BASIC COMPUTER OPERATIONS

A computer as shown in Fig. 2.1 performs basically five major operations or functions irrespective of their size and make. These are 1) it accepts data or instructions by way of input, 2) it stores data, 3) it can process data as required by the user, 4) it gives results in the form of output, and 5) it controls all operations inside a computer. We discuss below each of these operations.

1. **Input:** This is the process of entering data and programs in to the computer system. You should know that computer is an electronic machine like any other machine which takes as inputs raw data and performs some processing giving out processed data. Therefore, the input unit takes data from us to the computer in an organized manner for processing.

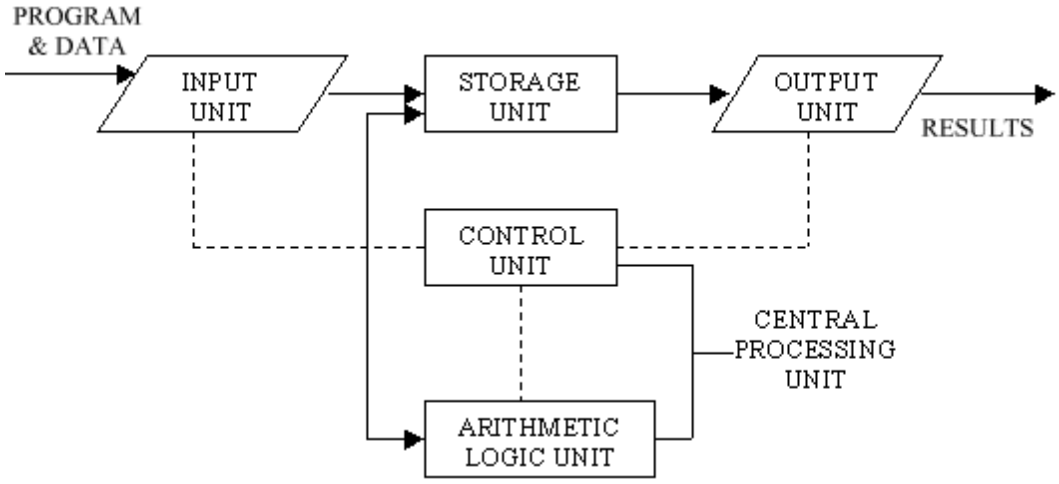


Fig. 2 Basic computer Operations

2. Storage: The process of saving data and instructions permanently is known as storage. Data has to be fed into the system before the actual processing starts. It is because the processing speed of Central Processing Unit (CPU) is so fast that the data has to be provided to CPU with the same speed. Therefore the data is first stored in the storage unit for faster access and processing. This storage unit or the primary storage of the computer system is designed to do the above functionality. It provides space for storing data and instructions.

The storage unit performs the following major functions:

- All data and instructions are stored here before and after processing.
- Intermediate results of processing are also stored here.

3. Processing: The task of performing operations like arithmetic and logical operations is called processing. The Central Processing Unit (CPU) takes data and instructions from the storage unit and makes all sorts of calculations based on the instructions given and the type of data provided. It is then sent back to the storage unit.

4. Output: This is the process of producing results from the data for getting useful information. Similarly the output produced by the computer after processing must also be kept somewhere inside the computer before being given to you in human readable form. Again the output is also stored inside the computer for further processing.

5. Control: The manner how instructions are executed and the above operations are performed. Controlling of all operations like input, processing and output are performed by control unit. It takes care of step by step processing of all operations in side the computer.

Data representation.

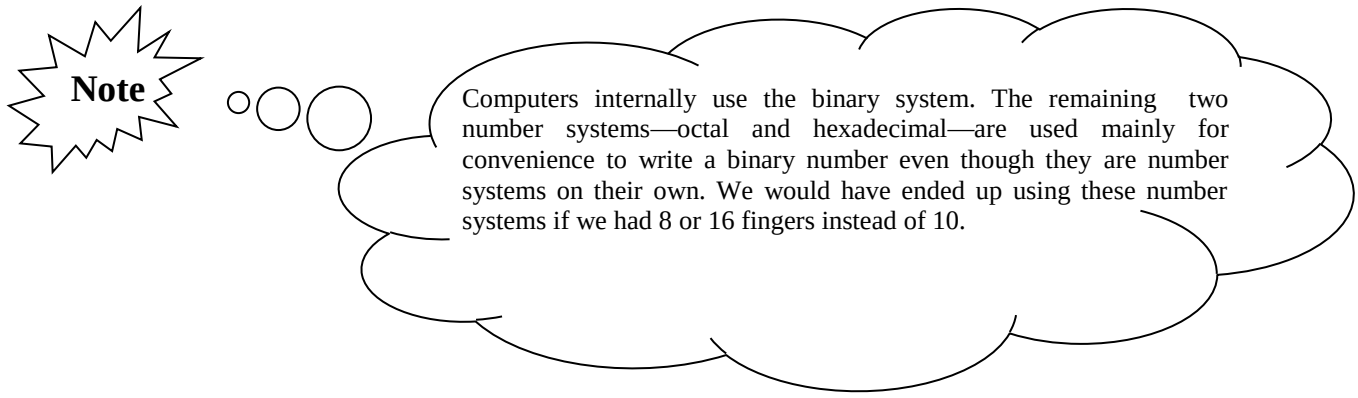
The basic building block of personal computers is the transistor. A ***transistor*** is an electronic device for controlling the flow of electrons in an electrical circuit. If electrons are allowed to flow, the circuit is on; conversely, if electrons are not allowed to flow, the circuit is off. Thinking of a transistorized circuit as a switch like a light switch at a home. The switch is either **on** or **off** and stays that way until it is flipped again. When a circuit is **on**, we say it is in the marking state and assign a **1** to it. Conversely, when it is **off,** we assign a **0** to it.

A modern digital computer is often said to be a binary computer because its most basic circuits can remember either one of ***two states: 0 and 1***. The binary digits 0 and 1 are called **bits**. Both the internal and external memory of computer are nothing more than **store-houses** for bits.

Numbers systems.

The number systems that we discuss here are based on positional number systems. The decimal number system that we are already familiar with is an example of a positional number system. In contrast, the Roman numeral system is not a positional number system. Every positional number system has a ***radix*** or ***base***, and an ***alphabet***. The base is a positive number. For example, the decimal system is a base-10 system. The

number of symbols in the alphabet is equal to the base of the number system. The alphabet of the decimal system is 0 through 9, a total of 10 symbols or digits. There are four number systems that are relevant in the context of computer systems and programming. These are the *decimal* (base-10), *binary* (base-2), *octal* (base-8), and *hexadecimal* (base-16) number systems.



System	Base	Possible digits
Binary	**2**	**0 1**
Octal	**8**	**0 1 2 3 4 5 6 7**
Decimal	**10**	**0 1 2 3 4 5 6 7 8 9**
Hexadecimal	**16**	**0 1 2 3 4 5 6 7 8 9 A B C D E F**

Digits	Binary
0	**0000**
1	**0001**
2	**0010**
3	**0011**
4	**0100**
5	**0101**
6	**0110**
7	**0111**
8	**1000**
9	**1001**
A	**1010**
B	**1011**
C	**1100**
D	**1101**
E	**1110**
F	**1111**

In a positional number system, a sequence of digits is used to represent a number. Each digit in this sequence should be a symbol in the alphabet. There is a weight associated with each position. If we count position numbers from right to left starting with zero, the weight of position n in a base b number system is b^n. For example, the number 579 in the decimal system is actually interpreted as

$$5 \times (10^2) + 7 \times (10^1) + 9 \times (10^0) .$$

(Of course, 10^0=1). In other words, 9 is in unit's place, 7 in 10's place, and 5 in 100's place.

Binary System:

The key to understanding computers is the binary number system. The binary number system has only two digits: **0 to 1**. Just as decimal notation is based on places that represent powers of ten, binary notation is based on power of two. For example, the decimal number **537** is really the sum of powers of ten:

$537 = (5\times10^2) + (3\times10^1) + (7\times10^0)$

$= (5\times100) + (3\times10) + (7\times1)$

$= 500 + 30 + 7 = 537$

Binary numbers are sums of powers of two in the same way that decimal numbers are sums of powers of ten. The following table shows the decimal numbers represented by some of more important powers of two in personal computing:

$2^{-2} = 0.25$

$2^{-1} = 0.5$

$2^0 = 1$

$2^1 = 2$

$2^2 = 4$

$2^3 = 8$

$2^4 = 16$

$2^5 = 32$

$2^6 = 64$

$2^7 = 128$

$2^8 = 256$

2^{16} = 65.536

2^{20} = 1.048.576

2^{24} = 16.777.216

A ***binary number*** is a string of 1s and 0s, each indicating the presence or absence of a power of two. For example, consider the binary number **101**. This number is converted to its decimal equivalents as follows:

binary number	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**
power of two	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
decimal number	**0**	**0**	**0**	**0**	**0**	**4**	**0**	**1**

101 binary = $(1\times2^2) + (0\times2^1) + (1\times2^0)$

= (1×4) + (0×2) + (1×1)

= 4 + 0 + 1 = 5 decimal

Thus, **101** is the binary number representation of the decimal number **5**. In a binary number such **101.1** the fractional number part is a sum of negative powers of **2.** For example, **101.1** binary is converted to its decimal equivalent as follows:

101.1 binary = $(1\times2^2) + (0\times2^1) + (1\times2^0) + (1\times2^{-1})$

= (1×4) + (0×2) + (1×1) + (1×(1/2))

= 4 + 0 + 1+ 1/2 = 5.5 decimal

Keep in mind that computers work **exclusively with binary numbers** because they can store only either a 1 or a 0. To do arithmetic and word processing they must convert from binary to decimal and back again.

In the binary system, using *n* bits, we can represent numbers from 0 through (2^n-1) for a total of 2^n different values. We need *m* bits to represent *X* different values, where

$$m = \lceil \log_2 X \rceil .$$

for example, 150 different values can be represented by using

$$\lceil \log_2 150 \rceil = \lceil 7.229 \rceil = 8 \text{ bits}.$$

In fact, using 8 bits, we can represent 2^8=256 different values (i.e., from 0 through 255).

6- Basic Terms and Notation.

The alphabet of computers, more precisely digital computers, consists of 0 and 1. Each is called a ***bit***, which stands for the binary digit. The term ***byte*** is used to represent a group of 8 bits. The term ***word*** is used to refer to a group of bytes that is processed simultaneously. The exact number of bytes that constitute a word depends on the system. For example, in the Pentium, a word refers to four bytes or 32 bits. We use the abbreviation "**b**" for bits, "**B**" for bytes, and "**W**" for words. Sometimes we also use ***doubleword*** and ***quadword***. A ***doubleword*** has twice the number of bits as the word and the ***quadword*** has four times the number of bits in a word.

2-bit → 2^2 = 4 possible states (00, 01, 10, 11)
3-bit → 2^3 = 8 possible states (000, - - -, 111)
8-bit →2^8 = 256 possible states (00000000, - - - , 11111111)

Bits in a word are usually ordered from right to left, as you would write digits in a decimal number. The rightmost bit is called the *least significant bit* (**LSB**), and the leftmost bit is called the *most significant bit* (**MSB**).

We use standard terms such as ***kilo*** (K), ***mega*** (M), ***giga*** (G), and so on to represent large integers. Unfortunately, we use two different versions of each, depending on the number system, decimal or binary. Table 1 summarizes the differences between the two systems. Typically, computer-related attributes use the binary version. For example, when we say 128 megabyte (MB) memory, we mean 128×2^{20} bytes. Usually, communication-related quantities and time units are expressed using the decimal system. For example, when we say that the data transfer rate is 100 megabits/second (Mb/s), we mean 100×10^6 Mb/s.

Table 1 Terms to represent large integer values

Term	Decimal (base 10)	Binary (base 2)
K (kilo)	10^3	2^{10}
M (mega)	10^6	2^{20}
G (giga)	10^9	2^{30}
T (tera)	10^{12}	2^{40}
P (peta)	10^{15}	2^{50}