# A History of Modern 64-bit Computing

Matthew Kerner
matthew.kerner@microsoft.com
Neil Padgett
npadgett@microsoft.com

CSEP590A

Feb 2007

# Background

By early 2002, the rumors had been swirling for months: despite the company's intense focus on its IA64 64-bit processor, Intel engineers in Oregon were busy updating the company's venerable x86/Pentium line with 64-bit EM64T (later Intel64) extensions. Not only would this mean Intel was potentially forgoing, or at least hedging, its bet on IA64 as its next-generation processor and entrance into the 64-bit processor space; reports [1] were also claiming that Intel's 64-bit x86 processor would be compatible with the x86-64 (later AMD64) architecture that rival AMD had announced just over a year before [2]. Intel boasted revenues of $26.5 Billion in 2001 [3], and was a giant compared to AMD, who sold a relatively tiny $3.89 Billion that year [4]. If true, Intel would be making a bold admission of AMD's coup by following the technology lead of its smaller rival.

This run-in was hardly the first time Intel and AMD crossed paths however; both had deep roots in Silicon Valley and in some ways they shared a common history. The relationship was a tortured one: the two companies were alternately friends advancing the x86 architecture and foes competing furiously.

Intel's founding traces back through Fairchild Semiconductor to Shockley Semiconductor Lab, the post-Bell Labs home of transistor co-inventor William Shockley. Founded in 1955 [5], Shockley Labs focused on the development and evolution of the transistor. However, in 1957 [6], fed-up with the management style of Shockley, eight young scientists left the company to set-up their own company [7]. With funding from the Fairchild Camera and Instrument Company, controlled by Sherman Fairchild [7], Fairchild Semiconductor was born. The company was successful, profitable within six months [8], and by 1958 a clear technology leader with its invention of the planar transistor [9]. In 1959, Sherman Fairchild took advantage of a buyout option in the funding agreement, buying out the eight for $300,000 each [7]. Cash rich and without a stake in the business, the eight began to drift away from the company, many starting new ventures of their own. In July of 1968, two of the last of the original eight to leave Fairchild, Gordon Moore and Bob Noyce, founded their own company, which they named Intel [7].

In 1969, Intel received a contract from a Japanese calculator manufacturer, Busicom, to produce chips to be used in business calculators [10]. In response to the Japanese request for calculator chips, Intel engineer Marcian "Ted" Hoff instead suggested a design for a more general-purpose chip. Hoff's chip, rather than being a specialized calculator chip, would be a generalized one that executed instructions fed from an external memory [10]. The chip, dubbed the 4004 (ostensibly the number of transistors the chip replaced [10]), was announced November, 1971 [7]. A second microprocessor, named the 8008, was released August of the next year [7]. In April of 1974, the 8080 [10] was announced. The 8080 served as the "brain" of the early personal computer, the MITS Altair [10], a machine credited by many as sparking the

personal computer revolution. The microprocessor was firmly established in the Intel product line. Updated products soon followed, with new variations such as microcontrollers introduced. Then, in 1978 Intel introduced another processor, the 16-bit 8086 [11], which would have a dramatic impact on the path of the company.

On August 12, 1981, IBM announced the IBM Personal Computer, or IBM PC [10]. At the heart of the machine was a microprocessor, Intel's 8088, a low-cost variant of the 8086. The machine also shipped with a BASIC interpreter and an operating system from Seattle software house Microsoft. (Microsoft had previously delivered BASIC for earlier PCs including the Altair.) Given its open design, the IBM PC was soon cloned, and these clones drove continued demand for the 8086 chips, as well as Microsoft's MS-DOS operating system.

However, Intel was not the only company selling these 8088 processors; in order to secure the contract, IBM required Intel to provide what was known as a "second-source," a second company granted a license to make the chips [7]. For this Intel turned to a company that had second-sourced its chips in the past, AMD.

Advanced Micro Devices, or AMD, was another company founded by Fairchild alumni. Jerry Sanders, a sales executive at Fairchild, left the company after a change in management saw him marginalized [12]. Soon after leaving, Sanders, along with several engineers who also left Fairchild, founded AMD [12]. Early on Sanders decided AMD would pursue a different strategy than that of other companies like Fairchild and National Semiconductor; rather than engineering new designs, Sanders focused his company on providing lower-priced chips compatible with the other companies' designs [12]. Adopting high quality standards and an understandable naming scheme for the competitive parts, AMD soon found a successful niche [12].

Intel and AMD's paths crossed in 1977 when Intel threatened to sue AMD over its unauthorized second-sourcing of an Intel-designed EPROM (Erasable-Programmable Read-Only-Memory) chip [7]. Though this episode caused some at AMD to worry that Intel wanted to drive AMD out of business, the two companies soon reached an agreement: AMD would make payments for the knockoff EPROMS and would give Intel the rights to second-source AMD designs, including their unique floating point unit, a specialized support chip that enabled microprocessor users to outsource math processing for greater speed [7]. In exchange, Intel would give AMD second sourcing rights for Intel's 8-bit 8085 microprocessor chip, as well as rights to the miniature computer programs, called "microcode," that Intel was beginning to incorporate into its chips [7]. A deal was struck.

The delight over the 8085 deal lasted only a little over a year [7]. By 1978, Intel was shipping samples of its new 16-bit processor, the 8086. However, Intel did not offer AMD second-source rights immediately. Not having heard from Intel, Sanders inked deals to become a second-source for Zilog [7], an Intel competitor that introduced a 16-bit chip, the Z8000, in 1979 [13]. However, Intel did come calling again once IBM demanded that Intel provide a second-source

for the 8086. The two companies reached an agreement in October 1981: AMD gained rights to manufacture Intel x86 processors (then the 8088 and the 8086) [12]. Under the terms of the agreement AMD would second-source Intel parts, first for cash and then, post-1985, based on a system of tit-for-tat product exchange with payments only for imbalances in the relationship [7]. The agreement, signed February 1982, was to last 10 years, with a cancellation clause allowing either party to cancel with one year's notice after year five [7].

As mentioned, the IBM PC design was a market success. Beyond IBM, hundreds of companies introduced clone models, some with added features, others with different pricing but similar designs. However, at the core, almost all needed an x86 processor. This meant success for Intel and also for its second-source, AMD.

However, Intel was not entirely happy with the AMD arrangement – some executives at Intel complained AMD had gotten a better deal; they compared Intel's high-volume, high-profit microprocessors to AMD's more prosaic portfolio of "peripheral chips" [12]. Internal Intel memos in 1985 and 1986 indicated Intel did not plan to let AMD second-source their upcoming 386 processor; Intel planned to cancel the second-source agreement. In 1987 AMD forced their hand — demands by AMD for arbitration over disputes about the agreement led to Intel invoking the one-year termination clause [7]. The subsequent arbitration would last 3 years and spanned 42,000 pages of written transcript [7]. Meanwhile, in 1987, Intel released their new 386. The 386 was an important step in x86 architecture evolution: with the 386 chip, Intel extended the 16-bit 8086 architecture to 32-bits. This allowed 386 systems to address more memory and deal with data in ways that the earlier x86 processors could not while still running existing programs written for the 16-bit x86 chips. This would also foreshadow an AMD strategy of almost a decade later.

Meanwhile, the cancellation of the Intel-AMD agreement marked a change of course for AMD. In 1989, Sanders directed AMD engineers to start building a clone of the Intel 386 based on only public information [7]. By August 1990, AMD engineers had working samples of their chip, dubbed the Am386 [7]. However, Intel and AMD would again clash on legal grounds; Intel sued claiming trademark rights to the name 386. A judge ruled against the company in March 1991, finding numeric chip numbers were effectively generic. AMD was free to call its chip 386. Also, due to timing, Intel was also unable to rename the by-then released successor to the 386, the 486, so competitors were free to use the 486 name as well [7].

Intel, however, was not pleased with this state of affairs and started a series of moves to cement their brand. In May 1991, the now familiar "Intel Inside" branding campaign debuted [7]. Intel also registered the trademark "Pentium" and announced that this, not 586, would be the name of their new processor.

AMD responded to Intel's moves by introducing their own branding for their successor to the Am486 (a clone of Intel's 486, the Am486 was also delayed at launch, in particular by renewed legal wrangling over AMD rights to Intel microcode) [12]. In June 1993, Sanders announced that

AMD's next x86 chip would be called K5. Further, this K5 would not be a reengineered Intel design, but rather would be designed by AMD. AMD would base the new design around a superscalar architecture, leveraging experience gained with RISC in building AMD's moderately successful non-x86 RISC processor line [14].

After the 1994 verdict that gave AMD rights to use microcode in the Am486, Intel approached AMD looking for a truce. Intel proposed that AMD give up rights to use Intel microcode in future processors [12]. After negotiation, the two companies reached an agreement, to run from 1995 to 2001, allowing AMD to build microprocessors compatible with Intel chips [12].

Intel shipped the Pentium processor March 1993 [15]. Updated models followed soon after. AMD's K5, however, reached market March 1996, months late; delayed due to design flaws [12]. The late product meant AMD fabs sat idle while potential AMD customers moved to Intel. AMD microprocessor sales fell 60% [12]. The chip saw marginal success: PC maker Compaq adopted it in low-end PCs briefly, however it was far from the hit AMD had hoped for [12].

Intel followed their Pentium with a sequence of new versions and products: the Pentium Pro, targeted at high end 32-bit applications, which was the first x86 processor to support superscalar, out-of-order execution with in-order retirement semantics[1]; the Pentium II, a mass market update of the Pentium based around the Pro; and Celeron, a value line of processors based on the Pentium II. A Pentium III eventually followed. AMD too introduced new processors; acquiring the firm NexGen and leveraging that talent to release a moderately successful K5 successor, called simply K6. The K6 release garnered moderate market share; reviews reported it was a solid performer [16]. AMD followed with updated versions, the K6-II and K6-III.

However, in 1996, Intel and AMD would again be back at the bargaining table; their patent agreement was to expire December 31 that year [7]. The two reached a renewed agreement. In it, AMD gave up their right to produce socket compatible chips beyond the Pentium generation (for AMD the K6 generation) [7]. The only remaining commonality between the two companies' chips beyond the x86 instruction set was soon to be gone.

AMD's next chip, codenamed K7, debuted in June, 1997. K7 used an AMD proprietary bus (based on technology licensed from DEC) and was named "Athlon." Athlon was a commercial success for AMD. Though AMD did not break into all of Intel's entrenched OEM customers, the influential gaming and enthusiast markets fell in love with the chip and its market-leading performance. By 1999 year end, AMD market share had risen from a 1997 estimate of 7.5-10% [17] to an estimated 14% [18]. With the release of Athlon, industry watchers began to consider AMD "the Intel competitor" in the processor space. But soon, Intel would attempt a leap to a

[1] Dileep Bhandarkar stated in a personal interview that he believed this development to be among the most significant developments in computer engineering.

revolutionary 64-bit design. Meanwhile, AMD, perhaps true to their roots, would take a page from the Intel playbook, extending the venerable x86 to 64-bit.

# IA64, AMD64 and Intel64

## *IA64 Development and Architecture*

In the early 90s, many engineers at Intel believed that x86 would soon hit a performance ceiling. In a personal communication, Dave Cutler elaborated, "When the IA64 project was started it was perceived that the performance of RISC machines would outstrip that of [Complex Instruction Set Computing (CISC)] machines. … They felt that they could get higher performance with a [Very Large Instruction Word (VLIW)] architecture than they could with a CISC architecture. … [T]he x86/x64 implementations have been able to master these complicated implementation techniques [out of order, superscalar, speculative execution] which was not thought possible 10-15 years ago in the RISC era." Bhandarkar confirmed that Intel perceived an x86 performance ceiling. He explained that the Pentium Pro design team was believed to be embarking on an ambitious and somewhat risky project to achieve performance advances with x86 that were only expected to be possible for RISC designs.

In response to this perceived ceiling, Intel began to look for alternatives. Bhandarkar described an internal Intel effort to develop an Alpha-like RISC chip. This chip would break through the x86 ceiling and compete with various RISC-based competitors including IBM PowerPC and Sun SPARC, which were common chips in scale-up SMP servers. At the time, Gordon Bell attempted unsuccessfully to convince Intel to base their new 64-bit chip on the Alpha itself [19]. Digital also considered approaching Intel with a different proposal: while Bhandarkar was still at Digital, he proposed that Digital collaborate with Intel to produce Alpha in volume. Since Digital had just invested $500M in a new fab, the idea was never pursued.

Meanwhile, HP was continuing to develop its PA-RISC line of processors. Josh Fisher, a Very-Long-Instruction-Word (VLIW) pioneer from Multiflow Computer joined HP in 1990 after Multiflow ceased operation [20]. Bhandarkar explained that Fisher convinced HP's management that a VLIW-based machine would outperform a traditional RISC machine. On that basis, Bhandarkar explained, HP decided to invent a new architecture based on VLIW to replace PA-RISC.

But HP was wary of continuing to invest in their own fabs. Cutler explained in a personal interview that Intel had the most advanced process, transistor, and fabrication technologies. Bhandarkar confirmed that HP looked to Intel as a potential fabrication technology partner. On the other hand, HP had expertise in processor architecture, while both Intel and HP had compiler and operating system development expertise. When HP management approached Intel management about a possible partnership, Bhandarkar said, Intel decided that the opportunity to gain access to HP's architectural expertise was too much to pass up.

Furthermore, there were some at Intel who believed that VLIW would deliver higher performance even as x86 implementations continued to improve. For these reasons, Intel agreed to work jointly with HP on what would eventually become IA64.

In addition to the 12- to 18-month performance lead associated with contemporary RISC architectures, Bob Davidson suggested in a personal interview that academic research was a major influence on the IA64 consortium. The VLIW work done by Fisher at Yale, and then at Multiflow, was focused on enabling the compiler to optimize across a larger instruction window than the processor could at runtime, and to expose the most efficient execution strategy through an Explicitly Parallel Instruction Computing (EPIC) Instruction Set Architecture (ISA).[2]

Davidson also pointed out two areas where academic research could create a blind spot for architecture developers. First, most contemporary academic research ignored CISC architectures, in part due to the appeal of RISC as an architecture that could be taught in a semester-long course. Since graduate students feed the research pipeline, their initial areas of learning frequently define the future research agenda, which remained focused on RISC. Second, VLIW research tended to be driven by instruction traces generated from scientific or numerical applications. These traces are different in two key ways from the average system-wide non-scientific trace: the numerical traces often have more consistent sequential memory access patterns, and the numerical traces often reflect a greater degree of instruction-level parallelism (ILP). Assuming these traces were typical could lead architecture designers to optimize for cases found more rarely in commercial computing workloads. Fred Weber echoed this latter point in a phone interview. Bhandarkar also speculated that the decision to pursue VLIW was driven by the prejudices of a few researchers, rather than by sound technical analysis.

Bhandarkar denied that competition from AMD drove Intel to pursue IA64. However, Richard Russell explained in a personal interview that AMD perceived a strategic interest at Intel in pursuing IA64. Russell elaborated that legal agreements ensured Intel's x86-related IP was available for inspection by competing processor companies, including AMD. As a result, Intel had to innovate constantly to avoid a commoditized market, as other players incurred less risk and lower research budgets than Intel did as the initial developer of the technology. By making a decisive move, Intel could establish itself as the uncontested market leader and gain a head start on AMD and other competitors from an IP perspective.

Weber similarly speculated that there were several lines of reasoning during Intel's internal debate on whether to pursue a new ISA in IA64: those who believed that a new ISA was necessary to deliver higher performance; those who believed that Intel had slowly lost its hegemony over x86, and that the invention of a new architecture with strong patent protection would leave the x86 competition behind; and those who believed that Intel had enough influence to make any architecture a success. He also speculated that the IA64 move was meant

[2] Bhadarkar explained that Digital licensed the Multiflow compiler, although not the associated VLIW technology.

to eliminate PA-RISC, Alpha, MIPS and SPARC from the field. It was, in fact, a successful move against PA-RISC and Alpha, which were retired by HP and Compaq in the face of IA64.

Intel did not embark on any public program to simultaneously extend x86 to a 64-bit address space, even though this was the next obvious step along the path that they had blazed when extending x86 from 16- to 32-bits. Bhandarkar explained that Intel perceived no pressing market need to extend the x86 for personal computers at the time. He elaborated further, "It was not a question of if we extend the x86, the question was when … [There was] a lot of feeling inside Intel that we should postpone extending x86 as long as possible to allow Itanium to establish itself to as large an extent as possible." That is, Intel chose not to cannibalize their potential IA64 business with an internally-developed competitor.

Instead, Intel and HP set out to build a new ISA that they hoped would be the foundation for the future of both client and server computing.[3] Briefly, IA64 is a RISC-based EPIC architecture with a dizzying array of features designed to expose ILP at the ISA level. The basic unit of scheduling consists of two sets of three-way VLIW instructions, also known as instruction bundles, providing six-way instruction issue [21]. The compiler is responsible for populating the open slots in the bundles in order to achieve superscalar performance. IA64 was also designed to target multiproc (later multicore and hyperthreaded) systems, providing thread-level parallelism as well.

As in traditional RISC architectures, IA64 contains a large register file exposed through the ISA and accessed directly by software [21]. A unique feature of IA64 is the idea of a register stack that automatically renames a subset of the registers across function calls. This feature provides each function with a fresh copy of as many registers as that function requires, without requiring the application to save or restore register contents across function calls. It also contains processor logic – the Register Stack Engine – to automatically spill register contents to memory on stack "overflow," and restore those contents on "underflow." The stack mechanism also allows functions to pass arguments in registers efficiently as part of the register rename operation. Finally, a variation on the rename scheme enables hardware-assisted software pipelining in loops by adjusting register names per loop iteration to simplify addressing of data across iterations.

The branch prediction and cache management features of the ISA are exposed through instructions that allow the compiler to provide hints to the processor about the behavior of the program [21]. For example, branch prediction instructions allow the compiler to indicate that a branch is likely to be taken, or to specify the likely behavior of a loop to assist the processor in keeping the pipeline full across every loop iteration. Similarly, cache management instructions allow the compiler to indicate whether a prefetch instruction should populate the L2 cache, L1

[3] While the consortium made the decision to pursue IA64, Bhandarkar explained that it was a controversial issue at Intel. Some believed that IA64 would not successfully unseat x86.

cache or both. The ISA also provides a mechanism to prefetch with the intention to write, which affects the disposition of the cache line in SMP systems with cache coherency protocols.

Both control and speculation are exposed in the IA64 ISA to assist the compiler in boosting loads to avoid pipeline stalls due to cache misses [21]. Exceptions on speculative loads (control speculation) are deferred and propagated in the data flow of the target register until the exception can be handled by the application either when the load's contents are consumed by an instruction whose effect cannot be deferred (e.g. a store), or when the speculative load's result is checked explicitly in the load's basic block of origin. In order to perform data speculation, the processor tracks stores to addresses that were previously speculatively loaded, so that they can be reloaded if the load results are consumed.

In addition to control and data speculation, IA64 also supports instruction predication [21]. Using predication, branches can be eliminated by the compiler (and thus the performance impact of mispredictions can be avoided) by tagging instructions with a precomputed predicate that is only true if the would-be branch condition is true. The processor will only commit the instruction if the predicate is true.

IA64 also includes some features that appeal in particular to high-end computing users. The multiprocessor memory model is defined as part of the ISA rather than as part of the microarchitecture or platform. Further, with features like reliable memory and hot-swap parts, the ISA implements reliability/system management features preferred in datacenters. Also, the floating-point support in the ISA is excellent. Finally, the ISA defines a new firmware mechanism, called the Extensible Firmware Interface (EFI) that is responsible for a variety of low-level platform functions like boot and hardware error correction.

The Itanium microarchitecture, the first microarchitecture for the IA64 ISA, provides many standard microarchitectural features such as pipelining, an on-chip cache hierarchy, cache coherency protocols for multiprocessor chips, branch prediction, hazard detection and recovery logic, and so on [22]. One key feature of Itanium was the bus structure, with high enough bandwidth to support impressive scale-up disk, network and memory IO requirements.

Compatibility with existing x86 applications was a concern in the IA64 ISA, but not a priority. The ISA specified a mechanism for running x86 programs in hardware, but the chip was optimized for the native ISA, and x86 execution was not to be at-speed [21]. Jeff Havens explained in a personal interview that the goal was to run x86 software as well as the previous generation of contemporary x86 processors. In fact, this goal was never met, and the x86 hardware emulation was eventually removed. Instead, it was replaced by a binary translation mechanism to convert x86 programs to native IA64 code. Havens recalled that the Intel team reported comparable performance using binary translation, and that this was the point at which hardware emulation was cut.

Bhandarkar elaborated on the evolution of x86 support in IA64. Initially, it was provided by dedicated hardware at about 50% of the execution speed of contemporary x86 chips. However, with the release of out-of-order x86 chips, the performance level dropped significantly relative to native x86 execution, and the team decided that the IA64 implementation could be cleaner if hardware support for x86 was removed in favor of binary translation. The McKinley and Madison releases of the Itanium processor used binary translation with an option to execute in hardware as a hedge, while Montecito dropped hardware support as binary translation was viewed as being mature enough to stand on its own.

The complexity of emulating x86 on IA64 was daunting. In addition to basic binary translation, the x86 compatibility model had to provide a strong ordering model for x86 programs on top of a weaker IA64 model. While this was possible, Havens described it as being extremely difficult to accomplish while still maintaining acceptable performance. To make matters worse the translated programs had to produce x86 exceptions with complete context on top of native IA64 exceptions. Havens said, "Our code does every trick in the book trying to make the 32-bit debugger not know that it's working on translated code." It is no wonder that the performance of x86 code on IA64 processors was an issue even with only occasional use.

The set of features in IA64 are appealing to some, and not to others, but most people agree that the number of features introduced in the ISA is quite astoundingly large [23]. Bhandarkar explained that the design process was a committee process, and that when presented with two options for a design, the committee often compromised by adopting both.[4]

Along with development of the ISA, the microarchitecture and the hardware itself, Intel and HP began to work with a variety of partners to prepare tool chains, operating systems and applications for IA64. For example, Intel hired Red Hat to complete the open-source tool chain port.

Intel also began to work with Microsoft to complete both tool chain and Windows ports. Intel accepted some input on the design of the processors (one engineer at Microsoft described sharing traces of large-scale database workloads with Intel for performance-tuning purposes), but the collaboration with Microsoft was largely limited to software design rather than the ISA, unlike the later AMD64 collaboration between AMD and Microsoft. Bhandarkar confirmed that Intel and HP essentially delivered a completed ISA manual to vendors, and made only minor adjustment in response to feedback.

Havens was the architect at Microsoft in charge of the IA64 Windows port. He explained that the initial porting work was done by a team of twelve OS developers at Intel. This was a significant investment by Intel, considering that Intel's source-control policies would prohibit these developers from working on subsequent OS projects for a period of years. The Intel

[4] As a Microsoft engineer pointed out, this approach was similar to the approach taken for plenty of other large technology projects (e.g. OS/2 development).

engineers' changes were submitted to Microsoft for review and approval. When Intel completed their initial work, the remaining work transitioned to Microsoft.

As Havens explained, the primary goal of the port was to keep changes minimized, and the changes that were made were broken up into four categories: common code, processor-architecture specific code, platform specific code and pointer-size specific code. Interestingly, the pointer-size and architecture specific work was not always split apart, since IA64 was the only active 64-bit Windows port at the time. Furthermore, the build system wasn't yet sophisticated enough to be able to handle the combination of options required to build all of the different flavors of the operating system. The port covered the operating system and server applications, but not many client components, although some of these could run with x86 emulation.

The bulk of the work had to do with defining and using types properly.[5] In addition to the type work, there were sections of code that were processor or platform dependent that had to be rewritten, including the addition of EFI support on the boot path. This work was particularly tricky because EFI was brand-new, and any bugs in the firmware prevented further testing until they were understood and fixed. Since Intel maintained exclusive access to the firmware source, this was sometimes a bottleneck for the Microsoft team.

The trap handling code in the kernel presented its own challenges. Much of this code could only be written in assembler (e.g. trap handler code that could not fault). Havens described spending a great deal of effort finding ways to achieve more than one instruction per cycle with hand-tuned VLIW assembly code. The team also focused on optimizing certain performance-sensitive routines in the OS, like cryptography routines, to take advantage of the EPIC features of the ISA.

Along with the Windows port, there were several categories of compiler work to be done to generate code for IA64. On the more trivial end, the compiler had to be changed to generate warnings for 64-bit portability violations (e.g. truncation and expansion). These warnings were useful markers for identifying areas where porting work needed to be done.

The compiler team also had to decide what code generation strategy to adopt. Davidson explained that Intel's own compiler team tended to use many of the new features of the ISA extensively. While these features enable the compiler to speculate more aggressively, they also result in significant instruction stream expansion. In fact, the more aggressively the compiler speculates, the more code expansion worsens. For example, predicated instructions must execute even when their results are not used, and control speculation requires recovery code.

[5] Pointer width and so-called parametric types were areas of focus. For example, pointer-sized ULONGs had to be handled properly across both 32- and 64-bit builds.

[23] estimated that an average IA64 instruction stream would be four times longer than the equivalent x86 instruction stream. In practice, Davidson and Havens described observing a two- to three-fold expansion. At any ratio, additional code impedes data hit rates in shared caches (a key concern on the cache-limited, functional-unit-rich Itanium), and consumes both memory bandwidth and memory footprint. Davidson explained that the Microsoft compiler team chose to speculate less in order to generate smaller binaries, given the importance of a small code footprint in mixed workload settings.

Davidson explained that it was hard to fill the bundles at first to achieve more than one instruction per cycle. Bhandarkar confirmed that Intel and HP struggled with the same issue: 20-30% of the instructions in their early IA64 instruction streams were No-Ops. Davidson recounted further that Microsoft's compiler and server application teams had work to do before the compilation process could take advantage of profile results during code generation to really leverage the EPIC features of the ISA. This effort paved the way for future application writers to get similar performance gains when using traces, but the biggest gains are still only achievable if the developer can generate accurate profiles of execution.

For OS developers, the debuggability of machine code is an important factor in the ease-of-use of any platform. Havens, who was intimately familiar with the IA64 ISA, found it to be easier to debug than some other platforms, such as DEC's Alpha.[6] In particular, two aspects of the ISA were helpful: the register stack ensured that the context for each frame was available in one well-known location. The other useful feature was the fact that the TLB was populated by software rather than hardware or firmware (as with Alpha). By inserting logging code into the page fault trap handler, developers could infer what code had been running prior to an exception and use that to investigate possible bugs.

Other developers expressed great distaste for debugging IA64 code. The mnemonics are different from the familiar x86 mnemonics, the bundle formats are unfamiliar, and speculation obscures the original behavior of the code as written. These complains illustrate the challenge that Intel and HP faced with the introduction of such a different and feature-rich ISA.

The Itanium processor was slated to be released in 1999, but after repeated delays it shipped in 2001 [24]. It was released with an 800 MHz clock speed, which was less aggressive than the original projected clock speed, and was also lower than the gigahertz speeds of other processors at the time [24].[7] Bhandarkar explained how Intel's expectation of IA64's penetration changed

[6] There were three exceptions to this. First, when the system was in the middle of a software divide (the ISA provides no hardware divide support [21]), the instruction stream was fairly obscure. Through repeat exposure to this, the developers working on the port started to recognize divides. Second, errors in recovery code were hard to debug since this code executed infrequently and only in unusual scenarios. Fortunately it was mostly compiler-generated, so the fixes could be made in one central place. Finally, debugging exceptions was difficult because of the degree of speculation and code motion that could occur, thus making it difficult to determine how the instruction stream related to the original program as written in a higher-level language.

[7] Multiple engineers we spoke with described the first boot of each new processor with great enthusiasm and detail. Havens explained that the first Itanium processors were being readied in Dupont, Washington. The Microsoft, Linux and HP/UX teams were all present on the same day in the same lab, bringing up the processor. Each team had a secure room for their software work,

over time. From 1995-2001, the IA64 team believed that it would penetrate to high-end workstations, but over time their outlook was revised to first the server market, and then high-end servers only. To a chip manufacturer, this was a big change: the expected market shrunk from an estimated 200 million units to an expected 200,000 units.

## *AMD64 Development and Architecture*

Unlike Intel and HP, AMD was not a market leader embarking on an aggressive new project to redefine the processor architecture landscape. In fact, AMD was a distant second competitor to Intel, with much more success in embedded processors, flash memory and value-focused client processors than high-end client or server chips. During most of x86's history, Intel had been the leader and AMD had been the follower.

Intel had a successful strategy to make it difficult for AMD to follow their lead using a combination of trade secrets, legal agreements and licensing terms. As Fred Weber explained, Intel successfully pushed AMD to produce its own design for the K5 chip for business, rather than technical reasons. This was also true for the K6. This process was again repeated with the K7 chip; AMD chose to build its own socket and chipset design to avoid having to license the P4 bus.

Ironically, Intel's success at forcing AMD to jump over business hurdles while following Intel's technical lead was a key factor in setting AMD up to be successful with AMD64. Weber and Russell described the painful process of striking out on their own with first the processor and later the bus and chipset design. Even though it was painful, they both felt that these Intel-induced steps set AMD up with the skills, experience, and confidence to develop its own technology.

Weber explained that the motivation to pursue a new chip design was AMD's interest in breaking into the server market. Although the K7 had been a successful chip from a performance perspective, it made only small inroads in the desktop market, and had zero server market share. AMD believed that if they could gain server market share, the consumer desktop market, where the real volume is, would follow.

AMD needed a 64-bit server chip design. Weber recounted that AMD considered many different options including Alpha, SPARC, MIPS, PowerPC and IA64. The key factor for AMD was installed base, so Alpha, PowerPC and SPARC were all tempting. AMD considered

---

but worked on the hardware in a shared lab. They all successfully brought the machine up on the same day. However, the Linux team brought the machine up a few hours before the Windows team did because Intel had forgotten to initialize the keyboard controller in the firmware. Linux accepted input from the serial port, while Windows did not, forcing the Windows team to wait for keyboard controller support before booting successfully.

shipping a processor in one of these architectures, with built-in support for executing x86 binaries.[8]

But Weber explained that AMD also saw an opportunity to capitalize on two large areas that Intel had left open: macro and micro/system architecture. AMD believed that Intel's plans for the IA64 ISA didn't make sense at a macroarchitectural level: they were too focused on numerical/scientific computing, and had missed the general architecture sweet spot. Even if IA64 had been a great general-purpose ISA, AMD knew that introducing *any* new ISA would be very hard. Weber himself had previously worked at Kendall Square Research, where he experienced customer reluctance to adopt a new ISA firsthand.

AMD believed that Intel had also missed an opportunity in micro/system architecture. If AMD could extend its design experience gained from the K7 socket design to provide the parts for a glueless two- to four-processor SMP system, the chips would sell. Weber saw a rare combination of macro- and micro-architectural opportunities, and as he put it, "By exploiting both of them we were able to come from 0% presence in the server market to 25% presence." Intel's strategy had "left a big wide open left flank for someone to exploit. It's by a small chance that AMD had the guys, the position and the willingness to try to exploit that left flank and it wasn't a slam dunk that it would succeed."

Dave Cutler explained another aspect of the microarchitectural opportunity in more detail. Intel chips at the time were designed with a short-tick model: they were heavily pipelined and ran at high frequencies (a "short tick"). AMD's processors were the opposite: they had fewer pipeline stages and ran at lower frequencies (a "long tick"). We know in retrospect that power leakage goes up exponentially with clock frequency at small feature sizes, and Intel's short tick design was reaching a power consumption ceiling even if it was a successful marketing technique. AMD's long tick design used much less power, and therefore provided further headroom for future clock speed increases.

Essentially, AMD had decided to take a big bet by "taking on the mantle of x86," said Weber, "If we didn't succeed we would be on an architecture that we couldn't get software behind. If we did succeed it would put the AMD vs. Intel story in a different light. It was a big bet, but possible to win, and if you did win, since it was a big bet, it was a clean sweep of the whole field." Interestingly, Russell described the bet much the same way, recalling an old AMD saying that "Intel plays chess, but AMD plays poker." At least there was the consolation that if the bet failed, AMD would still have produced "a heck of a 32-bit processor," according to Weber.

Unlike Intel, AMD began working on the 64-bit project after x86 chips had gone superscalar, speculative, and out-of-order (as mentioned above, Intel's Pentium Pro was the first chip to

[8] Despite the market share factor, it was not a slam-dunk decision to ignore IA64. As Weber put it, "If Intel's going in a direction, you have to think long and hard if you don't go in that direction." Like the decision between using RAMBUS and DDR, he explained, AMD would take a big risk if it didn't follow Intel's lead.

achieve this milestone). This allowed AMD to build upon x86, confident that they could continue to extend its performance beyond what had been generally believed to be possible at Intel at the beginning of the IA64 project.

The project began under the codename "Sledgehammer". Weber explained that the building used by the team had a wall down the middle that separated the team into two parts. One day, he and a coworker brought in sledgehammers and knocked holes in the wall. That was the origin of the codename. In fact, the name was so popular that the team had to be very disciplined in using the name Opteron once it was chosen by marketing, out of fear that Slegehammer's popularity would prevent the real product name from sticking.

Briefly, the AMD64 architecture ended up being a straightforward extension of the 32-bit x86 architecture to 64 bits of address space and 64-bit operand sizes [25]. Its Legacy mode supports running 32-bit x86 operating systems and applications at speed with 100% x86 compatibility. 64-bit-aware OSes can put the processor into Long mode, which extends the ISA to support a set of 64-bit extensions. In Long mode, the opcodes for a redundant set of increment/decrement register instructions are repurposed for use as instruction prefixes to signal 64-bit operand and address sizes. Long mode also provides a compatibility sub-mode to allow 32-bit binaries to run at speed, as written.

There are a number of small performance improvements to the x86 architecture in Long mode [25]. AMD64 widens the processor registers to accommodate 64-bit values, and it also doubles the number of registers to 16. In Legacy mode the upper half of each register and the 8 additional Long mode registers are simply ignored. The Long mode calling convention is updated to take advantage of the additional registers when passing arguments. The Long mode page table is extended by one level to accommodate the larger address space, and segmentation is largely removed, although GS and FS are retained to provide software with pointers to key OS structures at runtime. Virtual 8086 mode is unsupported in Long mode, and several other changes are active in Long mode (more details below).

AMD opted not to change the major part of the ISA from x86 when moving to AMD64, even though x86 is far from a "clean" architecture. Weber talked about the difficulty of convincing customers to adopt a new ISA, which were avoided by keeping Long mode close to the familiar x86 design. Furthermore, Cutler explained the positive engineering impacts of maintaining a largely consistent instruction set from x86 in AMD64. Not only are application, OS and tool chain ports simplified (as Havens said, one can *almost* assemble an x86 program to run natively on x64), but the same functional units and pipelines in the processor are used for both 32- and 64-bit execution. This means that any improvements made for one mode propagate to the other, and the designers do not have to make tradeoffs between 32- and 64-bit improvements. Finally, a single team can work on the chip, contributing to both 32- and 64-bit improvements simultaneously, instead of having to split the work across one 64-bit team and another 32-bit emulation team.

Russell, Cutler, and Weber highlighted the implementation of the K8 chip, AMD's first AMD64 processor, as a key success factor for the architecture. HyperTransport is a point-to-point, serial, bidirectional link used for inter-processor communication on the K8. Since it handles cache coherency messages and inter-processor interrupts, that traffic need not travel on the bus, eliminating the performance impact of the resulting bus contention. HyperTransport can also be used to connect a processor with a DRAM controller, and since there is no added bus contention, the memory bandwidth and capacity can scale up without hitting a contention ceiling.

Another K8 chip innovation identified by Russell, Cutler and Weber is the on-chip DRAM controller. As described above, commercial workloads are often dominated by the cost of memory latency and bandwidth, and putting the DRAM controller on-chip reduces that cost. Combining HyperTransport with an on-chip DRAM controller enables the creation of a glueless SMP system with NUMA, a key target in AMD's attempt to gain server chip market share. The K8 chips had enough HyperTransport links to support a 4-way glueless SMP system, AMD's original target for a system architecture win. Weber attributed the original idea for a glueless SMP chip to the Alpha EV7.

Weber described the early steps that AMD took once they formulated the AMD64 plan. They ran the idea by a few key reviewers: Rich Oehler and Marty Hopkins at IBM, Dave Cutler and Rob Short at Microsoft, and John Fowler at SUN. All of these reviewers were enthusiastic about the idea, although Sun would continue to promote its own SPARC architecture, and Microsoft was initially noncommittal while working on IA64 support for Windows.

Though Microsoft was officially noncommittal towards the early AMD64 effort, Cutler saw that AMD was open to feedback from Microsoft and he began to work on the project energetically. This difference in approach from the IA64 consortium, who had already finalized the ISA before presenting it to the software industry, was a key issue among the engineers we spoke with. They indicated that the input of the software industry helped to define AMD64 as an ISA that the industry could adopt with lower degrees of risk and uncertainty, and the lack of early consultation by the IA64 consortium in that regard hampered widespread adoption.

In parallel with the Microsoft collaboration, AMD also engaged the open source community to prepare for the chip. AMD contracted with both Code Sorcery and SuSE for tool chain work (Red Hat was already engaged by Intel on the IA64 tool chain port). Russell explained that SuSE produced C and FORTRAN compilers, and Code Sorcery produced a Pascal compiler. Weber explained that the company also engaged with the Linux community to prepare a Linux port. This effort was very important: it acted as an incentive for Microsoft to continue to invest in the AMD64 Windows effort, and also ensured that Linux, which was becoming an important OS at the time, would be available once the chips were released.

Weber goes so far as to say that the Linux work was absolutely crucial to AMD64's success, because it enabled AMD to produce an end-to-end system without the help of any other

companies if necessary. This possibility ensured that AMD had a worst-case survival strategy even if other partners backed out, which in turn kept the other partners engaged for fear of being left behind themselves.

There are several possible reasons why AMD might have taken a more collaborative approach with the software industry. Of course, they were an underdog in the processor market, which could have led them to look more aggressively for any willing partner. Also, while Intel and HP both had robust software development groups, AMD was primarily hardware focused, and had to rely on external partners for most of the software development for AMD64.

The partnership with Microsoft was unusual in two ways. First, Russell pointed out that there was no contract governing the partnership. This lack of a contract was an incentive for AMD, the smaller player, to make the partnership work. Second, Weber, Cutler and Russell explained that it was largely Dave Cutler who sponsored – and completed most of – the work at Microsoft. Had it not been for Cutler's influence on the project, it is not clear that it would have been successful.

An additional factor that may have facilitated AMD's success with Microsoft was the shared experience at DEC between AMD's Jim Keller and Dirk Meyer, and Microsoft's Dave Cutler and Rob Short. Not only did these engineers have previous chip design experience from DEC, but they also had experience working together. Weber felt that this helped to ensure a successful collaboration. In contrast, although Bhandarkar and Cutler had worked together on the PRISM architecture team, Bhandarkar recalled that the primary Microsoft liaison at Intel was not a DEC alum, and the DEC connection was not a significant factor in the development of either IA64 or Intel64. Of course, there were both engineering-level and executive-level sponsors at Microsoft and Intel who ensured the success of the IA64 project (as well as other projects), and there was significant collaboration on the performance, compatibility and other aspects of the IA64-based Windows design.

From the very beginning of the project, Microsoft's input was instrumental in shaping the AMD64 ISA. For example, Davidson describes recompiling Windows repeatedly with different options to identify the optimal design in terms of register count, instruction encoding, calling conventions, context switch mechanisms, and so on. Cutler also identified a list of key architectural features that were the direct result of the AMD/Microsoft collaboration:

- RIP-relative addressing, which allows most Long-mode instructions to reference global data using a 32-bit displacement that is applied to the instruction pointer to enable indirection [25]. This enables global data addressing on large-address machines elegantly, without requiring a Table of Contents (TOC). Previously, large-address machines stored the address of a TOC in a register, and then passed small displacements to index into the TOC. Not only did this require an indirect lookup in memory for the final address,

but it required updates to the TOC pointer when switching between modules, which added complexity and hurt performance.[9]

- CR8, which allows the processor to change the interrupt mask directly through an ISA-exposed register in Long mode [25]. This register is an abstraction of the APIC interface, which allows for a much faster read (~6 cycles) and write (~25 cycles) of the current interrupt mask than the TPR register in x86 (hundreds of cycles). This provides a significant performance win when processing interrupts or using synchronization primitives that require masking interrupts.
- SwapGS, which allows the kernel to switch the GS register between a user-mode pointer and a kernel-mode pointer in Long mode to handle mode switches more efficiently while maintaining pointers to key user- and kernel-mode structures in the same register where they had always been stored previously [25].
- Exception trap frame improvements that align the stack on all exceptions and traps in Long mode to simplify stack-manipulation and exception handling.
- No-Execute (NX), which provides page-level execution permissions in Long mode to mitigate many classes of malicious software attacks. Unlike previous x86 protection mechanisms that were segment-based, NX can protect noncontiguous blocks of memory [25]. This feature is used extensively in Windows to minimize attack surface.
- Fast floating point switch, which allows hardware to skip saving and restoring the floating-point register state (256 bytes) on every call by relying on the calling standard to do so [25].
- Transparent multicore support, which exposes multiple cores through the same mechanism used to expose HyperThreading on Intel processors. This allows legacy operating systems that support HyperThreading to automatically support multicore, and conveniently ensures that the licensing model is identical between multicore and HyperThreaded processors.

Havens explained that the Windows port to AMD64 was partially simplified by the previous work that had been done both on the Alpha and IA64 porting efforts. Unlike the IA64 port however, the AMD64 port was done completely by Microsoft engineers. Less type-related work was required in the OS itself, but comparable architecture and platform specific changes were required. Now that there were multiple active 64-bit ports of Windows, some work was done to separate the changes required per architecture from the changes required to support different pointer sizes. One new area of work was the addition of many intrinsics to the compiler [26]. These intrinsics exposed architecture and platform specific methods in a standard and often portable way.

[9] Since RIP-relative addressing uses a 32-bit address field, it cannot be used in binaries that are greater than 2Gbytes in size. As Cutler said, "even FORTRAN programmers wouldn't have a 2GB array."

In addition to the basic OS port, Cutler described other porting work done by the AMD64 team at Microsoft. The volume of this work was quite extensive. This included the design and implementation of the WOW64 layer that allows 32-bit programs to run on top of a 64-bit operating system transparently. By translating 32-bit wide API calls into 64-bit wide API calls (and the results back to 32-bit results), WOW64 ensured a complete 32-bit application compatibility story on a 64-bit OS. In fact, Microsoft's focus on application compatibility was shared by AMD: Russell described an extensive compatibility testing program at AMD where engineers manually verified that existing 32-bit applications would continue to work with AMD64 chips.[10]

Russell recounted that there was some speculation at AMD about code that was recompiled to AMD64: would it experience a performance speedup when compared to the x86 binary running on the same chip? It ended up that simply recompiling to AMD64 didn't deliver an appreciable performance win. Russell speculated that CISC architectures simply don't benefit from more registers in the same way that RISC architectures do. Cutler explained that although the calling conventions are further optimized, the ~40% expansion of the instruction and data streams due to 64-bit operands and pointers counterbalances this gain.

Both Cutler and Weber believed that AMD's decision to publish the AMD64 ISA early [2] was a critical step in AMD's successful AMD64 program. Weber explained that this wasn't an easy decision within AMD, but it did accomplish two goals: it showed the market that AMD was leading the AMD64 effort rather than Intel, and it enabled Intel to ramp up on Intel64 much sooner than if AMD had kept the ISA secret. The development of Intel64 by Intel was a key step in the success of AMD64, speculated Mark Russinovich in a personal interview, because it convinced customers that the architecture would be widely supported. AMD's early publishing decision therefore had a huge impact on the eventual project outcome.

Weber pointed out that Intel's timing was a critical factor for AMD: if Intel had moved earlier, AMD might not have gotten the credit for doing the work. On the other hand, if Intel waited too long, AMD might have had trouble getting partners and customers to take the effort seriously.

In retrospect, Intel's timing forced it to rush the development of the first round of chips, in Weber's view, giving AMD a performance and architecture advantage in the market. Weber believed that in hindsight the timing worked out serendipitously for AMD in part because of

[10] Russell described bringing up the first K8 part. Once AMD had a working gcc compiler from SuSE, they crafted a version of Linux that could run entirely from the on-chip cache, called DiagOS. The K8 chip was equipped with a debug port that could be used to load the miniature OS and operate it once the part had come up. Michael Wisor elaborated in a personal communication that DiagOS allowed for platform level diagnosis and functional testing of the CPU in the absence of a fully booted operating system. Furthermore, the firmware was simulated using a software model simulation that modeled a full system in addition to the CPU itself. The first K8 part came up well at low clock speed, and they worked out the kinks as they brought the clock speed up to spec over time. Dave Cutler described the day that the first AMD64 machine was brought to the Microsoft campus in February of 2002. It arrived at noon, and by 2PM, Windows was installed and booted. Previously, it had taken 24 hours to install the OS on the AMD64 simulator. Today, the Microsoft production build lab compiles, links and images all Windows builds for all architectures on AMD64 versions of Windows, running on AMD64 processors inside HP build servers.

Microsoft's early avoidance of a public commitment to AMD64. Had Microsoft committed earlier, Intel might have gotten involved earlier, reducing AMD's opportunity to innovate before any competitors had entered the field.

Despite the importance of Intel64 to AMD64's success, Weber believed that Intel could not have prevented AMD64's success: AMD had "gotten enough support from the Linux community and Microsoft that we were over the hump, and that led to the support of the OEMs," which was the final stroke.

Bhandarkar described Intel's decision to pursue Intel64 further. In November 1999, Weber presented Sledgehammer at the Microprocessor Forum, and Intel didn't feel they could ignore it. They also knew that AMD would get to silicon before Intel could, regardless of how they responded.

One option was to preannounce a competing ISA with a RISC-like 64-bit extension to x86. This would have been risky: Microsoft and other vendors were unlikely to develop softeware for another, non-compatible x86 extension without a major performance win. Furthermore, Intel did not want to damage the IA64 project, and disclosure of an alternative 64-bit plan so far in advance of the Itanium release would hurt the project and the HP relationship. Nor could Intel work on this alternative secretly: if Intel's x86 extensions were not compatible with AMD64, Intel would have to disclose the plan to vendors to enable them to develop compilers and operating systems for the platform.

Eventually, Bhandarkar was responsible for proposing the effort that began in June of 2000 to release an AMD64-compatible Intel ISA that was variously called Yamhill, Clackamas and finally EM64T during its secretive development cycle (it was eventually renamed Intel64). Intel knew that vendors would be able to release Intel64-compatible software quickly on the heels of their AMD64 development efforts. In fact, Intel monitored the Windows source for AMD64-releated changes, ran their own builds, and tested those builds on pre-silicon simulators for validation before sharing the plan with Microsoft so as to keep the possibility of leaks to a minimum. In January of 2002 they began to disclose their plans to partners, followed by testing on prototype systems in 2003, and production systems in early 2004.

## Market Reaction

The market reaction in the 64-bit area is worth discussing. Most software vendors did not express opinions about the relative merit of the architectures. Others, such as open source celebrity Linus Torvalds, lead developer of his namesake Linux kernel, were clear in their opinions: "[Intel] threw out all the good parts of the x86 [32-bit Pentium-class microprocessors] because people thought those parts were ugly. They aren't ugly, they're the 'charming oddity' that makes it do well," remarked Torvalds in 2003 [27]. Added Torvalds, "Right now Intel doesn't even seem to be interested in '64-bit for the masses', and maybe IBM will be. AMD

certainly seems to be serious about the 'masses' part, which in the end is the only part that really matters" [28].

Sales of Itanium also got off to a slow start. In the first full quarter of Itanium sales in 2003, Intel sold fewer than 2500 servers [29]. Some sources, which excluded demonstration units from reports, gave estimates of fewer than 500 sold [29]. Of course, Itanium systems tended to be scale-up SMP machines, so the revenue from a single Itanium server sale was much more substantial than the revenue from a single AMD64 server sale. Meanwhile, AMD's 64-bit sales were brisk: AMD64 Opteron processors sold about 150,000 units in their debut year, representing about half of their AMD sales of server chips that year [30]. That said, AMD's numbers were still tiny compared to the six million Intel 32-bit server processors sold in that same timeframe [30].

Itanium processors were certainly successful in penetrating the scale-up server market: Bhandarkar estimated that recent IA64-based scale-up SMP systems sales from HP, NEC, Fujitsu and Unisys are near 45% of competing PowerPC and SPARC sales, while AMD64-based scale up SMP servers are only recently available on the market and not yet widely deployed.

In the workstation market, however, IA64 clearly failed to deliver. The processor was displaced by Intel's own Intel64 offering. In September of Intel64's release year, 2004, IA64 partner workstation giant Hewlett-Packard announced it would cease production of Itanium2 workstations [31]. HP cited demand "switching" to Intel64 processors [31].

During 2004, support for AMD64 also grew. In early 2004, Sun Microsystems, progenitor of the SPARC server/workstation CPU, announced plans to ship AMD Opteron-based servers [32]. IBM was already onboard to sell Opteron servers, and HP soon followed [33]. By August 2004, Sun was claiming, right or wrong, that overall Opteron server sales were greater than IA64 sales [34]. Meanwhile, analysts covering the IA64 market were disappointed – compared to analyst IDC's year 2000 estimate for $28 Billion in 2004 Itanium sales [35], 2004's actual sales were abysmal, hitting only $606 Million by mid-2004 [36].

Overall, with AMD64, AMD was able to gain market share. By Q1 2006, "AMD had 22.9% server market share compared to Intel's 76.8 percent, comparing units sold of Opteron to Xeon" [37]. (Xeon is the product name for Intel's server x86 processor line. Both Opteron and Xeon are used in server and workstation lines.) Considering AMD's initial zero percent share in server processors prior to AMD64, the Opteron product line was a huge success. That said, Intel recovered swiftly. The introduction of Intel64 Xeons hampered AMD efforts to gain further server market share. Intel also acted to quell slipping sales in commodity workstation and desktop markets, starting a price war with AMD. By 2006, AMD issued earnings warnings about slowing sales amid questions about whether the aggressive Intel CPU price war was to blame [38]. AMD64 allowed AMD to gain market share in a market where it previously had none. Intel's gamble on IA64 left the door open, as Intel failed to rev their Xeon product lines until after initial AMD gains.

All told, IA64 was poorly received in the market. Instead, customers switched to AMD's lower-cost 64-bit offering, rather than jumping to IA64. And throughout, even prior to Intel64, large segments of the market ignored both, instead continuing to purchase Intel 32-bit server processors. The perhaps outspoken Linus Torvalds summarized the reaction of the entire market when he wrote in 2003 about Itanium: "Code size matters. Price matters. Real world matters. And ia-64 at least so far falls flat on its face on ALL of these" [28].

# Final Thoughts

Looking back on the story of IA64 and AMD64, it is worth considering what the key factors were in the relative success of AMD64, and, relatively speaking, the disappointment of IA64. In particular, why did IA64 fail to capture mass-market 64-bit computing? And, on the flip-side, why did the AMD-originated AMD64 ISA catch on to such a degree that Intel was forced to adopt it in its mainline products? Several factors influenced the situation. Firstly, the timing of the release of the AMD and Intel products affected the market reaction. Secondly, Intel and AMD's decisions regarding ISA design had a clear influence. Finally, the price-performance of each platform at launch drove application availability and correlated to success. All told, a combination of factors determined the relative success of AMD64 over IA64.

The timing of the release of AMD64 and IA64 influenced the success of each product. The IA64 project was one that was fraught with delays: Intel's original planned release date of "late-1999" slipped repeatedly, first to "early-2000" [39], and then continued to slip until the product's actual release in May 2001 [40]. So, when Intel finally released the processor it was a design targeted to a market that had since moved. In fact, even before the release of Merced, the first generation IA64 chip, Intel partner HP had announced plans to skip the chip in favor of waiting for the successor IA64 generation [41]. At the same time, AMD understood how controlling timing could influence success. Weber remarked that AMD was extremely conscious that, if AMD64 was successful, Intel would need to adopt it in their x86 product line. Accordingly, Weber explained that AMD tried to make decisions to maximize their first-mover advantage: if Intel adopted too late it could hold back the success of the platform; too early, and AMD's advantage could be negated. In fact it worked out perfectly, with AMD benefiting from the increased confidence of Intel's support and Intel suffering from a rushed development cycle; many in industry commented that Intel's early Intel64 designs were unrefined. All told, AMD benefited from timing while Intel suffered with product delays, thus giving an advantage to the AMD64 design.

Intel's decision to design a new ISA also influenced success. Intel chose to go with a revolutionary change in design with the move to EPIC. This decision had a large impact. Firstly, compilers had to change; Davidson agreed that IA64 demanded more of a compiler writer than most other modern architectures. Software writers were also impacted; as noted previously, debugging IA64 assembly code and optimizing for the platform required extensive learning of a

new architecture paradigm and instruction set, as well as development of new techniques. Finally, initial releases of the IA64 had to expend transistors to provide backwards compatibility with existing x86 software, yet failed to meet the performance of Intel's own x86 chips of the same generation. Later efforts to use binary translation also failed to deliver expected performance on legacy apps. These two issues combined to create problems with application software availability: legacy applications were slow at launch because of poor performance running x86 workloads, so users did not migrate; they were better off staying on x86. Meanwhile, developers gained little from porting to IA64 because of the small installed base; and that installed base would not grow without applications. As Intel technical leader for the earlier P6 project, Robert P. Colwell remarked comparing x86 to PowerPC: "Who cares how fast any chip is if it cannot execute the right kind of code?" [42]

Conversely, AMD had a smooth transition path: because their chip was x86 at its core, legacy 32-bit x86 applications ran well on it. Extending x86 to 64-bits was also familiar territory; many developers would recall the transition from 16-bit x86 to 32-bit x86, and hence could understand the cost and impact of making a change. All told, AMD's architecture represented an easier transition for users and developers than Intel's and was hence relatively more successful.

Finally, IA64 suffered from problems with price-performance, both perceived and real. At launch the Merced was panned as slow by developers who attempted to use its x86 emulation; some in the media dubbed it "Itanic," making a play-on-words of Itanium comparing it to the ill-fated Titanic. At the same time, it was difficult to realize the promised performance of the chip when running native IA64 code as Davidson explained. And, even when the performance was realized, it was often not enough to justify the costs of an architecture change (and the considerable dollar costs of scale up systems); in many cases the gains were small compared to a much lower-cost x86 processor. Noted Havens, "Unless you have a clear uncontested 2x performance advantage, people aren't going to go to … [a new processor]; ten percent doesn't matter." Meanwhile, AMD64 delivered performance both in 64-bit mode as well as on 32-bit; this was part of AMD's strategy. Good 64-bit performance, as well as microarchitectural developments such as HyperTransport and the on-chip DRAM controller further served to cement the reputation of AMD64 as a "fast architecture." While IA64 provided unmatched performance on scale-up workloads, its pricing failed to differentiate it in the marketplace, and this contributed to a scarcity of applications for the platform, a major deployment blocker. Inertia was in x86's favor and AMD64 delivering reasonable performance weighted clearly in AMD64's favor.

All told, a mix of elements; timing, ISA design choices, and price-performance combined to see the market favor AMD64 over IA64 (excepting the scale-up SMP market, where AMD64 has not yet penetrated deeply). Ultimately, IA64 saw modest market share in the server market and failed to break into the client. Meanwhile, with AMD64, former second-source AMD was able to rise from nearly 0% server market share to, as Weber described, nearly 25% share. Though not a complete failure for Intel, IA64 failed to deliver on its original vision as a replacement for x86. Meanwhile, by taking a page from Intel's past strategy and extending x86, underdog AMD

captured market share and ensured its place as a viable CPU maker. Ultimately, the two companies continue their on-going battle in the microprocessor marketplace.

## *Key Dates*

Key dates from the development of IA64, AMD64 and Intel64 are shown in the following table:

| Year | Event |
|---|---|
| 1994 | Intel & HP announced the IA64 project, targeting a 1998-1999 release [43] |
| 1996 | Microsoft announced development of IA64-compatible Windows [44] |
| 1999 | Intel & HP released the IA64 Instruction Set Architecture (ISA) [43] |
| 1999 | AMD announced the AMD64 architecture [45] |
| 2000 | AMD published the AMD64 architecture specification [2] |
| 2001 | Intel & HP shipped the first Itanium system [46] |
| 2001 | AMD announced the Hammer architecture [47] |
| 2002 | Microsoft released the first IA64 compatible version of Windows [48] |
| 2002 | AMD announced development of AMD64 compatible Windows [49] |
| 2003 | AMD shipped the first AMD64 system [43] |
| 2004 | Intel announced Intel64 and shipped Intel64 compatible processors [43] |
| 2005 | Microsoft released an AMD64 compatible version of Windows [43] |

**Interviews**

- Face to face interview with Richard Russell, 11/21/2006, Redmond, WA.
- Face to face interview with Dave Cutler, 11/30/2006, Redmond, WA.
- Face to face interview with Mark Russinovich, 12/6/2006, Redmond, WA.
- Face to face interview with Bob Davidson, 12/8/2006, Redmond, WA.
- Face to face interview with Jeff Havens, 12/8/2006, Redmond, WA.
- Telephone interview with Fred Weber, former CTO, AMD, 12/12/2006.
- Telephone interview with Dileep Bhandarkar, Director, Enterprise Architecture, Intel, 12/28/2006.
- Private written communication from Michael Wisor, Senior Director, System Software Development, AMD, 2/16/2007.

We would like to thank Richard, Dave, Mark, Bob, Jeff, Fred, Dileep, Michael and everyone else we spoke with for their time and their thoughtful comments both during the interviews and upon subsequent drafts of this paper.

**References**

[1] T. Poletti, "Intel's Plan B Chip Stirs Debate," San Jose Mercury News, 25 January 2002 Morning Final, p. 1A.

[2] "AMD Releases x86-64™ Architectural Specification; Enables Market Driven Migration to 64-Bit Computing," [Online Document], 10 August 2000, [cited 14 Dec 2006], Available HTTP: http://www.amd.com/us-en/Weblets/0,,7832_8366_7595~715,00.html.
[3] "Intel 2001 Annual Report," [Online Document], 2001, [cited 14 Dec 2006], Available HTTP: http://www.intel.com/intel/annual01/facts.htm
[4] "AMD 2001 Annual Report," [Online Document], 2001, [cited 14 Dec 2006], Available HTTP: http://www.amd.com/us-en/assets/content_type/DownloadableAssets/AMD_AR2001.pdf.
[5] "William Shockley," [Online Document], [cited 14 Dec 2006], Available HTTP: http://en.wikipedia.org/wiki/William_Shockley
[6] L. Goff, "1958: The birth of integrated circuits," [Online Document], 19 May 1999, [cited 14 Dec 2006], Available HTTP: http://www.cnn.com/TECH/computing/9905/19/1958.idg/index.html.
[7] T. Jackson, Inside Intel, New York: Penguin, 1997.
[8] [Online Document], 2006, [cited 14 Dec 2006], Available HTTP: http://www.fairchildsemi.com/mediaKit/history_1957.html.
[9] [Online Document], 2006, [cited 14 Dec 2006], Available HTTP: http://www.fairchildsemi.com/mediaKit/history_1958.html.
[10] P. Freiberger and M. Swaine, Fire in the Valley, R. R. Donnelley & Sons Company, 2000.
[11] "Our History of Innovation," [Online Document], 2006, [cited 14 Dec 2006], Available HTTP: http://www.intel.com/museum/corporatetimeline/index.htm?iid=about+ln_history.
[12] J.L. Rodengen, The Spirit of AMD, Fort Lauderdale : Write Stuff Enterprises, 1998.
[13] "Zilog 8000," [Online Document], [cited 14 Dec 2006], Available HTTP: http://en.wikipedia.org/wiki/Z8000
[14] "AMD K5", [Online Document], [cited 14 Dec 2006], Available HTTP: http://en.wikipedia.org/wiki/AMD_K5
[15] "Pentium", [Online Document], [cited 14 Dec 2006], Available HTTP: http://en.wikipedia.org/wiki/Pentium
[16] T. Pabst, "Intel's Enemy No. 1: The AMD K6 CPU," [Online Document], 6 Apr 1997, [cited 14 Dec 2006], Available HTTP: http://www.tomshardware.com/1997/04/06/intel/page10.html.
[17] K. Polsson, "Chronology of Microprocessors," [Online Document], 2006, [cited 14 Dec 2006], Available HTTP: http://www.islandnet.com/~kpolsson/micropro/proc1996.htm.
[18] K. Polsson, "Chronology of Microprocessors," [Online Document], 2006, [cited 14 Dec 2006], Available HTTP: http://www.islandnet.com/~kpolsson/micropro/proc1998.htm
[19] "Minicomputers: The Dec (aka Digital) Story. / Gordon Bell," 11 Nov 2006, [cited 14 Dec 2006], Available HTTP: http://www.cs.washington.edu/education/courses/csep590a/06au/lectures/asx/csep590a_06au_3.asx.
[20] "Biography: Josh Fisher," [Online Document], 2006, [cited 14 Dec 2006], Available HTTP: http://www.hpl.hp.com/about/bios/josh_fisher.html.
[21] J.C. Huck, D. Morris, J. Ross, A.D. Knies, H. Mulder, R. Zahir, "Introducing the IA64 Architecture," IEEE Micro 20(5), pp. 12-23.
[22] H. Sharangpani, K. Arora, "Itanium Processor Microarchitecture," IEEE Micro 20-5, Sept 2000, pp. 24-43

[23] M. Hopkins, "Guest Viewpoint: A Critical Look at IA64. Massive Resources, Massive ILP, But Can It Deliver?" Microprocessor Report, Feb 2000.
[24] "Itanium", [Online Document], [cited 14 Dec 2006], Available HTTP: http://en.wikipedia.org/wiki/Itanium.
[25] AMD x86-64 Architecture Programmer's Manual Volume 1: Application Programming, Advanced Micro Devices, Inc, 2002. Revision 3.00.
[26] "Compiler Intrinsics (C++)," [Online Document], [cited 19 December 2006], Available HTTP: http://msdn2.microsoft.com/en-us/library/26td21ds.aspx
[27] "Itanium is 'horrible', says Torvalds", [Online Document], 25 Feb 2003, [cited 14 Dec 2006], Available HTTP: http://www.information-age.com/article/2003/february/itanium_is_horrible,_says_torvalds
[28] "Linus Torvalds, Itanium 'threw out all the good parts of the x86'," [Online Document], 24 Feb 2003, [cited 14 Dec 2006], Available HTTP: http://www.theinquirer.net/default.aspx?article=7966
[29] M. Kanellos, "Itanium sales off to a slow start,"11 Dec 2001, [cited 14 Dec 2006], Available HTTP: http://news.com.com/2100-1001-276880.html
[30] J.G. Spooner, "A year old, Opteron serves notice,"[Online Document], 22 Apr 2004, [cited 14 Dec 2006], Available HTTP: http://news.com.com/2100-1006_3-5197394.html
[31] "HP Discontinues Itanium Workstation Line," [Online Document], 24 Dec 2004, [cited 14 Dec 2006], http://www.cfdreview.com/article.pl?sid=04/09/24/2041229&mode=nested.
[32] T. Prickett Morgan, "Sun Mulls its Options As It Readies Opteron Boxes", [Online Document], 27 Jan 2004, [cited 14 Dec 2006], Available HTTP: http://www.itjungle.com/breaking/bn012704-story02.html
[33] T. Prickett Morgan, "Opteron Learning to Walk, Ready to Run", [Online Document], 28 Apr 2004, [cited 14 Dec 2006], Available HTTP: http://www.itjungle.com/two/two042804-story03.html.
[34] "Sun claims Opteron sales outpace Itanium," [Online Document], 26 Aug 2004, [cited 14 Dec 2006], Available HTTP: http://www.theinquirer.net/default.aspx?article=18102
[35] A. Vance, "Intel and IDC at odds over Itanium's future," [Online Document], 13 Jan 2004, [cited 14 Dec 2006], Available HTTP: http://www.theregister.co.uk/2004/01/13/intel_and_idc_at_odds/
[36] A. Vance, "Itanium sales fall $13.4bn shy of $14bn forecast", [Online Document], 30 Aug 2004, [cited 14 Dec 2006], Available HTTP: http://www.theregister.com/2004/08/30/opteron_itanium_sales_q2/
[37] "AMD misses earnings target despite Opteron sales," [Online Document], 21 Jul 2006, [cited 14 Dec 2006], Available HTTP: http://www.itworld.com/Tech/5050/060721amd/
[38] C. Kraeuter, "AMD Fesses Up," [Online Document], 7 Jul 2006, [cited 14 Dec 2006], Available HTTP: http://www.forbes.com/technology/2006/07/07/amd-intel-warnings_cx_ck_0707amd.html
[39] M. Kanellos, "Problems delay Merced chip", [Online Document], 1 Jun 1998, [cited 15 Dec 2006], Available HTTP: http://news.com.com/2100-1001-211718.html?legacy=cnet.

[40] "Intel's 64-bit Itanium released ... at last", [Online Document], 30 May 2001, [cited 15 Dec 2006], Available HTTP: http://www.information-age.com/article/2001/may/intels_64-bit_itanium_released_..._at_last.
[41] M. Magee, "HP confirms Merced retreat", [Online Document], 5 May 1999, [cited 15 Dec 2006], http://www.theregister.co.uk/1999/05/05/hp_confirms_merced_retreat/.
[42] R.C. Colwell, The Pentium Chronicles: The People, Passion, and Politics Behind Intel's Landmark Chips, Hoboken, New Jersey: Wiley, 2006.
[43] "64-bit", [Online Document], [cited 14 Dec 2006], Available HTTP: http://en.wikipedia.org/wiki/64-bit.
[44] "Microsoft and Intel Announce Windows NT Operating System Development Plans to Support Intel's IA64 Architecture," [Online Document], 18 Sept 1996, [cited 14 Dec 2006], Available HTTP: http://www.microsoft.com/presspass/press/1996/sept96/intelpr.mspx.
[45] "AMD Showcases First 'Virtuhammer' Simulator at Linuxworld shows in New York, Paris," [Online Document], 31 Jan 2001, [cited 14 Dec 2006], Available HTTP: http://www.amd.com/us-en/Corporate/VirtualPressRoom/0,,51_104_543_4493~655,00.html.
[46] "HP Announces Broad Portfolio of Itanium-based Systems, Services and Solutions," [Online Document], [Online Document], 29 May 2001, [cited 14 Dec 2006], Available HTTP: http://www.hp.com/hpinfo/newsroom/press/2001/010529a.html
[47] "AMD Announces 8th-Generation Architecture for Microprocessors," [Online Document], 15 Oct 2001, [cited 14 Dec 2006], Available HTTP: http://www.amd.com/us-en/Corporate/VirtualPressRoom/0,,51_104_543~10742,00.html.
[48] "Windows XP 64-bit Edition", [Online Document], [cited 14 Dec 2006], Available HTTP: http://en.wikipedia.org/wiki/Windows_XP_64-bit_Edition.
[49] "AMD and Microsoft Collaborate to further 64-Bit Computing," [Online Document], 24 Apr 2002, [cited 14 Dec 2006], Available HTTP: http://www.amd.com/us-en/Corporate/VirtualPressRoom/0,,51_104_543_8001~19906,00.html