CHAPTER ONE

Digital Logic Circuits

IN THIS CHAPTER

1-1 Digital Computers

The digital computer is a digital system that performs various computational tasks. The word *digital* implies that the information in the computer is represented by variables that take a limited number of discrete values. These values are processed internally by components that can maintain a limited number of discrete states. The decimal digits 0, 1, 2, . . . , 9, for example, provide 10 discrete values. The first electronic digital computers, developed in the late 1940s, were used primarily for numerical computations. In this case the discrete elements are the digits. From this application the term *digital computer* has emerged. In practice, digital computers function more reliably if only two states are used. Because of the physical restriction of components, and because human logic tends to be binary (i.e., true-or-false, yes-or-no statements), digital components that are constrained to take discrete values are further constrained to take only two values and are said to be *binary*.

Digital computers use the binary number system, which has two digits: 0 and 1. A binary digit is called a *bit*. Information is represented in digital computers in groups of bits. By using various coding techniques, groups of bits can be made to represent not only binary numbers but also other discrete symbols, such as decimal digits or letters of the alphabet. By judicious use of binary arrangements and by using various coding techniques, the groups of bits are used to develop complete sets of instructions for performing various types of computations.

In contrast to the common decimal numbers that employ the base 10 system, binary numbers use a base 2 system with two digits: 0 and 1. The decimal equivalent of a binary number can be found by expanding it into a power series with a base of 2. For example, the binary number 1001011 represents a quantity that can be converted to a decimal number by multiplying each bit by the base 2 raised to an integer power as follows:

$$1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 75$$

The seven bits 1001011 represent a binary number whose decimal equivalent is 75. However, this same group of seven bits represents the letter K when used in conjunction with a binary code for the letters of the alphabet. It may also represent a control code for specifying some decision logic in a particular digital computer. In other words, groups of bits in a digital computer are used to represent many different things. This is similar to the concept that the same letters of an alphabet are used to construct different languages, such as English and French.

program

A computer system is sometimes subdivided into two functional entities: hardware and software. The hardware of the computer consists of all the electronic components and electromechanical devices that comprise the physical entity of the device. Computer software consists of the instructions and data that the computer manipulates to perform various data-processing tasks. A sequence of instructions for the computer is called a *program*. The data that are manipulated by the program constitute the *data base*.

A computer system is composed of its hardware and the system software available for its use. The system software of a computer consists of a collection of programs whose purpose is to make more effective use of the computer. The programs included in a systems software package are referred to as the *operating system*. They are distinguished from application programs written by the user for the purpose of solving particular problems. For example, a high-level language program written by a user to solve particular data-processing needs is an application program, but the compiler that translates the high-level language program to machine language is a system program. The customer who buys a computer system would need, in addition to the hardware, any available software needed for effective operation of the computer. The system software is an indispensable part of a total computer system. Its function is to compensate for the differences that exist between user needs and the capability of the hardware.

computer hardware

The hardware of the computer is usually divided into three major parts, as shown in Fig. 1-1. The central processing unit (CPU) contains an arithmetic and logic unit for manipulating data, a number of registers for storing data, and control circuits for fetching and executing instructions. The memory of a computer contains storage for instructions and data. It is called a random-access memory (RAM) because the CPU can access any location in memory at random and retrieve the binary information within a fixed interval of time. The input and

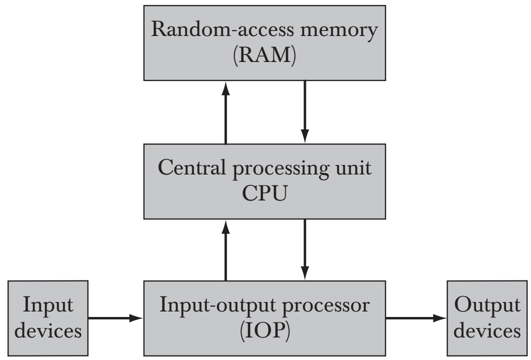


Figure 1-1 Block diagram of a digital computer.

output processor (IOP) contains electronic circuits for communicating and controlling the transfer of information between the computer and the outside world. The input and output devices connected to the computer include keyboards, printers, terminals, magnetic disk drives, and other communication devices.

This book provides the basic knowledge necessary to understand the hardware operations of a computer system. The subject is sometimes considered from three different points of view, depending on the interest of the investigator. When dealing with computer hardware it is customary to distinguish between what is referred to as computer organization, computer design, and computer architecture.

computer organization

Computer organization is concerned with the way the hardware components operate and the way they are connected together to form the computer system. The various components are assumed to be in place and the task is to investigate the organizational structure to verify that the computer parts operate as intended.

computer design

Computer design is concerned with the hardware design of the computer. Once the computer specifications are formulated, it is the task of the designer to develop hardware for the system. Computer design is concerned with the determination of what hardware should be used and how the parts should be connected. This aspect of computer hardware is sometimes referred to as *computer implementation*.

computer architecture

Computer architecture is concerned with the structure and behavior of the computer as seen by the user. It includes the information, formats, the instruction set, and techniques for addressing memory. The architectural design of a computer system is concerned with the specifications of the various functional modules, such as processors and memories, and structuring them together into a computer system.

Two basic types of computer architectures are von Neumann architecture and Harvard architecture. von Neumann architecture describes a general framework, or structure, that a computer's hardware, programming, and data should follow. Although other structures for computing have been devised and implemented, the vast majority of computers in use today operate according to the von

Neumann architecture. Von Neumann envisioned the structure of a computer system as being composed of the following components:

1. the central arithmetic unit, which today is called the arithmetic-logic unit (ALU). This unit performs the computer's computational and logical functions;
2. memory; more specifically, the computer's main, or fast, memory, such as random access memory (RAM);
3. a control unit that directs other components of the computer to perform certain actions, such as directing the fetching of data or instructions from memory to be processed by the ALU; and
4. man-machine interfaces; i.e., input and output devices, such as a keyboard for input and display monitor for output, as shown in Fig. 1.1.

Of course, computer technology has developed extensively since von Neumann's time. For instance, due to integrated circuitry and miniaturization, the ALU and control unit have been integrated onto the same microprocessor "chip", becoming an integrated part of the computer's central processing unit (CPU). The most noteworthy concept contained in von Neumann's first report was most likely that of the stored-program principle. This principle holds that data, as well as the instructions used to manipulate that data, should be stored together in the same memory area of the computer and instructions are carried out sequentially, one instruction at a time. The sequential execution of programming imposes a sort of 'speed limit' on program execution, since only one instruction at a time can be handled by the computer's processor. It means that the CPU can be either reading an instruction or reading/writing data from/to the memory. Both cannot occur at the same time since the instructions and data use the same signal pathways and memory.

The Harvard architecture uses physically separate storage and signal pathways for their instructions and data. The term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape (24-bits wide) and data in relay latches (23-digits wide). In a computer with Harvard architecture, the CPU can read both an instruction and data from memory at the same time, leading to double the memory bandwidth.

An example of computer architecture based on the von Neumann architecture is the desktop personal computer. Microcontroller (single-chip microcomputer)-based computer systems and DSP (Digital Signal Processor)-based computer systems are examples for Harvard architecture.

The book deals with all three subjects associated with computer hardware. In Chapters 1 through 4 we present the various digital components used in the organization and design of computer systems. Chapters 5 through 7 cover the steps that a designer must go through to design and program an elementary digital computer. Chapters 8 and 9 deal with the architecture of the central processing unit. In Chapters 11 and 12 we present the organization and architecture of the input–output processor and the memory unit.

1-2 Logic Gates

Binary information is represented in digital computers by physical quantities called *signals*. Electrical signals such as voltages exist throughout the computer in either one of two recognizable states. The two states represent a binary variable that can be equal to 1 or 0. For example, a particular digital computer may employ a signal of 3 volts to represent binary 1 and 0.5 volt to represent binary 0. The input terminals of digital circuits accept binary signals of 3 and 0.5 volts and the circuits respond at the output terminals with signals of 3 and 0.5 volts to represent binary input and output corresponding to 1 and 0, respectively.

gates

Binary logic deals with binary variables and with operations that assume a logical meaning. It is used to describe, in algebraic or tabular form, the manipulation and processing of binary information. The manipulation of binary information is done by logic circuits called *gates*. Gates are blocks of hardware that produce signals of binary 1 or 0 when input logic requirements are satisfied. A variety of logic gates are commonly used in digital computer systems. Each gate has a distinct graphic symbol and its operation can be described by means of an algebraic expression. The input–output relationship of the binary variables for each gate can be represented in tabular form by a *truth table*. The basic logic gates are AND and inclusive OR with multiple inputs and NOT with a single input. Each gate with more than one input is sensitive to either logic 0 or logic 1 input at any one of its inputs, generating the output according to its function. For example, a multi-input AND gate is sensitive to logic 0 on any one of its inputs, irrespective of any values at other inputs.

The names, graphic symbols, algebraic functions, and truth tables of eight logic gates are listed in Fig. 1-2, with applicable sensitivity input values. Each gate has one or two binary input variables designated by A and B and one binary output variable designated by x. The AND gate produces the AND logic function: that is, the output is 1 if input A and input B are both equal to 1; otherwise, the output is 0. These conditions are also specified in the truth table for the AND gate. The table shows that output x is 1 only when both input A and input B are 1. The algebraic operation symbol of the AND function is the same as the multiplication symbol of ordinary arithmetic. We can either use a dot between the variables or concatenate the variables without an operation symbol between them. AND gates may have more than two inputs, and by definition, the output is 1 if and only if all inputs are 1.

OR

The OR gate produces the inclusive-OR function; that is, the output is 1 if input A or input B or both inputs are 1; otherwise, the output is 0. The algebraic symbol of the OR function is +, similar to arithmetic addition. OR gates may have more than two inputs, and by definition, the output is 1 if any input is 1.

inverter

The inverter circuit inverts the logic sense of a binary signal. It produces the NOT, or complement, function. The algebraic symbol used for the logic complement is either a prime or a bar over the variable symbol. In this book we use a prime for the logic complement of a binary variable, while a bar over the letter is reserved for designating a complement microoperation as defined in Chap. 4.

The small circle in the output of the graphic symbol of an inverter designates a logic complement. A triangle symbol by itself designates a buffer circuit. A

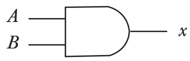
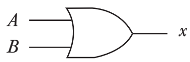
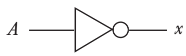
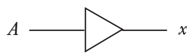
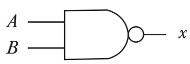
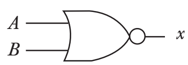
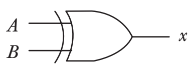
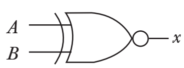
Name	*Graphic symbol*	*Algebraic function*	*Truth table*	*Input sensitivity*
AND		$x = A \cdot B$ or $x = AB$	$A\ B\ \vert\ x$ 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1	0
OR		$x = A + B$	$A\ B\ \vert\ x$ 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1	1
Inverter		$x = A'$	$A\ \vert\ x$ 0 | 1 1 | 0	Not Applicable
Buffer		$x = A$	$A\ \vert\ x$ 0 | 0 1 | 1	Not Applicable
NAND		$x = (AB)'$	$A\ B\ \vert\ x$ 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0	0
NOR		$x = (A + B)'$	$A\ B\ \vert\ x$ 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0	1
Exclusive-OR (XOR)		$x = A \oplus B$ or $x = A'B + AB'$	$A\ B\ \vert\ x$ 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0	Not Applicable
Exclusive-NOR or equivalence		$x = (A \oplus B)'$ or $x = \cancel{A'B + AB'}$	$A\ B\ \vert\ x$ A' B' 0 0 | 1 1 1 0 1 | 0 1 0 | 0 1 1 | 1 ✓	OR Not Applicable

Figure 1-2 Digital logic gates with applicable input sensitivity values.