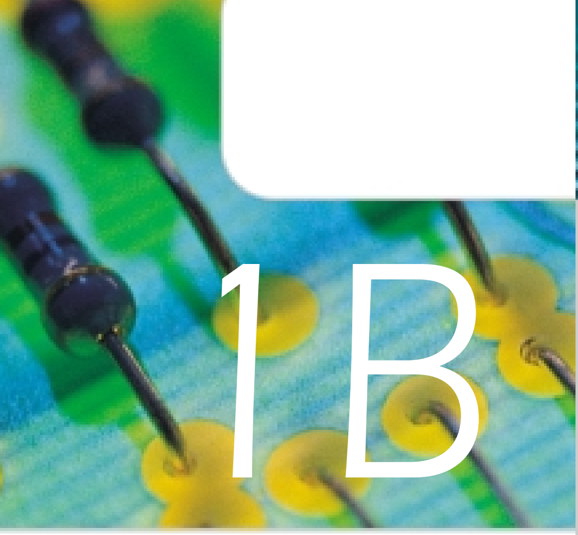


MODULE

HISTORY OF COMPUTERS AND THE INTERNET

OUTLINE

WHAT YOU'LL LEARN . . .

After reading this module, you will be able to:

1. Define the term "electronics" and describe some early electronic devices that helped launch the computer industry.
2. Discuss the role that the stored-program concept played in launching the commercial computer industry.
3. List the four generations of computer technology.
4. Identify the key innovations that characterize each generation.
5. Explain how networking technology and the Internet has changed our world.
6. Discuss the lessons that can be learned from studying the computer's history.

What would the world be like if the British had lost to Napoleon in the battle of Waterloo, or if the Japanese had won World War II? In *The Difference Engine,* authors William Gibson and Bruce Sterling ask a similar question: What would have happened if nineteenth-century inventor Charles Babbage had succeeded in creating the world's first automatic computer? (Babbage had the right idea, but the technology of his time wasn't up to the task.) Here is Gibson and Sterling's answer: with the aid of powerful computers, Britain becomes the world's first technological superpower. Its first foreign adventure is to intervene in the American Civil War on the side of the U.S. South, which splits the United States into four feuding republics. By the mid-1800s, the world is trying to cope with the multiple afflictions of the twentieth century: credit cards, armored tanks, and fast-food restaurants.

Alternative histories are fun, but history is serious business. Ideally, we would like to *learn* from the past. Not only do historians urge us to study history, but computer industry executives also say that knowledge of the computer's history gives them an enormous advantage. In its successes and failures, the computer industry has learned many important lessons, and industry executives take these to heart.

Although the history of analog computers is interesting in its own right, this module examines the chain of events that led to today's digital computers. You'll begin by looking at the computing equivalent of ancient history, including the first mechanical calculators and their huge, electromechanical offshoots that were created at the beginning of World War II. Next, you'll examine the technology—electronics—that made today's computers possible, beginning with what is generally regarded to be the first successful electronic computer, the ENIAC of the late 1940s. You'll then examine the subsequent history of electronic digital computers, divided into four "generations" of distinctive—and improving—technology. The module concludes by examining the history of the Internet and the rise of electronic commerce.

STEPS TOWARD MODERN COMPUTING

Today's electronic computers are recent inventions, stemming from work that began during World War II. Yet the most basic idea of computing—the notion of representing data in a physical object of some kind, and getting a result by manipulating the object in some way—is very old. In fact, it may be as old as humanity itself. Throughout the ancient world, people used devices such as notched bones, knotted twine, and the abacus to represent data and perform various sorts of calculations (see Figure 1B.1).

First Steps: Calculators

During the sixteenth and seventeenth centuries, European mathematicians developed a series of calculators that used clockwork mechanisms and cranks (see Figure 1B.1). As the ancestors of today's electromechanical adding machines, these devices weren't computers in the modern sense. A **calculator** is a machine that can perform arithmetic functions with numbers, including addition, subtraction, multiplication, and division.

The Technological Edge: Electronics

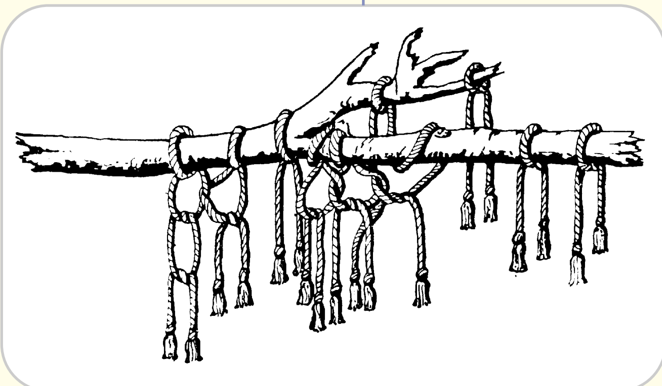
Today's computers are **automatic,** in that they can perform most tasks without the need for human intervention. They require a type of technology that was unimaginable in the nineteenth century. As Figure 1B.1 shows, nineteenth-century inventor Charles Babbage came up with the first design for a

Figure 1B.1

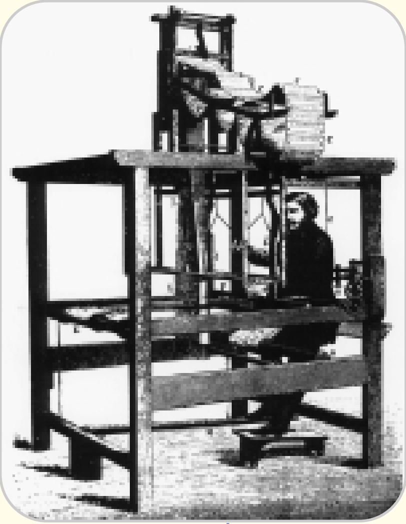
Steps Toward Modern Computing: A Timeline

quipa (15th and 16th centuries) At the height of their empire, the Incas used complex chains of knotted twine to represent a variety of data, including tribute payments, lists of arms and troops, and notable dates in the kingdom's chronicles.

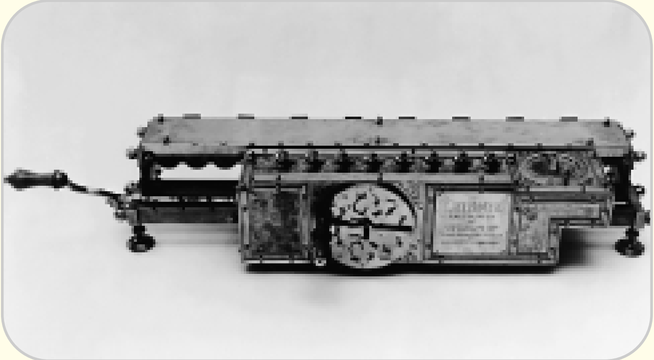
abacus (4000 years ago to 1975) Used by merchants throughout the ancient world. Beads represent figures (data); by moving the beads according to rules, the user can add, subtract, multiply, or divide. The abacus remained in use until a worldwide deluge of cheap pocket calculators put the abacus out of work, after being used for thousands of years.



Jacquard's loom (1804) French weaver Joseph-Marie Jacquard creates an automatic, programmable weaving machine that creates fabrics with richly detailed patterns. It is controlled by means of punched cards.

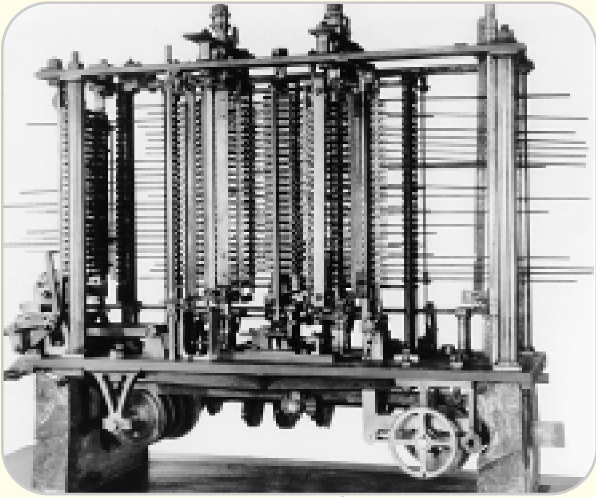


Pascal's calculator (1642) French mathematician and philosopher Blaise Pascal, the son of an accountant, invents an adding machine to relieve the tedium of adding up long columns of tax figures.



Leibniz's calculator (1674) German philosopher Gottfried Leibniz invents the first mechanical calculator capable of multiplication.

Figure 1B.1 (Cont.)

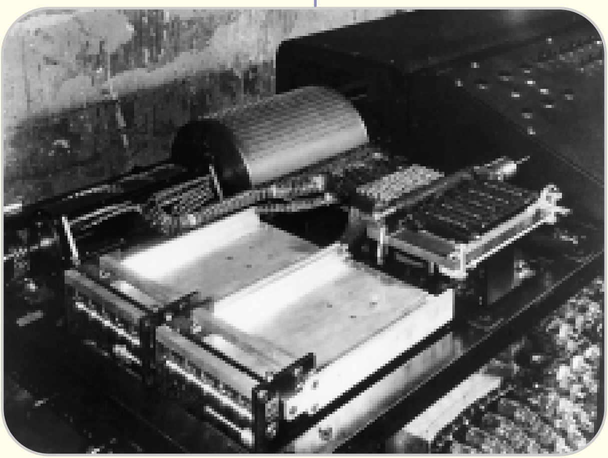


Hollerith's tabulating machine (1890) Created to tally the results of the U.S. Census, this machine uses punched cards as a data input mechanism. The successor to Hollerith's company is International Business Machines (IBM).

Babbage's difference engine (1822) English mathematician and scientist Charles Babbage designs a complex, clockwork calculator capable of solving equations and printing the results. Despite repeated attempts, Babbage was never able to get the device to work.

Mark I (1943) In a partnership with Harvard University, IBM creates a huge, programmable electronic calculator that used electromechanical relays as switching devices.

Zuse's Z1 (1938) German inventor Konrad Zuse creates a programmable electronic calculator. An improved version, the Z3 of 1941, was the world's first calculator capable of automatic operation.



recognizably-modern computer. It would have used a clockwork mechanism, but the technology of his day could not create the various gears needed with the precision that would have been required to get the device to work.

The technology that enables today's computer industry is called electronics. In brief, **electronics** is concerned with the behavior and effects of electrons as they pass through devices that can restrict their flow in various ways. The earliest electronic device, the **vacuum tube,** is a glass tube, emptied of air, in the flow of electrons that can be controlled in various ways. Created by Thomas Edison in the 1880s, vacuum tubes can be used for amplification, which is why they powered early radios and TVs, or switching, their role in computers. In fact, vacuum tubes powered all electronic devices (including stereo gear as well as computers) until the advent of **solid-state devices.** Also referred to as a **semiconductor,** a solid-state device acts like a vacuum tube, but it is a "sandwich" of differing materials that are combined to restrict or control the flow of electrical current in the desired way.

Putting It All Together: The ENIAC

With the advent of vacuum tubes, the technology finally existed to create the first truly modern computer—and the demands of warfare created both the funding and the motivation.

In World War II, the American military needed a faster method to calculate shell missile trajectories. The military asked Dr. John Mauchly (1907–1980) at the University of Pennsylvania to develop a machine for this purpose. Mauchly worked with a graduate student, J. Presper Eckert (1919–1995), to build the device. Although commissioned by the military for use in the war, the ENIAC was not completed until 1946, after the war had ended (see Figure 1B.2).

Although it was used mainly to solve challenging math problems, **ENIAC** was a true programmable digital computer rather than an electronic calculator. One thousand times faster than any existing calculator, the ENIAC gripped the public's imagination after newspaper reports described it as an "Electronic Brain." The ENIAC took only 30 seconds to compute trajectories that would have required 40 hours of hand calculations.

The Stored-Program Concept

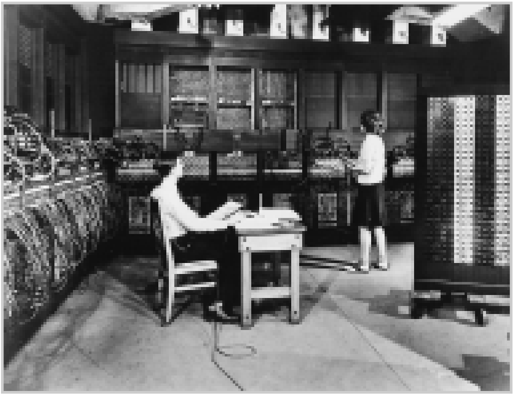


Figure 1B.2

Using 17,480 vacuum tubes, ENIAC was a true programmable digital computer that was one thousand times faster than any existing calculator.

ENIAC had its share of problems. It was frustrating to use because it wouldn't run for more than a few minutes without blowing a tube, which caused the system to stop working. Worse, every time a new problem had to be solved, the staff had to enter the new instructions the hard way: by rewiring the entire machine. The solution was the stored-program concept, an idea that occurred to just about everyone working with electronic computers after World War II.

With the **stored-program concept,** the computer program, as well as data, is stored in the computer's memory. One key advantage of this technique is that the computer can easily go back to a previous instruction and repeat it. Most of the interesting tasks that today's computers perform stem from repeating certain actions over and over. But the most important advantage is convenience. You don't have to rewire the computer to get it to do something different. Without the stored-program concept, computers would have remained tied to specific jobs, such as cranking out ballistics tables. All computers that have been sold commercially have used the stored-program concept.

Table 1B.1

The Generations of Computer Development

Generation	Years	Circuitry	Characterized by
First	1950s	Vacuum tubes	Difficult to program; used only machine language
Second	Early 1960s	Transistors	Easier to program (high-level languages); could work with business tabulating machines; cheaper
Third	Mid-1960s to 1970s	Integrated circuits	Timesharing, minicomputer (SSI, MSI, LSI)
Fourth	Mid-1970s to Present	VLSI and the Microprocessor	Personal computer; graphical user; user interface; LANs; Internet

THE COMPUTER'S FAMILY TREE

The PC that's sitting on your desk is, in many respects, a direct descendent of ENIAC-inspired research, including the stored-program concept. Of course, your computer is thousands of times faster and thousands of times less expensive than its room-filling, electricity-guzzling predecessors. When we're talking about a PC, the "computer" is the microprocessor chip, which is about the size of a postage stamp and consumes less energy than one of the desk lamps in ENIAC's operating room. How was this amazing transformation achieved?

Today's computers weren't achieved in a gradual, evolutionary process, but rather by a series of technological leaps, each of which was made possible by major new developments in both hardware and software. To describe the stage-by-stage development of modern computing, computer scientists and historians speak of computer generations. Each generation is characterized by a certain level of technological development. Some treatments of this subject assign precise dates to each generation, but this practice overstates the clarity of the boundary between one generation and the next. Table 1B.1 introduces the four generations of computing technology. In subsequent sections, you'll learn about each in more detail.

The First Generation (1950s)

Until 1951, electronic computers were the exclusive possessions of scientists, engineers, and the military. No one had tried to create an electronic digital computer for business. And it wasn't much fun for Eckert and Mauchly, the first to try. When the University of Pennsylvania learned of their plans to transform ENIAC into a commercial product, University officials stated that the university owned the duo's patent. Eckert and Mauchly resigned to form their own company, the Eckert-Mauchly Computer Company, and landed a government grant to develop their machine. They underestimated the amount of effort involved, however, and would not have delivered the computer if they hadn't been bailed out by Remington Rand, a maker of electric shavers. With Rand's financial assistance, Eckert and Mauchly delivered the first UNIVAC to the U.S. Census Bureau in 1951 (see Figure 1B.3).

Figure 1B.3

Eckert and Mauchly delivered the first UNIVAC to the U.S. Census Bureau in 1951. UNIVAC gained fame when it correctly predicted the winner of the 1952 U.S. presidential election, Dwight Eisenhower.

UNIVAC gained fame when it correctly predicted the winner of the 1952 U.S. presidential election, Dwight Eisenhower. Since then, computers have been used to predict the winners in every presidential election.

From today's perspective, first-generation computers are almost laughably primitive. For input, punched cards were used, although UNIVAC could also accept input on magnetic tape. Power-hungry vacuum tubes provided the memory (see Figure 1B.4). The problem with vacuum tubes was that they failed frequently, so first-generation computers were down (not working) much of the time.

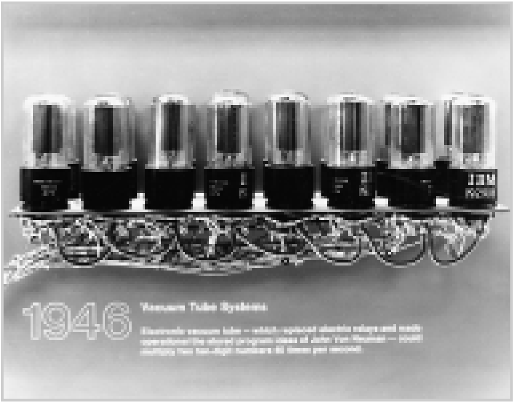


Figure 1B.4

The first generation of computers used vacuum tubes. Vacuum tubes failed frequently, so first-generation computers did not work most of the time.

For all the limitations of first-generation technology, UNIVAC was a much more modern machine than ENIAC. Because it used fewer vacuum tubes than ENIAC, it was far more reliable. It employed the stored-program concept, provided a supervisory typewriter for controlling the computer, and used magnetic tapes for unlimited storage. Because the stored-program feature enabled users to run different programs, UNIVAC is considered to be the first successful **general-purpose computer.** A general-purpose computer can be used for scientific or business purposes, depending on how it is programmed.

Although the stored-program concept made first-generation computers easier to use, they had to be programmed in **machine language,** which is composed of the numbers 0 and 1 because electronic computers use the binary numbering system, which contains only 0 and 1. People often find binary numbers difficult to read. Moreover, each type of computer has a unique machine language, which is designed to communicate directly with the processor's **instruction set,** the list of operations it is designed to carry out. Because machine language was difficult to work with, only a few specialists understood how to program these early computers.

Figure 1B.5

IBM's first commercial computer, the 701, wasn't popular because it didn't work with IBM's own punched-card equipment.

Realizing that Rand's new computers posed a threat to its core business, IBM reacted quickly. In 1953, the company announced its first commercial computer, the IBM 701, but it wasn't popular because it didn't work with IBM's own punched-card equipment (see Figure 1B.5). The 701 was quickly followed by the highly-successful (and more user-friendly) IBM 650, which interfaced with the most widely-used punched-card technology in the world. Thanks to IBM's aggressive sales staff, IBM sold over a thousand 650s in the first year of the computer's availability.

Destinations

Explore the history of computing visually at The History of Computing, an outstanding Web presentation created by the Institute of Electrical and Electronic Engineers (IEEE).

The Second Generation (Early 1960s)

First-generation computers were notoriously unreliable, largely because the vacuum tubes kept burning out. To keep the ENIAC running, for example, students with grocery carts full of tubes were on hand to change the dozens that would fail during an average session. But a 1947 Bell Laboratories invention, the **transistor,** changed the way computers were built, leading to the second generation of computer technology. A transistor is a small electronic device that, like vacuum tubes, can be used to control the flow of electricity in an electronic circuit, but at a tiny fraction of the weight, power consumption, and heat output of vacuum tubes. Because second-generation computers were created with transistors instead of vacuum tubes, these computers were faster, smaller, and more reliable than first-generation computers (see Figure 1B.6).

Second-generation computers looked much more like the computers we use today. Although they still used punched cards for input, they had printers, tape storage, and disk storage. In contrast to the first-generation computer's reliance on cumbersome machine language, the second generation saw the development of the first **high-level programming languages,** which are much easier for people to understand and work with than machine languages. A high-level programming language enables the programmer to write program instructions using English-sounding commands and Arabic numbers. Also, unlike assembly language, a high-level language is not machine-specific. This makes it possible to use the same program on computers produced by different manufacturers. The two programming languages introduced during the second generation, Common Business-Oriented Language (COBOL) and Formula Translator (FORTRAN), remain among the most widely-used programming languages even today. COBOL is preferred by businesses, and FORTRAN is used by scientists and engineers.

Figure 1B.6

The transistor heralded the second generation of computers.

A leading second-generation computer was IBM's fully transistorized 1401, which brought the mainframe computer to an increasing number of businesses. (A mainframe computer is a large, expensive computer designed to meet all of an organization's computing needs.) The company shipped more than 12,000 of these computers. A sibling, the 1620, was developed for scientific computing and became the computer of choice for university research labs.

In business computing, an important 1959 development was General Electric Corporation's **Electronic Recording Machine Accounting (ERMA),** the first technology that could read special characters. Banks needed this system to handle the growing deluge of checks. Because ERMA digitizes checking account information, it has helped to lay the foundation for electronic commerce (e-commerce).

In 1963, an important development was the **American Standard Code for Information Interchange (ASCII),** a character set that enables computers to exchange information and the first computer industry standard. Although ASCII didn't have much of an impact for 15 years, it would later help to demonstrate the importance of standardization to industry executives.

In 1964, IBM announced a new line of computers called System/360 that changed the way people thought about computers. An entire line of **compatible computers** (computers that could use the same programs and peripherals), System/360 eliminated the distinction between computers designed primarily for business and those designed primarily for science. The computer's instruction set was big enough to encompass both uses.

Figure 1B.7

Early second-generation computers were frustrating to use because they could run only one job at a time. Users had to give their punched cards to computer operators, who would run their program and then give the results back to the user.

The Third Generation (Mid-1960s to Mid-1970s)

It's possible to separate the first and second computer generations on neat, clean technological grounds: the transition from the vacuum tube to the transistor. The transition to the third generation isn't quite so clear-cut because many key innovations were involved.

One key innovation was **timesharing.** Early second-generation computers were frustrating to use because they could run only one job at a time. Users had to give their punched cards to computer operators, who would run their program and then give the results back to the user (see Figure 1B.7). This technique,

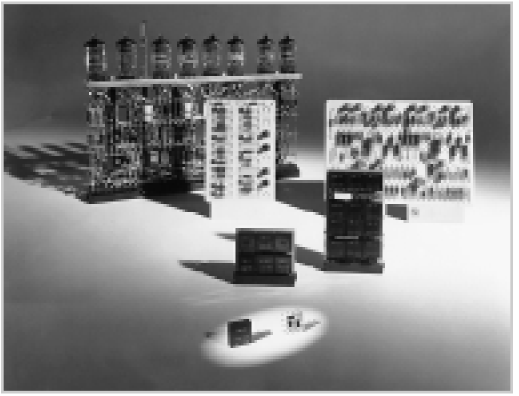


Figure 1B.8

Integrated chips are shown here with first-generation vacuum tubes and second-generation transistors.

called **batch processing,** was time-consuming and inefficient. In timesharing, however, the computer is designed so that it can be used by many people simultaneously. They access the computer remotely by means of **terminals,** control devices equipped with a video display and keyboard. In a properly-designed timesharing system, users have the illusion that no one else is using the computer.

In the third generation, the key technological event was the development of computers based on the **integrated circuit (IC),** which incorporated many transistors and electronic circuits on a single wafer or chip of silicon (see Figure 1B.8). Invented by Jack St. Clair Kirby and Robert Noyce in 1958, integrated circuits promised to cut the cost of computer production significantly because ICs could duplicate the functions of transistors at a tiny fraction of a transistor's cost. The earliest ICs, using a technology now called **small-scale integration (SSI),** could pack up to 10 to 20 transistors on a chip. By the late 1960s, engineers had achieved **medium-scale integration (MSI),** which placed between 20 and 200 transistors on a chip. In the early 1970s, **large-scale integration (LSI)** was achieved, in which a single chip could hold up to 5,000 transistors.

Integrated circuit technology unleashed a period of innovation in the computer industry that is without parallel in history. By the second generation, scientists knew that more powerful computers could be created by building more complex circuits. But because these circuits had to be wired by hand, these computers were too complex and expensive to build. With integrated circuits, new and innovative designs became possible for the first time.

With ICs on the scene, it was possible to create smaller, inexpensive computers that more organizations could afford to buy. Mainframe computer manufacturers such as IBM, however, did not perceive that this market existed. In the first of two key events that demonstrated the inability of large companies to see new markets, the mainframe computer manufacturers left the market for smaller computers open to new, innovative firms. The first of these was Digital Electronic Corporation (DEC), which launched the minicomputer industry. (A minicomputer is smaller than a mainframe and is designed to meet the computing needs of a small- to mid-sized organization or a department within a larger organization.)

Figure 1B.9

DEC's first commercially-available minicomputer, the PDP-8, did not require the attention of a full-time computer operator.

DEC's pioneering minicomputers used integrated circuits to cut down costs. Capable of fitting in the corner of a room, the PDP-8 (a 1965 model) did not require the attention of a full-time computer operator (see Figure 1B.9). In addition, users could access the computer from different locations in the same building by means of timesharing. This minicomputer's price tag was about one-fourth the cost of a traditional mainframe. For the first time, medium-sized companies (as well as smaller colleges and universities) could afford computers.

By 1969, so many different programming languages were in use that IBM decided to unbundle its systems and sell software and hardware separately. Before that time, computer manufacturers received software that was "bundled" (provided) with the purchased hardware. Now buyers could obtain software from sources other than the hardware manufacturer, if they wished. This freedom launched the software industry.

The minicomputer industry strongly promoted standards, chiefly as a means of distinguishing their business practices from mainframe manufacturers. In the mainframe industry, it was a common practice to create a **proprietary architecture** (also called a **closed architecture**) for connecting computer devices. In a proprietary architecture, the company uses a secret technique to define how the various computer components connect. Translation? If you want a printer, you have to get it from the same company that sold you the computer. In contrast, most minicomputer companies stressed **open architecture.** In open architecture designs, the various components connect according to nonproprietary, published standards. Examples of such standards are the RS-232c and Centronics standards for connecting devices such as printers.

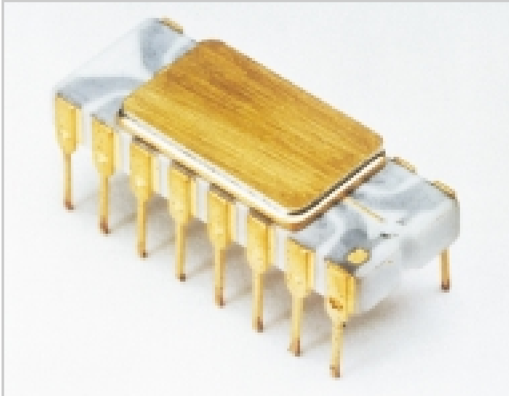


Figure 1B.10

The Intel 4004, the world's first microprocessor.

The Fourth Generation (1975 to the Present)

As the integrated circuit revolution developed, engineers learned how to build increasingly more complex circuits on a single chip of silicon. With **very-large-scale integration (VLSI)** technology, they could place the equivalent of more than 5,000 transistors on a single chip—enough for a processing unit. Inevitably, it would occur to someone to try to create a chip that contained the core processing circuits of a computer.

In the early 1970s, an Intel Corporation engineer, Dr. Ted Hoff, was given the task of designing an integrated circuit to power a digital watch. Previously, these circuits had to be redesigned every time a new model of the watch appeared. Hoff decided that he could avoid costly redesigns by creating a tiny computer on a chip. The result was the Intel 4004, the world's first **microprocessor** (see Figure 1B.10). A microprocessor chip holds the entire control unit and arithmetic-logic unit of a computer. Compared to today's microprocessors, the 4004 was a simple device (it had 2,200 transistors). The 4004 was soon followed by the 8080, and the first microcomputers—computers that used microprocessors for their central processing unit (CPU)—soon appeared. (The central processing unit processes data.)

Destinations

Learn more about the people who created the personal computer industry at "Triumph of the Nerds," a Public Broadcasting System (PBS) Web site created as a companion for the PBS documentary with the same title **(http://www.pbs.org/nerds).**

Repeating the pattern in which established companies did not see a market for smaller and less expensive computers, the large computer companies considered the microcomputer nothing but a toy. They left the market to a host of startup companies. The first of these was MITS, an Arizona-based company that marketed a microcomputer kit. This microcomputer, called the Altair, used Intel's 8080 chip.

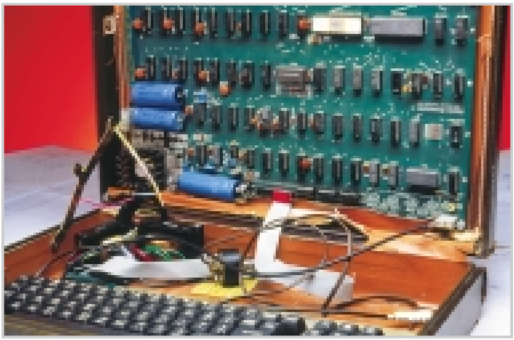


Figure 1B.11

The Apple I was intended for hobbyists, but the experience Apple gained in building it led to the highly-successful Apple II.

In the mid-1970s, computer hobbyists assembled microcomputers from kits or from secondhand parts purchased from electronics suppliers. However, two young entrepreneurs, Steve Jobs and Steve Wozniak, dreamed of creating an "appliance computer." They wanted a microcomputer so simple that you could take it out of the box, plug it in, and use it, just as you would use a toaster oven. Jobs and Wozniak set up shop in a garage after selling a Volkswagen for $1,300 to raise the needed capital. They founded Apple Computer, Inc., in April 1977. Its first product, the Apple I, was a processor board intended for hobbyists, but the experience the company gained in building the Apple I led to the Apple II computer system (see Figure 1B.11).

Figure 1B.12

The first IBM PC was released in 1981. Intel provided the microprocessor chip and Microsoft Corporation provided the operating system.

The Apple II was a huge success. With a keyboard, monitor, floppy disk drive, and operating system, the Apple II was a complete microcomputer system, based on the Motorola 6502 microprocessor. Apple Computer, Inc. soon became one of the leading forces in the microcomputer market, making millionaires out of Jobs, Wozniak, and other early investors. The introduction of the first electronic spreadsheet software, VisiCalc, in 1979 helped convince the world that these little microcomputers were more than toys. Still, the Apple II found its greatest market in schools and homes, rather than in businesses.

In 1980, IBM decided that the microcomputer market was too promising to ignore and contracted with Microsoft Corporation to write an operating system for a new microcomputer based on the Intel 8080. (An *operating system* is a program that integrates and controls the computer's internal functions.) The IBM Personal Computer (PC), with a microprocessor chip made by Intel Corporation and a Microsoft operating system called MS-DOS, was released in 1981 (see Figure 1B.12). Based on the lessons learned in the minicomputer market, IBM adopted an open architecture model for the PC (only a small portion of the computer's built-in startup code was copyrighted). IBM expressly invited third-party suppliers to create accessory devices for the IBM PC, and the company did not challenge competitors who created **IBM-compatible** computers (also called **clones**), which could run any software developed for the IBM PC. The result was a flourishing market, to which many hardware and software companies made major commitments.

IBM's share of the PC market soon declined. The decline was partly due to stiff competition from clone makers, but it was also due to IBM management's insistence on viewing the PC as something of a toy, used chiefly as a means of introducing buyers to IBM's larger computer systems. Ironically, thanks to IBM's reputation among businesses, the IBM PC helped to establish the idea that a PC *wasn't* just a toy or an educational computer, but could play an important role in a business.

Techtalk

look and feel
The on-screen visual ("look") and user experience ("feel") aspects of a computer program. Some software publishers claim that a program's "look and feel" are copyrightable, but courts have had a tough time distinguishing between truly original features and those that are in widespread usage (and therefore not subject to copyright protection). In 1988, Apple Computer sued Microsoft Corporation, alleging that Microsoft Windows infringed on the "look and feel" of the Macintosh interface. After six years of litigation, a Federal court ruled in Microsoft's favor.

The Apple II and IBM PC created the personal computer industry, but they also introduced a division that continues to this day. Because software must be tailored to a given processor's instruction set, software written for one type of machine cannot be directly run on another type. Apple chose Motorola processors for its line of computers, while IBM chose Intel. Today's PCs use advanced Intel microprocessors; the Apple II's successor, the Macintosh, uses PowerPC chips provided by Motorola.

Why were the Apple II and IBM PC so successful? Part of the reason was attributable to the lessons taught by the minicomputer industry. Computer buyers don't like it when manufacturers use proprietary protocols in an attempt to force them to buy the same brand's accessories. Both the Apple II and IBM PC were open architecture systems that enabled users to buy printers, monitors, and other accessories made by third-party companies. Although an open-architecture strategy loses some business initially, in the end it benefits a company because it promotes the growth of an entire industry focused around a given company's computer system. As more software and accessories become available, the number of users grows—and so do the profits.

The first microcomputers weren't easy to use. To operate them, users had to cope with the computer's **command-line user interface.** (A **user interface** is the means provided to enable users to control the computer.)

In a command-line interface, you must type commands to perform such actions as formatting a disk or starting a program. Although the Apple II and IBM PC were popular, computers would have to become easier to use if they were to become a common fixture in homes and offices. That's why the **graphical user interface (GUI)** was such an important innovation.

The first GUI was developed at Xerox Corporation's Palo Alto Research Center (PARC) in the 1970s. In a graphical user interface, users interact with programs that run in their own sizeable windows. Using a mouse (also developed at PARC), they choose program options by clicking symbols (called icons) that represent program functions. Within the program's workspace, users see their document just as it would appear when printed on a graphics-capable printer. To print these documents, PARC scientists also developed the laser printer.

It's difficult to underestimate the contribution that PARC scientists made to computing. Just about every key technology that we use today, including Ethernet local area networks (see Module 6B), stems from PARC research. But Xerox Corporation never succeeded in capitalizing on PARC technology, repeating a theme that you've seen throughout this module: big companies sometimes have difficulty perceiving important new markets.

Figure 1B.13

Apple Computer's Macintosh was the first commercial personal computer to offer a PARC-influenced graphical user interface.

The potential of PARC technology wasn't lost on a late-1970s visitor, Apple Computer's Steve Jobs. Grasping instantly what the PARC technology could mean, the brilliant young entrepreneur returned to Apple and bet the company's future on a new, PARC-influenced computer called the Macintosh. In 1984, Apple Computer released the first Macintosh, which offered all the key PARC innovations, including on-screen fonts, icons, windows, mouse control, and pull-down menus (see Figure 1B.13). Apple Computer retained its technological leadership in this area until Microsoft released an improved version of Microsoft Windows in the early 1990s. Windows is designed to run on IBM-compatible computers, which are far more numerous and generally less expensive than Macintoshes. Also showing the influence of PARC innovations, Windows is now the most widely-used computer user interface program in the world (see Figure 1B.14).

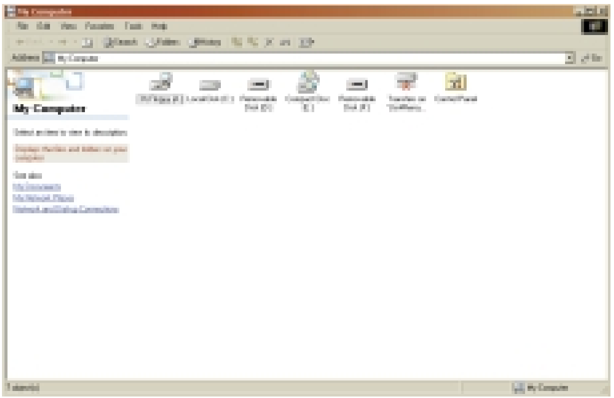


Figure 1B.14

Microsoft Windows 2000 includes the latest version of the world's most popular user interface.

Although fourth-generation hardware has improved at a dizzying pace, the same cannot be said for software. Throughout the fourth generation, programmers have continued to use high-level programming languages. In fact, COBOL, which dates to the dawn of the second generation, is still the most widely-used programming language in the world. High-level programming languages are inefficient, time-consuming, and prone to error. In short, software (not hardware) has slowed the development of the computer industry—at least, until very recently. You will learn about several improvements to computer programming languages, such as object-oriented (OO) programming, a method of dividing programs into reusable components, in Module 8C.

MOVERS & SHAKERS

Amazing Grace

The computer's history isn't an all-male story. Among the many women who have made significant contributions to the computer's development, Admiral Grace Murray Hopper (1906–1992) stands like a giant. She is admired for her considerable technical accomplishments and, perhaps most of all, for her insight, wisdom, and leadership.

Admiral Grace Hopper, the first woman to receive a doctorate in mathematics from Yale University, joined the U.S. Naval Reserve in 1943 and was assigned to Howard Aiken's Mark I computer project at Harvard University. Subsequently, Hopper joined the team that created UNIVAC, the first commercial computer system.

While working with the UNIVAC team in 1952, Hopper invented the first language translator (also called compiler), which for the first time freed programmers from the drudgery of writing computer programs in 1s and 0s. In 1955, Hopper led the development effort that created COBOL, the first high-level programming language that enabled programmers to use familiar English words to describe computer operations. COBOL is still the world's most widely-used programming language.

During her long career, Hopper lectured widely. Her favorite audience was young people, especially in the age group of 17–21. Hopper believed that young people were receptive to the idea of change—a good thing, in Hopper's view, because older people tended to fall into the trap of believing that change isn't possible. Hopper ought to know: experts at first refused to examine her compiler, claiming no such thing was possible. In her retirement speech, Admiral Hopper looked not to the past, but to the future. "Our young people are the future," she said. "We must give them the positive leadership they're looking for."

Hopper's observations inspired generations of computer science students, and seem particularly wise today. Going against the "bigger-must-be-better" philosophy of computer design, Hopper insisted that "we shouldn't be trying for bigger computers, but for more systems of computers." Subsequent years would see the demise of major supercomputer firms as networked computers surpassed the big machines' performance. Hopper also warned that computer systems needed to boil information down to just what's useful, instead of flooding people with more information than they can handle. And once the key information is obtained, Hopper insisted, the job isn't finished. "A human must turn information into intelligence or knowledge. We've tended to forget that no computer will ever ask a new question."

The recipient of more than 40 honorary doctorates from colleges and universities, Hopper received the U.S. Navy's Distinguished Service Medal in a retirement ceremony aboard the U.S.S. Constitution. In recognition of Admiral Hopper's accomplishments, President George Bush awarded her the 1991 National Medal of Technology, the nation's highest honor for technological leadership. Hopper died in 1992 and was buried in Arlington National Cemetery with full military honors.

Admiral Grace Hopper originated COBOL, which is still the world's most widely used programming language.

A FIFTH GENERATION?

If there is a fifth generation, it has been slow in coming. After all, the last one began in 1975. For years, experts have forecast that the trademark of the next generation will be **artificial intelligence (AI),** in which computers exhibit some of the characteristics of human intelligence. But progress towards that goal has been disappointing.

Technologically, we're still in the fourth generation, in which engineers are pushing to see how many transistors they can pack on a chip. This effort alone will bring some of the trappings of AI, such as a computer's capability to recognize and transcribe human speech. Although fourth-generation technology will inevitably run into physical barriers, engineers do not expect to encounter these for many years (perhaps decades).

What appears to truly differentiate the late 1990s from previous years is the rocket-like ascent of computer networking, both at the LAN and WAN levels. Many new homes now include local area networks (LANs) to link the family's several computers and provide all of them with Internet access. At the WAN level, the Internet's meteoric growth is creating a massive public computer network of global proportions, and it has already penetrated close to 50 percent of U.S. households. You'll learn more about the growth and development of the Internet in Module 7A.

Another third-generation innovation was the development of standards for computer networking. Since the late 1960s, the U.S. Advanced Research Projects Agency (ARPA) had supported a project to develop a **wide area network (WAN),** a computer network capable of spanning continents. Headed by Vincent Cerf, this project created a test network, called the ARPANET, that connected several universities that had Defense Department research contracts. The outcome of this project, the Internet, would later rock the world. Interestingly, the ARPANET proved a point that's been seen throughout the history of computing: innovators often cannot guess how people will use the systems they create. ARPANET was designed to enable scientists to access distant supercomputers. Most users, however, viewed it as a communications medium. They developed real-time chatting, electronic mail, and newsgroups. The Internet continues to play an important social role for users.

In 1973, ARPANET fully implemented the **Internet protocols** (also called **TCP/IP**), the standards that enable the Internet to work. Coincidentally, in the same year, Bob Metcalfe and other researchers at Xerox Corporation's Palo Alto Research Center (PARC) developed the standards for a **local area network (LAN),** a direct-cable network that could tie in all computers in a building. Called Ethernet, these standards are now the most widely-used in the world.

THE INTERNET REVOLUTION

As you've learned in this chapter, wartime needs played a crucial role in the computer's development. The same is true of computer networking. In this case, the impetus was the Soviet Union's 1957 launch of the first artificial satellite, Sputnik, during the Cold War. Perceiving a need to play catch-up with Soviet science, the U.S. Congress established the Advanced Research Projects Agency (ARPA). Equipped with generous funds and a mandate to explore cutting-edge science and technology, ARPA was to play a key role in the Internet's development. (You'll learn more about the Internet and its underlying technology in Module 7A; this section recounts the Internet's historical development.)

As Cold War tensions mounted, U.S. military officials became concerned about the survival of its command and control system in the event of a nuclear war. Computer networks were increasingly seen as the command and control system of the future, but the then-existing computer networks were based on a highly-centralized design. A direct hit to the network's central facility would knock out the entire network. A 1962 Rand Corporation study identified a new and unproven networking technology, called packet-switching, as the best bet for creating a decentralized network, one that could keep functioning even if portions of it were knocked out by an enemy hit.

In brief, a **packet-switching network** works by dividing messages up into small-sized units called **packets.** Each packet contains a unit of data as well as information about its origin, its destination, and the procedure to be followed to reassemble the message. While en route, the packets can travel more than one path to reach their destination. If some do not arrive, the receiving computer requests a re-transmission until it has received all of the packets.

In 1968, ARPA awarded a contract to Bolt, Beranak, and Newman (BBN), a technology consulting firm, to build a **testbed** network called ARPANET. In engineering, a testbed is a small-scale version of a product that is developed in order to test its capabilities. Originally, the ARPANET connected only four computers, which were located in California and Utah.

The network grew slowly at first, from an estimated 20 users in 1968 to millions of users today. In the beginning, no one dreamed that the network would one day span the globe. Still, the Internet surprised its creators right away. Originally, the ARPANET's designers thought the network would be used to give researchers remote access to high-powered computers. Instead, ARPANET users figured out how to use the network for communication. The first e-mail program was created in 1972, and was quickly followed by a set of topically-focused mailing lists. (A **mailing list** is an e-mail application in which every member of the list receives a copy of every message sent to the list.) Even though the ARPANET linked researchers at top universities and defense installations, the mailing list topics included many less-than-serious ones, including discussions of science fiction novels, romance and dating, and Star Trek.

The original ARPANET used a set of packet-switching standards that were closely tied to the network's physical medium. In 1973, work began on TCP/IP, a set of standards for packet switching that would enable data to be transferred over virtually any type of physical medium, including cable of all kinds, radio signals, and satellite transmissions. In 1983, every **host** on the ARPANET was required to convert to the TCP/IP standards. In ARPANET terms, a host is a computer that is fully connected to the Internet. (Since many Internet hosts are multi-user machines, the number of people actually using the Internet at a given time is many times larger than the number of hosts.)

By the mid-1970s, local area networks (LANs) were flourishing, and the ARPANET research team realized that the TCP/IP standards had an important strength: they could be used to connect networks as well as hosts. For this reason, Vincent Cert and Bob Kahn, the developers of the TCP/IP standards, began to refer to TCP/IP networks as *internets,* networks capable of linking networks.

Because the ARPANET was fast becoming indispensable for university researchers, the U.S. National Science Foundation (NSF) created a civilian version of ARPANET, called CSNET, in 1981. In 1984, this network was renamed NSFNET. NSF's contribution included construction and maintenance of the network's **backbone,** the long-distance transmission lines that transfer data over interstate and continental distances. Because NSFNET was publicly supported, commercial use of the network was forbidden, but it linked growing numbers of colleges and universities. Meanwhile, the military portion of the ARPANET was separated from the growing public network, and in 1990 the original ARPANET long-distance lines were taken out of service.

In the early 1990s, the term *Internet* was increasingly used to describe the growing network that relied on the NSFNET backbone—and increasingly, regional extensions of the network were being constructed by for-profit firms. In 1995, NSF announced that it would withdraw support for the Internet's backbone network. Commercial providers stepped in to take up the slack, and the restrictions on the Internet's commercial use were finally withdrawn completely.

What has happened since is the most important technological story of the twentieth century. From its origins as a Cold War concept for keeping the military in operation in the event of a nuclear war, the Internet has emerged as an unparalleled public medium for communication and commerce—and

SPOTLIGHT

COMPUTERS AND ELECTIONS: PICKING THE WINNER

On the eve of the 1952 U.S. presidential election, the polls suggested a tight race between Republican hopeful Dwight D. Eisenhower and his Democratic challenger, Adlai Stevenson. On the night of the election, the CBS television network featured a new guest commentator: a UNIVAC computer, which was asked to predict the outcome of the election based on the patterns seen in early returns from the East Coast. At 9 PM Eastern Standard Time, with only 7 percent of the votes tallied, UNIVAC forecasted an Eisenhower landslide. Eisenhower would win 43 states and 438 electoral votes, but Stevenson would win only 5 states and a meager 93 votes. But CBS did not report UNIVAC's prediction. Because most of the polls had called for a close race, the UNIVAC programmers feared they had made a programming error. Instead, they added fudge factors to the program in an attempt to make the results seem more like the close race that the polls were predicting. With the fudge factors added, UNIVAC called the election a toss-up, and that's what CBS viewers heard at 10 PM that evening.

But UNIVAC's program was right. Eisenhower indeed won the election by almost exactly the landslide that UNIVAC had originally predicted: the final tally was 442 electoral votes for Eisenhower, and 89 for Stevenson. "The trouble with machines," CBS commentator Edward R. Murrow later reflected, "is people."

All too often, the trouble with computers is software, too. In a 1981 provincial election in Quebec, Canada, a computer-based election eve forecast gave the nod to the all-but-written-off Union Nationale (UN), a small splinter party that no one thought had the slightest chance of winning the election. Asked to explain the UN's meteoric rise, television commentators came up with a slew of on-the-spot analyses that accounted for the party's sudden popularity. One of them concluded that the experts were wrong to write off the Union Nationale; "the people have spoken," he declared. But there was only one little problem. A software glitch had scrambled the results, leading to a wildly inaccurate prediction. In reality, the Union Nationale was trounced, just as the polls predicted.

As long as the software functions correctly, computers can indeed forecast election results with great accuracy—too great, according to some critics. On the night of the 1980 U.S. presidential election, computer predictions showed incumbent President Jimmy Carter headed for defeat against challenger Ronald Reagan, and the networks declared Reagan the winner. However, they did so before the polls closed on the West Coast, leading some Democrats from the western states to charge that the prediction harmed their chances in state and local elections; with Carter headed for defeat, they argued, Democrats stayed home instead of voting. Carter didn't help matters much by conceding defeat—again, before the West Coast polls closed.

Did computers affect the outcome of the 1980 election? Experts are still divided. Some point out that West Coast Republicans were just as likely as Democrats to skip voting: after all, Reagan had already won. Still, the major networks decided to hold off on releasing the computer projections until the last West Coast polling stations close, and that's their policy to this day.

---

it's changing our world. For example, growing numbers of people use the Internet to **telecommute** to work. In telecommuting, employees work at home, and stay in touch by means of computer-based communications. The Internet is proving indispensable in every conceivable professional field. For example, physicians use the Internet to stay in touch with colleagues around the world, and learn of life-saving new therapies. The growing role of **electronic commerce,** or **e-commerce,** is even changing the way we shop. In e-commerce, people use the Internet to view and order goods and services online.

The Internet has grown and changed in ways that its designers could not anticipate. But what about its effectiveness in its anticipated use: a military situation? In the Gulf War, the U.S. and its allies had a very difficult time knocking out Saddam Hussein's command and control system. After the war's conclusion, military officials learned the reason: Iraq's military was using a TCP/IP-based network—and the network passed its wartime test with flying colors.

LESSONS LEARNED

What's to be learned from the computer's history? Perhaps the most important lesson is an appreciation of the two forces that are currently driving massive changes in our society:

- **Moore's Law** Computers double in power roughly every two years, but cost only half as much.
- **Metcalfe's Law** A network's social and economic value increases steeply as more people connect to it.

These two laws explain why we're witnessing the distribution throughout society of incredibly inexpensive but powerful computing devices, and why the Internet is growing at such an impressive rate. At the same time that computers are rapidly becoming more powerful and less expensive, the rise of global networking is making them more valuable. The combination of these two forces is driving major changes in every facet of our lives.

TAKEAWAY POINTS

- The technology that enables today's computer industry is called electronics. Electronics is concerned with the behavior and effects of electrons as they pass through devices that can restrict their flow in various ways. The vacuum tube was the earliest electronic device.
- The first successful large-scale electronic digital computer, the ENIAC, laid the foundation for the modern computer industry.
- The stored-program concept fostered the computer industry's growth because it enabled customers to change the computer's function easily by running a different program.
- First-generation computers used vacuum tubes and had to be programmed in difficult-to-use machine languages.
- Second-generation computers introduced transistors and high-level programming languages, such as COBOL and FORTRAN.
- Third-generation computers introduced integrated circuits, which cut costs and launched the minicomputer industry. Key innovations included timesharing, wide area networks, and local area networks.
- Fourth-generation computers use microprocessors. Key innovations include personal computers, the graphical user interface, and the growth of massive computer networks.
- An unparalleled public medium for communication and commerce, the Internet has created a massive public computer network of global proportions. It has already penetrated close to 50 percent of U.S. households.
- As computers become more powerful and less expensive, the rise of global networking is making them more valuable. The combination of these two forces is driving major changes in every facet of our lives.

MODULE REVIEW

KEY TERMS AND CONCEPTS

American Standard Code for Information Interchange (ASCII)
artificial intelligence (AI)
automatic
backbone
batch processing
calculator
command-line user interface
compatible computers
electronic commerce or e-commerce
electronics
Electronic Recording Machine Accounting (ERMA)
ENIAC
general-purpose computer
graphical user interface (GUI)
high-level programming languages
host
IBM compatibles, or clones
instruction set
integrated circuit (IC)
Internet protocols, or TCP/IP
large-scale integration (LSI)
local area network (LAN)
machine language
mailing list
medium-scale integration (MSI)
Metcalfe's Law
microprocessor
Moore's Law
open architecture
packet-switching network
packets
proprietary architecture, or closed architecture
semiconductor
small-scale integration (SSI)
solid-state devices
stored-program concept
telecommute
terminals
testbed
timesharing
transistor
user interface
vacuum tubes
very-large-scale integration (VLSI)
wide area network (WAN)

TRUE/FALSE

Indicate whether the following statements are true or false.

1. Today's electronic computers are recent inventions, stemming from work that began during the Korean War.
2. Electronics is the technology that enables today's computer industry.
3. One key advantage of the stored-program concept is that the computer can easily return to a previous instruction and repeat it.
4. Although the stored-program concept made first-generation computers easier to use, they

had to be programmed in machine language, which is composed of the numbers 0 and 1.

5. Power-hungry transistors provided the memory for first-generation computers.
6. A high-level programming language enables programmers to write program instructions using Arabic-sounding commands and Roman numerals.
7. The key event in the third generation was the development of computers based on integrated circuits.
8. The first graphical user interface was developed at Apple Computer.
9. A third-generation innovation was the development of standards for computer networking.
10. The Advanced Research Projects Agency (ARPA), established by the U.S. Congress during the Cold War, played a key role in the Internet's development.

MATCHING

Match each key term from the left column to the most accurate definition in the right column.

_____ 1. calculator
_____ 2. vacuum tube
_____ 3. transistor
_____ 4. stored-program concept
_____ 5. instruction set
_____ 6. timesharing
_____ 7. integrated circuit
_____ 8. microprocessor
_____ 9. Internet protocols
_____ 10. backbone

a. the list of operations a processor is designed to carry out
b. a small, second-generation electronic device that can control the flow of electricity in an electronic circuit
c. a device that contains the entire control unit and arithmetic logic unit of a computer
d. a machine that can perform arithmetic functions
e. the standards that enable the Internet to work
f. a device that incorporates many transistors and electronic circuits on a single chip of silicon
g. the earliest electronic device that powered all electronic devices until the advent of solid-state devices
h. long-distance transmission lines that transfer data over interstate and continental distances
i. enables many people to use a computer simultaneously
j. the idea that the program and data should be stored in memory

MULTIPLE CHOICE

Circle the letter of the correct choice for each of the following.

1. Which of the following was considered the first true programmable digital computer?
   a. UNIVAC
   b. ERMA
   c. ENIAC
   d. Apple II
2. All computers that have been sold commercially have used which of the following?
   a. terminals
   b. transistors
   c. the stored-program concept
   d. vacuum tubes
3. What characterizes first-generation computers?
   a. vacuum tubes and punched cards
   b. magnetic tape and transistors
   c. minicomputers
   d. high-level programming languages
4. What kind of computer can be used for scientific or business purposes?
   a. timesharing computer
   b. general-purpose computer
   c. ENIAC
   d. abacus

5. Which of the following does not apply to high-level programming languages?
   a. They are easier to understand than machine languages.
   b. They are not machine-specific.
   c. They use English-sounding commands.
   d. They are composed entirely of the numbers 0 and 1.
6. What invention enabled developers to create microcomputers?
   a. integrated circuits
   b. transistor
   c. vacuum tube
   d. magnetic disk
7. What are Steve Jobs and Steve Wozniak known for?
   a. the first IBM-compatible computer
   b. UNIVAC
   c. the first Apple computer
   d. the stored-program concept
8. Which of the following is not true of computers as we progress from one generation to the next?
   a. computer size decreases
   b. computer cost decreases
   c. speed of processing increases
   d. memory and storage capacities decrease
9. Which technology describes people using the Internet to view and order goods and services online?
   a. electronic exchange
   b. home shopping network
   c. electronic commerce
   d. telecommuting
10. Which law states that a network's social and economic value increases steeply as more people connect to it?
   a. Moore's Law
   b. Metcalfe's Law
   c. Job's Law
   d. Mauchly's Law

FILL-IN

In the blank provided, write the correct answer for each of the following.

1. Also called a(n) ______________ , a solid-state device acts like a vacuum tube, but it is a "sandwich" of differing materials that combine to restrict or control the flow of electrical current in the desired way.
2. With the ______________ , the computer program, as well as data, is stored in the computer's memory.
3. UNIVAC is considered to be the first successful ______________ .
4. ______________ is composed entirely of the numbers 0 and 1.
5. Second-generation computers used ______________ instead of vacuum tubes and were faster, smaller, and more reliable.
6. COBOL and FORTRAN are examples of ______________ programming languages.
7. The ______________ is a character set enabling computers to exchange information.
8. In a(n) ______________ architecture, a company uses a secret technique to define how the various computer components connect.
9. With ______________ technology, engineers could place the equivalent of more than 5,000 transistors on a single chip.
10. A(n) ______________ network works by dividing messages up into small units called ______________ .

SHORT ANSWER

On a separate sheet of paper, answer the following questions.

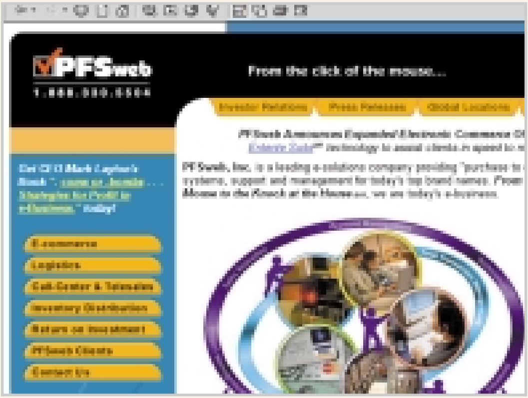
1. Explain why ENIAC is considered the first true programmable digital computer. What kinds of problems did it have?
2. Explain the stored-program concept. How did this concept radically affect the design of computers we use today?
3. What major hardware technology characterized each of the four generations of computers?
4. What are the differences between a command-line interface and a user interface? Which one is easier to use and why?
5. How does a machine language differ from a high-level programming language?
6. What were the various transistor capacities for small-scale, medium-scale, large-scale, and very-large-scale integration?
7. Explain the differences between an open architecture and a proprietary, or closed, architecture.
8. What differentiates the last 10 years of computing technology from the last 60?
9. How did the Cold War contribute to the growth of the Internet revolution?
10. In what ways has the Internet changed the way we work and live?

E-COMMERCE IN ACTION

PFSweb, Inc.

Have you ever purchased music, books, or clothes over the Web? Congratulations! You've participated in e-commerce! "E" what?? E-commerce. The "e" stands for electronic, and "commerce" means business. Doing business online, rather than in a "bricks-and-mortar" store, is what e-commerce is all about. It's taking the traditional buyer-seller relationship and moving it into cyberspace.

Lots of companies are getting into the dance, with hopes that their online stores will generate profits. It's easy for a company to put up a pretty marketing Web site in cyberspace to build their brands and promote their products. But moving it to the next level where customers can actually make purchases is a much trickier dance step. Behind the scenes, the site needs a way to process customer payments, check for fraudulent credit card usage, and get the merchandise shipped from the warehouse without missing a beat. It also needs a way for customers to ask questions—where's the order, how do I return something, and so on.



There's a company you've probably never heard of in Plano, Texas that helps e-commerce companies keep in step with the online buying and selling marketplace. It's called PFSWeb, Inc. The company's job is to orchestrate all the pieces that comprise an e-commerce site so that buying or selling is simple and seamless. Mark Layton, president of the company, has helped hundreds of growing e-commerce companies with their online stores by running all the "behind the scenes" tasks. The company can design Web sites, prepare online catalogs, process payments, check for fraud, calculate taxes, ship merchandise, and more for any size e-commerce site. Pretty much the same activities any physical business must manage if it's going to make money. The only difference is that online, all steps in buying are automated. The people at PFSweb have done their job if the buyer can't tell where the marketing Web site ends and the business transaction side begins. In fact, Mark's company has a saying that pretty much says it all: "From the Click of the Mouse, to the Knock at the House." They'll deliver. Now there's a reason to dance!

What do you think? Describe a recent online purchase you've made. Why did you buy online? Were the buying instructions clear? How did you pay for your purchase? Did the actual merchandise meet your expectations? Why? If you had problems or needed to make a return, how easy was it to take care of it?

WebLink Go to **www.prenhall.com/pfaffenberger** to see the video of Mark Layton and explore the Web.