. Int'l Symp. Computer Architecture (ISCA '00)*, pp. 248-259, June 2000.

[13] AMD Inc., X86-64 Technology White Paper, Sunnyvale, CA, 2000 http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/x86-64_wp.pdf.

[14] E. Sprangle and D. Carmean, "Increasing Processor Performance by Implementing Deeper Pipelines," *Proc. 29th Ann. Int'l Symp. Computer Architecture (ISCA '02),* pp. 25-34, May 2002.

[15] Lavalys Inc, Everest Ultimate Edition, www.lavalys.com.

[16] N. Tuck and D. M. Tullsen, "Initial Observations of the Simultaneous Multithreading Pentium 4 Processor," *Proc. 12th Intel Conference on Parallel Architectures and Compilation Techniques,* pp. 26-34, Sept. 2003.

[17] S. Gochman, R. Ronen, I. Anati, A. Berkovits, T. Kurts, A. Naveh, A. Saeed, Z. Sperber and R.C. Valentine, "The Pentium M Processor: Microarchitecture and Performance," *Intel Technology Journal*, vol. 7, no. 2, pp. 21-36, May 2003.

[18] O. Wechsler, "Inside Intel Core Microarchitecture: Setting New Standards for Energy-Efficient Performance," White Paper, Intel Corp., 2006.

[19] R. Ronen, "Inside the Intel Core Microarchitecture," *11th EMEA Academic Forum,* May 2006, Dublin, http://www.intel.com/corporate/education/emea/af11/agenda.htm.

[20] E. Grochowski and M. Annavaram, "Energy per Instruction Trends in Intel Microprocessors," *Technology @ Intel Magazine*, vol. 4, no. 3, pp. 1-8, 2006, http://www.intel.com/technology/magazine/research/energy-per-instruction-0306.htm.

[21] N.S. Kim, T. Austin, D. Blaauw, T. Mudge, K. Flautner, J. S. Hu, M. J. Irwin, M. Kandemir and V. Narayanan, "Leakage Current: Moore's Law Meets Static Power," *Computer*, vol. 36, no. 12, pp. 68-75, Dec. 2003.

[22] R. Ronen, "The Thermal Wall: Where it came from and how to live with it?," *10th Intel EMEA Academic Forum*, May 2005, http://download.intel.com/corporate/education/EMEA/academicforum/keynotes/Ronen.

[23] R. Hetherington, "The UltraSPARC T1 Processor – Power Efficient Throughput Computing," White Paper, *Sun Inc*., Dec. 2005, http://www.sun.com/processors/whitepaper/UST1_pwr_v1.0.pdf.

[24] T. Krazit and T. Mainelli, "Intel Changes Planes for Pentium 4," *PC World*, May 07, 2004, http://www.pcworld.com/article/id,116053-page,1/article.html.

[25] K. Krewell, "Intel Cancels 4GHz P4," *Microprocessor Report,* vol. 18. no. 14, pp. 1-3, Nov. 2004.

[26] D.M. Brooks, P. Bose, S. Schuster, H. Jacobson, P. Kudva, A. Buyuktosunoglu, J. Wellman, V. Zyuban, M. Gupta and P. Cook, "Power Aware Microarchitecture: Design and Modeling Challenges for Next Generation Microprocessors," *IEEE Micro*, vol. 20, no. 6, pp. 26-44, Nov./Dec. 2000.

[27] Intel Corp., "Balanced Technology Extended (BTX) Interface Specification, Version 1.0a," 2003.

[28] J. Crawford and J. Huck, "Next Generation Instruction Set Architecture," *Microprocessor Forum,* Oct. 14, 1997, San Jose, California, http://www.hpl.hp.com/speeches/mpforum.html.

[29] M.S. Schlansker and B.R. Rau, "EPIC: Explicitly Parallel Instruction Computing," *Computer*, pp. 37-45, Feb. 2000.

[30] M. Smotherman, "Understanding EPIC Architectures and Implementations," 2001, http://www.cs.clemson.edu/~mark/464/acmse_epic.pdf-

[31] K. Ebcioglu and E. Altman, "DAISY: Dynamic Compilation for 100% Architectural Compatibility," *Proc. 24th Ann. Int'l Symp. Computer Architecture (ISCA '97),* pp. 26-37, June 1997.

[32] K. Ebcioglu E. Altman, M. Gschwind and S. Sathaye, "Dynamic Binary Translation and Optimization," *IEEE Trans. Computers*, vol. 50, no. 6, pp. 529-548, June 2001.

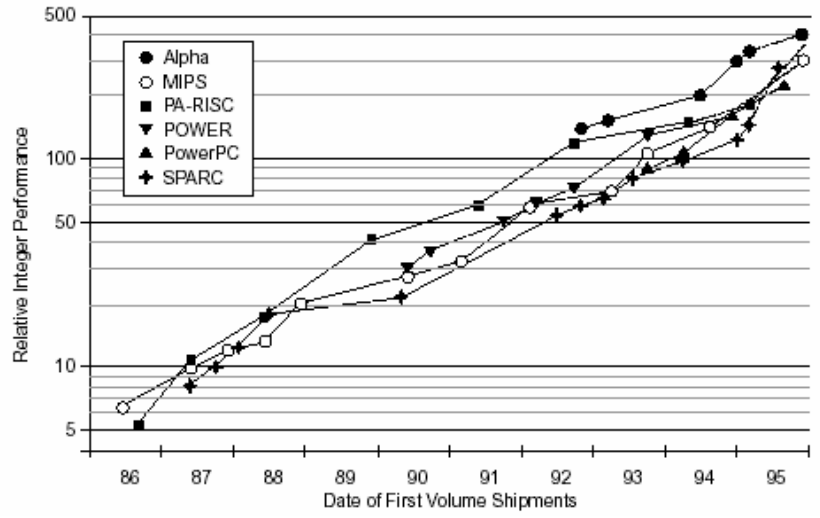
[33] L. Gwennap, “Intel’s Itanium and IA-64: Technology and Market Forecast,” *Microprocessor Design Resources*, 2000.

[34] G. L. Lancester, “Changing the Economics of Computing Thru Technology Leadership,” *IBM Forum* 2005, https://www-903.ibm.com/kr/event/download/ibmforum2005/day2/it3.pdf

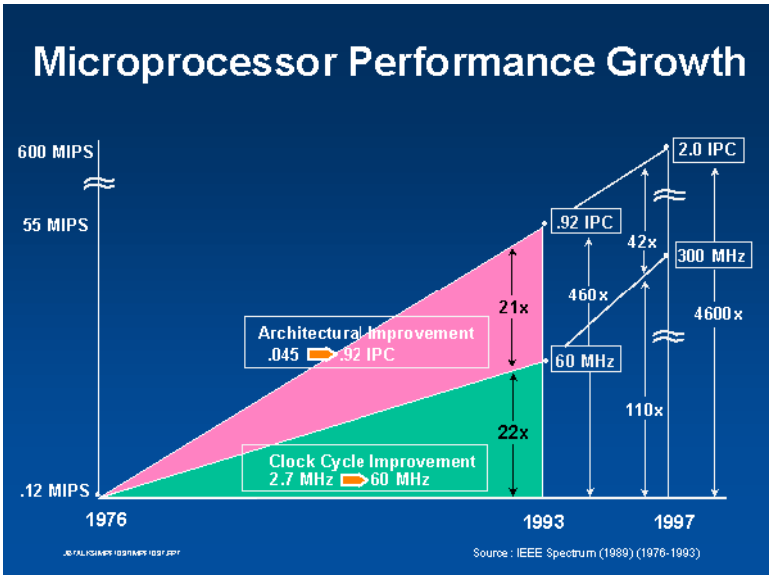
[35] D. Bhandarkar, “The Dawn of a New Era: Multi-Core Computing,” *11th EMEA Academic Forum*, May 2006, Dublin http://www.intel.com/corporate/education/emea/af11/agenda.htm

**Referenced figures**

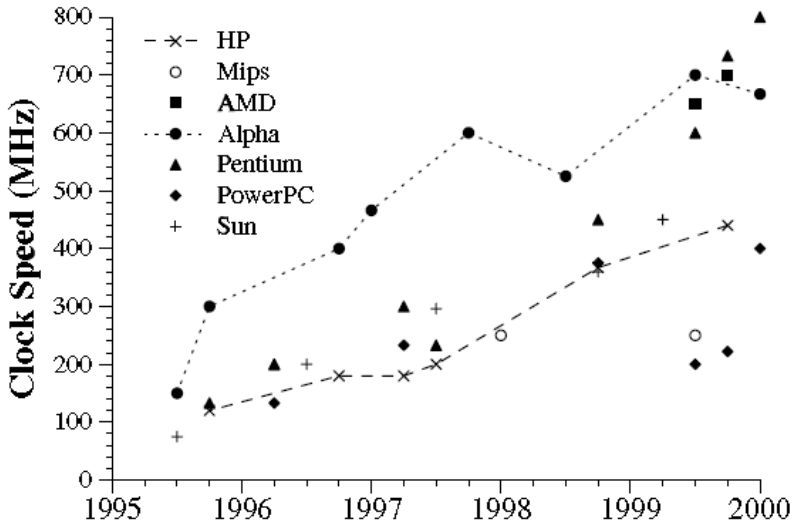
[1]: L. Gwennap, " Processor Performance Climbs Steadily," *Microprocessor Report*, vol. 9, no. 1, Jan. 23, 1995, pp. 17-23.



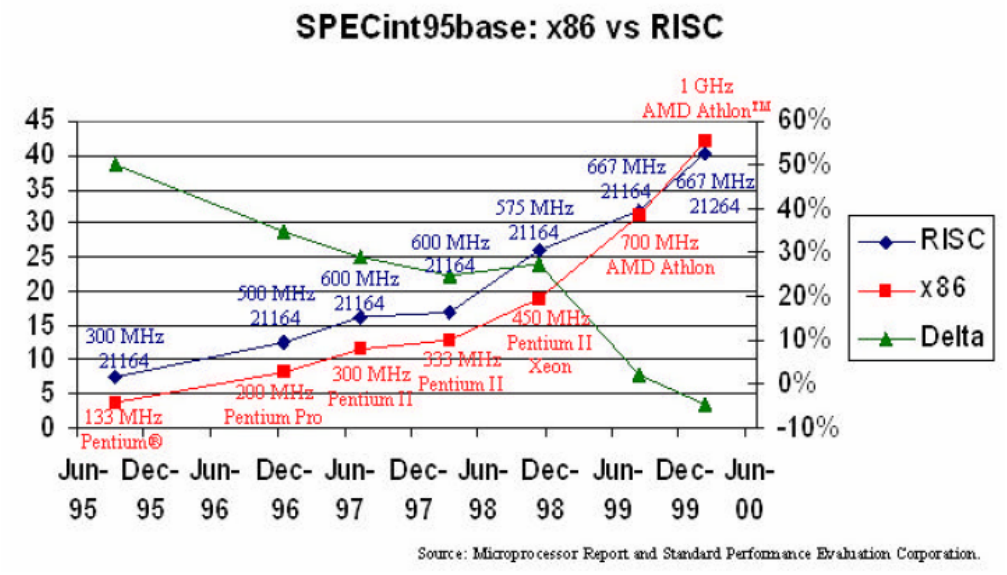

[2]: J. Birnbaum, "Architecture at HP: Two Decades of Innovation", *Microprocessor Forum*, October 14, 1997, San Jose, California, http://www.hpl.hp.com/speeches/mpforum.html.



[12]: V. Agarwal, M.S. Hrishikesh, S.V. Keckler et D. Burger, “Clock Rate versus IPC: The End of the Road for Conventional Microarchitectures”, *Proc. 27th ISCA*, 2000, pp. 248-259.




[13]: - X86-64 Technology White Paper, AMD Inc., Sunnyvale, CA, 2000, http://www.amd.com/us-/assets/content_type/white_papers_and_tech_docs/x86-64_wp.pdf.



[23] R. Hetherington, "The UltraSPARC T1 Processor – Power Efficient Throughput Computing', White Paper, *Sun Inc.*, Dec. 2005, http://www.sun.com/processors/whitepapers/UST1_pwr_v1.0.pdf.

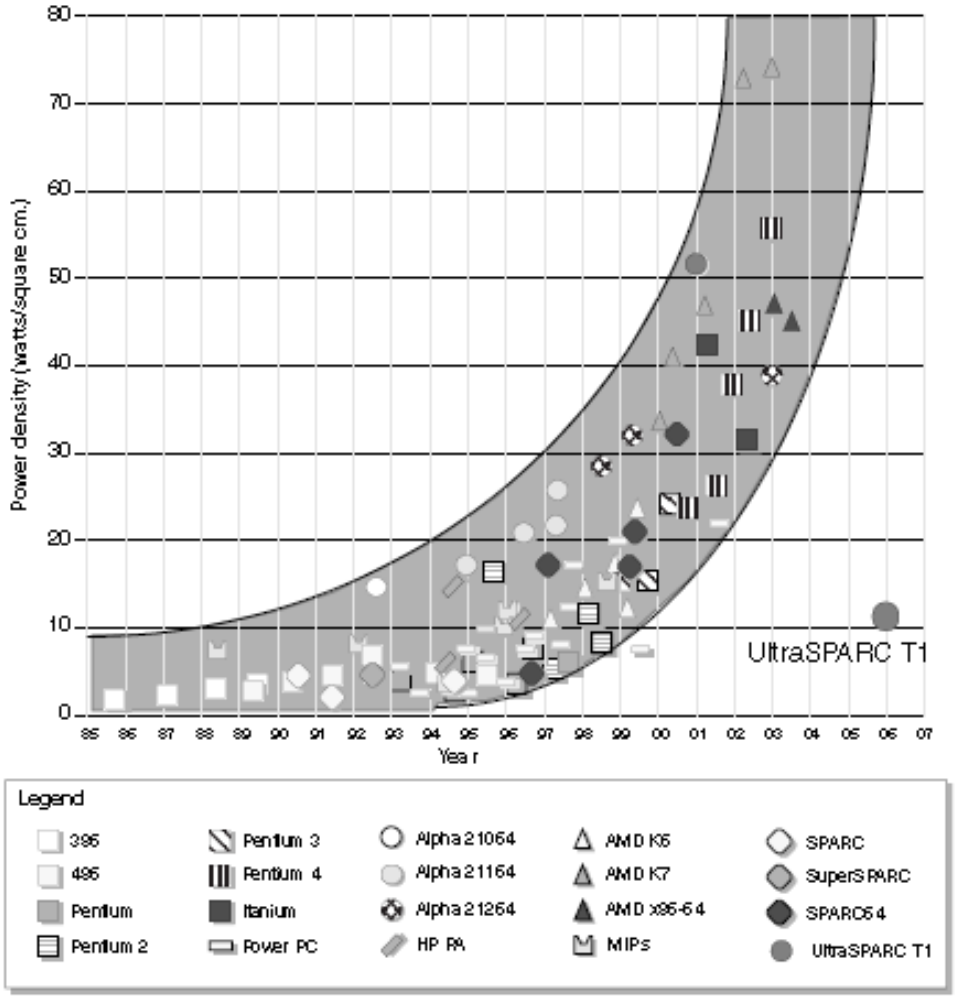
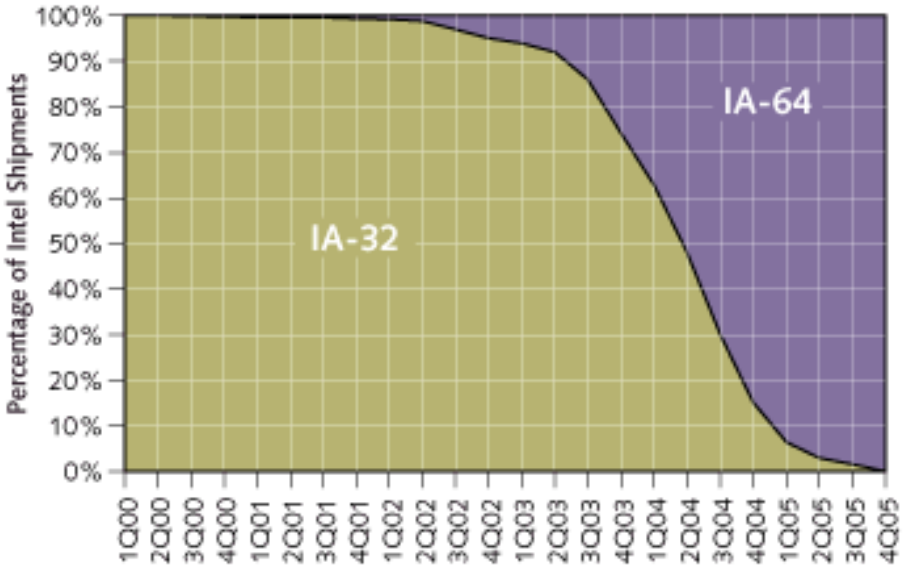
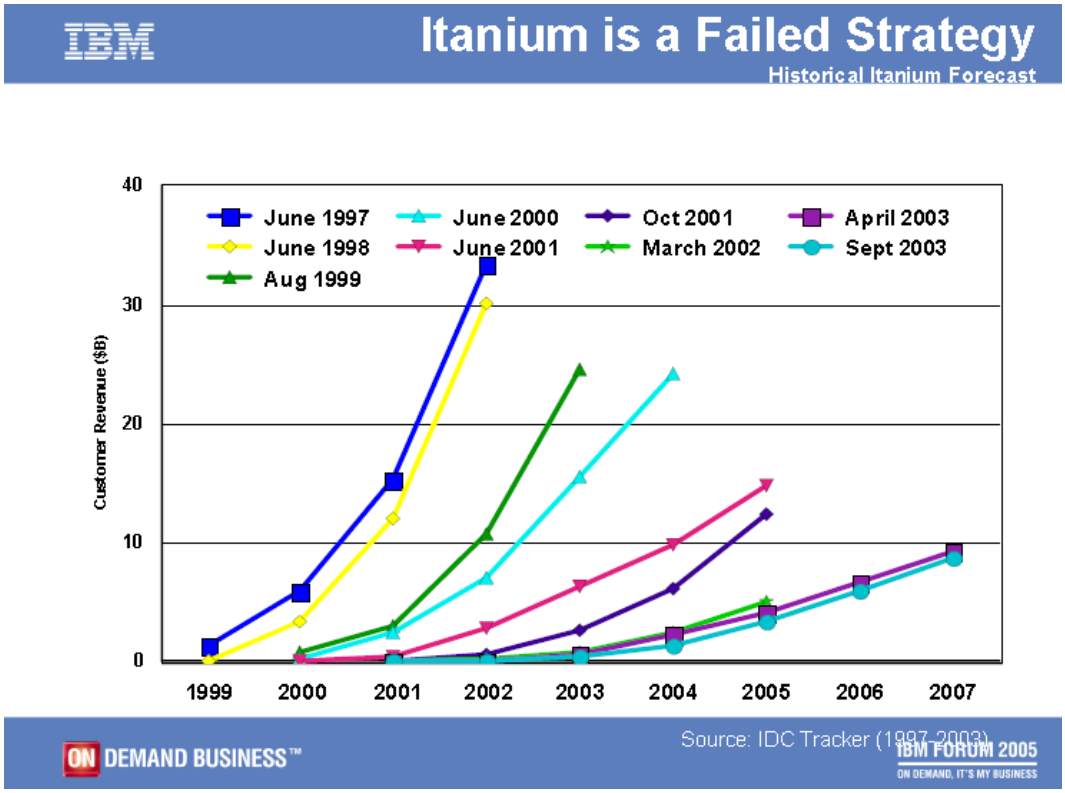



FIGURE 2 Power Density (W/cm²) Over Time

[33]: L. Gwennap, "Intel's Itanium and IA-64: Technology and Market Forecast", *Microprocessor Design Resources*, 2000.




[34]: G. L. Lancester, "Changing the Economics of Computing Thru Technology Leadership, *IBM Forum* 2005.



**Referenced data**

[15] — Everest Ultimate Edition, www.lavalys.com.

## Memory delay in systems

| | CPU | CPU Clock | Motherboard | Chipset | Memory | CL-RCD-RP-RAS | Latency |
|---|---|---|---|---|---|---|---|
| 54.4 ns | Sempron 2600+ | 1600 MHz | ASRock K8NF4G-SATA2 | GeForce6100 Int. | DDR400 SDRAM | 2.5-3-3-8 CR1 | 54.4 ns |
| 55.4 ns | Athlon64 3200+ | 2000 MHz | ASRock 939S56-M | SiS756 | Dual DDR400 | 2.5-3-3-8 CR2 | 55.4 ns |
| 56.2 ns | Athlon64 X2 4400+ | 2200 MHz | MSI RD480 Neo2-FI | RD480 | Dual DDR400 | 2-3-2-6 CR1 | 56.2 ns |
| 65.5 ns | Core 2 Extreme X6800 | 2933 MHz | Intel D975XBX | i975X | Dual DDR2-667 | 4-4-4-11 | 65.5 ns |
| 79.5 ns | P4EE | 3733 MHz | Intel SE7230NH1LX | iE7230 | Dual DDR2-667 | 4-4-4-10 | 79.5 ns |
| 80.3 ns | Opteron 248 | 2200 MHz | MSI K8T Master1-FAR | K8T800 | Dual DDR266R | 2-3-3-6 CR1 | 80.3 ns |
| 85.5 ns | Pentium EE 955 | 3466 MHz | Intel D955XBK | i955X | Dual DDR2-667 | 5-5-5-15 | 85.5 ns |
| 89.7 ns | Pentium M 730 | 1600 MHz | AOpen i915Ga-HFS | i915G Int. | Dual DDR2-533 | 4-4-4-12 | 89.7 ns |
| 91.1 ns | AthlonXP 3200+ | 2200 MHz | Asus A7N8X-E | nForce2-U400 | DDR400 SDRAM | 3-3-3-8 CR1 | 91.1 ns |
| 93.4 ns | Celeron M 320 | 1300 MHz | DFI 855GME-MGF | i855GME Int. | DDR333 SDRAM | 2.5-3-3-7 | 93.4 ns |
| 93.6 ns | Core Duo T2500 | 2000 MHz | Asus N4L-VM DH | i945GM Int. | Dual DDR2-667 | 5-5-5-15 | 93.6 ns |
| 98.3 ns | **P4 630** | **3000 MHz** | **[ TRIAL VERSION ]** | **i915GV Int.** | **Dual DDR2-533** | **4-4-4-12** | **98.3 ns** |
| 112.9 ns | P4 | 2800 MHz | MSI 848P Neo-S | i848P | DDR400 SDRAM | 2.5-3-3-8 | 112.9 ns |
| 115.7 ns | PIII-E | 667 MHz | MSI Pro266TD Master-LR | ApolloPro266TD | DDR266 SDRAM | 2-3-3-6 CR2 | 115.7 ns |
| 116.4 ns | PIII-E | 733 MHz | Tyan Thunder 2500 | ServerSet3HE | PC133R SDRAM | 3-3-3-6 | 116.4 ns |
| 116.6 ns | Xeon | 3066 MHz | Asus PCH-DL | i875P + PAT | Dual DDR333 | 2.5-4-4-7 | 116.6 ns |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 118.7 ns | Crusoe 5800 | 1000 MHz | ECS A530 DeskNote | Crusoe | DDR266 SDRAM | | 118.7 ns |
| 124.7 ns | P4EE | 3466 MHz | ASRock 775Dual-880Pro | PT880Pro | Dual DDR2-400 | 3-3-3-8 CR2 | 124.7 ns |
| 139.1 ns | K6-III | 400 MHz | Epox EP-MVP3G-M | MVP3 | PC100 SDRAM | 2-2-2-5 | 139.1 ns |
| 144.9 ns | Xeon | 3200 MHz | Intel SE7320SP2 | iE7320 | Dual DDR333R | 2.5-3-3-7 | 144.9 ns |
| 145.8 ns | Celeron D 326 | 2533 MHz | ASRock 775Twins-HDTV | RC410 Ext. | DDR2-533 SDRAM | 4-4-4-11 | 145.8 ns |
| 158.8 ns | Celeron | 1700 MHz | Asus P4B | i845 | PC133 SDRAM | 3-3-3-6 | 158.8 ns |
| 159.9 ns | C3 | 800 MHz | VIA EPIA | PLE133 Int. | PC133 SDRAM | 3-3-3-6 | 159.9 ns |
| 161.4 ns | Celeron | 2000 MHz | Gigabyte GA-8TRS350MT | RS350 Int. | Dual DDR400 | 2-2-4-6 CR1 | 161.4 ns |
| 162.3 ns | MediaGXm | 233 MHz | ALD NPC6836 | Cx5520 | PC60 SDRAM | 3-3-3-6 | 162.3 ns |
| 162.9 ns | P4 | 1600 MHz | Abit TH7II | i850 | Dual PC800 RDRAM | - | 162.9 ns |
| 166.0 ns | PIII | 500 MHz | Epox KP6-BS | i440BX | PC100R SDRAM | 3-3-3-? | 166.0 ns |
| 167.5 ns | C3 | 1333 MHz | VIA EPIA SP | CN400 Int. | DDR333 SDRAM | 2.5-3-3-7 CR2 | 167.5 ns |
| 170.8 ns | AthlonXP 1600+ | 1400 MHz | Acorp 7KMM1 | KM133A Int. | PC133 SDRAM | 3-3-3-6 | 170.8 ns |
| 172.1 ns | Athlon | 1333 MHz | PCChips M817LMR | MAGiK1 | DDR266 SDRAM | 2-3-3-7 | 172.1 ns |
| 174.1 ns | Duron | 1600 MHz | Biostar M7VIQ | KM266 Int. | DDR266 SDRAM | 2.5-2-2-6 CR2 | 174.1 ns |
| 177.9 ns | PentiumMMX | 200 MHz | Gigabyte GA-586DX | i430HX | Dual EDO | - | 177.9 ns |
| 182.5 ns | Duron | 600 MHz | Abit KG7-Lite | AMD-760 | DDR200R SDRAM | 2-2-2-5 | 182.5 ns |
| 194.0 ns | K6-2 | 333 MHz | Amptron PM-9100LMR | SiS5597 Ext. | PC66 SDRAM | 3-3-3-6 | 194.0 ns |
| 207.9 ns | Athlon | 750 MHz | Epox EP-7KXA | KX133 | PC133 SDRAM | 3-3-3-6 | 207.9 ns |
| 213.6 ns | PIII Xeon | 550 MHz | IBM Netfinity 8500R | Profusion | PC100R SDRAM | | 213.6 ns |
| 215.8 ns | PIII | 450 MHz | Asus P3C-S | i820 | PC600 RDRAM | - | 215.8 ns |
| 225.7 ns | PentiumPro | 200 MHz | Intel PR440FX | i440FX | Dual EDO | - | 225.7 ns |
| 251.4 ns | PII | 333 MHz | Intel DK440LX | i440LX | PC66 SDRAM | 3-2-2-? | 251.4 ns |
| 257.6 ns | Pentium | 166 MHz | Asus TX97-X | i430TX | PC66 SDRAM | 2-2-3-4 | 257.6 ns |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 275.8 ns | Celeron | 700 MHz | PCChips M758LT | SiS630ET Int. | PC100 SDRAM | 3-3-3-6 | 275.8 ns |
| 296.6 ns | K5 PR166 | 116 MHz | Asus P5A | ALADDiN5 | PC66 SDRAM | 3-3-2-6 | 296.6 ns |
| 299.7 ns | Celeron | 266 MHz | Epox P2-100B | ApolloPro | PC66 SDRAM | 2-2-2-5 | 299.7 ns |