1 A Look Back

In this chapter...

- Nonelectronic Computers
- Electronic Computers
- Personal Computers
- Workstation Computers
- Graphical User Interface
- The Central Processor Revolution
Would you spend $1,000 for a computer with 8K of random access memory (RAM), a black-and-white monitor, and a keyboard? No, there aren’t any typographical errors in my question. And, no, I didn’t leave out any equipment. I rushed out and bought one and so did thousands of other computer nerds.

Back in the late 1970s, Tandy’s TRS-80, affectionately known as the trash 80, was the hot computer of the day. (Tandy is the parent company of Radio Shack.) The computer itself was in the keyboard and the only permanent storage device was an audio-tape recorder—and that was unreliable.

Today no one, including a collector of historical computers, would pay $1,000 for such an underperforming computer that couldn’t run even Windows 2.0. Yet, in its day, the TRS-80 was arguably using the leading PC technology.

We tend to take computer technology for granted. And those of us who lived through and participated in a small way in the computer revolution probably forgot what it was like to crank up the PC and spend the next several minutes trying to load a simple game program from an audio tape into the computer.

In this chapter, I’ll jar a few memories of those who remember the not-so-good old days of early computers and for everyone else provide an entertaining and informative trip that will serve as a foundation to begin to learn about the inner workings of today's computer hardware. You'll learn the origins of:

- Mechanical computers
- Mainframe computers
- Minicomputers
- Personal computers
- Workstation computers
- Microprocessors
REALITY CHECK

Computers can be intimidating, as I discovered when my oldest daughter received her first high school English assignment. This was a time when computer training wasn't part of the daily routine in grammar schools.

Although computers have been in my home since before the IBM personal computer was introduced, my family considered computers a daddy thing. They probably referred to the computer as daddy’s toy when I was out of earshot.

However, the English assignment quickly changed my family’s attitude. My daughter needed an electric typewriter to write her English assignment. My arguments to use the word processor in my computer fell on deaf ears.

So we spent a couple of hundred dollars for an electronic typewriter and the necessary supplies—correction fluid, messy typewriter ribbons, and a dictionary for checking misspelled words. In fact the typewriter still exists today in my attic, nearly brand new and used for one English assignment.

The reality of using a typewriter soon became apparent to my daughter when she produced her first error-free page. It took her nearly an hour to do so. Then she inadvertently ripped the page when pulling it from the typewriter. From that day on the entire family used daddy’s toy whenever they had to write a letter, do a report for school, or play a game. They simply cannot live a day without using a computer. It is safe to say that none of us ever want to go back to a time without computers.

NONELECTRONIC COMPUTERS

Now for a brief test: Define the term computer. Don’t go racking your brain trying to compose an academic definition of a computer because I’m sure you know what a computer is . . . or should I say you can picture a computer in your mind.

For most of us a computer is the box that sits on our desk or on the floor or something we lug around on trips. Beyond that we probably haven’t a notion of how a computer works and the meaning behind the advertising buzzwords that bombard us by those trying to sell us a computer. By the end of this book you’ll be able to cut through the hype and know how modern computer hardware works—and be able to give computer sales representatives a run for their money.

For now think of a computer as a counting machine, a machine that can add, subtract, multiply, and divide numbers—and nothing more. Computers don't seem intimidating when you realize all a computer does is simple math that you learned in elementary school. Even acts performed by computers that seem to appear as near human intelligence are nothing more than grammar school math.
One of the functions of a computer that baffled me when I first began learning about computers in the TRS-80 days is how a computer knows when two things are the same or are different. You've seen this function performed when you enter an ID and password into a computer to access your corporation's network.

At first I thought comparisons involved some abstract scientific formula that was electronically embedded into a chip inside the computer. Then I learned how computers compared two things—by using subtraction.

Here is how this works. Information such as an ID and password are represented as numbers. I'll show you how this is done in the next chapter. The computer, following instructions from a programmer, subtracts the information entered into the keyboard from known information stored inside the computer.

Let's say my ID is 12345. This ID is stored in a database that contains my ID and other employees' IDs. The network administrator, the person responsible for the network, assigns IDs to all employees and enters them into the database. After I enter my ID when I log on to my company's computer network, the computer compares my ID against each ID stored in the database.

In this example, the first ID the computer comes across is 00001. The computer subtracts 12345 from 00001. The result is –12344. However, the computer cares only if the result is zero or not zero because a zero means the ID entered at the keyboard is the same as the ID in the database. A nonzero result means they are different.

In this case, the result is not zero. The ID I entered is different than the first ID in the database. When the computer tries the second ID in the database, which is 12345, the result of subtraction is zero (12345 – 12345 = 0). The IDs match.

Computers Are Not Electronic

I will bet that when I asked you to define a computer you envisioned some type of box filled with electronic components. If so, then you are in the majority of people who associate computers with electronic devices. Actually a computer is anything that can help us count.

Fingers and toes are the computers I (and probably you) used when I taught my daughters how to count. At first glance, these digits may not seem sophisticated since the maximum value that can be counted using your hands is ten. Yet that is not entirely true because it depends on how you represent numbers with those digits.

Time for another quiz. What is the maximum number that you can count with your fingers? You don't need to use any of your lifelines because I'll give you the answer. The answer is 30. Puzzled? Here's how this is done.

Each time you finish counting digits on your right hand, hold up a digit on your left hand. Begin counting again on your right hand. Each digit on your left hand repre-
sents five for a total of 25. You still have five fingers on your right hand extended when you finish counting. This gives you a combined total of 30.

This is more than a dinner table trick because it demonstrates the basic concept of number places that was used in the original mechanical computing device called the abacus. The abacus consists of rows of 10 beads each. Each row represents a number place (see Figure 1-1).

*Figure 1-1*
An abacus is an ancient mechanical computer that used beads to represent digits of a number

**Tech Talk**

Abacus: The first computer that used rows of beads to assist in adding and subtracting numbers.

In 3000 B.C., the abacus was used as a way of extending the use of fingers as a counting tool. The first row in the abacus is similar to our right hand and the second row takes the place of our left hand in our example. However, instead of five digits the abacus uses ten beads. And instead of two hands, the abacus has many rows.

Fundamentally the process is the same. The count begins by moving each bead from one end of the row to the other end. When all the beads are at the other end of the row, one bead in the next row is moved to the other end of the second row. All the
beads in the first row are returned to their original starting position and the beads are counted again. Subtraction is also possible with an abacus by moving the beads in the opposite direction that you moved them when adding numbers together.

A merchant who is proficient using the abacus can be stiff competition for anyone who is using an electronic calculator. The merchant can zip through moving the beads long before the calculator is powered up and the first numbers entered into the keyboard.

The Illusive Dream

While the abacus was an improvement over the fingers, it still left much to be desired, especially when there was a need to perform complex calculations in science and business. A more efficient machine was required.

Leonardo da Vinci was one of the first to recognize this need. In the 16th century, da Vinci drew the first plans for a mechanical calculator that could perform addition and subtraction. However, he never lived long enough to see those plans become reality.

It wasn't until the 17th century that a mechanical adding machine was developed. And we can thank the tax collector for this marvel. Actually, it was the tax collector’s son, Blaise Pascal, a French mathematician who designed and built the first working adding machine called the Pascaline. Pascal built the Pascaline (see Figure 1-2) to help his father collect taxes.

Although the French taxpayers may not have been pleased by the efficiency the Pascaline gave to collecting taxes, the Pascaline did prove to the world that a machine could add and subtract faster and more accurately than people.

The Pascaline used a similar concept as the abacus. However, instead of beads, the Pascaline used a series of wheels interlocked with gears. Each wheel had 10 digits (0 through 9) displayed on the wheel. When the first wheel reached past nine and returned to zero, the second wheel moved to one. A similar effect occurred for each wheel.

Figure 1-2
The Pascaline used interlocking gears to reproduce the basic functionality used in the abacus
Pascal surely impressed his father, but the Pascaline couldn’t overcome a major hurdle that still faces the computing industry. That is, the early versions of any computer are very costly and are not seen as cost-effective.

The Pascaline was hand built and not durable for the daily ordeal of tax collecting. In fact only the inventor himself could keep the Pascaline running when it broke down. The cost of building and maintaining the Pascaline adding machine was dramatically more costly than hiring additional clerks to perform math by hand.

The Pascaline may not have been a business success but it paved the way for the success of computing. The concepts developed by Pascal were used in every mechanical computing device until the middle of the 1960s.

With improvements in manufacturing, the Pascaline gave birth to assorted mechanical adding machines that were widely used in universities, business, and government in the 18th century. However, these machines were limited to addition and subtraction and were unable to handle the sophisticated calculations that were needed to drive the Industrial Revolution in the 19th century.

These calculations required several steps and involved addition and subtraction as well as multiplication and division of very large and very small numbers such as required by bankers when creating a repayment schedule.

In those days, mathematicians created calculation tables that reduced the number of calculations a banker or scientist had to perform by hand. For example, a calculation table (see Figure 1-3) might have the interest payments for a series of interest rates based on one dollar of principal. A banker looked up the table for the appropriate interest paid for a specific interest rate, then manually multiplied the interest payment by the number of dollars in principal of the loan.

**Figure 1-3** This table contains the interest payment for a dollar at various interest rates and was used as a way to speed calculations

<table>
<thead>
<tr>
<th>Years</th>
<th>6%</th>
<th>8%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.060000</td>
<td>1.080000</td>
</tr>
<tr>
<td>2</td>
<td>1.123600</td>
<td>1.166400</td>
</tr>
<tr>
<td>3</td>
<td>1.191016</td>
<td>1.259712</td>
</tr>
<tr>
<td>4</td>
<td>1.262477</td>
<td>1.360489</td>
</tr>
</tbody>
</table>
Calculation tables increase the efficiency of bankers, but calculation tables were not without errors. Calculation tables were created by hand and were error prone. There was still a need for a machine that could perform complex calculations without errors.

Charles Babbage expanded Pascal’s concept of wheels and gears and designed a mechanical computer that could also multiply and divide as well as add and subtract. The machine was called the Difference Engine and the Analytical Engine.

**Tech Talk**

**Difference Engine and the Analytical Engine:** The first mechanical computer that was designed to add, subtract, multiply, and divide, but was never commercially built. Principles used in this computer are still found in modern electronic computers.

As with the Pascaline, Babbage’s design never got off the ground because of cost and the lack of technology to build the engine. The design called for thousands of wheels and gears powered by a railroad engine. Babbage’s mechanical computer would have been the size of a football field—and could perform 60 addition calculations per second. Babbage’s design, however, did become the foundation of today’s general computer.

**Can You Repeat That?**

Pascal and Babbage gave birth to mechanical computers that could perform calculations, but that’s only part of the historic development of computing. People had to perform each calculation themselves rather than having the computer automatically perform repeating calculations.

Let’s say that you wanted to create an identification code that used 10 letters (A through J). How many individual codes can you create using 10 letters? You can answer this question by multiplying 9 times. Here’s how it’s done.

Multiply $10 \times 10$, then multiply the product by 9. Multiply that product by 8 and continue the process until you multiply the final product by 1. Mathematicians call this factorial. The answer is there are 3,628,800 combinations that can be made using 10 letters.

**Tech Talk**

**Factorial:** A mathematical formula used to determine the number of unique combinations of a set of objects such as letters of the alphabet.

You’d need to perform nine separate calculations if you used a mechanical calculator based on Babbage’s concept. This wasn’t efficient because the mechanical calculator could not repeatedly multiply results on its own.
In this example, it would be nice to enter the number of letters in the identification code (10), then press a button that instructs the calculator to automatically determine the number of combinations.

Today’s computers and calculators perform factorial calculations and other similar repeating calculations nearly automatically because those devices store instructions. One of the many challenges facing 19th century inventors was to devise a way for a mechanical calculator to repeat instructions automatically.

The answer loomed in the loom. Joseph-Marie Jacquard, a French weaver at the turn of the 19th century, inadvertently solved this computing problem. Jacquard developed a loom that used holes in cards to control the operation of the loom. Holes punched into a card directed the loom to create a specific design. Each new design required a new card with holes punched according to the design requirements.

The loom could re-create any design automatically as long as Jacquard inserted the corresponding punch card. Although Jacquard’s objective was to run the loom longer hours and employ workers with less skill—and less pay—than weavers could work and get paid, Jacquard inadvertently found the solution to having a mechanical calculator repeat instructions automatically.

It wasn’t until later in the 19th century that Jacquard’s invention was considered applicable to mechanical calculators. Lady Ada Lovelance, a mentor to Babbage, was the first to put forward the idea of using punch cards to instruct a mechanical calculator. Although Babbage’s computer never became fully operational, the idea of using punch cards to instruct mechanical—and later electronic—computers was later a success.

Lovelance’s place in computer history was firmly established when the U.S. Department of Defense named one of its first programming languages—Ada—after her. Many consider Lovelance the first programmer.

**Tech Talk**

**Programming language:** A set of words recognized by a computer and used to instruct a computer to perform specific tasks.

It wasn’t until almost the turn of the 20th century that punch cards took center stage—but it was the railroad and not Lovelance that provided the inspiration. However, punch cards were first used to store information, not instructions.

**Tech Talk**

**Information:** One or more words that are used to describe something.

The U.S. Bureau of the Census needed an efficient way to count people. Herman Hollerith decided punch cards and a punch card reading machine was the answer and won a bid to count the population.
Hollerith invented the punched card tabulating machine based partly on the way tickets were punched by railroad conductors. The railroad devised a clever system to catch would-be fare cheats. A railroad ticket was preprinted with characteristics of their customers such as gender, hair color, eye color, etc. Railroad conductors punched out on the ticket the characteristics that described the customer before giving the customer the ticket. The tickets were called a punched photograph.

Census takers punched specially designed punch cards for the Bureau of the Census that described a person. Hollerith created an electrically (not electronically) operated machine that read and sorted the punch cards and tabulating dials that summarized the count based on characteristics of each person.

If Only I Invested in TMC

Hollerith founded one of the first successful computer manufacturing and servicing companies, the Tabulating Machine Company (TMC). TMC was built on Hollerith’s success with the census and gradually branched out to service the business community.

TMC later evolved into the Computer-Tabulating-Recording Company with the merger of TMC and complimentary businesses a decade into the 20th century. By 1924 the company again had a makeover. This time the Computer-Tabulating-Recording Company was renamed the International Business Machines Corporation (IBM).

The first half of the 20th century is the era of the electro-mechanical accounting machine, which was the improved version of Hollerith’s original tabulating computer. The electro-mechanical accounting machine built by IBM consisted of eight separate components: an accounting machine, card punch, collator, interpreter, reproducer, sorter, summary punch, and verifier.

Each component was a separate machine that performed specific instructions using data that was recorded on thousands of punch cards. Instructions used to control the electro-mechanical accounting machine were programmed by inserting prewired control panels into each component.

Programmers in that era didn’t learn computer languages as they do today. Instead, they plugged wires called patch wires into holes on each component’s control panel based on the task that the programmer wanted the machine to perform.

There were no disk drives, tape drives, floppy disks, or computer networks in those days. Instead a machine room operator had to carry thousands of punch cards encoded with data from component to component. And the whole process came crashing down if the pile of punch cards was dropped.
ELECTRONIC COMPUTERS
........................................

As IBM was providing computing services to governments and businesses using its electro-mechanical accounting machine, a professor and his graduate student at Iowa State University took a unique approach to computing. Instead of using an electro-mechanical technology, they used the new electronic technology.

Dr. John Atanasoff and his student Clifford Berry spent seven years developing an electronic computer to aid students in performing complex physics calculations. They hit many roadblocks along the way. It wasn't until the winter of 1937 that the plan came together over a late night drink in an Illinois tavern.

Atanasoff and Berry figured out how to use vacuum tubes (see "Inside Vacuum Tubes and Transistors") and other electronic components to form logic circuits (equivalent to the central processing unit [CPU] in today's computer) and memory (see Figure 1-4). They also figured out how to use the binary numbering system (see "Computers Count to Two") to represent data and instructions.

Figure 1-4
A vacuum tube is an electronic switch that redirects the flow of electricity using a charged plate

Tech Talk
Vacuum Tube: An electronic switch that uses a charged plate to change the flow of electricity in a circuit.
By 1942, they had a working prototype called the Atanasoff Berry Computer (ABC), which was the first automatic electronic digital computer and the beginning of the electronic computer revolution.

There were two major blunders. First, neither Iowa State University nor Atanasoff and Berry patented the ABC computer. This not only had obvious financial repercussions, it failed to document Atanasoff and Berry’s discovery. Decades of legal battles over who is credited with inventing the electronic computer finally culminated in 1973 when the courts ruled that Atanasoff invented the automatic electronic digital computer.

IBM made the second blunder. Atanasoff tried in vain to interest IBM in ABC computing technology. IBM’s response: “IBM will never be interested in an electronic computing machine.” I guess you can never say never.

Computers Count to Two

You’ve probably heard the term binary used whenever anyone talks about the zeros and ones used to represent data inside a computer. Binary refers to a mathematical system used to count and perform math functions very similar to the method we use to balance our checking account.

Let’s take the mystery out of binary math by first reviewing our common system of mathematics called the decimal system. Our system contains 10 digits—zero through nine—which is referred to as base 10 because there are 10 digits before we must carry over a value.

When we reach the ninth digit, we carry over one value, which gives us 10. Notice the right-most digit is 0, the digit we started with. We count sequentially and when we reach 19—notice the right-most digit is 9—we carry over another value to give us 20.

A similar process occurs within the binary system except it contains only two digits—zero and one. Counting begins with 0 just like in the decimal system, but a value is carried over when the value reaches two because there isn’t a digit 2 in the binary system.

Let’s say we wanted to write the value two in decimal. We simply use the digit 2. However, we’d write 10 to represent the same value in binary. No, this is not 10, although it resembles the value 10 in decimal.

Here’s how it works. We begin counting with 0. Remember, this is binary so we must stop counting when the right-most digit reaches 1, then carry over a value and reset the right digit to 0.
Computers Count to Two (continued)

Don’t feel embarrassed if you are confused by binary math. This baffles even computer science majors until they become use to dealing with the concept of using two digits instead of 10. However, there is a reason computer scientist settled on binary math when dealing with computers.

Computers manipulate information by performing addition and subtraction on data stored inside the computer. Remember, data is represented by a setting of eight switches and each switch can be on or off. Sometimes more than eight switches are used, but don’t concern yourself with that right now.

Binary math lends itself to representing the state of a switch and enables standard mathematical operations to be performed on those values. You can perform the same math using binary values that can be performed using our decimal values.

First Generation of Computers

Atanasoff and Berry’s work became the foundation for Dr. John Mauchly’s efforts in 1946 to solve a perplexing problem for the military—the creation of accurate trajectory tables. Errors in trajectory tables caused bombing mistakes during World War II. Eighty percent of the bombs came outside of 1,000 feet of their targets.

Mauchly and his associates at the University of Pennsylvania developed a vacuum tube-based computer called the Electronic Numerical Integrator and Computer (ENIAC) that dramatically increased the accuracy of trajectory tables.

ENIAC was the fastest computer available and performed 500 multiplication calculations and 5,000 addition calculations per minute using 18,000 vacuum tubes. The ENIAC weighed 30 tons and took up almost an entire floor of an office building. Lights in Philadelphia dimmed whenever the ENIAC was working.

Although the ENIAC advanced computer technology appreciably, some computer technology historians believe Mauchly made one blunder. Instead of using the base-2 binary numbering system as Atanasoff and Berry did, Mauchly used the base-10 decimal numbering system.

Tech Talk

Base-10 numbering system: A system of numbers that uses 10 digits from zero to nine. This is commonly called the decimal system and is the prevalent system used in everyday counting.
The base-10 numbering system didn't compromise the accuracy of the ENIAC results. It did, however, require the use of more vacuum tubes to represent numeric values. For example, 10 vacuum tubes were required to represent the number one in the ENIAC. One vacuum tube represented the same value in the ABC computer using the base-2 (binary) numbering system.

Neither the ABC nor the ENIAC computer was designed for commerce. The ABC computer was used to help students at Iowa State University with complex physics problems, and ENIAC created computerized trajectory tables for the military.

It wasn’t until 1951 that the first commercially available electronic digital computer was introduced and formally began the first generation of computers. Mauchly and his associates at the University of Pennsylvania were enlisted by the Remington-Rand Corporation to develop the Universal Automatic Computer, better known as the UNIVAC I.

UNIVAC I was quickly put to work counting the 1951 census, but was more widely known projecting results of the Presidential election early on election night. With only 5 percent of the votes counted, UNIVAC I awarded the election to Dwight Eisenhower. The projection was broadcast by CBS News, which was the first media outlet to introduce computer technology into the average home four years after Chuck Yeager broke the sound barrier.

The UNIVAC I was also a wake-up call for IBM. Electronic computer technology proved to be a commercial success with many a high-profile business demanding to have its own UNIVAC I. IBM saw the handwriting on the wall—and the profits to be made in the commercial electronic computer market.

In 1954, fewer than 50 large corporations and government agencies employed IBM’s punch card–based computers to help manage their operations. This was the size of the commercial computer market. IBM sought to capitalize on its existing customer base by designing an electronic computer that became a natural upgrade for its customers. The computer was called the IBM 650.

IBM expected 50 sales of the IBM 650. IBM made 1,000 sales and the first wave of the commercial computer revolution had begun.

**Saving Grace**

The first-generation computers were the first to use electronic instructions that didn’t involve prewired panels. You’ll recognize these electronic instructions as programs. Prewired panel programming was on its way out and a more convenient punch card program took its place.

**Tech Talk**

Program: A set of instructions that tells a computer how to process information.
Programmers encoded their instructions for the computer onto punch cards that could be read by the computer. A common problem of the day was that there wasn’t a standard programming language that could be used across various models of computers. This meant that programmers needed to specialize in writing programs for a particular make and possibly model of computer.

This didn’t make sense to Grace Hopper, who was involved with computer systems for the Department of the Navy. She set out to streamline the task of writing a program and became the catalyst that created the first standard programming language called Common Business-Oriented Language (COBOL).

Here’s how her idea worked. Consider that each computer is like a country that has its own language. The language is used to instruct the computer on how to process information such as what numbers to add together.

Let’s say one computer understood instructions written in French and another computer understood instructions in German. Of course French and German are not computer languages, but I’ll pretend that they are so you can easily understand the problem and Hopper’s solution.

If you were to have both computers add two numbers, you would need to write instructions in French and rewrite instructions in German. Hopper and most of the computer industry at that time thought this wasted the programmer’s time.

Hopper’s solution was to create a third language that every programmer would learn. She called it COBOL. No computer understood the COBOL language, which was fine. However, Hopper and her associates created a special program called a compiler that translated the COBOL language into a language understood by each computer.

**Tech Talk**

*Compiler: A program that translates an English-like programming language into the unique language of each model computer.*

A programmer who wanted to instruct a computer to add two numbers needed only to write these instructions in COBOL, then have a compiler translate COBOL instructions into the computer’s language, the results of which could be run on the appropriate computer.

Hopper’s idea revolutionized the way computer instructions were written. COBOL later gave way to other programming languages such as FORTRAN, C, C++, and Java, all of which use the same basic principle: Translate a standardize set of instructions into a set of computer specific instructions.

Hopper has another claim to fame in computer history. She was the first person to coin the term “bug” to describe a computer malfunction. One of her computers was producing questionable results. She investigated and discovered a moth in a relay in the computer. Hopper removed the “bug” and the computer worked fine from then on.
Second Generation of Computers

A new generation of computers was ushered in as the ball dropped in Times Square to welcome 1960. The monster size computers of the first generation that used energy-hungry vacuum tubes and required enormous cooling systems gave way to trimmer, more efficient, and less expensive second-generation computers.

This was all made possible by the invention of the transistor at Bell Laboratories in New Jersey. Transistors replaced vacuum tubes as the electronic switches (see Figure 1-5) used to store and process data (see “Inside Vacuum Tubes and Transistors”).

Transistors were tiny compared to vacuum tubes and could be manufactured faster with higher quality control because, among other reasons, they had fewer parts than a vacuum tube. The slim size of a transistor enabled more electronic switches to be placed inside a second-generation computer than the number of vacuum tubes that could fit into a first-generation computer. This meant more processing power.

**Tech Talk**

Relay: A switch that is turned on and off electrically.

Transistor: A silicon-based device that performs as an electronic switch.

*Figure 1-5*

A transistor uses the properties of silicon to become an electronic switch
Computers were still pricey, ranging from a few hundred thousand to upward of a million dollars in 1960 dollars. Although the price still placed computers out of reach for most medium and small businesses, an increasing number of larger businesses found computers affordable.

The market for computers was expanding and so was competition. IBM now faced challenges from Honeywell, UNIVAC, Burroughs, and NCR, among others. Each was staking claim to a piece of the computer market.

First- and second-generation computers were mainframe computers. But that began to change in 1963, just around the time when the Beatles were taking over the American music industry. That was when a smaller version of the mainframe computer called the minicomputer was introduced.

As the name implies, a minicomputer was to some degree a scaled-down version of a mainframe computer in size, power, and price. For around $20,000 in 1963 dollars a business could harness the power of a computer to run a portion of its operation.

Digital Equipment Corporation was the first computer maker to successfully claim a major piece of the growing demand for minicomputers with the introduction of the PDP-8 minicomputer. Data General and 20 or so other manufacturers soon joined the revolution by the early 1970s.

---

**INSIDE VACUUM TUBES AND TRANSISTORS**

A computer is a box of electronic switches that are connected to form logical circuits. Information is stored in a computer by setting one or a series of switches a certain way, very similar to a two-way light switch.

You can set the light switch on or off and each of these settings can have a different meaning. For example, in my household we turn on the outside light by the front door until everyone arrives home for the evening, at which time we turn off the light. Anyone knowing this rule can determine if the entire family is home or if at least one member of the family is not home by seeing if the light is on or off.

Each setting of a switch inside a computer is associated with a binary value. A switch in the off state represents a zero and in the on state represents a one. And by grouping switches a computer can store information such as numbers, letters, data, and instructions, which you’ll learn more about in the next chapter.
A Look Back

Third and Fourth Generation of Computers

Computer technology doesn’t stand still for very long. It seems that every five years or less advances in technology quickly antiquate existing technology. Case in point occurred in the mid-1960s when second-generation computer technology based on transistors became outdated with the invention of integrated circuits. Hundreds of transistors that populated second-generation computer circuit boards, making them look like little cities, were now reduced to a circuit the size of a pin head.

Tech Talk

Integrated circuit: A computer chip that contains many microscopic transistors connected together with the wires etched into the chip.

Integrated circuits enabled computer manufacturers to build smaller, more powerful, and less expensive computers. The power of many ABC computers that once
filled a floor of an office building was now available on the head of a pin in the form of an integrated circuit.

IBM became one of the major computer manufacturers that was quick to incorporate integrated circuits into its computers, the first of which was the IBM System 360. The IBM System 360 was a family of computers that set a high standard for commercial computers.

Integrated circuits and a strategic plan enabled IBM to offer customers computer power that made it uneconomical to continue using second-generation computers. IBM’s plan centered on the expectation that advances in computer technology will continue to make existing computers out of date.

IBM introduced the concept of upward compatibility, something that wasn’t heard of in the computer industry until then. Today we call this backward compatibility. Customers simply didn’t have the time or finances to re-create their programs and data each time computer technology changed and made their existing computers obsolete.

Engineers at IBM assured customers that beginning with the IBM System 360 all new IBM computers would use the same programs and data with little or no modification. This promise plus the vast leap integrated circuits gave to computer technology encouraged many IBM customers to abandon their existing computers for the IBM System 360.

In 1971, two years after Apollo 11 landed on the moon, another evolution of computer technology was born with the onset of large-scale integrated circuits. Engineers were now able to increase the number of transistors that could be placed on an integrated circuit.

**Tech Talk**

Large-scale integrated circuits: A chip that contains many integrated circuits.

Large-scale integrated circuits technology enabled an entire computer circuitry to be placed on one computer chip. This followed the trend set by previous generations, making computers ever more powerful and smaller at very affordable prices.

**PERSONAL COMPUTERS ........................................

It’s hard for most of us to fathom a time when there weren’t personal computers. However, personal computers didn’t join the computer evolution until 1975 with the introduction of the Altair 8800 personal computer on the cover of *Popular Electronics*.

The Altair hardly resembled today’s personal computers. It didn’t have a disk drive or any of the other features that we expect to find on a personal computer. In
fact, the Altair wasn’t designed for consumers. Electronic hobbyists, people who enjoyed building radios and other electronic gizmos, were the target audience.

The Altair was a diamond in the rough, and four young men set out in the same year to cut that diamond into their own design. They were Bill Gates, Paul Allen, Steven Jobs, and Steve Wozniak. Gates and Allen formed Microsoft while Jobs and Wozniak launched Apple Computer.

Although Gates, Allen, Jobs, and Wozniak are the names most recognized as pioneers of personal computers, there were many others, such as Tandy Corporation’s TRS-80, which I spoke about at the beginning of this chapter; the Atari Commodore; and the Osborne Computer, which became the first all-in-one transportable personal computer.

Early PCs suffered from a problem similar to those of early mainframe computers: Each PC understood instructions written in its own language. Simply stated, a program written for the TRS-80 couldn’t run on an Apple computer and vice versa.

As you’ll learn in the next chapter, a computer is nothing more than a box of electronic switches. A special set of programs called an operating system such as Windows brings the computer to life by turning switches on and off based on a set of instructions by programmers who wrote the operating system.

**Tech Talk**

*Operating system: A set of programs that operate the computer hardware and interfaces between the computer user and the computer hardware.*

The TRS-80 and the Osborne Computer used a variation of an operating system called CP/M, and Apple Computer used the Applesoft operating system. Notice that I didn’t mention DOS because DOS arrived in 1981. More about this in a minute.

Personal computers needed to adopt Grace Hopper’s concept of a standardized programming language. Two professors at Dartmouth College, Dr. Thomas Kurtz and Dr. John Kemeny, developed the BASIC programming language.

BASIC enables programmers to write instructions in a standard language that could be translated into specific machine languages for each type of computer. Gates built on Kurtz’s and Kemeny’s work and developed a version of BASIC that ran on personal computers.

Although commercial programs of the day such as word processing and very simple games were still written in machine specific language, other more customizable programs were written in BASIC.

About 800,000 computers were sold by 1981 with Tandy and Apple Computer taking the lion’s share of the market. Then the market changed. IBM decided to join the fray—and made one of the biggest blunders in the history of computers.
The IBM Gambit

IBM again saw the handwriting on the wall. Personal computers were small and relatively powerful computing devices that were slowly finding their way on the desktops of corporations. Actually, very few corporations purchased personal computers because there were few business-oriented programs and even fewer employees who knew how to use a personal computer.

In the late 1970s, I worked for Volkswagen of America, the North American arm of the German car manufacturer. I also owned one of the first TRS-80s. At that time, my attic was more computerized in terms of personal computers than Volkswagen of America. I created a pricing model on the TRS-80 that was used by Volkswagen of America to determine optimum pricing. Paper, pencils, and a calculator were used for pricing until that time. Spreadsheets hadn’t been developed yet.

In 1981, IBM handed Gates and Allen the multibillion dollar lottery ticket. IBM needed an operating system for the new IBM personal computer. After failed attempts to acquire the CP/M operating system, they visited Gates, who was already known in this fledging industry because of his development of BASIC for personal computers, and asked if he had an operating system.

Gates and his associates said yes when in reality the answer was no. Gates knew someone who created an enhanced version of the CP/M operating system and called it DOS. Gates quickly purchased DOS for $50,000, then cut a deal that IBM has been kicking itself ever since.

Instead of selling DOS to IBM, Gates licensed DOS to IBM. The license gave IBM the rights to install DOS on its personal computers in exchange for a per-unit charge. Furthermore, the license was nonexclusive. This meant that Gates and his company received a piece of every personal computer sold by IBM and retained the right to license DOS to other competing computer manufacturers—which they did.

IBM’s management thought this was a good deal because they considered personal computers ancillary to their existing mainframe business. In keeping with that idea IBM dug itself deeper into a financial hole by changing their design philosophy.

Computers made by IBM, until the IBM personal computer was introduced, were considered closed systems. That is, no manufacturer except IBM made components for IBM computers. Customers could have any feature they wanted for their IBM computer as long as they purchased it from IBM.

This gave IBM a near-monopoly on the mainframe computer industry because of their market share and the closed-system design philosophy. It was commonly said among corporate computer managers that no one every got fired for buying IBM computers.

IBM moved to an open-system philosophy when it came to the IBM personal computer. IBM freely shared specifications for their IBM personal computer with
component manufacturers in hopes of minimizing their investment and risk by en-
couraging other manufacturers to enhance the features of the IBM personal computer.

Customers who purchased an IBM personal computer had the option of upgrad-
ing by buying additional hardware from IBM or from a third party. Third-party com-
puter component manufacturers were quick to rise to the occasion in anticipation that
IBM personal computers would become the industry standard.

What IBM Didn’t Anticipate

When IBM release the technical specifications for the IBM personal computer to the
industry, they practically gave away the design for their personal computer. Many
third-party component manufacturers stopped making IBM personal computer com-
patible components. They made computers that were practically the same as the IBM
personal computer. We know these today as IBM PC clones.

Manufacturers were able to modify the IBM’s personal computer design suffi-
ciently to avoid patent violations and still enable the computers to use IBM or third-
party components.

IBM soon realized that the deal with Microsoft wasn’t a smart move. Clone
manufacturers rushed to Microsoft and licensed the DOS operating system for use in
their computers. Technically IBM used Microsoft’s PC-DOS and the clone manufac-
turers used Microsoft’s MS-DOS. Although the name implies two different operating
systems, PC-DOS and MS-DOS were basically the same.

This meant that IBM PC clones looked and acted just like an IBM personal
computer and could run any program that could be run on an IBM PC. One of the
main differences between the IBM personal computer and a clone was the price.
Clones were sold at a much lower price than the IBM product, yet delivered the same
quality and computing power.

And I can’t leave out the fact that now Microsoft received compensation for ev-
ery personal computer sold except for Apple computers. This didn’t matter much be-
cause Apple’s market share never reached beyond 10 percent of the personal computer
market.

Corporations gradually became convinced that personal computers had a place
in the business community; they were no longer just a hobby. IBM’s endorsement of
personal computer technology was a major catalyst in adopting PCs for business use.

Likewise, IBM defined the standard for personal computers. Soon computers
such as the TRS-80 and the Osborne fell to the pressure of IBM. Apple Computer felt
the pressure, too, but was and remains able to hold on to a share of the market.
Software: The Key to Success

Even IBM realized that there wouldn’t be much of a market for personal computers in business unless there was software that could streamline business operations. There was a need for a killer application, an application that businesses couldn't live without. And the killer application for the personal computer was Lotus 1-2-3.

In the same year IBM launched the IBM personal computer, Mitchell Kapor created an enhanced version of the first spreadsheet program called VisiCalc. Kapor called his spreadsheet Lotus 1-2-3, which was designed to run on an IBM PC. Lotus 1-2-3 was just what IBM needed to entice businesses to purchase its product.

Personal computer programming was in its infancy. Corporations immediately found Lotus 1-2-3 a revolutionary business tool. Around the same time word processing software such as Word Star expanded the use of a personal computer and practically replaced the office typewriter.

Just as the IBM PC was gaining a foothold in the business community, I had the opportunity to help introduce personal computer programming to the technology community in my Programmer’s Notebook column that appeared in Popular Electronics, the same publication that introduced the Altair (see “Personal Computers”).

Soon the business community realized the personal computer was more than a computerized typewriter and electronic spreadsheet. With the right software, PCs could handle many of the routine, small computation and data processing jobs that were processed by mainframe computer systems.

WORKSTATION COMPUTERS ..................................

IBM, Digital Equipment Corporation, and other established computer manufacturers didn’t see personal computers as a threat to their central line of business—mainframe computers and minicomputers. Personal computers at that time were simply underpowered and lacked connectivity to each other to make an impact in the mainframe and minicomputer market. In those days, there weren’t any computer networks, as we know them today.

This would change in 1982, Sun Microsystems introduced the first workstation. Today the term workstation has many meanings from any computing device such as a personal computer that is used to perform your primary work to a particular class of computers.

However, the term workstation back in 1982 was used for a new kind of computer, a computer that was based on the reduced instruction set computer (RISC) technology developed by John Cocke in 1980. Sun Microsystems positioned the
workstation to capture the market segment held by minicomputers and low-end mainframe computers.

RISC requires fewer instructions to process the same amount of information as non-RISC computers. Personal computers did not use RISC. Workstations were capable of processing information more efficiently than minicomputers and at less cost.

The industry began to feel the pressure of this new wave of computers with the introduction of the SPARCstation™ 1 in 1989. The SPARCstation™ 1 was the size of a pizza box and was quickly adopted by major Wall Street firms such as Salomon, Inc., as its computer of choice for running mission-critical database applications. IBM and other computer manufactures soon followed with their own entries into the workstation market.

**Tech Talk**

Mission-critical database applications: Computer programs that manage data that is important to a business. The business would stop running if a mission-critical database application stopped running.

UNIX: A computer operating system that can manage multiple users and multiple applications using the computer at the same time.

Proprietary operating system: A computer OS that was specifically designed for a particular computer and not made available for use with other computers.

Various versions of UNIX were developed at universities and research labs but it never gained a wide commercial acceptance until Sun Microsystems adopted the UNIX operating system as the OS for its workstations.

Sun Microsystems created its own version of UNIX called Solaris. IBM followed suit calling its UNIX version AIX. Except for workstations and PCs, other computers such as mainframe computers used a proprietary operating system.

Sun Microsystems workstations later became a computer of choice for specialty applications such as computer animation for Hollywood. In 1995, Sun Microsystems workstations were used to create Disney’s *Toy Story*, which is the first all computer-generated feature film. This was the same year that Sun Microsystems introduced the Java programming language.

**Tech Talk**

Java: The first universal software language designed for use on the Internet and corporate intranets that enables programmers to write an application once that can be run without modification on any computer.
Until the mid-1980s personal computers were not user friendly. Anyone who used a PC needed to learn special commands since the personal computer presented only a prompt (\texttt{c: >}) on the screen.

This became imposing for many people who soon believed that special skills were required to put the personal computer through its paces. Gates, Allen, Jobs, and Wozniak realized this problem early on and began looking for a solution. The solution resided at the Xerox research center.

Xerox engineers designed a primitive graphical user interface (GUI) that used a mouse in addition to a keyboard, enabling a person to interact with the computer’s operating system. No longer was the user presented with a prompt and had to learn special commands.

**Tech Talk**

**Graphical user interface:** A method of enabling users to interact with the computer using small graphic images called icons rather than typing commands at a prompt.

In the face of objections from a number of Xerox engineers, the company invited Gates, Allen, Jobs, and Wozniak, among other technologists, to see a demonstration of their user interface. This group was more than casual observers. They took notes, asked questions, and learned how the GUI worked.

Jobs and Wozniak saw the GUI as a way to make their personal computers compete against the IBM PC. In 1984, Jobs and Wozniak introduced their new Apple computer that was designed to take advantage of their own GUI, which was based on technology observed at Xerox. The computer was called the Macintosh, and it was a hit.

IBM depended on Microsoft for DOS and DOS was not a graphical user interface. Microsoft rushed to create its own GUI and introduced Microsoft Windows in 1985. However, the early version of Windows wasn’t everything customers expected. Windows wasn’t fully integrated into the operating system.

Windows was simply another program running in DOS, a program that required more personal computer resources than other programs. Windows required a faster PC with much more memory than was available on the average IBM personal computer. Windows was ahead of its time and had to wait until hardware improvements caught up.

Apple didn’t have this problem because of several unique situations. First, Apple’s GUI ran on a new computer. The company didn’t try to retrofit the existing Apple II computers to run the GUI.
Also, Apple Computer controlled both the GUI and the design and manufacturing of the Macintosh. Microsoft had to build around the existing IBM personal computer limitations. Furthermore, the GUI was fully integrated into the Macintosh operating system, something Microsoft didn’t do. These factors gave Apple the edge.

Another problem that faced Microsoft and IBM was complications involved in writing a Windows program. Many more lines of instructions are needed to create a Windows program than is required to create a similar DOS program.

This meant that Windows programs cost more to create than DOS programs. Therefore, software manufacturers and corporations that built their own software were not rushing to write Windows programs. In addition, programmers had to be retrained to acquire the necessary skills to write Windows programs.

In 1985, two years after CDs were introduced by the recording industry, Microsoft took its Windows roadshow to Wall Street with Windows on Wall Street where Gates and his crew highlighted Windows applications built for internal use by Wall Street firms. I was on a Windows development team at Merrill Lynch, whose Windows application was featured at the show. At that time Gates had just entered the billionaire club.

While Gates’s PR people caused much excitement about Windows in the press and among those who attended the roadshow, Windows got off to a very slow start. It was another five years before Windows was widely adopted. By that time the IBM PC became more powerful and the tools to make it easier to develop Windows applications became more prevalent. Microsoft had trained an extensive number of Windows programmers, and Microsoft itself sold many of the key Windows applications, such as Word and Excel, under its own brand name.

**THE CENTRAL PROCESSOR REVOLUTION**********

The brain of a computer is the central processor, commonly known as the CPU or the microprocessor, such as the Pentium III. This is a chip inside the computer that executes instructions written by a programmer. The CPU is the major component that defines a computer, as you’ve probably realized looking at advertisements for personal computers. You’ll learn more about the functions of a CPU in the next chapter.

Companies, such as Intel, which design and manufacture processors determine the power that can be packed into a computer. More robust processors mean computer manufacturers can offer more powerful computers—and software manufacturers can create dazzling programs.

It’s hard to believe with all the hype about Intel that there were other processor-manufacturers that designed CPUs for the early personal computers. These included
Motorola Zilog. Zilog manufactured the Z80 processor, which was the most desirable processor for computers that used the CP/M operating system. The TRS-80 computer was one of the first computers to use the Z80 processor as the name implies (T is for Tandy; RS is for Radio Shack; 80 is for the Z80 processor).

However, the processor industry was in for a radical shake-up in late 1981 when IBM made its choice of processor for the IBM personal computer. The winner was Intel’s 8088 processor. IBM had made the 8088 processor the de facto industry standard.

An interesting point is that the 8088 processor was not the best processor Intel had to offer. In 1979, Intel designed the more powerful 8086 processor. IBM decided to choose second best for its personal computer because IBM wanted to save time and money.

Here’s why. An important characteristic of a processor is the amount of information it can accept for processing at the same time. This is analogous to a highway toll plaza where information is the number of cars and processing is the number of cars that can pass through the toll plaza at the same time.

In computer terms, the processor is the toll plaza, and a bit (i.e., one or zero) represents data. Wires etched into the computer’s motherboard are similar to lanes of a highway except instead of cars the wires provide a path for data to reach the processor.

The 8086 processor accepted 16 bits (16 cars) at the same time and the 8088 processor accepted 8 bits (8 cars) at a time. Therefore, the 8086 processor could receive more information per second than the 8088 processor.

The problem IBM experienced was with the data path (lanes of the highway) that had to be etched onto the IBM personal computer’s motherboard. It was faster and less costly to build a motherboard with 8 data lines (an 8-lane highway) than it was to build one with 16 data lines (a 16-lane highway).

Keep in mind that IBM was under pressure to make a presence in the fledgling personal computer market as quickly as possible. And the strategy was to use the personal computer as an ancillary product to the mainframe product line.

The Processor Race Was On

Computers could become more powerful only if improvements were made in the design of the processor. A key characteristic of power in the early days of personal computers was the amount of random access memory (RAM) installed in the computer.

**Tech Talk**

**RAM:** The place inside the computer where programs and data are stored.
Simply stated, the more RAM a computer had, the more instructions and data could be stored inside it and the less time it spent reading data from an alternative storage location such as a floppy disk.

A computer that used the 8088 processor was limited to 64 KB of memory. The storage area inside the processor that is used to hold a memory address mathematically established the limit.

Each memory location is identified by an address, which is very similar to how each house in your town is identified by an address. A memory address is a number that is represented as a series of binary numbers such as 10011011 (see “Computers Count to Two”).

Inside the processor there is a place called a register that holds the memory address of the next instruction to process. The processor looks at the memory address stored in the register, then goes out to that memory location and retrieves the instruction for processing.

The register in the 8088 processor could hold 8 bits, which meant that 64 KB was the largest number that could be held in the register. Therefore, computer manufacturers could include more than 64 KB of memory in their computers, but the processor couldn’t access that memory.

Microsoft and computer manufacturers initially worked around this problem by implementing a technique called page switching. Page switching required the operating system to exchange the contents of a block of memory called a page with a similar size block on a disk. Page switching increased access time (not a good thing) whenever a switch from disk to memory was made, however this enabled the processor to handle 128 KB of memory, which was a good thing.

The first IBM personal computer gave the business community a taste of the computing power that can be placed on the desktop. Business leaders knew that the personal computer could revolutionize the way to do business. They also knew the job they wanted the personal computer to perform, but the early PCs were underpowered to meet their needs.

Intel’s response was to increase the size of the address register from 8 bits to 16 bits, which set the upper memory address limit to 1 MB. This is a far cry from today’s 4-gigabyte (GB) memory addresses, but was revolutionary at the time.

**Intel, Meet the Competition**

Although Intel was chosen to be the processor for the first IBM personal computers, Intel management realized that IBM could easily use a processor built by a different manufacturer in future models. Intel strategy to maintain the IBM contract was to aggressively incorporate new features into its processor while maintaining back-
ward compatibility with existing processors. Simply said, programs running on existing processors would work without modification on new additions to the Intel line of processors.

Intel identified its line of processors as the x86 family with 80286 becoming the first member of the family to appear in the IBM personal computer. The x86 family began in force when the 80286 replaced the 8088 processor. From then on new processors were given similar family names such as the 80386 and the 80486 although advertising copy typically dropped the 80 and referred to them as the 286, 386, and 486.

A frequent computer trivia question is why did Intel begin with 80286 rather than the 80186? There was an 80186, but because of the rapid development in processor technology the 80186 never found its way into the IBM personal computer.

Using numbers as the name for a family of processors seemed a good idea. The name implied that the higher numbered processor was a greater value than a lower numbered one. While this is true, using numbers almost proved to be disastrous.

Numbers could not be protected under the trademark laws. This meant a competitor could reverse engineer the technology used in the Intel processor and use it to create a functionally compatible processor—and give it the same numbered name as the Intel processor.

This is similar to an automobile manufacturer calling a model a 4X4. The expression 4X4 can be used freely by any carmaker. Intel’s problem was that it had already established the processor name as a functional trademark. Now competitors could capitalize on the established processor name.

Intel discontinued using numbers as product names with the onset of what would have been the 80586. Names such as the Pentium were used rather than numbers because names could be legally protected by a trademark.

**Improvements Along the Way**

New features that increase performance are incorporated in the processor each time the name of the processor changes. The 8088 processor read information in 8-bit chunks. The 80286 doubled this capacity by reading information in 16-bit chunks.

The 80286 also introduced the concept of protected mode. Protected mode placed restrictions on memory access to programs and enabled the computer to access 16 MB of memory. You’ll learn more about protected mode in the next chapter.

In 1985, Intel introduced the 80386 processor. The 80386 enhanced memory access by using a feature called paging, which used a combination of real memory (chips inside the computer) and disk space to create 4 GB of virtual memory. The 80386 also read 32-bit chunks of information at a time, doubling that of the 80286. This dramatically increased the speed of processing many programs.
A Look Back

Tech Talk

Virtual memory: Memory that appears to exist to the computer, but doesn’t physically exist.

Four years after the introduction of the 80386, Intel came out with the next generation processor called the 80486. The 80486 included a math coprocessor and special memory called a level 1 cache built into the processor.

Computer programs were gradually becoming graphical, which involves the calculation of very large and very small numbers quickly. Earlier processors required many steps to perform these calculations and that caused a noticeable delay in the presentation of the graphic on the computer screen.

Not every customer required sophisticated graphics. Those that did purchased an additional processor that could perform these calculations using fewer steps than the main processor. The additional processor was called a math coprocessor. Any time the CPU was required to perform sophisticated calculations it turned over processing to the math coprocessor.

Intel realized that PCs were moving to a GUI, which would require every personal computer to perform these sophisticated calculations. Therefore, the math coprocessor was fully integrated into the CPU with the introduction of the 80486.

The level 1 cache also introduced with the 80486 reduced the time necessary for the processor to read information from memory by storing the most frequently used information inside the processor. I’ll explain more about this in the next chapter.

Tech Talk

Cache: Computer memory set aside for a special purpose such as storing frequently used data.

The next generation of Intel processors came in 1993 with the introduction of the Pentium. The Pentium processed 64 bits of information at a time; it also included two onboard caches of memory.

Intel followed the Pentium with the Pentium Pro and the Pentium MMX. The Pentium Pro incorporated a level 2 cache memory while the Pentium MMX contains features for multimedia. And in 1996 the Pentium II made its debut, which among other design enhancers combined the features of the Pentium Pro and the Pentium MMX into one processor.

Marketing managers at Intel soon realized that the computer market was dividing into five segments: low-price computers for home use, desktop computers for general use, mobile computers for business use, servers for computer networks, and workstations for general business use.
Each segment had its own special needs and price point. Intel’s competitors such as Motorola, Digital Equipment Corporation, and Advanced Micro Devices, among others, saw an opportunity to design comparable processors and sell them below Intel’s prices. They could do this because their chips removed features not required for specific market segments.

Intel introduced new processors to meet this competitive strategy. The Pentium II Xeon and the Pentium III Xeon provided the high processing speed that is required for servers. The Pentium III is targeted for desktop and workstations. And the Celeron processor is designed for the home market.

**SUMMARY** .................................................................

Although most of us take for granted how well computers help us in everyday chores, computer technology followed a long road to get us where we are today. Regardless of its power, a computer is a counting machine that can add two numbers and subtract two numbers. And that is where the first computer began.

The first successful computer began in 3000 B.C. It was called the abacus. The abacus was a mechanical computer that used rows of beads to represent numbers. Moving a bead in one direction facilitated addition and, in the other direction, subtraction.

It wasn’t until the 17th century that Blaise Pascal, a French mathematician, developed a mechanical adding machine. The machine was called the Pascaline and used wheels and gears to perform addition and subtraction based on the same principle used in the abacus.

Charles Babbage expanded Pascal’s concept and designed the Difference Engine and the Analytical Engine that could multiply and divide as well as perform addition and subtraction. However, the Difference Engine and the Analytical Engine was never built because it simply cost too much.

It wasn’t until almost the turn of the 20th century that Herman Hollerith created the first electrical computer that used punch cards to read data and later instructions. His company was called Tabulating Machine Company, which later changed its name to the International Business Machines Corporation, more commonly known as IBM.

In 1937, Dr. John Atanasoff and Clifford Berry, a professor and his graduate student at Iowa State University, created the first electronic computer that used vacuum tubes to store information and perform mathematical calculations. The computer was called the ABC computer. They were also the first people to apply the binary numbers system to store and manipulate information.

In 1946, Dr. John Mauchly of the University of Pennsylvania used technology developed by Atanasoff and Berry to build the first generation of computers called the
Electronic Numerical Integrator and Computer (ENIAC). The ENIAC had one purpose—to create accurate trajectory tables for the Department of Defense.

Five years later, Mauchly built the Universal Automatic Computer, better known as the UNIVAC 1, for the Remington-Rand Corporation, which became the first well-known commerce electronic computer.

The second generation of computers was ushered in when scientists at Bell Laboratories in New Jersey developed the transistor. The transistor provided the same electronic switching capabilities as a vacuum tube but at lower power, a reduced size, and lower cost.

The third generation of computers was launched in the mid-1960s with the discovery of integrated circuits. Hundreds of transistors that populated second-generation computer circuit boards, making the circuit board look like a little city were now reduced to a circuit the size of a pin head. Computer manufacturers were then able to build smaller, more powerful, and less expensive computers than second-generation computers.

The year 1963 saw the first challenge to mainframe computers. Digital Equipment Corporation introduced the first minicomputer called the PDP-8. A minicomputer was to some degree a scaled-down version of a mainframe computer in size, power, and price. For around $20,000 in 1963 dollars a business could harness the power of a computer to run a portion of its operation.

Fourth-generation computers were born in 1971 with the introduction of large-scale integrated circuits. Large-scale integrated circuits technology enabled an entire computer circuitry to be placed on one computer chip.

Four years into the fourth-generation computers, the first personal computer was introduced, called the Altair 8800. The Altair was design for the hobbyist. However, a few short years later more refined personal computers entered the market. These included the TRS-80 and, of course, the Apple computer.

It wasn’t until 1981 that personal computers were taken seriously by the business community. It was that year when IBM launched its own personal computer with the aid of Microsoft and Intel. IBM set the standard for personal computers.

The year 1984 was another turning point in the history of computers. The first commercial graphical user interface was introduced for the Macintosh line of Apple computers. Microsoft followed the next year with the introduction of Windows. Computers became easier to use because users clicked icons rather than typed commands at a prompt.

Around the same time, Sun Microsystems began developing its first workstation. Sun Microsystems incorporate the reduced instruction set computer technology developed by John Cocke in 1980. Workstations became widely adopted in 1989 with the introduction of the SPARCstation™ 1. Workstations became a more desirable alternative to minicomputers and low-end mainframe computers.
Summary Questions

1. Why are transistors important to computer technology?

2. How did IBM freeze out competitors?

3. What blunders did IBM make when signing the operating system contract with Microsoft?

4. What is the difference between a minicomputer and a workstation?

5. Why were early versions of Microsoft Windows not widely accepted?

6. Why is there an advantage of using the base-2 numbering system over the base-10 numbering system?

7. Why were mechanical computers that performed multiplication and division not widely accepted?

8. What technique did competitors use to obtain technology employed in Intel processors?

9. What is the difference between integrated circuits and large-scale integrated circuits?

10. Why did IBM select an inferior processor from Intel for the first IBM personal computer?