

How Do Computers Work?

Human beings do not spend money on expensive items unless such items somehow improve human capabilities. Cars are great. They move faster than humans, they do not get tired, and they keep you comfortable in bad weather. They are expensive, but the expense is worth it. Computers process information and do this processing better in many areas compared to human beings. The three areas in which a computer is superior to a human being are shown in figure 1.1.

Figure 1.1

3 Areas Where Computers are Superior to Human Beings
<ul style="list-style-type: none">• Computers are faster• Computers are more accurate• Computers do not forget

You may be quick to accept that *computers are faster*, but you are not so sure about the other two. Too often you have heard the term *computer error* and you also remember hearing about data that was lost in the computer.

Well, let us start our computer lesson right now by clearing up some basic myths. *Computers do not make errors*. Sure, it is possible for a computer to give erroneous information. However, the computer is nothing but a stupid machine that faithfully, and always accurately, follows instructions. If the instructions given by a human being to a computer are faulty, then the computer will produce errors. At the same time, many so-called *computer errors* are caused by sloppy data entry. A person who receives an outrageous electric bill is told that the computer created an erroneous bill. True, the computer printed the bill, but not until a data-entry clerk had slipped an extra zero in the amount of electricity used for the previous month.

Perhaps you are still not convinced. After all, what about the situation when a computer breaks down? Won't that cause problems? Broken computers will certainly cause problems. However, your computer will not work at all. Your computer applications will not work and you are stuck, but the computer does not suddenly start adding $2 + 2 = 5$.

You may also have heard that people lose their computer information because of problems with disk drives. Once again this happens, but computer users who keep their computers and diskettes in a proper environment, along with a sensible backup system, do not have such problems.

With everything that we see computers do today, it is not surprising that some people think that computers are also more intelligent than human beings. Yes, computers can do amazing things, but what must be understood before we go on is that COMPUTERS ARE STUPID. They have no intelligence. They also have no creativity. All a computer can do is to follow your directions.

Well, you give up. No point arguing with a stupid book that cannot hear you. Fine, the computer is *faster*, the computer is *more accurate*, and sure the computer *does not forget*. But how is this managed electronically? You know that electricity is incredibly fast, and you have every confidence that the flip of a switch turns on a light or a vacuum cleaner. Today's computers are electronic. Just how does electricity *store information*? How does a computer perform *computations*? How does a computer translate keyboard strokes into desirable output? These are all good questions and an attempt will be made here to explain this in a manner that does not become too technical.

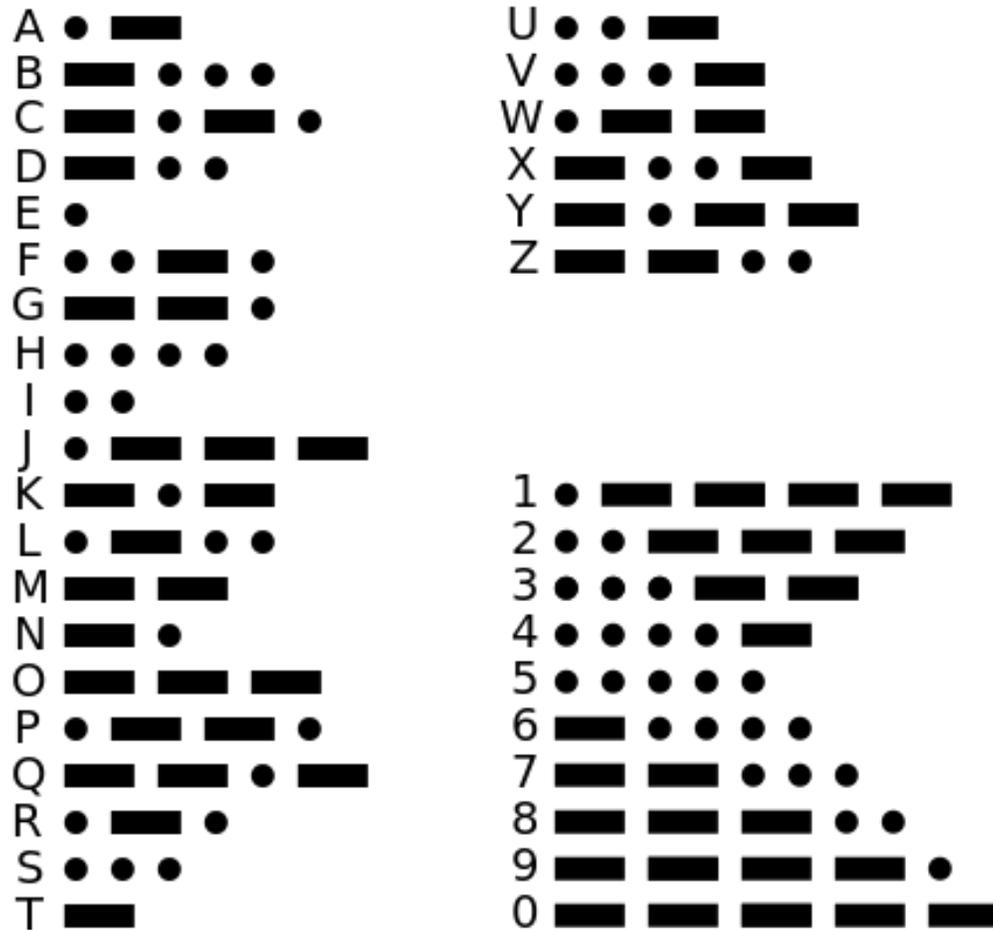
Messages with Morse Code

Unless you are a Boy Scout or Navy sailor, you probably have little experience with *Morse Code*. Today's communication is so much better than Morse code, but there was a time when Morse code was an incredible invention and allowed very rapid electronic communication.

Imagine the following situation. Somehow, you have managed to connect an electric wire between the home of your friend and yourself. You both have a buzzer and a push button. Each of you is capable of "buzzing" the other person, and the buzzer makes a noise as long as the button is pressed. You have no money for a microphone, you have no amplifier and you have no speakers. Furthermore, your mean parents have grounded you to your room without use of the telephone. But you do have your wires, your buzzers and your buttons. Can you communicate? You certainly can communicate if you know Morse code or develop a similar system.

Morse code is based on a series of *short* and *long* signals. These signals can be sounds, lights, or other symbols, but you need some system to translate signals into human communication. Morse code creates an entire set of short and long signal combinations for every letter in the alphabet and every number. Usually, a long signal is three times as long as a short signal. In the diagram shown in figure 1.2 a long signal is shown with a bar and a short signal is indicated by a circle.

Figure 1.2



You, and your buddy, can now send messages back and forth by pressing the buzzer with long and short sounds. Letters and numbers can be created this way. For instance the word **EXPO** would be signaled as follows:



The secret of Morse code is the fact that electricity can be *turned on*, and it can be *turned off*. This means that a flashlight can send long and short beams of light and a buzzer can send long and short buzzing sounds. With an established code, such as Morse code, we can now send combinations of long and short impulses electronically. Very, very brief pauses occur between the *shorts* and *longs* of a letter. Longer pauses indicate the separation between letters. This basically means that electronically we can send human messages by turning electricity on and off in a series of organized pulses. Does this mean that Samuel Morse invented the computer? No, he did not get credit for starting the computer revolution, but it does serve as a simple example to illustrate how electricity can process letters by translating **on** and **off** situations into letters and numbers.

Electronic Memory

Fine, Morse code explains how letters can be translated into electronic impulses. This explains electronic communication, but Morse code does not store any letters. Morse code signals are sent and they are gone, followed by the next signal. If you doze off, you miss the signal and it is too bad. Luckily, somebody became clever and a special device was invented that printed **dots** (short signals) and **dashes** (long signals) on a paper tape as the message was received. Now that explains a paper memory, but we still have not achieved electronic memory.

Suppose you line up a series of light bulbs. How about picking eight bulbs? Each light bulb is capable of being turned **on** and **off**. With these **8** light bulbs we can create 2^8 or 256 different combinations. Two tables are shown in figure 1.3 below. The first diagram shows **on** and **off**. The second diagram uses **1** and **0**. In Computer Science, **1** means **on** and **0** means **off**.

Figure 1.3

off	on	off	off	on	off	off	off
0	1	0	0	1	0	0	0

In this particular example, the **second** and **fifth** bulbs are **on**, and all the other bulbs are **off**. This represents only one of the **256** different combinations. Figure 1.6 will show three more combinations. It certainly is not Morse code, but by using the Morse code example, we can imagine that each of the 256 combinations is assigned to a letter, a number, or some other type of character.

Before we go on, we need to truly understand our own number system. The number system that we use is called the *decimal* number system or **base-10**. It is called “base-10” because it has 10 digits (**0 – 9**). Rumor has it that people developed a base-10 system, because of our ten fingers. Aside from 10 digits, there is something else that is significant about base-10 numbers. Every digit in a base-10 number represents a multiple of a *power of 10*. Consider the base-10 number **2,345,678** as it is shown in figure 1.4:

Figure 1.4

10^6	10^5	10^4	10^3	10^2	10^1	10^0
1,000,000	100,000	10,000	1,000	100	10	1
2	3	4	5	6	7	8

Mathematically speaking, counting and computing are possible in other bases besides base-10. The number system used by computers is the *binary* number system or **base-2**. Only the digits **0** and **1** are used. Remember that modern computers use electricity, which is either **on** or **off**. This is perfectly represented with a binary **1** or **0**. The first 32 **base-2** numbers, with their equivalent **base-10** values, are shown in figure 1.5.

Figure 1.5

Base-10	Base-2	Base-10	Base-2
0	0	16	10000
1	1	17	10001
2	10	18	10010
3	11	19	10011
4	100	20	10100
5	101	21	10101
6	110	22	10110
7	111	23	10111
8	1000	24	11000
9	1001	25	11001
10	1010	26	11010
11	1011	27	11011
12	1100	28	11100
13	1101	29	11101
14	1110	30	11110
15	1111	31	11111

Now consider these three “8-light-bulbs” combinations in figure 1.6. Each of these combinations of **on** and **off** light bulbs can be viewed as a **base-2** number. In the same way that every digit in a base-10 number represents a multiple of a *power of 10*, every column in a base-2 number represents a *power of 2*. The math is identical. The only thing that changed is the *base*.

Figure 1.6

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
0	1	0	0	0	0	0	1

01000001 (base-2) = **65** (base 10)

Figure 1.6 Continued

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
0	1	0	0	0	0	1	0

01000010 (base-2) = **66** (base 10)

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
0	1	0	0	0	0	1	1

01000011 (base-2) = **67** (base 10)

You are looking at **A**, **B**, **C** on the majority of today's personal computers. By convention, at least the convention of the *American Standard Code for Information Interchange* (ASCII), number **65** is used to store the letter **A**. Combinations **0** through **127** are used for the *standard* set of characters. The second group, from **128** through **255**, is used for the *extended* set of characters.

Now we are finally getting somewhere. We can use eight lights for each character that needs to be stored. All we have to do is place thousands of light bulbs in a container and you can store bunches of information by using this special *binary* code. There is another big bonus. Mathematically speaking, computations can be performed in any base. With our clever binary system, we now have a means to store information and make electronic calculations possible as well.

We have now learned that information can be stored in base-2 numbers. Base-2 numbers can store characters by using a system that equates numbers like the base-2 equivalent of **65** to **A**. At the same time, mathematical operations now become an electronic reality. In other words, the magic of on/off switches allows both the electronic storing of information as well as electronic computation.

It should be noted that in a first year computer science class, students are not required to be able to convert numbers between bases. You will not be expected to figure out that **201** in base-10 converts to **11001001** in base-2 or vice-versa. However, if you are planning a career in technology, especially in the area of networking, then it is definitely an essential skill.

We can also add some terminology here. A single bulb can be **on** or **off** and this single light represents a single digit in **base-2**, called a **binary digit**, which is abbreviated as *bit*. We also want to give a special name to the row of eight light bulbs (bits) that make up one character. This row shall be called a *byte*. Keep in mind that *byte* is not plural for *bit*. There is one problem with ASCII's system of storing each character in a single byte. You only have access to 256 different combinations or characters. This may be fine in the United States, but it is very

inadequate for the international community. *Unicode* is now becoming very popular and this code stores characters in 2 bytes. The result is **65,536** different possible characters. Java has adopted Unicode, as have many technical organizations. The smaller ASCII code is a *subset* of Unicode.

Bits, Bytes and Codes

Bit is a **binary digit** that is either **0** (off) or **1** (on).

1 Byte = 8 bits

1 Nibble = 4 bits ($\frac{1}{2}$ a byte)

1 Byte has 2^8 or **256** different numerical combinations.

2 Bytes has 2^{16} or **65,536** different numerical combinations.

ASCII uses one byte to store one character.

Unicode uses two bytes to store one character.

Early computers did in fact use one vacuum tube for each bit. Very large machines contained thousands of vacuum tubes with thousands of switches that could change the status of the tubes. Miles of wires connected different groups of vacuum tubes to organize the instructions that the computer had to follow. Early computer scientists had to walk around giant computers and physically connect wires to different parts of the computer to create a set of computer instructions.

The incredible advances in computer technology revolve around the size of the bit. In the forties, a bit was a single vacuum tube that burned out very rapidly. Soon large vacuum tubes were replaced by smaller, more reliable, vacuum tubes. A pattern was set that would continue for decades. Small is not only smaller, it is also better. The small tube gave place to the pea-sized transistor, which was replaced by the integrated circuit. Bits kept getting smaller and smaller. Today, a mind-boggling quantity of bits fits on a single microchip.

This is by no means a complete story of the workings of a computer. Very, very thick books exist that detail the precise job of every component of a computer. Computer hardware is a very complex topic that is constantly changing. Pick up a computer magazine, and you will be amazed by the new gadgets and the new computer terms that keep popping up. The intention of this brief introduction is to help you understand the essence of how a computer works. Everything revolves around the ability to process enormous quantities of binary code, which is capable of holding two different states: **1** and **0**.

Memory and Secondary Storage

Electronic appliances used to have complex – cables everywhere – dusty interiors. Repairing such appliances could be very time consuming. Appliances, computers included, still get dusty on the inside, but all the complex wires and vacuum tubes are gone. You will now see series of boards that all have hundreds and thousands of copper lines crisscrossing everywhere. If one of these boards is bad, it is pulled out and replaced with an entire new board. What used to be loose, all over the place, vacuum tubes, transistors, resistors, capacitors and wires, is now neatly organized on one board. Electronic repair has become much faster and cheaper in the process.

In computers the main board with all the primary computer components is called the *motherboard*. Attached to the motherboard are important components that store and control information. These components are made out of chips of silicon. Silicon is a semiconductor, which allows precise control of the flow of electrons. Hence we have the names memory chip, processing chip, etc. We are primarily concerned with the *RAM* chip, the *ROM* chip and the *CPU* chip.

I mentioned earlier that information is stored in a binary code as a sequence of ones and zeroes. The manner in which this information is stored is not always the same. Suppose now that you create a group of chips and control the bits on these chips in such a way that you cannot change their values. Every bit on the chip is fixed. Such a chip can have a permanent set of instructions encoded on it. These kinds of chips are found in cars, microwaves, cell phones and many electronic appliances that perform a similar task day after day.

Computers also have chips that store permanent information. Such chips are called **Read Only Memory** chips or *ROM* chips. There is a bunch of information in the computer that should not disappear when the power is turned off, and this information should also not be altered if the computer programmer makes some mistake. A ROM chip can be compared to a music CD. You can listen to the music on the CD, but you cannot alter or erase any of the recordings.

Another type of chip stores information temporarily. Once again, information is stored in many bytes, each made up of eight bits, but this information requires a continuous electric current. When the power is gone, so is the information in these chips. Computer users also can alter the information of these chips when they use the computer. Such chips can store the data produced by using the computer, such as a research paper or it can store the current application being used by the computer. The name of this chip is **Random Access Memory** chip or *RAM* chip. Personally, I am not happy with that name. I would have preferred something that implies that the chip is **Read and Write**, but then nobody asked for my opinion when memory chips were named.

Computer terminology has actually borrowed terms from the *Metric System*. We all remember that a *kilometer* is 1000 meters and a *kilogram* is 1000 grams. This

is because the Metric System prefix *kilo* means *1000*. In the same way, a *kilobyte* is about 1000 bytes. Why did I say “about”? Remember that everything in the computer is based on powers of 2. If you are going to be really technical and picky, a *kilobyte* is exactly 2^{10} or 1024 bytes. For our purposes, 1000 bytes is close enough. Other metric system prefixes are shown in figure 10.7.

Figure 1.7

Measuring Memory			
KB	Kilo Byte	1 thousand bytes	1,000
MB	Mega Byte	1 million bytes	1,000,000
GB	Giga Byte	1 billion bytes	1,000,000,000
TB	Tera Byte	1 trillion bytes	1,000,000,000,000
PB	Peta Byte	1 thousand <i>terabytes</i>	1,000,000,000,000,000
EB	Exa Byte	1 million <i>terabytes</i>	1,000,000,000,000,000,000
ZB	Zetta Byte	1 billion <i>terabytes</i>	1,000,000,000,000,000,000,000
YB	Yotta Byte	1 trillion <i>terabytes</i>	1,000,000,000,000,000,000,000,000

Modern computers now have memory is measured in *gigabytes* and hard drive space is measured in *terabytes*. Kilobytes and megabytes are rapidly fading from the computer terminology. Your children will probably be working with *petabytes* and *exabytes*. Your grandchildren will probably be working with *zetabytes* and *yottabytes*.

The most significant chunk of silicon in your computer is the *CPU* chip. *CPU* stands for **C**entral **P**rocessing **U**nit and this chip is the *brains* of the computer. You cannot call this chip *ROM* or *RAM*. On this tiny little chip are lots of permanent instructions that behave like ROM, and there are also many places where information is stored temporarily in the manner of a RAM chip. The CPU is one busy little chip. You name it, the CPU does the job.

A long list of operations could follow here but the key notion is that you understand that all the processing, calculating and information passing is controlled by the Central Processing Unit. The power of your computer, the capabilities of your computer, and the speed of your computer is based on your CPU chip more than any other computer component.

Secondary Storage

I just know that you are an alert student. ROM made good sense. RAM also made sense, but you are concerned. If the information in RAM is toast when you turn off the computer . . . then what happens to all the stored information, like your research paper? Oh, I underestimated your computer knowledge. You do know that we have hard drives, diskettes, zip diskettes, tapes, and CDs that can store information permanently.

We have stored information with *rust* for quite some time. Did I say *rust*? Yes, I did. Perhaps you feel more comfortable with the term *iron oxide*. Tiny particles of iron oxide on the surface of a tape or floppy disk are magnetically charged positively or negatively. Saving information for later use may be a completely different process from simply storing it in memory, but the logic is still similar.

Please do keep in mind that this information will not disappear when the power is turned off, but it can be easily altered. New information can be stored over the previous information. A magnetic field of some type, like a library security gate, heat in a car, dust in a closet, and peanut butter in a lunch bag can do serious damage to your information.

You might be confused about the currently popular *CD-ROMs*. You can see that they are external to the computer, but *ROM* implies Read Only Memory. CDs store enormous amount of information. The information is permanent and thus behaves like ROM. When you use a CD with a computer it behaves as if you had added extra ROM to your computer internally. CDs do not use rust; they are far too sophisticated for such a crude process. The CD is coded with areas that reflect and absorb laser light. Once again we can create a code system because we have two different states, on and off.

The on/off state is the driving force of the digital computer. What is *digital*? Look at your watch. You can see digits, and you see the precise time. There is no fractional time. A clock with hour, minute and second hands is an *analog* device. It measures in a continuous fashion. A measuring tape is also *analog*, as is a speedometer with a rotating needle. What is the beauty of digitizing something? With digital information it is possible to always make a precise copy of the original.

It is easy to transfer, store and use digitized information. Entire pictures can be converted to a digitized file and used elsewhere. I am sure you have been in movie theaters where “digital” sound is advertised. So digital is the name of the game. Just remember that not all digitizing is equally fast. The internal memory of the computer is digital and it uses electronics. The access of a hard disk involves electronics, but the information is read off a disk that rotates and only one small part of the disk is “readable” at one time. Accessing a disk drive is much slower than accessing internal memory.

Hardware and Software

Computer science, like all technical fields, has a huge library of technical terms and acronyms. Volumes can be filled with all kinds of technical vocabulary. Have no fear; you will not be exposed to volumes, but you do need some exposure to the more common terms you will encounter in the computer world. Some of these terms will be used in the following section on the history of computers.

For starters, it is important that you understand the difference between *hardware* and *software*. Computer *hardware* refers to any physical piece of computer equipment that can be seen or touched. Essentially, hardware is *tangible*. Computer *software*, on the other hand, is *intangible*. Software refers to the set of computer instructions which make the computer perform a specific task. These computer instructions, or *programs*, are usually encoded on some storage device like a CD, jump drive or hard drive. While CDs, jump drives and hard drives are examples of tangible hardware, the programs stored on them are examples of intangible software.

Computer Hardware and Peripheral Devices

There are big, visible hardware items that most students know because such items are difficult to miss. This type of hardware includes the main computer box, the monitor, printer, and scanner. There are additional hardware items that are not quite as easy to detect.

It helps to start at the most essential computer components. There is the CPU (Central Processing Unit), which controls the computer operations. The CPU together with the primary memory storage represents the actual computer. Frequently, when people say to move the CPU to some desk, they mean the big box that contains the CPU and computer memory. This “box” is actually a piece of hardware called the *system unit* and it actually contains a lot more than just a bunch of memory chips. There are also many *peripheral devices*.

What does periphery mean? It means an imprecise boundary. If the computers are located on the periphery of the classroom, then the computers are located against the walls of the classroom. Computer hardware falls into two categories. There are *internal* peripheral devices and *external* peripheral devices.

External peripheral devices are located outside the computer and connected with some interface, which is usually a cable, but it can also be wireless. The first external peripheral device you see is the monitor. In the old days a monitor was called a CRT (**C**athode **R**ay **T**ube). This was appropriate with the bulky monitors that looked like old televisions. Today many monitors use LCD (**L**iquid **C**rystal

Display) or Plasma screens. It is common now for monitors to be 17, 24, or even 28 inches. (Right now, I am actually looking at a 48 inch plasma screen as I edit the 2014 version of this chapter.) Things have changed considerably since the days of 10 inch monochrome computer monitors.

Other external peripheral devices include a printer, keyboard, mouse, scanner, and jump drive. There are many *internal* peripheral devices that are connected to the computer inside the system unit. These devices include the disk drive, CD ROM drive, hard drive, network interface card and video card.

Computer Software

Computer software provides instructions to a computer. The most important aspect of this course is to learn how to give correct and logical instructions to a computer with the help of a programming language. Software falls into two categories. There is *system software* and *application software*. Usually, students entering high school are already familiar with applications software.

Applications software refers to the instructions that the computer requires to do something specific for you. The whole reason why a computer exists is so that it can assist people in some type of *application*. If you need to write a paper, you load a *word processor*. If you need to find the totals and averages of several rows and columns of numbers, you load an *electronic spreadsheet*. If you want to draw a picture, you load a *paint program*. Word processors and electronic spreadsheets are the two most common *applications* for a computer. Currently, there are thousands of other applications available which assist people in every possible area from completing tax returns to designing a dream home to playing video games.

NOTE: People often talk about the “apps” on their cell phone. *App* is just an abbreviation for *application software*.

System software refers to the instructions that the computer requires to operate properly. A common term is **Operating System** (OS). The major operating systems are Windows, UNIX, Linux and the MAC OS. It is important that you understand the *operation* of your operating system. With an OS you can store, move and organize data. You can install new external devices like printers and scanners. You can personalize your computer with a desktop appearance and color selections. You can execute applications. You can install additional applications. You can also install computer protection against losing data and viruses.