1

Key concepts

- Data and Information
- Data processing
- Functional units of a computer
- Computer as a data processor
  - Characteristics of computer
- Number system
  - Decimal
  - Binary
  - Octal
  - Hexadecimal
- Number conversions
  - Decimal to binary
  - Decimal to octal
  - Decimal to hexadecimal
  - Binary to decimal
  - Octal to decimal
  - Hexadecimal to decimal
  - Octal to binary
  - Hexadecimal to binary
  - Octal to hexadecimal
- Binary addition
- Data representation
  - Representation of numbers
  - Representation of characters
  - Representation of audio, image and video

Fundamentals of Computer

Computers have now become an integral part of our daily life. People use computers for a variety of reasons and purposes. Be it education, business, entertainment, communication, government service or transportation, computers are inevitable today. As far as students are concerned, computers are used for learning different subjects effectively and for carrying out learning activities apart from their primary functions of computing. Try to recollect the situations where we used computers and identify the benefits you got from it. Therefore it is essential to know more about computers and its applications. This chapter presents the concepts of data processing and functional units of computer. Different data representation methods used in computers are also discussed in this chapter.

1.1 Data and Information

Many of us are familiar with the terms - data and information. We often use these terms interchangeably in our daily life. But there exists fundamental differences between these two. As part of our attempt to explore the field of computers, it is very essential to distinguish between these two terms.

Figure 1.1 shows a portion of the class diary of a teacher. Can you make out the words and numbers? Since it is a teacher's diary, these can be the names of some students. What do the numbers mean? One cannot be sure. It can be the marks scored by students in tests, their attendance for some months, or something similar. We call these facts and figures data, because they do not give a complete idea. **Data** denotes raw facts and figures such as numbers, words, amount, quantity etc. that can be processed or manipulated.

Fig. 1.1 : Sample data

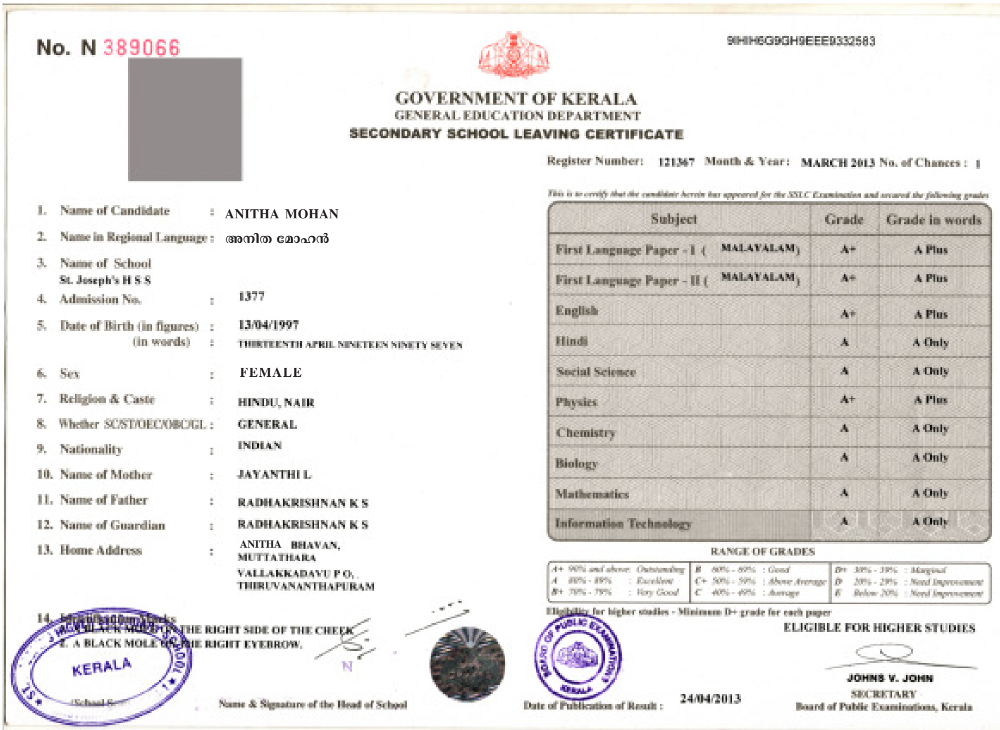
If these facts and figures were written as shown in Figure 1.2, there would be no confusion as to what they mean. It is clear that the figures show the scores obtained by students in Continuous Evaluation (CE) activities. We can see that when the data is arranged in a meaningful way, we get a clear-cut idea about these facts and figures. This is known as **information**. It is a meaningful and processed form of data.

Roll	Name	Scores out of 20			
		Asgmt	Test	Sem	Project
1	Anitha	19	19	20	19
2	Adarsh	20	18	18	19

Fig. 1.2 : Sample information

Information may also act as data in other contexts. In our example of preparation of CE scores of students, the teacher converts these numbers into a consolidated score out of 10. Similarly, after the evaluation of answer scripts of the public examination, each of these students will be awarded a score out of 40. During the preparation of results of examination, the scores of all the subjects are collected and corresponding grades are granted. The personal details and grades are put in an appropriate format with suitable labels and it becomes the mark sheet of the student, which is again information.

Figure 1.3 shows the score sheet of a student issued after the public examination. It contains personal details of the student and the grades obtained in each subject. The personal details like Anitha Mohan, Female, 13/04/1997, etc. are printed against proper labels such as Name, Sex, Date of Birth, etc. Here the facts and figures that represent the personal details and grades are the data. When these data are specified with suitable labels, it becomes information about that student. Thus we can say that the score sheet contains the information about the performance of a student in an examination. This information adds to our knowledge about the level of achievement of the students in various subjects. It also helps them to take decisions about their higher studies or to plan their future.



No. N 389066

9IHH6G9GH9EEE9332583

GOVERNMENT OF KERALA
GENERAL EDUCATION DEPARTMENT
SECONDARY SCHOOL LEAVING CERTIFICATE

Register Number: 121367 Month & Year: MARCH 2013 No. of Chances : 1

1. Name of Candidate : ANITHA MOHAN
2. Name in Regional Language : അനിത മോഹൻ
3. Name of School
St. Joseph's H S S
4. Admission No. : 1377
5. Date of Birth (in figures) : 13/04/1997
(in words) : THIRTEENTH APRIL NINETEEN NINETY SEVEN
6. Sex : FEMALE
7. Religion & Caste : HINDU, NAIR
8. Whether SC/ST/OEC/OBC/GL : GENERAL
9. Nationality : INDIAN
10. Name of Mother : JAYANTHI L
11. Name of Father : RADHAKRISHNAN K S
12. Name of Guardian : RADHAKRISHNAN K S
13. Home Address : ANITHA BHAVAN, MUTTATHARA VALLAKKADAVU P O, THIRUVANANTHAPURAM
14. Identification Marks
1. A BLACK MOLE ON THE RIGHT SIDE OF THE CHEEK
2. A BLACK MOLE ON THE RIGHT EYEBROW.

This is to certify that the candidate herein has appeared for the SSLC Examination and secured the following grades

Subject	Grade	Grade in words
First Language Paper - I (MALAYALAM)	A+	A Plus
First Language Paper - II (MALAYALAM)	A+	A Plus
English	A+	A Plus
Hindi	A	A Only
Social Science	A	A Only
Physics	A+	A Plus
Chemistry	A	A Only
Biology	A	A Only
Mathematics	A	A Only
Information Technology	A	A Only

RANGE OF GRADES

A+ 90% and above : Outstanding	B 60% - 69% : Good	D+ 30% - 39% : Marginal
A 80% - 89% : Excellent	C+ 50% - 59% : Above Average	D 20% - 29% : Need Improvement
B+ 70% - 79% : Very Good	C 40% - 49% : Average	E Below 20% : Need Improvement

Eligibility for higher studies - Minimum D+ grade for each paper

ELIGIBLE FOR HIGHER STUDIES

(School Seal) Name & Signature of the Head of School

Date of Publication of Result : 24/04/2013

JOHNS V. JOHN
SECRETARY
Board of Public Examinations, Kerala

Fig. 1.3 : SSLC score sheet

Information is always generated by performing some operations on data. In other words, data is like raw material to generate information. Now let us try to distinguish between these two terms. Table 1.1 summarises the comparison between data and information.

Data	Information
• Raw facts and figures	• Processed data
• Similar to raw material	• Similar to the finished product
• Cannot be directly used	• Adds to knowledge and helps in taking decisions
• Does not give precise and clear sense	• Clear and meaningful

Table 1.1 : Comparison between data and information

As we know, information always adds to knowledge. One can apply this knowledge to solve problems or in decision making. Generally, the ability to draw useful inferences from the acquired knowledge is known as intelligence. It depends on how we process knowledge and apply it in various situations. Recent advancement in Computer Science and technology have attempted to make computers do things, which at the moment people do better, incorporating knowledge and intelligence. This is referred to as artificial intelligence.

Let us do

- *Examine a telephone bill, electricity bill or water bill and identify the data contained in it.*
- *Think of the purchase of some items from a shop. Identify the data involved and see how it is converted into information.*
- *Identify data and information in any real life situation. Make sure that you can clearly distinguish them.*

1.2 Data processing

In the case of preparation of the score sheet mentioned in the previous section, the scores given to each subject as part of Continuous Evaluation (CE) and Terminal Evaluation (TE) are added together, and grades are determined based on some predefined criteria. The activities or operations to generate information can collectively be termed as process. **Data processing** refers to the operations or activities performed on data to generate information. So we can say that information is the result of data processing.

As shown in Figure 1.4, data is supplied for processing and information is obtained after processing. In other words, data is the input to the process and information is the output from the process.

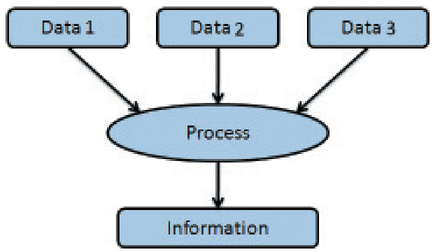


Fig. 1.4 : Data processing

Let us consider the case of the Single Window System (*Ekajaalakam*) - the admission procedure for higher secondary courses in Kerala. We can briefly list out its activities as follows:

1. The authority collects the data from applicants through application forms, in which score sheet of Class X examination will be referenced to furnish the required details. Note that in this context the facts and figures in the score sheet become the data.
2. The collected data is then fed to the computer.
3. The input data is stored and will later be retrieved for processing.
4. The data within the computer is used for performing operations such as calculations, comparisons, categorisation, sorting, filtering, etc.
5. The allotment slips for candidates and allotment lists for schools are generated. The slips and lists are printed and may be stored for later reference. It may be used as data to generate information in some other situation.
6. The slips are distributed to the applicants and lists are forwarded to schools.

Thus it is clear that data processing proceeds through six stages, as listed below:

(a) Capturing data
(b) Input of data
(c) Storage of data
(d) Processing / manipulating data
(e) Output of information
(f) Distribution of information

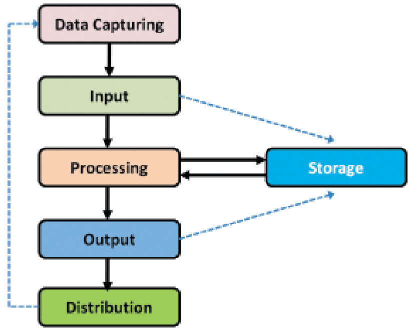
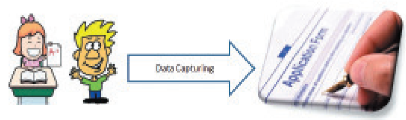


Fig. 1.5 : Stages of data processing

The thick arrow marks in Figure 1.5 indicate the flow of the activities in data processing and the dotted lines specify the flow of activities that are optional. Let us take a close look at these stages.

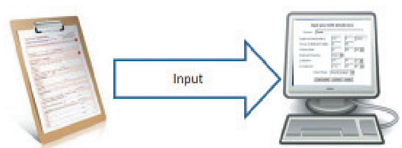
a. Capturing data



When we apply for admission to the higher secondary course, we usually provide details through a prescribed application form. The authority is actually collecting the required data for the admission process through the proforma. This is the first stage in data processing. The proforma, also known as the source document, is so designed that all relevant data to be recorded in proper order and format. Thus, preparation of hard copy of source document and data collection are the activities that take place in this stage. Today, prescribed application forms are not used for collecting data. Instead, data are directly entered through on-line facility.

b. Input

In the case of seeking admission, we submit the filled up application form to the school. There the data is extracted and fed into the computer. Sometimes, we may enter these details directly into the computer. Feeding data to the computer for processing is known as input. The input data is usually stored in computers before it is processed.

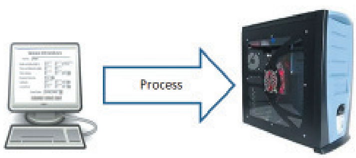


c. Storage

In many cases, the amount of data given to the computers will be large. Besides, the data entry may not be completed in a single session or a day. In the case of admissions, the data of lakhs of applicants is input to the computer. It usually takes a few weeks to complete the data entry. So the data input at different times should be stored then and there. The processing will start only after the entire data is stored. The information obtained as a result of processing is also stored in the computer. This stored data and information can be used in future for various purposes.

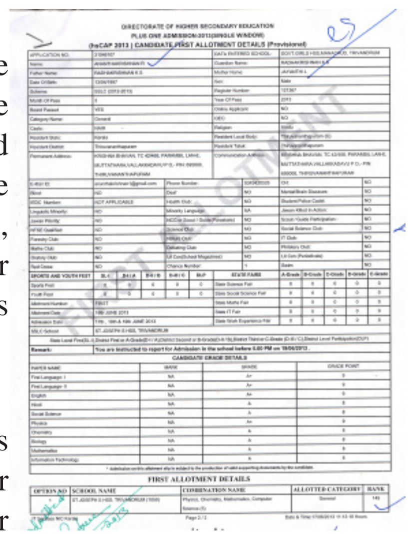
d. Process

The data stored in computers is retrieved for processing. Various operations like calculation, classification, comparison, sorting, filtering, summarising etc. are carried out as part of processing. In the case of admission to the higher secondary course, Weighted Grade Point Average (WGPA) of each applicant is calculated. Then the applicants are listed under various categories based on the descending order of WGPA. Here, school of choice, course, and performance in various co-curricular activities are considered. Finally, allotment lists for schools and allotment slips for applicants are prepared.



e. Output

The information obtained after processing will be available in this stage. Output stage should provide the information in such a form that the beneficiary should be able to take decision or solve the problem. In the case of admission to the higher secondary course, allotment slip for the applicant and allotment list for the school are generated in the desired format as outputs.


DIRECTORATE OF HIGHER SECONDARY EDUCATION
PLUS ONE ADMISSION 2013 (SINGLE WINDOW)
(hsCAP 2013 | CANDIDATE FIRST ALLOTMENT DETAILS (Provisional)

CANDIDATE GRADE DETAILS

PAPER NAME	MARK	GRADE	GRADE POINT

FIRST ALLOTMENT DETAILS

OPTION NO	SCHOOL NAME	COMBINATION NAME	ALLOTTED CATEGORY	RANK
[illegible]	[illegible]	Physics, Chemistry, Mathematics, Computer Science	[illegible]	[illegible]

f. Distribution of information

The information obtained in the output stage is distributed to the beneficiaries. They take decisions or solve problems according to the information. For example in higher secondary admission, the allotment slips are distributed to applicants for joining the school allotted and allotment lists are issued to the schools for admitting the eligible applicants. The allotment slips may be used to prepare admission register or roll list of classes. The allotment lists may be used to prepare nominal roll for registering the students for public examination.

Let us do

- *Identify and write the data processing activities in (i) opening an account in a bank and (ii) applying for scholarships*
- *Identify data processing cases in any other real life situations and write the activities performed in each stage.*

Check yourself

1. Raw facts and figures are known as ______ .
2. Processed data is known as _______.
3. Which of the following helps us to take decisions?

   (a) data (b) information (c) knowledge (d) intelligence
4. Manipulation of data to get information is known as _______.
5. Arrange the following in proper order:

   Process, Output, Storage, Distribution, Data Capture, Input
6. Pick the odd one out and give reason:

   (a) Calculation (b) Storage (c) Comparison (d) Categorization
7. Why do we store information?
8. Information may act as data. State True or False.
9. Which is the final stage in data processing?
10. What is a source document?

1.3 Functional units of a computer

Even though computers differ in size, shape, performance and cost over the years, the basic organisation of a computer is the same. It is based on a model proposed by John Von Neumann, a mathematician and a computer scientist. It consists of some functional units namely Input Unit, Central Processing Unit (CPU), Storage Unit and Output Unit. Each of these units is assigned to perform a particular task. Let us discuss the functions of these units. Figure 1.7 shows the basic functional units of a computer.

Fig. 1.6 : John Von Neumann (1903 - 1957)

1. Input unit

The collected data and the instructions for their processing are entered into the computer through the input unit. They are stored in the memory (storage unit). The data may be in different forms like number, text, image, audio, video, etc. A variety of devices are

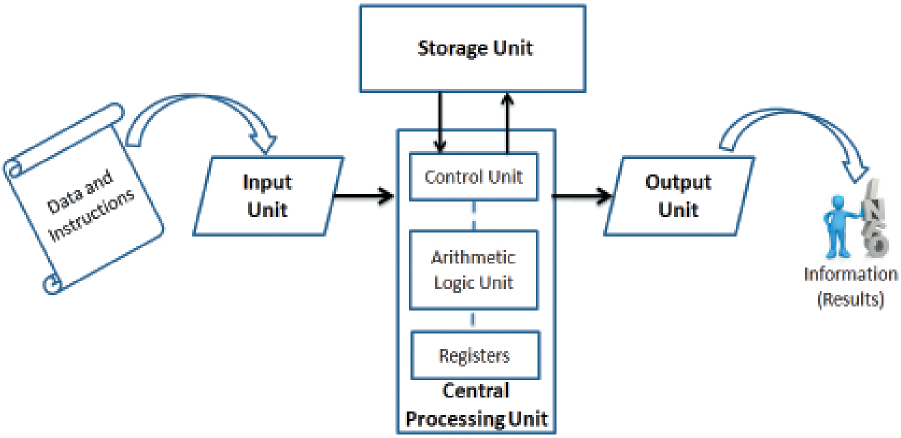


Fig. 1.7 : Functional units of a computer

available to input the data depending on its nature. Keyboard, mouse, scanner, mic, digital camera, etc. are some commonly used input devices. In short, the functions performed by input unit are as follows:

1. Accepts instructions and data from the outside world.
2. Converts these instructions and data to a form acceptable to the computer.
3. Supplies the converted instructions and data to the computer for processing.

2. Central Processing Unit (CPU)

The CPU is the brain of the computer. In a human body, all major decisions are taken by the brain and other parts of the body function as directed by the brain. Similarly, in a computer system, all major computations and comparisons are made inside the CPU. It is also responsible for activating and controlling the operations of other units of the computer. The functions of CPU are performed by three components - Arithmetic Logic Unit (ALU), Control Unit (CU) and registers.

a. Arithmetic Logic Unit (ALU)

The actual operations specified in the instructions are carried out in the Arithmetic Logic Unit (ALU). It performs calculations and logical operations such as comparisons and decision making. The data and instructions stored in the storage unit are transferred to the ALU and the processing takes place in it. Intermediate results produced by the ALU are temporarily transferred back to the storage and are retrieved later when needed for further processing. Thus there is a data flow between the storage and the ALU many times before the entire processing is completed.

b. Control Unit (CU)

Each of the functional units has its own function, but none of these will perform the function until it is asked to. This task is assigned to the control unit. It invokes the other units to take charge of the operation they are associated with. It is the central nervous system that manages and co-ordinates all other units of the computer. It obtains instructions from the program stored in the memory, interprets the operation and issues signals to the unit concerned in the system to execute them.

c. Registers

These are temporary storage elements that facilitate the functions of CPU. There are variety of registers; each designated to store unique items like data, instruction, memory address, results, etc.

3. Storage unit

The data and instructions entered in the computer through input unit are stored inside the computer before actual processing starts. Similarly, the information or results produced after processing are also stored inside the computer, before transferring to the output unit. Moreover, the intermediate results, if any, must also be stored for further processing. The storage unit of a computer serves all these purposes. In short, the specific functions of storage unit are to hold or store:

1. data and instructions required for processing.
2. intermediate results for ongoing processing.
3. final results of processing, before releasing to the output unit.

The storage unit comprises of two types as detailed below:

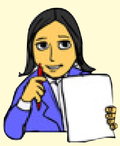
Primary storage: It is also known as main memory. It is again divided into two - Random Access Memory (RAM) and Read Only Memory (ROM). RAM holds instructions, data and intermediate results of processing. It also holds the recently produced results of the job done by the computer. ROM contains instructions for the start up procedure of the computer. The Central Processing Unit can directly access the main memory at a very high speed. But it is costly and has limited storage capacity.

Secondary storage: It is also known as auxiliary storage and it takes care of the limitations of primary storage. It has a huge storage capacity and the storage is permanent. Usually we store data, programs and information in the secondary storage, but we have to give instruction explicitly for this. Hard disk, CDs, DVDs, memory sticks, etc. are some examples.

4. Output unit

The information obtained after data processing is supplied to the outside world through the output unit in a human-readable form. Monitor and printer are the commonly used output devices. The functions performed by output unit can be concluded as follows:

1. Receives the results produced by the CPU in coded form.
2. Converts these coded results to human-readable form.
3. Supplies the results to the outside world.



Let us do

Fill up the following table by comparing human beings and the computer in the context of data processing. In the case of operations, the organs or components may be specified and for characteristics, performance may be indicated. You can also add more features.

Features	Human being	Computer
Operations		
Input	*Eyes, Ears*	*Keyboard, Mouse*
Output		
Calculation & Comparison		
Temporary Storage		
Permanent Storage		
Controlling		
Characteristics		
Speed		
Accuracy		
Reliability		
•		
•		

1.4 Computer - as a data processor

We have seen the activities involved in data processing and identified the different stages in data processing. Imagine the situation where humans are involved in these stages for performing the operations. It is sure that we will not get the information always in time and without any error all the time. We always need accurate, comprehensive, reliable and timely information in proper format and media so that it can be applied to the context concerned to formulate knowledge. Only then problems can be solved and/or decisions be made appropriately. From the discussions we had so far, computers can be considered as the best data processing machine. In short, **computer** may be defined as an electronic machine designed to accept the data and instructions, performs arithmetic and logical operations on the data according to a set of instructions and output the results or information.

1.4.1 Characteristics of computers

We have already recognized some of the characteristics of computers by performing the learning activity of filling the comparison table given in the Let us do box. As we know, computers can execute millions of instructions in a second. The results produced after processing the data are very accurate, but computers do not have adequate knowledge or intelligence to interpret the results. They only carry out instructions like an obedient servant. The computer gives correct results only if the data and instructions given are correct. The term Garbage In Garbage Out (GIGO) is used to mean this feature. That is, if a wrong input is given to the computer, it will give a wrong output. Look at Table 1.2 and identify the advantages and limitations of computer.

Computers	
Advantages	**Limitations**
Speed: A computer can perform millions of operations in a second or in fraction of second. It can do in a minute, as much work as a man do taking months and years. ***Accuracy***: A computer can perform arithmetic operations with a very high degree of accuracy. By accuracy, we mean fewer errors in the output and precision with which computations are performed. ***Diligence***: Since computer is a machine, it can operate for long hours untiringly. Unlike human beings, it will not show any emotion or disobey you. Hence computers are best suited for routine jobs. ***Versatility***: Computer can be used to perform many different kinds of processing tasks. It is a general purpose data processing machine. ***Huge memory***: Computer has enormous memory capacity. Huge volume of data can be stored in its memory for processing. The storage capacity can also be increased as per requirement.	***Lack of IQ***: Many people think that computer has super human capabilities. However this is not true. A computer does not have natural intelligence as humans have. ***Lack of decision making power***: Computer cannot decide on its own and it does not possess intuitive capabilities like human beings.

Table 1.2 : Advantages and limitations of computers

Check yourself

1. Who proposed the model of modern computers?
2. Name the components of CPU.
3. Which of the functional units of the computer is not directly involved in data processing?
4. What is meant by 'execution of an instruction'?
5. Which part of a computer can be compared to the human brain?

1.5 Number system

A number is a mathematical object used to count, label and measure. A number system is a systematic way to represent numbers. The number system we use in our day to day life is the decimal number system that uses ten symbols or digits. The number 289 is pronounced as two hundred and eighty nine and it consists of the symbols 2, 8 and 9. Similarly there are other number systems. Each has its own symbols and method for constructing a number. A number system has a unique base, which depends upon the number of symbols. The number of symbols used in a number system is called **base** or **radix** of a number system.

Let us discuss some of the number systems.

1.5.1 Decimal number system

The decimal number system involves ten symbols 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9 to form a number. Since there are 10 symbols in this number system, its base is 10. Therefore, the decimal number system is also known as base-10 number system.

Consider two decimal numbers 743 and 347

743 $\rightarrow$ seven hundred + four tens+ three ones ($7\times10^2 + 4\times10^1 + 3\times10^0$)

347 $\rightarrow$ three hundreds + four tens + seven ones ($3\times10^2 + 4 \times10^1 + 7\times10^0$)

Here, place value (weight) of 7 in first number 743 is 10^2=100. But weight of 7 in second number 347 is 10^0=1. The weight of a digit depends on its relative position. Such a number system is known as ***positional number system***. All positional number systems have a base and the place value of a digit is some power of this base.

Place value of each decimal digit is power of 10 (10^0, 10^1, 10^2, ...). Consider a decimal number 5876.

This number can be written in expanded form as

Weight	10^3	10^2	10^1	10^0
Decimal Number	5	8	7	6

$= 5\times10^3 + 8\times10^2 + 7\times10^1 + 6\times10^0$

$= 5\times1000 + 8\times100 + 7\times10 + 6\times1$

$= 5000 + 800 + 70 + 6$

$= 5876$

In the above example, the digit 5 has the maximum place value, 10^3=1000 and 6 has the minimum place value, 10^0=1. The digit with most weight is called Most Significant Digit (**MSD**) and the digit with least weight is called Least Significant Digit (**LSD**). So in the above number MSD is 5 and LSD is 6.

Left most digit of a number is MSD and right most digit of a number is LSD

For fractional numbers weights are negative powers of 10 (10^{-1}, 10^{-2}, 10^3,…..) for the digits to the right of decimal point. Consider another example 249.367

Weight	10^2	10^1	10^0	10^{-1}	10^{-2}	10^{-3}
Decimal Number	2	4	9	3	6	7

MSD (.) LSD

$= 2\times10^2 + 4\times10^1 + 9\times10^0 + 3\times10^{-1} + 6\times10^{-2} + 7\times10^{-3}$

$= 2\times100 + 4\times10 + 9\times1 + 3\times0.1 + 6\times0.01 + 7\times0.001$

$= 200 + 40 + 9 + 0.3 + 0.06 + 0.007$

$= 249.367$

So far we have discussed a number system which uses 10 symbols. Now let us see the construction of other number systems with different bases.

1.5.2 Binary number system

A number system which uses only two symbols 0 and 1 to form a number is called binary number system. Bi means two. Base of this number system is 2. So it is also called base-2 number system. We use the subscript 2 to indicate that the number is in binary.

e.g. $(1101)_2$, $(101010)_2$, $(1101.11)_2$

Each digit of a binary number is called bit. A **bit** stands for **binary digit**. The binary number system is also a positional number system where place value of each binary digit is power of 2. Consider an example $(1101)_2$. This binary number can be written in expanded form as shown below.

Weight	2^3	2^2	2^1	2^0
Binary Number	1	1	0	1

MSB LSB

$$= 1\times2^3 + 1\times 2^2 + 0\times2^1 + 1\times2^0$$
$$= 1\times8 + 1\times4 + 0\times2 + 1\times1$$
$$= 8 + 4 + 0 + 1$$
$$= 13$$

The right most bit in a binary number is called Least significant Bit (**LSB**). The leftmost bit in a binary number is called Most significant Bit (**MSB**).

The binary number 1101 is equivalent to the decimal number 13. The number 1101 also exists in the decimal number system. But it is interpreted as one thousand one hundred and one. To avoid this confusion, base must be specified in all number systems other than decimal number system. The general format is

$$\textbf{(Number)}_{\textbf{base}}$$

This notation helps to differentiate numbers of different bases. So a binary number must be represented with base 2 as $(1101)_2$ and it is read as "one one zero one to the base two".

If no base is given in a number, it will be considered as decimal. In other words, specifying the base is not compulsory in decimal number.

For fractional numbers, weights are negative powers of 2 ($2^{-1}, 2^{-2}, 2^{-3}$, ...) for the digits to the right of the binary point. Consider an example $(111.011)_2$

Weight	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
Binary Number	1	1	1	0	1	1

MSB (.) LSB

$$= 1\times2^2 + 1\times2^1 + 1\times2^0 + 0\times2^{-1} + 1\times2^{-2} + 1\times2^{-3}$$
$$= 1\times4 + 1\times2 + 1\times1 + 0\times\frac{1}{2} + 1\times \quad + 1\times$$
$$= 4 + 2 + 1 + 0 + 0.25 + 0.125$$
$$= 7.375$$

Importance of binary numbers in computers

We have seen that binary number system is based on two digits 1 and 0. The electric state ON can be represented by 1 and the OFF state by 0 as shown in Figure 1.8 Because of this, computer uses binary number system as the basic number system for data representation.

Fig. 1.8 : Digital representation of ON and OFF states

1.5.3 Octal number system

A number system which uses eight symbols 0, 1, 2, 3, 4, 5, 6 and 7 to form a number is called octal number system. Octa means eight, hence this number system is called octal. Base of this number system is 8 and hence it is also called base-8 number system. Consider an example $(236)_8$. Weight of each digit is power of 8 ($8^0, 8^1, 8^2, 8^3, \ldots$). The number $(236)_8$ can be written in expanded form as

$\frac{1}{8}$

Weight	8^2	8^1	8^0
Octal Number	2	3	6

$$= 2\times8^2 + 3\times8^1 + 6\times8^0$$
$$= 2\times64 + 3\times8 + 6\times1$$
$$= 128 + 24 + 6$$
$$= 158$$

For fractional numbers weights are negative powers of 8, i.e. ($8^{-1}, 8^{-2}, 8^{-3}, \ldots$) for the digits to the right of the octal point. Consider an example $(172.4)_8$

Weight	8^2	8^1	8^0	8^{-1}
Octal Number	1	7	2	4

$$= 1\times8^2 + 7\times8^1 + 2\times8^0 + 4\times8^{-1}$$
$$= 64 + 56 + 2 + 4\times$$
$$= 122 + 0.5$$
$$= 122.5$$

1.5.4 Hexadecimal number system

A number system which uses 16 symbols 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E and F to form a number is called hexadecimal number system. Base of this number system is 16 as there are sixteen symbols in this number system. Hence this number system is also called base-16 number system.

In this system, the symbols A, B, C, D, E and F are used to represent the decimal numbers 10, 11, 12, 13, 14 and 15 respectively. The hexadecimal digit and their equivalent decimal numbers are shown below.

Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Consider a hexadecimal number $(12AF)_{16}$. Weights of each digit is power of 16 (16^0, 16^1, 16^2, ...).

This number can be written in expanded form as shown below.

Weight	16^3	16^2	16^1	16^0
Hexadecimal Number	1	2	A	F

$$= \quad 1\times16^3 + 2\times16^2 + 10\times16^1 + 15\times16^0$$
$$= \quad 1\times4096 + 2\times256 + 10\times16 + 15\times1$$
$$= \quad 4096 + 512 + 160 + 15$$
$$= \quad 4783$$

For fractional numbers, weights are some negative power of 16 (16^{-1}, 16^{-2}, 16^{-3}, ...) for the digits to the right of the hexadecimal point. Consider an example $(2D.4)_{16}$

Weight	16^1	16^0	16^{-1}
Hexadecimal	2	D	4

$$= \quad 2\times16^1 + 13\times16^0 + 4\times$$
$$= \quad 32 + 13 + 0.25$$
$$= \quad 45.25$$

Table 1.3 shows the base and symbols used in different number systems:

Number System	Base	Symbols used
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

Table 1.3 : Number systems with base and symbols

Importance of octal and hexadecimal number systems

As we have discussed, digital hardware uses the binary number system for its operations and data. Representing numbers and operations in binary form requires too many bits and needs lot of effort. With octal, the bits are grouped in threes (because $2^3 = 8$), and with hexadecimal, the binary bits are grouped in fours (because $2^4 = 16$) and these groups are replaced with the respective octal or hexadecimal symbol. This conversion processes of binary numbers to octal and hexadecimal number systems and vice versa are very easy. This short-hand notation is widely used in the design and operations of electronic circuits.

Check yourself

1. Number of symbols used in a number system is called ____.
2. Pick invalid numbers from the following

   i) $(10101)_8$ ii) $(123)_4$ iii) $(768)_8$ iv) $(ABC)_{16}$
3. Define the term 'bit'.
4. Find MSD in the decimal number 7854.25.
5. The base of hexadecimal number system is ________.

1.6 Number conversions

After having learnt the various number systems, let us now discuss how to convert the numbers of one base to the equivalent numbers in other bases. There are different types of number conversions like decimal to binary, binary to decimal, decimal to octal etc. This section discusses how to convert one number system to another.

1.6.1 Decimal to binary conversion

The method of converting decimal number to binary number is by repeated division. In this method the decimal number is successively divided by 2 and the remainders are recorded. The binary equivalent is obtained by grouping all the remainders, with the last remainder being the Most Significant Bit (MSB) and first remainder being the Least Significant Bit (LSB). In all these cases the remainders will be either 0 or 1 (binary digit).

Example:

Find binary equivalent of decimal number 25.

		Remainders	
2	25		
2	12	1	↑ LSB
2	6	0	
2	3	0	
2	1	1	
	0	1	MSB

$(25)_{10} = (11001)_2$

Find binary equivalent of $(80)_{10}$.

		Remainders	
2	80		
2	40	0	↑ LSB
2	20	0	
2	10	0	
2	5	0	
2	2	1	
2	1	0	
	0	1	MSB

$(80)_{10} = (1010000)_2$

Hint: Binary equivalent of an odd decimal number ends with 1 and binary of even decimal number ends with zero.

Converting decimal fraction to binary

To convert a fractional decimal number to binary, we use the method of repeated multiplication by 2. At first the decimal fraction is multiplied by 2. The integer part of the answer will be the MSB of binary fraction. Again the fractional part of the answer is multiplied by 2 to obtain the next significant bit of binary fraction. The procedure is continued till the fractional part of product is zero or a desired precision is obtained.

Example: Convert 0.75 to binary.

	$0.75 \times 2 = 1.50$
1	$.50 \times 2 = 1.00$
1	$.00$

$(0.75)_{10} = (0.11)_2$

Example: Convert 0.625 to binary.

	$0.625 \times 2 = 1.25$
1	$.25 \times 2 = 0.50$
0	$.50 \times 2 = 1.00$
1	$.00$

$(0.625)_{10} = (0.101)_2$

Example: Convert 15.25 to binary.

Convert 15 to binary.

2	15	Remainders
2	7	1
2	3	1
2	1	1
	0	1

Convert 0.25 to binary.

	$0.25 \times 2 = 0.50$
0	$.50 \times 2 = 1.00$
1	$.00$

$(15.25)_{10} = (1111.01)_2$

1.6.2 Decimal to octal conversion

The method of converting decimal number to octal number is also by repeated division. In this method the number is successively divided by 8 and the remainders are recorded. The octal equivalent is obtained by grouping all the remainders, with the last remainder being the MSD and first remainder being the LSD. Remainders will be either 0, 1, 2, 3, 4, 5, 6 or 7.

Example: Find octal equivalent of decimal number 125.

8	125	Remainders	
8	15	5	LSD
8	1	7	
	0	1	MSD

$(125)_{10} = (175)_8$

Example: Find octal equivalent of $(400)_{10}$.

Divisor	Number	Remainders
8	400	
8	50	0
8	6	2
	0	6

$(400)_{10} = (620)_8$

1.6.3 Decimal to hexadecimal conversion

The method of converting decimal number to hexadecimal number is also by repeated division . In this method, the number is successively divided by 16 and the remainders are recorded. The hexadecimal equivalent is obtained by grouping all the remainders, with the last remainder being the Most Significant Digit (MSD) and first remainder being the Least Significant Digit(LSD). Remainders will be 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E or F.

Example: Find hexadecimal equivalent of decimal number 155.

Divisor	Number	Remainders	
16	155		
16	9	11 (B)	→ LSD
	0	9	→ MSD

$(155)_{10} = (9B)_{16}$

Example: Find hexadecimal equivalent of 380.

Divisor	Number	Remainders
16	380	
16	23	12 (C)
16	1	7
	0	1

$(380)_{10} = (17C)_{16}$

1.6.4 Binary to decimal conversion

A binary number can be converted into its decimal equivalent by summing up the product of each bit and its weight. Weights are some power of 2. (2^0, 2^1, 2^2, 2^3, ...)

Example: Convert $(11011)_2$ to decimal.

Weight	2^4	2^3	2^2	2^1	2^0
Bit	1	1	0	1	1

$$
\begin{aligned}
(11011)_2 &= 1\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 \\
&= 16 + 8 + 2 + 1 \\
&= 27
\end{aligned}
$$

$(11011)_2 = (27)_{10}$

Example: Convert $(1100010)_2$ to decimal.

Weight	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Bit	1	1	0	0	0	1	0

$(1100010)_2 = 1\times2^6 + 1\times2^5 + 0\times2^4 + 0\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0$

$= 64 + 32 + 2$

$= 98$

$(1100010)_2 = (98)_{10}$

Table 1.4 may help us to find powers of 2.

2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1024	512	256	128	64	32	16	8	4	2	1

Table 1.4 : Powers of 2

Converting binary fraction to decimal

A binary fraction number can be converted into its decimal equivalent by summing up the product of each bit and its weight. Weights of binary fractions are negative powers of 2 ($2^{-1}, 2^{-2}, 2^{-3}, ...$) for the digits after the binary point.

Example: Convert $(0.101)_2$ to decimal.

Weight	2^{-1}	2^{-2}	2^{-3}
Bit	1	0	1

$(0.101)_2 = 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3}$

$= 0.5 + 0 + 0.125$

$= 0.625$

$(0.101)_2 = (0.625)_{10}$

Example: Convert $(1010.11)_2$ to decimal.

Weight	2^3	2^2	2^1	2^0
Bit	1	0	1	0

$(1010)_2 = 1\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0$

$= 8 + 0+2+0$

$= 10$ $\quad (1010)_2 = (10)_{10}$

Weight	2^{-1}	2^{-2}
Bit	1	1

$(0.11)_2 = 1\times2^{-1} + 1\times2^{-2}$

$= 0.5 + 0.25$

$= 0.75$ $\quad (0.11)_2 = (0.75)_{10}$

$(1010.11)_2 = (10.75)_{10}$

Table 1.5 shows negative powers of 2.

2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
0.5	0.25	0.125	0.0625	0.03125

Table 1.5 : Negative powers of 2

1.6.5 Octal to decimal conversion

An octal number can be converted into its decimal equivalent by summing up the product of each octal digit and its weight. Weights are some powers of 8 (8^0, 8^1, 8^2, 8^3, ...).

Example: Convert $(157)_8$ to decimal.

Weight	8^2	8^1	8^0
Octal digit	1	5	7

$(157)_8 = 1\times8^2 + 5\times8^1 + 7\times8^0$

$= 64 + 40 + 7$

$= 111$

$$(157)_8 = (111)_{10}$$

Example: Convert $(1005)_8$ to decimal.

Weight	8^3	8^2	8^1	8^0
Octal digit	1	0	0	5

$(1005)_8 = 1\times8^3 + 0\times8^2 + 0\times8^1 + 5\times8^0$

$= 512+5$

$= 517$

$$(1005)_8 = (517)_{10}$$

1.6.6 Hexadecimal to decimal conversion

An hexadecimal number can be converted into its decimal equivalent by summing up the product of each hexadecimal digit and its weight. Weights are powers of 16 (16^0, 16^1, 16^2, ...).

Example: Convert $(AB)_{16}$ to decimal.

Weight	16^1	16^0
Hexadecimal digit	A	B

$(AB)_{16} = 10\times16^1+11\times16^0$

$= 160 + 11$

$= 171$

A = 10 B = 11

$$(AB)_{16} = (171)_{10}$$

Example: Convert $(2D5)_{16}$ to decimal.

Weight	16^2	16^1	16^0
Hexadecimal digit	2	D	5

$(2D5)_{16} = 2\times16^2+13\times16^1+5\times16^0$

$= 512+208+5$

$= 725$

D = 13

$$(2D5)_{16} = (725)_{10}$$

1.6.7 Octal to binary conversion

An octal number can be converted into binary by converting each octal digit to its 3 bit binary equivalent. Eight possible octal digits and their binary equivalents are listed in Table 1.6.

Octal digit	0	1	2	3	4	5	6	7
Binary equivalent	000	001	010	011	100	101	110	111

Table 1.6 : Binary equivalent of octal digit

Example: Convert $(437)_8$ to binary.

3-bit binary equivalent of each octal digits are

4	3	7
↓	↓	↓
100	011	111

$(437)_8 = (100011111)_2$

Example: Convert $(7201)_8$ to binary.

3-bit binary equivalent of each octal digits are

7	2	0	1
↓	↓	↓	↓
111	010	000	001

$(7201)_8 = (111010000001)_2$

1.6.8 Hexadecimal to binary conversion

A hexadecimal number can be converted into binary by converting each hexadecimal digit to its 4 bit binary equivalent. Sixteen possible hexadecimal digits and their binary equivalents are listed in Table 1.7.

Example: Convert $(AB)_{16}$ to binary.

4-bit binary equivalent of each hexadecimal digits are

A	B
↓	↓
1010	1011

$(AB)_{16} = (10101011)_2$

Example: Convert $(2F15)_{16}$ to binary.

4-bit binary equivalent of each hexadecimal digits are

2	F	1	5
↓	↓	↓	↓
0010	1111	0001	0101

$(2F15)_{16} = (10111100010101)_2$

Hexa decimal	Binary equivalent
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

Table 1.7 : Binary equivalent of hexadecimal digits

1.6.9 Binary to octal conversion

A binary number can be converted into its octal equivalent by grouping binary digits to group of 3 bits and then each group is converted to its octal equivalent. Start grouping from right to left.

Example: Convert $(101100111)_2$ to octal.

We can group the given binary number 101100111 from right as shown below.

101	100	111
↓	↓	↓
5	4	7

$$(101100111)_2 = (547)_8$$

Example: Convert $(10011000011)_2$ to octal.

We can group the given binary number 10011000011 from right as shown below.

After grouping, if the left most group does not have 3 bits, then add leading zeros to form 3 bit binary.

010	011	000	011
↓	↓	↓	↓
2	3	0	3

$$(10011000011)_2 = (2303)_8$$

1.6.10 Binary to hexadecimal conversion

A binary number can be converted into its hexadecimal equivalent by grouping binary digits to group of 4 bits and then each group is converted to its hexadecimal equivalent. Start grouping from right to left.

Example: Convert $(101100111010)_2$ to hexadecimal.

We can group the given binary number 101100111010 from right as shown below.

1011	0011	1010
↓	↓	↓
B	3	A

$$(101100111010)_2 = (B3A)_{16}$$

Example: Convert $(110111100001100)_2$ to hexadecimal.

We can group the given binary number 110111100001100 from right as shown below.

After grouping, if the left most group does not have 4 bits, then add leading zeros to form 4 bit binary.

0110	1111	0000	1100
↓	↓	↓	↓
6	F	0	C

$$(110111100001100)_2 = (6F0C)_{16}$$

1.6.11 Octal to hexadecimal conversion

Conversion of an octal number to hexadecimal number is a two step process. Octal number is first converted into binary. This binary equivalent is then converted into hexadecimal.

Example: Convert $(457)_8$ to hexadecimal equivalent.

First convert $(457)_8$ into binary.

$$
\begin{array}{rccc}
(457)_8 = & 4 & 5 & 7 \\
 & \downarrow & \downarrow & \downarrow \\
 & 100 & 101 & 111
\end{array}
$$

$$= (100101111)_2$$

Then convert $(100101111)_2$ into hexadecimal.

$$
\begin{array}{rccc}
(100101111)_2 = & 0001 & 0010 & 1111 \\
 & \downarrow & \downarrow & \downarrow \\
= & 1 & 2 & F
\end{array}
$$

$$= (12F)_{16}$$

$$\boxed{(457)_8 = (12F)_{16}}$$

1.6.12 Hexadecimal to octal conversion

Conversion of an hexadecimal to octal number is also a two step process. Hexadecimal number is first converted into binary. This binary equivalent is then converted into octal.

Example: Convert $(A2D)_{16}$ into octal equivalent.

First convert $(A2D)_{16}$ into binary.

$$
\begin{array}{rccc}
(A2D)_{16} = & A & 2 & D \\
 & \downarrow & \downarrow & \downarrow \\
 & 1010 & 0010 & 1101
\end{array}
$$

$$= (101000101101)_2$$

Then convert $(101000101101)_2$ into octal.

$$
\begin{array}{rcccc}
(1010\ 00101101)_2 = & 101 & 000 & 101 & 101 \\
 & \downarrow & \downarrow & \downarrow & \downarrow \\
 & 5 & 0 & 5 & 5
\end{array}
$$

$$= (5055)_8$$

$$\boxed{(A2D)_{16} = (5055)_8}$$

Table 1.8 summarises the procedures for all types of number conversions that are discussed in this chapter.

Conversion	Procedure
Decimal to Binary	Repeated division by 2 and grouping the remainders
Decimal to Octal	Repeated division by 8 and grouping the remainders
Decimal to Hexadecimal	Repeated division by 16 and grouping the remainders
Binary to Decimal	Multiply binary digit by place value(power of 2) and find their sum
Octal to Decimal	Multiply octal digit by place value (power of 8) and find their sum
Hexadecimal to Decimal	Multiply hexadecimal digit by place value (power of 16) and find their sum
Octal to Binary	Converting each octal digit to its 3 bit binary equivalent
Hexadecimal to Binary	Converting each hexadecimal digit to its 4 bit binary equivalent
Binary to Octal	Grouping binary digits to group of 3 bits from right to left
Binary to Hexadecimal	Grouping binary digits to group of 4 bits from right to left
Octal to Hexadecimal	Convert octal to binary and then binary to hexadecimal
Hexadecimal to Octal	Convert hexadecimal to binary and then binary to octal

Table 1.8: Procedure for number conversions

1.7 Binary addition

As in the case of decimal number system, arithmetic operations are performed in binary number system. When we give instruction to add two decimal numbers, the computer actually adds their binary equivalents. Let us see how binary addition is carried out. The rules for adding two bits are given in Table 1.9

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Table 1.9: Binary addition rules

Note that a carry bit 1 is created only when two ones are added. If three ones are added (i.e. 1+1+1), then the sum bit is 1 with a carry bit 1.

Example: Find sum of binary numbers 1011 and 1001.

1 0 1 1 +
<u>1 0 0 1</u>
10 1 0 0

Example: Find sum of binary numbers 110111 and 10011.

1 1 0 1 1 1 +
<u>1 0 0 1 1 0</u>
1 0 1 1 1 0 1

1.8 Data representation

Computers process different types of data such as numbers, characters, images, audios and video. We know that computer is an electronic device functioning on the basis of two states - ON and OFF. The two-state operation is called binary operation. Hence the data given to computer should also be in binary form. Data representation is the method used internally to represent data in a computer. Computer uses a fixed number of bits to represent a piece of data which could be a number, a character, image, audio, video etc. Let us see how various types of data can be represented in computer memory.

1.8.1 Representation of numbers

Numbers can be classified into integer numbers and floating point numbers. Integers are whole numbers or fixed point numbers without any fractional part. A floating point number or a real number is a number with fractional part. These two numbers are treated differently in computer memory. Let us see how integers are represented.

a. Representation of integers

There are three methods for representing an integer number in computer memory.

i) Sign and magnitude representation
ii) 1's complement representation
iii) 2's complement representation

The following data representation methods are based on 8 bit word length.

A **word** is basically a fixed-sized group of bits that are handled as a unit by a processor. Number of bits in a word is called **word length**. The word length is the choice of computer designer and some popular word lengths are 8, 16, 32 and 64.

i) Sign and magnitude representation

In this method, first bit from left (MSB) is used for representing sign of integer and remaining 7-bits are used for representing magnitude of integer. For negative integers sign bit is 1 and for positive integers sign bit is 0. Magnitude is represented as 7 bit binary equivalent of the integer.

Example: Represent + 23 in sign and magnitude form.

Number is positive, so first bit (MSB) is 0.

7 bit binary equivalent of 23 = $(0010111)_2$

So + 23 can be represented as $(00010111)_2$

Sign	← Magnitude →						
0	0	0	1	0	1	1	1

Example: Represent -105 in sign and magnitude form.

Number is negative, so first bit(MSB) is 1.

7 bit binary equivalent of 105 = $(1101001)_2$

So -105 can be represented as $(11101001)_2$

Sign	← Magnitude →						
1	1	1	0	1	0	0	1

Note: In this method an 8 bit word can represent 2^8-1 =255 numbers (i.e. -127 to +127). Similarly, a 16 bit word can represent 2^{16}-1 = 65535 numbers (i.e. -32767 to +32767). So, an *n*-bit word can represent 2^n-1 numbers i.e., –(2^{n-1}-1) to +(2^{n-1}-1). The integer 0 can be represented in two ways: +0 = 00000000 and –0 = 10000000.

ii) 1's complement representation

In this method, first find binary equivalent of absolute value of integer. If number of digits in binary equivalent is less than 8, provide zero(s) at the left to make it 8-bit form. 1's complement of a binary number is obtained by replacing every 0 with 1 and every 1 with 0. Some binary numbers and the corresponding 1's compliments are given below:

Binary Number	1's Complement
11001	00110
10101	01010

If the number is negative it is represented as 1's complement of 8-bit form binary. If the number is positive, the 8-bit form binary equivalent itself is the 1's complement representation.

Example: Represent -119 in 1's complement form.

Binary of 119 in 8-bit form = $(01110111)_2$
-119 in 1's complement form = $(10001000)_2$

Example: Represent +119 in 1's complement form.

Binary of 119 in 8-bit form = $(01110111)_2$
+119 in 1's complement form = $(01110111)_2$
(*No need to find 1's complement, since the number is positive*)

Note: In this representation if first bit (MSB) is 0 then number is positive and if MSB is 1 then number is negative. So 8 bit word can represent integers from -127 (represented as 10000000) to +127 (represented as 01111111). Here also integer 0 can be represented in two ways: +0 = 00000000 and -0 = 11111111. An *n*-bit word can represent 2^n-1 numbers i.e. -(2^{n-1}-1) to +(2^{n-1}-1).

ii) 2's complement representation

In this method, first find binary equivalent of absolute value of integer and write it in 8-bit form. If the number is negative it is represented as 2's complement of 8-bit form binary. If the number is positive 8-bit form binary itself is the representation. 2's complement of a binary number is calculated by adding 1 to its 1's complement.

For example, let us find the 2's complement of $(10101)_2$.

1's complement of $(10101)_2$ = $(01010)_2$
So 2's complement of $(10101)_2$ = 01010 +
1
= $(01011)_2$

Example: Represent -38 in 2's complement form.

Binary of 38 in 8-bit form $= (00100110)_2$

-38 in 2's complement form $=$ 11011001+
1

$= (11011010)_2$

Example: Represent +38 in 2's complement form.

Binary of 38 in 8-bit form $= (00100110)_2$

+38 in 2's complement form $= (00100110)_2$

(No need to find 2's complement)

Note: In this representation if first bit (MSB) is 0 then number is positive and if MSB is 1 then number is negative. Here integer 0 has only one way of representation and is 00000000. So an 8 bit word can represent integers from -128 (represented as 10000000) to +127(represented as 01111111). It is the most common integer representation. An *n*-bit word can represent 2^n numbers $-(2^{n-1})$ to $+(2^{n-1}-1)$.

Table 1.10 shows the various representation methods of integers in 8 bit word length.

Features	Sign & Magnitude	1's complement	2's complement	Remarks
Range	-127 to +127	-127 to +127	-128 to +127	Range is more in 2's complement
Total Numbers	255	255	256	
Representation of integer 0	Two ways for representations	Two ways for representations	Only one way for representation	In 2's complement there is no ambiguity in 0 representation
Representation of positive integers	Binary equivalent of integer in 8 bit form	Binary equivalent of integer in 8 bit form	Binary equivalent of integer in 8 bit form	All three forms are same.
Representation of negative integers	Sign bit 1 and magnitude is represented in 7 bit binary form	Find 1's complement of 8 bit form binary	Find 2's complement of 8 bit form binary	For all negative numbers MSB is 1

Table 1.10 : Representation of integers in 8-bit word length

To compare the three types of representations let us consider the following table. For clarity and easy illustration, 4-bits are used to represent the numbers in this table .

Number	Sign & Magnitude	1's complement	2's complement
-8	Not possible	Not possible	1000
-7	1111	1000	1001
-6	1110	1001	1010
-5	1101	1010	1011
-4	1100	1011	1100
-3	1011	1100	1101
-2	1010	1101	1110
-1	1001	1110	1111
0	1000 or 0000	0000 or 1111	0000
1	0001	0001	0001
2	0010	0010	0010
3	0011	0011	0011
4	0100	0100	0100
5	0101	0101	0101
6	0110	0110	0110
7	0111	0111	0111

From this table, it is clear that the MSB of a binary number indicates the sign of the corresponding decimal number irrespective of the representation. That is, if the MSB is 1, the number is negative and if it is 0, the number is positive. The table also shows that only 2's complement method can represent the maximum numbers for a given number of bits. This fact reveals that, a number below -7 and above +7 cannot be represented using 4-bits in sign & magnitude form and 1's complement form. So we go for 8-bit representation. Similarly in 2's complement method, if we want to handle numbers outside the range -8 to +7, eight bits are required.

In 8-bits implementation, the numbers from -128 to +127 can be represented in 2's complement method. The range will be -127 to +127 for the other two methods. For the numbers outside this range, we use 16 bits and so on for all the representations.

b. Representation of floating point numbers

A floating point number / real number consists of an integer part and a fractional part. A real number can be written in a special notation called the floating point notation**.** Any number in this notation contains two parts, ***mantissa*** and ***exponent*.**

For example, 25.45 can be written as 0.2545×10^{2}, where 0.2545 is the mantissa and the power 2 is the exponent. (In normalised floating point notation mantissa is between 0.1 and 1). Similarly -0.0035 can be written as $-0.35\text{x}10^{-2}$, where -0.35 is mantissa and -2 is exponent.

Let us see how a real number is represented in 32 bit word length computer. Here 24 bits are used for storing mantissa (among these the first bit for sign) and 8 bits are used for storing exponent (first bit for sign) as in Figure 1.10. Assumed decimal point is to the right of the sign bit of mantissa. No separate space is reserved for storing decimal point.

Consider the real number 25.45, that can be written as 0.2545×10^{2}, where 0.2545 is the mantissa and 2 is the exponent. These numbers are converted into binary and stored in respective locations. Various standards are followed for representing mantissa and exponent. When word length changes, bits used for storing mantissa and exponents will change accordingly.

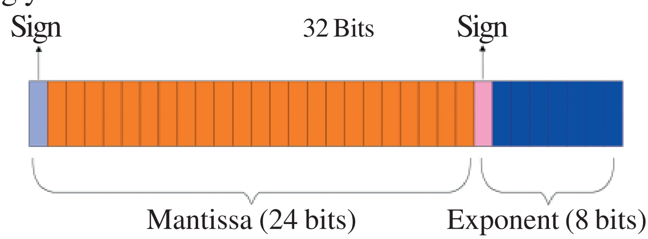


Fig 1.10: Representation of floating point numbers

In real numbers, binary point keeps track of mantissa part and exponent part. Since the value of mantissa and exponent varies from number to number the binary point is not fixed. In other words it floats and hence such a representation is called floating point representation.

1.8.2 Representation of characters

We have discussed methods for representing numbers in computer memory. Similarly there are different methods to represent characters. Some of them are discussed below.

a. ASCII

The code called ASCII (pronounced "AS-key"), which stands for American Standard Code for Information Interchange, uses 7 bits to represent each character in computer memory. The ASCII representation has been adopted as a standard by the U.S. government and is widely accepted. A unique integer number is assigned to each character. This number called ASCII code of that character is converted into binary for storing in memory. For example, ASCII code of A is 65, its binary equivalent in 7-bit is 1000001. Since there are exactly 128 unique combinations of 7 bits, this 7-bit code can represent only128 characters.

Another version is ASCII-8, also called extended ASCII, which uses 8 bits for each character, can represent 256 different characters. For example, the letter A is represented by 01000001, B by 01000010 and so on. ASCII code is enough to represent all of the standard keyboard characters.

b. EBCDIC

It stands for Extended Binary Coded Decimal Interchange Code. This is similar to ASCII and is an 8 bit code used in computers manufactured by International Business Machine (IBM). It is capable of encoding 256 characters. If ASCII coded data is to be used in a computer which uses EBCDIC representation, it is necessary to transform ASCII code to EBCDIC code. Similarly if EBCDIC coded data is to be used in a ASCII computer, EBCDIC code has to be transformed to ASCII.

c. ISCII

ISCII stands for Indian Standard Code for Information Interchange or Indian Script Code for Information Interchange. It is an encoding scheme for representing various writing systems of India. ISCII uses 8-bits for data representation. It was evolved by a standardisation committee under the Department of Electronics during 1986-88, and adopted by the Bureau of Indian Standards (BIS). Nowadays ISCII has been replaced by Unicode.

d. Unicode

Using 8-bit ASCII we can represent only 256 characters. This cannot represent all characters of written languages of the world and other symbols. Unicode is developed to resolve this problem. It aims to provide a standard character encoding scheme, which is universal and efficient. It provides a unique number for every character, no matter what the language and platform be.

Unicode originally used 16 bits which can represent up to 65,536 characters. It is maintained by a non-profit organisation called the Unicode Consortium. The Consortium first published the version 1.0.0 in 1991 and continues to develop standards based on that original work. Nowadays Unicode uses more than 16 bits and hence it can represent more characters. Unicode can represent data in almost all written languages of the world.

1.8.3 Representation of audio, image and video

In the previous sections we have discussed different data representation techniques and standards used for the computer representation of numbers and characters. While we attempt to solve real life problems with the aid of a digital computer, in most cases we may have to represent and process data other than numbers and characters. This may include audio data, images and videos. We can see that like numbers and characters, the audio, image and video data also carry information. In this section we will see different file formats for storing sound, image and video.

Digital audio, image and video file formats

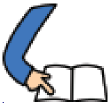
Multimedia data such as audio, image and video are stored in different types of files. The variety of file formats is due to the fact that there are quite a few approaches to compressing the data and a number of different ways of packaging the data. For example an image is most popularly stored in Joint Picture Experts Group (JPEG) file format. An image file consists of two parts - header information and image data. Information such as name of the file, size, modified data, file format, etc are stored in the header part. The intensity value of all pixels is stored in the data part of the file.

The data can be stored uncompressed or compressed to reduce the file size. Normally, the image data is stored in compressed form. Let us understand what compression is. Take a simple example of a pure black image of size 400×400 pixels. We can repeat the information black, black,…., black in all 16,0000(400×400) pixels. This is the uncompressed form, while in the compressed form black is stored only once and information to repeat it 1,60,000 times is also stored. Numerous such techniques are used to achieve compression. Depending on the application, images are stored in various file formats such as BMP (Bitmap file format), TIFF (Tagged Image File Format), GIF (Graphics Interchange Format), PNG - (Portable (Public) Network Graphic).

What we said about the header file information and compression is also applicable for audio and video files. Digital audio data can be stored in different file formats like WAV, MP3, MIDI, AIFF, etc. An audio file describes a format, sometimes referred to as the 'container format', for storing digital audio data. For example WAV file format typically contains uncompressed sound and MP3 files typically contain compressed audio data. The synthesized music data is stored in MIDI(Musical Instrument Digital Interface) files. Similarly video is also stored in different files such as AVI (Audio Video Interleave) - a file format designed to store both audio and video data in a standard package that allows synchronous audio with video simultaneous playback, MP3, JPEG-2, WMV, etc.

Check yourself

1. Which is the MSB of representation of -80 in sign and magnitude method?
2. Write 28.756 in mantissa exponent form.
3. ASCII stands for_______________.
4. Represent -60 in 1's complement form.
5. Define Unicode.
6. List any two image file formats.



Let us sum up

Data processing is a series of activities by which data is converted into information. The limitations of manual data processing are overcome by electronic data processing and the computer is the best electronic data processor. A computer has five functional units such as input unit, storage unit, arithmetic logic unit, control unit and output unit. Though the data supplied to computers is of different forms, internally these are represented using bits. Different number systems are associated with computer and any quantity on one system has an equivalent form in another system. Different types of coding systems are available to represent characters in computer. Data in the form of audio, image and video are also stored in binary form and different file formats are used to represent them.

Learning outcomes

After the completion of this chapter the learner will be able to

- distinguish between data and information.
- identify various stages in data processing.
- list the functional units of a computer and explian the functions of each.
- explain why the computer is the best electronic data processing machine.
- infer the concept of data representation inside computers.
- convert a number from one system to another.
- list the features of various coding systems to represent characters.

Sample questions

Very short answer type

1. What is data?
2. Processed data is known as ________.
3. __________ is the place value of 9 in 29610.
4. Hexadecimal number system uses ______symbols and octal number system uses _________ symbols.
5. Find octal equivalent of the decimal number 55.
6. EBCDIC stands for ________.
7. Name the coding system that can represent almost all characters used in the human languages in the world.
8. In sign and magnitude form, if the number is negative, sign bit is _______ and if number is positive, sign bit is ____.

Short answer type

1. Distinguish between data and information.
2. The application form for Plus One admission contains your personal details and your choice of groups and schools.
   a. Identify the data and information in the admission process.
   b. Explain how the information helps the applicants and the school authorities.
   c. Write down the activities involved in the processing of the data.
3. How is a computer superior to human in data processing?
4. Explain the role of storage in data processing activities.
5. List down the functions of input unit of a computer.
6. Is secondary storage essential for a computer? Justify your answer.
7. Write down the role of control unit in a computer?
8. How does the memory unit help CPU to perform its function?
9. "Computers are slaves, humans are masters". Do you agree with this? Give reasons.
10. List down the characteristics of computers.
11. Computer is a versatile machine. How?
12. What is meant by the term diligence, the characteristic of a computer?
13. Define the term data representation.
14. What do you mean by a number system? List any four number systems.
15. Find missing terms in the following series
   a. 101_2 , 1010_2, 1111_2, _________, _________.
   b. 15_8 , 16_8, 17_8, _________, _______.
   c. 18_{16}, $1A_{16}$, $1C_{16}$, ______, ______.
16. Binary number system is used for data representation in computer memory. Why?
17. Convert the following decimal numbers into binary equivalent:
   a. 25 b. 128 c. 255 d. 19.875 e. 89.25
18. Convert the following binary numbers into decimal equivalent:
   a. 1011_2 b. 111001_2 c. 1000001_2 d. 1001.11_2 e. 1111.111_2
19. Convert the following decimal numbers into octal and hexadecimal numbers
   a. 17 b. 75 c. 100 d. 199 e. 256
20. Convert the following binary numbers into octal and hexadecimal
   a. 1011_2 b. 101001_2 c. 11100011_2 d. 110001110_2 e. 10000010001_2
21. Convert the following octal numbers into decimal.
   a. 57_8 b. 101_8 c. 77_8 d. 245_8 e. 1205_8

22. Convert the following hexadecimal numbers into decimal.

    a. $2D_{16}$ b. 101_{16} c. AB_{16} d. $1F8_{16}$ e. ABC_{16}

23. Convert the following octal numbers into binary and hexadecimal.

    a. 67_8 b. 123_8 c. 167_8 d. 745_8 e. 1054_8

24. Convert the following hexadecimal numbers into binary and octal.

    a. $7F_{16}$ b. 207_{16} c. AB_{16} d. $9F8_{16}$ e. ABC_{16}

25. If $(X)_2 = (Y)_8 = (Z)_{16} = (28)_{10}$, then find X, Y and Z.

26. Arrange the following numbers in descending order

    a. $(101)_{16}$ b. 110_{10} c. 111000_2 d. 251_8

27. What are the methods of representing integers in computer memory?

28. Represent the following numbers in sign and magnitude method

    a. -19 b. +49 c. -97 d. -127

29. Represent the following numbers in 1's complement method

    a. -24 b. +69 c. -100 d. -127

30. Represent the following numbers in 2's complement method

    a. -33 b. +71 c. -111 d. -127

31. Represent - 83 in all the three number representation forms.

32. Find out the decimal number which is represented as $(10011001)_2$ in sign and magnitude method.

33. Explain the method of representing a floating point number in a 32 bit computer.

34. What are the methods of representing characters in computer memory?

35. Write short notes on ASCII and ISCII.

36. Briefly explain the significance of Unicode in character representation.

Long answer type

1. Taking the example of a real life situation like banking, briefly describe the activities involved in each stage of data processing.

2. With the help of a block diagram, explain the functional units of a computer.

3. Write down the following numbers in expanded notation

   a. $(1011.101)_2$ b. $(65356)_{10}$ c. $(A5F)_{16}$ d. $(67.4)_8$ e. $(763.452)_{10}$

4. Briefly explain different methods for representing numbers in computer memory.

5. Briefly explain different methods for representing characters in computer memory.