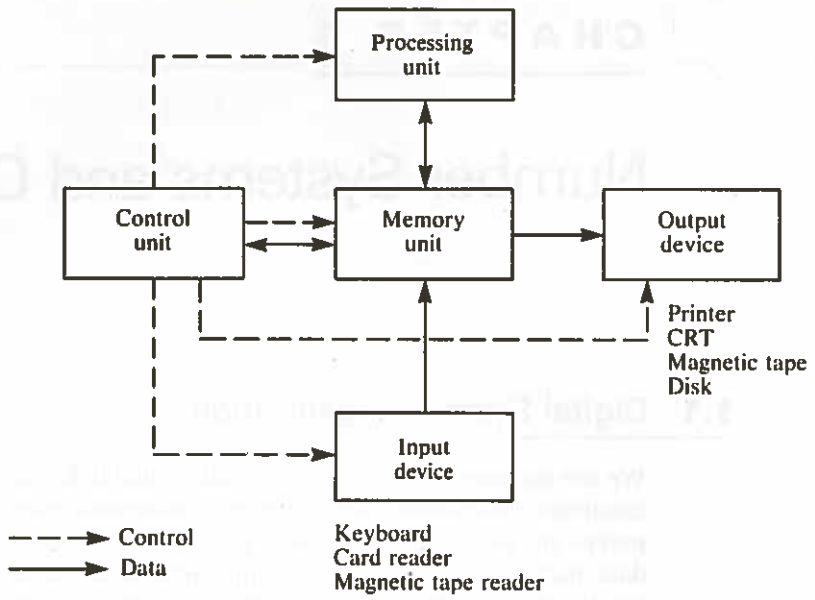

Figure 1.1 Components of a digital computer

Figure 1.1 shows the components of a digital computer, the most general digital device. The program to manipulate the data is first brought into the *memory unit* through the *input device*. The data to be processed are then brought into the memory unit, also through the input device. The *control unit* fetches instructions from the program stored in the memory one at a time, analyzes each instruction, and instructs the *processing unit* to perform the operations called for by the instruction. The results produced by the processing unit are forwarded to the memory unit for storage and then transferred to the *output device*.

As mentioned earlier, the elements in the discrete data representation correspond to discrete voltage levels or current magnitudes in the digital system hardware. If the digital system is required to manipulate only numeric data, for instance, it will be best to use 10 voltage levels, with each level corresponding to a decimal digit. But the noise introduced by multiple levels makes such representation impractical. Therefore, digital systems typically use a two-level representation, with one voltage level representing a 0 and the other representing a 1. To represent all 10 decimal digits using this *binary* (two-valued) alphabet of 0 and 1, a unique pattern of 0s and 1s is assigned to each digit. For example, in an electronic calculator, each keystroke should produce a pattern of 0s and 1s corresponding to the digit or the operation represented by that key.

Because the data elements and operations are all represented in binary form in all practical digital systems, a good understanding of the binary number system and data representation is basic to the analysis and design of digital system hardware. In this chapter, we will discuss the binary number system in detail. In addition we will discuss two other widely used systems: *octal* and *hexadecimal*. These two number systems are useful in representing binary information in a compact form. When the human user of the digital system works with data manipulated by the system, either to verify it or to communicate it to another user, the compactness provided by these systems is helpful. As we will see in this chapter, data conversion from one number system to the other can be performed in a straightforward manner.

The data to be processed by the digital system are made up of decimal digits, alphabetic characters, and special characters, such as +, −, *. The digital system uses a unique pattern of 0s and 1s to represent each of these digits and characters in the binary form. The collection of these binary patterns is called the *binary code*. Various binary codes have been devised by digital system designers over the years. Some popular codes will be discussed in this chapter.

1.2 Number Systems

Let us review the decimal number system, the system with which we are most familiar. There are 10 symbols (0 through 9), called digits, in the system—along with a set of relations defining the operations of addition (+), subtraction (−), multiplication (×), and division (/). The total number of digits in a number system is called the *radix* or *base* of the system. The digits in the system range in value from 0 through $r - 1$, where r is the radix. For the decimal system, $r = 10$ and the digits range in value from 0 through $(10 - 1) = 9$.

In the so-called *positional notation* of a number, the radix point separates the "integer" portion of the number from the "fraction" portion. If there is no fraction portion, the radix point is not explicitly shown in the positional notation. Furthermore, each position in the representation has a weight associated with it. The weight of each position is equivalent to the radix raised to a power. The power starts with a 0 at the position immediately to the left of the radix point and increases by 1 as we move each position toward the left, and decreases by 1 as we move each position toward the right. A typical number in the decimal system is shown in the following example.