Overview

Computers are complex machines. But unless they are given instructions, all that great hardware is wasted. In order to be useful, computers must be programmed.

Compared to the computers of today, the first computers were simple devices. They were programmed by flipping switches or by inserting cards with holes punched in them. Early computers were only used to do simple math or tabulation. Today's computers, however, are used to display pictures, play sounds, and perform very complex tasks. To make programming modern computers possible, more sophisticated languages had to be developed.

In this chapter, you will see how the circuits in a computer are used to represent data and give instructions. Also, you will learn about programming languages that free programmers to concentrate on the function of the program rather than the computer's circuits.

Finally, you will learn about the five-step process that is essential for every programmer to follow in order to produce good computer programs.

CHAPTER 2, SECTION 1

The Computer's Language

As you learned in Chapter 1, computers are complex devices where electrical signals pulse through the system millions of times per second. These electrical signals are what the computer uses to represent data and give instructions to system components.

REPRESENTING DATA

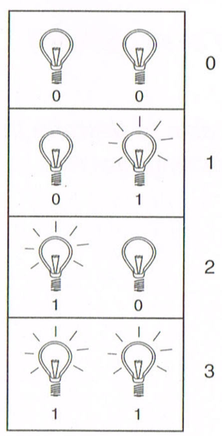

FIGURE 2-1
There are four light combinations possible with two light bulbs.

Data is a computer representation of something that exists in the real world. For example, data can be values such as money, measurements, quantities, or a high score. Data can also be alphabetic, such as names and addresses or a business letter.

In a computer, all data is represented by numbers, and the numbers are represented electronically in the computer. To understand how electrical signals become numbers, let's begin by looking at a simple electric circuit that everyone is familiar with: a switch controlling a light bulb.

FROM CIRCUITS TO NUMBERS

When you think of an electric circuit, you probably think of it being either on or off; for example, a light bulb is turned on and off by a switch. The light bulb can exist in two conditions: on or off. In technical terms, the light bulb has two ***states***.

Imagine you had two light bulbs on two switches. With two light bulbs there are four possible states, as shown in Figure 2-1. You could assign a number to each of the states and represent the numbers 0 through 3.

You can't do much using only the numbers 0 through 3, but if more circuits are added, the number of states increases. For example, Figure 2-2 shows how three circuits can represent the numbers 0 through 7 because there are eight possible states.

If you are the mathematical type, you may have noticed that the number of states is determined by the formula 2^n where n is the number of circuits (see Table 2-1).

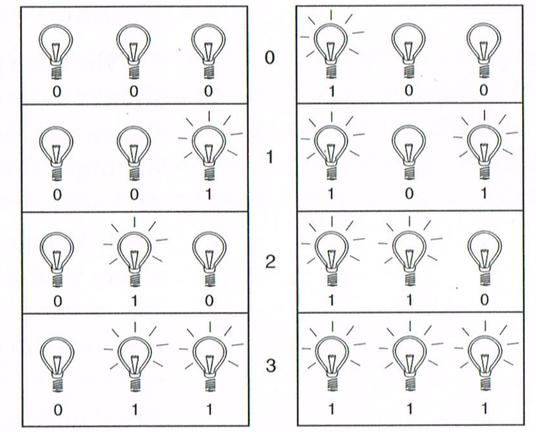

FIGURE 2-2
There are eight light combinations possible with three light bulbs.

TABLE 2-1

NUMBER OF CIRCUITS	NUMBER OF STATES	NUMBERS THAT CAN BE REPRESENTED
1	2^1=2	0,1
2	2^2=4	0..3
3	2^3=8	0..7
4	2^4=16	0..15
5	2^5=32	0..31
6	2^6=64	0..63
7	2^7=128	0..127
8	2^8=256	0..255

Now instead of lights, think about circuits in the computer. A single circuit in a computer is like a single light; it can be on or off. A special number system, called the ***binary number system***, is used to represent numbers with groups of these circuits. In the binary number system each binary digit, called a ***bit*** for short, is either a 0 or a 1. As shown in Figure 2-3, circuits that are off are defined as 0, and circuits that are on are defined as 1. Binary digits (bits) are combined into groups of eight bits called ***bytes***.

If a byte is made up of eight bits, then there are 256 possible combinations of those eight bits representing the numbers 0 through 255 (see Table 2-1). Even though 255 is not a small number, it is definitely not the largest number you will ever use. So to represent larger numbers, computers group bytes together. You will learn more about how the computer uses bytes to represent numbers in Chapter 4.

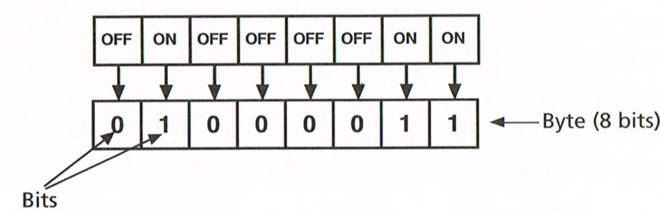

FIGURE 2-3
In the computer, signals that are off are defined as 0 and signals that are on are defined as 1.