



Technical Insight, Business Relevance.

Jim Turley

A N A L Y S I S

WHITE PAPER

64-Bit Microprocessors

A Technical and Market Evaluation
for Desktop, Server, and Workstation Applications



Primary Business Address
Phone: (831) 375-8086

861 Jewell Avenue
Fax: (831) 375-8087

Pacific Grove, CA 93950
Email: info@jimturley.com

Website: <http://www.jimturley.com>

USA

Introduction

The race among 64-bit high-end processors has many competitors grouped into two basic camps: traditional processors using the RISC architecture and a set of new and emerging, more volume-oriented, “industry standard” merchant processors. The RISC processors were developed originally by large computer vendors for their customers’ high-performance system needs. Examples include Hewlett Packard (PA-RISC), Digital (Alpha), IBM (Power4), MIPS (R20K) and Sun (UltraSPARC). Challenging these proprietary offerings from the system vendors are a pair of upstarts from Intel (Itanium) and AMD (Opteron). Among these 64-bit processors, the Intel and AMD products clearly stand out as upsetting the processor market dynamics. The following paper compares the features and opportunities for these architectures with special emphasis on the new Intel Itanium Processor Family (IPF) and emerging AMD Opteron products.

Classic RISC Processors

RISC saw its heyday in the early 1990s as an alternative to the increasingly complicated microprocessors of the day. Ironically, RISC’s guiding principles – simpler design, lower complexity, and reduced functionality – ran counter to the prevailing (and current) trends in semiconductor manufacturing. In other words, RISC advocates economizing on transistors just as Moore’s Law showers more and more transistors on chip designers. Nonetheless, many RISC architectures achieved some measure of commercial success, although few are used today for their original purpose. RISC processors dominate embedded processor and UNIX server and workstation shipments, but they have all but disappeared from the desktop computers and volume workstations for which they were originally conceived.

Digital Alpha

There have been only two major generations of Alpha processors despite the number of models and products announced over the lifespan that Digital, then Compaq, and now HP owned the design. The 21164 and 21264 (EV6) supplied the core for all the other processors. The 21364 (EV7) is essentially the same as the ’264 internally but with a different external system interface.

Alpha processors have always been fast but not only because of architecture. Alpha’s instruction set is equivalent to that of other RISC architectures. Its pipelines are not especially long nor does it embody any unusually clever microarchitectural techniques. Alpha’s lead is (or was) almost entirely due to its top-notch manufacturing process and the minute tweaking that went into tuning Digital Semiconductor’s process. Alpha was exquisitely designed for the exact semiconductor technology Digital used to build it.

Alpha was the first 64-bit processor to reach 1 GHz, in 2001. Ironically, this milestone came just a few short weeks after Compaq proclaimed it was discontinuing Alpha development and prior to HP's announcement that it was acquiring Compaq.

HP PA-RISC

Hewlett Packard's Precision Architecture RISC (PA-RISC) is among the oldest commercial RISC designs, driving HP's UNIX servers and workstations since the very early 1990s. As with SPARC, the PA-RISC architecture was originally offered to other vendors under license, and saw some success in Japanese computer offerings from NEC, Hitachi, Mitsubishi, and Oki Electronics.

PA-RISC processors never lacked for technical innovation. Announced in 1992, the PA-7200 processor added multiprocessing support, allowing vendors (primarily HP itself) to create single-, dual-, and quad-processor systems easily. It also was among the first CPUs to add a second execution unit and multilevel caches. The PA-8000 had four-instruction dispatch, aggressive out-of-order execution, and 64-bit data paths, all high-end features that give the PA-RISC the performance lead in the mid-90s. Subsequent designs such as the PA-8200 and PA-8500 pushed performance still further, but by the late 90s HP's customers had turned their attention to the upcoming IPF architecture, and PA-RISC development waned.

Although the PA-RISC architecture was never broadly licensed beyond the Japanese computer vendors, the unit volume of PA-RISC chips often exceeded those of Power- or MIPS-based processors due to HP's large share of the UNIX server and workstation market.

IBM Power4

IBM's Power4 embodies some astounding technology. The chip (or, more accurately, the module) includes some 680 million transistors, has 5,200 pins, and draws an astonishing 500 watts of power. The module comprises two processor cores per die and four die per module, making eight 64-bit processors in one unit.

The Power4 pipeline is 12 stages long and feeds eight execution units, including two integer units, two floating-point units, two load/store units, one branch unit, and one condition-evaluation unit. Like current Athlon and Pentium machines, Power4 cracks its instructions into an internal format that is more easily digested by the pipeline. Since Power4 is nominally a RISC architecture, it's a bit odd that IBM would see fit to "decode" RISC operations into this intermediate form.

Instructions are dispatched in order but reorganized in the pipeline. Individual instructions progress through the pipeline at various rates until they are reunited with their siblings and retired in order. In all, more than 200 instructions may be in flight in one Power4 processor – and there are eight processors in the module.

Level-1 caches measure 32K for data and 64K for instructions; L2 cache is 1.5 MB; L3 cache controller is included on the Power4 chip, although the 32 MB cache memory itself is off-chip but within the module.

MIPS64

The 20K processor core and chip is MIPS' current high end. The packaged 20K processor has 7.2 million transistors on a 34mm² die and includes dual 32K caches and a new system bus that can move 3.6 GB/sec at 150 MHz.

The 20K dispatches two instructions through its simple seven-stage, in-order pipeline. For a 64-bit processor, the 20K is not a particularly aggressive design. MIPS lost its workstation business once Silicon Graphics adopted IPF; there's now no point in designing high-end MIPS processors. MIPS is now firmly in the embedded camp, and that means sacrificing a few interesting features for lower power consumption and easier manufacturing. The 20Kc doesn't even have any advanced branch prediction. Its seven-stage pipe will probably limit the 20K to about 500 MHz, and that suits most embedded ASIC designers just fine.

The 20K has three execution units (one full integer, one partial integer, one floating-point) but can dispatch and execute only two instructions per cycle. Because the second integer ALU is a somewhat limited version of the first, the 20K cannot dispatch two load/stores simultaneously. It can have only one floating-point operation in flight at any one time, although the 20K can maintain two integer operations while it waits for the FP instruction to complete.

Although 20K won't embarrass any other 64-bit processors, it does have the advantage of ASIC availability, a vital element for its embedded customers. Those without the need, or the budget, for an ASIC project can buy the 20K chip for development or moderate-volume production.

Sun UltraSPARC

SPARC represents one of the purer RISC architectures still in existence. It's also about the only RISC still used for its original purpose as a workstation processor. And it's been the slowest RISC processor family for several years running.

Sun has added visual- and media-processing features to SPARC with VIS (visual instruction set). Similar to MMX or 3DNow!, VIS handles packed RGB-alpha data for compression, decompression, and video-processing. Even with those enhancements, and fully eight generations of design, SPARC processors are all still software compatible, from the first to the most recent. But in an age of PC processors running at 2.0 GHz and up, UltraSPARC-III barely squeaked past 1.05 GHz in January 2002 – and it took TI's latest six-layer copper-interconnect process to do it.

UltraSPARC-III has a 14-stage pipeline, the longest of any of the 64-bit processors reviewed here. In common with most 64-bit CPUs it has six execution units: two for

integer, two for floating-point, one load/store unit and one address-generation unit. With only one load/store unit, UltraSPARC-III can't process multiple memory transactions the same as Opteron or Itanium. UltraSPARC-III can have multiple loads and/or stores outstanding, thanks to its buffers and queues; it just can't dispatch more than one at a time from the code stream.

UltraSPARC-III has average-sized L1 caches of 32K for instructions and 64K for data. The chip contains the L2 cache tags, but not the cache memory itself. That cache is built off-chip using standard SRAMs. There is no provision for an L3 cache at all. Without the L2 caches on the chip, UltraSPARC-III's price is somewhat artificially lower than its competitors.

SPARC's register windowing, an inherent and permanent feature of every SPARC, has so far made it impossible to add multithreading and difficult to keep clock speeds up. Fortunately for Sun, most SPARC processors are buried inside its workstations, where the value of Sun's software and systems-level expertise outshine the relative shortcomings of its processors.

New and Emerging High-Performance Merchant Processors

Intel's Itanium is a clean break, while AMD's Opteron (under the codenames ClawHammer, SledgeHammer and the Hammer family) is essentially an extension to the ages-old x86 architecture. Itanium is a new VLIW design that has x86 compatibility tacked on; Opteron is a real x86 processor – albeit one with 64-bit extensions – just like Athlon, K6, and AMD's other processors before it. The two product lines are now heading down separate paths. Intel's road leads to high-performance servers and workstations, with PCs as an afterthought. AMD's avenue advances PCs as far as the company can take them, with servers benefiting from coherent system bus technology.

Both the IPF and the x86-64 architectures demonstrate elegance combined with surprising eccentricities. The initial response to the debut Itanium processor was muted. Itanium seemed to fail to deliver on years of promises. Itanium 2, however, has silenced the critics by delivering best-in-class benchmarks in almost all categories. It remains to be seen if ClawHammer, AMD's initial x86-64 implementation, will raise the bar yet again.

Itanium Paves a New Path

Itanium is a six-issue VLIW processor, although Intel prefers the term EPIC: explicitly parallel instruction-set computing. Itanium has nine execution units, grouped as two integer units, two combination integer-and-load/store units, two floating-point units, and three branch units. These four groups are significant, as we shall see in a moment.

The processor has a 10-stage pipeline, which is respectable but not impressive by today's standards. For comparison, Pentium III and AMD's Opteron both have 12-stage pipelines but the Alpha 21264 has just eight stages.

Itanium 2 (McKinley) triples system bandwidth over Itanium and increases clock frequency by 25% yet has a shorter pipeline (eight stages versus ten). The newer chip also has a larger L2 cache and a new L3 cache, as well as somewhat different execution resources. Most of Itanium 2's improvements over its predecessor have to do with latency and bandwidth, not raw execution resources, suggesting that the original Itanium was data-starved, not fundamentally flawed as some critics had suggested.

Itanium 2 does have new resources, though, in the form of two additional integer units, bringing the total to four (eleven execution units total). The decode and execute logic is largely unchanged, however, so Itanium's six-issue limitation is still in force for Itanium 2.

Both processors have a massive register set: 328 registers in all, with 128 general-purpose integer registers (each 64 bits wide) and another 128 floating-point registers (each 82 bits wide). The remaining 72 registers are scattered among several different functions, including some for x86 backward compatibility. The first 32 integer registers are global, available to all tasks at all times. The other 96 can be framed, rotated, or both. Before a function call, Itanium's ALLOC instruction (which is unrelated to the C function of the same name) can shift the apparent arrangement of the general-purpose registers so parameters appear to be passed from one function to another. In reality, ALLOC changes the logical-to-physical mapping of the registers, much like SPARC does with its register windows. The size of these registers "frames" is programmable to suit the needs of the calling or called routines.

In addition to the frames, Itanium supports register rotation, a feature that helps loop unrolling more than parameter passing. With rotation, Itanium can shift up to 96 of its general-purpose registers by one or more apparent positions. This allows iterative loops to be unrolled without causing congested access to the same registers. Each instance of the loop actually targets different physical registers, allowing them all to be in flight at once.

Itanium's register-rotation feature is less generic than all-purpose register renaming like Athlon's, so it's easier to implement and faster to execute. Chip-wide register renaming adds multiplexers, adders, and routing, one of the big drawbacks of Athlon's out-of-order machine.

Itanium Debuts New Instruction Set

As a VLIW architecture, Itanium's concept of "instruction" is different from that of more conventional processors such as MIPS, Alpha, or Pentium. Itanium implements the concepts of instructions, bundles, and groups. Instructions are 41 bits long. Each register identifier requires 7 bits to specify one of the 128 general-purpose (or floating-point) registers, so two source-operand fields and a destination field consume 21 bits in

each 41-bit word. Another 6 bits specify the 64 combinations of predication, if any. The remaining 14 bits are the opcode.

Instructions are delivered to the processor in “bundles.” Bundles are 128 bits, consisting of three 41-bit instructions and one 5-bit template (described below). Bundles are a physical convenience only; the three instructions in a bundle are not necessarily dispatched or executed together. Itanium instructions are delivered in 128-bit bundles because that’s a reasonable width for internal buses and decode circuitry. An Itanium bundle happens to hold three complete instructions, but this is not an architectural requirement.

Instruction “groups” are collections of instructions that can execute at once. The instruction groups are the compiler’s way of showing the processor which instructions can be dispatched simultaneously without dependencies or interlocks. Grouping is up to the compiler; the processor will neither optimize groups nor will it check for inter-relationships or dependencies. Groups may be of any arbitrary length.

Bundles and groups are not related. Groups define a logical relationship among instructions; bundles are just 128-bit delivery mechanisms with some added decode information. Instruction groups can be large or small and their grouping will affect the processor’s performance. Finally, optimal grouping depends on the particular IPF chip (Itanium, Itanium 2, Madison, Montecito, Deerfield, etc.).

All Itanium instructions fall into one of four categories: integer, floating-point, load/store, and branch operations. These categories are significant in how they map onto the chip’s hardware resources. Different Itanium processors will have different hardware resources that affect what instructions can be dispatched within a group.

Strange Features of EPIC Proportions

It appears Intel’s track record of baroque and ungainly features affected even Itanium. For example, Itanium opcodes are not unique. The same 41-bit pattern decodes into four completely different and unrelated instructions depending on whether it’s sent to an integer, floating-point, memory, or branch unit for execution. Even with a generous supply of 2^{41} different opcodes, Itanium’s designers apparently decided to skimp on this resource.

A second eccentric feature essentially covers for the first eccentric feature. The five-bit template at the start of each 128-bit bundle helps route the three-instruction payload to the correct execution units. But five bits is insufficient for all the possible permutations of instructions, so instead these bits are used to select one of 24 different “templates” (the other eight combinations are reserved). It is the template that specifies how the instructions are grouped. Yet even the templates don’t support most combinations of instructions; several combinations of integer, FP, branch, and memory operations are simply not permitted.

Strangely, Itanium’s two floating-point units can neither add nor multiply. The FPU is designed for multiply-accumulate (MAC) operations only, so a conventional FP

multiplication is programmed as an FP MAC with an adder of zero. Likewise, simple FP addition is another MAC with a multiplier of 1.0, plus the value being added.

It gets better. Itanium has no integer multiply function at all. *Any* multiplication, whether integer or floating-point, uses the floating-point unit. Unfortunately, that means transferring a pair of integers from the general-purpose registers to the floating-point registers, then transferring the result back again. The Itanium instruction set includes a few operations specifically for this purpose.

New Opteron Features

At its core, Opteron is a nine-way superscalar, massively out-of-order, CISC-into-RISC processor not too different in architecture from Athlon. In many ways, the change from K6 to Athlon was greater than from Athlon to Opteron.

Opteron has nine execution units, the same as Athlon and, coincidentally, the same as the first Itanium. These are grouped into three integer units, three address-generation units, and three floating-point units. Like Athlon and K6 before it, Opteron converts every x86 instruction into one or more internal RISC operations (ROPs). Beyond the first few stages of the pipeline, Opteron is a RISC machine with no concept of x86 instructions, architecture, or machine state.

Opteron can decode up to three x86 instructions and dispatch up to nine ROPs per cycle, if each ROP happens to map to one execution unit. Most ROPs execute directly in hardware, but even after conversion some x86 operations are too complex for that. Like Java byte codes, these are trapped and emulated in Opteron's micro-ROM.

Opteron's pipeline is longer than Itanium's, at 12 stages. Unsurprisingly, most of the difference is spent decoding x86 instructions and converting them to more digestible ROPs.

Extensions to 64 Bits and Multiprocessing

AMD extended the original x86 instruction set to 64 bits, whereas Intel treats 32-bit x86 code as the end of the road. Opteron accomplishes this feat the same way the '386 did: with a size-override byte. Any standard (pre-Opteron) x86 instruction can be prefixed with the one-byte REX pseudo-instruction. This byte tells the decoder that the operands in this instruction should be interpreted as 64-bit quantities. The '386 worked the same way, introducing the 0x66 prefix byte to turn 16-bit operations into 32-bit operations without actually changing the instruction set.

AMD also developed a new set of floating-point operations that use a new flat FP register file of sixteen 128-bit registers. This will be a huge improvement over the spectacularly clumsy register-stack architecture of the original 8087 and all x87 FPUs that followed it.

AMD clearly had multiprocessing in mind for ClawHammer and SledgeHammer, and expects that to be a big differentiator over Itanium 1 and 2. SledgeHammer will have an ambitious system interface that supports two-way, four-way, and eight-way multiprocessing. Opteron chips may have multiprocessing features that Itanium and Itanium 2 lack, but no one doubts that Intel could add those features at any time. The company hardly lacks the wherewithal; it just doesn't see the market demand at present. It'll be much easier for Intel to add multiprocessing features to IPF chips than it would for AMD to add Itanium compatibility to a future Opteron.

Summary Analysis: IPF versus x86-64

Opteron can dispatch nine of its internal ROPs (RISC operations) compared to Itanium's six, which would appear to give Opteron a 50% lead in work-per-cycle. On the other hand, Opteron's ROPs are nothing but decimated x86 instructions, so they don't count for much individually. It usually takes a handful of ROPs to equal one "real" x86 instruction. But the same could be said for Itanium; its native instructions are essentially ROPs as well. *Advantage: Neither.*

Looking deeper, three of Itanium's seven (nine for Itanium 2) non-FP units are branch units. Itanium really has just four integer units to Opteron's three. Two of those do double duty as load/store units, so it's not quite fair to say Itanium could run four integer operations at once – it has to do loads and stores sometime. Opteron, on the other hand, sets aside three address-generation units to this task, so it really *can* execute three integer operations at once. *Advantage: Neither.*

All this changes with Itanium 2. Its two additional integer units give it the edge over Opteron. Yet Itanium 2 can issue "only" six instructions per cycle, the same as Itanium, so the extra integer units aren't as helpful as one might think. *Advantage: Intel.*

Although Opteron and Itanium both have nine execution units, it's hard to say that they'd both accomplish the same amount of work per cycle. Itanium 2, of course, has eleven execution units. Opteron has three address-generation units, which are necessary to the architecture but don't really contribute to forward progress. The Itanium chips have no address-generation units because they support only one simple addressing mode. *Advantage: Intel.*

Three of Opteron's nine execution units are for floating-point operations, leaving six for integer code. Two of Itanium's are used for FP, with the remaining seven free for integer code. That balance, combined with the fact that Itanium's integer units do more useful work suggest that the Intel chip will make more headway on normal code. The balance tips even further in Intel's favor with Itanium 2. *Advantage: Intel.*

Opteron has one more floating-point unit than Itanium has. On the other hand, Itanium's are both equivalent and able to handle any FP operation, whereas all three of Opteron's are different. Itanium wins on elegance but Opteron can potentially get more floating-point work done. *Advantage: AMD.*

AMD's sixteen 64-bit registers are a far cry from Intel's 128 general-purpose plus 128 floating-point registers. Even in its 64-bit mode, Opteron has one-sixteenth the quantity of registers that Itanium has. Most programmers agree that more registers is better, although there comes a point of diminishing returns. *Advantage: Intel.*

Software Support the Key Decider

Will that be enough? It depends on what customers want. Both processor families (IPF with Itanium and Itanium 2) and x86-64 (with ClawHammer and SledgeHammer) are totally, completely, and inarguably backward compatible with x86 binaries. Anything less would be a punishable offense. Customers simply looking for a faster PC may find that Opteron-based systems will run old (or upcoming) PC applications faster than an Itanium- or Itanium 2-based system. That's fine, for as long as customers run x86 binaries. But the day may come when Microsoft Windows version n+1, or Quake XVII, is released for Itanium but not for AMD's x86-64. Microsoft has publicly announced Windows XP support for both IPF and x86-64, so the day of reckoning is postponed for now.

Running existing binaries on either Itanium or Opteron is straightforward, but what about new code? Now, for the first time, software vendors will have to decide: do they support Intel, AMD, or both? Porting major applications and operating systems to Opteron will not be trivial – but neither is supporting Itanium. Backing Intel's newest and heavily promoted next-generation architecture is a foregone conclusion for vendors that want to stay in business. Supporting AMD becomes more problematic. Will the added market share be worth the effort? Suddenly AMD finds itself in the same boat as Apple with a different, yet competitive, product that requires dedicated software support to survive.

Grimly, AMD itself lived through this tragedy not so many years ago, and the wound was self-inflicted. AMD unceremoniously axed its entire 29000 family, one of the most popular processors of the early 1990s, due to the cost of software support. The company decommissioned the second-best-selling RISC in the world because subsidizing the independent software developers was sapping all the profits from 29K chip sales. As "successful" as it was, AMD had to abandon the 29K, the only original CPU architecture it ever created.

Squeezing Blood From the x86 Turnip

Opteron is not a VLIW machine and it doesn't expose parallelism (or anything else) to the compiler. It's just another turbocharged x86: really fast at x86 code but really nothing new in architecture. Newly written Opteron code can access all the new registers, and even treat them as a flat register file, but it can't break free of the inherent awkwardness of the x86 instruction set. The problem is not the binary encoding of x86 instructions; AMD and others have shown they can build blazing fast RISC machines even with that handicap. It's the nonparallel nature of x86 code that's impossible to overcome.

In contrast, Itanium compilers have all the time in the world to locate parallelism, find dependencies, optimize loads, organize branches, develop predicate conditions, and much more, and then express all that richness and intelligence explicitly to the processor. Poor Opteron has to pry what secrets it can from an inherently serial binary stream, searching for tiny fragments of instruction-level parallelism in hardware – and do it all in a handful of nanoseconds without slowing the critical path or impacting the clock frequency. Plus, Opteron has to do all this with no foreknowledge, and no memory, of the entire program. Opteron's scope for reassignment and optimization is limited to the few instructions already in its pipeline. It's like decoding the Rosetta Stone while skydiving, or trying to express Internet routing protocols using Latin or Sanskrit.

Opteron and Itanium are complete opposites when it comes to rescheduling. Itanium does none whatsoever and is damn proud of it. That's the compiler's job and the hardware is just a dumb servant of the compiler, doing what it's told. Opteron, on the other hand, is – has to be – aggressive about locating and exploiting weaknesses (for lack of a better term) in the compiled output, stealthily reorganizing the occasional integer or FP instruction to take advantage of its hardware resources.

Which do you suppose has more headroom, more upside growth potential? It's inconceivable that the Itanium processor family can't be extended and enhanced for another decade or so. Opteron, on the other hand, *is* the result of a decade of stretching, tweaking, and cajoling a Paleolithic architecture into modern form. It's difficult to believe there would be as much life left in Father Time as in the New Year's baby.

Branch latency and its related pipeline bubbles are the bane of high-speed microprocessors. Itanium is the hands-down winner here. It offers complete software and compiler control over branch prediction, hinting, and prefetching, all in addition to its hardware branch-prediction logic. Opteron has to make do with the x86 instruction set, which has no concept of branch hinting and never will, unless AMD chooses to invent new branch instructions.

Itanium's instruction clustering means the hardware doesn't have to check for dependencies; that's the compiler's responsibility. This eliminates a lot of the nastiest, most convoluted hardware from the processor's critical path – exactly the place where Opteron spends most of its effort.

Intel threw out the baby with the bathwater, creating an entirely new microprocessor and gluing x86 compatibility onto the side for sentimental value. AMD keeps straining the same old bathwater. Odd as it seems, AMD now carries the torch for x86, extending it this way and that, while Intel heads down another path. In a sense, AMD will "own" the x86 architecture for high-end processing.

Itanium Was Slow Out of the Gate

If EPIC is so new, why does Itanium run at only 800 MHz, and Itanium 2 at just 1 GHz? Intel's current PC processors run at more than double that speed. Intel could be sandbagging to make the first Itanium processors a bit slower than Pentium 4. As with

Pentium Pro, the idea might be to encourage software developers to switch to the new code base as quickly as possible with a promise of better performance to come. One way to make the new product look good is to make the old one look bad, even if it's your own product.

It could be a heat problem. Lots of transistors generate lots of heat and evacuating that thermal effluvium can limit speed. Still, UltraSPARC-III, MIPS R10000, and even other Intel chips have more transistors than Itanium 2 and they're not soaking in liquid nitrogen.

To some extent, Itanium – and especially Itanium 2 – doesn't have to run at very high clock speeds. Itanium 2 at 1 GHz yields better SPEC benchmark scores than any nearly other processor, at any speed. Its SPECint and SPECfp scores are both better than Power4's, even though the IBM behemoth runs 30% faster and has far more hardware. Itanium 2 cranks out SPECint scores that are similar to a Pentium 4 running at more than twice the clock rate. It appears that Itanium is twice as efficient, clock for clock, as Intel's "other" high-end processor, though this will surprise few observers.

Itanium code density should be as bad as x86 code density is good. That's a bonus for AMD's Opteron, though it's scant consolation; server manufacturers don't choose high-end processors based on code density.

Nor are there many high-end server makers AMD could reasonably hope to woo. IBM and Sun obviously have their own in-house processors; HP/Compaq is just as obviously committed to Itanium. That leaves Dell, usually a staunch Intel supporter, and a handful of second- and third-tier vendors for AMD to convince of Opteron's advantages.

There is a glimmer of hope Opteron from recent history. IBM was losing ground in the 1980s to the PC clones so it took its ball and went home. It changed the game, and called it PS/2 – and promptly lost almost the entire market. Instead of following IBM and switching platforms, the world went right on using PC clones, and IBM never regained the dominance it once had. Maybe Itanium is just the PS/2 of processors, a futile attempt to change the game. Maybe the world really wants a faster x86 instead of a new and different family of processors. Maybe lightning will strike twice. AMD had better hope so.

64-Bit Summary

Processor architecture and design passes through distinct generations, and so it is with today's crop of high-end microprocessors. IPF is the third or latest generation, though it does not always embody the latest academic thinking on CPU architecture. Opteron is the oldest generation, revitalizing an ancient architecture. However, it's really an elaborate RISC machine internally that cracks x86 opcodes, making Opteron perhaps a 1.99-generation machine. Alpha is generation 2. It's pure RISC, has some exotic features, and is very well architected. SPARC belongs in this group, too, though its basic architecture is older than Alpha's and less scalable than PA-RISC. MIPS is of the same

vintage as SPARC, PA-RISC and Alpha, and all bear some similarities. These stand in between Opteron and Itanium in terms of modern thinking. Power4 is in a class by itself, pulling a middle-aged instruction set with it as it develops on-chip multiprocessing and other system-level features.

Although all of these processors are 64-bit machines, most are destined for different and mutually exclusive markets. MIPS has given up the desktop and become a very successful embedded architecture. SPARC has done just the opposite: Sun relies on SPARC for its workstations and maintains its characteristically defiant attitude about other CPUs. Alpha and PA-RISC will fade away on their own, though not because of their performance, development, features, or software support. Alpha and PA-RISC are disappearing by fiat. They have no long-term place in their new environment so they must be eliminated. Power4 is in another world, one dominated by scientists with white lab coats and bulging foreheads.

That leaves Opteron to compete with Itanium *n* for the mainstream server market. Two roads are diverging in the yellow wood of industry standard processors and Intel, for a change, is taking the one less traveled. Whether that's blazing a new trail or abandoning the road to riches remains to be seen. What's certain is that both sides will claim to be on the shining path.

30