

Happy Birthday x86! An Industry Standard Turns 30

Gary Anthes
(ComputerWorld)

Intel's x86 microprocessor architecture has dominated large swaths of computing for three decades. Here's why.

Thirty years ago, on June 8, 1978, Intel Corp. introduced its first 16-bit microprocessor, the 8086, with a splashy ad heralding "the dawn of a new era." Overblown? Sure, but also prophetic. While the 8086 was slow to take off, its underlying architecture -- later referred to as x86 -- would become one of technology's most impressive success stories.

"X86" refers to the set of machine language instructions that certain microprocessors from Intel and a few other companies execute. It essentially defines the vocabulary and usage rules for the chip. X86 processors -- from the 8086 through the 80186, 80286, 80386, 80486 and various Pentium models, right down to today's multicore chips and processors for mobile applications -- have over time incorporated a growing x86 instruction set, but each has offered backward compatibility with earlier members of the family.

In the three decades since the introduction of the 8086, the x86 family has systematically progressed from desktop PCs to servers to portable computers to supercomputers. Along the way, it has killed or held at bay a host of competing architectures and chip makers. Even some markets that had seemed locked up by competitors, such as Apple's use of Motorola PowerPCs in the Macintosh computer, have yielded to x86 in recent years.

How did Intel's architecture come to dominate so much of the computing world? Let's take a look.

In the beginning

Intel's first microprocessor was the 4-bit 4004, which was made for a Japanese calculator in 1971. That was quickly followed by the 8-bit 8008 and in 1975 by the 8-bit 8080 chip. The 8080 went into the Altair 8800 PC, which was sold as a mail-order kit. (Bill Gates and Paul Allen founded Microsoft Corp. to sell their version of Basic for the Altair 8800.)

Intel, memory maker?

Intel made its first microprocessor in 1971 in response to a request from Busicom, a Japanese calculator maker. But Intel's founders in 1968 had semiconductor memory foremost in mind, and such a chip became the company's first product, in 1969.

For almost 20 years, Intel focused on memory products. But by 1984, according to Albert Yu, a retired Intel senior vice president, the company was getting killed by the Japanese memory makers. It was deriving 100% of its profits from microprocessors but spending 80% of its R&D budget on memory. "Our strategy and investments were completely out of line with reality," Yu recalls in his book, *Creating the Digital Future*. The following year, Intel reluctantly exited the memory business.

Yu recalls: "We finally overcame the emotional burden of letting go of a failing business that we had invented and focused all our energy on the business we would build our future on. It was tough. It was gut-wrenching. But it was right." The following year, Intel's sales dropped from \$1.6 billion to \$1.2 billion, and the company lost \$250 million from restructuring.

Three years later, the 16-bit 8086 made its debut. IBM's selection of the 8088, an 8086 variant, to power its PC in the early '80s gave the x86 architecture tremendous momentum and helped it become an industry standard that persists today.

Happy Birthday x86! An Industry Standard Turns 30

Gary Anthes
(ComputerWorld)

Patrick Gelsinger, electrical engineer, chip designer and now executive vice president at Intel, says the critical turning point for the PC industry -- the thing that really sent the industry into overdrive -- was the introduction of the 32-bit 80386 in 1985. It was not obvious at the time that the x86 needed to be upgraded from the 16-bit address space of the earlier models, he says. "People said, 'What do you mean 32 bits? That's for minicomputers and mainframes.' They derided us at the time for being extravagant."

At about the same time, Compaq Computer Corp. announced a 386-based PC, which lessened IBM's death-grip control of the personal computer market. The IBM PC at the time ran the 16-bit 80286, which was more than three times slower than the 386.

According to Intel, IBM spurned the 386 because there was not yet any 32-bit software to take advantage of it. IBM was also developing a proprietary 16-bit operating system called OS/2.

x86 memories



"IBM owned the architecture from top to bottom. It was their applications, their operating system and their hardware design," says Gelsinger, who was a member of the 386 design team. "When they went to the next generation, they would be the only company able to offer the top-to-bottom solution, with no guarantee of compatibility from one generation to the next."

All that changed with the advent of the 386, Gelsinger says. "We moved from a vertical industry to a horizontal industry, and that really opened up the world."

The 386 was followed by the 486 in 1989. Finding that it couldn't trademark numbers, however, Intel broke from its earlier naming convention in 1993, when it named its fifth-generation processor the Pentium rather than the 586. Numerous generations of chips have carried on the Pentium brand (e.g., Pentium Pro, Pentium II and Pentium D), and Intel has since added the low-end Celeron and the high-end Core 2 brands to its x86 offerings.

Despite the name changes -- not to mention design improvements that led to exponential increases in speed, power and efficiency -- all of these chips are based on the x86 instruction set that began with the 8086 and continues to expand today.

Happy Birthday x86! An Industry Standard Turns 30

Gary Anthes
(ComputerWorld)

Ingredients in a recipe for success

Why has the x86 been so successful for so long, beating back and in some cases completely vanquishing competing microprocessor architectures? For starters, the x86 came along at just the right time. By 1978, computing had been migrating from huge, expensive mainframes to smaller, cheaper minicomputers for several years. The desktop was the logical next frontier.

Moreover, the x86 demonstrated a property that had been predicted in 1965 by Gordon Moore, who would one day become Intel's chairman and CEO. Moore said, in essence, that microprocessors would double in performance every two years at no increase in cost. His prediction, later dubbed Moore's Law, proved to be correct, and the x86 went on to dominate large swaths of computing, from the data center to the workplaces and homes of end users.

And the 8086 and its successors continued to cement the relationship between two early giants of the desktop computer industry. Bill Gates and Paul Allen had tried but failed to develop their Basic programming language for the wimpy 8008 processor in 1972. But they made it work on the more powerful 8080 that they soldered into the Altair microcomputer in 1975.

That marked the beginning of a de facto partnership between Intel and Microsoft that would create a gargantuan base of software that continues to drive the industry today. Of all the factors that have led to the success of the x86 architecture, probably none is so important as that software inventory -- and no example better demonstrates this fact than the RISC processor scare.

The RISC risk

In the late 1980s and early 1990s, a serious threat to the x86 arose in the form of reduced instruction set computing (RISC) processors such as the Sun Sparc, the IBM/Apple/Motorola PowerPC and the MIPS processors. The idea was that a processor could be made to run blindingly fast if it worked on very simple instructions, with one instruction executed each clock cycle, rather than with the elaborate, multicycle instructions used in complex instruction set computing (CISC) processors like the x86.

The software spiral

At the time the 80486 processor debuted, Intel CEO Andy Grove developed a kind of corollary to Moore's Law called the "software spiral." Patrick Gelsinger, lead architect for the 486, explains:

"The idea of the spiral was we'd introduce hardware that was faster than the existing software and then allow the software to catch up, which requires the hardware to jump ahead again. So software begets hardware, hardware enables software, and that spiral is really what's been driving the industry for many, many years."

Indeed, the spiral effect is very much in evidence today, with multicore chips available but with little software that can really take advantage of them by running on multiple processors at once. It's why Intel is spending millions of dollars internally and at universities to develop tools and techniques for parallel programming.

Pundits, the press and Intel competitors widely predicted the demise of CISC at the time. "It was a difficult time for us," Gelsinger acknowledges. Indeed, Intel rushed to develop its own RISC workstation processor, the i860. But neither the 860 nor any other RISC processor came close to dislodging the hegemony of the x86.

Happy Birthday x86! An Industry Standard Turns 30

Gary Anthes
(ComputerWorld)

Here's why, according to Gelsinger, who was the lead architect for the 80486 processor: "The day before the 486 was announced [April 10, 1989], there was already billions of dollars of software waiting to run on the chip. Even though the [x86 CISC] architecture was a little bit slower, by the time you could develop software for the RISC machine, we could make the [x86] machine that much faster. We had an overwhelming economic advantage because we had so much of an installed base and so many people developing. The RISC machine could never catch up."

Ironically, the lack of software for RISC machines -- plus big performance gains on the 80486 and Pentium processors -- doomed Intel's i860 along with other RISC processors. Trying to introduce a second major microprocessor architecture was a mistake, Intel would later admit.

But RISC spurred much innovation, says David Patterson, a computer science professor at the University of California, Berkeley, and one of the key RISC innovators in the 1980s. "The [Digital Equipment Corp.] VAX architecture, for example, could not keep up with RISC, and it more or less disappeared. But Intel was able to incorporate the ideas that were becoming popular in RISC while maintaining their old architecture with its large software base. And they did that in part with their superior manufacturing."

The floating-point fiasco

Perhaps as gut-wrenching as the RISC threat was a crisis that began in the summer of 1994, when Intel test engineers discovered a tiny flaw in the floating-point division circuitry of its new Pentium chip. The flaw occurred so rarely and was so minor in its impact that Intel elected to just fix it and put the chip back into production without recalling the flawed chips.

But a few months later, Thomas Nicely, a math professor at Lynchburg College in Virginia, discovered the flaw in his PC. He was unable, Intel was to admit later, to find anyone at Intel who would even listen to his complaint. So he posted his findings on the Internet, and before long, Intel was engulfed in a firestorm of criticism that would ultimately lead to a public relations disaster and a \$475 million recall of the chip.

"It was a painful rite of passage, but we finally learned to behave like a consumer company," recalls Albert Yu, a former Intel senior vice president, in his book, *Creating the Digital Future*.

Mixing and matching

Another defining moment in x86 history occurred in 1995, says Todd Mowry, a computer science professor at Carnegie Mellon University and an Intel research consultant. That's when Intel introduced the Pentium Pro, a microprocessor with some radical new features, such as the ability to look ahead in a stream of instructions, guess which ones would be needed and then execute them out of order. That kept the processor busy a larger percentage of time and, combined with a new, extremely fast on-chip cache, it offered huge performance gains in some applications.

"The thing that was radically different," Mowry says, "was that they used the benefits of RISC without changing the instruction set. They did that by translating the x86 instructions into micro-operations that are more like RISC instructions. So what you had was a RISC machine inside an x86 machine, and overnight, that eliminated the performance gap."

Mowry says the Pentium Pro resulted from a top-down design process. "They started out with the design of a fast machine and then figured out how to make the x86 run on it," he says.

Happy Birthday x86! An Industry Standard Turns 30

Gary Anthes
(ComputerWorld)

That approach -- finding good ideas in non-x86 architectures and working backward from them -- was just how it worked, Gelsinger says. "The Pentium was a dramatic architectural leap. We took the best ideas from minis and mainframes and just implemented them better, because we had a superior canvas to paint them on, called silicon."

Unlike a mainframe, which spreads processing components over a wide area inside the box, putting everything on a single, tiny, tightly integrated chip gives microprocessor designers more flexibility and their designs more power, he says. Indeed, over the years, the performance of silicon chips has marched smartly along according to Moore's Law, while systems of interconnected components have not improved as fast.

Microprocessor for a parallel universe

Intel introduced the 8-bit 8080 microprocessor in 1975, and immediately CEO Gordon Moore began to worry about what would come next. He wanted it to be an architecture that would carry the company forward for decades. Many at Intel favored extending the 8080 architecture to 16 bits while others, including ultimately Moore, favored developing a brand new and much more advanced architecture.

So while engineers in Santa Clara, Calif., plugged along with the x86 architecture (the 16-bit successor to the 8080 architecture), Moore hired some bright engineers and started a lab in Portland, Ore., to design a processor that would become known as the iAPX 432 "micro-mainframe." But the design was so advanced -- for example, it included features for fault tolerance and for multicore chips, which wouldn't be needed for 25 years -- that the processor became too big and complex.

Introduced finally as a three-chip 32-bit processor in 1981, the 432 was expensive and slow, and it didn't sell. Customers for the time being seemed happy with the 16-bit 8088 and 80286 models.

"The Santa Clara and Portland efforts were rivals; they were competitors," says UC Berkeley professor David Patterson. Although the x86 designers were able to borrow some ideas from the 432 -- such as its floating-point math operations -- the x86 designs at the time lacked the elegance of the 432 architecture, he says.

In 1985, Intel made another stab with the 432, forming a joint partnership with Siemens, called BiiN, to build a fault-tolerant computer. But that didn't go anywhere, either.

What if the 432 had succeeded back in the late 1970s? "The x86 wouldn't have been invented," Patterson says. "Had the group that built the 432 built something less ambitious, but which was out in the market ready to go, that's what we would be celebrating today. In some parallel universe, that's what Intel would be shipping today."

The competition heats up

Intel has not enjoyed immunity from competition even on its x86 home turf. For example, Taiwan-based VIA Technologies was founded in Silicon Valley in 1987 to sell core logic chip sets, some using x86 technology, for use in motherboards and other electronic components. VIA now makes a wide variety of products and aims its x86 processors at low-power mobile and embedded markets.

Advanced Micro Devices Inc., the world's No. 2 maker of microprocessors, has become a competitive thorn in Intel's side since about 2000. Throughout most of the 1980s and 1990s, AMD had been a me-too maker of x86 chips and was hardly any concern to Intel. (It still has only about 15% of the x86-compatible desktop and mobile market, according to Mercury Research.)

Happy Birthday x86! An Industry Standard Turns 30

Gary Anthes
(ComputerWorld)

But AMD scored a technical and public relations coup in 2003 with its introduction of x86-64, a 64-bit superset of the x86 instruction set. As a superset, it meant that users of new x86-64 machines could use them to natively run their old 32-bit software.

At the time, Intel's 64-bit offering was Itanium, an architecture developed by Intel and Hewlett-Packard Co. for superscalar execution on big iron, and it was not directly compatible with 32-bit x86-based software. Intel responded to the AMD threat with its own 64-bit x86 instruction superset, the EM64T, in 2004. AMD, and the press, made much of the fact that the company had beaten Intel to the 64-bit market that mattered most.

"It's an example of where the flexibility of the x86 instruction set was used against Intel," says Patterson. "So even though Intel dominates the market, another company can change directions for the x86."