VIEWPOINT

The x86 Is Here to Stay—Get Used to It

PC Industry Status, Microprocessor Futures, and RISC vs. x86

by Brian Case

From 1984 to 1987, I was an architect of AMD's 29000 RISC architecture, worked on the microarchitecture of the first chip, and wrote an optimizing C compiler for internal use. Over the past decade, I have been an outspoken RISC proponent. Recent developments in the microprocessor industry, however, have made me reflect on my career as a RISC architect and RISC proponent more seriously than ever before. My reflections led to a couple of startling realizations—at least they were startling to me.

Before addressing those realizations, however, I hope many readers will enjoy the results of my "find the RISC-PC needle in the haystack" search. For years, I have been waiting impatiently for a RISC PC because I believed such a machine would blow away the price/performance of CISC-based PCs and give me a new and richer computing experience. Was it worth the wait?

RISC PCs Appear on Radar

I spent several hours poring over the pages of two of Silicon Valley's free computer publications, the 160-page *Computer Currents* and *MicroTimes,* weighing in at 340 pages. Studying these publications makes the state of computer retailing readily discernable.

Out of 140 ads in *MicroTimes*, I found seven that offered Apple PowerMac machines and exactly four that featured systems based on other RISCs. Of those four in *MicroTimes*, MIPS R4400-based systems were available from three vendors. All but a couple of Mac-only vendors sell x86 PC clones. One advertiser offered x86-, MIPS-, and Alpha-based systems.

The results of my search are summarized in Table 1, which contains vital statistics for a selection of Power Mac, x86, Alpha, MIPS, and SPARC systems. For the PC clones, I did not scrutinize each advertisement; there may be cheaper x86 systems.

It should be noted that the level of technical sophistication varies greatly among these vendors. For example, the Symphony dual-Pentium machine and the ASA MIPS box both come with DOS and Windows 3.1. This is strange because Windows 3.1 cannot take advantage of the dual Pentiums and it simply will not run at all on a MIPS processor. Windows NT is needed for each of these systems. (I called the vendors and explained the contradiction, but both insisted the systems came with DOS and Windows 3.1. I got the impression that the sales people have limited experience with the RISC systems.)

PowerMacs Lose to PC Clones

One of the first things to notice is that although Apple has dramatically improved the price/performance of the Macintosh line, the PC-clone market has done even better. A full 90-MHz, 101-SPECint92 Pentium system can be had for less than the price of a 60-MHz,

Vendor, System Name	Processor	Clock (MHz)	Off-Chip Cache	RAM	Disk	Expan. Bus	Keyboard, Monitor	CD-ROM	OS Software	SPEC92 int	SPEC92 fp	Price
PowerMac 6100/60	PPC 601	60	—	8M	250M	—	yes, 14"	2×	Mac OS	66[2]	74[2]	$2,259
PowerMac 6100/60 + Accel.	PPC 601	80	256K	8M	250M	—	—, —	—	Mac OS	88[2]	98[2]	$1,838
PowerMac 8100/110	PPC 601	110	256K	16M	2G	NuBus	—, —	—	Mac OS	121[2]	135[2]	$5,888
Mega Tek[1], Mega System	Pentium	90	256K	4M	210M	PCI/ISA	yes, 14"	—	DOS/Win	101[2]	73[2]	$1,499
Konicom[1], Pentium 90	Pentium	90	256K	8M	540M	PCI/ISA	yes, 15"	—	DOS/Win	101[2]	73[2]	$1,949
Symphony[1], Advanced Dual 9	Pentium × 2	90	256K tot.	16M	1G	PCI/EISA	yes, 15"	—	DOS/Win(!)	101[2]	73[2]	$4,328
Now Elec., DEC Alpha PC	Alpha	200	2M	16M	1G	PCI/ISA	yes, 17"	4×	WinNT	108[3]	135[3]	$6,795
Now Elec., DEC Alpha PC	Alpha	233	2M	16M	1G	PCI/ISA	yes, 17"	4×	WinNT	158[3]	184[3]	$7,695
Now Elec., DEC Alpha PC	Alpha	275	2M	16M	1G	PCI/ISA	yes, 17"	4×	WinNT	200	291	$8,195
Top Data, RISC PC	R4400	135	512K	—	—	ISA	—, —	—	WinNT	85[2]	86[2]	$2,399
ASA Computers, Thoroughbred	R4400	150	512K	16M	1G	EISA	yes, 17"	—	DOS/Win(!)	95[2]	96[2]	$4,121
Now Elec., NEC SuperStation	R4400	150	512K	16M	1G	EISA	yes, 21"	3×	WinNT	95[2]	96[2]	$8,595
Now Elec., NEC RISCServer	R4400 × 2	150	1M/CPU	64M	3G	EISA	yes, 14"	3×	WinNT	100[2]	101[2]	$15,995
PromoX, SS20/61 clone	SuperSparc	60	1M	32M	540M	S-Bus	yes, 15"	—	Solaris	77	98	$9,265
PromoX, SS20/2×61 clone	S-Sparc × 2	60	1M total	32M	540M	S-Bus	yes, 15"	—	Solaris	77	98	$12,355

Table 1. Computer systems found in advertisements from the free computer publications *MicroTimes*, 1/2/95, and *Computer Currents*, 12/13/94. [1]"No-name" clone. (Source: vendors except [2]Estimates based on available performance data; [3]Performance data from a DEC advertisement in *PC/Computing*, 1/95, p. 123)

66-SPECint92 PowerMac system. The Pentium system has more than twice the disk space and offers PCI expansion slots to boot. If you are willing to settle for less disk, less RAM, and a lower-quality display, you can get a 90-MHz Pentium box for $1,500.

For less than the price of the 110-MHz PowerMac 8100/110, you can get a dual-processor Pentium system. The Pentium system has less disk, but it includes a monitor. True, the 8100/110 system can potentially deliver 20% better performance for applications that cannot take advantage of two processors, but for applications that can use both processors—e.g., a Windows NT file server—the price/performance of the dual-processor Pentium box easily beats that of the PowerMac.

Most Other RISC Boxes Are Embarrassing

Beyond the PowerMacs, the sad state of RISC PCs is apparent from the data in Table 1. The cheapest way to get a MIPS box would be to add peripherals to the Top Data RISC PC. The configured price for a system with a 540M disk, 16M of RAM, and a 15" monitor would be about $3,500. That seems reasonable, except for the fact that a Pentium system that costs 34% less delivers about 18% more integer performance.

The next cheapest system is the ASA 150-MHz MIPS box at $4,121. Unfortunately, this system still delivers less integer performance than the much cheaper Pentium. The NEC single-processor MIPS box from Now Electronics is even more outrageous.

The dual-processor MIPS machine is truly embarrassing. For your $16,000, you get a lot of RAM, a big disk, larger off-chip caches, and a triple-speed CD-ROM. To bring the Symphony dual-Pentium box up to the same specs would raise its price to about $8,000, or one-half of the MIPS system's cost. The MIPS box will probably have a slight integer performance advantage because its second-level cache is much better.

While the dual-processor MIPS box may be an embarrassment, the SparcStation clones make me want to avert my eyes. The integer performance of the single-processor box is not even close to that of a 90-MHz Pentium machine—Pentium delivers 30% better SPECint92—and the SPARC clone costs more than three times as much as a similarly configured Pentium. The dual-processor box needs no comment. A $2,100 SPARC clone in a *Computer Currents* ad caught my eye until I called to find out it includes no RAM, no disk, no monitor, and—get this—no microprocessor. Unless you have SPARC-specific software or a SPARC-specific environment, run—don't walk—in some other direction.

While the cheapest Alpha system beats the Pentium in integer performance—108 to 101 SPECint92—it unfortunately costs two and one-half times as much! (I think I can get a 4×speed CD-ROM drive for less than $4,000.) On the other hand, a really fast Alpha system—the 275-MHz box—costs only another $1,400. For someone with a genuine need for 200-SPECint92 performance for a non-parallelizable problem, DEC offers a solution. This 275-MHz system is the only non-Macintosh RISC box that seems to have a justified existence.

Note that most of the RISC systems come with Ethernet standard and that the Pentium performance shown in Table 1 requires an optimized design. On the other hand, you can get a PC Ethernet board for between $50 and $150, and many RISC PCs probably don't deliver the performance of their optimized brethren either.

Architecture Advantage Less Than Predicted

What is so disappointing to me is not only are RISC boxes not cheaper, but they do not, in general, deliver higher performance than comparable x86 machines. Some say that the problem with RISC PCs is not their performance but their price, but I say that even if they were *cheaper* than x86 PCs, few people—buyers, software developers, or clone makers—would be interested.

What happened to the guaranteed, "blow your socks off" potential of RISC microprocessors? I cannot speak for other RISC proponents, but I believed that RISC architectures were a *qualitative* improvement in the basic art of microprocessor design, just as automobiles were a qualitative improvement over horse-drawn buggies in transportation. RISCs *are* a distinct improvement—there is clearly no advantage to designing a new architecture as a CISC—but they are only a quantitative improvement, and the magnitude of that improvement seems to be smaller than I believed.

Next Generation: More of the Same

The data from Table 1 shows that for shipping, readily available systems, the x86 is less than a factor of two behind all but one RISC system in integer performance. If the 100-MHz Pentium is included, the difference is even less. A new generation of microprocessors, however, is imminent. Will RISC chips pull away from x86 implementations? No, not in shipping, readily available systems that could be purchased and used like PCs.

Table 2 shows performance estimates for next-generation implementations of all major general-purpose microprocessor architectures. The next-generation RISC implementations show strong growth in integer performance, and they all set a floating-point standard that leaves Pentium-class x86 chips in the dust. (I have a feeling, however, that x86 vendors do not care.)

The 120-MHz Intel Pentium is certain to appear this year, and with AMD set to make an impact in the high-end market, Intel should have extra incentive to stick to its plan to introduce the P55C Pentium at 150 MHz. AMD plans to position the 100-MHz K5 against the 120-MHz Pentium this year; just for comparison, the table shows where a 150-MHz K5 would fit.

As Table 2 makes clear, despite the gains made by the RISCs in integer performance, the x86 will stay within a factor of two (plus or minus a few months). Further, if AMD should somehow be able to ship a 150-MHz K5, or if Intel can ship the P6, or if NexGen ships its 686, the x86 camp will actually move closer to the high-end RISCs. Remember, this is for high-volume, readily available PC-like boxes; I wonder if the 366-MHz 21164 will ever achieve truly high volume production.

Which brings me to my next point: the highest-performance RISC chips in Table 2 are somewhat like concept cars in the auto industry. They can be built—and will be built—in some volume. If Intel or AMD wanted to spend similar amounts of money, they could build some fast, huge P6 or K5 chips and ship them soon. In fact, Intel could probably ship a few 120-MHz Pentiums, and maybe even some 150-MHz chips, right now. Doubling K5's instruction cache would dramatically improve performance and is something a low-volume RISC vendor might do, but that would price K5 out of its market. So I think Table 2, except for PowerPC, is comparing RISC apples to x86 oranges.

Clock-Rate Gap Will Shrink

Intel-architecture chips have lagged RISCs in two important areas: floating-point performance and clock speed. Until there is a major shift toward floating-point computation in mainstream PC applications, x86 implementations will probably continue to lag RISCs in megaflops by a huge margin.

I think the clock-speed differential, however, will shrink. RISCs have had faster clocks than x86 chips for three primary reasons. First, the complexity of the x86 has limited clock speed. To fit all the x86-specific logic in a reasonable number of pipeline stages requires that the amount of logic in some of those stages be fairly large.

Now, with plenty of transistors, thinner wires, and more chip area, it is possible to implement good branch prediction and some amount of predecode information in the instruction cache. Good branch prediction mitigates the negative effects of long pipelines, and predecode information in the instruction cache reduces the amount of logic in some pipeline stages. Together, these two implementation techniques should enable designers to build x86 implementations with faster clock rates.

Processor	Ship Date	Clock Speed	SPECint92	SPECfp92
MIPS R10000	1Q96	200 MHz	>300	> 600
PowerPC 620	3Q95	133 MHz	225	300
UltraSPARC	3Q95	167 MHz	275	305
Alpha 21164	4Q95*	366 MHz*	400*	600*
Alpha 21164	1Q95	300 MHz	330	500
PA-8000	1Q96	200 MHz*	375*	500*
P6	4Q95	133 MHz	200	200*
Pentium (P55C)	4Q95*	150 MHz*	150*	100*
Pentium	2Q95*	120 MHz*	130*	90*
AMD K5	2Q96*	150 MHz	200*	100*
AMD K5	3Q95*	100 MHz	130*	75*
NexGen Nx686	4Q95*	150 MHz*	200*	100*
NexGen Nx586	4Q94	100 MHz	110*	n/a

Table 2. Performance of next-generation RISC and expected x86 processors. (Source: vendors except *MDR estimates)

The second reason x86 clock rates have lagged RISC clock rates is because of system-design issues. Cheap systems demand a certain degree of pedestrian system design (low frequency and power). This is one reason Intel has stepped up its board business: to allow "Joe's Clones" to build an effective Pentium system.

The third reason x86 chips have lagged RISCs in clock speed is that there has been a simple lack of competition. Now that Intel has AMD, NexGen, and Cyrix/IBM to worry about—although Intel still has a huge lead in sheer production capacity—I think we will see x86 chips with clock rates equal to at least midrange RISC chips in less than two years.

A RISC Market Opportunity

Given that the x86 is not going to roll over, die, and name RISC as the inheritor of the PC market in its will, where can RISCs find a volume market? Certainly, in the areas where they are already successful: special-purpose workstations, floating-point-intensive scientific work, and high-end mainframe replacements. I think, in addition, RISCs have a chance to dominate high-bandwidth content servers for interactive cable systems, whatever they turn out to look like.

Wanna-be content providers and the traditional providers of information infrastructure smell profit in interactive services like video on demand. Experimental video-on-demand systems rely on high-speed file servers and a huge array of disks to store a big content library with low latency and high bandwidth. One system (see *Time*, 12/26/94, p. 125) allows the home viewer to prowl the entire library as if with a computer terminal. The viewer can freeze a movie, change attention to another service or title in the library, and when finished there, return to the original movie just where it was frozen.

To provide this flexibility to thousands (or millions) of viewers simultaneously requires huge amounts of storage and compute bandwidth to manage massive data-transfer rates. Direct addressability for kilo-terabytes of storage might even give the high-end RISCs—with 64-bit addressing—a compelling advantage. The volumes in this video-server market, however, are relatively small—no bigger than current RISC markets.

Decoupled Superscalar from Now On

What will microprocessor implementations look like over the next few years? High-end chips will use decoupled, asynchronous superscalar organizations with

register renaming and out-of-order execution *(see **081102.PDF**)*. Processors using some of these techniques are already available, and full-fledged designs are coming soon in the form of the K5, P6, R10000, and others.

Decoupled designs will be used for the next few years because they are so scalable. It is possible to add function units, enlarge reorder buffers, increase the sophistication of branch prediction and instruction predecode, add operand and result buses, etc., without changing the basic philosophy of the design.

The step from single-pipeline designs of the late 1980s to the first decoupled superscalar implementations was a big one. Now, advances in circuit density and speed will be exploited by relatively small changes in microarchitecture. First-level caches will get bigger until they are the clock-rate bottleneck. Then, as Digital has already done with the 21164, designers will add on-chip second-level caches. Special function units for graphics and multimedia are also starting to make appearances.

Before the designers can go wild with large numbers of function units and wider dispatch capabilities, application developers and compiler optimization will need to catch up with the current capabilities of superscalar processors. Proper source coding and better instruction scheduling will be required to fully exploit the hardware parallelism of the next few generations of high-end microprocessors.

VLIW Will Not Displace x86 or RISC

I think VLIW microprocessors will be successful in the embedded market, but I do not think they are appropriate for general-purpose computers. Advances in process technology allow more concurrency (more operations per cycle) in implementations. With traditional architectures—RISCs and the x86—this is exploited by higher degrees of superscalar capability. To exploit better technology with a second-generation VLIW, either the architecture must change (a wider instruction word to accommodate more operations) or designers must implement a superscalar VLIW. Neither of these options is good, because widening the instruction breaks binary compatibility, and implementing a superscalar VLIW has no advantage over a superscalar RISC or x86.

Opportunity Knocked, I Didn't Answer

OK, now for the cathartic realizations I mentioned at the beginning of this article. Over the past year, it has become obvious to me that the transition to decoupled superscalar implementations has put the x86 on a much more level playing field than it has been in the past. When implementation technology was not dense enough to permit these sophisticated superscalar organizations, RISCs had a clear advantage. The advantage was especially profound in the 1980s when it looked, at least to me, as if CISCs were on their deathbeds.

Now, we can implement microarchitectures that extract parallelism from instruction streams and schedule low-level operations in a near-optimal fashion. RISC instruction sets do have advantages for encoding parallelism and simplifying the hardware that extracts it, but the x86 instruction set can encode the same algorithms, if less elegantly.

Given this knowledge, I recently realized that we who were designing and building the first 29000 RISC microprocessor missed one of the greatest opportunities of our lives: the chance to build a pipelined x86 before—or at least concurrently with—Intel. While the 1.2-micron CMOS technology we were using wouldn't have allowed a full-fledged Intel 486 design, we probably could have implemented something close to the C&T Super386 or the original Cyrix 486.

If we had done a pipelined x86, the microprocessor industry would likely be profoundly different now. First, the litigation between AMD and Intel might have been even worse. AMD's annual revenues might be two, three, or even more times bigger than they are now. With AMD as competition, Intel might currently be attempting to phase out Pentium-class chips in hopes of steering the market to its next-generation P6-class, which might just now be achieving volume production. Or, since AMD is actually introducing its K5 *now*, might Intel be phasing out its P6-like chip in favor of a P7-like chip? Well, maybe not, since competition would not have accelerated process technology, upon which these processor generations depend, but I do think the pace of design innovation would have been quickened.

In any case, I wonder why I did not see the writing on the wall years ago. I can take only a little comfort from the fact that few, if any, other microprocessor designers were able to see the future any better. I wonder now, "Where were all the academic papers describing the coming technology and what it would mean for microprocessor implementation?" Even Mike Johnson's *Superscalar Microprocessor Design* textbook only waves its hands at a superscalar x86 implementation; it does not even seriously discuss a K5-like chip.

And I do not want to hear any CISC bigots gloating "I told you so." The past few years have been filled with a lot of bickering, but I never once heard anyone make the appropriate technical argument, which would have been something like, "RISCs have a definite advantage for a simple pipeline, but wait until technology allows decoupled superscalar organizations. Then the x86 won't look so bad."

So the x86 is here to stay, at least for the next few years, thanks to advances in implementation technology. RISCs are better—but not by enough—and the x86 is supported by insurmountable market forces. I might not like it, but I can get just as much work done with an x86 as with a RISC, and cheaper to boot. ♦

