

# CPU Technology Has Deep Roots

*Groundwork for Today's Microprocessors Set in Early 1970s*



by Linley Gwennap

As a span of human history, 25 years is a brief interval, just over a single generation. Microprocessor generations, however, occur roughly ten times faster than human ones. By this measure, November will mark 250 microprocessor years since the Intel 4004, the first commercial microprocessor, began shipping in 1971. As with a similar period in human history, the microprocessor era has seen the rise and fall of empires; perhaps more interesting is the evolution of the underlying technology that has changed the world.

For software developers, the x86 instruction set is the most widely used on the planet, but few realize that key aspects of this instruction set were created as much as 25 years ago, before there was a single shipping microprocessor. Microprocessor designers are always eager to try the latest out-of-order superscalar methods, but fundamental hardware techniques used in today's CPUs also date back to the dim recesses of early microprocessor history (and, indeed, to even earlier mainframe systems).

It isn't possible to discuss all of the significant chips that have appeared during the 250 microprocessor years of history. The chips discussed here have made significant contributions to CPU design by bringing to market new techniques that have become fundamental parts of most commercial microprocessors today.

## Primordial Soup

In 1961, Fairchild released the first commercial integrated circuits, three years after the IC was created in a laboratory. These first components contained just a few transistors connected by on-chip wiring, but engineers at Fairchild and other companies set forth to rapidly reduce the size of the devices placed on these semiconductor chips, thereby increasing their capacity. Within a few years, Gordon Moore, then R&D director at Fairchild, made his famous forecast that the number of transistors able to be placed on a single chip would double every 18 months, a forecast so accurate for so long it became known as Moore's Law.

The first ICs were used in many types of electronic devices. One of the least significant in terms of volume was the computer. In 1961, the entire computer industry, dominated by IBM mainframes, sold perhaps a thousand systems per year, hardly a burgeoning market.

By the late 1960s, the minicomputer, exemplified by Digital Equipment Corp.'s PDP-8, had become a popular lower-cost alternative to mainframes, and both mainframes and minicomputers had adopted IC technology. These

systems typically contained hundreds of chips with about 100 transistors each (MSI), plus many smaller chips, to implement their complex 16-bit architectures. Moore's Law made it clear the day would come when a complete mainframe or minicomputer CPU could be built on a single chip, but the semiconductor capacity curve would not intersect with the transistor count of these processors until the late 1970s.

As a result, the computer companies of the day did not pursue the idea of a single-chip CPU, or microprocessor. They focused instead on increasing the performance of their systems, trying virtually every computer architecture technique used today, including reduced instruction sets, pipelining, superscalar dispatch, register renaming, and out-of-order execution. It was left to the emerging semiconductor makers to develop the microprocessor.

## Founding Fathers

In 1968, Robert Noyce left Fairchild in a management dispute. He founded Intel along with Moore and Andy Grove, who had worked for Moore at Fairchild. Within a year, the company introduced its first products: a 64-bit bipolar SRAM and a 256-bit PMOS SRAM. The following year, Intel introduced the first commercial DRAM, a 1,024-bit device, and began work on its first EPROM. Noyce realized the ongoing improvements in IC technology would rapidly make these devices much less expensive than the discrete flip-flops and core memories then used for storage.

Noyce also knew the same improvements would make other types of large-scale integrated (LSI) circuits attractive in the near future. Fortuitously, a Japanese company named Busicom met with Intel in 1969, requesting a set of several custom chips for a new calculator. The Intel team realized the calculator could instead be built around a single general-purpose processor chip along with ROM chips to hold the programming and RAM chips for the data. These generalized chips could then be sold to other companies to solve other problems, giving Intel a new product line. The Busicom processor became the Intel 4004, designed by Ted Hoff, Stan Mazor, and Federico Faggin, along with Busicom's Masatoshi Shima, who later joined Intel.

## First CPU on a Chip

Hoff knew he couldn't fit a complete mainframe-type CPU onto a single chip with only 2,300 transistors. Instead, he selected a 4-bit architecture, the size of a single decimal (BCD) digit. This choice eliminated 75% of the transistors in the ALU and register file as compared with a 16-bit CPU. As Table 1 shows, the 4004 had only 45 instructions, each encoded in either 8 or 16 bits. Intended for a calculator, the chip had no logical operations and no multiply or divide


Mnemonic	Operation	Mnemonic	Operation
SRC	Send address	LDM/FIM	Load 4/8-bit immediate
RDM/WRM	Read/write A to RAM	ADD/SUB	Add/subtract register to A
RDR/WRR	Read/write A to ROM port	LD	Move register to A
WMP	Write A to RAM port	XCH	Exchange register with A
WR[0-4]	Write A to RAM register	IAC/DAC	Increment/decrement A
RD[0-4]	Read RAM register to A	CMA/CMC	Complement A/carry
ADM/SBM	Add/subtract RAM to A	CLC/STC	Clear/set carry
FIN	Load indirect	CLB	Clear both A and carry
JUN	Jump unconditional	TCC	Copy carry to A
JCN	Jump on condition	TCS	Subtract carry from A
JIN	Jump indirect	RAL/RAR	Rotate A left/right
JMS	Jump to subroutine	DAA	Convert A to BCD
BBL	Return from subroutine	KBP	Keyboard conversion
INC	Increment register	DCL	Designate command line
ISZ	Inc. and jump if not zero	NOP	No operation

**Table 1.** The 4004 instruction set is quite primitive compared with that of today's processors.

instructions. Before reading or writing memory, a separate "send address" instruction was required.

Most instructions operated on the accumulator, but the 4004 included 16 general-purpose registers of 4 bits each. It also contained a 12-bit program counter and a four-entry stack to hold subroutine return addresses. Most large systems kept the stack in main memory, but the 4004 relied on custom RAM chips optimized for floating-point BCD data, so Intel placed the stack on the CPU chip instead.

The 320-bit RAM chips each held four floating-point numbers in an 80-bit format, allowing 16 decimal digits for the fraction plus two digits of exponent and two control digits. Today's Intel chips continue to use an 80-bit floating-point format (although the data is kept in binary rather than BCD form), while most of the rest of the industry has adopted a 64-bit format.

The 4004 die, shown in Figure 1, measured just 12 mm<sup>2</sup> (about this big: ) and was built on 2" wafers. The largest available package had only 16 pins, which restricted the designers to a 4-bit external bus. Executing a single-byte instruction required eight clock cycles, including three to send the 12-bit address and two to load the instruction across the narrow bus; two-byte instructions executed in 16 cycles. The chip had a clock speed of 750 kHz, which may not seem like much but gave it performance similar to that of ENIAC, the first electronic computer. Intel shipped the first devices to Busicom in February 1971; the 4004 was introduced for public consumption in November of that year.

### Intel 8008 Builds Foundation

About the time it began the Busicom project, Intel also developed a custom shift register for Datapoint (then Computer Terminal Corp.), a company building its own 8-bit processor from custom MSI chips. Several months later, with the 4004 progressing well, Intel proposed building a complete 8-bit processor on a single chip. This data size, twice that of the 4004, fit the alphanumeric data of a computer terminal.

The device, which became the 8008, took a detour from the 4004 instruction set to offer as much compatibility as possible with Datapoint's existing processor. Instead of the orthogonal register set of the 4004, the 8008 had an 8-bit accumulator (A) and six general-purpose registers: B, C, D, E, H, and L. Memory was addressed only through the HL register pair. These registers later became the A, B, C, D, and index registers in the 8086 architecture, with their remnants in today's P6. The 8008 also added the concepts of logical operations and interrupts to the microprocessor world. The interrupts worked poorly, however, since the CPU lost its state after the interrupt.

Instructions and data shared the same off-chip memory, which was built from standard memory chips instead of from the custom devices used with the 4004. For compatibility with Datapoint's existing design, 16-bit addresses and data were stored in memory with the low byte first, the genesis of Intel's reliance on little-endian byte ordering.

Datapoint wanted a second source for the 8008 and contracted with Texas Instruments, which began work on the part as well. Because Intel's patent on the 4004 does not claim the single-chip processor as an invention, and the small company did not bother to patent the 8008, TI's 1971 patent on its work with Datapoint is recognized as the first claim on a microprocessor (*see 1009MSB.PDF*). Ironically, Datapoint lost interest in the project and never used the chip; TI also gave up on the device before bringing it to market. Intel persevered and began selling the 8008 in April 1972.

### Intel 8080 Extends Instruction Set

With the 8008 enjoying moderate success, Intel began receiving requests for improvements. At the same time, the company's engineers had developed a new fabrication technology, 6-micron NMOS, that increased transistor speed and density beyond the 10-micron PMOS designs in the 4004 and 8008. Faggin and Shima led the 8080 design team, which set out to increase performance by 10× over the 8008.

The new process boosted the transistor budget to about 6,000; more complicated control logic enabled the chip to process instructions in as few as five clock cycles. The designers added many new instructions and redefined the old 8008 opcodes, although 8008 programs could be easily cross-assembled to run on the new chip. In all, the 8080 defined 244 of the 256 possible opcodes and included one-, two-, and three-byte instructions.

For compatibility, the team retained the 8008 register set. In the new design, the BC and DE register pairs could be used to address memory, like the HL pair; in addition, the 8080 included 16-bit arithmetic instructions that operated on these register pairs, although the chip relied on an 8-bit ALU and thus required extra cycles for these operations. The 8080's interrupt structure was much more useful than the

8008's, providing vectored interrupts and allowing the processor to save state before processing the interrupt.

At the time, 40-pin plastic packages had become available, and the team eagerly adopted the larger package. The extra pins allowed separate 16-bit address and 8-bit data buses, simplifying system design. The 8080 debuted in 1974 at a clock speed of 2 MHz. Among other applications, it appeared in the Altair home computer kit, the first personal computer. The 8080 became popular enough to spawn several unauthorized "clones."

### Competition Blossoms

Expanding interest in the microprocessor, both from embedded applications and the emerging home computer market, led other semiconductor companies to develop their own products. Shima and Faggin left Intel to form Zilog, which deployed the Z80 in 1975. This chip ran at 2.5 MHz and executed all 8080 instructions as well as many new ones, including block move and block I/O instructions. The Z80 also added a second register set; after an interrupt or operating-system call, the processor could simply switch register sets, avoiding the need to save registers on the stack (as long as these events were not nested).

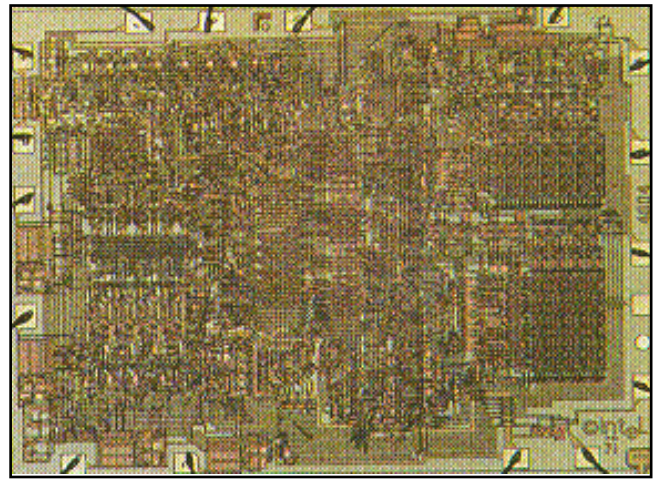
The Zilog chip simplified system design as well. The chip was the first to generate the refresh signals needed for DRAM, which had become the predominant memory type. It was also the first microprocessor with an on-chip clock circuit, requiring only an external crystal. The Z80 became very successful, outselling the Intel 8080.

Motorola completed its first microprocessor, the 6800, in 1974, shortly after the 8080. Unlike the 8080, this 8-bit processor was optimized for operating on data in memory rather than in registers. The 6800, designed by Chuck Peddle and Charlie Melear, had only two 8-bit general-purpose registers plus an index register.

Paralleling the 8080's history, Peddle left Motorola and joined MOS Technology, where he developed the 6502. This chip, which shipped in 1976, was quite similar to the 6800 but not binary compatible. At a time when most other microprocessors cost a few hundred dollars in small quantities, MOS sold the 6502 for less than \$100, which attracted companies like Commodore, Atari, and Apple to use the chip in low-cost computers, including the original Apple II.

### Early Innovation

Other vendors produced some innovative designs during this period, although many failed to achieve lasting commercial success. Texas Instruments created the microcontroller market in late 1972 with its TMS1000, which contained 1K of ROM and 32 bytes of RAM along with a very simple 4-bit CPU. No external memory could be added. The processor used 8-bit instructions; its only registers were a 4-bit accumulator, a 6-bit data pointer (enough to address the RAM with 4-bit granularity), and a 10-bit program counter. Only a single level of subroutine calls was allowed. This simple



**Figure 1.** The Intel 4004, the first commercial microprocessor, measured just  $3 \times 4$  mm and was built using 10-micron PMOS with one metal layer. The photo clearly shows individual traces and transistors; if the transistors of a Pentium Pro were printed at this scale, the entire chip would be more than 25 feet on a side.

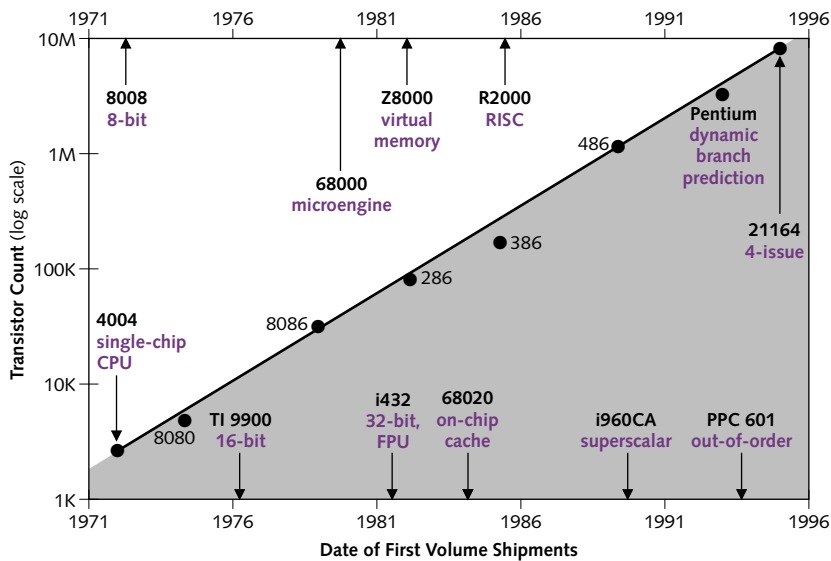
design was adequate for many embedded-control applications, and the part was very inexpensive, helping make it one of the highest volume processors of the 1970s.

TI upgraded to the TMS9900 in 1976. The first 16-bit microprocessor, the 9900 implemented its 16-bit architecture for compatibility with the TI 990 minicomputer. The company could not fit a full 16-bit register file and ALU on a single chip, however, so it took the unusual step of placing the entire register file in external memory. This technique made implementing register windows easy; the pointer to the register file in memory could be quickly changed upon an interrupt or subroutine call, improving context-switch time. SPARC would later implement a different version of register windows using on-chip storage.

National Semiconductor released its SC/MP ("Scamp") chip in 1976. From a software standpoint, the SC/MP was an 8-bit processor, but to reduce cost the internal ALU was only one bit wide, extending the basic execution time by several cycles. The chip was unique in being able to share the system bus with other SC/MP processors or with intelligent I/O devices, thus becoming the first microprocessor to support glueless multiprocessing.

The Signetics 8x300, designed by Scientific Micro Systems, appeared in 1978. A single 16-bit instruction could take data from one of eight registers, rotate it by up to 7 bits, mask off the upper 0–7 bits, and perform an arithmetic or logical operation with the accumulator. The chip also included a shift-and-merge unit. This design was ideal for signal processing. Instructions such as increment and logical OR, however, were missing from the 8x300 instruction set.

The RCA 1802 "COSMAC" processor implemented an unusual instruction set with all instructions encoded in just 8 bits (except for immediate values). It had 16 registers of 16 bits each. Arithmetic instructions combined the accumulator with the memory value pointed to by the data



**Figure 2.** This chart shows when key microprocessor features (in purple) were first introduced and the commercial products that first incorporated them. The trend line shows nearly linear (in log terms) increases in transistor count over a 25-year period. (Sources: vendors, MDR)

pointer, storing the result in the accumulator. Any register could be preselected as the data pointer. Any of the registers could also act as the program counter, providing interesting options for branches and subroutine calls.

Because of the simplicity of the design and its reliance on new CMOS technology, the 1802 was rated at 6.4 MHz, faster than any contemporary device. Even with the overhead of manipulating the data pointer, the 1802 delivered very good performance. The chip was used in a few kit computers and video games and, in a silicon-on-sapphire version, in several NASA space probes. Without a compiler, however, it was extremely difficult to program the chip.

### The 432: Intel's False Start

In contrast to the 1802, the 432 was in some ways the most complex microprocessor ever to reach the market. Intel originally called the project the 8800 and intended it to be the company's mainstream processor after the 8080. Designed to take advantage of the Ada programming language, the chip included operating-system support, such as scheduling and interprocess messaging, in hardware. All memory references were object oriented—that is, each load and store went through one or more levels of indirection, and all accesses were checked to ensure the data was of the correct type (character, integer, pointer, etc.).

As the name implies, the 432 was a full 32-bit processor. The programming model was a stack architecture: all data was kept on the stack, and instructions operated on the stack. Like the TMS 9900, this design avoided the need for a wide on-chip register file. The 432 had a huge number of instructions, many very complex, that were encoded in variable lengths ranging from 6 bits to an astonishing 321 bits. These instructions were left completely unaligned

in memory. The chip included full support for floating-point data and an on-chip FPU; this work led to the definition of the IEEE 754 standard for floating-point math.

The 432 CPU was part of a four-chip set that included a memory controller, I/O controller, and bus interface. These chips provided advanced features such as memory interleaving, ECC, fault tolerance, and multiprocessor support. The complete design, however, was far more expensive than competitors' microprocessors. Furthermore, the system was incredibly slow, due mainly to the lack of on-chip registers and the extensive time required to fetch and check each memory reference. Finally, the project was very late: initially expected to ship in 1978, the 432 processor finally hit the market in 1981, leaving barely a ripple.

### 8086 Rushed to Market

When Intel realized the 432 was in a deep hole, it rushed to staff a new project. The 8086 instruction set, the basis for all x86 processors in PCs today, was created by two engineers in only three weeks. Intel's first 16-bit processor, the 8086 had a transistor budget of about 30,000, giving the designers plenty of room to improve on the 8080 while retaining assembly-code compatibility.

The 8086 added several new features, including multiply and divide instructions. The A, B, C, and D registers from the 8080 were extended to 16 bits, giving the processor four 16-bit registers or eight 8-bit registers. The 8086 also had four 16-bit index registers, including the stack pointer, that replaced the old HL pair. Several instructions operated only on specific registers or implicit operands, making register allocation for the 8086 a challenging task. This weakness was only somewhat remedied in later x86 processors.

The designers wanted to push beyond the 64K address space of the 8080 but needed to retain compatibility with that chip's 16-bit addresses. This led to a segmented addressing model that can access up to 1M of memory, but only 64K at a time. Programmers dealing with objects larger than 64K had to add complicated code to juggle multiple segments. Although the 80386 implemented 32-bit addresses in 1985, for compatibility reasons segmentation continued to be the bane of programmers for another decade.

Although segmentation allowed some crude implementations of virtual addressing, the 8086 had no memory protection scheme and no concept of a page fault. These features were later added in the 80286 and 80386, respectively. Zilog's original Z8000, introduced in 1979, also did not handle page faults, but it supported two modes, user and protected, and could address up to 8M of memory with non-overlapping segments. A later version of the Z8000, released in 1982, was the first microprocessor with a paged memory-management unit and full virtual addressing.

## Motorola Pushes 68000 Forward

In a forward-looking move, Motorola created a new 32-bit instruction set first implemented in the 68000, which began shipping in 1979. There is some dispute as to whether the 68000 was actually a 32-bit processor; all the registers were 32 bits wide, but the ALU was only 16 bits wide, so most 32-bit operations took an extra clock cycle. Furthermore, the external data bus was only 16 bits, so 32-bit memory accesses took an extra bus cycle. (The Z8000 implemented a similar combination of 32-bit operations and 16-bit ALU). These differences were not visible to software, however, and the 68020 later implemented the same 32-bit operations without any cycle penalties.

The 68000 was the first microprocessor to reach the market that was built around a microengine. Instead of using a state machine fed by a PLA or other logic, instructions were executed step by step using sequences of microinstructions. This simplified designing and debugging the chip; although hardwired logic is generally more compact, burgeoning transistor budgets allowed this type of design.

Unlike the 8086, the 68000 had an orthogonal register set, eschewing special-purpose registers. With its 32-bit registers, the 68000 had no need for segmented addressing; it had a linear 24-bit address space. This design made the 68000 much easier to program than the 8086 and allowed it to access single arrays larger than 64K. The 68000 included user and supervisor modes but did not support page faults. This oversight was fixed in the 68010, which supported true virtual memory using an external MMU.

The 68000, Z8000, and 8086 were among the processors considered by IBM for its first PC. IBM eventually selected the 8088, a version of the 8086 with an 8-bit external bus. The narrow bus reduced system cost and leveraged existing 8080 and Z80 interface chips. This design win made the 8086 and its successors the dominant PC processors; the 68000 ended up in workstations, embedded applications, and Apple's personal computers.

## RISC Emerges from Laboratories

In the early 1980s, while Intel and Motorola focused on improving their memory management and moving to 32-bit ALUs, the next advance in microprocessor design was being cooked up by several research teams. The first RISC project is generally considered to be the 801, developed at IBM Research by John Cocke and others. This processor, which was never commercialized, established the general RISC paradigms of fixed-length instructions, large register files, three-operand instructions, no microcode, and single-cycle execution. The designers felt the tradeoff of increased code size was justified by the potential performance improvements.

Additional RISC development ensued at U.C. Berkeley and Stanford University, led by David Patterson and John Hennessy, respectively. As the results of this research proved promising, it moved into the commercial sphere. Much of Patterson's work was adopted by Sun in its SPARC architec-

## For More Information

Try two excellent articles direct from the people that invented the microprocessor: "Intel—Memories and the Microprocessor," by Gordon Moore, *Dædalus* (Journal of the American Academy of Arts and Sciences), Spring 1996; "The History of the Microcomputer—Invention and Evolution," by Stanley Mazor, *Proceedings of the IEEE*, December 1995.

ture. Hennessy and others founded MIPS. Hewlett-Packard hired several members of the 801 team to develop PA-RISC. The major microprocessor makers, however, held back in fear of jeopardizing their lucrative traditional processors, which became known as CISC designs.

RISC concepts go as far back as the Control Data 6600, designed by Seymour Cray in the late 1960s. Companies such as Pyramid and Ridge built proprietary RISC processors in the early 1980s. The MIPS R2000, which began shipping in systems in June 1986, was the first RISC microprocessor to reach the market, as Figure 2 shows. Within a year, it was joined by several others.

The R2000 gained several advantages from its RISC design. Simplifying the instruction set and forcing most instructions to execute in a single cycle made it relatively easy to pipeline the processor, overlapping the execution of several instructions at once. The initial R2000 operated at 8 MHz but, at its peak rate, completed an instruction every cycle. The contemporary 386, in contrast, used a 16-MHz clock but took at least four cycles to complete an instruction. On the Dhrystone benchmark, the R2000 outperformed the 386 by roughly 50%.

The Stanford project defined MIPS as a "microprocessor without interlocked pipeline stages," which left software the burden of inserting other instructions before using the results of long-latency calculations. To simplify software, the R2000 added interlocks for all instructions except loads, which had a single delay cycle; the R4000 eventually added these interlocks as well. Other RISC processors debuted with fully interlocked pipelines. Within a few years, CISC chips such as the 486 and 68040 would adopt pipelining.

Compared with the diverse pioneers of the 1970s, the first RISC devices were remarkably similar, although their proponents argued fiercely over details such as the number of addressing modes and lack of integer divide instructions. SPARC and AMD's 29000 were perhaps the most deviant with their large windowed register files. Acorn's ARM chip included predicated execution; each instruction could be executed conditionally based on a set of condition flags. HP's PA-RISC and IBM's later POWER architecture initially focused on multichip implementations and thus included more complex instructions in their designs.

Many of the RISC vendors gained initial success in the

workstation market, which required hardware floating-point implementations. As early as the 8087, mainstream microprocessor vendors had offered add-on FPU chips (as known as math coprocessors) for their CPUs. These vendors were slow to integrate the CPU and FPU on a single chip, however, because the FPU would have added cost but no value for most of their customers. Except for the ill-fated i432, RISC processors were the first to offer an on-chip FPU, starting with the Inmos T800 Transputer in 1986. Within the next few years, many other microprocessors followed suit.

### Intel Does RISC, Too

While pooh-poohing RISC as “the last hope of the have-nots,” Intel realized the new guys might have found a good idea. The i960KB, which shipped in 1988, barely qualified as a RISC: its variable-length and multicycle instructions handled by microcode violated basic RISC tenets. A more traditional RISC design, the i860, followed in mid-1989.

This chip was capable of executing two instructions per cycle, a first among microprocessors. The i860 is not considered superscalar, however, because the dual-issue capability required putting the chip into a special mode and carefully aligning the instructions, making it more of a VLIW processor. Because of these restrictions, this mode was rarely used. The i860 set performance records upon its release but garnered few CPU design wins, although it appeared in several specialized graphics accelerators.

### Today's Superscalar Out-of-Order Chips

Intel's i960CA, which appeared late in 1989, was the first truly superscalar microprocessor, able to pair two instructions on the fly. This technique has become widely used in the 1990s, with state-of-the-art processors today executing up to four instructions per cycle.

To feed these wide-issue machines, out-of-order execution has become popular. Released in 1993, the PowerPC 601, codeveloped by IBM and Motorola, was the first microprocessor to issue and execute instructions even if a preceding instruction was blocked due to a register dependency or other conflict. The follow-on PowerPC 603, which shipped in 1994, implemented a more generalized method of out-of-order execution that has since been adopted in most high-performance microprocessors.

Wide issue rates have also placed a premium on accurately predicting branches. Intel's Pentium, in 1993, was the first processor to incorporate dynamic branch prediction. In 1995, NexGen's Nx586 became the first chip to implement a more advanced two-level prediction algorithm (see [090405.PDF](#)).

Cache architectures have evolved since 1984, when the 68020 and the Z80,000 became the first microprocessors with small (<1K) on-chip caches. The R2000 typically used a relatively large (64K) off-chip cache and included the necessary control logic. In 1989, Intel's 486 and i860 chips included sizable (8K) on-chip caches. By 1992, the R4000 featured not only 16K of on-chip cache but the control logic for an external level-two cache. Today's Alpha 21164, with more transistors than any other processor in production, includes both level-one and level-two cache on chip as well as the control mechanisms for an external level-three cache.

This trend is typical of microprocessor history, where the processor chip has consumed FPU, MMU, cache, and bus-interface circuits that previously were implemented externally. This trend may continue with main-memory control and graphics acceleration subsumed into the main processor in the future. History shows that as transistor counts increase, microprocessor designers always find ways to take advantage of them. 