VOLUME 13, NUMBER 3
MARCH 8, 1999

MICROPROCESSOR REPORT

THE INSIDERS' GUIDE TO MICROPROCESSOR HARDWARE

Pentium III = Pentium II + SSE

Internet SSE Architecture Boosts Multimedia Performance

by Keith Diefendorff

The name, Pentium III, belies the processor's modest changes. Extrapolating from the differences between Pentium and Pentium II—out-of-order execution, long pipeline, backside L2 cache, and a split-transaction bus—one might have expected Pentium III to be a completely new microprocessor. But the reality is less meaty; Katmai, the first member of the Pentium III family, is essentially a Pentium II with an enhanced multimedia capability. Consumers, however, may not be so disappointed: Katmai's new features will enable improvements in multimedia performance large enough to spawn new applications that are simply intractable on the current generation of Pentium II processors.

Katmai was announced at 450 and 500 MHz, an 11% frequency boost over the 450-MHz Deschutes processor, the current top-of-the-line Pentium II. A 550-MHz version will come out in the second quarter and later this year the second Pentium III, Coppermine, will surface. That processor will be implemented in Intel's upcoming 0.18-micron P858 process (see MPR 1/25/99, p. 22), boosting its frequency to 600 MHz—and eventually to 733 MHz or more—while also making space for at least 256K of on-chip L2 cache.

The new processor uses the same pipeline, same caches, same bus, and same IC process as Deschutes. So, not surprisingly, it achieves the same per-cycle performance on existing software. All of Katmai's new features, save one, are focused on multimedia, as Figure 1 shows. The new features, including 70 new instructions (alias KNI) and a new memory-streaming architecture, are officially dubbed the "Internet streaming-SIMD extensions" (Internet SSE).

The Internet prefix is probably just a ploy by Intel marketeers to catch the wave of Internet enthusiasm and transfer its momentum to their chip. But, while SSE may have nothing to do with the Internet per se, content providers are finding it useful for building soft delivery mechanisms, like soft ADSL modems, and for achieving high data-compression ratios with techniques like 3D NURBS (nonuniform rational B-splines), which are impractical on Pentium II.

The only new feature—or misfeature, depending on your view—in Katmai not related directly to multimedia is the processor serial number (see MPR 2/15/99, p. 3). Contrary to some reports, Katmai does not implement a random-number generator, which would have been a far more useful feature than the serial number.

Streaming SIMD Increases Multimedia Capability

The SSE architecture is likely to be more significant to multimedia performance than MMX ever was. SSE represents a major improvement over MMX for processing video, sound, speech, and 3D graphics. As we pointed out earlier (see MPR 10/5/98, p. 1), Intel did about as well as anyone could expect in defining the architecture, given the x86 starting point (a hole).

What is disappointing about Katmai, however, is that it implements only half of SSE's 128-bit architectural width, double-cycling the existing 64-bit data paths. This approach limits Katmai to only half of the architecture's potential performance, leaving it with little advantage over the K6 III's (see page 22) implementation of 3DNow, which also delivers

Continued on page 6

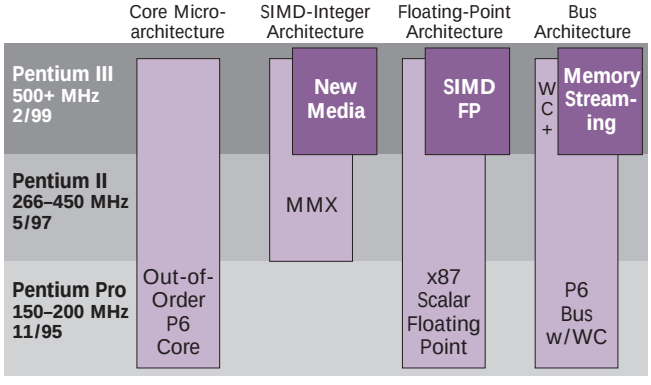


Figure 1. The first Pentium III, Katmai, is based on the same core microarchitecture as Pentium Pro and Pentium II, adding SIMD floating point, new MMX instructions, and memory-streaming features, including prefetch and improved write combining (WC).

AT A GLANCE

The Slater Perspective and Recent ICs will return in the next issue.

EMBEDDED

MICROPROCESSOR REPORT

Founder and Executive Editor
Michael Slater
mslater@mdr.zd.com

Publisher and Editorial Director
Linley Gwennap
linley@mdr.zd.com

Editor in Chief
Keith Diefendorff
keithd@mdr.zd.com

Senior Editor
Jim Turley
jturley@mdr.zd.com

Senior Analyst
Peter N. Glaskowsky
png@mdr.zd.com

Senior Editor
Tom R. Halfhill
thalfhill@mdr.zd.com

Associate Editor: Laurie Masters

Editorial Board

Dennis Allison	Rich Belgard
Brian Case	Jeff Deutsch
Dave Epstein	Don Gaubatz
John Novitsky	Bernard Peuto
Nick Tredennick	John F. Wakerly

Editorial Office
298 S. Sunnyvale Avenue
Sunnyvale, CA 94086-6245
Phone: 408.328.3900 **Fax:** 408.737.2242

Microprocessor Report (ISSN 0899-9341) is published every three weeks, 17 issues per year. Rates are: N. America: $695 per year, $1,295 for two years. Europe: £495 (¤695) per year, £925 (¤1,300) for two years. Elsewhere: $795 per year, $1,495 for two years. Back issues are available.

Published by
MICRODESIGN RESOURCES

Business Office
874 Gravenstein Hwy. So., Suite 14
Sebastopol, CA 95472
Phone: 707.824.4004 **Fax:** 707.823.0504
Subscriptions: 707.824.4001
cs@mdr.zd.com

World Wide Web: *www.MDRonline.com*

Computer Press Award, Best Newsletter, Winner, 1993, 1994, and 1997

Printed on recycled paper with soy ink.

■ THE EDITORIAL VIEW

What's the Best Way to Benchmark?

As EEMBC Wrestles With Testing Conditions, Philosophical Issues Arise

A benchmark is like sex. Everybody wants it, everybody is sure of how to do it, but nobody can agree on how to compare performance.

Part of the problem lies in the fact that microprocessor performance is not a one-dimensional vector. Microprocessor drag racing is all very nice, but the average embedded designer is looking to balance the often-contradictory demands of power consumption, performance, code density, price, interrupt response, and probably other factors. A combination that's good for one application may be unusable for another.

Benchmarking embedded chips is tough, no doubt about it. That's why we have not progressed beyond Dhrystone, the accepted lowest common denominator that any microprocessor can run. Unfortunately, Dhrystone tells us very little about what a microprocessor is good at. One could argue that Dhrystone scores say more about the marketing efforts behind a chip than about its technical features.

By now you've probably heard about the embedded benchmarking work under way at EEMBC (see MPR 4/20/98, p. 13). EEMBC's laudable goal is to eradicate the scourge of Dhrystone in our lifetime. EEMBC (*www.eembc.org*) counts 24 CPU makers, large and small, among its members. For such a diverse group, they've made amazing progress toward standardizing embedded benchmarks. But there may still be some crumbs between the sheets.

Realizing that no single metric can hope to capture the many varied aspects of a chip's performance envelope, the EEMBC benchmark suite consists of dozens of smaller benchmarks. Each test contains a core algorithm taken from real-world code. There are tests for automotive-engine control, codecs, pixel manipulation, task switching, and lots of others. All the tests have been written in ANSI C for architecturally neutral portability.

EEMBC is following a path somewhat similar to that taken by SPEC (*www.specbench.org*), which is a good thing in my opinion. Specifically, EEMBC will allow its members to report two scores for every benchmark: the "out-of-the-box" score and the flat-out, fully tweaked, downhill-with-a-tailwind score. The two scores allow potential users of these chips both to evaluate competing processors under controlled conditions (the basic scores) and to see what each chip is fully capable of, given some care and attention.

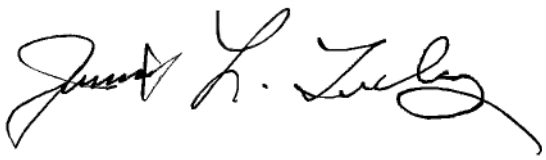
Nobody disputes the need for controlled, nonnegotiable, standardized testing. But I expect some controversy over how best to handle the "tweaked" scores. Exactly how much tweaking is allowed, or desirable? Should testers be allowed to alter the source code of the benchmark? Can they rewrite key algorithms? Can they take shortcuts, like hard-coding lookup tables or—dare we say it—the predetermined results of complex calculations?

The question boils down to deciding what is important to test and what is extraneous. The Heisenberg Uncertainty Principle suggests that the less you want to know, the more accurately you can know it. If your goal is to pin down a given microprocessor's abilities in real-world situations, make sure that's what you're measuring. I believe there should be (almost) no holds barred. Any optimization, from rewriting all the C code, to creating shortcuts, to using unusual chip-specific features or instructions, is fair game in my book. This approach encourages creative and unusual solutions, which are representative of the real world of creative and unusual embedded programmers. As long as the benchmark delivers the correct answers in a reliable and repeatable manner, the details of generating the results shouldn't matter.

It's that "reliable and repeatable" part that makes people nervous. Obviously, simply hard-coding the answers to the benchmark after a few NOPs isn't meaningful. And here's where EEMBC's sister organization, the EEMBC Certification Labs (ECL; *www.embedded-benchmarks.com*), comes in. ECL must first approve every EEMBC member's benchmark scores before those scores can be published. "Approval" in this case means duplicating the same scores in ECL's own facilities. Part of ECL's role is to prove that "tweaked" scores aren't arrived at by nefarious means. That proof includes pumping alternative data sets into the chip under test to be sure that it's really executing the correct algorithm and not just regurgitating prearranged answers.

To me, it seems that a benchmark should test the abilities of a chip, not the skill of EEMBC's programmers. Wide-open testing promotes creativity and allows vendors to exploit the unusual features of their processors. As long as the chip returns the correct result under all conditions, I don't believe that what's inside the black box matters.

Forcing a particular coding convention onto dozens of different microprocessors only discourages programmer innovation and reduces everything to the lowest common denominator. And then we'd be right back to Dhrystone. M



SGI Extends MIPS Map

Silicon Graphics has acknowledged that it is developing new members of its R10000 family in response to the delays in the Merced program announced last year (see MPR 6/22/98, p. 1). In converting from MIPS to IA-64, the company's original plan (see MPR 4/20/98, p. 1) had been to stop shipping MIPS systems by the end of 2000. With Merced now scheduled for a mid-2000 launch, SGI now expects to continue shipping MIPS systems through 2002, even after the debut of the second IA-64 chip, McKinley.

The previous MIPS roadmap ran out after the R14000, now due in mid-2000. To provide its MIPS customers with additional performance upgrades, SGI now plans a follow-on device for early 2001. The unnamed processor (R16000 would be appropriate) will use the same CPU core as the R12000 and R14000, but SGI will revise its layout to improve clock speed. The company expects the revised core will run at 600 MHz, 33% faster than the R14000. Such a sizable increase will involve significant design effort and, potentially, tweaks to the process technology as well.

The new device will take advantage of the R14000's 0.18-micron-generation process to add SRAM to the CPU core. This will probably include moving the L2-cache tags on board, but SGI may not take the PA-8500's approach of including the entire L2 cache. A 0.18-micron process supports a maximum of about 2M of cache on a reasonable die, and SGI prefers larger caches for its big systems.

For further performance enhancements, the plan also includes a 0.13-micron shrink of the new design as soon as that process is available, probably around the end of 2001. If the original design reaches 600 MHz, we estimate the shrink should hit 800 MHz. These extensions to the roadmap give MIPS users more headroom than before, making it more attractive to purchase a new MIPS-based system today. But without a new CPU core, the plan is still likely to leave MIPS falling further and further behind the leaders in uniprocessor performance. —*L.G.*

Chartered, HP Go HiPer (MOS)

Chartered Semiconductor (*www.csminc.com*) and Hewlett-Packard have lined up squarely behind Motorola's copper HiPerMOS process. In a complex multilevel agreement, the two-year-old Chartered/HP joint venture, Chartered Silicon Partners, will license Motorola's copper 0.22-micron HIP5 process (which Motorola labels 0.15 micron after the L_{eff}) as well as its future HIP6 and -7 processes. CSP's new 200-mm fab—now under construction in Singapore—expects to sample its first parts early next year and to be in production by 2Q00. Under this agreement, Motorola will use the CSP fab as a foundry for a portion of its manufacturing operations.

Each company gains something from the arrangement. Chartered, which currently trails rivals TSMC and UMC in process technology, suddenly jumps into the lead with the most advanced process of any foundry in the world, save for IBM. Chartered also locks in two large customers, including one, HP, that will share in the cost of operating its new fab. Chartered will be able to use any excess capacity from the CSP fab to serve high-end customers that its current fabs cannot satisfy. The new agreement also allows Chartered to implement HiPerMOS in its own fabs.

HP gains guaranteed access to a state-of-the-art fab while shedding the awesome cost of operating its own fabs and developing its own processes. HP will continue to build a variety of mixed-signal devices in its existing fabs, but its PA-8500 processor is already too advanced for these fabs. The 8500 is being built elsewhere—we believe at IBM. If so, the similarity between IBM's and Motorola's process technologies might allow HP to shift PA-8x00 fabrication to CSP. HP's aggressive PA-RISC roadmap (see MPR 11/16/98, p. 4) would be well served by the HiPerMOS technology.

Motorola gains an enormous vote of confidence for its process technology. It also gains royalties and licensing fees that it and process-development partner AMD can plow back into technology development, further strengthening the HiPerMOS roadmap. Licensing its technology to Chartered is a critical step toward Motorola's goal of outsourcing 50% of its manufacturing by 2001 (see MPR 1/25/99, p. 10).

What Motorola potentially loses in the deal is the competitive advantage one gains from the proprietary ownership of a high-value technology. Not wishing to cede this advantage completely, Motorola will first use each new process in its own MOS 13 fab for an unspecified period of time before transferring it to CSP. But that period will have to be short, maybe six months, if it is still to be of value to CSP. —*K.D.*

Dixon To Be Intel's First 0.18-Micron Chip

Intel says the first processor to use its new 0.18-micron P858 process (see MPR 1/25/99, p. 1) will be a shrink version of Dixon. The new chip will extend the Mobile Pentium II line (see MPR 1/25/99, p. 20) as an interim step until Mobile Pentium III, aka Coppermine, comes out. The 0.18-micron Dixon will push the high end of the mobile line to 433 MHz. Actually, the part could reach much higher clock speeds, but Intel is likely to fetter the faster Dixon in favor of Coppermine once the Pentium III part is available.

We expect the 433-MHz Mobile Pentium II to appear this summer, as Intel has said it expects to ship its first 0.18-micron products in mid-1999. The company had hoped to go straight to Coppermine, but that design won't be ready to ship until the fall, as Intel is still working out the kinks of its on-die cache. Thus, a simple die shrink of Dixon makes an excellent vehicle for bringing up the new process. In the long run, however, the 0.18-micron Dixon will be a small speed bump in Intel's mobile roadmap. —*L.G.*

Sun to Give Away PicoJava, SPARC Cores

In an unusual move that signals either genius or desperation, Sun Microelectronics is giving away synthesizable models of its SPARC and PicoJava microprocessor cores. The company will make RTL descriptions, verifications tools, and reference materials for its processors available for download beginning the end of this month. Anyone can download, modify, and synthesize the processors for free; Sun will charge a royalty only if customers ship the processors for revenue.

The maneuver is not unlike the open-source movement that is growing in popularity among software developers. Like Linux, Apache, Netscape's Communicator, and other software products, the "source code" for synthesizing Sun's processors will be free for the asking. After enduring the 300-Mbyte download, users may alter the core of the PicoJava or SPARC processors in any way, even if they break binary compatibility with other SPARC or Java processors. Users will be encouraged—but not required—to give any such modifications back to the community, so that third parties may benefit from the enhancements.

Unlike with Linux, this design freedom does not extend to shipping products. Before customers can fabricate and ship for revenue, they must demonstrate compliance with the verification suite included in the download package. Incompatible products cannot be shipped, under Sun's licensing terms.

Users must negotiate royalty terms with Sun before they can ship any chips based on the downloaded designs. Royalty rates are negotiated on a case-by-case basis, so serious customers may wish to arrange terms up front before they begin development in earnest. The PicoJava-I core (see MPR 10/28/96, p. 28) will be available for download at the end of March. Sun expects to make SPARC v8–based cores available by midyear, with SPARC v9 cores coming on line by the end of 1999.

Although Sun's decision to emulate the open-source movement with hardware IP is certainly innovative, it is not clear what effect this move will have on the processor-IP market as a whole. On the surface, it appears to be a good move to broaden the appeal of Sun's two processor families. Developers can evaluate SPARC and/or Java processors with no up-front cost or risk. Sun's license agreement even permits customers to fabricate limited quantities of the chips for internal evaluation. A license for ARM or MIPS, in contrast, generally costs millions of dollars. Lexra, ARC Cores, and Tensilica (see page 12) also charge significant up-front licensing fees for access to their CPU designs.

On the other hand, there's little up-front cost in evaluating these other microprocessors, either. Standard off-the-shelf ARM and MIPS (and, by extension, Lexra) CPUs are available. Both ARC Cores and Lexra allow users to download synthesized designs into FPGAs for development, testing, and evaluation. ARC Cores even lends users the development systems for free. It is only semiconductor vendors, not individual ASIC developers, who must pay the multimillion-dollar fees for MIPS and ARC licenses.

What Sun's unusual community-source arrangement will allow customers to do is tinker with the RTL description of the microprocessor for free, something its competitors charge real money for. Until a customer produces a real SPARC- or Java-based ASIC, no funds are committed. Unlike the case of Tensilica, ARC Cores, or Lexra, however, that tinkering cannot substantially alter the processor, because Sun requires all production chips to pass its compatibility test. In the end, Sun's free-source distribution may appeal primarily to hobbyists, academics, tire-kickers, and frustrated CPU architects—classes of users not known to generate lucrative licensing deals. But it may also encourage grass-roots support for Sun's two CPU families, something that might pay off in a more indirect, long-term way. While the rewards may not be great, the risks to Sun are minimal. —*J.T.*

Tom Halfhill Joins MDR

I'm pleased to announce that Tom R. Halfhill has joined the MDR staff as our new senior editor for embedded microprocessors. Many readers know Tom from *Byte* magazine, where he was senior editor, writing nearly 200 articles about microprocessors, data compression, thin clients, Java, computer reliability, broadband communications, and a variety of other topics.

Before *Byte,* Tom was the editor of several magazines covering home computing and electronic games, such as *COMPUTE!* and *Game Player's*. He has been a full-time technical journalist since 1982, having started his career at daily newspapers in 1977. Tom has coauthored and edited several books, and he still writes a monthly technology column that appears in four magazines.

With Tom aboard, MDR will broaden its coverage of embedded systems in both Microprocessor Report and Embedded Processor Watch. —Jim Turley, Senior Editor

MIPS Signs Texas Instruments

MIPS Technologies has signed DSP powerhouse Texas Instruments as its newest licensee. TI will use MIPS processor cores in future "system-level integration" devices that combine the CPU with TI's own DSP cores. TI will also make the MIPS cores available to its ASIC customers as early as next month.

Continued on page 15

Pentium III

Continued from page 1

4 FLOPS/cycle. Why Intel would sacrifice this much performance for a few measly mm^2 of silicon is a mystery.

The cost of SSE is indeed small. At 128 mm^2, Katmai is only 15 mm^2 larger than Deschutes. According to the MDR Cost Model, the additional silicon adds only 7% to the manufacturing cost of a Katmai module—bringing it to $75. But a full implementation of SSE might not have cost much more. The four-wide AltiVec SIMD-FP unit on Motorola's G4 (see MPR 11/16/98, p. 17), for example, would occupy only 8 mm^2 in Katmai's process; even if it were added on top of Katmai's 128 mm^2, it would add only $3 to the manufacturing cost.

Surprisingly, Katmai's 700,000 extra transistors do not add to its power consumption. To the contrary, improvements to the design actually reduced power dissipation by 7%, to 25 W (max at 450 MHz).

It Won't Fly Without Software

The new chip has the potential to significantly outperform Deschutes on everything multimedia. To realize that potential, however, Katmai must have software written expressly for its new features. Intel has seen to it that the obvious graphics APIs, such as Direct3D and OpenGL, are in place, but the most significant benefits will depend on applications being created or modified for the new features.

The last time Intel made a processor transition that depended on software (when it added MMX to Pentium), it failed to get a critical mass of applications ready in time, blunting the appeal of the new feature. Fortuitously, Pentium/MMX had larger L1 caches to power it through the transition. But such is not the case this time—if Intel wants its new chip to look good, it must get applications in place.

To this end, Intel has, over the past year, mounted a massive campaign—three times the size of any previous Intel software-enablement program—to assist software vendors. The $300 million launch effort was clearly successful, as more than 300 Pentium III–ready applications were shown at the preview event in February. Most of these applications used SSE to some extent, although some just use the serial number.

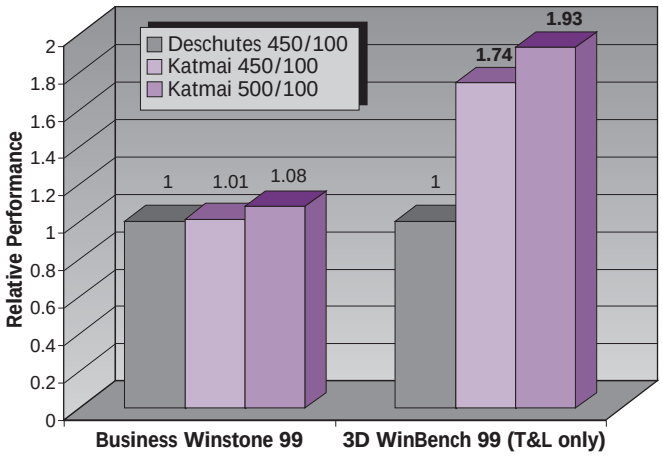


Figure 2. On the 3D WinBench 99 Transform and Lighting CPU Test (rendering nulled out), Intel says that Katmai is 74% faster than Deschutes (although we expect that the overall 3D WinBench speedup will probably be less than 30%). (Source: Intel)

Last summer, Intel decided to pull in the Katmai launch from June to February. The option to do so resulted from better-than-expected silicon, but the impetus was probably a desire to curb the growing software support for AMD's 3DNow and to preempt the announcement of the K6 III.

Although the schedule pull-in is good for consumers, it did foul Intel's 133-MHz-bus plans. Katmai was initially planned to sync up with the Camino chip set in June. But now, without Camino, Katmai is stuck with the BX chip set, limiting it to the same 100-MHz bus as Pentium II. The situation may be a blessing in disguise, as it allows Intel to introduce a 550-MHz Katmai rather than the 533-MHz part it would have been limited to with a 133-MHz bus. Sources now indicate that Camino will be delayed until September, presumably mating up with the 600-MHz Coppermine.

Seventy New Instructions

The SSE features in Katmai divide into two categories: new SIMD instructions and new memory-streaming features. The purpose of SIMD instructions is to increase the rate at which vectorizable data—the kind most common in multimedia applications—can be processed. SIMD instructions give the software a way to express the parallelism that is inherent in this type of data, so the hardware can easily process it in parallel.

SSE introduces new integer and new floating-point SIMD instructions. The integer instructions are really just extensions to the MMX instruction set. Intel, however, refers to these as "new-media instructions." The name makes them easier to explain to customers and avoids lending any credence to the notion that MMX had shortcomings, such as the lack of video-compression capability that we pointed out earlier (see MPR 3/5/96, p. 1).

The new-media instructions, listed in Table 1, will accelerate important multimedia tasks that were poorly served by MMX. For example, the PMAX and PMIN instructions, which are important to the Viterbi-search algorithm used in speech recognition, were notably absent in MMX. Average (PAVG) instructions were added to accelerate video decoding, and Sum of Absolute Differences (PSADBW) was added to speed motion-search in video encoding.

To save silicon, Intel used a clever trick to implement PSADBW. The instruction is issued as three µops: the first computes the differences (A_i–B_i) and carry-outs (C_i) of each of the byte elements in the two source operands; the second computes the absolute values ($|A_i–B_i|$) of the intermediate results; and the third sums the eight absolute values ($\sum_{i=0\ldots7}|A_i–B_i|$). The trick was to use the Wallace tree in the SIMD multiplier to perform the final summation. With this approach, PSADBW added only 2% to the area of the SIMD integer unit.

Mnemonic	Description	SIMD	Scalar	Integer	FP	Port	Latency	Thruput
Floating-Point Instructions (use SIMD-FP registers)								
ADD	Add elements ($D_i = D_i + S_i$, i = 0…3)	•	•		•	1	4	2
SUB	Subtract elements ($D_i = D_i - S_i$)	•	•		•	1	4	2
MUL	Multiply elements ($D_i = D_i \times S_i$)	•	•		•	0	5	2
DIV	Divide elements ($D_i = D_i \div S_i$)	•	•		•	0	18–36	
SQRT	Square root of elements ($D_i = f(S_i)$)	•	•		•	0	29–58	
MAX	Maximum of elements ($D_i = f(S_i)$)	•	•		•	1	4	2
MIN	Minimum of elements ($D_i = f(S_i)$)	•	•		•	1	4	2
RSQRT	Reciprocal square-root estimate (12-bit accuracy) ($D_i = f(S_i)$)	•	•		•	1	2	2
RCP	Reciprocal estimate (12-bit accuracy) ($D_i = f(S_i)$)	•	•		•	1	2	2
CMP	Compare elements using eq, lt, le, unord, neq, nlt, or nle returning Boolean mask	•	•		•	1	4	2
COMISS/ UCOMISS	Ordered/unordered compare scalar element setting condition flags		•		•	1	1	1
ANDPS	Bitwise logical AND of elements	•			•	1	2	2
ANDNPS	Bitwise logical AND of elements w/complement of one operand	•			•	1	2	2
ORPS	Bitwise logical OR of elements	•			•	1	2	2
XORPS	Bitwise logical XOR of elements	•			•	1	2	2
New-Media Instructions (use MMX registers)								
PINSRW	Insert 16-bit value from general register into one of four elements (specified by immediate)	•		•		0,1	4	1
PEXTRW	Extract one of four elements (specified by immediate) to a general register	•		•		0,1	2	2
PMULHU	Multiply four 16-bit unsigned elements returning most significant 16 bits of each	•		•		1	3	1
PSHUFW	Full shuffle of 16-bit elements under control of 8-bit immediate mask	•		•		1	1	1
PMOVMSB	Move 8-bit mask composed of MSbs of byte elements to a general register	•		•		1	1	1
PAVGB	Average of byte elements ($D_i = -(D_i + S_i + 1)/2$, i = 0…7)	•		•		0,1	1	0.5
PAVGW	Average of 16-bit elements ($D_i = -(D_i + S_i + 1)/2$, i = 0…3)	•		•		0,1	1	0.5
PSADBW	Sum of absolute value of differences of 16-bit elements ($D_{1,0} = \sum_{i=0\ldots3} \lvert D_i - S_i \rvert$)	•		•		0,1 *	5	2
PMINSW	Minimum of signed 16-bit elements	•		•		0,1	1	0.5
PMINUB	Minimum of unsigned byte elements	•		•		0,1	1	0.5
PMAXSW	Maximum of signed 16-bit elements	•		•		0,1	1	0.5
PMAXUB	Maximum of unsigned byte elements	•		•		0,1	1	0.5

Mnemonic	Description	SIMD	Scalar	Integer	FP	Port	Latency	Thruput
Type-Conversion Instructions (use MMX and SIMD-FP registers)								
CVTSS2SI	Convert FP scalar to integer in MMX register		•	•	•	1,2	3	1
CVTTSS2SI	Convert FP scalar to integer in MMX register		•	•	•	1,2	3	1
CVTSI2SS	Convert integer in MMX register to FP scalar		•	•	•	1,2	4	2
CVTPS2PI	Convert two FP elements to integers in MMX register	•		•	•	1	3	1
CVTTPS2PI	Convert two FP elements to integers in MMX register	•		•	•	1	3	1
CVTPI2PS	Convert two integers in MMX register to FP scalars	•		•	•	1	3	1
Data-Movement Instructions (use SIMD-FP registers)								
MOVA	Load/store/move an aligned 128-bit vector or 32-bit scalar (fault on misaligned)	•	•		•	2 3/4 0,1	4 1 4	2 1 2
MOVUPS	Load/store an unaligned 128-bit vector from/to memory	•			•	2 3/4	4 5	2 3
MOVLPS/ MOVHPS	Load/store 64-bit mem operand to/from low/high half of register	•			•	2 3/4	3	1
MOVLHPS/ MOVHLPS	Move lower/upper 64 bits of src reg into upper/lower 64 bits of dest reg	•			•	0,1	1	1
MOVMSKPS	Move 4-bit mask composed of MSbs of four elements to a general register	•			•	0	1	1
SHUFPS	Move any two elements from two source registers to dest register under control of 8-bit mask	•			•	1	2	2
UNPCKHPS	Interleave high two elements from two source regs to dest register	•			•	1	3	2
UNPCKLPS	Interleave low two elements from two source regs to dest register	•			•	1	3	2
State Save/Restore Instructions (use SIMD-FP registers)								
FXSAVE	Save registers to memory	•			•		µcode	
FXRSTORE	Load registers from memory	•			•		µcode	
STMXCSR	Save the SIMD-FP status and control register to memory		•	•			µcode	
LDMXCSR	Load the SIMD-FP status and control register from memory		•	•			µcode	
Streaming/Prefetching Instructions								
MOVMSKQ	Store MMX register to memory under byte mask	•		•		0,1 3/4	4	1
MOVNTQ	Store MMX register to aligned mem minimizing cache pollution	•		•		3/4	3	1
MOVNTPS	Store SIMD-FP register to aligned memory minimizing cache pollution	•			•	3/4	4	2
PREFETCH	Load cache line containing addressed datum into specified levels of the cache hierarchy	•	•	•	•	2	2	1
SFENCE	Ensure all prior stores are globally visible (flush WC buffers)					3/4	3	1

Table 1. The SSE architecture introduces new SIMD floating-point instructions, new SIMD integer instructions, and new memory-streaming instructions. The SIMD floating-point instructions operate on four-element vectors of IEEE-754 single-precision values in a new 128-bit eight-entry register file. The SIMD-FP instructions also have a scalar mode that operates on only the rightmost element of vectors. Intel calls the SIMD integer instructions new-media instructions. SIMD-FP multiplies are dispatched to port 0, allowing them to be issued in parallel with SIMD-FP adds, which are dispatched to port 1. Most of the new-media instructions can be dispatched to either port 0 or port 1 (shown as 0,1). *PSADBW requires three µops; the first two can execute on either port 0 or 1, while the third requires port 0. Load instructions use port 2, while store instructions require both ports 3 and 4 (shown as 3/4). Instruction latencies are shown in cycles and throughputs in cycles per instruction. In scalar mode, floating-point instruction latencies are one cycle shorter than shown. (Source: Intel)

SIMD-FP: the Main Attraction

The most significant new feature of Katmai is SIMD floating point. Initially, the feature will be used mainly to accelerate 3D graphics, as Figure 2 shows (see page 6). But Intel expects Katmai's high SIMD-FP performance to spur the use of floating-point data in many other signal-processing algorithms as well, making the feature broadly applicable.

Architecturally, the SIMD-FP feature introduces a new register file containing eight 128-bit registers, each capable of holding a vector of four IEEE single-precision floating-point data elements. The SIMD-FP instructions perform arithmetic on the respective elements in two registers, returning a four-element result vector to one of the two registers. Thus, the architecture allows four single-precision floating-point operations to be carried out with a single instruction and, in a full implementation, to achieve a throughput of four multiply or four add operations per cycle.

To avoid completely redesigning the core microarchitecture, however, Intel had to shoehorn the new SIMD-FP architecture into the Deschutes core. Since Katmai is built in the same 0.25-micron process as Deschutes, it also had to implement the feature using as little silicon as possible. To achieve these goals, Intel implemented the 128-bit architecture by double-cycling the existing 64-bit data paths and by merging the SIMD-FP multiplier with the x87 scalar floating-point multiplier into a single unit, as Figure 3 shows.

To utilize the existing 64-bit data paths, Katmai issues each SIMD-FP instruction as two µops. To avoid the possibility of leaving the machine in an imprecise architectural state between µops, Intel devised a check-next-µop mechanism to hold off the register update from the first µop until it is determined that the second µop will complete without generating an exception (e.g., overflow). The check-next-µop feature has no effect on the execution pipeline but can block retirement for an extra cycle. This potential penalty is avoided when exceptions are disabled, which is the normal mode.

To partially compensate for implementing only half of SSE's architectural width, Katmai implements the SIMD-FP adder as a separate unit on the second dispatch port, as Figure 3 shows. This organization allows half of a SIMD multiply and half of an independent SIMD add to be issued together, as Figure 4 shows, bringing the peak throughput back to four floating-point operations per cycle—at least for code with an even distribution of multiplies and adds.

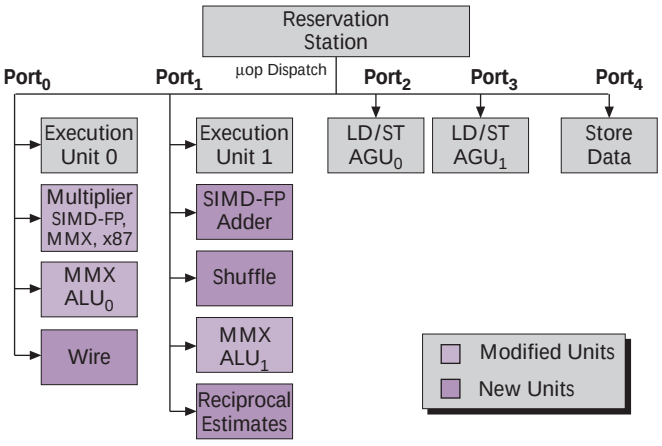


Figure 3. Katmai adds new execution units and modifies others while preserving the P6's basic five-port dispatch mechanism.

Although this situation is common in DSP algorithms, the adds are usually data dependent on the multiplies. Thus, for Katmai to realize its peak throughput, the code will have to be scheduled to cover the latencies. With only eight registers and a destructive two-operand (i.e., $R_D \leftarrow R_D$ op R_S) instruction format, however, such a code schedule will be difficult to achieve. To help with this problem, Katmai can issue two register-move instructions simultaneously.

The Katmai implementation of the SIMD-FP architecture adds new units for the shuffle instructions and reciprocal estimates. The new units increase the loading on the dispatch ports, which would have compromised frequency if not for the enormous effort Katmai's engineers put into tuning critical circuit paths. Apparently they were successful, as Katmai will actually be 50 MHz faster than Deschutes.

Although Katmai's implementation leaves some of the potential of the SSE architecture on the table, the tradeoff for silicon area is understandable. But the danger with Katmai's method is that the hardware has implemented a different model of parallelism than is implied by the architecture. This sets up a code-scheduling dilemma: Should the code be scheduled for Katmai to maximize near-term performance, or would it be better to schedule for the architecture in anticipation of a full implementation in a future processor? Sources indicate that Coppermine will use the same core as Katmai, so a full implementation of SSE is not likely until Willamette, which isn't expected until 2H00.

SIMD-FP Architecture Is Full IEEE-754

Pentium III's SIMD-FP instructions adhere to the IEEE-754 specification, including all four rounding modes and the maskable numeric exceptions. Unlike the x87 scalar FPU, which implements numeric exceptions in microcode, Katmai's SIMD-FP units implement all exceptions in hardware,

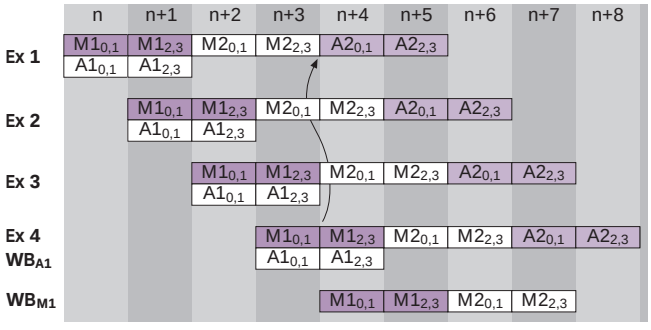


Figure 4. Katmai can execute a SIMD multiply (M1) and an independent add (A1) every two cycles. A dependent add (A2←M1) suffers a four-cycle delay.

except underflows and denormalized results. This is crucial to SIMD performance, since any exception within a four-element vector can trigger a microcode-recovery routine.

To prevent microcode processing from hampering performance, the architecture provides a mode to flush underflows to zero (FTZ). Although operation in this mode is not IEEE-754 compliant, it is acceptable for most 3D-graphics applications, and it will be useful in many other algorithms where IEEE accuracy is less important than raw speed.

A new control-and-status register (the MXCSR) is provided to report the IEEE exception flags and to control the FTZ mode, the directed-rounding modes, and the exception enables. Unmasked exception traps are delivered through interrupt 19, a new x86 interrupt assigned for that purpose.

More Than Just SIMD-FP Arithmetic

By performing operations on multiple data elements simultaneously, SIMD architectures impose difficulties not present with scalar machines. Means must be provided to make control-flow decisions on individual data elements, to perform operations on a subset of elements within a vector, and to reorganize the data so the SIMD units can be fully utilized.

For data-dependent decisions, the architecture provides instructions that compare elements in two SIMD-FP vectors according to one of the relations: equal, less-than, less-than-or-equal, unordered, not-equal, not-less-than, or not-less-than-or-equal. The result of a compare instruction is a four-element Boolean-result vector (true = 0xFFFFFFFF, false = 0x00000000) that can be manipulated with a set of SIMD logical instructions (AND, ANDN, OR, XOR) to mask elements within a vector. A MOVMSKPS instruction allows the most significant bit of each Boolean element to be deposited into an x86 general register. From there, control-flow decisions can be made on any of these four bits.

The need often arises to transpose data elements from a row to a column organization so the SIMD parallelism can be exploited. Consider the example of 3D coordinates organized as WZYX vectors in memory, as Figure 5 shows. To transpose these into vectors of like coordinates, the architecture provides MOVHPS, MOVLPS, and SHUFPS instructions. With these instructions, 3D transformations (without clipping) can be performed with only a 25% overhead, compared with data that has been preorganized and stored as $X_3X_2X_1X_0$. Additional arithmetic operations further amortize the overhead, typically into the 5–10% range.

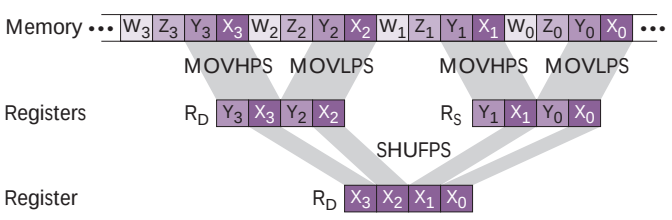


Figure 5. With the MOVHPS, MOVLPS, and SHUFPS instructions, Pentium IIIs can transpose vectors with only a small overhead.

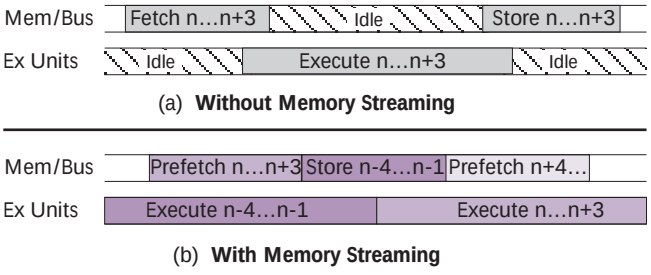


Figure 6. Prefetch instructions allow memory fetches to overlap execution, increasing bus and execution-unit utilization.

Memory Streaming Feeds Execution Engine

A downside of SIMD engines is that they can easily increase the processing rate to a level above the memory system's ability to supply data. When that happens, the execution units stall or become poorly utilized, thus limiting the SIMD speedup. Such would have been the case for Katmai, as the 3D-transform-and-lighting example in Figure 6(a) illustrates, had Intel done nothing to improve memory performance.

But Intel increased the throughput of the memory system and the P6 bus, as Figure 6(b) shows, with a set of features it collectively calls memory streaming. These features include prefetch instructions, streaming stores, and enhanced write combining (WC). As with the SIMD code itself, the effectiveness of these measures will depend on software's use of the new instructions and on a good code schedule.

Memory-streaming features have been used before in RISC processors. Although they have proved effective in many specific situations, they have suffered in general from three problems: they do not lend themselves to use with the random memory-access patterns found; they are notoriously difficult to apply effectively due to the dynamics of the run-time environment; and compilers have not been trained to use them, so they do not have broad applicability.

Multimedia applications, however, rarely have random access patterns, and Katmai's prefetch cachability should help control the dynamics. Rapid advances are also being made in data prediction, allowing compilers to use prefetch instructions more effectively than in the past.

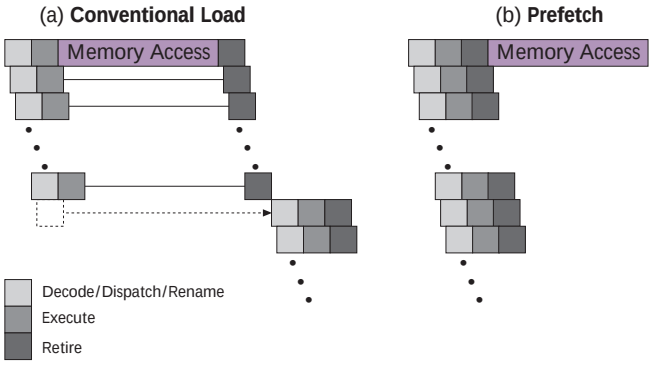


Figure 7. In-order retirement can cause completion resources to saturate following a load, stalling the pipeline (a). Prefetches retire ahead of the memory access, freeing completion resources (b).

Prefetch Hides Latency, Frees Resources

For multimedia streams, loads have shortcomings. Because they can fault (e.g., on a protection violation), loads cannot generally be hoisted (moved up in the code sequence) far enough to cover very much memory latency. When they are hoisted, they often cause the machine's in-order completion resources (e.g., reorder-buffer entries) to fill up, limiting the number of instructions that can be in flight, as Figure 7(a) shows. Furthermore, loads give no clue to the temporal locality of their data, so cache optimization is impossible.

Katmai's prefetch instructions address these shortcomings. Being only hints, prefetch instructions do not fault or modify architectural state. As a result, they can be hoisted arbitrarily far and retired before the memory access completes, as Figure 7(b) shows, freeing completion resources so more instructions can be dispatched. Plus, Katmai's prefetch instructions provide suggestions to the hardware regarding the cache level to which the addressed line should be allocated. Options include all cache levels, L1 only, all levels above L1, and cache bypass (fetching to a temporary read buffer).

Katmai also adds a streaming store, which, on a miss, bypasses the cache without allocating a new line. This mechanism allows software to avoid polluting the caches when it knows the data being stored will not be accessed again soon.

To improve the performance of streaming stores, Katmai enhanced the write-combining (WC) feature that has been in all P6-bus-based processors. Previous P6 processors used a single cache-line buffer (32 bytes) to collect sequential write-through stores so they could be pushed out as a single bus transaction, improving bus efficiency. Katmai increases to four the number of WC buffers, as Figure 8 shows, and also improves the buffer-management policies to increase their effectiveness. (Actually, Katmai doesn't add new buffers; it just uses Deschutes' existing load buffers for double duty.)

The WC buffer allocation and flush policies were improved to reduce the average buffer occupancy. In Deschutes, the WC buffer was not flushed until a new request forced it to be pushed out; Katmai, on the other hand, automatically empties the buffer as soon as it fills. Intel insists that the new WC structure is effective, citing simulations that show it rarely, if ever, degrades load performance but often substantially improves write-through performance. An SFENCE instruction was also added to give software a means of manually flushing the WC buffers when it is necessary to ensure that all prior stores are globally visible (e.g., for screen updates or for software-managed coherence).

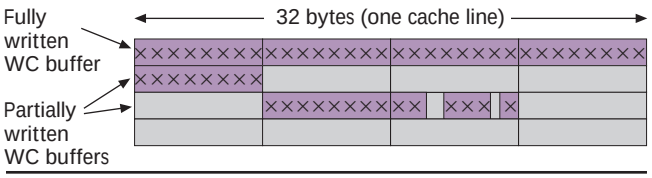


Figure 8. Katmai reuses Deschutes' load buffers for write combining. Intel says that, in practice, conflicts between the two uses rarely occur.

Not a New Core, But Not Chopped Liver Either

Intel's choice of name set the expectation of a new microprocessor core. Against that expectation, Katmai fails. But that goal may have been wrong anyway. A new core may have—with luck—gained a 50% speedup. While that speedup may have made for faster word processors and spreadsheets, it would hardly have enabled anything fundamentally new.

By focusing efforts on multimedia features instead of a new core, however, Katmai will accelerate—by several times—the critical underlying multimedia algorithms. In one speech application, training time improved 60×. Real-time MPEG-2 encoding, impossible on a Pentium II, is achievable on Katmai. Such speedups can, and will, enable new applications, a feat that no practical degree of superscalarity or frequency boost could have achieved.

From this perspective, it's hard to argue that Intel made a mistake. With SSE in Katmai, Intel has set the software stage for the big frequency and memory-bandwidth improvements that will come with Coppermine. But if the focus was on SSE, Intel should have gone all the way. By shoehorning SSE into the existing 64-bit Deschutes core, and by being so stingy with silicon, Intel sacrificed too much.

Unwillingness to revamp Deschutes' instruction decoder forced Intel to stick with the x86 instruction format, limiting to eight the number of registers that could be included in SSE, and constraining the instructions to a destructive two-operand format. Reuse of Deschutes' 64-bit data paths sacrificed nearly half of Katmai's multimedia-performance potential and set up a code-scheduling dilemma with long-term consequences—all for marginal silicon savings.

Not Enough Distance Between P III and K6 III

Although processor architects may find Katmai underwhelming, it is sure to be a commercial success. On competitive grounds, however, the chip may be somewhat of a tactical blunder. In too many ways, AMD's K6 III (see page 22) is an even match for Katmai.

SSE does have advantages over 3DNow. SSE uses full IEEE-754 arithmetic, allowing it to serve scientific applications that 3DNow can't. And 3DNow lacks both SSE's new-media instructions, which boost video performance, and its memory-streaming features, which, along with the P6 bus, give Katmai substantially higher delivered bus bandwidth. SSE's new register file gives it a big theoretical advantage over 3DNow, although that advantage is diminished by the destructive two-operand instruction format.

Despite Katmai's half-wide implementation, SSE's largest advantage may still be its 128-bit architectural width. This width allows twice the parallelism to be expressed per instruction as with 3DNow—a fact that reduces the degree of loop unrolling needed to expose parallelism and puts less pressure on the register file, the instruction cache, and instruction issue and reordering resources. This headroom should allow Katmai to sustain higher throughput in practice, even though the K6 III may have the same peak rating.

Price & Availability

The Pentium III processor (Katmai) is available now at a quantity-1,000 list price of $696 for the 500-MHz version and $496 for the 450-MHz version. Both are offered in the SECC2 (Slot 1) module.

A Xeon version of Katmai, code-named Tanner, is expected to be announced in March, with delivery a month later. Intel has also disclosed that a 550-MHz Katmai will be available in 2Q99.

But while Katmai's advantages are not insignificant, they are also not overpowering. By not pulling out the stops on the architecture and on Katmai's implementation of it, Intel may have blown its chance to knock out 3DNow and to regain complete architectural control of the x86 platform.

Diffusion Key to Success

Katmai's technical advantages do not seem so compelling that they alone can overcome 3DNow's 15-million-unit installed-base head start. To complete that job, Intel must rely on its persuasive powers to divert software developers' attention from 3DNow. It must also—quickly—drive the SSE architecture across its product line and down into the Celeron space, where it can build a large enough installed base to attract software developers on its own.

As it normally does with new microprocessors, Intel introduced Pentium III at the top of its product line. The company has set the initial 1,000-unit list price of the 500-MHz part at $696; $496 for the 450-MHz version. At the same time, it dropped the Pentium II-450 price to $476, while the -400 and -300 fell to $264 and $192.

Thus, at the same frequency, Intel is charging only a $20 (4%) premium for SSE, indicating that it is serious about a rapid transition to SSE-based Pentium IIIs. We expect the SSE premium to disappear quickly and Pentium III to come rapidly down Intel's price curve, approaching $200 by year end. By that time, the 600-MHz Coppermine will have replaced Katmai at the higher price points.

The SSE architecture, however, will not make it into the Celeron line until 2000. This should happen in the first quarter, as we expect Intel to introduce a version of Coppermine, possibly with half the on-chip cache, into that space. Once the Celeron space is breached, SSE will serve as a powerful differentiator that could leave AMD in the lurch.

All Hopes Riding on SSE

Even if SSE bests 3DNow, Katmai will still not claim the desktop-multimedia high ground. Apple, whose desktop market share is once again on the rise, will this summer field G4-based platforms. The G4 processor provides a full implementation of Motorola's 128-bit AltiVec multimedia architecture (see MPR 5/11/98, p. 1). With 32 registers, four-operand nondestructive instructions, multiply-add, 32 × 16 bytewise permute, select, multistream prefetching, and a host of other powerful features, the G4 should outperform Katmai on all multimedia algorithms, regardless of frequency differential. Although PowerPC is not a direct competitor, the pesky Apple could sap some software development resources that Intel would prefer to have on SSE and Katmai.

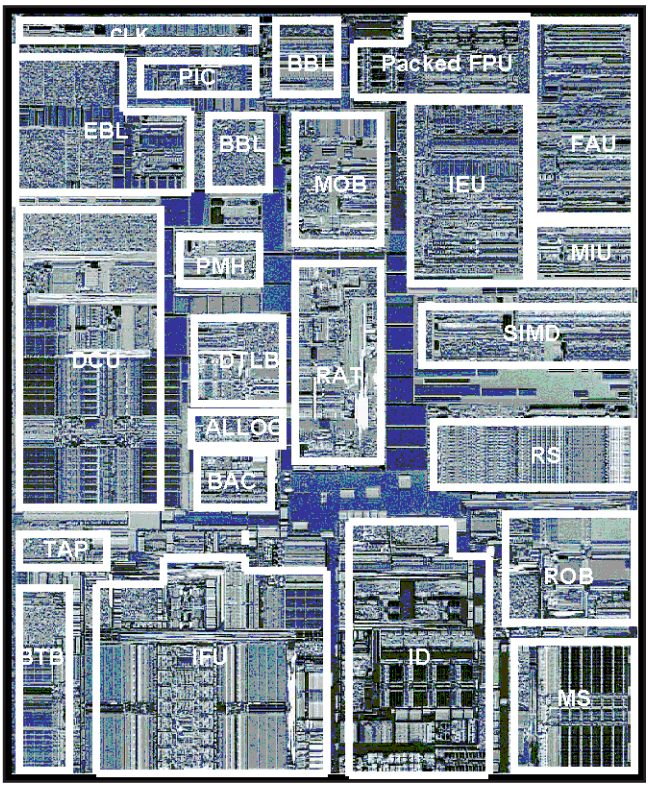


Figure 9. The first Pentium III processor, Katmai, implements 9.5 million transistors and measures 12.3 × 10.4 mm in 0.25-micron P856.5 with five layers of aluminum interconnect. (Source: Intel)

With the K6 III nipping at Katmai's heels, Intel must also contend with the K7 this summer. If AMD can execute, which is not yet a sure thing, the K7 could actually grab the performance lead on general applications, and possibly even 3D. Intel's secret wish that AMD will stumble getting K7 into production at frequency, as it did with K6, appears less likely to come true with each passing day. AMD is probably breathing a sigh of relief that Intel was not more aggressive with Katmai. For the first time, it is possible that AMD could have Intel surrounded. How this could happen is a question that Intel execs are undoubtedly contemplating.

But Intel can potentially blunt the impact of the K7 by getting quickly to a 0.18-micron process with Coppermine. This move should allow Intel to stay ahead of the K7 on frequency and within marketing distance of it on performance. If Intel can avoid missteps, Katmai and Coppermine should safely bridge the company to Willamette, with only a few scrapes and bruises along the way. If it makes the transition safely, we expect that the company will never again yield to the complacency that, in this round, has allowed a competitor to get so dangerously close. M

Tensilica CPU Bends to Designers' Will

Xtensa Processor Can Be Configured by Customers Before ASIC Synthesis

by Jim Turley

Proving there's no shortage of new ideas in processor design, Silicon Valley startup Tensilica has made the first public announcement of its new processor design. Tensilica's CPU core differs from most others in that it can be configured and customized by an individual ASIC developer. Tensilica bundles its basic CPU design with design tools that allow ASIC developers to create their own application-specific extensions.

Tensilica's RISC-inspired processor mixes 16-bit and 24-bit instructions, an orthogonal register file, and 78 immutable "base case" instructions. Customers are then free to add their own special-purpose instructions by defining them, using a subset of the Verilog hardware-description language. The resulting mix is then synthesized and fabricated as part of the larger ASIC design process.

Tensilica is poised to compete with better established 32-bit core vendors such as ARM and MIPS as well as fellow newcomers like Lexra and ARC Cores. Predictably, Tensilica claims its processor design is both more powerful than any of the existing processor-core alternatives and more flexible. Any performance claims are so far tough to verify, but in terms of flexibility, Tensilica may have a point.

Headed by All-Star Cast and Directors

Tensilica's org chart reads like a who's who of RISC and synthesis luminaries. The management and engineering teams include MIPS/SGI alumni Ashish Dixit, Earl Killian, Woody Lichtenstein, Dror Maydan, Chris Rowen, John Ruttenberg, and Keith van Sickle as well as Synopsys graduates Harvey Jones, Bernie Rosenthal, and Albert Wang. Beatrice Fu, late of Intel, serves as Vice President of Engineering. John Hennessey (Stanford), Kurt Kreutzer (UC Berkeley), and Monica Lam (Stanford) serve on Tensilica's technical advisory board.

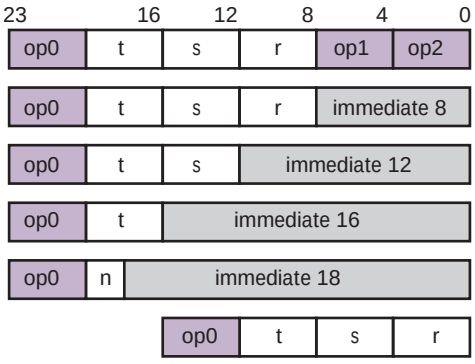


Figure 1. Xtensa instructions are encoded in either 16 or 24 bits, with the most significant bit determining word size.

The assembled combination of processor-design and synthesis-tool experience was a natural for Tensilica's goal: to push CPU design decisions down the microprocessor food chain to the ASIC designer. Rather than create yet another embedded microprocessor and offer it up for licensing, Tensilica has chosen to develop a set of tools that lets ASIC designers create their specialized microprocessor, merge it with their custom logic and memory, and synthesize the whole thing to create an ASIC.

To that end, Tensilica (the name is a play on tensile, as in flexible, silicon) created a CPU core descriptively named Xtensa. Tensilica president Rowen believes that "traditional" licensed CPU cores, such as MIPS, ARM, SPARC, and PowerPC, force the ASIC customer to "design around" a fixed processor that has been licensed to a limited number of semiconductor fabricators. The customer's algorithms and software must then be mapped onto the architecture and instruction set of that processor, a situation that has been true from the earliest days of digital computers.

A user-configurable microprocessor, the argument goes, can be customized by those who know the application's requirements and algorithms best: the customers. Rather than map the software onto a fixed instruction set, the instruction set can, to some extent, be mapped onto the algorithm.

All Processors Include 78 Basic Instructions

Looking at just the base-level Xtensa processor, one finds little to differentiate it from dozens of other RISC-inspired CPUs. One oddity is that Xtensa mixes 16- and 24-bit instruction words; there are no 32-bit instructions. The most-significant bit in each opcode identifies the instruction size, as Figure 1 shows. Most instructions specify three register operands, or two registers and an immediate (literal) value. Most operations are nondestructive, and Xtensa follows a load/store memory model. Tensilica believes Xtensa's reliance on short instruction words enhances the CPU's code density, negating the need for code compression.

The base instruction set, which all Xtensa processors will share, consists of 78 instructions, listed in Table 1. The base ISA includes the usual logical and arithmetic operations, as well as some not-so-common conditional moves and zero-overhead loop instructions. By maintaining a core set of instructions for all Xtensa processors, Tensilica can guarantee at least some level of software compatibility among different customers' implementations. This allows a single operating-system port to run on all Xtensa-based CPUs.

Register Windows Make a Return

Tensilica has revived register windowing with Xtensa, though not the same way that SPARC processors implement the feature. The register file is logically maintained as a circular queue, as in SPARC processors, though only 16 registers are visible at any one time. On subroutine calls, programmers can choose to overlap the called procedure by zero, four, eight, or twelve registers. The overlapped registers pass parameters into and out of called procedures without pushing and popping values on a stack. The Xtensa C compiler automatically selects the maximum amount of overlap needed to preserve static data.

With no register overlap, Xtensa's 32-register file can maintain two concurrent processes without overflowing. (Developers can optionally double the number of registers, to 64, with a synthesis option.) Specifying a nonzero register overlap increases the potential number of concurrent processes. When the number of physical registers is exceeded, Xtensa spills 4, 8, or 12 registers to an external stack. The physical register file is refilled when code executes an RETW to a call frame that is not in the register file.

Register windows have never been a popular architectural feature because they are either huge, awkward to maintain, and take a long time to switch contexts (as in SPARC), or they are so small that they provide little benefit. Windowed registers are a bit more useful in embedded systems, which generally have few (or no) independent tasks or contexts. The shallower call/return stack of embedded systems makes small circular register files (as in PicoJava, TriCore, and the PSC1000) more fruitful.

Designer-Defined Instructions

The real charm of Xtensa is not its basic architecture but its extensibility. Tensilica allows ASIC designers to extend Xtensa's instruction set using a software tool called TIE (Tensilica instruction extension language), a subset of Verilog. Customers write TIE scripts that define the mnemonic, the binary opcode, and the input and output functions of the new instruction. Tensilica's synthesis tools then automatically create the decoding logic required to implement the new instruction.

Tensilica's flexibility is not total. There are significant limits to the changes a customer can make. All Xtensa processors start with the same hardware resources (ALU, multiplier, data paths, etc.) and the same basic set of 78 instructions. To this baseline architecture, customers can add an arbitrary number of new instructions (at least, until they deplete the unused opcodes).

Mnemonic	Description	Mnemonic	Description	Mnemonic	Description
Arithmetic		**Conditional Branches**		**Data Transfer**	
ADD	Add	BEQ	Branch if equal	L8UI	Load 8 bits, unsigned
ADDI	Add immediate	BEQI	Branch if equal, immediate	L16SI	Load 16 bits, signed
ADDX2	Add, shift by 1	BEQZ	Branch if equal to zero	L16UI	Load 16 bits, unsigned
ADDX4	Add, shift by 2	BNEZ	Branch if not equal to zero	L32I	Load 32 bits
ADDX8	Add, shift by 3	BGE	Branch if greater/equal	L32R	Load 32 bits, PC-relative
ADDMI	Add immediate, shift by 8	BGEI	Branch if greater/equal, immediate	S8I	Store 8 bits
SUB	Subtract	BGEU	Branch if greater/equal, unsigned	S16I	Store 16 bits
SUBX	Subtract, shift by 1	BGEUI	Branch if greater/equal, uns, imm	S32I	Store 32 bits
SUBX4	Subtract, shift by 2	BGEZ	Branch if greater/equal to zero	MOVEQZ	Move if equal to zero
SUBX8	Subtract, shift by 3	BLT	Branch if less than	MOVGEZ	Move if greater/equal to zero
Shift and Logical		BLTI	Branch if less than, immediate	MOVI	Move immediate
AND	Logical AND	BLTU	Branch if less than, unsigned	MOVLTZ	Move if less than zero
OR	Logical OR	BLTUI	Branch if less than, uns, immediate	MOVNEZ	Move if not equal to zero
XOR	Logical exclusive-OR	BLTZ	Branch if less than zero	**Flow Control**	
NEG	Negate	BNE	Branch if not equal	CALL0	Call, zero register overlap
SLL	Logical shift left	BNEI	Branch if not equal, immediate	CALLX0	Call via register
SLLI	Logical shift left, immediate	BALL	Branch if all bits set	RET	Return from CALL
SRA	Arithmetic shift right	BNALL	Branch if not all bits set	J	Jump, unconditional
SRAI	Arithmetic shift right, immediate	BANY	Branch if any bits set	JX	Jump via register
SRC	Shift right through carry	BNONE	Branch if no bit set	**Miscellaneous**	
SRL	Logical shift left	BBC	Branch if bit clear	RSR	Read special register
SRLI	Logical shift left, immediate	BBCI	Branch if bit clear, immediate	WSR	Write special register
SSA8B	Set shift amount, little-endian	BBS	Branch if bit set	DSYNC	Synchronize load/store
SSA8L	Set shift amount, big-endian	BBSI	Branch if bit set, immediate	ESYNC	Serialize execution
SSAI	Set shift amount, immediate	LOOP	Loop	ISYNC	Synchronize fetch
SSL	Set shift amount for left shift	LOOPGTZ	Loop if greater than zero	RSYNC	Register read, synchronized
SSR	Set shift amount for right shift	LOOPNEZ	Loop if not equal to zero	EXTUI	Extract unsigned immediate

Table 1. All Xtensa processors share a baseline instruction set of 78 instructions, which users can extend using a Verilog-like compiler. The base set includes normal arithmetic and logical operations but no multiply-accumulate, media, or SIMD operations.

Price & Availability

Tensilica's Xtensa microprocessor generator is available now. Licensing fees start at $250,000 for Web-based access to the tools and a manufacturing license. Royalties are negotiable.

For more information, contact Tensilica (Santa Clara, Calif.) at 408.986.8000 or visit *www.tensilica.com.*

There are only a few basic limitations on new instructions: they must be encoded in either 16 or 24 bits; they must execute in a single cycle; and they must access only register-based operands. All new instructions will, by definition, require additional hardware resources. If the designer can describe the necessary hardware function in TIE, the synthesis tools will generate it as necessary.

Within these boundaries, users are free to develop their own special-purpose instructions. Rowen cites common examples of arithmetic minimum, maximum, or sum-of-absolute-differences instructions. Users can also create their own simple SIMD instructions that, for example, sum two eight-bit operands in parallel. This would be implemented with a new adder that has a segmented carry chain, allowing two unrelated addition operations at once.

Predefined Options Provide a la Carte Selection

Tensilica has already defined about two dozen prepackaged options for the basic Xtensa core. Option packages consist of a few specialized instructions and the additional hardware resources needed to support them. Some examples add support for prioritized interrupts, timers, a 32-bit multiply/divide unit, a floating-point coprocessor, a larger register file, debug visibility, or increased code density.

Many of these options are roughly analogous to the T, D, M, and I options for the ARM7 and ARM9 designs. ARC Cores has a similar package of specialized ISA options, and MIPS licensees have MIPS-16 code compression. Motorola's ever-evolving ColdFire product line includes chips with MAC units and other options, although these modifications are controlled by Motorola, not (directly) by its customers.

Synthesis Tools Are Key

The entire Xtensa processor core is synthesized from Verilog code. Rowen points out that the processor can be synthesized using virtually any standard-cell library and memory generator. Tensilica neither recommends nor requires any special libraries; even the register file is implemented as gates, not as a specially hand-packed multiported macro cell.

Such a level of generic, lowest-common-denominator synthesis would normally exact a heavy toll on performance, a situation Tensilica claims to have overcome. The company suggests a practical target frequency of 150–175 MHz in a "generic" 0.25-micron CMOS process. Speeds of up to 250 MHz are possible under less pessimistic process, voltage, and temperature conditions, according to Tensilica.

Tensilica's clock-speed claims don't seem out of line with what other vendors are producing. Microprocessors with optimized 0.25-micron layouts run at about 300 MHz today. ARM claims that its synthesized ARM7TDMI-S core can reach 90% of the clock speed of a hand-optimized design (though at a 2× to 3× penalty in die size), so a 175-MHz Tensilica part doesn't seem unreasonable.

Regarding size, the base implementation should require fewer than 25,000 gates and need less than 1 mm^2 of silicon, according to Tensilica. Power consumption is likewise modest, estimated to be less than 0.5 mW per MHz. All these parameters put Xtensa at the low end of the power, size, and gate-count scales compared with other 32-bit processors. Realistically, at those scales, Xtensa's die size, transistor count, and power consumption are nearly irrelevant in the larger environment of a complete ASIC. Putting multiple processors on a single die becomes practical at these scales.

Software Tools As Important As the Hardware

Hardware engineers may applaud Xtensa's flexibility, but software developers may abandon their medication regimen at the prospect of creating and debugging code for a chip with no fixed instruction set. Tensilica claims this eventuality has been more than covered with a software tool chain that is self-configuring, keeping pace with the CPU changes.

As part of the hardware-synthesis process, Tensilica's tools also modify the supplied GNU compiler, assembler, linker, debugger, profiler, simulator, and libraries to support any new instructions. The C language definition does not change, of course, but the compiler is altered with intrinsics that access new functions. The assembler, debugger, and other tools also will understand user-defined instructions and correctly disassemble and profile them. Users can thus define, synthesize, profile, and tune their processor in a big hardware/software feedback loop, looking for the best combination that achieves the desired optimization.

Third-party operating systems and other software tools are ported to only the basic Xtensa instruction set, ensuring compatibility with any Xtensa implementation. Currently, Tensilica has struck deals with ISI and Wind River for the pSOS and VxWorks real-time operating systems.

User Configurability the Start of a Trend

Tensilica's first public licensee is Zilog, an often overlooked processor supplier. Zilog intends to use Tensilica's processor in its own line of communications-related processors, making tools available to its many far-flung design centers.

Tensilica's business is to license its CPU core and related development tools to semiconductor houses, EDA companies, and large ASIC developers. For the most part, this is not a significantly different model than that adopted by Rambus, MIPS, or other newly minted IP companies.

One difference between Tensilica and MIPS is that Tensilica must maintain software-development tools and keep them in sync with its processor products. Unlike a strict processor-IP company, Tensilica cannot simply update and license its hardware cores; it must also maintain a watch on its software tools or risk diluting the advantage of its extensible processor.

The nearest extant example of Tensilica's strategy is ARC Cores (see MPR 7/8/96, p. 8), which also licenses an extensible microprocessor. Like Tensilica, ARC's CPU is synthesized and user extensible. Also like Tensilica, ARC has developed a portfolio of tested instruction-set extensions, its software tools track changes to the hardware architecture, and users can download interim CPU designs to an FPGA. ARC has a few years' head start over Tensilica and the advantage of revenue from more than 30 existing licensees.

Making technical comparisons between configurable microprocessors is a bit like pushing a rope. Tensilica and the other synthesizable or configurable processors all promise roughly equal performance (in clock speed) in a given process. In each case, real performance differences will be due to customer-designed extensions. And assuming such extensions could be implemented on any underlying architecture, we're back where we started.

Decisions of this sort might be based on design tools, licensing costs, degree of configurability, or ease of use. Processor-IP companies are typically mum on detailing their licensing fees, though most follow the standard practice of charging an upfront fee followed by royalties.

Tensilica is among the first in a new wave. ASIC designers who are comfortable with synthesis and who can benefit from configurability will find Xtensa an interesting alternative to traditional fixed CPU cores. Tensilica's tool chain is complete and well thought out. Xtensa's underlying architecture is sound, but it's only the starting point for custom departures. It is the customers—not the vendor—that make configurable processors valuable. ASIC designers are simply choosing a palette on which to create their perfect chip. M

Embedded News

Continued from page 5

The company has signed on for the Jade and Opal processor cores (see MPR 12/7/98, p. 10), MIPS's forthcoming low-end (32-bit) and midrange (64-bit) designs. Jade, which will be disclosed more fully at Embedded Processor Forum in May, adds general-purpose computing and Windows CE compatibility to the TI product line. MIPS is not the first CPU core that TI has licensed; the company also has licenses from ARC Cores and ARM.

TI hinted at plans for highly integrated devices that could combine its strengths in DSP, mixed-signal, and connectivity (i.e., FireWire and USB) with the new MIPS processors. The first such devices would likely combine the Jade CPU with a DSP from TI's 'C54x family.

MIPS's CPUs and TI's well-supported DSPs should provide a daunting combination especially well suited for portable devices with wireless communication. No schedule was given for product introductions; although ASIC designs could start shortly, products are not likely to ship before the end of this year. —*J.T.*

MoSys To Offer MDRAM Technology

Specialty memory manufacturer MoSys has decided to enter the intellectual-property business by licensing its unusual multibank memory architecture (see MPR 12/25/95, p. 17) to semiconductor vendors. The MoSys memory design, which it calls 1T-SRAM, is an embedded-DRAM cell that can be manufactured in either a pure-logic process or a process designed for embedded DRAM. In either case, MoSys claims a 4× improvement in power at equivalent speed and as much as a 3–9× improvement in area.

The MoSys MDRAM is a true DRAM design in that it uses one transistor per bit. It achieves its power savings because only a small portion of the entire array—one bank—is active at one time. Other banks are either powered down or periodically refreshed. The invisible refresh behavior makes 1T-SRAM look like an SRAM to system logic, while the very small bank size gives it SRAM-like speed.

MoSys (*www.mosys.com*) has not enjoyed enormous success with its SRAM-like DRAMs. Licensing the same technology for embedded-DRAM ASIC usage may be just the new approach the company needs. —*J.T.*

SST—Yet Another 8051 Supplier

Silicon Storage Technology (SST) has entered the crowded market for 8051-compatible microcontrollers with its own line of chips with flash memory. SST's chips, which are part of a planned family with the difficult-to-pronounce name of FlashFlex51, are pin and software compatible with the plethora of 8051 microcontrollers available from various sources. Clock speeds range up to 33 MHz, and supply voltages down to 2.7 V. The chips are available with from 16K to 64K of flash memory, plus an additional 4K block of E^2PROM. SST (*www.ssti.com*) is sampling these chips now, with production scheduled for 2Q99. Prices start at $4.85 in "large quantities."

Flash memory has slowly made inroads into various microprocessors, large and small, over the past 10 years. Customer demand has been consistent, but manufacturing difficulties have kept the price of flash-based microcontrollers much higher than those of their one-time-programmable (OTP) equivalents until recently. Over the next few years, flash-based processors may eventually displace OTP devices for most high-volume applications. —*J.T.* M

Intel Discloses New IA-64 Features

Rotating Registers Reduce Code Expansion; Merced Touted for Big Servers

by Linley Gwennap

In a series of talks at the recent Intel Developers Forum, the company tantalized industry watchers by dribbling out a few more details about its IA-64 instruction set and its first implementation, Merced. In a joint presentation by Intel's John Crawford and Hewlett-Packard's Jerry Huck, the two architects shed additional light on the IA-64 design. They provided further details on the architecture's support for predication and speculation and also described IA-64's branch architecture. A newly disclosed feature, rotating registers, provides an efficient way to unroll loops while minimizing code expansion.

In other talks, Intel disclosed that Merced and its first chip set, the 460GX, will support high-availability features required in large servers. The company asserts that four-processor Merced servers will deliver more performance on the TPC-C benchmark than four-way servers using 1-GHz Alpha 21264 processors or 750-MHz UltraSparc-3 processors, two key Merced rivals that are expected to ship next year. But it has yet to disclose any details about clock speed, bus bandwidth, or other metrics to support this position.

Register Renaming Implemented in Software

One of the key philosophies of IA-64 is the idea of moving complexity from the hardware to the software. Register renaming is one example. Most high-end processors map a small number (8–32) of logical registers onto a larger set of physical registers (up to 80 in the case of the 21264). Because software can access only the logical registers, the hardware must assign mappings and translate accesses using an associative lookup table. This complexity increases die size and often the pipeline depth as well.

IA-64 eliminates this hardware complexity with its large register file (128 integer, 128 floating-point) that is directly accessible by software. Specifying the physical register names in software works well except in the case of tight loops, a common occurrence. In these short code sequences, there may not be enough instructions in the loop to cover the latency of load instructions, resulting in unwanted stalls.

An out-of-order processor reorders instructions to cover the latency of the loads. The reordering naturally overlaps instructions from two or more iterations of the loop until enough instructions are found to overcome the latency (or the hardware runs out of resources). This overlap will cause register conflicts, since each loop iteration references the same registers, but these conflicts are resolved by hardware register renaming.

An IA-64 processor can address the latency problem by unrolling the loop in software. This common compiler technique duplicates the loop instructions, often several times, to generate enough instructions to cover the load latencies. Each duplicate set of instructions, however, must use a different set of registers to avoid collisions. IA-64 has plenty of registers available, but all of these duplicate instructions can create massive code expansion.

Rotating Registers Compact Code

To reduce code expansion, IA-64 uses its rotating registers. With this technique, the upper three-quarters of each register file (integer, FP, and predicates) rotates, leaving the lower registers for global variables. Accesses to these upper registers are offset by the value in the corresponding RRB (rotating register base) register. A special instruction, BR.CTOP, decrements each of the RRBs by one at the end of each loop iteration, allowing the next iteration to use a new set of physical registers. (With proper spacing, several variables can be rotated through the register file at once.)

The rotating predicate registers provide a simple way to handle loop setup (prologue) and termination (epilogue). If the prologue and epilogue instructions are appropriately predicated, and the predicate registers rotated, the prologue instructions are executed only during the initial iteration(s) of the loop, and the epilogue instructions are executed only

MEMCPY LOOP: for (i=0; i<n; i++) {*b++ = *a++}

(a) PA-RISC with hardware reordering

```
; Set up r2=loop count, r10=source addr, r11=destination addr

loop:  LDWM    r1, (r10)       ; Load into r1, inc addr
       STWM    (r11), r1       ; Store from r1, inc addr
       ADDIB,> r2, -1, loop    ; Decr loop count and branch
```

(b) IA-64 with rotating registers

```
; Set up LC=loop count–1, r10=source addr, r11=destination addr
; Clear predicate registers, set p16, set EC=epilogue count

loop:  (p16) LD8      r34 = [r10], 8   ; Load into "r34," inc addr
       (p17) ST8      [r11] = r35, 8   ; Store from previous "r34," inc addr
             BR.CTOP  loop             ; Decr loop count and branch
```

Figure 1. In a simple memory-copy loop, a PA-RISC processor with hardware reordering will cover the latency of the first load by launching subsequent loads, creating multiple versions of "r1" using hardware renaming. Without adding instructions to the loop, an IA-64 processor will accomplish the same effect by rotating its registers; in this case, "r35" refers to the previous iteration of "r34."

during the final iteration(s) of the loop. Some setup is still required to properly initialize the predicates, but this can be done well in advance of beginning the loop, removing this setup from the critical path.

Eschewing an orthogonal register set, HP and Intel added several special registers to implement this process. The 64-bit LC (loop count) register performs its eponymous function. The 6-bit EC (epilogue count) register controls the execution of epilogue instructions. Three RRBs (each 6 or 7 bits) rotate the integer, FP, and predicate registers, as described above. The use of special registers allows the BR.CTOP instruction to specify several operations at once, but in the common case of nested loops, register rotation can be used in only one of the loops.

This method of register renaming allows a single copy of the loop code to be unrolled in hardware rather than software, eliminating most of the code expansion, as Figure 1 shows. Rotating the registers adds some complexity (a few 7-bit registers and adders) to the hardware, but it adds far less than the fully generic renaming hardware in a reordering CPU. The rotating register concept dates back to Cydrome's Cydra-5, one of the original VLIW processors; not coincidentally, its architect, Bob Rau, is now on staff at HP.

By handling epilogue and prologue issues in a simple fashion, IA-64's rotating registers are appropriate even for loops that iterate only a few times. Thus, this technique can be broadly applied. In current processors, loop unrolling is rarely used, except in scientific code, where iteration over long vectors amortizes prologue and epilogue overhead.

Static, Dynamic Prediction Combined

The basic IA-64 branch instruction uses a 21-bit relative offset. Branch targets must be the first instruction in a bundle, allowing branching within ±16M. Aligning the targets simplifies the branch hardware and extends the target range, but it will cause some code expansion by adding an average of one NOP instruction per branch target.

Because conditional branches use the predicate field to specify the condition, they have the same long offset. Indirect branches (including call/return) use a special set of eight BRs (branch registers), instead of the integer registers, to hold target addresses. These special registers are likely to be physically located near the fetch unit, not the ALUs, and thus they could obviate the call/return stack used in most high-end processors to fetch the target of subroutine returns.

The LC register improves branch prediction for loops. Most branch predictors mispredict the final (fall-through) iteration of a loop-closing branch, but IA-64 hardware can easily and accurately predict these branches by looking at the LC register. BR.CLOOP is a degenerate form of BR.CTOP that simply decrements LC and loops if LC ≠ 0. Again, this mechanism can be used for only one loop at a time, as there is only one LC register.

IA-64 branches include at least two bits to give the compiler more control over branch prediction. Like many RISC architectures, IA-64 provides a "hint" as to whether a branch is likely to be taken or not taken. The hardware can use this hint to initialize the dynamic branch predictor. Some branches, however, are easily predicted by software, as they nearly always branch the same way. The second bit indicates that software prediction should be used; the hardware predictor can ignore these branches, freeing entries for more difficult branches. (The hardware may enter static taken branches into its target-address predictor.) This combination of software and hardware prediction should provide more accuracy than today's hardware-only branch predictors.

Like PA-RISC, IA-64 can combine a comparison and a branch in a single cycle. PA-RISC uses a compact compare-and-branch instruction. IA-64 instead combines a predicate-generating CMP instruction and a branch predicated on that result into a single group for parallel execution.

A branch instruction must be at the end of a parallel-instruction group. As a special case, two or more branches can be placed together at the end of a group to form a multiway branch. All the branches can be processed in a single cycle (assuming the hardware has enough resources), as it is a simple matter to check the predicates and determine which, if any, branch should be taken. This construction is useful when several short code blocks have been combined using predication; all the exit cases can be processed at once.

Flexible Design Allows Massive Speculation

The recent disclosures indicate that IA-64's predication and speculation capabilities are more extensive than previously indicated (see MPR 10/27/97, p. 1). Speculation is used to hoist loads above branches, giving the compiler more flexibility to reorganize code. To handle exceptions, each IA-64 register is tagged with an associated NaT (not a thing) bit that is set when a LD.S (speculative load) encounters an exception. The actual exception is deferred until a CHK.S instruction is encountered.

The CHK.S instruction is simply a conditional branch that tests the NaT bit. If NaT is true, it branches to fixup code that reexecutes the load and handles the exception. The IA-64 architects did not take advantage of the target register (which is undefined when NaT is true) to store the load address; thus, the fixup code must have access to any registers needed to recreate the load address.

The NaT mechanism allows instructions that use speculatively loaded data to be hoisted as well. Any computation instruction sets the target register's NaT bit if any source is NaT. This NaT propagation allows entire routines to be executed speculatively, with exceptions later handled by a single CHK.S. Note that the recovery code must also redo any speculative calculations after reloading the correct data. HP's Huck estimates that half of the instructions in a typical program are likely to execute speculatively.

IA-64 also includes a mechanism for hoisting loads above stores. In a traditional architecture, the compiler can

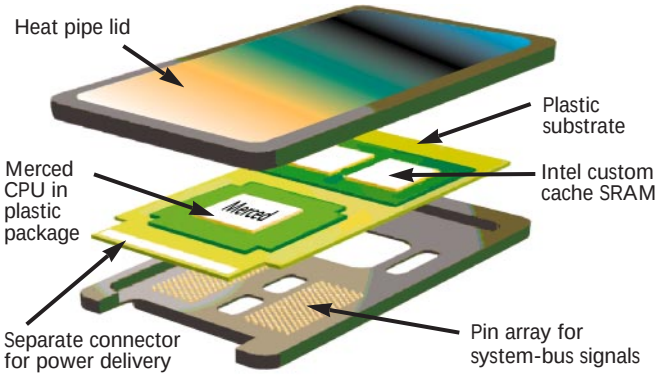


Figure 2. The Merced CPU is combined with up to 4M of SRAM in a module that attaches horizontally to the motherboard.

do this only if it can guarantee that the load and store use different physical addresses. With indirect addressing, however, this pointer disambiguation can be impossible at compile time. Reordering processors handle this task easily, since loads and stores are reordered at runtime, after addresses have been calculated.

To hoist a load above a store, IA-64 uses the LD.A (advanced load) instruction. In addition to performing a normal load, this instruction inserts the load address into the ALAT (advanced load address table). Subsequent store addresses are associatively checked against the ALAT; if a match is found, the offending entry is removed. Before using the data from an LD.A instruction, an LD.C is needed to see if the entry associated with that target register is still in the ALAT. If so, the LD.C is a zero-cycle NOP; if not, the LD.C simply reexecutes the load. A CHK.A instruction allows speculative computation based on the result of an LD.A.

The size of the ALAT is implementation dependent. If an LD.A bumps a "live" address from the ALAT, the LD.C (or CHK.A) will reload the data, causing a performance loss but no error. Similar structures, such as the P6's MOB (memory reorder buffer), are found in reordering processors. The MOB, however, is invisible to software, whereas the ALAT is directly manipulated by the IA-64 compiler.

Predicates Better Than Traditional CMOV

Predicates can be generated using CMP (compare) or TBIT (test bit) instructions. Special versions of CMP combine a comparison with a predicate value to create compound conditions (e.g., A = B or A = C). CMP always generates two predicates, the requested value and its complement, either of which can be stored in any predicate register (p0–p63). The first register, p0, is hardwired "true" and can be used as a target to discard unwanted predicates.

Many programs are limited by control flow; that is, they have many conditional branches that are hard to predict. By combining short routines, the compiler can use predicates to eliminate branches, avoiding costly mispredictions. Traditional architectures can eliminate branches using the simpler CMOV (conditional move) instruction, but this method requires longer code sequences and often results in lower performance than a fully predicated architecture such as IA-64.

Merced Targets Big Systems

In other presentations, Intel left no doubt that the primary target of Merced is high-end commercial servers. The first IA-64 processor and its associated system-logic chip set, the 460GX, will include a host of features to deliver the performance and reliability needed by these expensive systems.

For example, all of the caches and system buses are protected from data loss using ECC or other techniques. Corrupted data is corrected when possible; if not, it can be marked as bad and the affected process terminated without crashing the entire system. For a fully fault-tolerant system, two Merced CPUs can operate in lockstep and crosscheck each other. This feature is not likely to see significant use, as Tandem, the leading vendor of fault-tolerant systems, has chosen Alpha over Merced for its next-generation boxes.

The 460GX supports ECC on the system bus and in the main-memory subsystem and can map out failed DRAMs. It handles up to four Merced processors and can be used as a building block in larger systems, although several Intel customers are developing their own system logic to connect eight or more Merced processors. The 460GX supports hot plugging on up to four PCI buses, each at up to 64 bits and 66 MHz for extra bandwidth. The multichip set can also be used in workstations, as it includes an AGP 4× port.

Sources indicate the 460GX allows at least 16G of SDRAM interleaved four ways, as Direct RDRAM will not provide adequate density in 2000. This bandwidth may be wasted on Merced, however, as Intel says the processor's system bus will carry significantly less than 3.2 Gbytes/s.

As Figure 2 shows, the processor itself is housed in a module containing the CPU chip and custom cache chips. Using both sides of the substrate, the module appears to have room for four SRAMs. Using its 1-Mbyte SRAM, known as CK1 (see MPR 7/13/98, p. 1), Intel should be able to fit up to 4M of full-speed cache on the Merced module.

The module design is functionally similar to the Xeon module but mechanically quite different. The module lies horizontally above the motherboard rather than vertically and uses two sets of connectors. A pin array carries the bus signals and provides better electrical performance than the edge connector in the Slot 2 module; this method should enable faster bus speeds. Power delivery is handled through an edge connector (presumably connected to a wire harness) to efficiently deliver plenty of amps.

The module lid is hollow, forming a heat pipe. This design spreads the intense heat from the CPU across the entire lid, reducing heat density. The system maker must attach a large heat sink to the module to further dissipate the heat. Intel has not disclosed the power of the processor, but the package design clearly implies that it will be high; we estimate the Merced module will dissipate more than 70 W.

Merced Nearing Tapeout

The Merced design team has made much progress since the major slip announced last summer (see MPR 6/22/98, p. 1). The processor is finally nearing tapeout, and the company expects to receive first silicon around the middle of this year. The schedule allows about 12 months to bring up and verify the design before it is ready for system shipments. This schedule seems somewhat aggressive for a high-end processor implementing a new instruction set and aimed exclusively at multiprocessor-capable systems; we would not be surprised if system shipments slip toward late 2000.

Intel has been working hard in an attempt to ensure a smooth bringup process. Given the focus on multiprocessing, the company is already performing MP verification on a presilicon RTL model. Seven operating systems—64-bit Windows NT, SCO UnixWare, Novell Modesto (NetWare), Compaq Tru64 Unix, Silicon Graphics Irix, Sun Solaris, and HP-UX—have booted in MP mode using a system simulator and are expected to be ready for mid-2000 shipments. An eighth port, Linux, is in progress. Because the system designers were aiming for 1999 shipments, the Merced slip has put them ahead of the CPU. Some platforms are already being tested and await only the processor.

Intel has begun working with software vendors to ensure that key applications will be available when Merced ships or shortly after. Leading server applications from Oracle, Informix, SAS, Baan, SAP, and others are already on tap. Technical applications are in the pipe from vendors such as Cadence, Mentor, Synopsys, Softimage, Avid, and Adobe. These vendors have already received software development kits, including the system simulator; several applications are already running on the simulator.

Given that Merced is likely to be superseded by its successor fairly quickly, Intel is trying to build a smooth migration path to McKinley. Although McKinley will use a much faster system bus than Merced's, Intel hopes vendors will be able to reuse much of the system infrastructure. For example, Intel's 870 chip set for McKinley will support legacy memory and I/O from the 460GX. McKinley will also use no more power or board space than Merced, avoiding chassis redesign.

IA-64 Performance Debate Unsettled

The new details of IA-64 highlight its philosophy of moving complexity from the hardware to the compiler. With these new features, an IA-64 compiler can perform most of the code motions handled by hardware in a reordering processor. The compiler can perform these code motions across an arbitrarily large group of instructions, whereas reordering hardware is limited to a window of no more than 80 instructions in today's implementations. Without predication and access to large register files, RISC compilers cannot perform the same optimizations and must rely on the hardware.

Initial criticisms of IA-64 focused on its emphasis on static instruction scheduling, which ignores dynamic information available to the hardware at runtime. Some of the newly disclosed features address these issues and show how IA-64 combines static and dynamic scheduling. The ALAT, for example, allows loads to pass stores in a manner impossible in a purely static machine. IA-64's branch prediction is another example of a combination of static and dynamic methods.

The tradeoff is that dynamic features add hardware complexity. Initially, it appeared that an IA-64 design might be more compact than an out-of-order processor by eliminating the instruction-reordering and register-renaming logic. IA-64 processors, however, still require features such as dynamic branch prediction, rotating registers, and the ALAT, which consume die area. Although these particular features may be smaller than their RISC counterparts, when combined with other IA-64 features such as predication and the large register files, they are likely to limit any die-size advantage IA-64 might have over RISC.

Code size remains a concern. The rotating registers avoid the massive code bloat of loop unrolling, but speculation is another issue. Every speculative instruction must be duplicated in a fixup routine, although these infrequently executed routines aren't likely to do much, other than take up disk space. More critical are the aligned branch targets and the 41-bit instructions themselves, which are 33% larger than RISC instructions. An increase in code size reduces the effectiveness of the instruction cache and requires more bandwidth from the system bus and from main memory.

The large register file improves performance by reducing data-cache accesses, but it creates a problem on context switches. Saving state requires storing 128 integer registers (64 bits each), 128 FP registers (at least 80 bits each), 63 predicate bits, 256 NaT bits, 8 branch registers, 3 RRBs, the LC, EC, and Intel knows what else.

The upside of all these features is real, but it remains to be quantified. We expect the net performance advantage of IA-64 over RISC will be around 20–30%, significant but not impossible for competitors to overcome. If Intel delivers strong implementations, they should match or exceed the performance of the fastest competitive chips.

Merced may not be the best implementation of IA-64, but it should be very competitive, enough to establish the architecture in high-end servers and some workstations. The design is progressing well, and we expect first systems to appear in 2H00. Support from both system and software vendors is strong and unwavering. As we have seen with x86, this support, more than any technical merits or demerits, determines the fate of a new microprocessor. M

For More Information

For more information on IA-64, visit Intel's Web site at *http://developer.intel.com/design/processor/future/ia64.htm.*

CPU Business to Rebound in 1999

Improving Industry, New Products, and Y2K Upgrades Point to High Growth

by Mel Thomsen

The 32-bit microprocessor business should be much better this year than last. We expect a revenue growth of approximately 41%, from $24.3 billion in 1998 to $32.1 billion this year. Unit shipments should grow by 45%, from 217 million units in 1998 to 315 million in 1999, as Figure 1 shows. Although this is a high growth rate, it is due to low unit shipments in 1998. Average selling prices will continue to decline along the historical trend, from $112 in 4Q98 to about $102 by 4Q99, as Figure 2 shows. This forecast is for worldwide sales of all 32-bit microprocessors, including both embedded and nonembedded processors.

Semiconductor Industry Recovers in 1999

The worldwide recession that has plagued the semiconductor industry since 1996 finally reached bottom during July and August last year. The global semiconductor industry is now on a gradual recovery that will accelerate in the second half of this year, as seasonal strength adds to the fundamental recovery.

A look at other global factors shows that the economic crisis in Asia that surprised the industry last year is not over but has stabilized. A year ago, the sudden collapse of the Asian economy seemed so bleak as to engender serious talk about triggering a global depression. That doomsday forecast is less pervasive now, but the fear still lurks in the shadows.

Although PC sales won't be booming in Asia this year, any microprocessor manufacturer or PC supplier with a reasonably conservative sales forecast (meaning no growth or very low growth) should be safe from downside surprises from sales in that region. At the same time, European sales are stable and probably will remain so, as the currency unification seems to be giving a psychological boost to the economy throughout that region. Finally, PC sales in the United States should resume their historical rates. PC sales growth this year should be slightly better than last year, which suffered from a first-half inventory glut.

Seasonal patterns will be typical, although the first quarter of this year will be slightly stronger than historical trends suggest. There are two underlying reasons for this prediction. First, despite the stronger-than-expected surge in 4Q98, there is not as much inventory in the PC channel. Second, Intel's new Pentium III (see cover story) will give a slight boost to 1Q99 CPU sales as PC makers purchased early production units before the formal introduction of Katmai-based systems in February.

ASPs Trending Downward

Average selling prices (ASPs) will continue to decline as in the past—perhaps even faster. This is driven by renewed competition at the low end for PC processors, as Intel has become more aggressive in Celeron pricing than it was last year (see MPR 1/25/99, p.18). If there is any downside to our forecast, it could come from even lower ASPs caused by a price war at the low end of the market.

Also contributing to declining ASPs for 32-bit processors is the changing mix between PC processors and the low-priced embedded processors that now represent approximately 65% of all 32-bit units. The embedded processors typically sell in the $5–$70 range, while most PC processors are priced in the $60–$400 range.

One upward force on ASPs is the growth in shipments of Xeon-class server processors. Another upward factor is the shift to units with on-chip L2 cache, as MPU makers take more system dollars from the SRAM suppliers. Neither of

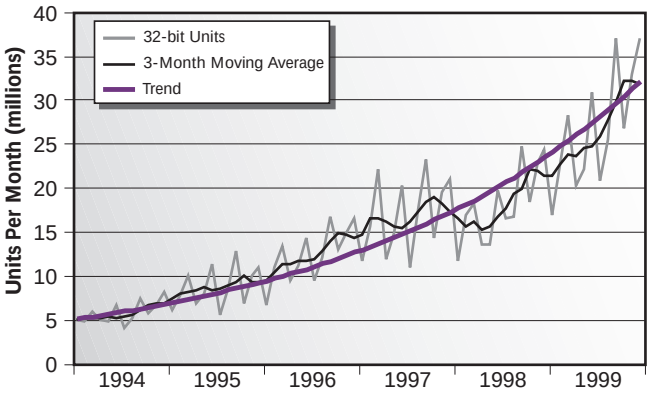


Figure 1. CPU unit shipments should remain below the trend line for the first half of 1999, giving a unit growth of 45% this year.

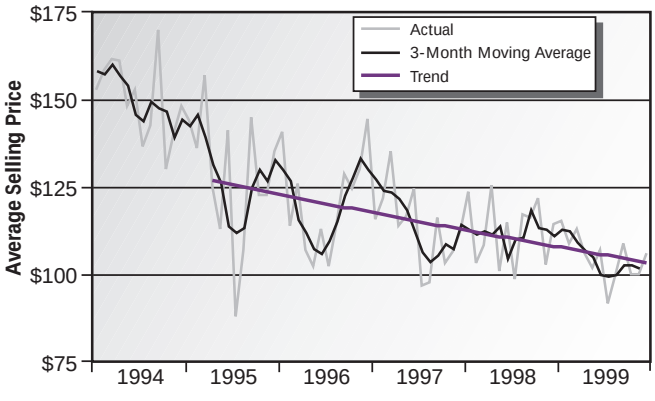


Figure 2. CPU ASPs will decline throughout the first half of the year before slightly recovering in the fourth quarter.

these trends, however, is strong enough to counterbalance the downward pressures.

End-Use Demand Remains Strong

The bright spot in the outlook is that end-use demand remains strong for PCs and products such as laser printers, handheld PCs, DVD players, automotive-engine controllers, and cellular phones that use 32-bit microprocessors. Sales of video games, while still large, fell off in 1998 and may decline further this year because the technology is old. New video-game consoles will not enter the market early enough this year to have a significant impact. Sales of digital cameras are growing rapidly, but their volume is still too small to have a significant impact on CPU demand this year.

Y2K Creates Uncertainty

The Y2K problem creates a large uncertainty in the sales growth of PCs and microprocessors this year. There are two scenarios. The negative scenario suggests that most IT budgets will be reserved for solving the software aspects of the Y2K problem, with little left for PCs, networks, or communications hardware—all of which drive semiconductor consumption.

While those IT managers are consumed with fixing the software aspect of the amorphous problem, there is little mental bandwidth remaining to evaluate new hardware, let alone purchase any. A corollary to this scenario suggests that by midyear, IT departments may give up trying to fix software and decide to replace equipment in desperation, strengthening demand in the second half. Although there may be some cases where that scenario occurs, we don't give it much credence in our outlook.

The other, more positive, scenario is that IT managers will decide that the best way to solve the Y2K problem is to replace old Y2K-noncompliant PCs. Although problems with PCs are a minor facet of the Y2K issue, there are some older, noncompliant systems in the field. Given the cost of upgrading, testing, and requalifying those systems, many IT managers are opting to replace them. They may do so early so they can exit the year with a stable environment.

Although large enterprises have been waiting (hoping, actually) for Windows 2000, many have given up any hope of deploying it this year. If we assume an evaluation period of three to six months, a shipping release of Windows 2000 would have to be ready now to be in a first-half upgrade cycle. And if there is a stable equipment environment during the second half, we won't see a significant push from Windows 2000 upgrades, even if it is released before June. Buyers are abandoning the idea of installing Windows 2000 in 1999, making it an easy choice to upgrade noncompliant PCs now.

Although it is too early to tell if either scenario will be strong enough to significantly alter the business pattern this year, there are signs that the early PC-upgrade scenario is happening now.

1998 Is Behind Us—Thankfully

This forecast is much more optimistic than the year just ended would indicate. In 1998, worldwide 32-bit microprocessor revenue grew a dismal 4.8% from $23.2 billion in 1997, making it the worst year in the past decade. However, unit shipments in 4Q98 rebounded quite strongly. During the same year, shipments grew a mere 7.1% from 203 million units.

As 1998 began, the industry consensus was for approximately 15% revenue growth. But the world was stunned into a different reality when, in the first quarter last year, Intel announced that its revenue would fall 10% short of expectations and that it would allow its worldwide workforce to decline by about 3,000. Other companies followed with their own bad news. Although the microprocessor business recovered more strongly in the fourth quarter than we anticipated, that surge came too late to offset the dismal first half.

Several factors contributed to 1998's poor showing for this premiere segment of the chip industry. The first was the extremely robust shipments in the last quarter of 1997, as PC makers built excessive inventory. That action may have been motivated by a desire to stockpile Socket 7 microprocessors to maintain that configuration in the face of a shift toward the more expensive Pentium II module. Or perhaps it was merely overzealous forecasting by PC suppliers. Whatever the reason, it caused a severe inventory excess in the PC channel during most of the first half of 1998.

Second, PC OEMs made a rapid shift to a build-to-order business style, causing a sharp slowdown in microprocessor shipments as they depleted existing inventory. Fortunately, the build-to-order transition is nearly complete, and this one-time event will not be a factor this year.

Adding to the situation was the Asian Flu—the rapidly worsening economic situation throughout the countries of Asia—which caused a slowdown in PC sales in those previously fast-growing regions. Finally, computer retailers, knowing that Windows 98 and 100-MHz bus systems were coming by mid-1998, were not willing to restock their shelves with computers that they feared would rapidly become obsolete.

Sunny Outlook for '99

For 1999, we see more positive signs than negative, with Y2K effects, real or imagined, the big question mark. This year will usher in major new products, such as AMD's K6-III (see page 22) and K7 (see MPR 10/26/98, p.1), and Intel's Pentium III (see cover story).

If there are any surprises this year, and there always are, they are likely to be upside surprises. The high end of our forecast model showed as much as 55% unit growth for 1999, but we need to see stronger activity before making that call. M

Mel Thomsen, Director of Market Analysis at MicroDesign Resources, has been tracking the semiconductor market for more than 10 years. Mel will present MDR's Semiconductor Industry Forecast at the March 18 MDR dinner meeting. For more information, visit www.MDRonline.com/events/sve/.

AMD Gets the IIIrd Degree

K6 III Positioned Against Pentium III, But Will PC Makers Buy It?

by Linley Gwennap

Seeking to escape the profitless pit of Celeron competition, AMD has begun shipping the long-awaited K6 III. With both performance and nomenclature, the company shows that its part is competitive not only with Intel's Pentium II but with the just-released Pentium III (see cover story). The new K6 is a convincing argument that consumers, and even businesses, should consider an AMD processor instead of an Intel chip. But PC makers are a staid lot, and it will take time for AMD to convert its technical prowess into design wins in more expensive PCs.

With the new name, the company has adopted the Roman numerals that are *sine qua non* for Super Bowls and PC processors, what with Pentium II and III, Cyrix's MII (that's M2, not 1002), and Rise's forthcoming mP6 II. (Any day now, IDT will surely release a WinChip II.) The K6's III is both a numerical increment from the K6-2 and an unsubtle indication that the processor is ready to rumble with Pentium III, Intel's top-of-the-line PC processor.

The AMD chip—previously known as the K6+ 3D, the K6-3, and Sharptooth—has been in the works for some time; initial details of the part were disclosed at Microprocessor Forum 1997, and AMD has had working silicon since last summer. Its introduction one week before Pentium III is no coincidence; the company had been waiting until it needed to deploy the chip, as its larger die size will reduce AMD's processor output. But with the K6-2 under fire from Celeron, and with a juicy target like Pentium III available, the time is right for AMD to advance its technology.

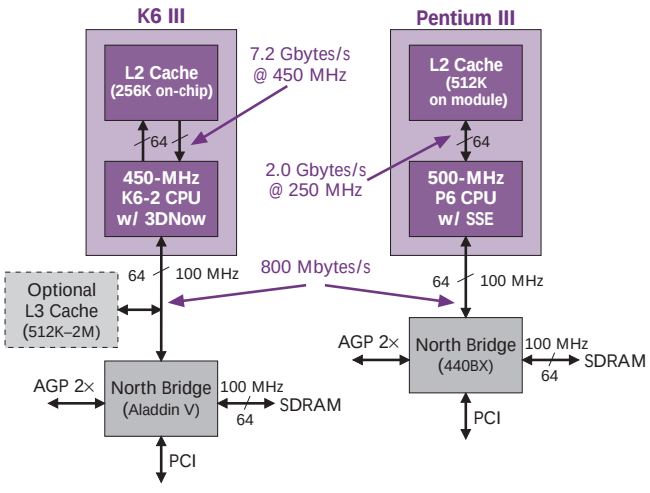


Figure 1. The K6 III matches the dual-bus structure of Pentium III; in fact, its dual-ported full-speed cache delivers nearly four times the bandwidth of Intel's single-port half-speed cache.

Integrated Cache Breaks Socket 7 Bottleneck

As previously disclosed (see MPR 10/27/97, p. 19), the K6 III combines the CPU core from the K6-2 with 256K of on-die level-two (L2) cache. This design breaks a major constraint of the Socket 7 architecture, giving the K6 III the same dual-bus architecture used in Intel's P6 processors. As Figure 1 shows, a K6 III system has a nearly the same bus structure as a typical Pentium III system. The only external difference is that the K6 III supports an optional level-three (L3) cache, which is impractical in a Pentium III system.

By placing the L2 cache on the CPU chip, AMD can run it at the full CPU speed, delivering at least twice the bandwidth of Intel's half-speed external cache. Taking further advantage of the integration, AMD made its L2 cache dual ported, allowing it to process one read and one write per cycle. This design further increases the peak cache bandwidth, delivering 7.2 Gbytes/s in a 450-MHz K6 III. This is four times the bandwidth achieved in a Pentium III at the same clock speed.

We expect Intel to integrate an L2 cache in the 0.18-micron Pentium III, code-named Coppermine, in 2H99. This cache, like AMD's, should run at the full CPU speed. If Intel uses the same cache design as in its current Mendocino (see MPR 8/24/98, p. 1) and Dixon chips, however, it will not be dual ported. This situation will prevent Coppermine from matching the K6 III's L2-cache bandwidth.

Like Pentium III's, the K6 III's L2 cache is four-way set-associative. A read and a write must access different sets to execute in the same cycle. The cache is nonblocking; up to two cache misses can be in progress without stalling the CPU on a cache hit. The L2 cache has a latency of three cycles; at 450 MHz, this is 4.5× faster than an external L2 cache on the K6-2. AMD did not implement 256-bit on-chip buses, so it still takes four cycles to transfer an entire cache line.

The K6 III continues to use the "Super" Socket 7 bus to access main memory and I/O, maintaining compatibility with K6-2 systems. Like Pentium III's Slot 1 bus, this bus runs at 100 MHz and delivers a peak bandwidth of 800 Mbytes/s. One drawback of the Socket 7 bus is that only one transaction can be pipelined. This limitation can bog down performance when many lines are being flushed to DRAM.

The L3 cache helps reduce the impact of this problem. AMD's measurements indicate that a 512K L3 cache improves the K6 III's application performance over a system with no L3 cache by about 4%. This sizable boost is surprising, given an L3 cache that is only twice as large the L2 cache. But the L3 acts as a cast-out buffer, holding lines that are flushed from the L2 cache without stalling the bus during a long main-memory write.

Performance Comes at a Cost

As Figure 2 shows, the integrated L2 cache consumes a third of the K6 III's 118-mm^2 die. According to the MDR Cost Model, the new chip costs about $45 to build, $10 more than the K6-2. Worse yet, the increase in die size reduces the net die per wafer by 40%, according to our model. The reduction would be greater had AMD not used redundant rows to map out some defects in the cache array. The lower yield means that every wafer AMD devotes to K6 III produces just over half as many good chips as a K6-2 wafer.

This explains the vendor's hesitation in announcing the new chip. During 1998, AMD's shipments were limited by the number of chips it could produce. By focusing on the K6-2, AMD boosted its output from 1.5 million in 1Q98 to more than 5 million in 4Q98. It could not have achieved such rapid growth had it shifted production to the larger K6 III.

Even in 1999, AMD will probably phase in the new part slowly to prevent a decline in unit shipments. Ironically, Intel's newly aggressive Celeron tactics (see MPR 1/25/99, p. 18) will help AMD ramp the K6 III more quickly. If Intel succeeds in blocking AMD's unit-share gains, AMD can devote more of its growing wafer capacity to the larger die. If AMD continues to rapidly gain design wins, it will have to rely more heavily on the K6-2 to meet demand.

The increase in manufacturing cost is less significant. If all goes well, the K6 III's average selling price (ASP) will be much higher than that of the K6-2, far outweighing the $10 increase in cost. In fact, the company would be better off selling fewer processors if such action results in greater profits.

Even at $45, the K6 III costs about $10 less to build than Intel's Celeron (Mendocino) and $25 less than the initial Pentium III (Katmai) modules. These cost advantages are due to two factors. First, the K6 core is smaller than the P6 core by about a third. Intel argues that the P6 core is more powerful, but most benchmarks show little performance difference between the two.

Second, AMD's L2-cache design is also compact; the K6 III's 256K cache is a third smaller than the 256K cache on Intel's Dixon (see MPR 1/25/99, p. 20). The metal pitches of AMD's CS-44 process (see MPR 9/16/96, p. 11) are similar to those of Intel's P856.5 process, but CS-44 provides a local interconnect layer not found in P856.5. This extra feature is valuable for reducing SRAM cell size.

In fact, AMD's die-size advantage is so significant that the 0.25-micron K6 III is likely to be smaller than Intel's 0.18-micron Pentium III (Coppermine), which we expect will also include 256K of on-chip cache. In previous processor generations (such as the 386 and 486), AMD's parts were always larger than Intel's, because the smaller company couldn't devote as many engineers to circuit design and IC process development. Today, the shoe is on the other foot, as AMD is trying to kick Intel's booty with a more compact CPU and a better manufacturing process.

It seems odd that AMD is introducing on-die cache at the top of its product line while Intel is using on-die cache mainly in its low-end Celeron parts. Intel, however, had already made the leap of including L2 cache as part of the "processor." Because the original Pentium II featured a discrete L2 cache on a module, moving that cache onto the die provided a cost reduction. In AMD's case, the L2 cache is a performance enhancement that adds cost, so the company is using it as a premium feature. With its Coppermine Pentium III, Intel will also introduce on-chip cache at the high end.

Price & Availability

The AMD K6 III is available now at 400 MHz; the 450-MHz version is sampling, with volume shipments set for March. In 1,000-unit quantities, the two parts list for $284 and $476, respectively. For more information, access *www.amd.com/K6.*

K6 III's Performance Matches Pentium III's

According to AMD's testing, the K6 III is more than a match for Pentium III at the same clock speed. On Winstone 99, which represents typical PC productivity applications, a 450-MHz K6 III rated 5% faster than a 450-MHz Pentium II with Windows 98. (Intel's tests show a Pentium II–450 and a Pentium III–450 are equivalent on Winstone 99.) This difference seems small, but AMD claims the K6 III is actually faster than a Pentium III-500 as well, as Figure 3 shows.

Previous versions of the K6 fared less well on Windows NT, which generates more memory traffic and thus encounters the Socket 7 bottleneck. With its dual-bus architecture, the K6 III does well on NT, rating 4% better than Pentium II at the same clock speed. In this case, however, the new 500-MHz Pentium III barely tops the K6 III-450.

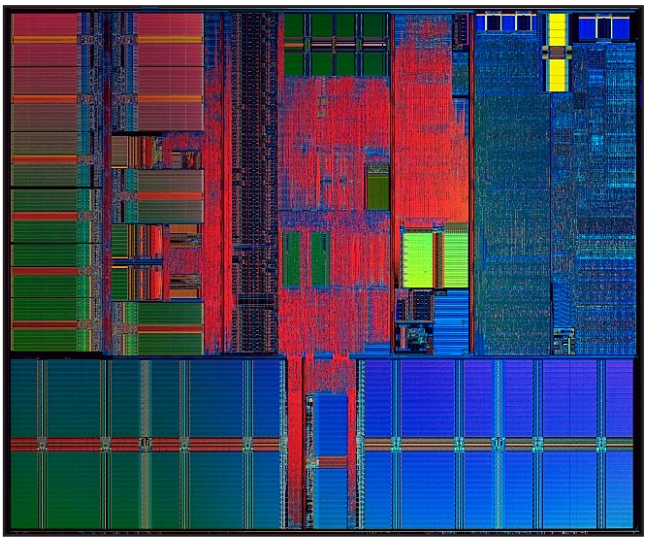


Figure 2. The K6 III contains 21.3 million transistors and measures 9.8 × 12.0 mm in AMD's 0.25-micron five-layer-metal process. The 256K L2 cache (at bottom) consumes 34% of the die area and 56% of the transistors.

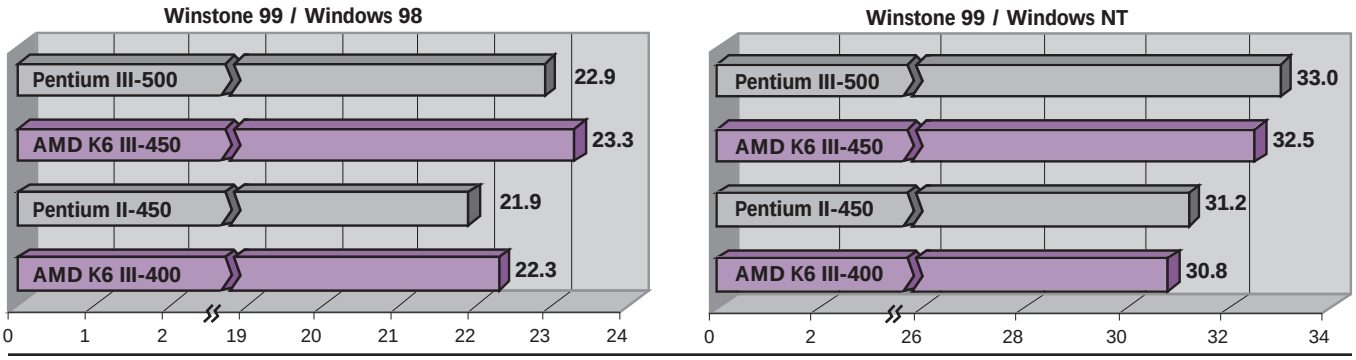


Figure 3. When running Ziff-Davis's Winstone 99 benchmark, the K6 III-450 is faster than a Pentium III-500 on Windows 98 and faster than a Pentium II-450 on Windows NT. Both systems use a high-end Maxtor DiamondMax 2500 Plus IDE disk, an STB Lightspeed 3300 16M AGP graphics card, and 64M of SDRAM (128M for NT). The K6 III uses the ALi Aladdin V chip set with 512K of L3 cache on an Asus P5A motherboard, while the Pentium II/III uses the 440BX chip set and no L3 on an Asus P2B motherboard. (Source: AMD)

At this point, it's too early to say how the two chips will compare on 3D applications. AMD claims that 3DNow delivers 20% better performance than Pentium II on applications that have been enhanced for 3DNow. Although AMD touts an impressive list of 3DNow applications, many are only partially optimized and thus do not currently see the full performance benefit.

Pentium III contains Intel's new SSE instructions, which also boost 3D performance for applications that take advantage of them. Intel has not published 3D application benchmarks for Pentium III, but we expect SSE to deliver a boost of between 15% and 30%, similar to 3DNow's. Both new processors have a peak execution rate of 4 FLOPS per cycle, so neither has an inherent advantage on optimized 3D code. Pentium III has an edge on unoptimized 3D applications, as it delivers about 25% better performance on standard x86 floating-point code.

The performance data in Figure 3 was measured with an expensive cached hard drive typical of a high-end configuration. AMD would not provide results for a more mainstream configuration. On the other hand, AMD is giving up a bit of performance by benchmarking its chip with a 512K L3 cache. The Aladdin V chip set supports up to 2M of L3 cache, which would boost performance by another 4%, according to AMD. Conversely, the K6 III can be used with no L3 cache, with only 4% performance loss.

The K6 III is likely to be used without an L3 cache in notebook systems, where the extra performance isn't worth the power consumed by the cache. AMD has not yet announced notebook versions of the K6 III, but the new chip should fit into the mobile power envelope. By definition, the K6 III CPU core dissipates the same power as the K6-2; the cache adds 2 W, about what is used by the external L2 cache in a K6-2 system.

AMD Assumes the Position

With a $476 list price, the K6 III-450 sets a new high-water mark for AMD. More important, this price puts the chip in direct competition with Pentium III. In fact, the K6 III-450 is only $20 less expensive than the Pentium III-450 and the same price as the Pentium II-450. Similarly, the K6 III-400, at $284, has the same price as the Pentium II-400. This seems untenable, as AMD has always had to offer a 20% or greater discount for the same performance as Intel's.

AMD's plan is to position the K6 III against Intel's premium products while the K6-2 continues to get down and dirty with Celeron. The K6-2 will provide the bulk of AMD's shipments for some time, as it will continue to thrive in the sub-$1,000 PC market, where only MHz matters. With Intel keeping even the fastest Celerons priced below $150, AMD needs to target Pentium II and Pentium III to boost its prices. The K6 III's role is to take on these chips, appealing to more sophisticated PC buyers who have bigger budgets.

As Figure 3 shows, the K6 III has the performance to compete well with Pentium II and even Pentium III. AMD hopes that the new chip's stronger performance on Windows NT will help it break into the business PC market, where AMD has had little success to date. Because businesses buy two-thirds of all PCs, this market is crucial to AMD's hopes of securing a 30% market share.

Although AMD assumes this position will hold, the flaw is that in the PC market performance is becoming less relevant than branding. On most applications, Celeron, a Pentium II, and a Pentium III at the same clock speed all have virtually the same performance, yet Intel positions these parts very differently. Because of this performance overlap among its own lines, Intel could easily show that a 400-MHz Celeron delivers nearly the same Winstone 99 performance as a 400-MHz K6 III.

Intel must be careful in making such an argument, however, as it would undercut its own fragile positioning of Celeron and Pentium II/III. AMD would like to duplicate this positioning with the K6-2 and K6 III, respectively. We expect the K6 III will succeed in pushing AMD into new PC segments, but volumes will ramp slowly, as building momentum in high-end markets takes time. On its technical merits alone, the K6 III clearly deserves to be more than the core of a sub-$1,000 PC. M

LITERATURE WATCH

AUDIO/VIDEO

Displays for electronic imaging. Electronic displays are seemingly pervasive, but their suitability for digital imaging depends on both underlying technology and end-user application, as this overview of displays indicates. Paul Alt, IBM; *IEEE Micro*, 11–12/98, p. 42, 11 pp.

BUSES

Multiprocessor architecture drives high-bandwidth real-time applications. Traquair's new communications architecture, Heart, focuses on getting data where it needs to go quickly, simply, and with minimal delay. Stephen Bradshaw, Traquair Data Systems; *RTC*, 1/99, p. 71, 3 pp.

NGIO and RACE: benefits of switched fabric interconnects. NGIO provides a scalable way to connect desktop processors, allowing both high bandwidth and scalability. Gérard Vichniac, Mercury Computer Systems; *RTC*, 1/99, p. 61, 3 pp.

Switches, switched fabric, bridges drive new MP architectures. New technologies and mechanisms continue to shift high-end multiprocessing from today's shared-memory model to a distributed model. Ray Weiss, *RTC*, 1/99, p. 51, 5 pp.

DEVELOPMENT TOOLS

Effective C++ memory allocation. Using features of C++, the author presents a framework for resource allocation that is temporally deterministic and provides for callback, memory pools, and deadlock prevention. Aaron Dailey, Chaparral Technologies; *Embedded Systems Programming*, 1/99, p. 44, 6 pp.

C compilers and development tools simplify DSP assembly-language programming. With multiple execution units operating in parallel, DSPs demand a level of programming dexterity that is difficult to achieve. Thanks to C compilers and development tools, your task is getting simpler. Manju Nath, *EDN*, 1/21/99, p. 103, 5 pp.

UML statecharts. Here is an examination of statechart development using the Unified Modeling Language. The author describes the event metamodel in UML and some of the more interesting features of statecharts, including nested states and orthogonal regions. Bruce Douglass, *Embedded Systems Programming*, 1/99, p. 22, 11 pp.

MISCELLANEOUS

Standards-based embedded computer industry headed for banner year. The overall market is expected to climb more than 23% for the year, topping last year's growth of about 18%—provided there are no major hiccups in the economy. Warren Andrews, *RTC*, 1/99, p. 39, 4 pp.

Survey: foundries. Foundries stand at the heart of the semiconductor industry's paradigm shift away from vertically integrated companies. Richard Quinnell, *Silicon Strategies*, 1/99, p. 34, 5 pp.

Taking Moore's law into the next century. "Cooperation" is not a word associated with the competitive world of semiconductor manufacturing. Yet the Semiconductor Research Corporation not only gets all the major players to the table, it also helps set the course for the chip industry's future. Scott Hamilton, *Computer*, 1/99, p. 43, 6 pp.

PERIPHERAL CHIPS

RTCs and the Y2K bug. The system's real-time clock (RTC), or lack thereof, plays a leading role in ensuring that the computer will perform correctly in the next century. Jim Lott, Dallas Semiconductor; *RTC*, 1/99, p. 23, 3 pp.

The broad sweep of integrated microsystems. Micromachining mechanical, chemical, and optical components onto the same wafers as electronic circuits produces powerful systems that can understand and influence their environments. S. Tom Picraux and Paul McWhorter, Sandia National Laboratories; *IEEE Spectrum*, 12/98, p. 24, 10 pp.

PROCESSORS

The pressure is on. As applications become more demanding, computer-systems research must not only redefine traditional roles but also unite diverse disciplines in a common goal: to make quantum leaps toward next-generation systems. Krishna Kavi et al., *Computer*, 1/99, p. 30, 4 pp.

A chip called Katmai. Intel's Katmai New Instructions (KNI) promise technology that can rip through 3D graphics twice as fast. Unfortunately, Intel's first KNI-infused processor will take only partial advantage of KNI. Tom Halfhill, *Maximum PC*, 2/99, p. 68, 2 pp.

PROGRAMMABLE LOGIC

Generating reliable embedded processors. This approach to designing fault-tolerant embedded systems—using PLDs to duplicate application-specific hardware—significantly reduces the costs of classical fault-tolerance techniques. Matthias Pflanz and Heinrich Vierhaus, Brandenburg Technical University; *IEEE Micro*, 9–10/98, p. 33, 9 pp.

SYSTEM DESIGN

Color processing in digital cameras. In seconds, a digital camera performs full-color rendering that includes color filter array interpolation, color calibration, antialiasing, infrared rejection, and white-point correction. This article describes the design decisions that make this processing possible. Jim Adams et al., Eastman Kodak; *IEEE Micro*, 11–12/98, p. 54, 10 pp.

Apple Macintosh's energy consumption. Much of a portable computer's utility depends on how long it can run off the battery. We measure Apple Macintosh's current power consumption (and how much of that power goes to each system component) using built-in measuring tools. Jacob Lorch and Alan Smith, UC Berkeley; *IEEE Micro*, 11–12/98, p. 54, 10 pp.

PATENT WATCH

by Rich Belgard, Contributing Editor

The following U.S. patents related to microprocessors were issued recently. Please send e-mail to belgard@umunhum.stanford.edu *with comments or questions.*

5,784,585

Computer system for executing instruction stream containing mixed compressed and uncompressed instructions by automatically detecting and expanding compressed instructions
Filed: January 7, 1997 Issued: July 21, 1998
Assignee: Motorola Claims: 4
Inventor: Paul W. Derman
Disclosed is an instruction decoder that may decompress compressed instructions fetched from memory. A condition-selecting code, normally controlling execution of a conditional instruction, is used to control the decompressor. The code "never," which would normally cause the instruction to be used as a "NOP," is used to indicate that the decoder should decompress the instruction. Otherwise, the decoder does not decompress the instruction.

5,784,584

High performance microprocessor using instructions that operate within instruction groups
Filed: June 7, 1995 Issued: July 21, 1998
Assignee: Patriot Scientific Claims: 29
Inventors: Charles H. Moore et al.
A microprocessor or system that employs instruction-block-relative operand and branch addressing. Blocks of multiple instructions are fetched. A "SKIP" instruction in the current block can cause the processor to branch within the next block of instructions. In a similar fashion, immediate-operand references can refer to offsets relative to blocks of instructions.

5,781,758

Software emulation system with reduced memory requirements
Filed: March 23, 1995 Issued: July 14, 1998
Assignee: Apple Claims: 13
Inventor: John E. Morley
An instruction-emulation system generates instruction semantic routines on demand during emulation, rather than statically storing all routines. A static portion of the emulator comprises only one copy of each different type of semantic routine. Dispatch-table entries for other emulated instructions of the same type contain pointers to a semantic-routine generator for instructions that have the same number of operands. On demand, the semantic-routine generator makes a copy of the static routine, but it substitutes the appropriate operands in place of those in the statically stored routine, generating a new routine.

5,781,753

Semi-autonomous RISC pipelines for overlapped execution of RISC-like instructions within the multiple superscalar execution units of a processor having distributed pipeline control for speculative and out-of-order execution of complex instructions
Filed: March 13, 1995 Issued: July 14, 1998
Assignee: AMD Claims: 61
Inventors: Harold L. McFarland et al.
A pipeline-control system for implementing a virtual architecture having a complex instruction set is distributed over RISC-like semiautonomous function units in a processor. Decoder logic fetches instructions of the target architecture and translates them into simpler RISC-like operations. These operations are issued to the function units. Operations are executed by the units in a manner that is generally independent of the other units, but the units do not irrevocably change the state of the machine. The decoder logic retires normally terminated operations in order.

5,781,750

Dual-instruction-set architecture CPU with hidden software emulation mode
Filed: January 11, 1994 Issued: July 14, 1998
Assignee: Exponential Technology Claims: 20
Inventors: James S. Blomgren et al.
A central processing unit for processing instructions from two separate instruction sets. The CPU has separate instruction decoders for each of the instruction sets but uses common execution resources to execute either of the instruction sets.

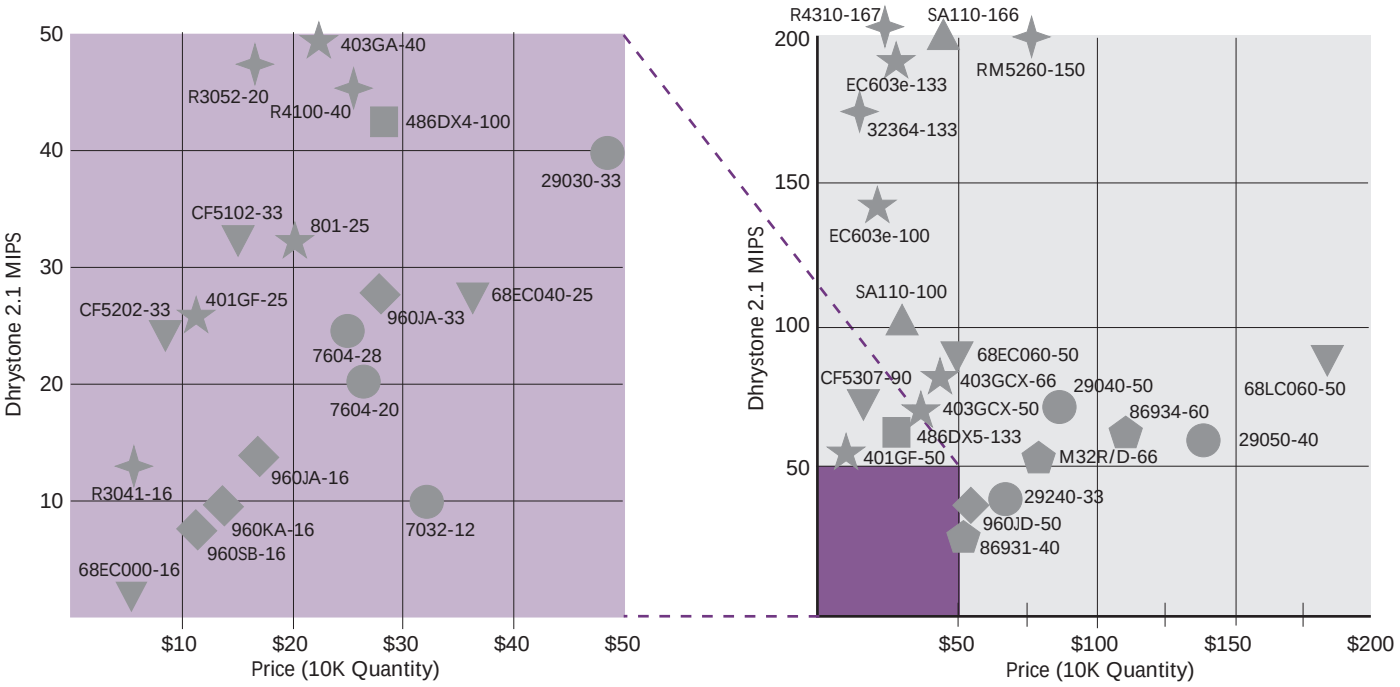
5,778,219

Method and system for propagating exception status in data registers and for detecting exceptions from speculative operations with non-speculative operations
Filed: February 7, 1996 Issued: July 7, 1998
Assignee: HP Claims: 20
Inventors: Frederic C. Amerson et al.
A method for speculative execution includes designating operations as speculative or nonspeculative. Speculative exceptions are deferred; nonspeculative exceptions are reported immediately. Deferred exceptions are detected and reported using a check operation, either incorporated into a nonspeculative operation or inserted as a separate check operation.

OTHER ISSUED PATENTS

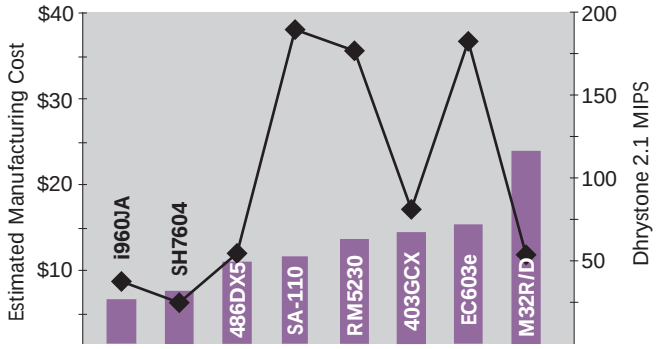
5,781,457 *Merge/mask, rotate/shift, and Boolean operations from two instruction sets executed in a vectored mux on a dual-ALU*
5,778,247 *Multi-pipeline microprocessor with data precision mode indicator* M

CHART WATCH: EMBEDDED PROCESSORS



This Chart Watch covers 32-bit embedded microprocessors, excluding low-power processors. The table below and the chart at the upper right show the performance and price of a number of embedded CPUs; the chart above is an inset for the lowest-priced products.

The chart on the right compares integer performance (diamonds) and estimated manufacturing costs (bars) for a number of embedded microprocessors according to the MDR Cost Model.



	NEC VR5464	IDT 32364	National 486SXL	AMD 486DX5	Intel 960JT	Motorola CF5307	Motorola EC603e	IBM 403GCX	AMD 29040
Architecture	MIPS	MIPS	x86	x86	i960	ColdFire	PowerPC	PowerPC	29K
Clock rate	250 MHz	133 MHz	25 MHz	133 MHz	100 MHz	90 MHz	200 MHz	66 MHz	50 MHz
I/D cache	**32K/32K**	8K/2K	1K	16K	16K/4K	8K†	16K/16K	16K/8K	8K/4K
FPU?	Yes	No	No	Yes	No	No	No	No	No
MMU?	Yes	Yes	No	Yes	No	No	Yes	Yes	Yes
Bus width	64 bits	32 bits	16 bits	32 bits	32 bits	32 bits	64 bits	32 bits	32 bits
Bus frequency	**100 MHz**	66 MHz	25 MHz	33 MHz	33 MHz	45 MHz	**100 MHz**	33 MHz	25 MHz
MIPS	**519 MIPS**	175 MIPS	12 MIPS	57 MIPS	80 MIPS*	70 MIPS	282 MIPS	81 MIPS	67 MIPS
Voltage	2.5/3.3 V	3.3 V	5 V	3.3 V	3.3 V	3.3 V	2.5/3.3 V	3.3 V	3.3 V
Power (typ)	4.4 W	0.85 W	0.6 W	n/a	1.2 W*	n/a	3.6 W	0.4 W	1.7 W
MIPS/watt	118	**206**	20	n/a	66	n/a	78	202	39
MIPS/price	5.46	**10.6**	0.8	1.90	2.29	4.07	6.27	1.80	0.78
Transistors	5,800,000	1,000,000	350,000*	n/a	1,200,000*	n/a	2,600,000	1,820,000	1,200,000
IC process	0.25µ 3M	0.35µ 3M	0.65µ 3M	0.35µ 3M	0.35µ 3M	0.35µ 3M	0.35µ 5M	0.45µ 3M	0.7µ 3M
Die size	47 mm²	29 mm²	n/a	43 mm²	89 mm²	27 mm²	79 mm²	n/a	119 mm²
Est mfg cost*	$25	$9	$8	$11	$8	$8	$18	$14	$20
Availability	3Q98	Now	Now	Now	Now	Now	Now	Now	Now
Price (10K)	$95	$17	$15	$30	$35	$17	$45	$45	$86

n/a: information not available

†with 4K of SRAM

(Source: vendors except *MDR estimates)

RESOURCES

Be Logical and Fast

Logical Effort: Designing Fast CMOS Circuits is the title of a new book by Ivan Sutherland, Bob Sproull, and David Harris. The 230-page softcover book may start a little slowly for seasoned logic designers, covering AND, OR, and XOR gates and discussing the concepts of fan-out and fan-in. Successive chapters then move on to asymmetric logic gates, delays, domino circuits, Muller C-elements, and the authors' concept of "logical effort."

The book (ISBN 1055860-557-6) sells for $43. For more information, contact Morgan Kaufmann Publishers (San Francisco) at *www.mkp.com*.

Demystifying AGP

Tom Shanley's MindShare Press has published *AGP System Architecture,* a 240-page treatise on the popular graphics-expansion bus. Covering version 2.0 of the specification, author Dave Dzatko delves into signaling, timing models, arbitration, commands, ordering and transactions, and a comparison of AGP versus PCI.

The $33 book (ISBN 0-201-37964-3) is available directly from the publisher at *www.awl.com/cseng/series/mindshare.*

Remystifying FireWire

For those who prefer their buses serial, there's *FireWire System Architecture, 2nd edition.* Covering IEEE-1394a, author Don Anderson (who gives his dedication in the form of an org chart) elucidates both the physical and electrical arcana of the IEEE-1394 standard. Tellingly, the $40 book is 500 pages long.

To order copies (ISBN 0-201-48535-4), contact the publishers at *www.awl.com/cseng/series/mindshare.*

Where PC Developers Confer

The fifth annual PC Developers' Conference and Expo offers four days of immersion in PCI, UCB, SCSI, I_2O, PCMCIA, and other PC-related acronyms. Wireless and home-networking technologies will also be on the agenda this year. Comfortably situated after Embedded Processor Forum, PC Developers' Conference is scheduled for May 24–27 in Santa Clara.

For more information, or to register, call Annasoft (San Diego, Calif.) at 800.462.1042 or visit *www.annabooks.com/confer/pcitop.htm.*

Automation. Information. Hannover.

The annual CeBIT, the world's largest business fair, comes again to Hannover March 18–24. As usual, fair exhibits will cover office automation, CAD/CAM, communications (including wireless), networking, peripherals, computers, "bank technology," and research and technology transfer.

For more information (in English and German), point your browser to *www.messe.de/cb99.*

SUBSCRIPTION INFORMATION

To subscribe to *Microprocessor Report*, contact our customer service department by phone, 707.824.4001; fax, 707.823.0504; e-mail, *cs@mdr.zd.com;* or Web, *www.MDRonline.com.*

For European orders, contact Parkway Gordon by phone, 44.1491.875386, or fax, 44.1491.875524; or send e-mail to *parkway@rmplc.co.uk,* or visit the Web at *www.parkway.co.uk.*

	U.S. and Canada*	Europe	Elsewhere
One year	$695	£495 or ¤695	$795
Two years	$1,295	£925 or ¤1,300	$1,495

**Sales tax applies in the following states: GA, KY, MA, TX, and WA. GST tax applies in Canada.*

Microprocessor Report (ISSN 0899-9341) is published every three weeks, 17 issues per year. Back issues are available on paper and CD-ROM. Volume reprints of individual articles are also available.

Ship to: