◆ The History of the Microprocessor

Michael R. Betker, John S. Fernando, and Shaun P. Whalen

Invented in 1971, the microprocessor evolved from the inventions of the transistor (1947) and the integrated circuit (1958). Essentially a computer on a chip, it is the most advanced application of the transistor. The influence of the microprocessor today is well known, but in 1971 the effect the microprocessor would have on everyday life was a vision beyond even those who created it. This paper presents the history of the microprocessor in the context of the technology and applications that drove its continued advancements.

Introduction

The microprocessor, which evolved from the inventions of the transistor and the integrated circuit (IC), is today an icon of the information age. The pervasiveness of the microprocessor in this age goes far beyond the wildest imagination at the time of the first microprocessor. From the fastest computers to the simplest toys, the microprocessor continues to find new applications.

The microprocessor today represents the most complex application of the transistor, with well over 10 million transistors on some of the most powerful microprocessors. In fact, throughout its history, the microprocessor has always pushed the technology of the day. The desire for ever-increasing performance has led to the rapid improvements in technology that have enabled more complex microprocessors. Advances in IC fabrication processes, computer architecture, and design methodologies have all been required to create the microprocessor of today.

As we trace the history of the microprocessor, we will explore its evolution and the driving forces behind this evolution. In the earliest stages, microprocessors filled the needs of embedded applications. It was not long, however, before advances in microprocessors and computers drove the capabilities and needs of both. We will discuss these and other forces behind the history of the microprocessor, including the impact of individuals and companies.

The history of the microprocessor can be divided into five stages:

- The birth of the microprocessor,
- The first microcomputers,
- A leading role for the microprocessor,
- The promise of reduced instruction set computer (RISC), and
- Microprocessors of the 1990s.

These five stages define a rough chronology, with some overlap. Each stage could be said to reflect a generation of microprocessors, with corresponding generations of applications. For each stage, we discuss representative microprocessors and their key applications. **Figure 1** shows a timeline of the development of the microprocessor, starting with the Intel* 4004.

The information in this paper was taken from many sources, including other overviews of the history of the microprocessor.[1,2,3,4] We have selected the microprocessors discussed in this paper based on their innovation and their success in the marketplace. Embedded processors are given limited coverage since, in many cases, the microprocessors mentioned in more detail have led to versions for embedded applications. We have not covered digital signal processors (DSPs), even though they could be considered a type of microprocessor. However, we have included in the appendix of the paper a history of microprocessors at Bell Labs, which has designed microprocessors since the latter half of the 1970s.

The Birth of the Microprocessor

"Announcing a New Era of Integrated Electronics"
—Headline, Intel 4004 ad

The history of the microprocessor begins with the birth of the Intel 4004, the first commercially available microprocessor (see **Panel 2**). The roots of this development can be traced directly back to the inventors of the transistor. In 1955, William Shockley founded Shockley Semiconductor in Palo Alto, California (arguably the birth of Silicon Valley). This company eventually employed Gordon Moore and Robert Noyce, who left with others to form Fairchild Semiconductor in 1957. While at Fairchild, Noyce played a significant role in the development of the IC, first commercially available in 1961. In 1968, Moore and Noyce left Fairchild to form Intel Corporation. Intel's focus at that time was the development of memory chips, but Intel's history was forever changed by the events leading to the development of the 4004 for the Busicom calculator company. The first fully functional 4004 parts were available in March 1971, with the first public announcement in November 1971.

Around the same time Intel developers began working on the 4004, they also began work on the 1201 project for Computer Terminal Corporation (CTC). The 1201 was intended to be a single metal-oxide semiconductor (MOS) chip that would replace a similar processor designed using medium-scale-integration components. The 1201 was later renamed the Intel 8008. The 8008 was the first 8-bit microprocessor and laid the foundation for future microprocessors from Intel. The 8008 was designed in 10-micron PMOS (metal-oxide semiconductor using p-type transistors) technology, and required approximately 3,500 transistors. The die for the 8008 measured 4.9 mm × 6.7 mm. The 8008 was packaged in an 18-pin dual inline package, ran at 200 kHz, and was capable of 60,000 instructions per second.

While the 8008 was being developed, a June 1971 Texas Instruments (TI) advertisement in *Electronics* magazine showing a "Computer On A Chip" revealed that CTC had also contracted with TI to produce a chip similar to the 8008. This presented a difficult situation for Intel, which had not yet announced the 4004 and presumed it was ahead of the competition. As it

---

Panel 1. Acronyms, Abbreviations, and Terms

ARM—Advanced RISC Machines
BiCMOS—bipolar complementary metal-oxide semiconductor
BIOS—basic input/output system
BIU—bus interface unit
CISC—complex instruction set computer
CMOS—complementary metal-oxide semiconductor (with n- and p-type transistors)
CPI—cycles per instruction
CP/M—control program/monitor
CPP—communications protocol processor
CPU—central processing unit
CTC—Computer Terminal Corporation
DEC—Digital Equipment Corporation
DMA—direct memory access
DRAM—dynamic random access memory
DSP—digital signal processor
EU—execution unit
FPU—floating-point unit
GaAs—gallium arsenide
GUI—graphical user interface
IC—integrated circuit
IEEE—Institute of Electrical and Electronics Engineers
I/O—input/output
MIPS—millions of instructions per second
MIPS—microprocessor without interlocking pipe stages
MMU—memory management unit
MOS—metal-oxide semiconductor
MPEG—Motion Picture Experts Group
MSI—medium-scale integration
NMOS—MOS with n-type transistors
OS—operating system
PC—personal computer
PMOS—MOS with p-type transistors
RAM—random access memory
RISC—reduced instruction set computer
ROM—read only memory
SC/MP—single-chip microprocessor
SCP—Seattle Computer Products
SPICE—simulation program integrated circuit emphasis
SRAM—static random access memory
TI—Texas Instruments
VLIW—very long instruction word
VLSI—very large scale integration
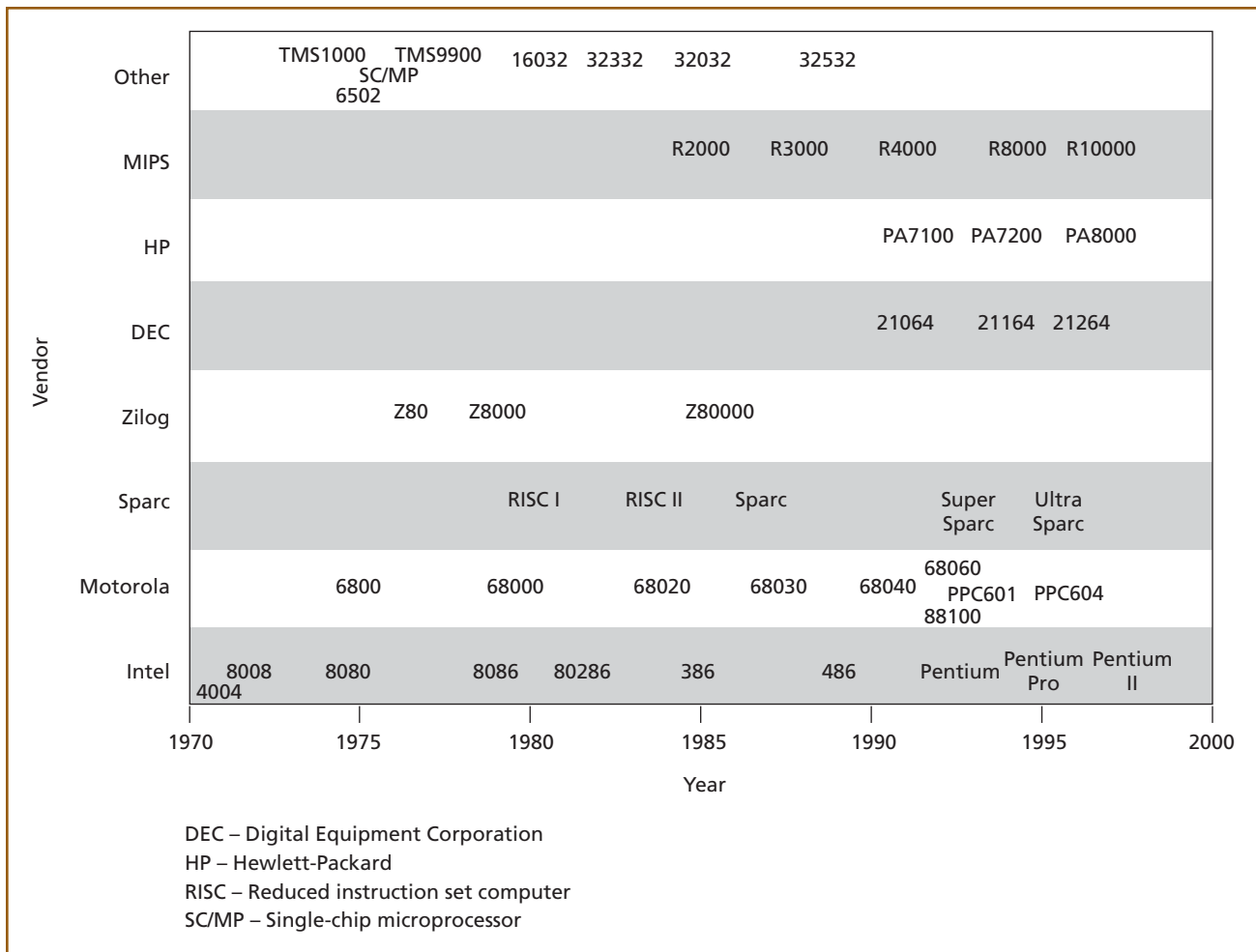
Figure 1.
Microprocessor timeline.

turned out, the TI chip was not operational. TI dropped the project when CTC decided not to use either the 8008 or the TI chip.

The architecture of the 8008 was based on the existing CTC processor and had a single 8-bit accumulator (A), along with six general-purpose 8-bit registers (B, C, D, E, H, and L). It supported a 14-bit address and included logical operations and interrupts. The 8008 was designed to interface with standard memory chips. Information on the 8008 was publicly available as early as December 1971, followed by the official introduction in April 1972.

A significant result of TI's efforts was a 1971 patent application,[5] which in 1978 resulted in the first patent issue covering a microprocessor. Intel never applied for a patent covering the microproces-

sor. In 1969, prior to either TI's or Intel's microprocessor efforts, an engineer named Gilbert Hyatt filed for a patent[6] that covered a computer on a single integrated chip. Twenty-one years later, when the patent was finally awarded, it would cause a great deal of turmoil and legal action.

In Search of Applications

The first commercially available microprocessors, the Intel 4004 and 8008, were developed with specific applications in mind. The 4004 was intended for an electronic calculator, and the 8008 was designed for a computer terminal. They were intended to replace a number of smaller devices wired together to perform the desired function. Beyond their original applications, it was unclear what the market was for these first microprocessors.

Panel 2. Intel 4004, The Birth of an Age[19,20]

Bob Noyce and Gordon Moore left Fairchild Semiconductor Corporation in 1968 and founded Intel Corporation for the express purpose of producing proprietary memory products. However, as in most start-up companies, there was a desire, for cash flow reasons, to do a certain amount of custom work. It was thought that custom products would ramp up to volume production faster than would proprietary products.

In April 1969, Busicom, a Japanese manufacturer, approached Intel with a need for a metal-oxide semiconductor (MOS) engine for its printing calculator products. A family of products using read-only memory (ROM)-programmable variations of the basic calculator design was in view. Ted Hoff, a new Intel employee with badge number 12, was assigned to act as liaison to the Busicom engineers. Busicom sent three engineers to Intel to finalize the logic design of the calculator chip set and transfer the design to Intel. Although Hoff was supposed to act only as liaison to the Busicom team, his curiosity led him to study their design.

Hoff was amazed at the complexity and I/O requirements of the proposed design and became concerned that the project's cost objectives could never be met. When he explained his concerns to Intel management, he was encouraged to pursue an alternative design.

Hoff began to consider the design of a general-purpose computer that would be programmed to perform calculator functions. Hoff's vision was of a computer that would fetch instructions from ROM into an arithmetic chip. The arithmetic chip, using local registers, would interpret the instructions, reading and writing to dynamic random access memory (DRAM) as necessary. (At this time Intel was developing the first DRAM.) While the arithmetic chip was fetching instructions, the DRAM would be refreshed.

In September of 1969, Stanley Mazor joined Intel from Fairchild and progress on the architecture accelerated. At this time, Intel marketing was sufficiently confident of the design to present it to Busicom as a superior alternative to their original approach. The Busicom managers saw the advantages and by October an agreement was reached to build the proposed Intel chip set.

Intel was now committed, but neither Hoff nor Mazor had ever designed chips and they realized that the complexity of these chips would require someone with extensive experience. The design languished for three months, with the customer getting increasingly concerned about the schedule. Early in 1970, Leslie Vadasz, who headed Intel's MOS design group, announced that he had found someone to design the calculator chip set, Federico Faggin.

Faggin joined Intel in April of 1970 to take on the design of one of the most complex chip sets attempted to date. The project was behind schedule and the Busicom engineer, Masatoshi Shima, was disappointed. He felt strongly that the program schedule and product introduction were hopelessly compromised by Intel's slow start. However, Shima stayed at Intel for the next six months to assist Faggin with the project.

After resolving the remaining architectural details, Faggin laid down the design methodology to be used, based on Intel's silicon gate process. An important element in the methodology was the use of bootstrap loads, which were fast and allowed switching to the full supply voltage. This approach further allowed the use of simple pass transistors, thereby reducing the transistor count needed to perform the logic.

The chip set consisted of four chip types: the 4001 ROM, the 4002 random access memory (RAM) register memory, the 4003 I/O shift register, and the 4004 central processing unit (CPU). Faggin decided to design the 4001 first, followed by the 4003, the 4002, and the 4004 last. There was very little design automation in those days. Graphical analysis was based on static and dynamic device characteristics. These characteristics were usually based on measurements from the most recent process runs. A slide rule was used for most calculations.

At the peak of the design effort, Faggin and Shima worked simultaneously on all four chips in different stages of their development. The first 4001 wafers were processed in October of 1970 and were fully functional from the start. One

month later, the 4002 and 4003 wafers were tested, with the 4002 needing only minor changes. When the first 4004s were tested in December, they found that a process step had been omitted and the chips did not work. New 4004 wafers were rushed through processing and by January of 1971, they were under test. Two minor bugs necessitated a mask change and the next iteration in March yielded fully functional CPU chips. While all this was happening, Shima returned to Japan to prepare the rest of the prototype calculator for the first chips. By April of 1971, the software was complete and the Busicom calculator was a fully functional product. Production ramp-up was rapid and they began shipping calculators by July. The only portions of the calculator system that were not part of the Intel chip set were the printer driver circuit and the clock generator.

At this point, the design belonged exclusively to Busicom. However, Faggin and Hoff were convinced that the chip set had commercial value beyond the Busicom sales. Unfolding events would have a way of solving this problem because Busicom found itself in business difficulties. Faggin and Hoff pleaded with Intel marketing to offer a price concession to Busicom in exchange for the right to market the chip set to companies not in the calculator business.

By May of 1971, Intel had negotiated the right to sell the chip set to non-calculator manufacturers. Initially Intel marketing was reluctant to push the MCS*-4 (as it was then called for "Micro Computer System 4-bit") for fear of not being able to provide customer support on such a complex product. To correct this, Hoff, Faggin, Mazor, and Hal Feeney worked on support. Data sheets, application information, a programmer's manual, and a printed circuit board were developed to support sales. The issue of good product support was later to be a hallmark of the Intel processor and microcontroller product line.
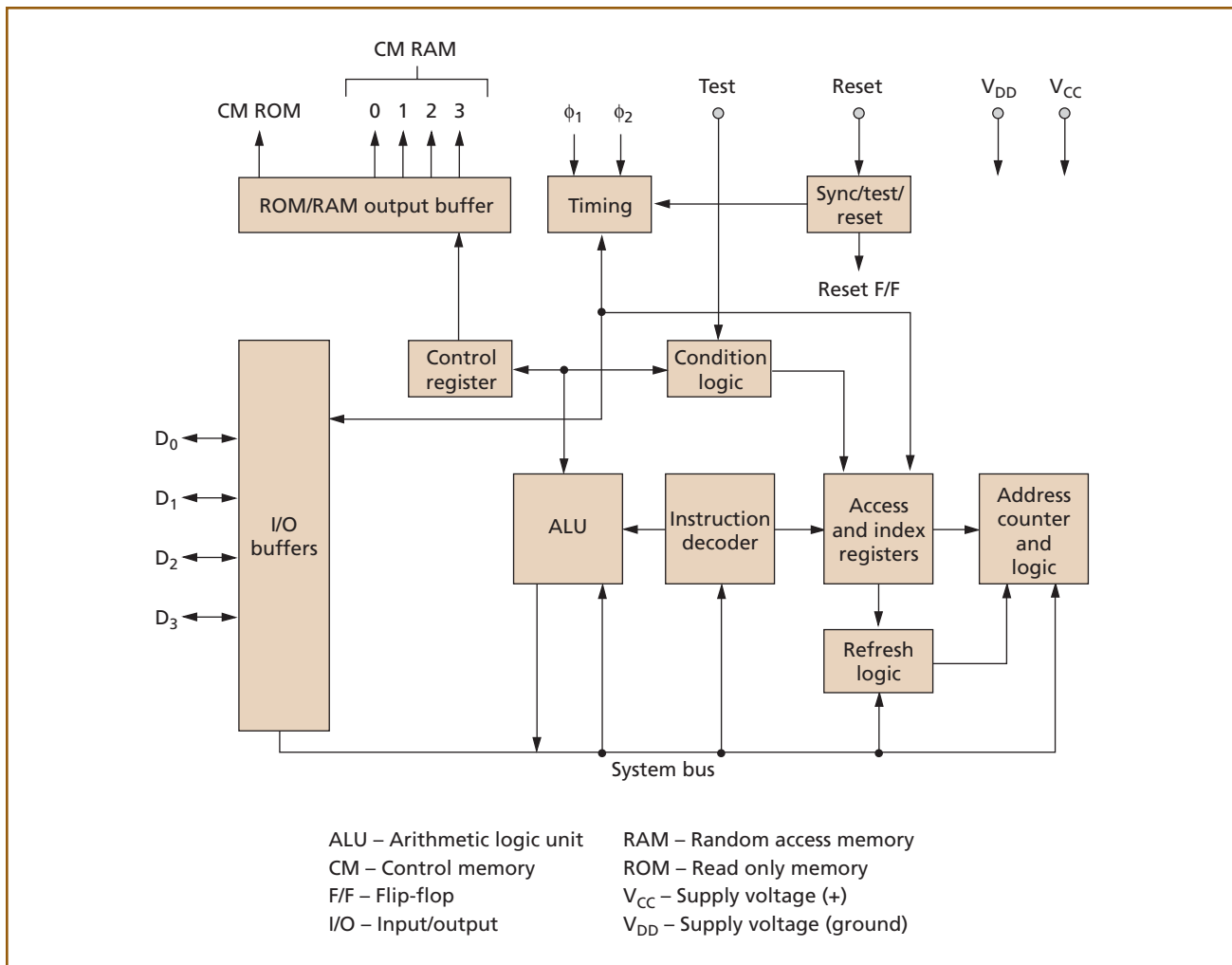
Figure 2 shows a block diagram of the 4004 CPU and **Figure 3** shows a 4004 system containing typical quantities of all four chips. The initial 4004 CPU chip measured 3.0 x 4.0 millimeters, used 2,300 transistors, and was supplied in a 16-pin dual inline package. The entire circuit was laid out by hand using a Rubylith* process. Each Rubylith layer was then photo-reduced by a factor of ten to the actual size of the 4004. A photographic step-and-repeat process was used to make the photo mask for device fabrication. Only six masks were required to define the 4004. The other three chips in the set used a five-mask process. Today, if the 4004 were built using a 0.35-micron process, it would be tenths of a square millimeter in area (without wire bond pads) and cost less than one cent to fabricate.

The calculators of the early 1970s were the most advanced form of computing available to the masses, costing hundreds of dollars. The closest general-purpose computer, the minicomputer, cost several tens of thousands of dollars at the time. The calculator received a huge amount of coverage in the press and, over time, created a revolution of its own, eventually replacing the engineer's trademark slide rule. With increasing demand came competition, which created constant pressure to reduce cost. Given this situation, it is obvious why the calculator market would require the eventual cost and size advantages of the microprocessor.

The question at that time was whether a hard-wired or general-purpose approach provided the best solution for the advance of the calculator. Implementations requiring fewer and fewer chips eventually led to a calculator on a chip and, as we have seen, the first commercial microprocessor. The question still remained—What were the other possible applications of the microprocessor?

The impact of the 4004 at the time was actually quite small, with little press attention. The 4004 and 8008 microprocessors, along with Intel's push to market the new invention, were greeted with little fanfare well into 1972. Few chips were actually being sold at first, with more interest in the design tools and test boards being offered. Intel's efforts to generate interest in its new chips were initially met with skepticism. Many thought the applications of the microprocessor were limited to a few niche areas. They did not see the potential of the microprocessor to revolutionize com-

CM ROM, CM RAM 0 1 2 3, φ₁ φ₂, Test, Reset, V_DD, V_CC

ROM/RAM output buffer — Timing — Sync/test/reset

Reset F/F

Control register — Condition logic

D₀, D₁, D₂, D₃ — I/O buffers

ALU — Instruction decoder — Access and index registers — Address counter and logic

Refresh logic

System bus

ALU – Arithmetic logic unit   RAM – Random access memory
CM – Control memory   ROM – Read only memory
F/F – Flip-flop   V_CC – Supply voltage (+)
I/O – Input/output   V_DD – Supply voltage (ground)

Figure 2.
Block diagram of the 4004.

puting. Through extensive marketing and publicity, interest in the microprocessor grew. Articles in trade and technical publications started to appear in the middle of 1972, with coverage of the microprocessor becoming commonplace in 1973. In a short time, the microprocessor had gone from an interesting technology to one that would change the way engineers design electronic products and systems. The promise of the microprocessor was now recognized. The next step was to start to fulfill this promise.
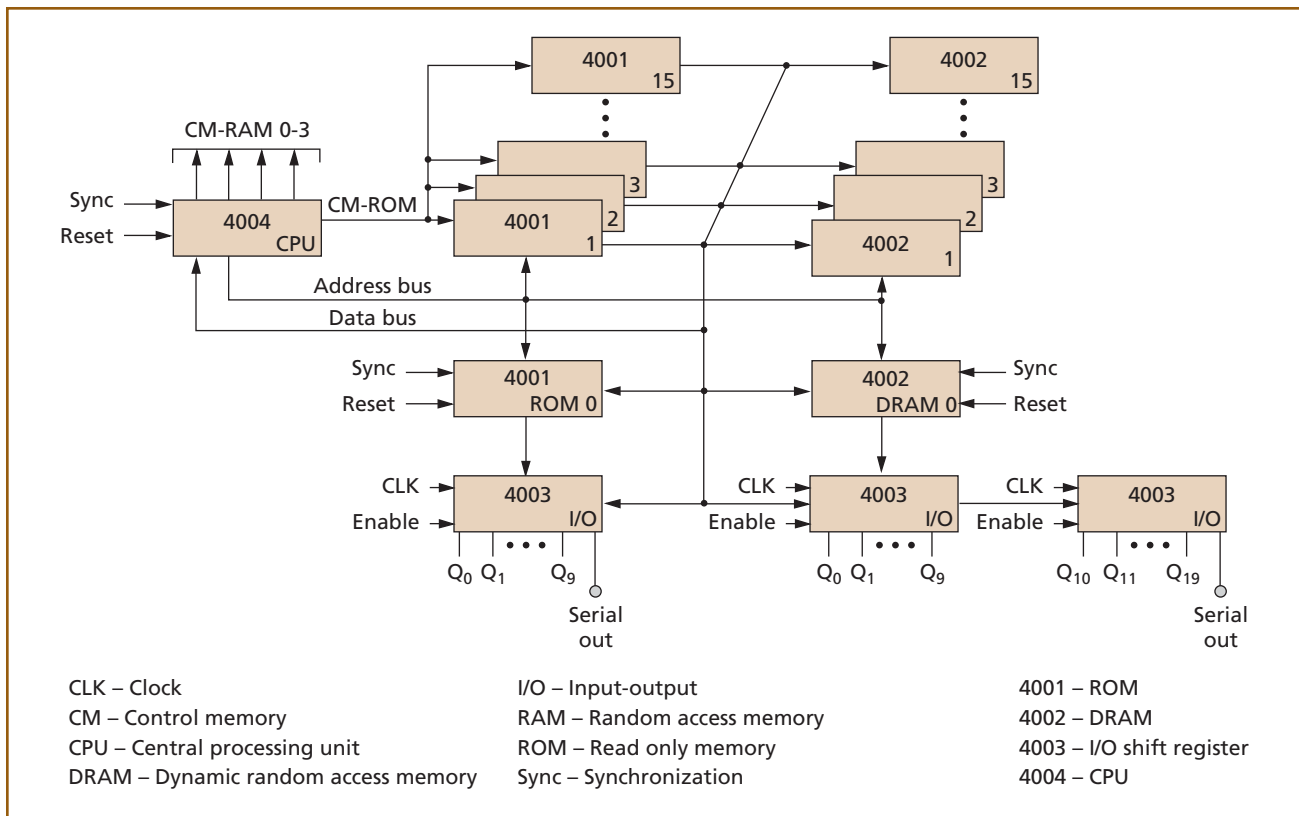
The First Microcomputers

"Project Breakthrough! World's First Minicomputer Kit"
—*Popular Electronics* cover, January 1975

The introduction of the Intel 4004 and 8008 demonstrated the possibility of putting an entire cen-

tral processing unit (CPU) on a chip, but it was not until the next generation of processors that a true microprocessor market was realized.

The initial applications for the microprocessor were mostly embedded applications. The application that would ultimately drive the continued advances in microprocessors was the microcomputer. The 8008 was used in a variety of microcomputer kits, as well as pre-assembled systems. The first microprocessor-based pre-assembled computer was the Micral, built in France using the 8008. Another early microcomputer was the Scelbi-8H,* also using the Intel 8008, which was available in kit and non-kit form. These computers were not very successful, but they did show the potential of the microprocessor.

Figure 3.
Typical 4004 system.

The following years would see a series of microprocessors that powered the first microcomputers to gain widespread acceptance.

The experience of the initial microprocessors and the continued advances in IC technology led to the development of more advanced chips. Among the next generation of chips were the first microcontroller and a series of more advanced 8-bit microprocessors from numerous companies.

TI's TMS1000, the First Microcontroller

The first commercially available microprocessor-based product from TI, the TMS1000, was introduced in late 1972.[7] The TMS1000 was the first microcontroller, integrating a simple 4-bit microprocessor, 1K read only memory (ROM), and 32-byte random access memory (RAM) on a single chip. This chip was inexpensive and saw numerous applications in embedded systems. An important application within TI was the Silent 700* series of terminals.

Intel's 8080 and the Altair

Intel's experience with the 8008 provided a tremendous source of ideas on how to improve on the microprocessor. Starting in the middle of 1972, these ideas were used to define the Intel 8080 microprocessor. The improvements in the 8080 included more instructions, a 64-KB address space, 256 I/O ports, 16-bit arithmetic instructions, and vectored interrupts. The designers of the 8080 included some of the key individuals responsible for the 4004 and 8008, Federico Faggin and Masatoshi Shima. The 8080 was introduced in early 1974 with a price tag of $360. The 8080 was designed in 6-micron MOS with n-type transistor (NMOS) technology and required 6,000 transistors. The 40-pin package allowed for separate address and data buses. The first 8080 ran at 2 MHz and was rated at 0.64 millions of instructions per second (MIPS).

Unlike the 4004 and 8008, the 8080 was quickly adopted by designers. It was incorporated into numerous products, the most significant being the Altair

8800* microcomputer kit from a company called MITS. First advertised in the January 1975 edition of *Popular Electronics,* the Altair 8800 offered an affordable "personal computer," or PC. The quick popularity of the Altair spurred interest in microcomputers and what one could do with them. Clubs such as the Homebrew Computer Club in California and the Amateur Computer Group of New Jersey were formed at the same time. The Altair showed there was a market for microprocessors beyond traditional embedded applications.

Motorola's 6800

Motorola entered the microprocessor market in 1974 with the 8-bit 6800. The 6800 required 4,000 transistors and was fabricated in NMOS technology. The 6800 offered some significant benefits over the 8080, including improved performance and the need for only a single 5-volt supply. The 6800 contained two 8-bit general-purpose registers and a single index register, which meant that it operated on data primarily in memory. Because the memory technology at the time was faster than the microprocessor, accessing memory did not impose a performance penalty.

The 6800 saw limited use in the microcomputers of the day, although in 1976 MITS did offer a 6800 version of its microcomputer, the Altair 6800.* The most significant application of the 6800 was initially the automotive market. Motorola first produced a custom version of the 6800 for General Motors and later for Ford. This was the beginning of a huge market for embedded processors in cars, which Motorola has since dominated. Variants of the 6800 have been introduced over the years, including the 6809 in 1977, the 6801, the 68HC11, and the 68HC16.

The Competition Heats Up

The 8080 and the 6800 provided excellent examples of the state of the art of microprocessors in the mid-1970s, but they were in some way surpassed by the continued work of some of their creators. Chuck Peddle left Motorola to join MOS Technologies, which would produce the 6502. Faggin and Shima left Intel in 1975 to form Zilog, which would produce the Z80. The 6502 and Z80 would become the microprocessors that powered the first microcomputers to reach beyond the hobbyist.

MOS Technologies' 6502, released in 1975, was loosely based on the 6800. The 6502 supported a 16-bit address bus and contained one 8-bit general-purpose register, two 8-bit index registers, and an 8-bit stack pointer. The most significant feature of the 6502 when it was introduced was its price. While a microprocessor such as the 8080 cost about $150 at the time, the 6502 was available for about $25. The low cost led to its use in microcomputers such as the Apple* II and Commodore PET. Variations of the original 6502 were also used in the Commodore 64, Atari 2600, the Nintendo Entertainment System* (NES), and the Super NES.*

The 2.5-MHz Zilog Z80 was released in 1976 and offered compatibility with the 8080, along with many significant enhancements. The instruction set was expanded and included block move and block I/O instructions. A second register set was added to better support interrupts and operating systems (OSs). The Z80 interface simplified the system design by providing dynamic random access memory (DRAM) refresh signals and an on-chip clock circuit, which could be connected directly to an external crystal. **Figure 4** shows a block diagram of the Z80.[8]

The Z80 would outsell the 8080 as it became the microprocessor of choice in many applications. The most significant microcomputer application, the Tandy TRS-80, was introduced in 1977. The TRS-80 contained a Z80, 4-KB RAM, 4-KB ROM, a keyboard, a black and white video display, and a tape cassette, all for $600. Thousands were sold in the first few months, exceeding all projections. To this day, the Z80 continues to be a popular microprocessor in embedded applications.

The Apple II, introduced at the First West Coast Computer Fair in April 1977, provided the next big leap in capability for the microcomputer. The Apple II included a 6502 microprocessor, 4-KB RAM, 16-KB ROM, a keyboard, an eight-slot motherboard, game paddles, built-in BASIC, and a graphics/text interface to a color display. The Apple II saw great success from the start, but it did not penetrate into wider markets until the introduction in 1979 of the "killer app" VisiCalc,* the first spreadsheet program. The combination of the
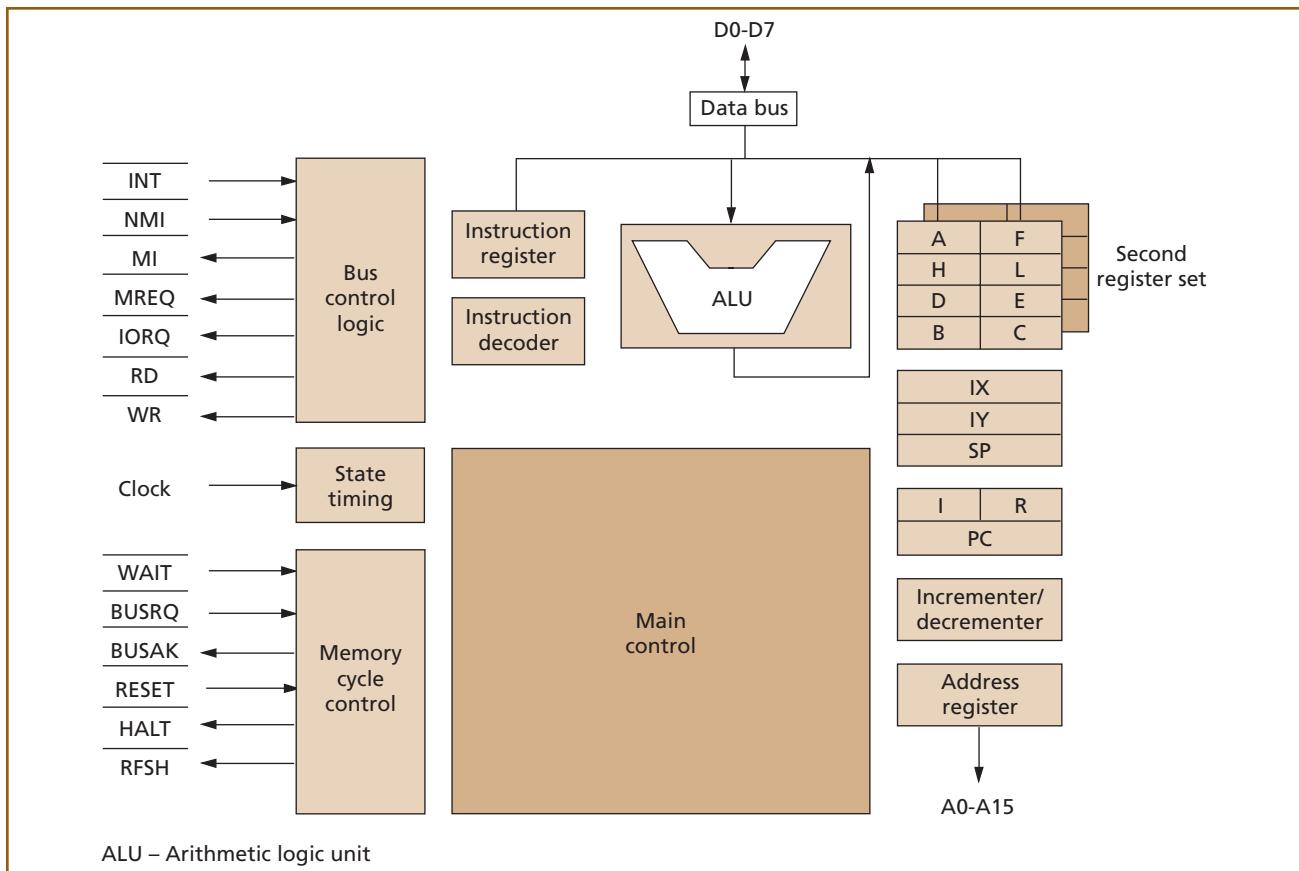
Figure 4.
Block diagram of the Z80.

Apple II and VisiCalc created a compelling reason for businesses to take notice. One of those businesses would be IBM.

Other Noteworthy Microprocessors

The 8-bit RCA 1802, introduced in 1974, was one of the first microprocessors designed using complementary MOS (CMOS) technology. The 1802 ran at 6.4 MHz with a 10-volt supply, making it one of the fastest microprocessors of its time. Its simple design included sixteen 16-bit registers, which were also usable as thirty-two 8-bit registers. It used an 8-bit opcode to implement the limited instruction set. The most significant applications of the 1802 were in several NASA space probes. It was used in those cases because a version that used the radiation-resistant silicon-on-sapphire technology was available.

The 8-bit National Semiconductor single-chip microprocessor (SC/MP), introduced in 1976, was the first microprocessor to support multiple bus masters on its system bus. This feature supported multiple SC/MPs and other bus masters, such as a direct memory access (DMA) controller. Arbitration was controlled by a "daisy chain" connecting the bus masters in priority order. The ENOUT (enable out) and ENIN (enable in) signals of the SC/MP were used to chain the processors together. Another unique feature of the SC/MP was its bit serial arithmetic logic unit (ALU).

The 16-bit TI TMS9900, introduced in 1976, was the first single-chip 16-bit microprocessor. Its architecture was based on the TI 990 minicomputer. The TMS9900 had only two 16-bit internal registers, with one of them pointing to the memory-resident register set. The speed of memory at the time made it feasible to use external memory for the register set. A simple adjustment of the internal register could be used to save the registers for a procedure call or interrupt. A version of the TMS9900, the TMS9940, was used in

the TI 99/4 PC, introduced in 1979.

A Leading Role for the Microprocessor

"Now, a computer on every desk, ..."
    —*Wall Street Journal,* August 1981 (IBM PC Introduction)

The early to mid-1980s marked the period when microprocessors, through desktop systems, came to be known to a wider public than the microcomputer hobbyists and embedded system developers. Desktop systems such as PCs and workstations prominently featured their microprocessors. The microcontrollers contained in a myriad of embedded applications were largely anonymous. This period saw a shakeout in the microprocessor industry. Critical markets, such as the PC market, quickly established dominant vendors. However, by the end of this period, new processor architectures were challenging the established players. Significant developments in OSs and software, which would greatly change the microprocessor landscape in the future, occurred at this time.

By the late 1970s, many of the early microprocessors were already fading from the center stage. Many semiconductor manufacturers had developed 4-bit and 8-bit microprocessors. Many of these devices were profitable in embedded applications (see **Panel 3**), but none had the impact of later 16-bit devices from Intel and Motorola. Early embedded applications such as watches and calculators offered ever-decreasing profits as these markets matured. A recession from 1981 to 1984 did not help either, forcing retrenchment by most large and small microprocessor vendors. The rise of desktop computers offered a market that, like embedded applications, consumed high volumes, but also offered high profit margins.

The development of the 16-bit Intel 8086 (and its relative, the 8088) and the 16/32-bit Motorola 68000 catalyzed the growth of the microprocessor industry. As so often happens in the semiconductor world, critical markets make or break a microprocessor. The 8088 and 68000 were not the first microprocessors to benefit from this phenomenon. However, the desktop computer market differed in significant ways from earlier microprocessor applications. The primary requirement for embedded applications such as calculators and watches was low cost. Because the customer was oblivious to the identity of the microprocessor in these products, the system maker could choose the lowest-cost vendor, thereby eliminating the possibility of high profit margins for the microprocessor vendor. Desktop computers introduced end customers to software and the notion of compatibility. As soon as end customers had invested in a library of software, the identity of the microprocessor (and OS) in their system became all too important. Once the end customer was wedded to a particular microprocessor, the profit margin in the vendor chain accrued primarily to the microprocessor manufacturer and the OS vendor.

Desktop Market Emphasizes Price and Performance Over Elegance

The desktop computer market also required ever-increasing microprocessor performance. Embedded applications tended to use a processor no more powerful than absolutely necessary. This was appropriate for a fixed-function appliance with little or no upgrade capability.

The situation in the desktop market was quite different. The desktop computer was a general-purpose device for running application software. The vendors of this software would have a poor business model if end users were to buy only one copy of the application. By introducing successive versions with more features (and bug fixes), the software industry drove end users to demand more performance. Thus, unlike the embedded space, the desktop market demanded a never-ending stream of higher-performance microprocessors. Vendors supplying the desktop parts could, in turn, demand premium prices for the latest introduction.

Even though the desktop market placed an emphasis on technology more than previous embedded applications, the microprocessor with the best technology was not necessarily the marketplace winner. The classic illustration of this phenomenon was the Intel 8086 and the Motorola 68000. Although the 68000 is widely regarded as a better example of computer architecture, it did not have the success of the 8086 in the desktop market. In fairness, it was not apparent in the early to mid-1980s that the x86 family had won the desktop architecture wars. However, it is significant that Intel was able to per-

Panel 3. Embedded Microprocessors

Although the media spotlight shines most brightly on desktop microprocessors, the workhorses and volume leaders by an overwhelming margin are embedded microprocessors. Embedded microprocessors find use in all manner of appliances, automobiles, consumer products, and even in the subsystems (such as keyboards and disk drives) of desktop computers. At present, the 64-bit and 32-bit microprocessors hold most of the mind share, but the bulk of the embedded processor market is made up of 4-bit, 8-bit, and 16-bit devices, in that order.

Intel's 4004, the first microprocessor, was an embedded microprocessor. Many early microprocessors were designed for watch or calculator applications. As the level of integration increased, more elements of the embedded system were integrated on chip with the microprocessor. This gave rise to the microcontroller: incorporating the central processing unit (CPU), read only memory (ROM), random access memory (RAM), and peripheral devices on one chip.

The Texas Instruments TMS 1000 was the first microcontroller, integrating 32 bytes of RAM, a 1-KB ROM, a clock, and I/O support on one chip. Intel's first microcontroller device was the 8048, followed by the 8051, which used two-byte instructions rather than the single byte of the 8048. The 8051 was unique in its ability to address practically any register or memory address at the bit level. Licensed widely, the 8051 is one of the most successful microcontrollers.

The 8096 was the 16-bit successor to the 8048. Intel later came out with the i860 and 80960. The i860 incorporated several innovative features such as an early version of dual-instruction issue. It found some applications as a graphics accelerator, but its programming complexity inhibited wider popularity. The 80960 has been one of the highest-volume 32-bit microcontrollers until overtaken by more recent video game processors. It found applications in printers and network equipment and was

one of the first true superscalar microprocessors, with the CA version introduced in 1989.

Motorola entered the embedded market early when it was approached by General Motors for an engine controller. The resulting 6800 in 1974 started a long line of successful 8-bit products for the automotive market, particularly the 6805 and 68HC11. The 68000 was also extensively used in higher-performance embedded applications such as telecommunications. Motorola was one of the first successful core-based vendors. With its inter-module bus and the 68000 core, Motorola produced many devices (most notably its 683xx series) with varying complements of peripherals.

Many reduced instruction set computer (RISC) vendors have introduced variants targeted to the embedded market. The Advanced RISC Machines (ARM) architecture was one of the first commercial RISC architectures. It is notable in being offered for most of its history by a vendor that is neither a system maker nor a semiconductor manufacturer. The ARM architecture was one of the first RISCs to incorporate conditional execution. The SPARC core is an example of a workstation RISC that has been widely licensed for use in the embedded market. In some cases, these embedded versions far outpace the volume of their desktop cousins.

Versions of the MIPS architecture have been used in the Sony PlayStation* and Nintendo 64 game systems. Other RISC architectures, such as the Hitachi SH family, have been catapulted to the top spot (for a time) because of their incorporation into a single high-volume product, such as Sega's Saturn* game system. The volume of the video game system market has introduced new pressure on microprocessor architectures. The Hitachi SH-4 incorporates floating-point performance seldom seen outside the engineering workstation or supercomputing market, in the quest for the most realistic three-dimensional gaming experience for the world's youth.

suade IBM to adopt the 8088 in spite of its technical deficiencies. It is largely accepted that Intel achieved this with superior marketing.

Intel's "Operation CRUSH" emphasized better customer support, documentation, and development tools for its processors.[3] Furthermore, the 8088 enabled the use of a wide library of 8-bit peripheral chips, which the 68000 lacked. By marketing a system approach,

Intel made the 8088 easier to include in product designs. Finally, IBM already had the right to manufacture the 8086 in exchange for bubble-memory technology to Intel. Thus, although the IBM PC development group unwittingly chose the path of the desktop industry, they may have done so simply to reduce the development effort and for little technical reason.

The period from about 1979 to 1984 saw an unprecedented convergence of events that set the stage for future growth in high-performance microprocessors. In addition to the beginnings of desktop computers as the significant driving application, developments in technology and software, as well as economic forces, laid the foundation for future architecture wars.

New Methods for VLSI Design

Prior to the early 1980s, the semiconductor design process was largely manual. However, the publication of *Introduction to VLSI Systems* in 1980 by Carver Mead and Lynn Conway[9] marked a turning point in design methodologies. Mead and Conway's methodology provided a generation of university students the technical knowledge of how to design VLSI systems, enabling a proliferation of microprocessor architectures. Their book abstracted the complex layout of NMOS transistors into "stick diagrams" to compose circuits with an eye toward their physical arrangement on the silicon and not just their electrical function. Mead and Conway explained the concepts of pipelining and regularity, enabling management of the growing complexity of large chips, namely microprocessors.

As Mead and Conway educated new designers, universities such as the University of California at Berkeley and Stanford University in Palo Alto were developing design tools to support very large scale integration (VLSI). Layout and composition tools were developed to computerize the physical design of VLSI chips. Analysis tools such as switch-level simulators and static-timing analyzers enabled designers to verify functionality based on the transistor netlist, without the need for full SPICE analysis. Other analysis tools such as layout-to-schematic verifiers, design-rule checkers, and electrical-rules checkers enabled devices to be produced that were fully (or at least largely) functional when first fabricated.

Driving the need for new design methodologies was the inexorable migration to smaller transistor geometries. The decade began with 3-micron technology in wide use. By 1985, transistor channel lengths had reached 1.25 micron and even shorter.[10] The Intel 386DX was introduced in October of 1985 with 1-micron gate lengths. The level of integration enabled essentially the entire CPU core to reside on a single die. However, floating-point units (FPUs) and memory management units (MMUs) were still typically external chips. The first microprocessors with on-chip MMUs and caches started to appear after the middle of the 1980s. CMOS was becoming the dominant technology over the earlier NMOS. The primary advantage of CMOS was low power consumption. Early packaging limited power dissipation to a couple of watts. Integration had reached a point where an NMOS-based chip (with non-zero static power dissipation) could not fit in the power budget of these packages. Clock speeds were still low enough that the dynamic power dissipation of CMOS devices was not a problem.

The mid-1980s saw experiments with gallium arsenide (GaAs) as a replacement for silicon. However, even at this point, the economies of scale gave MOS processing a huge advantage over GaAs. Companies such as Vitesse Semiconductor succeeded in finding a niche for GaAs devices. However, others such as GigaBit did not last, even after being purchased by Cray Computer Corporation for its Cray 3.

Microprocessors up to this point had been designed and manufactured by semiconductor vendors, the only ones with both design knowledge and fabrication capability. However, the advent of the 1980s saw the introduction of a new semiconductor business model and new technology for would-be microprocessor vendors—the silicon foundry. An early example of this model was LSI Logic, founded in 1981. With the availability of foundries, non-semiconductor manufacturers could become microprocessor design houses. This became particularly significant for workstation manufacturers later in the 1980s. Foundries lowered the threshold for introducing new microprocessor architectures. Conversely, as foundries showed the success of a business model without design resources, the "fab-less" semiconductor vendor illustrated the possibility of a semiconductor vendor without fabrication capacity. These business models

were exploited by early reduced instruction set computer (RISC) vendors.

Software for a New Industry

As desktop microprocessors experienced consolidation, systems and software were undergoing similar activity while driving microprocessor choices. The desktop industry was moving from systems primarily intended for hobbyists and home use to systems for business. The most popular desktop OS of the day was not Microsoft's MS-DOS.* Although many desktop systems featured BASIC as their primary programming language, the wide use of UNIX* and C on minicomputers influenced the development of the next generation of microprocessor architectures. The engineering workstation became a key application for advanced microprocessors and a development platform for future microprocessors.

Early microcomputer systems of the late 1970s and early 1980s were agnostic in their choice of processors, using the MOS Technologies 6502, Zilog Z80, Intel 8080, and others. However, as systems based on newer 16-bit processors appeared, the choice of CPU became more important.

Although the first 16-bit microprocessors became available in 1979, few desktop systems used these more-powerful chips. In 1979, TI introduced the TI99/4 PC based on the TI 9940 16-bit microprocessor. Most other systems continued to use 8-bit microprocessors. In 1980, Apple introduced the Apple III, again based on a 6502, but at a much higher price than the Apple II. Significant peripherals such as modems, hard disk drives, and floppy disk drives first appeared about this time.

Meanwhile, IBM was considering entering the PC market. Although initially it considered the 8080, IBM switched to the 8086 and later to the 8088 for the final product. In 1981, IBM brought its product to market with the 4.77-MHz Intel 8088, featuring 64-KB RAM, 40 KB-ROM, a 5.25-inch floppy drive, PC-DOS 1.0 (Microsoft's MS-DOS), and a monochrome monitor. Although downplayed by competitors Apple and Tandy, IBM's entry in the market legitimized the PC industry, giving it much more credibility in the eyes of business customers.

Before the year was out, the first third-party add-on peripherals for the IBM PC appeared. By June of 1982, the first IBM clone PC, from Columbia Data Products, was released. These developments emphasized the open nature of the platform. The key to the clone market was the availability of "clean room" basic input/output system (BIOS) code. Once this code was available (legally), it soon became possible for just about anyone to assemble a PC.[4] IBM continued to develop the platform with the XT in 1983, which included a 10-MB hard drive, more expansion slots, and 128-KB RAM. IBM introduced the AT in 1984 with a 6-MHz 80286, a 5.25-inch 1.25-MB floppy drive, and 256-KB RAM (no hard drive or monitor), running PC-DOS 3.0.

Although IBM introduced the business user to PCs, the home market was still a significant consumer. In 1981, Commodore announced the VIC-20, with a full-size keyboard, 5-KB RAM, and a 6502A CPU. It provided an inexpensive color home computer, using a television as the monitor, for $300. Its production peaked at 9,000 units per day. Commodore followed this with the Commodore 64 in 1982. This product included a 6510 (still 8-bit) CPU, 64-KB RAM, 20-KB ROM, custom sound, color graphics, and Microsoft BASIC for $600. After dropping in price to $200 in 1983, the Commodore 64 went on to become the best selling PC of all time, with sales estimated at 17 to 22 million units. Commodore introduced models intended for business users, but the venture enjoyed little commercial success.

The first significant desktop platform to use the 68000 was the Apple Lisa in 1983. The Lisa had a 5-MHz 68000, 1-MB RAM, 2-MB ROM, a black and white monitor, dual 5.25-inch floppy drives, and a 5-MB hard drive. The Lisa's introductory price was $10,000, after costing Apple $50 million for the hardware development alone. Lisa was the first personal computer to feature a graphical user interface (GUI). At the same time, Apple introduced the much-lower-priced IIe, still with a 6502 CPU, at $1,400.

With an Orwellian ad during the 1984 Super Bowl, Apple introduced the Macintosh* computer, based on an 8-MHz 68000 CPU. The Macintosh featured 128-KB RAM, a built-in black and white screen,

a 400-KB 3.5-inch floppy drive, and a mouse. The Macintosh GUI became Apple's primary competitive advantage for several years and the chief alternative to IBM-compatible PCs.

Although there was some early activity in producing Apple II clones by a few manufacturers, it was nowhere near the scale seen with IBM-compatible PCs. The IBM-compatible scene gave birth to Compaq, whose PCs were so successful that they propelled it into the Fortune 500 faster than any other company to date. Apple, on the other hand, through legal and technical means, discouraged the growth of a clone market. It was not until 1987, with the introduction of Nubus-based Macintoshes, that Apple endorsed even a limited third-party hardware market.

In these early years of desktop systems, software and OSs were available for a wide variety of platforms. Through the early to mid-1980s, existing application areas advanced with the introduction of WordPerfect* for DOS (Satellite Software International) in 1982, Lotus 1-2-3* spreadsheet in 1983, and Microsoft Word, also in 1983. Aldus PageMaker* created the desktop publishing market in 1985. As these applications added features, they overwhelmed the memory and processing power of early desktop systems, creating a pull for more powerful microprocessors and ways to address more memory.

Was There Life Before MS-DOS?

The development of desktop OSs has probably had the most impact on the microprocessor landscape. At the beginning of the 1980s, Digital Research's CP/M* (control program/monitor) was probably the most popular OS for microprocessors. Initially available on the Intel 8080, it was later ported to the Z80, the 8086, and the 8088. In 1980, Microsoft was in the interesting position of promoting both CP/M and Apple when it introduced the Z-80 SoftCard for the Apple II, enabling the latter to run CP/M and greatly contributing to its success. Also in 1980, IBM approached Digital Research about using CP/M-86 for an upcoming microcomputer product. They were not interested. This lack of interest would consign Digital Research to the desktop sidelines. It would be another 13 years before a cross-platform desktop OS other than UNIX (Microsoft's Windows NT*) became avail-

able, too late for non-x86 microprocessors.

Microsoft at this time was largely a programming language vendor. It had success in selling BASIC and FORTRAN compilers for early microcomputer systems, supporting a variety of microprocessors. Although Microsoft had an internal OS project (XENIX*) at the time, in 1980 it went outside for what was to become MS-DOS.

Seattle Computer Products (SCP) had developed a disk operating system for the 8086 earlier in 1980 because of delays in Digital Research's introduction of CP/M-86. Microsoft and SCP had worked on other projects before and SCP showed Microsoft its 86-DOS* in September of 1980. Microsoft was already discussing programming language products with IBM, as well as an OS for IBM's upcoming desktop product. Coincidentally, IBM was planning an 8086-based microcomputer. Microsoft licensed 86-DOS from SCP and bought non-exclusive marketing rights. Eventually, Microsoft bought all rights to the product and changed its name to MS-DOS in 1981. Soon after, Microsoft ported MS-DOS to a wide variety of (almost) IBM-compatible PCs, thus contributing to the proliferation of the x86 installed base.

In 1985, Microsoft delivered Windows* 1.0 for x86 PCs (two years after it was initially announced). Although Microsoft tried to interest IBM in Windows, IBM declined in favor of an internally developed GUI, which became Presentation Manager for OS/2. Windows, in spite of its shortcomings, sustained the x86 platform in the face of the threat from the Macintosh GUI and non-x86 desktop platforms.

"PCs" for Engineers

The engineering workstation industry was founded during the early 1980s and became an important force for innovation in the microprocessor industry. Apollo introduced its first workstation in 1980 based on the 68000. Sun, Silicon Graphics, and Hewlett-Packard (HP) also offered products based on the 68000. High-level-language programming, particularly in C, was growing in popularity, and the 68000 provided an efficient target for a C compiler. Prior to this time, assembly language or interpreted languages such as BASIC were popular for microcomputers. As compilation became more important, microprocessor

Table I. Microprocessor features.

Microprocessor	Date of introduction	Clock speed (MHz)	Architectural width	Addressable memory	Features
Intel 8086	6/78	4.77/8	16-bit	1 MB, segmented	16-bit successor to 8080/8085
Intel 8088	6/79	5/8	16-bit, 8-bit external	1 MB, segmented	CPU for IBM PC
Intel 80286	2/82	8/10/12	16-bit	16 MB, protected mode	Virtual memory
Intel iAPX432	1980/1983	8	32-bit	1 TB, segmented	Object oriented
Motorola 68000	9/79	4 to 12.5	32-bit, 16-bit external	16 MB, linear	First with 32-bit programmer's view
Motorola 68010	3/82	4 to 12.5	32-bit, 16-bit external	16 MB, linear	Virtual memory
Motorola 68020	3/84	16.67	32-bit	4 GB	3-stage pipeline, instruction cache
Zilog Z8000	1979	4	16-bit	8 MB, segmented	Incompatible successor to Z80
UC Berkeley RISC I/II	1980/1982	8/12	32-bit	4 GB	First RISC microprocessors
Stanford MIPS	1981	8	32-bit	4 GB	Advanced compiler techniques

RISC – Reduced instruction set computer

architecture research (outside the x86 arena) began to consider how to design microprocessors to execute compiled code more efficiently.

The popularity of C (developed with UNIX from 1969 through 1973) was intertwined with UNIX. UNIX became popular for software and hardware development in industry and academia outside the PC space, offering a productive environment for building tools.

Invented at Bell Labs, UNIX was available to others for study and modification. The versions developed at the University of California at Berkeley were particularly influential, producing Berkeley Software Distributions (BSD). UNIX became the development platform for the infant electronic design automation industry, feeding a synergistic relationship between microprocessor development tools and microprocessor development platforms. Early in the 1980s, the combination of C, UNIX, and university research gave rise to a new architecture paradigm, RISC. New industry players, such as MIPS Technologies in 1984, brought such microprocessors to market.

16-Bit, 32-Bit, and Early RISC Microprocessors

Although the systems and software defined microprocessors of this era to end users, the engineers designing these chips were grappling with internal details such as compatibility with 8-bit predecessors, extending memory addressing to more than 64 KB, virtual memory, instruction caches, and even new architecture paradigms. A survey of the significant microprocessors of the period illustrates the technical decisions that were made. **Table I** shows the basic features of these microprocessors.[11,12,13]

The 8086 microprocessor was structured as a bus interface unit (BIU) and an execution unit (EU). The BIU handled instruction and operand fetches from memory. The BIU fed opcodes to and requested operands from the EU, which performed the instructions. **Figure 5** shows a block diagram of the 8086.[14]

The BIU and EU constituted a simple pipeline, with the BIU fetching instructions concurrently with processing in the EU. The 8086 was source-code-compatible with the 8080/8085. It used variable-length instructions of one or more bytes fetched into the prefetch queue. The four 16-bit registers could be used as either 16-bit or 8-bit registers. The 8086 instituted an unusual form of segmented addressing. Within a segment, addressing was limited to 64 KB. Addressing was expanded to 1 MB by the addition of the segment
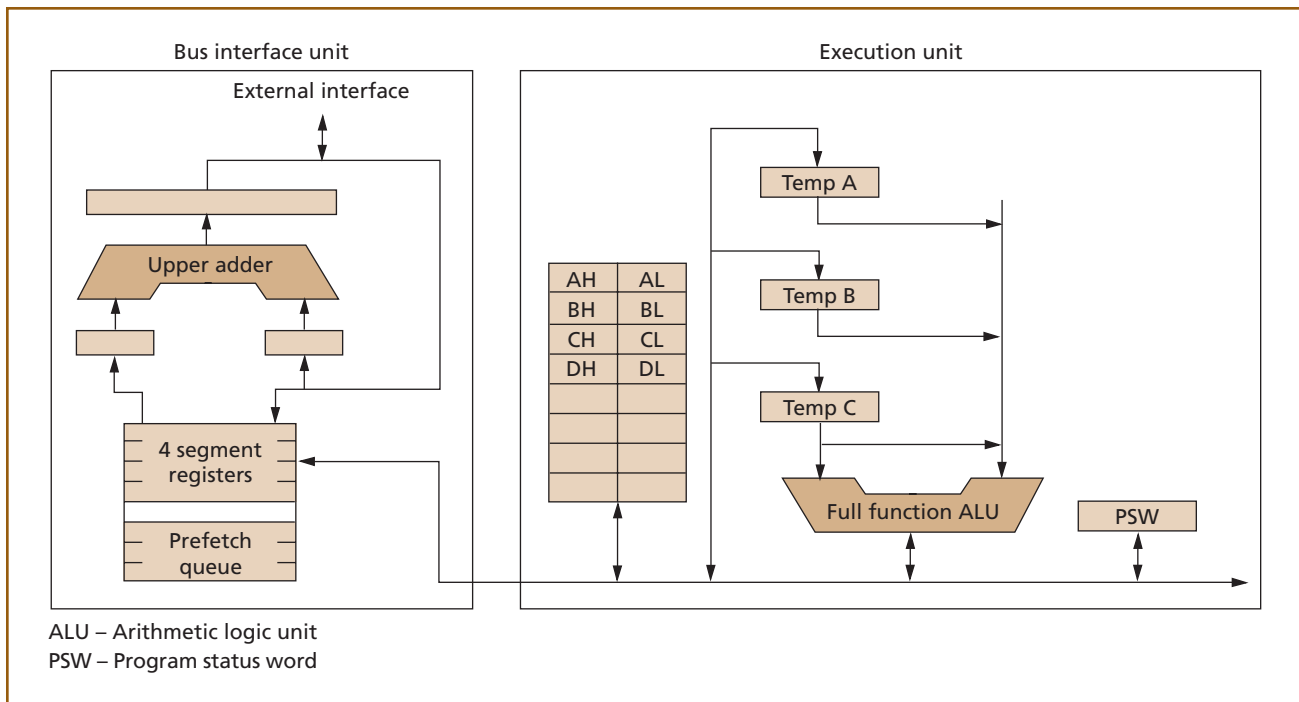
Figure 5.
Block diagram of the 8086.

register shifted by four to a 16-bit address. The 80286 extended addressing to 16 MB, but still through segments of no more than 64 KB and only in "protected" mode as opposed to the 8086's "real" mode. The 8086 had a companion floating-point chip, the 8087. The 8087 introduced Intel's 80-bit floating-point format, greatly influencing the IEEE floating-point standard 754, issued in 1985.

The 68000 had a more orthogonal architecture than the 8086. The 68000 fetched instructions of one or more 16-bit words. It featured 32-bit address and data registers, providing a linear address space and a path to future full 32-bit implementations. The 68000 had a simple pipeline, overlapping instruction fetch and execution. The 68010 added virtual memory support through the ability to restart instructions on a page fault. The 68020 was one of the first true 32-bit processors with a true pipeline, overlapping operand access with internal execution. It also was one of the first microprocessors with an on-chip instruction cache of 256 bytes.

The Z8000 was Zilog's follow-on to the successful Z80. However, the Z8000 sacrificed the compatibility

of the Z80 to make better use of a 16-bit external bus to memory and to make the instruction set orthogonal with respect to its 16 general-purpose registers.[15] The Z80's 8-bit opcodes could not encode more than one of the 16 registers as an operand. The Z8000's 16-bit registers could also be used as thirty-two 8-bit registers, eight 32-bit registers, and even as four 64-bit registers. The Z8000 was not pipelined because it was felt that the fixed 16-bit instruction format and simple address calculation eliminated the need for prefetching. The Z8000 was also singular in using hardwired logic instead of microcode ROM, in spite of increasing the instruction set from 128 instructions in the Z80 to 414. This may have contributed to its lack of success, since it suffered from initial bugs.

Another similarly notable processor of this period was the Intel iAPX432.* The 432 implemented many advanced features, unfortunately before the technology could support them. The 432 was positioned as an ideal Ada processor, incorporating many object-oriented features. Implementing these features in hardware slowed memory access with multiple segment lookups. The instruction set was bit-aligned in mem-

ory, virtually ensuring slow access and decoding. The 432 included support for multiprocessor implementations and fault-tolerance mechanisms. However, its complexity delayed introduction of the ultimately five-chip system until 1983, when the last two chips came out. The first three chips, a two-chip decoder/execution unit and an I/O controller, were introduced in 1980. The complexity also resulted in its being much slower than the 8086 and 68000.

The Beginning of the RISC Argument

In the early 1980s, the stage was being set in academia for the next phase of microprocessor evolution. Projects at the University of California at Berkeley and Stanford University in nearby Palo Alto were developing RISC microprocessors. Although the 8086/8088 and 68000 were well established with significant desktop bases, the field of computer architecture was much wider and older than microprocessors alone. The RISC movement began in reaction to the complexity of a minicomputer architecture, the VAX* from Digital Equipment Corporation (DEC).

The basic tenets of RISC were evident in earlier non-microprocessor architectures such as the IBM 801 by John Cocke and Control Data's 6600 by Seymour Cray. Unlike contemporary and earlier complex instruction set computer (CISC) processors, the RISC projects endorsed fixed-length, 32-bit instructions, no memory-to-memory instructions (RISC used a load/store architecture), large, general-purpose register files, and pipelining. In particular, the RISC projects formalized a fundamental performance metric for computer architectures, namely the amount of CPU time required to execute a given task. This was expressed by the equation *CPU time = instruction count x clock cycles per instruction (CPI) x clock cycle time*. A typical CISC had three or four, while RISCs approached the goal of achieving one cycle per instruction.

Professor David Patterson's project at Berkeley coined the term "RISC" with the RISC I microprocessor. Patterson's experience with VAX microcode at DEC may have led to the notion of compiling from C directly to microcode. However, the RISC philosophy, in some respects, was born of necessity. A university project had to meet the constraints of graduate students with little VLSI training and the limited duration of the academic year. The RISC projects popularized the idea of quantitative analysis of applications.

It was well known that the VAX, IBM 370, and other CISC architectures were characterized by a small subset of frequently used instructions, with many other instructions rarely used. The project teams at Berkeley and Stanford extensively analyzed the instruction usage characteristics of compiled programs. They found that most applications had surprising commonality in their instruction execution and data access patterns. From this analysis, the Berkeley group designed RISC I and II based on a large register file, divided into overlapping windows for the stack frames used by the compiler. The RISC processors led in introducing pipelining in microprocessors, with a two-stage pipeline for RISC I and a three-stage pipeline for RISC II. The RISC I/II ideas found later commercial application in Sun's SPARC* architecture.

The Berkeley team recognized the need to tailor the architecture to the compiler and to tune the compiler to the needs of the hardware. The notion of using the compiler to address the problem of branch latency (branch delay slots) was used at both Berkeley and Stanford. These projects were among the first attempts to treat the compiler and microprocessor as a single system, trading hardware for compiler complexity. At Stanford, the microprocessor *without* interlocking pipe stages (MIPS) project took optimizing compiler technology further.

The MIPS architecture required the compiler to manage all interlocks and data dependencies between instructions as well as the control dependencies of branches. The Stanford MIPS even introduced some compiler capabilities similar to very long instruction word (VLIW), packing two instruction pieces into a single 32-bit instruction word. The Stanford team emphasized compiler register allocation to handle the stack frames of compiled code in 32 general-purpose registers without resorting to a large windowed register file, as the Berkeley team had done. Some of these innovations were scaled back when the Stanford group ventured into the commercial world to found MIPS Technologies, Inc.

The Promise of RISC

"RISC: any computer designed after 1985"
—Stephen Przybylski (a designer of the Stanford MIPS)

The claim of RISC's superiority over CISC, outlined by Berkeley RISC and Stanford MIPS, led to the first commercial RISC CPUs in the second half of the 1980s. The workstation manufacturers abandoned Motorola 68K CPUs in favor of their own RISC CPUs. The first commercial RISC CPU, the MIPS* R2000, was based on the Stanford MIPS and was introduced in 1986. With the threat of RISC looming large, even Intel and Motorola designed their own RISC processors, while continuing to supply their flagship CISC processors in increasing volumes to the cost-sensitive PC market, which required compatibility. The RISC processors, on the other hand, were targeted at the performance-oriented UNIX workstation market, where price was secondary. This set the stage for the battle between price and performance.

The lower cost of IBM-compatible PCs compared to Apple's proprietary Macintosh computers increased the volume of Intel's 80386 (introduced in 1985) and 80486 (1989) processors. Much of the success of the x86 processors was based on the fact that the IBM PC used an open standard, which enabled hundreds of manufacturers to produce low-cost computers. The x86 CPUs were also licensed to several vendors, although the leading edge was confined to Intel.

Architectural Features

Several architectural features that defined the second and third generations of microprocessors were introduced. Pipelines deepened from the simple overlap of fetch, decode, and execute stages (characteristic of the Intel 80386 and Motorola 68030 CPUs) to over five stages (typical of the RISC CPUs). Data and instruction caches were incorporated on chip, along with memory management and cache-control functions. FPUs were also integrated by the late 1980s. The push to integrate was more pronounced in the CISC processors. The RISC CPUs, which attempted to execute one instruction per cycle, relied on large, fast caches. All these architectural features were enabled by the predictable advance of IC technology. For example, the number of transistors increased from 275K in the Intel 80386DX to 1.2M in the Intel 80486DX. The various processor families are considered in some detail below, with the focus on the major players.

Intel and Motorola CPUs

Intel produced its first true 32-bit processor, the 80386DX, in 1985, a year after the Motorola 68020, which already had 32-bit registers and 32-bit internal address and data buses. The Intel 80386 and the Motorola 68030 (introduced in 1987) were considered to be second-generation CISC processors with limited pipelining. The 80386 provided a fully binary-compatible upgrade to Intel's first-generation processors (the 8086, 80186, and 80286). The new *base+index+displacement* addressing mode allowed the full 32-bit memory space to be easily addressed, a great improvement over the 64-KB segment limitation of the previous generation. More than 30 new instructions were added, along with an MMU that provided four modes of privilege. Motorola introduced the 68030 in 1987 to succeed the three-year-old 68020, which already featured 32-bit external address and data buses and a 256-byte cache. The 68030 had an MMU with two levels of paging and dynamic bus sizing. Internally, it had a Harvard architecture (separate buses for fetching data and instructions) with separate 256-byte caches. Both the 80386 and the 68030 had three-stage pipelines and were clocked at 20 MHz.

Until 1989, the FPUs (implementing the IEEE754 floating-point standard) were separate chips called math coprocessors. Floating-point computations that were previously implemented in software were greatly accelerated by the coprocessors. Intel introduced the 80387 math coprocessor in 1987 as an adjunct to the 80386. Weitek, a company known for floating-point chips, introduced the Weitek 3167 math coprocessor in early 1988 for the 80386. With the introduction of the Intel 80486 in 1989, the FPU was integrated with the CPU. With an 8-KB cache, the 80486 exceeded one million transistors in one-micron CMOS technology and was clocked at 25 MHz. At 20 MIPS, it produced over twice the performance of the 80386 at 25 MHz. In 1991, Motorola introduced the 68040, which had 1.2 million transistors, two 4-KB caches, and an FPU. Although Weitek offered the 4167 as an enhancement to the 80486, the integration of the FPU in all microprocessors made the external

FPUs redundant. Both microprocessors had more pipe stages than did their predecessors.

RISC CPUs

The new commercial RISC CPUs were remarkably similar, following the design established by Berkeley RISC and Stanford MIPS. Instructions were all 32 bits wide. Register files typically had thirty-two 32-bit general-purpose registers. The opcodes provided only the basic instructions. The only instructions that accessed memory and the memory-mapped I/O space were load and store instructions, hence the name load/store architecture. Memory was addressed by *register plus displacement* or *register plus register*. The number of addressing modes was fewer than previous CISC CPUs and few data types were supported. Most RISC CPUs had a separate register file for floating-point operands. Operations that were neither loads nor stores typically specified two source registers and one destination register for the result. This allowed the source registers to be reused, unlike in CISC CPUs, where the result destroyed (wrote over) one of the source operands.

The typical RISC CPU had a five-stage pipeline, as shown in **Figure 6.** Each stage of the pipeline performed its processing in one clock, taking inputs, stored in registers, from the previous stage and storing its results in registers to be processed by the next stage. In the absence of branches, assuming all instructions and data were in cache, and all instructions took only one clock to execute, the pipeline remained full and proceeded without stalling, yielding an ideal CPI of one. Note that the goal of the processor designer and compiler writer is to prevent stalls as much as possible.

A crucial component of the processor was the register file. The larger the register file and the more ports it has, the slower it is. The basic RISC register file was required to perform two reads and one write in a clock cycle. The consistent placement of the 5-bit register values in the opcode facilitated quick reading of the register file. A significant number of comparisons were made with the value zero. Thus, R0 was hardwired to the value zero in many RISC processors. Absolute addressing could be achieved by using R0 as the base. Using R0 as the destination allowed subtract instructions to be used in place of compare instructions. Specifying three registers consumed 15 bits of the 32-bit instruction.
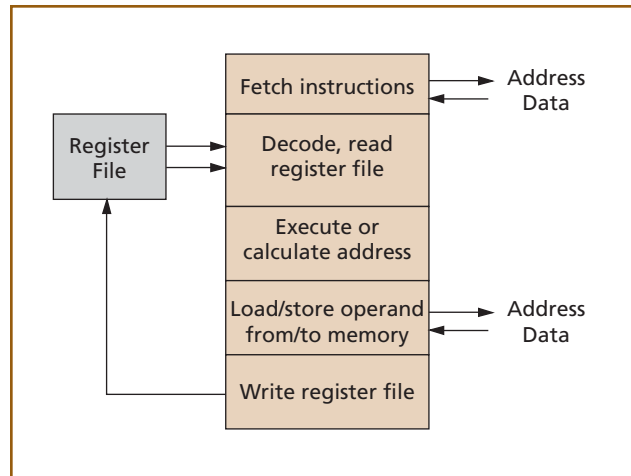
Figure 6.
Basic five-stage processor pipeline.

Decoding was also simplified compared to the CISC CPUs, by having fewer opcodes and eliminating complex instructions. All RISC processors had none of the microcode that their CISC counterparts required to execute complex instructions. The various RISC CPUs also had unique features, which are outlined next.

MIPS R2000 was the first commercial VLSI RISC processor and was an extension of the Stanford MIPS processor. Pipeline interlocks, which ensured that registers always had the latest values, were omitted in the R2000. This caused a one-clock delay between a register load and its use in the next instruction. The compiler was responsible for inserting a NOP (no operation instruction) between reads to ensure correct operation. It had only *register plus displacement* addressing.

The MIPS architecture also eliminated condition code bits for integer relations. The result of a comparison could be written as a zero or one into any register. A unique feature of MIPS allowed misaligned data (a word placed on a non-word boundary) to be loaded or stored correctly using only two instructions. It also had two dedicated registers, HI and LO, which held a 64-bit integer product or the quotient and remainder after integer division. MFHI and MFLO instructions were then used to transfer the required word into a general-purpose register. The MIPS architecture had only 16 floating-point registers.

The MIPS architecture was designed with efficient pipelining in mind. The compiler was responsible for scheduling the pipeline to avoid hazards, since the

machine had no interlocks. The R3000 was offered in 1988 and had comparators on chip to perform tag matching so that off-the-shelf static RAMs could be used for the external cache. In 1989, the MIPS R3000 was offered in a 144-pin package containing a 56-mm^2 die clocked at 25 MHz for about $300. In comparison, the Intel 80486 (with FPU and 8-KB cache), introduced one year later, measured 165 mm^2, had 168 pins, was clocked at 33 MHz, and cost $950.

Sun Microsystems developed the SPARC architecture based on Berkeley RISC for its own workstations, displacing the Motorola 68K CPU. SPARC was an open specification for a RISC processor and was fabricated by licensees. The first SPARC (1987) was the CY7C600 chip set by Cypress. The unique feature of SPARC was the windowed register file (a feature used in Berkeley RISC), which reduced memory traffic caused by saving and restoring registers on procedure calls. Each window allowed each procedure access to 32 registers (24 in window and 8 globals). An implementation could scale the number of windows from one to a maximum of 32. Each window had eight registers each for inputs, locals, and outputs, facilitating parameter passing from the called procedure to the callee procedure. The CY7C601 integer unit, which implemented all instructions except floating-point and coprocessor operations, had 136 registers. The current window pointer pointed to the window currently in use and was stored as five bits in the processor status register.

The SPARC design also included tagged addition and subtraction to aid languages such as LISP, Prolog, and Smalltalk. Tagged data was declared as an integer data type and was handled as unsigned words. The two least-significant bits were used for the tag. Integer and floating-point execution could be overlapped. A square-root operation was also included in the floating-point instruction set. Multiplication and division were supported by providing multiply-step and divide-step instructions. A swap instruction executed an atomic swap of a register with memory to support multiprocessor systems.

RISCs from Intel, Motorola, and AMD

The pervasiveness of the RISC philosophy prompted Intel, Motorola, and AMD to offer their own RISC CPUs about the time the workstation vendors announced their new CPUs. Each of the CPUs had unique features worth mentioning. Intel designed the 80960K and AMD the 29000 to serve the embedded market; both achieved great success. Extensive support for debugging and monitoring, superior exception handling, and quick context switching were requirements for the embedded CPU. Moreover, the memory subsystems were slower because of the cost constraints on embedded systems. The Intel 80960 (1988) register file had 32 global registers and 4 register banks (later expanded to 16) and 32 special-purpose registers. Thus, quick context switching was possible by reducing memory accesses. The efficient interrupt model saved the state of the processor and restored it without software intervention. A separate interrupt stack was also provided. The instruction set supported bit-field operations and floating-point operations, including several trigonometric operations. The design used register scoreboarding to allow multiple instructions to be executed. The 80960CA, introduced in 1989, was superscalar.

AMD's 29000 succeeded the 2900 bit-slice series and was derived from the Berkeley RISC. Introduced in 1987, it had a large register file—64 global registers, plus 128 local registers managed as a stack cache. The top of the run-time stack was mapped to the local registers to avoid memory accesses during procedure calls. Like the 80960K, it had tracing and breakpointing to support debugging. The floating-point instructions did not include trigonometric functions. The four-stage pipeline was interlocked. For many years, the 29000 was the most popular embedded processor, before being overtaken by the 80960 series. After the 29040 was produced in 1995, AMD abandoned the 29K series to focus on the lucrative x86 market.

Motorola's 88100 failed to achieve the success that Intel and AMD did. It had a single 32-bit register file for integer and floating-point operations. Extensive bit-manipulation instructions were provided. Instructions after multicycle instructions could be issued if no data hazard occurred. The four execution units (instruction fetch, data access, floating point, and integer unit) could operate in parallel. Load and store operations were pipelined and the Harvard architecture allowed two caches for instruction and data.

By the end of the 1980s, several CPU vendors discontinued their 32-bit processors, which had failed in the marketplace. Notable among those were Zilog and National. National began with the 16032 in 1980 and produced a compatible series of 32-bit processors—the 32032, which was similar to Motorola's 68000, the 32322, and finally the 32532. Fairchild produced several versions of the Clipper,* a RISC CPU, and was bought out by National.

The RISC philosophy had found a firm foothold in computer architecture. Several new RISC vendors emerged, such as Advanced RISC Machines (ARM) and Hitachi, targeting their processors at embedded niche applications. The RISC vendors had the advantage of not having to be compatible with previous architectures. A seminal textbook, *Computer Architecture: A Quantitative Approach* by Hennessy and Patterson,[16] who played lead roles in Berkeley RISC and Stanford MIPS, educated thousands of students and designers on the latest approach in processor design. The successes of the processors that emerged were largely based on the volume of the systems that used them and less on technical merits. System sales in turn were influenced strongly by price and application software base. The success of the x86 processors prompted others to produce clones, the first being AMD with the 80386.

Microprocessors of the 1990s

"Intel Inside"
                    —Intel advertising slogan

We now look at the evolution of the high-performance CPUs and their design features since 1992. Increasing performance requires reducing the CPI, the number of instructions in a program, and the clock period. The problem is that reducing any one factor increases the others, and improving performance requires artful balancing of the features that affect these factors. The second generation of RISC processors appeared in the early 1990s, and the similarities with the first generation disappeared as each vendor adopted different features.

In 1992, DEC produced the first Alpha* microprocessor, the 21064, which was clocked at an astounding 150 MHz. Recognizing that the success of a processor line depends on the volume of systems using

the CPU, IBM, Motorola, and Apple formed an alliance to design the PowerPC* processors based on IBM's Power architecture. IBM brought its RISC experience, Motorola its multiprocessor bus interface developed for the 88100, and Apple a ready consumer base that would redesign the Macintosh around the new processor. The alliance could hardly fail and was expected to mount a serious threat to Intel's x86. The x86 itself adopted many RISC ideas and the distinctions between RISC and CISC became less important than success in the marketplace. Although they thrived in the embedded market, the 68K family of processors left the desktop after the 68040 was replaced by the PowerPC601. The last in the line was the 68060.

The need for comparative evaluation of their RISC CPUs prompted several manufacturers to adopt the SPECmarks rating, based on benchmarks defined by Standard Performance Evaluation Corporation. Introduced in 1989, the ratings consisted of two numbers—SPECint for integer performance, based on six applications, and SPECfp for floating point, based on 14 floating-point kernels. Each number was a measure of the speedup of the CPU (in a UNIX system) relative to a VAX 11/780. The SPEC numbers were influenced by the compiler and system features such as cache size. The newer rating system, SPECint95 and SPECfp95, gave more weight to these factors.

Alpha and PowerPC

The Alpha21064 and PowerPC601 best illustrate the contrasting designs of the various RISC CPUs and are considered below in some detail.[17] Both processors were load-store architectures, with 32-bit instructions and two 32/64-bit register files for floating point and integer. The Alpha designers focused on very fast clocks, a simple instruction set that would enable fast clocking, and deep pipelines. The PowerPC instruction set had powerful instructions that did more in each clock. Of the three factors that affect performance, Alpha chose to reduce the clock period and CPI at the expense of the number of instructions. The PowerPC601 took a more balanced approach.

The clock rate of a CPU depends on the amount of logic in each pipeline stage. Thus, longer pipelines reduce the amount of logic in each stage and allow

faster clocks. Unfortunately, branches in program execution cause greater penalties in deep pipelines. Therefore, prediction of branches has become critical to high performance.

Examining the simplicity of Alpha relative to the PowerPC reveals some of the choices all CPU designers face. Alpha began with a 64-bit architecture and PowerPC601 defined a 32/64-mode bit that would allow 64-bit processors in the future. Alpha provided only the *register plus displacement* addressing mode, while PowerPC601 had *register plus register* as well, with post-modification of the index register. Thus, the PowerPC601 needed more ports on its register file. Alpha loaded and stored data in 32-bit or 64-bit words and did not align misaligned data in hardware. The PowerPC601 had byte loads and stored and handled misaligned data. Byte alignment was performed with separate instructions in the Alpha. Thus, the Alpha load/store pipeline was simpler and allowed faster access to its two direct-mapped 8-KB caches for instruction and data. The PowerPC601 had a 32-KB unified eight-way set associative cache that was slower but yielded a higher hit rate. Like previous RISC CPUs, Alpha had no condition code register, and the results of a comparison were written into any integer register. Conditional branches could test for zero or odd/even. The PowerPC had defined a condition code register and instructions had the option of modifying the condition code. It had a single instruction to test a counter and branch back to the top of a loop. Thus, some PowerPC instructions replaced two Alpha instructions.

Besides pipelining, performance improvements can be made by using several functional units and issuing more than one instruction. The Alpha had a load/store pipe and integer pipe with seven stages in each. The floating-point pipeline contained 10 stages. The heavy pipelining required 38 bypasses to hide latencies. The PowerPC had shorter pipelines in its branch, as well as in its integer and floating-point units, and had buffering to allow dispatch to busy units. It could also dispatch instructions out of order. The Alpha did not read register files in the decode stage as the PowerPC did. Deep pipes increase the branch latency (number of idle cycles that are caused by a conditional branch), which seriously affects per-

formance (branches may be 20% of general-purpose code). The Alpha designers, therefore, included dynamic branch prediction and conditional move instructions. Dynamic branch prediction was implemented, with a history table storing the result of the most recent branches. On the other hand, the PowerPC implemented the less-effective static branch prediction in its branch unit, whereby a bit is set by the compiler, predicting the probable outcome of the branch.

MIPS, Sun, and HP

At its introduction in 1992, the MIPS R4000 was one of the fastest single-chip processors, with a super-pipelined 64-bit architecture. This architecture was engendered by the high-end graphics market that Silicon Graphics dominated. The external clock (50 MHz) was doubled in the CPU to clock the deep pipelines at 100 MHz. Address and data buses were 64-bit and multiplexed. The R4000 had separate direct-mapped instruction and data caches of 8 KB and a second-level cache controller on chip. Several variations of the R4000 were made in the following years and the MIPS architecture became popular in the embedded marketplace.

The 64-bit architecture was particularly useful in game machines, which required good graphics. In contrast to MIPS, Sun was the laggard in the high-performance CPU race. Its first 64-bit superscalar CPU, the SuperSPARC,* was unimpressive. Sun used dual processors in its workstations to compensate for poor uniprocessor performance.

In 1995 the UltraSPARC,* fabricated by TI, put Sun back in the race. The 167-MHz UltraSPARC could issue four instructions in order to any of the nine units: two integer units, a branch unit, a load/store unit, and five floating-point/graphics units. Caches were 16K, direct-mapped for data, and two-way set associative for instructions. The UltraSPARC introduced the visual instruction set (VIS) to support pixel processing. Pixels, the units of which a picture is comprised, are expressed as three 8-bit scalars for color pictures. Pixels were recognized, much as floating-point numbers were, as an important new data type. Block move instructions in the UltraSPARC could bypass the cache since pixel data were not reused. The 64-bit

arithmetic units could operate simultaneously on eight 8-bit values stored in the 64-bit floating-point registers. This capability provided a significant increase in the speed-of-motion estimation computations in the Motion Picture Experts Group (MPEG) standards. Instructions for graphics support appeared earlier in the M88100 and PA-RISC,* but VIS was more extensive and was targeted towards MPEG.

Unlike Sun and MIPS, HP manufactured its own processors for its workstations. Therefore, PA-RISC was more proprietary than MIPS or SPARC. The PA-RISC 7100 and 7200 were 32-bit processors with external cache that required systems to use very-high-speed static RAMs (SRAMs). The 7200, produced in 1994, had two integer units and one FPU. It dispatched two instructions to any of the three units. The first 64-bit architecture from HP was the 180-MHz PA-RISC 8000, produced in 1996. For a short time it surpassed Digital's 333-MHz Alpha 21164 in integer performance. The pattern for most of the 1990s has been that every new processor introduced tends to surpass its older rivals. The only exception has been Alpha, which has held the top spot for most of the decade. The threat to these traditional RISC vendors is the proliferation of x86 and the encroachment of Windows NT into the UNIX market.

Dominance of Intel and Microsoft

Over a decade after Apple introduced the mouse and Windows, Microsoft produced Windows 3.0 for the IBM PC. The enormous volume of the so-called Wintel PCs made Intel the envy of other CPU manufacturers. It lost the copyright of the 8086/88 microcode in a dispute with NEC in 1989 and the 80486 and 80386 were cloned by its licensees, including AMD and Cyrix. To avoid trademark problems with the numerical naming convention, Intel called its successor to the 80486 the Pentium.* It was fully binary compatible with the installed base of over 100 million x86 systems. Shipped in early 1993, the 60-MHz Pentium was a 32-bit superscalar CPU with a 64-bit external bus and two integer units. Many of the features of the RISC CPUs were incorporated: dual instruction issue, deeper pipelines, separate 8-KB data and instruction caches, and support of external caches. Intel used BiCMOS technology to achieve higher

speeds than CMOS. In a year, clock rates increased to 100 MHz and a variety of PC manufacturers offered Pentium-based systems at several price/performance points. With its two integer units, the Pentium offered excellent integer performance that speeded up many desktop applications.

In contrast, the PowerPC-based Macintosh systems were more expensive and Apple continued to lose market share. Within two years of Pentium, Nexgen introduced the Nx586 (without the FPU) and Cyrix followed with its 5x86. AMD ran into trouble with its Pentium-class CPU called the K5 and ended up buying Nexgen to launch the K6. The difficulty of implementing the x86 instruction set caused Intel's competitors to map the x86 instructions into RISC-style micro-operations (also called ROPS). Complex instructions took several micro-operations. Thus, the underlying CPU architecture was very similar to RISC CPUs, blurring the distinction between RISC and CISC.

In 1995, Microsoft launched its Windows 95 OS with great fanfare. The 32-bit multi-tasking OS emphasized ease of use. It recognized all devices connected to the system and made installation of peripherals such as printers, CD-ROMS, and modems easy for average users. More significant for the workstation vendors was the prior introduction of Windows NT, a reliable, secure, multi-tasking, 32-bit OS for business and enterprise servers. It ran all the Windows software such as spreadsheets and database applications required by business users. With the price advantage of x86 systems, the low-end workstation market was under attack. Intel pushed Pentium performance further in 1996 with its superpipelined Pentium Pro. It used micro-operations like its competitors, translating x86 instructions into micro-operations using three decoders. With many of the same features used by the RISC vendors, the Pentium Pro's integer performance was better than some of the RISC processors. Its floating-point performance lagged as it always had. Recognizing the need to speed-up multimedia applications, Intel added 57 new pixel-processing instructions (less extensive than Sun's VIS) to the Pentium instruction set. The inclusion of the new instructions is advertised by the term MMX.* With these advances in performance and its policy of cutting prices on older

Table II. Specifications of high-performance microprocessors.

	DEC Alpha 21164	PowerPC 604e	SUN Ultra-2	HP PA-8000	MIPS R10000	Intel PentiumPro
Available	4Q96	2Q97	Limited	2Q96	1Q96	2Q96
Transistors	9.3M	5.1M	3.8M	3.9M	5.9M	5.5M
Die size (mm²)	209	.96	149	345	298	196
IC process	0.35 4M	0.27 5M	0.29 4M	0.5 4M	0.35 4M	0.35 4M
Pins	499	255	521	1085	527	387
Clock rate (MHz)	500	233	250	180	200	200
Maximum power (W)	25	15	20	> 40	30	35
Issue rate	4	4	4	4	1+FP	3
Pipe stages	7	6	6/9	7/9	5	12–14
Out of order	6 loads	16 instr	0	56	32	40 ROPs
Cache size (KB)	8/8/96	32/32	16/16	Not on chip	32/32	8/8
BHT entries	2K x 2-bit	512 x 2-bit	512 x 2-bit	256 x 2-bit	512 x 2-bit	>512
SPEC95 (int/fp)	12.6/18.3	9.0/8.5	8.5/15	10.8/18.3	10.7/17.4	8.7/6.0

BHT – Branch history table
FP – Floating point
int/fp – Integer/floating point
ROP – RISC opcode

processors, Intel continues to stay ahead of its x86 rivals and threatens the application domain of workstation vendors.

Performance improvements in the first half of the 1990s have been realized by using more of the same: more functional units, more pipeline stages, higher issue rates, more out-of-order instructions, more bandwidth and pins. **Table II** shows the high-performance desktop processors currently in production.[18] Omitted from the list are the x86 compatibles from AMD and Cyrix, which still trail the leading edge defined by Intel.

The increased clock speed brought new thermal problems for chip designers. The several million transistors of a processor clocked at several hundred MHz consume 30 to 40 watts. The thermal problems were first faced by Alpha, with its high clock rates. Power consumption has been significantly reduced by dropping the voltage. The 5-volt standard of the 1980s has yielded to voltages between 2 and 3 volts. At the system level, thermal problems have been addressed with heat sinks and fans dedicated to cooling the CPU. The

high out-of-order issue rate and techniques such as register renaming, branch prediction, and speculative execution have increased complexity, making it difficult to ship a bug-free processor. The most famous of these bugs was the Pentium floating-point division bug, which embarrassed Intel and forced it to replace defective CPUs.

Future Directions

The expected performance enhancements delivered by microprocessors may slow down in the future because of problems associated with IC technology, computer architecture, and market forces. The shrinking line widths of the next-generation ICs will require new lithographic techniques to draw finer lines. Thus far, the intrinsic delay of the transistor itself has been reduced to enable commensurate increases in clock speed. With finer geometries, the resistance-capacitance delay caused by interconnects becomes the limitation. To reduce this requires basic changes in the IC process itself. Resistance must be lowered by replacing aluminum with copper or gold and capacitance reduced

by using insulators with lower dielectric constants. With most of the delay in interconnects, delay models must also become more sophisticated to predict the clock speed of an entire processor. Alternatives to CMOS such as silicon-germanium may appear.

Limitations in exploiting parallelism must also be overcome. For example, simply increasing the maximum issue rate eventually produces diminishing returns. Applications need to be written in a manner that exposes more parallelism. This is already under way with multi-threading. Advances in compilers will find parallelism across larger sections of a program than previously possible. Approaches such as VLIW put the complexity back into the compiler, much as RISC did over a decade ago, reducing the silicon overhead now spent in extracting limited parallelism. VLIW promises to reduce the complexity of modern microprocessors, which is a problem in itself. The compatibility and code expansion issues associated with VLIW need to be overcome. Feeding a high-performance processor requires fast buses to all levels of memory. This, in turn, requires finely tuned buses as in the two-chip Pentium II processor, which has its level-two cache and CPU on a small printed circuit board.

The greatest influence on the development of microprocessors may come from market forces. New fabrication lines are becoming very expensive, requiring collaborative efforts. New applications such as the Internet and multimedia interfaces are expected to drive the microprocessor in new directions. Java* processors are already being touted by Sun. The microprocessor may likely lose some of its prominence in systems that are increasingly focused on communications and graphics, which require coprocessors to provide the differentiation that is visible to the user.

Appendix. The History of the Microprocessor at Bell Labs

Bell Labs has been engaged in the design of microprocessors since the latter half of the 1970s. The collection of microprocessors developed at Bell Labs include 4-, 8-, and 32-bit microcontrollers, a traditional 32-bit complex instruction set computer (CISC) microprocessor, and an advanced 32-bit reduced instruction set computer (RISC) microprocessor.

One of the common threads running through these processors is that they were all designed for complementary metal-oxide semiconductor (CMOS) technology. While this is commonplace now, in the 1970s and early 1980s this was quite unusual. Most microprocessors of that day were designed using NMOS (MOS using n-type transistors) technology. One reason for the early focus on CMOS within Bell Labs was the constant concern over power consumption within the telecommunications systems designed here.

The first microprocessor designed at Bell Labs was the Mac-8, a general purpose 8-bit microprocessor announced February 17, 1977. The Mac-8 was designed in 5-micron CMOS, requiring 7,500 transistors in an area of 32.45 mm^2. It was packaged in a 40-pin dual inline package and ran at 3 MHz, providing 0.2 million instructions per second (MIPS) in performance.

The Mac-8 was used in a variety of internal embedded applications within the Bell System. One of its unique features was the mapping of the register set to external memory, similar to the TMS9900. The Mac-8 was also one of the first microprocessors to provide an extensive development environment supporting the C programming language.

The Mac-4 was a 4-bit microcontroller intended for more cost-sensitive applications. Available in 1979, the Mac-4 was designed in 3.5-micron CMOS, requiring 30,000 transistors in an area of 28.56 mm^2. It ran at 2 MHz with a 9-volt supply and was available in a 40-pin package. The Mac-4 included the capability for 4-, 8-, 12-, and 16-bit arithmetic, and offered an instruction to put the chip into a low-power state. One of the unique features of the Mac-4 was a mask programmable logic array (PLA) encoder, which performed application-specific decoding or demultiplexing.

At the end of the 1970s, a project to leap from 4-bit and 8-bit parts to a full-blown 32-bit microprocessor was started. This microprocessor, named the BellMac-32,[21] was intended to be introduced in 1980. The first prototype was fabricated in 3.5-micron CMOS technology and was 146 mm^2 in area, requiring about 100,000 transistors. The production version, the BellMac-32A, was available in 1982. The BellMac-32A central processing unit (CPU) chip was fabricated in 2.5-micron CMOS technology and was about 100 mm^2 in area, requiring about 150,000 transistors. It was packaged on a module with four bus interface devices. A subsequent version added an additional chip, the memory management unit (MMU). This module was used in the 3B5 minicomputer.

The BellMac-32A was a pure CISC microprocessor. The instruction set included opcodes for such things as process switches and string operations, which were implemented in a special ROM on the chip. The control of the BellMac-32A was implemented using eight different PLAs, each with its own functions and state machines. The first version of the BellMac-32A ran at 6.5 MHz at 5 volts. This was the first CMOS 32-bit microprocessor and the advantage over NMOS was apparent when compared to a Hewlett-Packard (HP) processor announced at about the same time. The BellMac-32A dissipated less than one watt of power, while the HP processor dissipated about seven watts.

During 1982, it was realized that numerous improvements were needed to make the BellMac-32A a competitive product. After a series of studies of alternative solutions, it was decided to design a single-chip replacement for the BellMac-32A module. This replacement was originally called the BellMac-32B, but was later renamed the WE32100.

The WE32100 was designed in 2.5-micron CMOS technology, requiring about 180,000 transistors. The WE32100 offered improved performance through the inclusion of a 256-byte instruction cache, one of the first microprocessors to integrate a cache on chip. The WE32100 also added a coprocessor interface to support chips such as the WE32106 math accelerator unit. Internally, the WE32100 was used in the 3B2 minicomputer and the Teletype 5620 bit-mapped terminal. The WE32100 also became the first Bell

Labs microprocessor sold to outside companies.

At the same time the WE32100 was being designed, efforts had begun on more advanced microprocessor architectures. A group was defining an architecture for a C-machine that would offer much higher performance. Among this group was Dave Ditzel, one of the first proponents of RISC and an eventual key contributor to the SPARC architecture. The C-machine, named CRISP, demonstrated several advanced architectural features. Among those features were branch prediction, branch folding, single-cycle execution of most instructions, a decoded instruction cache, and a stack cache. The first version of CRISP was fabricated in 1986 using a 1.75-micron CMOS technology. It required about 172,000 transistors and measured about 126 mm^2.

In 1988, Apple selected the CRISP architecture for use in the personal digital assistant (PDA), which would evolve to become the Newton. This project led to the creation of the Hobbit* microprocessor, announced in 1990. Apple subsequently dropped Hobbit from its plans, but the design continued and was the microprocessor inside the EO personal communicator. The Hobbit microprocessor refined the CRISP design and added on-chip support for virtual memory. The first Hobbit chip was fabricated in 0.9-micron CMOS, requiring 413,000 transistors in an area of 94.4 mm^2. The Hobbit offered an attractive combination of high performance and low power.

Following the demise of the EO personal communicator, the experience gained from the Hobbit chip was applied to a microcontroller targeted for embedded applications with AT&T's—later Lucent's—successful line of digital signal processors (DSPs). This work led to the creation of the 32-bit communications protocol processor (CPP), which is a general-purpose RISC microprocessor core. The CPP core is currently being used in the CPP-Cellular™ chip, a microcontroller designed for protocol and human-machine interface processing within digital cellular phones. The CPP-cellular chip was first fabricated in 1996 using 0.5-micron CMOS. The CPP core requires about 60,000 transistors in an area of 3.2 mm^2. The CPP core provides two register banks to support fast context switching for interrupts and

system calls, with each bank containing sixteen 32-bit registers. The CPP core itself is capable of 45 million instructions per second (MIPS) when run at 40 MHz. The variable-length (16- and 32-bit) instruction set encodings offer the unique advantage of superior code density without sacrificing performance.

Moving forward, microprocessor activities at Bell Labs are focusing on the key applications within the communications industry. Foremost among these activities is the continued development of Lucent's successful DSP chips, a close cousin to the microprocessor. The DSP1600 family of devices continues to be a leader in performance, power, and cost. The new DSP16000 family promises continued expansion with an even higher level of performance. Building on Bell Labs history of innovation, on-going work is focused on defining the architectures and implementations required to support the rapid increase in capability needed for communications systems of the future.

Acknowledgments

We would like to thank Doug Haggan and Bill Troutman for contributing the panel concerning the Intel 4004 and Figures 2 and 3. We would also like to thank Jim Boddie and Bob Cutler for their comments on the draft.

*Trademarks

1-2-3 is a registered trademark of Lotus Development Corporation.

86-DOS is a trademark and CP/M is a registered trademark of Digital Research, Inc.

Altair 8800 and Altair 6800 are trademarks of MITS Corporation.

Alpha and VAX are trademarks of Digital Equipment Corporation.

Apple and Macintosh are registered trademarks of Apple Corporation.

Clipper is a trademark of Computer Associates International, Inc.

Hobbit is a trademark of the Saul Zaentz Company dba Tolkien Enterprises.

iAPX432 and MMX are trademarks and Intel, MCS, and Pentium are registered trademarks of Intel Corporation.

Java is a trademark of Sun Microsystems.

MIPS is a registered trademark of MIPS Computer Systems, Inc.

Nintendo Entertainment System (NES) and Super NES are trademarks of Nintendo of America, Inc.

PageMaker is a registered trademark of Adobe Systems, Inc.

PA-RISC is a registered trademark of Hewlett-Packard Company.

PlayStation is a trademark of Sony Computer Entertainment Inc.

PowerPC is a trademark and OS/2 is a registered trademark of International Business Machines Corporation.

Rubylith is a registered trademark of Diagravure Film Manufacturing Corporation.

Saturn is a trademark of Sega of America, Inc.

Scelbi-8H is a trademark of Scelbi Computers.

Silent 700 is a trademark of Texas Instruments.

UltraSPARC is a trademark and SPARC and SuperSPARC are registered trademarks of SPARC International.

UNIX is a registered trademark of The Open Group.

VisiCalc is a registered trademark of Personal Software, Inc.

Windows is a trademark and Microsoft, MS-DOS, Windows NT, and XENIX are registered trademarks of Microsoft Corporation.

WordPerfect is a registered trademark of Corel Corporation.

References
1. N. Tredennick, "Microprocessor-Based Computers," *Computer,* Vol. 29, No. 10, Oct. 1996, pp. 27–37.
2. *Microprocessor Report,* Vol. 10, No. 10, Aug. 5, 1996, pp. 9–13, 24.
3. M. S. Malone, *The Microprocessor: A Biography,* Springer-Verlag, New York, 1995.
4. "Triumph of the Nerds," narrated by Robert X. Cringely, National PBS Broadcast, June 12, 1996, 8:00 p.m. ET.
5. Gary W. Boone, "Variable Function Programmed Calculator," U.S. Patent 4,074,351, first filed July 19, 1971, issued Feb. 14, 1978.
6. Gilbert P. Hyatt, "Single Chip Integrated Circuit Computer Architecture," U.S. Patent 4,942,516, first filed Nov. 24, 1969, issued July 17, 1990.
7. http://www.ti.com/corp/docs/history/hist_tabs.htm
8. M. Shima, F. Faggin, and R. Ungermann, "Z-80 Chip Heralds Third Microprocessor Generation," *Electronics,* Vol. 49, No. 17, Aug. 19, 1976, pp. 89–93.
9. Carver Mead and Lynn Conway, *Introduction to VSLI Systems*, Addison-Wesley, Menlo Park, California, 1980.

10. http://www.intel.com/intel/museum/25anniv/html/hof/techspecs.htm
11. S. Kelly-Bootle and R. Fowler, *68000, 68010, 68020 Primer,* Howard Sams, Co., Indianapolis, Indiana, 1985, p. 49.
12. M. G. H. Katevenis, "Reduced Instruction Set Computer Architecture," Report No. UCB/CSD83/141, University of California, Berkeley, Oct. 1983.
13. D. Patterson, "Reduced Instruction Set Computers," *Communications of the Association for Computing Machinery,* Vol. 28, No. 1, Jan. 1985, p. 14.
14. J. McKevitt and J. Bayliss, "New Options from Big Chips," *IEEE Spectrum,* Vol. 16, No. 3, Mar. 1979, p. 33.
15. F. Faggin, "How VLSI Impacts Computer Architecture," *IEEE Spectrum,* Vol. 15, No. 5, May 1978, pp. 28–31.
16. John L. Hennessy and David A. Patterson, *Computer Architecture: A Quantitative Approach,* Morgan Kaufman Publishers, Inc., San Mateo, California, 1990.
17. J. E. Smith and S. Weiss, "PowerPC601 and Alpha21064: A Tale of Two RISCs," *Computer,* Vol. 27, No. 6, June 1994, pp. 46–58.
18. *Microprocessor Report,* Vol. 11, No. 5, Apr. 1997, p. 23.
19. F. Faggin, M. Hoff, S. Mazor, and M. Shima, "The History of the 4004," *IEEE Micro,* Vol. 16, No. 6, Dec. 1996, pp. 10–20.
20. "Finding A Beginning," Special Issue: The 30th Anniversary of the Integrated Circuit, *EE Times,* Issue No. 503A, Sept. 1988, pp. 14–24.
21. J. Kreiling, "The Mighty Micro—What It Is and How It Works," *Bell Laboratories Record,* Vol. 59, No. 3, Mar. 1981, pp. 72–74.

MICHAEL R. BETKER is a technical manager in the Processor Architecture Department of Lucent's Microelectronics Group in Allentown, Pennsylvania. He is responsible for future digital signal processor architectures to support the needs of the Wireless and Multimedia organization in Microelectronics. Before his assignment in Allentown, he was part of the BellMac-32 design group in Holmdel and subsequently a lead designer on the WE32100. He was also involved in future development of Hobbit microprocessors prior to working on the team responsible for developing the CPP microprocessor. Mr. Betker earned M.S. and B.S. degrees in computer engineering from the University of Michigan at Ann Arbor.

JOHN S. FERNANDO is a member of technical staff in the Processor Architecture Department of Lucent's Microelectronics Group in Allentown, Pennsylvania. He is responsible for developing digital signal processor architectures. He holds a Ph.D. in computer science from the University of California at Los Angeles, an M.S.E.E. from the University of Texas at Austin, and a B.Sc. in engineering from the University of Sri Lanka. Dr. Fernando's paper "A Microcomputer-based Interactive Transmission Line Simulator," published several years ago in IEEE Transactions on Education, won an Outstanding Transactions Paper Award from the IEEE.

SHAUN P. WHALEN is a distinguished member of technical staff in the Processor Architecture Department of Lucent's Microelectronics Group in Allentown, Pennsylvania. He is responsible for developing digital signal processor and multi-chip unit core architectures, on-chip debugging architectures, and software and hardware development tools. Mr. Whalen has an M.S.E.E. from the University of California at Berkeley and a B.S.E.E. from the University of Notre Dame in Indiana. ◆