UNIT I

Syllabus:
Basic Functional units of Computers: functional units, basic Operational concepts, Bus structures. Software, Performance, Multiprocessors, Multicomputer.
Data Representation: Signed number representation, fixed and floating point Representations. Computer Arithmetic: Addition and subtraction, multiplication Algorithms, Division Algorithms. Error detection and correction codes.

Basic Structure of Computers

Computer Architecture in general covers three aspects of computer design namely: Computer Hardware, Instruction set Architecture and Computer Organization.

Computer hardware consists of electronic circuits, displays, magnetic and optical storage media and communication facilities.

Instruction set Architecture is programmer visible machine interface such as instruction set, registers, memory organization and exception handling. Two main approaches are mainly CISC (Complex Instruction Set Computer) and RISC (Reduced Instruction Set Computer)

Computer Organization includes the high level aspects of a design, such as memory system, the bus structure and the design of the internal CPU.

Computer Types

Computer is a fast electronic calculating machine which accepts digital input, processes it according to the internally stored instructions (Programs) and produces the result on the output device. The internal operation of the computer can be as depicted in the figure below:

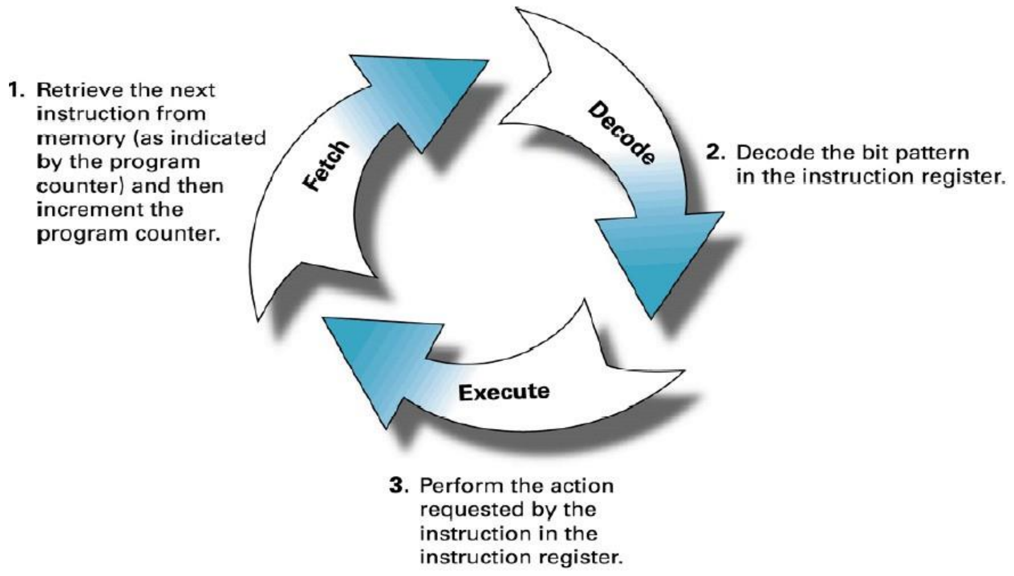

Figure 1: Fetch, Decode and Execute steps in a Computer System

The computers can be classified into various categoriesas given below:

- Micro Computer
- Laptop Computer
- Work Station
- Super Computer
- Main Frame
- Hand Held
- Multi core

Micro Computer: A personal computer; designed to meet the computer needs of an individual. Provides access to a wide variety of computing applications, such as word processing, photo editing, e-mail, and internet.

Laptop Computer: A portable, compact computer that can run on power supply or a battery unit. All components are integrated as one compact unit. It is generally more expensive than a comparable desktop. It is also called a Notebook.

Work Station: Powerful desktop computer designed for specialized tasks. Generally used for tasks that requires a lot of processing speed. Can also be an ordinary personal computer attached to a LAN (local area network).

Super Computer: A computer that is considered to be fastest in the world. Used to execute tasks that would take lot of time for other computers. For Ex: Modeling weather systems, genome sequence, etc (Refer site: http://www.top500.org/)

Main Frame: Large expensive computer capable of simultaneously processing data for hundreds or thousands of users. Used to store, manage, and process large amounts of data that need to be reliable, secure, and centralized.

Hand Held: It is also called a PDA (Personal Digital Assistant). A computer that fits into a pocket, runs on batteries, and is used while holding the unit in your hand. Typically used as an appointment book, address book, calculator and notepad.

Multi Core: Have Multiple Cores – parallel computing platforms. Many Cores or computing elements in a single chip. Typical Examples: Sony Play station, Core 2 Duo, i3, i7 etc.

GENERATION OF COMPUTERS

Development of technologies used to fabricate the processors, memories and I/O units of the computers has been divided into various generations as given below:

- First generation
- Second generation
- Third generation
- Fourth generation
- Beyond the fourth generation

First generation:
1946 to 1955: Computers of this generation used Vacuum Tubes. The computes were built using stored program concept. Ex: ENIAC, EDSAC, IBM 701.
Computers of this age typically used about ten thousand vacuum tubes. They were bulky in size had slow operating speed, short life time and limited programming facilities.

Second generation:
1955 to 1965: Computers of this generation used the germanium transistors as the active switching electronic device. Ex: IBM 7000, B5000, IBM 1401. Comparatively smaller in size About ten times faster operating speed as compared to first generation vacuum tube based computers. Consumed less power, had fairly good reliability. Availability of large memory was an added advantage.

Third generation:
1965 to 1975: The computers of this generation used the Integrated Circuits as the active electronic components. Ex: IBM system 360, PDP minicomputer etc. They were still smaller in size. They had powerful CPUs with the capacity of executing 1 million instructions per second (MIPS). Used to consume very less power consumption.

Fourth generation:
1976 to 1990: The computers of this generation used the LSI chips like microprocessor as their active electronic element. HCL horizen III, and WIPRO’S Uniplus+ HCL’s Busybee PC etc.
They used high speed microprocessor as CPU. They were more user friendly and highly reliable systems. They had large storage capacity disk memories.

Beyond Fourth Generation:
1990 onwards: Specialized and dedicated VLSI chips are used to control specific functions of these computers. Modern Desktop PC’s, Laptops or Notebook Computers.

Functional Unit

A computer in its simplest form comprises five functional units namely input unit, output unit memory unit, arithmetic & logic unit and control unit. Figure 2 depicts the functional units of a computer system.

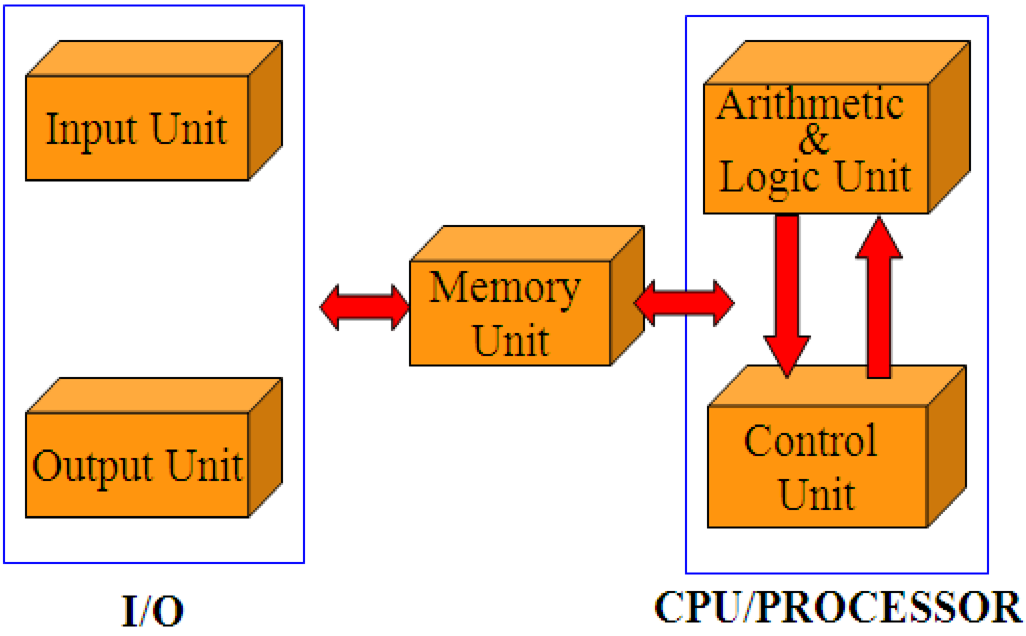

Figure 2: Basic functional units of a computer

Let us discuss about each of them in brief:

1. **Input Unit:** Computer accepts encoded information through input unit. The standard input device is a keyboard. Whenever a key is pressed, keyboard controller sends the code to CPU/Memory.

   Examples include Mouse, Joystick, Tracker ball, Light pen, Digitizer, Scanner etc.

2. **Memory Unit:** Memory unit stores the program instructions (Code), data and results of computations etc. Memory unit is classified as:

   - Primary /Main Memory
   - Secondary /Auxiliary Memory

Primary memory is a semiconductor memory that provides access at high speed. Run time program instructions and operands are stored in the main memory. Main memory is classified again as ROM and RAM. ROM holds system programs and firmware routines such as BIOS, POST, I/O Drivers that are essential to manage the hardware of a computer. RAM is termed as Read/Write memory or user memory that holds run time program instruction and data. While primary storage is essential, it is volatile in nature and expensive. Additional requirement of memory could be supplied as auxiliary memory at cheaper cost. **Secondary memories** are non volatile in nature.

3. **Arithmetic and logic unit:** ALU consist of necessary logic circuits like adder, comparator etc., to perform operations of addition, multiplication, comparison of two numbers etc.

4. **Output Unit:** Computer after computation returns the computed results, error messages, etc. via output unit. The standard output device is a video monitor, LCD/TFT monitor. Other output devices are printers, plotters etc.

5. **Control Unit:** Control unit co-ordinates activities of all units by issuing control signals. Control signals issued by control unit govern the data transfers and then appropriate operations take place. Control unit interprets or decides the operation/action to be performed.

The operations of a computer can be summarized as follows:

1. A set of instructions called a program reside in the main memory of computer.

2. The CPU fetches those instructions sequentially one-by-one from the main memory, decodes them and performs the specified operation on associated data operands in ALU.

3. Processed data and results will be displayed on an output unit.

4. All activities pertaining to processing and data movement inside the computer machine are governed by control unit.

Basic Operational Concepts

An Instruction consists of two parts, an Operation code and operand/s as shown below:

OPCODE	OPERAND/s

Let us see a typical instruction

ADD   LOCA, R0

This instruction is an addition operation. The following are the steps to execute the instruction: Step 1: Fetch the instruction from main memory into the processor

Step 2: Fetch the operand at location LOCA from main memory into the processor

Step 3: Add the memory operand (i.e. fetched contents of LOCA) to the contents of register R0 Step 4: Store the result (sum) in R0.

The same instruction can be realized using two instructions as

Load   LOCA,
R1   Add   R1,
R0

The steps to execute the instructions can be enumerated as below:

Step 1: Fetch the instruction from main memory into the processor

Step 2: Fetch the operand at location LOCA from main memory into the processor Register R1

Step 3: Add the content of Register R1 and the contents of register R0 Step 4: Store the result (sum) in R0.

Figure 3 below shows how the memory and the processor are connected. As shown in the diagram, in addition to the ALU and the control circuitry, the processor contains a number of registers used for several different purposes. The instruction register holds the instruction that is currently being executed. The program counter keeps track of the execution of the program. It contains the memory address of the next instruction to be fetched and executed. There are n general purpose registers R0 to R_{n-1} which can be used by the programmers during writing programs.

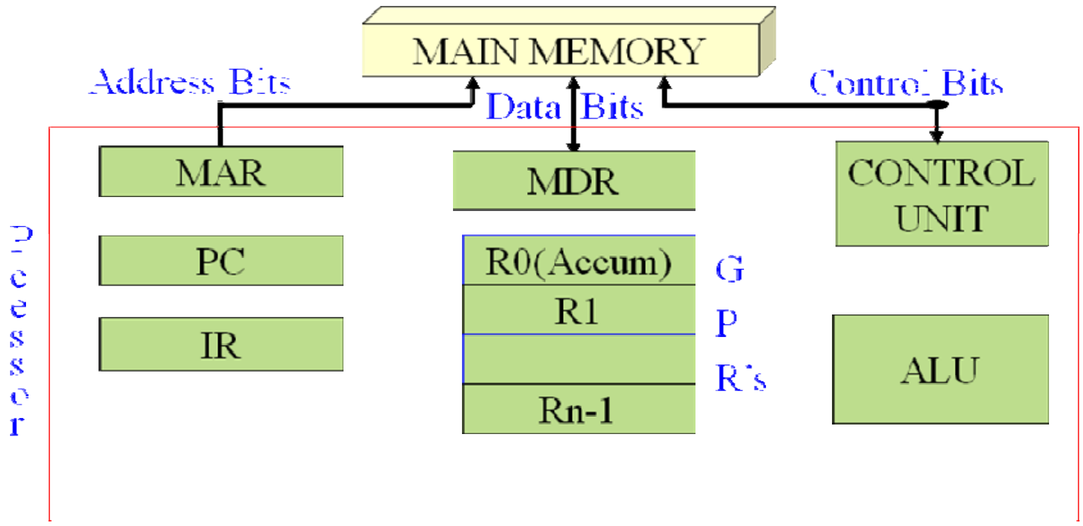

Figure 3: Connections between the processor and the memory

The interaction between the processor and the memory and the direction of flow of information is as shown in the diagram below:

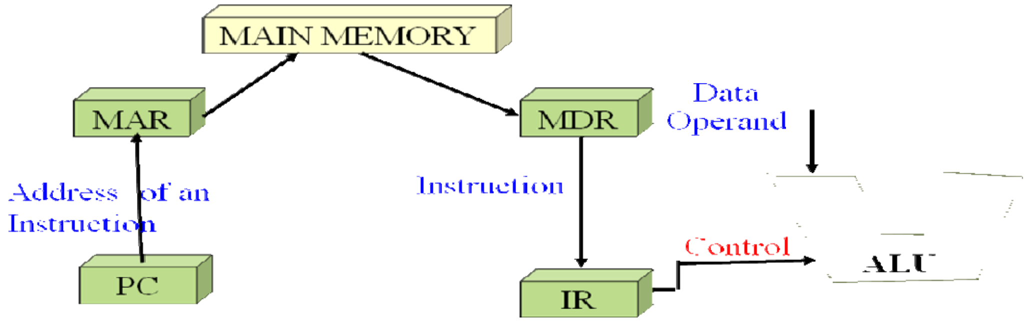

Figure 4: Interaction between the memory and the ALU

BUS STRUCTURES

Group of lines that serve as connecting path for several devices is called a bus (one bit per line). Individual parts must communicate over a communication line or path for exchanging data, address and control information as shown in the diagram below. Printer example – processor to printer. A common approach is to use the concept of buffer registers to hold the content during the transfer.

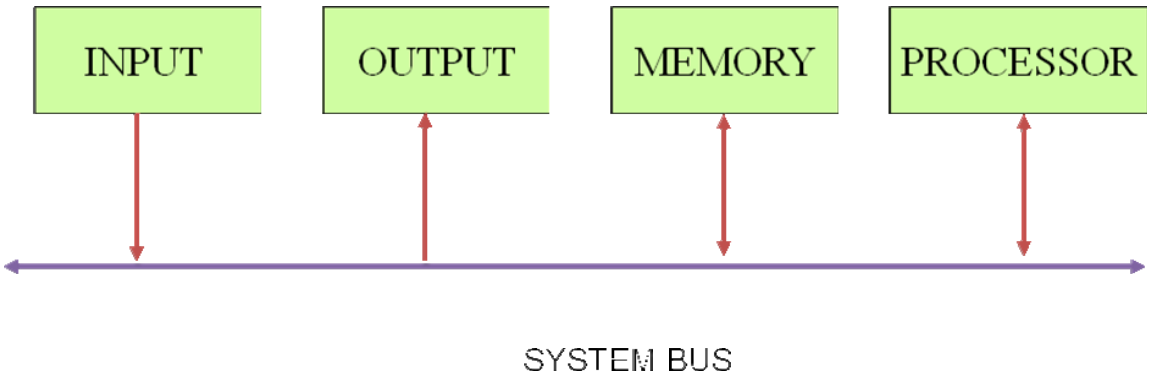

Figure 5: Single bus structure

SOFTWARE

If a user wants to enter and run an application program, he/she needs a System Software. System Software is a collection of programs that are executed as needed to perform functions such as:

- Receiving and interpreting user commands
- Entering and editing application programs and storing then as files in secondary storage devices
- Running standard application programs such as word processors, spread sheets, games etc…

Operating system - is key system software component which helps the user to exploit the below underlying hardware with the programs.

USER PROGRAM and OS ROUTINE INTERACTION

Let's assume computer with 1 processor, 1 disk and 1 printer and application program is in machine code on disk. The various tasks are performed in a coordinated fashion, which is called multitasking. t0, t1 …t5 are the instances of time and the interaction during various instances as given below:

t0: the OS loads the program from the disk to memory t1: program executes
t2: program accesses disk
t3: program executes some more t4: program accesses printer
t5: program terminates

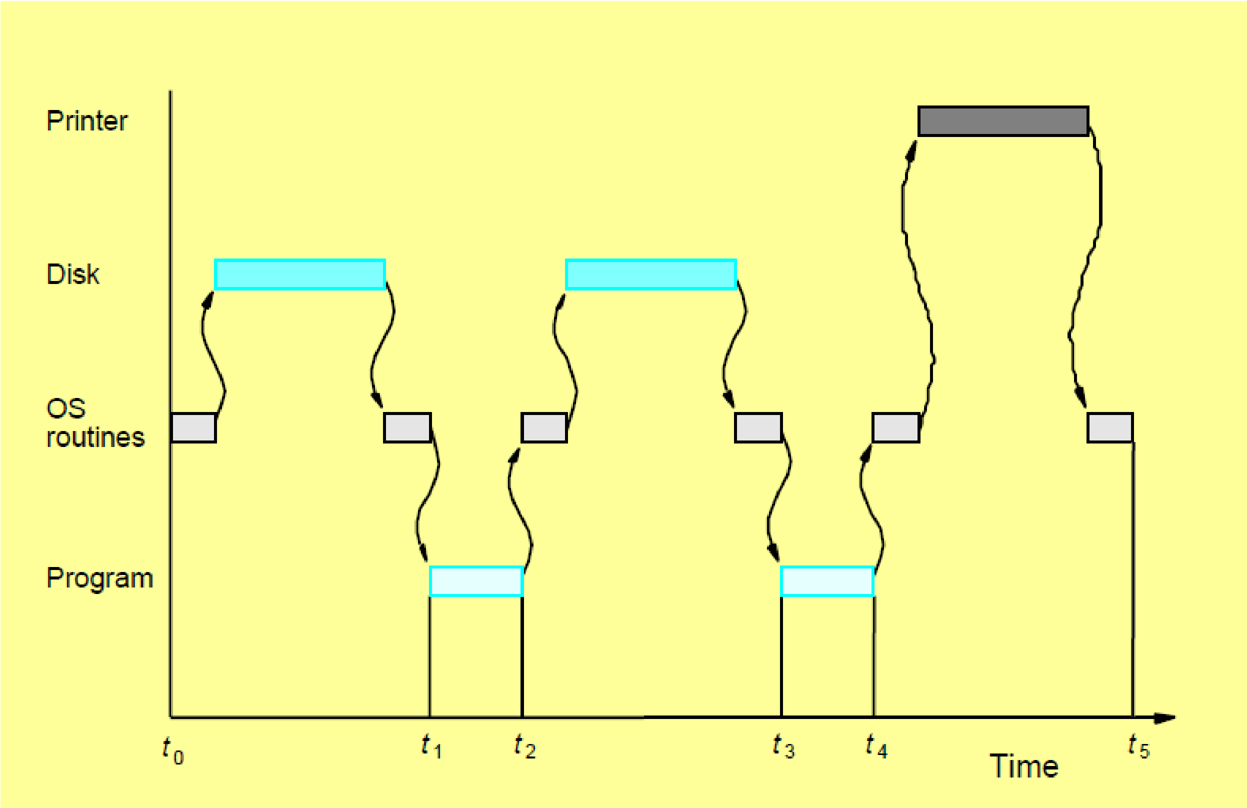

Figure 6 :User program and OS routine sharing of the processor

PERFORMANCE

The most important measure of the performance of a computer is how quickly it can execute programs. The speed with which a computer executes program is affected by the design of its hardware. For best performance, it is necessary to design the compiles, the machine instruction set, and the hardware in a coordinated way.

The total time required to execute the program is elapsed time is a measure of the performance of the entire computer system. It is affected by the speed of the processor, the disk and the printer. The time needed to execute a instruction is called the processor time.

Just as the elapsed time for the execution of a program depends on all units in a computer system, the processor time depends on the hardware involved in the execution of individual machine instructions. This hardware comprises the processor and the memory which are usually connected by the bus.

The pertinent parts of the fig. c is repeated in fig. d which includes the cache memory as part of the processor unit.

Let us examine the flow of program instructions and data between the memory and the processor. At the start of execution, all program instructions and the required data are stored in the main memory. As the execution proceeds, instructions are fetched one by one over the bus into the processor, and a copy is placed in the cache later if the same instruction or data item is needed a second time, it is read directly from the cache.
The processor and relatively small cache memory can be fabricated on a single IC chip. The internal speed of performing the basic steps of instruction processing on chip is very high and is considerably faster than the speed at which the instruction and data can be fetched from the main memory. A program will be executed faster if the movement of instructions and data between the main memory and the processor is minimized, which is achieved by using the cache.

For example:- Suppose a number of instructions are executed repeatedly over a short period of time as happens in a program loop. If these instructions are available in the cache, they can be fetched quickly during the period of repeated use. The same applies to the data that are used repeatedly.

Processor clock:

Processor circuits are controlled by a timing signal called clock. The clock designer the regular time intervals called clock cycles. To execute a machine instruction the processor divides the action to be performed into a sequence of basic steps that each step can be completed in one clock cycle. The length P of one clock cycle is an important parameter that affects the processor performance.
Processor used in today's personal computer and work station have a clock rates that range from a few hundred million to over a billion cycles per second.

Basic performance equation:

We now focus our attention on the processor time component of the total elapsed time. Let 'T' be the processor time required to execute a program that has been prepared
in some high-level language. The compiler generates a machine language object program that corresponds to the source program. Assume that complete execution of the program requires the execution of N machine cycle language instructions. The number N is the actual number of instruction execution and is not necessarily equal to the number of machine cycle instructions in the object program. Some instruction may be executed more than once, which in the case for instructions inside a program loop others may not be executed all, depending on the input data used.
Suppose that the average number of basic steps needed to execute one machine
cycle instruction is S, where each basic step is completed in one clock cycle. If clock rate is 'R' cycles per second, the program execution time is given by

$$T=N*S/R$$

this is often referred to as the basic performance equation.
We must emphasize that N, S & R are not independent parameters changing one may affect another. Introducing a new feature in the design of a processor will lead to improved performance only if the overall result is to reduce the value of T.

Performance measurements:

It is very important to be able to access the performance of a computer, comp designers use performance estimates to evaluate the effectiveness of new features.
The previous argument suggests that the performance of a computer is given by the execution time T, for the program of interest.
Inspite of the performance equation being so simple, the evaluation of 'T' is highly complex. Moreover the parameters like the clock speed and various architectural features are not reliable indicators of the expected performance.
Hence measurement of computer performance using bench mark programs is done to make comparisons possible, standardized programs must be used.
The performance measure is the time taken by the computer to execute a given bench mark. Initially some attempts were made to create artificial programs that could be used

as bench mark programs. But synthetic programs do not properly predict the performance obtained when real application programs are run.
A non profit organization called SPEC- system performance evaluation corporation selects and publishes bench marks.
The program selected range from game playing, compiler, and data base applications to numerically intensive programs in astrophysics and quantum chemistry. In each case, the program is compiled under test, and the running time on a real computer is measured. The same program is also compiled and run on one computer selected as reference.
The 'SPEC' rating is computed as follows.

$$\text{SPEC rating} = \frac{\text{Running time on the reference computer}}{\text{Running time on the computer under test}}$$

If the SPEC rating = 50

Multiprocessor & microprocessors:

Large computers that contain a number of processor units are called multiprocessor system. These systems either execute a number of different application tasks in parallel or execute subtasks of a single large task in parallel. All processors usually have access to all memory locations in such system & hence they are called shared memory multiprocessor systems. The high performance of these systems comes with much increased complexity and cost. In contrast to multiprocessor systems, it is also possible to use an interconnected group of complete computers to achieve high total computational power. These computers normally have access to their own memory units when the tasks they are executing need to communicate data they do so by exchanging messages over a communication network. This properly distinguishes them from shared memory multiprocessors, leading to name message-passing multi computer.

Data Representation:

Information that a Computer is dealing with

Data

Numeric Data

Numbers(Integer, real)
Non-numeric Data
Letters, Symbols

Relationship between data elements

Data Structures

Linear Lists, Trees, Rings, etc
Program(Instruction)

Numeric Data Representation

Fixed Point

Decimal	Binary	Octal	Hexadecimal
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	7
08	1000	10	8
09	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Representation:

It's the representation for integers only where the decimal point is always fixed. i.e at the end of rightmost point. it can be again represented in two ways.

1. Sign and Magnitude Representation

In this system, he most significant (leftmost) bit in the word as a sign bit. If the sign bit is 0, the number is positive; if the sign bit is 1, the number is negative.

The simplest form of representing sign bit is the sign magnitude representation.

One of the draw back for sign magnitude number is addition and subtraction need to consider both sign of the numbers and their relative magnitude.

Another drawback is there are two representation for 0(Zero) i.e +0 and -0.

2. One's Complement (1's) Representation

In this representation negative values are obtained by complementing each bit of the corresponding positive number.

For example 1s complement of 0101 is 1010 . The process of forming the 1s complement of a given number is equivalent to subtracting that number from 2n -1 i.e from 1111 for 4 bit number.

Two's Complement (2's) Representation Forming the 2s complement of a number is done by subtracting that number from 2n . So 2s complement of a number is obtained by adding 1 to 1s complement of that number.

Ex: 2's complement of 0101 is 1010 +1 = 1011

NB: In all systems, the leftmost bit is 0 for positive number and 1 for negative number.

Floating-point representation

Floating-point numbers are so called as the decimal or binary point floats over the base depending on the exponent value.

It consists two components.

- Exponent
- Mantissa

Example: Avogadro's number can be written as 6.02x1023 in base 10. And the mantissa and exponent are 6.02 and 1023 respctivly. But computer floating-point numbers are usually based on base two. So 6.02x1023 is approximately (1 and 63/64)x278 or 1.111111 (base two) x 21001110 (base two)

Error Detection Codes

Parity System

Hamming Distance

CRC

Check sum