

# **C-Programming Viva Questions**

## **1) What do you mean by Hardware and Software?**

All the electronic/electrical components and circuits used in a computer system are called hardware. A computer is actuated and controlled with the help of computer programs called software.

## **2) Mention the main components of a computer and their functions.**

- CPU (central processing unit) – to process the data
- Input Device – to enter the data into the computer.
- Output Device – to display / print results by the computer.

## **3) What is Operating System (OS)?**

An operating system is a collection of programs used to connect the user with the hardware it has the set of programs which controls the operations of the hardware components such as CPU, main memory, keyboard, monitor, and printer and so on.

## **4) What is Algorithms?**

Algorithms refer to the step by step instructions written to solve any problem.

## **5) What is Flowchart?**

A flowchart is a diagrammatic or symbolic representation of algorithms. It uses various symbols to represent the operations to be performed.

## **6) Name the four basic data types in “C” language?**

The four basic data types in “c” language are as follows

- char – a character
- int – an integer, in the range -32,767 to 32,767
- long int – a larger integer (up to +-2,147,483,647)
- float – a floating-point number
- double – a floating-point number, with more precision and perhaps greater range than float

## **7) Describe at least five different format specifies?**

- %d: -An integer whole number
- %f: -a floating point number
- %c: -a single character
- %s: -a string of value of characters.

### **8) Define and explain scanf () function?**

The Scanf () function can be used to get input into a program and it requires two arguments. First a format specified defines the type of data to be entered, then the name of the variable in which the input will be stored. This scanf () function is responsible for giving input into the program.

### **9) Define and explain printf () function?**

The printf () function is used to display/output values of variable in the monitor. The printf function has general form: printf ("format specifies", variables)

### **10) What are the maximum and minimum possible ranges of values for long and short type?**

If the int variable is created by default as a 'long' type it typically will have a possible range of values from a maximum of +214748347 and a minimum of -2147483648. For 'short' type these are the maximum and minimum values +327676 and minimum -32768.(While answering this question you can specify the approximate value raised to power).

### **11) What is preprocessor?**

The preprocessor is a program which is executed before the compilation of a source program written by the user. The preprocessor directives begins with hash (#) followed by the command. e.g #includes – it is a directive to include file.

### **12) What exactly is a 'variable scope', 'local variables' and 'global variables'?**

The extent to which a variable is accessible in a program is called the 'variable scope'. Variables declared internally inside a function are known as 'local' variables. Variables declared externally outside a function are known as 'global' variables.

### **13) What are signed values?**

When an int variable is declared it can by default contain either positive or negative integer values. These are known as 'signed' values. The range of positive values is determined by your system.

### **14) Define reserved words.**

C programs are constructed from a set of reserved words which provide control and from libraries which perform special functions. The basic instructions are built up using a reserved set of words, such as main, for, if, while, default, double, extern, for, and int, to name just a few.

### **15) What is the purpose of type declaration in C?**

All variables used in a C program are declared using the appropriate data types to enable the compiler to allocate the required number of bytes in RAM to store values of these variables in memory.

### **16) What is identifier?**

An identifier is a name used to identify a variable, function, symbolic constant and so on.

### **17) Mention the different types of operators used in C?**

- Arithmetic operator
- Relational operators
- Logical Operators
- Increment and decrements operators
- 5. Assignment operators
- 6. Conditional operator
- Bitwise operators

### **18) What are Loop control statements?**

Loop control structures are used to execute and repeat a block of statements depending on the value of a condition. There are 3 types of loop control statements in C

- for loop
- while loop
- do – while loop

### **19) Explain while loop.**

A while loop has one control expression, and executes as long as that expression is true. The general syntax of a while loop is

```
while ( expression ){  
    Statements;  
}
```

We use a while loop when a statement or group of statements which may have to be executed a number of times to complete their task. The controlling expression represents the condition

### **20) Explain for loop.**

A for loop is used to execute and repeat a block of statements depending on a condition. The syntax of a for loop is

```
for( ; ; )  
{  
    Statements;  
}
```

### **21) What do mean by network?**

Computer networking refers to connecting computers to share data, application software and hardware devices. Networks allow sharing of information among various computers and permit users to share files

## **22) List a few unconditional control statements in C.**

- break statement
- continue statement
- goto statement
- exit() function

## **23) What is an array?**

An array is a collection of values of the same data type. Values in array are accessed using array name with subscripts in brackets []. Syntax of array declaration is

Data type array\_ name [size];

## **24) What is Multidimensional Arrays?**

An array with more than one index value is called a multidimensional array. To declare a multidimensional array you can do follow syntax

data type array\_ name[] [] []....;

## **25) Define string?**

An array of characters is known as a string. For example

charst[8]; this statement declares a string array with 80 characters .

## **26) Mention four important string handling functions in C languages.**

There are four important string handling functions in C languages.

- strlen();
- strcpy();
- strcat();
- strcmp();

The header file #include is used when these functions are called in a C program.

## **27) Explain about the constants which help in debugging?**

A #if directive test can be offered with #else and #else if directives. This allows conditional branching of the program to run sections of the code according to the result. Constants defined with a #define directive can be undefined with the #undef directive. The #ifdef directive has a companion directive #ifndef. These commands can be useful when debugging problem code to hide and unhide sections of the program.

## **28) Define and explain about! Operator?**

The logical operator! NOT is a unary operator that is used before a single operand. It returns the inverse value of the given operand so if the variable “c” had a value of true then! C would return value of false. The not operator is very much useful in C programs because it can change the value of variables with successful iterations. This ensures that on each pass the value is changed.

## **29) What is operator precedence?**

Operator precedence defines the order in which C evaluates expressions.

e.g. in the expression  $a=6+b*3$ , the order of precedence determines whether the addition or the multiplication is completed first. Operators on the same row have equal precedence.

## **30) Explain about the functions strcat() and strcmp()?**

This function concatenates the source string at the end of the target string. Strcmp() function compares two strings to find out whether they are the same or different. The two strings are compared character by character until there is a mismatch or end of one of the strings is reached, whichever occurs first. If in case two strings are identical, a value of zero is returned. If there is no matches between two strings then a difference of the two non-matching values are returned according to ASCII values.

## **31) Define function**

A function is a module or block of program code which deals with a particular task. Each function has a name or identifier by which is used to refer to it in a program. A function can accept a number of parameters or values which pass information from outside, and consists of a number of statements and declarations, enclosed by curly braces { }, which make up the doing part of the object

## **32) Differentiate built-in functions and user – defined functions.**

Built – in functions are used to perform standard operations such as finding the square root of a number, absolute value and so on. These are available along with the C compiler and are included in a program using the header files math.h, string.h and so on.

User defined functions are written by the user or programmer to compute a value or perform a task. It contains a statement block which is executed during the runtime whenever it is called by the main program.

## **33) Distinguish between actual and formal arguments.**

Actual arguments are variables whose values are supplied to the function in any function call. Formal arguments are variables used to receive the values of actual arguments from the calling program.

### **34) Explain the concept and use of type void.**

A function which does not return a value directly to the calling program is referred as a void function. The void functions are commonly used to perform a task and they can return many values through global variable declaration.

### **35) What is recursion?**

A function calling itself again and again to compute a value is referred to as recursive function or recursion. Recursion is useful for branching processes and is effective where terms are generated successively to compute a value.

### **36) Mention the types of network.**

A simple network consist of computers connected using network interface cards, networking software and network cables. There are two main networking arrangements

- client / server – a powerful computer is used as the server which works as the interpreter between the clients and helps sharing files.
- peer to peer – there is no server and all the workstations are treated equally.

### **37) What are Library functions?**

Library functions are built in programs available along with the compiler which perform some standard mathematical operations.

### **38) How does the type float differ from double in C language?**

Float data type refers real number in single precision and has 6 decimal digits. It takes 4 bytes in memory to refer values ranging from  $3.4e-38$  to  $3.4e+38$

double data type also refers to real number but in double precision and has 12 decimal digits. It takes 8 bytes of memory to refer values ranging from  $1.7e-308$  to  $1.7e+308$

### **39) What is an operator and operand?**

An operator performs an operation like addition, subtraction and so on and produces a value. Variables and constants upon which operations are performed are called operands.

### **40) What is RAM?**

RAM – Random Access Memory is a temporary storage medium in a computer. RAM is a volatile memory i.e all data stored in RAM will be erased when the computer is switched off.

### **41) What is ROM?**

ROM – Read Only Memory is permanent storage medium which stores start up programs (operating system programs) and BIOS programs which are recorded by the manufacturer of the compiler system. ROM is a non-volatile memory.

#### **42) Define system software.**

System software is a collection of programs which are used to assist the user to handle the computer hardware like printer, disk and so on and execute the application programs.

#### **43) Define application software**

Application software's are programs which are used to solve specific problems /tasks. Examples include railway reservation, banking and so on.

#### **44) What are control statements?**

All the statements written in a program are executed from top to bottom one by one. Control statements are used to execute / transfer the control from one part of the program to another depending on conditions.

#### **45) What is Parallel Computation?**

Computations that use multi-processor computers and/or several independent computers interconnected in some way, working together on a common task.

- Examples: CRAY T3E, IBM-SP, SGI-3K, Cluster of Workstations.

#### **46) Why use Parallel Computation?**

- Computing power (speed, memory)
- Cost/Performance
- Scalability
- Tackle intractable problems

#### **47) What does Open MP stand for?**

Short version: Open Multi-Processing

Long version: Open specifications for Multi-Processing via collaborative work between interested parties from the hardware and software industry, government and academia.

#### **48) Explain increment and decrements operators.**

++ increment operator which add one to the value,  
example : i++ (which adds one to i and results is scored back to)  
— decrement operator which subtracts one from the value  
example — i ( which subtracts one from i)

#### **49) Mention the types of memory**

Two major types of memory storage is primary memory and secondary memory. Primary storage (or main memory or internal memory), often referred to simply as memory, is the only one directly accessible to the CPU.

Secondary memory (or external memory) differs from primary storage in that it is not directly accessible by the CPU. Some of the example for secondary memory includes floppy disks, flash memory, magnetic tape, hard drives etc.

#### **50) What are input and output device?**

Input and Output Devices: Input devices are the hardware that are used for providing information to the computer like mouse and keyboard and output devices are the hardware that are used for receiving information from computer like monitor, printer or the sound system.

#### **51) What are different storage class specifiers in C?**

Auto, register, static, extern

#### **52) What is scope of a variable? How are variables scoped in C?**

Scope of a variable is the part of the program where the variable may directly be accessible. In C, all identifiers are lexically (or statically) scoped. See [this](#) for more details.

#### **53) How will you print “Hello World” without semicolon?**

```
intmain(void)

{
    if printf("Hello World");
}
```

#### **54) What is NULL pointer?**

NULL is used to indicate that the pointer doesn't point to a valid location. Ideally, we should initialize pointers as NULL if we don't know their value at the time of declaration. Also, we should make a pointer NULL when memory pointed by it is deal located in the middle of a program.

#### **55) When should we use pointers in a C program?**

- To get address of a variable
- For achieving pass by reference in C: Pointers allow different functions to share and modify their local variables.
- To pass large structures so that complete copy of the structure can be avoided.
- To implement “linked” data structures like linked lists and binary trees.



# **C++ VIVA Question and Answer**

## **1. What is the full form of OOPS?**

Object Oriented Programming System.

## **2. What is a class?**

Class is a blue print which reflects the entities attributes and actions. Technically defining a class is designing a user defined data type.

## **3. What is an object?**

An instance of the class is called as object.

## **4. List the types of inheritance supported in C++.**

Single, Multilevel, Multiple, Hierarchical and Hybrid.

## **5. What is the role of protected access specifier?**

If a class member is protected then it is accessible in the inherited class. However, outside the both the private and protected members are not accessible.

## **6. What is encapsulation?**

The process of binding the data and the functions acting on the data together in an entity (class) called as encapsulation.

## **7. What is abstraction?**

Abstraction refers to hiding the internal implementation and exhibiting only the necessary details.

## **8. What is inheritance?**

Inheritance is the process of acquiring the properties of the exiting class into the new class. The existing class is called as base/parent class and the inherited class is called as derived/child class.

## **9. Explain the purpose of the keyword volatile.**

Declaring a variable volatile directs the compiler that the variable can be changed externally. Hence avoiding compiler optimization on the variable reference.

## **10. What is an inline function?**

A function prefixed with the keyword inline before the function definition is called as inline function. The inline functions are faster in execution when compared to normal functions as the compiler treats inline functions as macros.

## **10. What is a storage class?**

Storage class specifies the life or scope of symbols such as variable or functions.

**11. Mention the storage class's names in C++.**

The following are storage classes supported in C++: auto, static, extern, register and mutable

**12. What is the role of mutable storage class specifier?**

A constant class object's member variable can be altered by declaring it using mutable storage class specifier. Applicable only for non-static and non-constant member variable of the class.

**13. Distinguish between shallow copy and deep copy.**

Shallow copy does memory dumping bit-by-bit from one object to another. Deep copy is copy field by field from object to another. Deep copy is achieved using copy constructor and or overloading assignment operator.

**14. What is a pure virtual function?**

A virtual function with no function body and assigned with a value zero is called as pure virtual function.

**15. What is an abstract class in C++?**

A class with at least one pure virtual function is called as abstract class. We cannot instantiate an abstract class.

**16. What is a reference variable in C++?**

A reference variable is an alias name for the existing variable. Which means both the variable name and reference variable point to the same memory location. Therefore updation on the original variable can be achieved using reference variable too.

**17. What is the role of static keyword on class member variable?**

A static variable exists though the objects for the respective class are not created. Static member variables share a common memory across all the objects created for the respective class. A static member variable can be referred using the class name itself.

**18. Explain the static member function.**

A static member function can be invoked using the class name as it exists before class objects come into existence. It can access only static members of the class.

**19. Name the data type which can be used to store wide characters in C++.**

wchar\_t

**20. What are/is the operator/operators used to access the class members?**

Dot (.) and Arrow (->)

**21. Can we initialize a class/structure member variable as soon as the same is defined?**

No, defining a class/structure is just a type definition and will not allocate memory for the same.

**22. What is the data type to store the Boolean value?**

bool, is the new primitive data type introduced in C++ language.

**23. What is function overloading?**

Defining several functions with the same name with unique list of parameters is called as function overloading.

**24. What is operator overloading?**

Defining a new job for the existing operator w.r.t the class objects is called as operator overloading.

**25. Do we have a String primitive data type in C++?**

No, it's a class from STL (Standard template library).

**26. Name the default standard streams in C++.**

cin, cout, cerr and clog.

**27. Which access specifier/s can help to achieve data hiding in C++?**

Private & Protected.

**26. When a class member is defined outside the class, which operator can be used to associate the function definition to a particular class?**

Scope resolution operator (::)

**27. What is a destructor? Can it be overloaded?**

A destructor is the member function of the class which is having the same name as the class name and prefixed with tilde (~) symbol. It gets executed automatically w.r.t the object as soon as the object loses its scope. It cannot be overloaded and the only form is without the parameters.

**28. What is a constructor?**

A constructor is the member function of the class which is having the same as the class name and gets executed automatically as soon as the object for the respective class is created.

**29. What is a default constructor? Can we provide one for our class?**

Every class does have a constructor provided by the compiler if the programmer doesn't provides one and known as default constructor. A programmer provided constructor with no parameters is called as default constructor. In such case compiler doesn't provides the constructor.

**30. Which operator can be used in C++ to allocate dynamic memory?**

'new' is the operator can be used for the same.

**31. What is the purpose of 'delete' operator?**

'delete' operator is used to release the dynamic memory which was created using 'new' operator.

**32. Can I use malloc() function of C language to allocate dynamic memory in C++?**

Yes, as C is the subset of C++, we can all the functions of C in C++ too.

**33. Can I use 'delete' operator to release the memory which was allocated using malloc() function of C language?**

No, we need to use free() of C language for the same.

**34. What is a friend function?**

A function which is not a member of the class but still can access all the member of the class is called so. To make it happen we need to declare within the required class following the keyword 'friend'.

**35. What is a copy constructor?**

A copy constructor is the constructor which takes same class object reference as the parameter. It gets automatically invoked as soon as the object is initialized with another object of the same class at the time of its creation.

**36. Does C++ support exception handling? If so what are the keywords involved in achieving the same.**

C++ does support exception handling. Try, catch & throw are keyword used for the same.

**37. Explain the pointer – this.**

This, is the pointer variable of the compiler which always holds the current active object's address.

**38. What is the difference between the keywords struct and class in C++?**

By default the members of struct are public and by default the members of the class are private.

**39. What is the block scope variable in C++?**

A variable whose scope is applicable only within a block is said so. Also a variable in C++ can be declared anywhere within the block.

**40. What is the role of the file opening mode ios::trunk?**

If the file already exists, its content will be truncated before opening the file.

**41. What is the scope resolution operator?**

The scope resolution operator is used to

- Resolve the scope of global variables.
- To associate function definition to a class if the function is defined outside the class.

#### 42. What is a namespace?

A namespace is the logical division of the code which can be used to resolve the name conflict of the identifiers by placing them under different name space.

#### 43. What are command line arguments?

The arguments/parameters which are sent to the main() function while executing from the command line/console are called so. All the arguments sent are the strings only.

#### 44. What is a class template?

A template class is a generic class. The keyword template can be used to define a class template.

How can we catch all kind of exceptions in a single catch block?

The catch block with ellipses as follows

```
catch(...)  
  
{  
  
}
```

#### 45. What is keyword auto for?

By default every local variable of the function is automatic (auto). In the below function both the variables 'i' and 'j' are automatic variables.

```
void f()  
  
{  
  
int i;  
  
  
auto int j;  
  
}
```

**NOTE** – A global variable can't be an automatic variable.

#### 46. What is a static variable?

A static local variables retains its value between the function call and the default value is 0. The following function will print 1 2 3 if called thrice.

```
void f()  
  
{  
  
static int i;
```

```
++i;  
printf("%d ",i);  
}
```

If a global variable is static then its visibility is limited to the same source code.

#### **47. What is the purpose of extern storage specifier?**

Used to resolve the scope of global symbol

```
#include<iostream>  
  
using namespace std;  
main(){  
extern int i;  
  
cout<<i<<endl;  
}  
  
int i =20;
```

#### **48. What is the meaning of base address of the array?**

The starting address of the array is called as the base address of the array.

#### **49. When should we use the register storage specifier?**

If a variable is used most frequently then it should be declared using register storage specifier, then possibly the compiler gives CPU register for its storage to speed up the look up of the variable.

#### **50. Can a program be compiled without main() function?**

Yes, it can be but cannot be executed, as the execution requires main() function definition.

#### **51. Where an automatic variable is stored?**

Every local variable by default being an auto variable is stored in stack memory

#### **52. What is a container class?**

A class containing at least one member variable of another class type in it is called so.

#### **53. What is a token?**

A C++ program consists of various tokens and a token is either a keyword, an identifier, a constant, a string literal, or a symbol.

#### **54. What is a preprocessor?**

Preprocessor is a directive to the compiler to perform certain things before the actual compilation process begins.

#### **55. What are command line arguments?**

The arguments which we pass to the main() function while executing the program are called as command line arguments. The parameters are always strings held in the second argument (below in args) of the function which is array of character pointers. First argument represents the count of arguments (below in count) and updated automatically by operating system.

```
main(intcount,char*args[]){  
}
```

#### **56. What are the different ways of passing parameters to the functions? Which to use when?**

- **Call by value** – We send only values to the function as parameters. We choose this if we do not want the actual parameters to be modified with formal parameters but just used.
- **Call by address** – We send address of the actual parameters instead of values. We choose this if we do want the actual parameters to be modified with formal parameters.
- **Call by reference** – The actual parameters are received with the C++ new reference variables as formal parameters. We choose this if we do want the actual parameters to be modified with formal parameters.

#### **57. What is reminder for 5.0 % 2?**

Error, It is invalid that either of the operands for the modulus operator (%) is a real number.

#### **58. Which compiler switch to be used for compiling the programs using math library with g++ compiler?**

Option -lm to be used as > g++ -lm <file.cpp>

#### **59. Can we resize the allocated memory which was allocated using 'new' operator?**

No, there is no such provision available.

#### **60. Who designed C++ programming language?**

BjarneStroustrup.

**61. Which operator can be used to determine the size of a data type/class or variable/object?**

sizeof

**62. How can we refer to the global variable if the local and the global variable names are same?**

We can apply scope resolution operator (::) to the for the scope of global variable.

**63. What are valid operations on pointers?**

The only two permitted operations on pointers are

- Comparison ii) Addition/Substraction (excluding void pointers)

**64. What is recursion?**

Function calling itself is called as recursion.

**65. What is the first string in the argument vector w.r.t command line arguments?**

Program name.

**66. What is the maximum length of an identifier?**

Ideally it is 32 characters and also implementation dependent.

**67. What is the default function call method?**

By default the functions are called by value.

**68. What are available modes of inheritance to inherit one class from another?**

Public, private & protected

**69. What is the difference between delete and delete[]?**

Delete[] is used to release the array allocated memory which was allocated using new[] and delete is used to release one chunk of memory which was allocated using new.

**70. Does an abstract class in C++ need to hold all pure virtual functions?**

Not necessarily, a class having at least one pure virtual function is abstract class too.

**71. Is it legal to assign a base class object to a derived class pointer?**

No, it will be error as the compiler fails to do conversion.

**72. What happens if an exception is thrown outside a try block?**

The program shall quit abruptly.

**73. Are the exceptions and error same?**

No, exceptions can be handled whereas program cannot resolve errors.

**74. What is function overriding?**

Defining the functions within the base and derived class with the same signature and name where the base class's function is virtual.



**75. Which function is used to move the stream pointer for the purpose of reading data from stream?**

seekg()

**76. Which function is used to move the stream pointer for the purpose of writing data from stream?**

seekp()

**77. Are class functions taken into consideration as part of the object size?**

No, only the class member variables determine the size of the respective class object.

**78. Can we create an empty class? If so what would be the size of such object.**

We can create an empty class and the object size will be 1.

**79. What is 'std'?**

Default namespace defined by C++.

**80. What is the full form of STL?**

Standard template library

**81. What is 'cout'?**

cout is the object of ostream class. The stream 'cout' is by default connected to console output device.

**82. What is 'cin'?**

cin is the object of istream class. The stream 'cin' is by default connected to console input device.

**83. What is the use of the keyword 'using'?**

It is used to specify the namespace being used in.

**84. If a pointer declared for a class, which operator can be used to access its class members?**

Arrow (->) operator can be used for the same

**85. What is difference between including the header file with-in angular braces <> and double quotes “ “**

If a header file is included with in <> then the compiler searches for the particular header file only with in the built in include path. If a header file is included with in “ “, then the compiler searches for the particular header file first in the current working directory, if not found then in the built in include path

**86. S++ or S = S+1, which can be recommended to increment the value by 1 and why?**

S++, as it is single machine instruction (INC) internally.

**87. What is the difference between actual and formal parameters?**

The parameters sent to the function at calling end are called as actual parameters while at the receiving of the function definition called as formal parameters.

**88. What is the difference between variable declaration and variable definition?**

Declaration associates type to the variable whereas definition gives the value to the variable.

**89. Which key word is used to perform unconditional branching?**

goto.

**90. What is the purpose of #undef preprocessor?**

It will be used to undefine an existing macro definition.

**91. Can we nest multi line comments in a C++ code?**

No, we cannot.

**92. What is a virtual destructor?**

A virtual destructor ensures that the objects resources are released in the reverse order of the object being constructed w.r.t inherited object.

**93. What is the order of objects destroyed in the memory?**

The objects are destroyed in the reverse order of their creation.

**94. What is a friend class?**

A class member can gain accessibility over other class member by placing the class declaration prefixed with the keyword 'friend' in the destination class.

# **Java Viva Questions**

1. Define Class in Java.

In Java, a class is a template used to create objects and define the data type. It acts as a building block for Java language-oriented systems.

2. What Is the Difference Between Static and Dynamic Loading?

Static class loading involves the creation of objects and instances using new keywords, and dynamic class loading is done when the name of the class is not known at compile time.

3. What Is Multi-Threading?

Multi-threading is a programming concept used to run multiple tasks in a concurrent manner within a single program.

4. When and by whom was Java Developed?

Java was developed by James Gosling in Sun Microsystem in 1995.

5. What Do JDK, JRE, and JVM Stand for?

- JVM stands for Java Virtual Machine
- JRE stands for Java Runtime Environment
- JDK stands for Java Development Kit

6. Does Java Use Pointers?

No. Java doesn't use pointers. It has a tough security. Instead of pointers, references are used in Java as they are safer and more secure when compared to a pointer.

7. How Do You Connect to a Database in Java?

The steps to connect to a database in Java are:

- Registering the driver class
- Creating connection
- Creating statement
- Executing queries
- Closing connection

#### 8. What Are the Functions of JVM and JRE?

JVM provides a runtime environment for Java Byte Codes to be executed. JRE includes sets of files required by JVM during runtime.

#### 9. What Is the Difference Between Overloading and Overriding?

When there are two methods of the same name but different properties, it is overloading. Overriding occurs when there are two methods of the same name and properties, one is in the child class and the other is in the parent class. Check out this [post](#) for a more in-depth analysis.

#### 10. What Is the Default Size of the Load Factor in Hashing a Based Collection?

The default size is 0.75, and the default capacity is computed as:

Initial capacity \* Load factor

#### 11. What Is a Package?

A package is a collection of related classes and interfaces.

#### 12. What's the Base Class of all Exception Classes?

Java.lang.Throwable is the superclass of all exception classes, and all exception classes are derived from this base class.

#### 13. What Is the Difference Between **equals()** and **==** ?

Equals() method is used for checking the equality of two objects defined by business logic.

== or the equality operator is used to compare primitives and objects.

#### 14. State Two Differences Between an Inner Class and a Subclass

While Inner classes are in the same file, subclasses can be in another file. With that in mind, while subclasses have the methods of their parent class, inner classes get the methods they want.

#### 15. How Are Destructors Defined in Java?

Since Java has its own garbage collection, no destructors are required to be defined. Destruction of objects is automatically carried by the garbage collection mechanism.

#### 16. Define JSON.

JSON is an acronym for JavaScript Object Notation. It uses JavaScript syntax, and the format is text only.

#### 17. Name the Most Important Feature of Java

Java is a platform independent language.

### 18. What Is an Anonymous Class?

The class defined without a name in a single line of code using new keyword is known as an anonymous class.

### 19. What Is a JVM?

JVM is the Java Virtual Machine, which is a runtime environment for compiled Java class files.

### 20. Can a Dead Thread Be Started Again?

No, a thread that is in the dead state can't be started again.

### 21. Are Arrays of Primitive Data Types?

No. In Java, Arrays are objects.

### 22. What Are Constructors in Java?

In Java, the constructor is a block of code used to initialize an object.

### 23. What Are the Types of Constructors?

There are two types of constructors:

#### 1. Default constructor:

A constructor that has no parameter is known as default constructor. If we don't define a constructor in a class, the compiler creates **a default constructor (with no arguments)** for the class, as shown below:

```
// no-argument constructor
import java.io.*;
class Sprint
{
    int num;
    String name;
    // this would be invoked while object
    // of that class created.
    Sprint()
    {
        System.out.println("Constructor called");
    }
}
class Z
{
    public static void main (String[] args)
    {
        // this would invoke default constructor.
        Sprint zeal1 = new Sprint();
        // Default constructor provides the default
```

```
// values to the object like 0, null
System.out.println(zeal1.name);
System.out.println(zeal1.num);
    }
}
```

**2. Parameterized constructor:** A constructor that has known parameters is a **parameterized constructor**. If we want to initialize fields of the class with your own values, then use a parameterized constructor.

```
// Java Program to illustrate calling of
// parameterized constructor.
import
java.io.*;
class Sprint
{
// data members of the class.
String name;
int id;
// constructor would initialize data members
// with the values of passed arguments while
// object of that class created.
Sprint(String name, int id)
    {
this.name = name;
this.id = id;
    }
}
class Z
{
public static void main (String[] args)
    {
// this would invoke parameterized constructor.
Sprint zeal1 = new Sprint("adam", 1);
System.out.println("Sprint Name : "+zeal1.name+" and Sprint Id : "+zeal1.id);
    }
}
```

24. Explain Garbage Collection in Java.

In Java, when an object is no longer used or referenced, garbage collection is called and the object is destroyed automatically.

**25. What's the Difference Between Stack and Queue?**

The difference between a stack and a queue is that the stack is based on the Last in First out (LIFO) principle, and a queue is based on FIFO (First In, First Out) principle.

# **Assembly Language**

## **What is Assembly Language?**

An assembly language is a low-level programming language for microprocessors and other programmable devices. It is not just a single language, but rather a group of languages.

## **Describe Advantages of Assembly Language**

Having an understanding of assembly language makes one aware of –

- How programs interface with OS, processor, and BIOS;
- How data is represented in memory and other external devices;
- How the processor accesses and executes instruction;
- How instructions access and process data;
- How a program accesses external devices.

Other advantages of using assembly language are –

- It requires less memory and execution time;
- It allows hardware-specific complex jobs in an easier way;
- It is suitable for time-critical jobs;
- It is most suitable for writing interrupt service routines and other memory resident programs.

## **Describe Basic Features of PC Hardware**

The main internal hardware of a PC consists of processor, memory, and registers. Registers are processor components that hold data and address. To execute a program, the system copies it from the external device into the internal memory. The processor executes the program instructions.

## **Supporting number system by processors**

The processor supports the following data sizes –

- Word: a 2-byte data item
- Double word: a 4-byte (32 bit) data item
- Quad word: an 8-byte (64 bit) data item
- Paragraph: a 16-byte (128 bit) area
- Kilobyte: 1024 bytes
- Megabyte: 1,048,576 bytes

## What is Binary Number System?

Every number system uses positional notation, i.e., each position in which a digit is written has a different positional value. Each position is power of the base, which is 2 for binary number system, and these powers begin at 0 and increase by 1.

## Describe Addressing Data in Memory

The process through which the processor controls the execution of instructions is referred as the **fetch-decode-execute cycle** or the **execution cycle**. It consists of three continuous steps –

- Fetching the instruction from memory
- Decoding or identifying the instruction
- Executing the instruction

When the processor gets the numeric data from memory to register, it again reverses the bytes. There are two kinds of memory addresses –

- Absolute address - a direct reference of specific location.
- Segment address (or offset) - starting address of a memory segment with the offset value.

## Section of an Assembly Program

An assembly program can be divided into three sections –

- The **data** section,
- The **bss** section, and
- The **text** section.

## What are Memory Segments in Assembly?

A segmented memory model divides the system memory into groups of independent segments referenced by pointers located in the segment registers. Each segment is used to contain a specific type of data. One segment is used to contain instruction codes, another segment stores the data elements, and a third segment keeps the program stack.

In the light of the above discussion, we can specify various memory segments as –

- **Data segment** – It is represented by **.data** section and the **.bss**. The **.data** section is used to declare the memory region, where data elements are stored for the program. This section cannot be expanded after the data elements are declared, and it remains static throughout the program.

The **.bss** section is also a static memory section that contains buffers for data to be declared later in the program. This buffer memory is zero-filled.



- **Code segment** – It is represented by **.text** section. This defines an area in memory that stores the instruction codes. This is also a fixed area.
- **Stack** – this segment contains data values passed to functions and procedures within the program.

## What is processor and register?

Processor operations mostly involve processing data. This data can be stored in memory and accessed from thereon. However, reading data from and storing data into memory slows down the processor, as it involves complicated processes of sending the data request across the control bus and into the memory storage unit and getting the data through the same channel.

To speed up the processor operations, the processor includes some internal memory storage locations, called **registers**.

The registers store data elements for processing without having to access the memory. A limited number of registers are built into the processor chip.

## What are Processor Registers?

There are ten 32-bit and six 16-bit processor registers in IA-32 architecture. The registers are grouped into three categories –

- General registers,
- Control registers, and
- Segment registers.

The general registers are further divided into the following groups –

- Data registers,
- Pointer registers, and
- Index registers.

## What are Data Registers?

**AX is the primary accumulator**; it is used in input/output and most arithmetic instructions. For example, in multiplication operation, one operand is stored in EAX or AX or AL register according to the size of the operand.

**BX is known as the base register**, as it could be used in indexed addressing.

**CX is known as the count register**, as the ECX, CX registers store the loop count in iterative operations.

**DX is known as the data register**. It is also used in input/output operations. It is also used with AX register along with DX for multiply and divide operations involving large values.

## What are Pointer Registers?

The pointer registers are 32-bit EIP, ESP, and EBP registers and corresponding 16-bit right portions IP, SP, and BP. There are three categories of pointer registers –

- **Instruction Pointer (IP)** – The 16-bit IP register stores the offset address of the next instruction to be executed. IP in association with the CS register (as CS:IP) gives the complete address of the current instruction in the code segment.
- **Stack Pointer (SP)** – The 16-bit SP register provides the offset value within the program stack. SP in association with the SS register (SS:SP) refers to the current position of data or address within the program stack.
- **Base Pointer (BP)** – The 16-bit BP register mainly helps in referencing the parameter variables passed to a subroutine. The address in SS register is combined with the offset in BP to get the location of the parameter. BP can also be combined with DI and SI as base register for special addressing.

## What are Index Registers?

The 32-bit index registers, ESI and EDI, and their 16-bit rightmost portions, SI and DI, are used for indexed addressing and sometimes used in addition and subtraction. There are two sets of index pointers –

- **Source Index (SI)** – It is used as source index for string operations.
- **Destination Index (DI)** – It is used as destination index for string operations.

## What are Segment Registers?

Segments are specific areas defined in a program for containing data, code and stack. There are three main segments –

- **Code Segment** – It contains all the instructions to be executed. A 16-bit Code Segment register or CS register stores the starting address of the code segment.
- **Data Segment** – It contains data, constants and work areas. A 16-bit Data Segment register or DS register stores the starting address of the data segment.
- **Stack Segment** – It contains data and return addresses of procedures or subroutines. It is implemented as a 'stack' data structure. The Stack Segment register or SS register stores the starting address of the stack.

## Simply Describe Common Flags.

The common flag bits are:

- **Overflow Flag (OF)** – It indicates the overflow of a high-order bit (leftmost bit) of data after a signed arithmetic operation.

- **Direction Flag (DF)** – It determines left or right direction for moving or comparing string data. When the DF value is 0, the string operation takes left-to-right direction and when the value is set to 1, the string operation takes right-to-left direction.
- **Interrupt Flag (IF)** – It determines whether the external interrupts like keyboard entry, etc., are to be ignored or processed. It disables the external interrupt when the value is 0 and enables interrupts when set to 1.
- **Trap Flag (TF)** – It allows setting the operation of the processor in single-step mode. The DEBUG program we used sets the trap flag, so we could step through the execution one instruction at a time.
- **Sign Flag (SF)** – It shows the sign of the result of an arithmetic operation. This flag is set according to the sign of a data item following the arithmetic operation. The sign is indicated by the high-order of leftmost bit. A positive result clears the value of SF to 0 and negative result sets it to 1.
- **Zero Flag (ZF)** – It indicates the result of an arithmetic or comparison operation. A nonzero result clears the zero flag to 0, and a zero result sets it to 1.
- **Auxiliary Carry Flag (AF)** – It contains the carry from bit 3 to bit 4 following an arithmetic operation; used for specialized arithmetic. The AF is set when a 1-byte arithmetic operation causes a carry from bit 3 into bit 4.
- **Parity Flag (PF)** – It indicates the total number of 1-bits in the result obtained from an arithmetic operation. An even number of 1-bits clears the parity flag to 0 and an odd number of 1-bits sets the parity flag to 1.
- **Carry Flag (CF)** – It contains the carry of 0 or 1 from a high-order bit (leftmost) after an arithmetic operation. It also stores the contents of last bit of a *shift* or *rotate* operation.

## Conditional Execution in Assembly

Conditional execution in assembly language is accomplished by several looping and branching instructions. These instructions can change the flow of control in a program. Conditional execution is observed in two scenarios –

### Conditional Instructions

#### Unconditional jump

This is performed by the JMP instruction. Conditional execution often involves a transfer of control to the address of an instruction that does not follow the currently executing instruction. Transfer of control may be forward, to execute a new set of instructions or backward, to re-execute the same steps.

#### Conditional jump

This is performed by a set of jump instructions `j<condition>` depending upon the condition. The conditional instructions transfer the control by breaking the sequential flow and they do it by changing the offset value in IP.

## What is File Descriptor?

A **file descriptor** is a 16-bit integer assigned to a file as a file id. When a new file is created or an existing file is opened, the file descriptor is used for accessing the file.

File descriptor of the standard file streams - **stdin**, **stdout** and **stderr** are 0, 1 and 2, respectively.

## What is File Pointer?

A **file pointer** specifies the location for a subsequent read/write operation in the file in terms of bytes. Each file is considered as a sequence of bytes. Each open file is associated with a file pointer that specifies an offset in bytes, relative to the beginning of the file. When a file is opened, the file pointer is set to zero.

# Computer Organization and Architecture

## Architecture and Organization

- **Computer Architecture** refers to those attributes of a system visible to a programmer or that have a direct impact on the logical execution of a program.
- **Computer Organization** refers to the operational units and their interconnection that realize the architectural specifications
- Structure: The way in which the components are interrelated.
- Function: The operation of each individual component as part of the structure. The basic functions that a computer can perform
  - Data processing
  - Data storage
  - Data movement
  - Control above functions

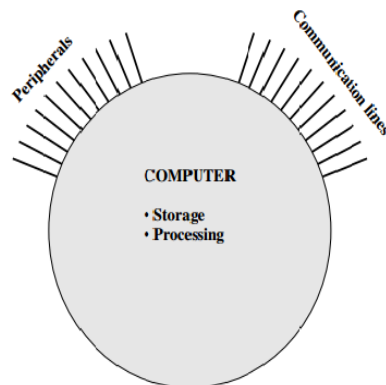


Figure 1.3 The Computer

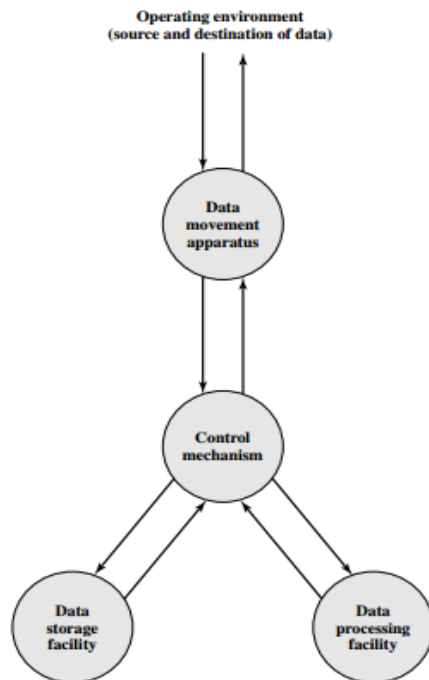


Figure 1.1 A Functional View of the Computer

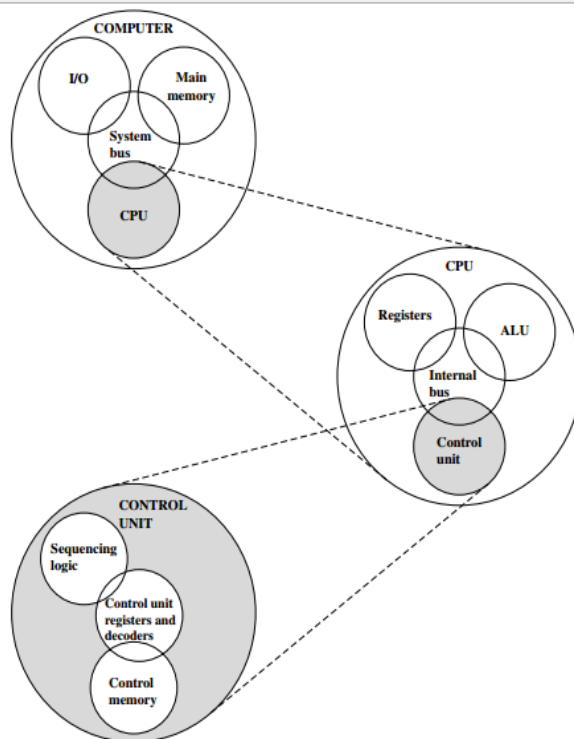


Figure 1.4 The Computer: Top-Level Structure

Main structural components of Computer:

**Central processing unit (CPU):** Controls the operation of the computer and performs its data processing functions; often simply referred to as processor.

- **Main memory:** Stores data.

- **I/O:** Moves data between the computer and its external environment.

- **System interconnection:** Some mechanism that provides for communication among CPU, main memory, and I/O. A common example of system interconnection is by means of a system bus, consisting of a number of conducting wires to which all the other components attach.

### Major structural components of CPU:

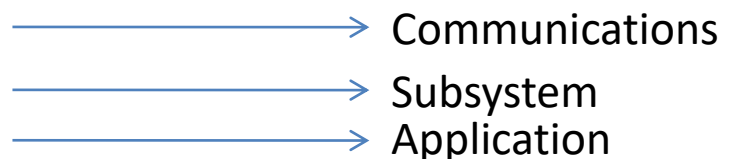
- **Control unit:** Controls the operation of the CPU and hence the computer
- **Arithmetic and logic unit (ALU):** Performs the computer's data processing functions
- **Registers:** Provides storage internal to the CPU
- **CPU interconnection:** Some mechanism that provides for communication among the control unit, ALU, and registers

### Hierarchic Principles

Architecture



Control



## Mode of data representation

- The only two values that a digital computing device has available are the values 0 and 1.
- The electronic signals that represent a 0 or a 1 are called bits.
- The bits may be organized into pure binary sequences where each consecutive bit represents a higher power of the base 2 numbering system.
- Thus the string of bits 101001 represents the value 41 in the decimal system in the following way:

$2^{**5}$	$2^{**4}$	$2^{**3}$	$2^{**2}$	$2^{**1}$	$2^{**0}$
(32)	(16)	(8)	(4)	(2)	(1)
1	0	1	0	0	1

## Number of Address Bits

- The number of bits in an instruction determines the range of direct addressability.
- Direct addressability is the number of addresses available to a program.
- The range of direct addressability determines the maximum size of a program segment. It bounds the size of a program and limits the amount of memory a program can profitably use.

## Number of addresses

- The decision about how many addresses to use is basic to an architecture. The architect may decide that it is not necessary in all circumstances to specify two operands and a result address.
- Sometimes a result address may be implied by an operand address. We may always place the result in the memory address of the 1<sup>st</sup> operand. It reduces the number of bits that must be used for addressing in certain operations.

## Definition of specialized units

- The performance of a processor is determined by the number of instructions that must be executed to accomplish a computation times the length of time to perform each instruction. The length of time is determined by the basic speed of the technology times the number of machine



cycles required to execute the instruction. A cycle is a fixed hardware period in which a set of logical operations are performed. We speak of an instruction as requiring N cycles of T units of time. Thus performance time for a program equals

- $PT = \text{No. Instructions} * \text{No. Cycles} * \text{Cycle time}$

## **Memory Organization**

- Memory organization means how much memory is appropriate for different levels of processing power. Faster processor needs more memory because they use the data and instruction in memory at a faster rate and consequently need more instructions and /or data from I/O devices at a faster rate.
- The amount of memory required is a function of the efficiency of the architecture and the speed of the I/O system.
- The designer must make some decisions about the basic organization of physical memories and how they will interconnect to processing units. A memory can be organized so that all of the address space is held in a single physical unit.
- Connecting memories to processors involves two additional fundamental decisions. The designer must determine the topology of the interconnect network. As with any network, star, mesh or bus structures may be used. The designer must also determine the width of the data paths from the memory to the processor, just as he or she determines the width of data paths within a processor.

## **I/O Organization**

- The designer must determine how I/O devices are to be attached to the system and the split of functions across different components of the I/O subsystem.
- Small systems have relatively simple I/O interconnection designs. Large systems may have tremendously elaborate I/O structures containing processors with large amounts of memory within the I/O structure itself.
- The designer must determine how many different types of devices to place along the I/O pathway and how functions should be split among them to get the best possible performance at various cost levels.

## **Buffering**

- Memory may be thought of as a buffer between the processor and I/O. Processor speeds are dramatically faster than I/O speeds, and the memory acts as a buffer so that references to instructions and data may be satisfied by memory accesses with a reduced need to do I/O.
- The memory speeds are characteristically slower than processor speeds. Thus a memory reference may require two, three, four or even more processor cycles.
- A designer must determine how to address the speed imbalance problem between processors and memory as well as between processors and I/O.

## Functional Implementation

- A machine may be completely designed in logic units that comprise the basic functions of switching. Patterns of such basic units representing the Boolean functions (AND, OR, NAND, NOR) can completely define the data and control flows of a system.
- The “Target Architecture” the architecture that is the goal of the compiler, is completely designed in basic logic and built with specialized units.

## Computer Components: Top-Level View:

A memory module consists of a set of locations, defined by sequentially numbered addresses. Each location contains a binary number that can be interpreted as either an instruction or data. An I/O module transfers data from external devices to CPU and memory, and vice versa. It contains internal buffers for temporarily holding these data until they can be sent on.

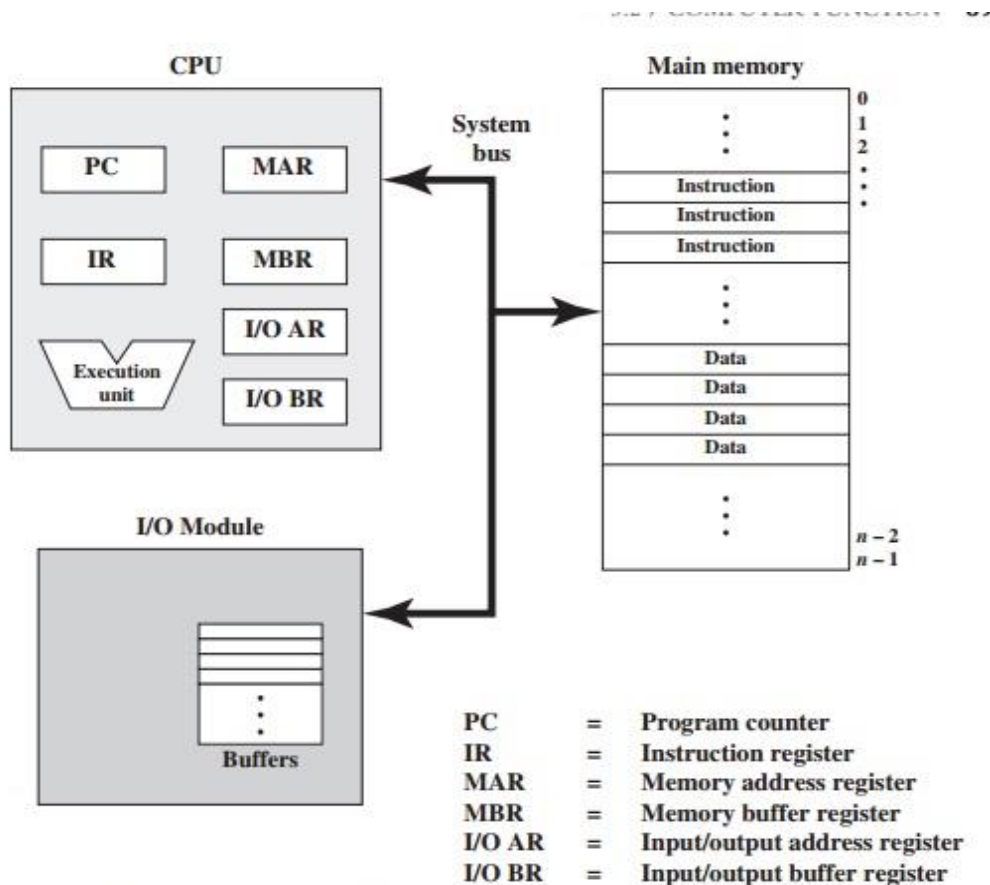


Figure 3.2 Computer Components:Top-Level View

Instruction Fetch and Execute:

- The processing required for a single instruction is called an instruction cycle. The two steps are referred to as the fetch cycle and the execute cycle.

- At the beginning of each instruction cycle, the processor fetches an instruction from memory. In a typical processor, a register called the program counter (PC) holds the address of the instruction to be fetched next.
- The fetched instruction is loaded into a register in the processor known as the instruction register (IR).

### Basic Instruction cycle diagram

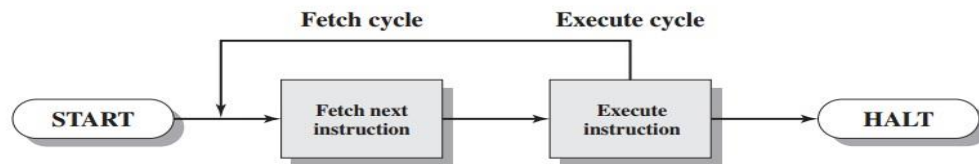


Figure 3.3 Basic Instruction Cycle

### Bus Structure

- A system bus consists typically about 50 to hundreds of separate lines. Each line is assigned a particular meaning or function.
- 3 functional groups of buses: data, address, and control lines.
- The data lines provide a path for moving data among system modules. These lines, collectively, are called the data bus. The data bus may consist of 32, 64, 128, or even more separate lines, the number of lines being referred to as the width of the data bus.
- The width of the data bus is a key factor in determining overall system performance. For example, if the data bus is 32 bits wide and each instruction is 64 bits long, then the processor must access the memory module twice during each instruction cycle.

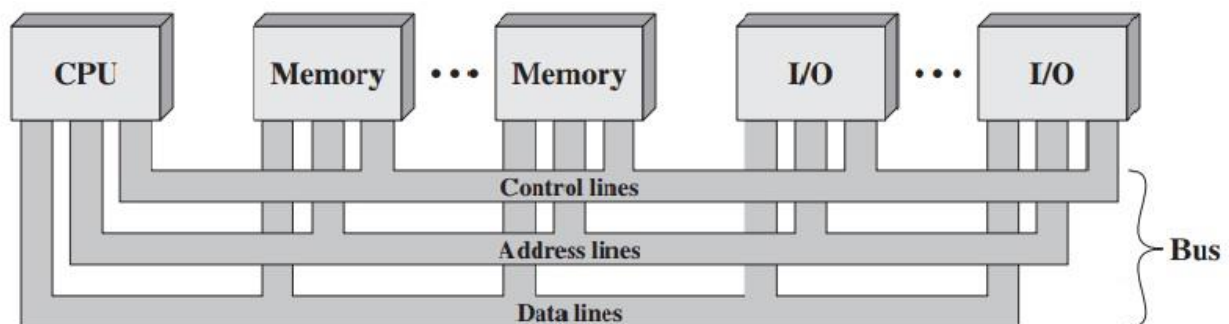


Figure 3.16 Bus Interconnection Scheme

### PCI

The peripheral component interconnect (PCI) is a popular high-bandwidth, processor-independent bus that can function as a peripheral bus.

PCI delivers better system performance for high-speed I/O subsystems

PCI is specifically designed to meet economically the I/O requirements of modern systems; it requires very few chips to implement and supports other buses attached to the PCI bus.

## **RAM**

- ▶ Random-access memory(RAM)
- ▶ It is possible both to read data from the memory and to write new data into the memory easily and rapidly.
- ▶ RAM is volatile, it must be provided with a constant power supply. If the power is interrupted, then the data are lost. Thus, RAM can be used only as temporary storage.
- ▶ The two traditional forms of RAM used in computers are DRAM and SRAM.

## **DYNAMIC RAM**

- ▶ A dynamic RAM (DRAM) is made with cells that store data as charge on capacitors. The presence or absence of charge in a capacitor is interpreted as a binary 1 or 0. Because capacitors have a natural tendency to discharge, dynamic RAMs require periodic charge refreshing to maintain data storage. The term dynamic refers to this tendency of the stored charge to leak away, even with power continuously applied.
- ▶ Although the DRAM cell is used to store a single bit (0 or 1), it is essentially an analog device. The capacitor can store any charge value within a range; a threshold value determines whether the charge is interpreted as 1 or 0.

## **STATIC RAM**

- ▶ Static RAM (SRAM) is a digital device that uses the same logic elements used in the processor. In a SRAM, binary values are stored using traditional flip-flop logic-gate configurations. A static RAM will hold its data as long as power is supplied to it.

## **SRAM VERSUS DRAM**

- ▶ Both static and dynamic RAMs are volatile; that is, power must be continuously supplied to the memory to preserve the bit values.
- ▶ A dynamic memory cell is simpler and smaller than a static memory cell. Thus, a DRAM is more dense (smaller cells more cells per unit area) and less expensive than a corresponding SRAM.
- ▶ On the other hand, a DRAM requires the supporting refresh circuitry. For larger memories, the fixed cost of the refresh circuitry is more. Thus, DRAMs tend to be favored for large memory requirements.
- ▶ A final point is that SRAMs are generally somewhat faster than DRAMs. Because of these relative characteristics, SRAM is used for cache memory (both on and off chip), and DRAM is used for main memory.

## **ROM**

- ▶ Read-only memory(ROM) contains a permanent pattern of data that cannot be changed. A ROM is nonvolatile; that is, no power source is required to maintain the bit values in memory. While it is possible to read a ROM, it is not possible to write new data into it.
- ▶ Important applications of ROMs:
  - ▶ microprogramming
  - ▶ Library subroutines for frequently wanted functions
  - ▶ System programs
  - ▶ Function tables

## **PROM**

- ▶ Programmable ROM(PROM) is nonvolatile and may be written into only once. For the PROM, the writing process is performed electrically and may be performed by a supplier or customer at a time later than the original chip fabrication. Special equipment is required for the writing or “programming” process. PROMs provide flexibility and convenience. The ROM remains attractive for high-volume production runs.
- ▶ Also called read-mostly memory
- ▶ There are three common forms of read-mostly memory: EPROM, EEPROM, and flash memory.

## **EPROM**

- ▶ Erasable programmable read-only memory(EPROM) is read and written electrically, as with PROM. However, before a write operation, all the storage cells must be erased to the same initial state by exposure of the packaged chip to ultraviolet radiation. Erasure is performed by shining an intense ultraviolet light through a window that is designed into the memory chip. This erasure process can be performed repeatedly. The EPROM is more expensive than PROM, but it has the advantage of the multiple update capability.

## **EEPROM**

- ▶ Electrically erasable programmable read-only memory(EEPROM). This is a read-mostly memory that can be written into at any time without erasing prior contents; only the byte or bytes addressed are updated. The write operation takes considerably longer than the read operation, on the order of several hundred microseconds per byte. The EEPROM combines the advantage of non-volatility with the flexibility of being updatable in place, using ordinary bus control, address, and data lines. EEPROM is more expensive than EPROM and also is less dense, supporting fewer bits per chip.

## **FLASH memory**

- ▶ Flash memory is intermediate between EPROM and EEPROM in both cost and functionality. Like EEPROM, flash memory uses an electrical erasing technology. An entire flash memory can be erased in one or a few seconds, which is much faster than EPROM. In addition, it is possible to erase just blocks of memory rather than an entire chip. Flash memory gets its name because the microchip is organized so that a section of memory cells are erased in a single action or “flash.” Flash memory uses only one transistor per bit, and so achieves the high density of EPROM.

## **Advantages of using a glass substrate for a magnetic disk:**

The glass substrate has a number of benefits, including the following

- Improvement in the uniformity of the magnetic film surface to increase disk Reliability
- A significant reduction in overall surface defects to help reduce read-write errors
- Ability to support lower fly heights (described subsequently)
- Better stiffness to reduce disk dynamics
- Greater ability to withstand shock and damage.

## **RAID (Redundant Array of Independent Disks)**

3 common characteristics:

- ▶ RAID is a set of physical disk drives viewed by the operating system as a single logical drive.
- ▶ Data are distributed across the physical drives of an array in a scheme known as striping, described subsequently.
- ▶ Redundant disk capacity is used to store parity information, which guarantees data recoverability in case of a disk failure.
- ▶ The details of the second and third characteristics differ for the different RAID levels. RAID 0 and RAID 1 do not support the third characteristic.

## **Compact Disk**

- ▶ A compact disc [sometimes spelled disk] (CD) is a small, portable, round medium made of molded polymer (close in size to the floppy disk) for electronically recording, storing, and playing back audio, video, text, and other information in digital form.

## DVD - digital versatile disc

- ▶ *Digital versatile disc* or *digital video disc*, a type of optical disk technology similar to the CD-ROM. A DVD holds a minimum of 4.7GB of data, enough for a full-length movie. DVDs are commonly used as a medium for digital representation of movies and other multimedia presentations that combine sound with graphics.
- ▶ The DVD specification supports disks with capacities of from 4.7GB to 17GB and access rates of 600KBps to 1.3MBps. One of the best features of DVD drives is that they are backward-compatible with CD-ROMs, meaning they can play old CD-ROMs, CD-I disks, and video CDs, as well as new DVD-ROMs. Newer DVD players can also read CD-R disks.

## Logic Gates

- AND
- OR
- NOT
- NAND
- NOR
- XOR
- EQU

## DE Morgan's Law

Allows a digital designer to convert an AND function to an equivalent OR function and vice versa. By using DE Morgan's Law to generate equivalent functions, a digital designer may be able to simplify complex functions.

- $(ab)' = a' + b'$
- $(a+b)' = a'b'$

## PROCESSOR ORGANIZATION

The requirements placed on the processor:

- Fetch instruction: The processor reads an instruction from memory (register, cache, main memory).
- Interpret instruction: The instruction is decoded to determine what action is required.
- Fetch data: The execution of an instruction may require reading data from memory or an I/O module.

- Process data: The execution of an instruction may require performing some arithmetic or logical operation on data.
- Write data: The results of an execution may require writing data to memory or an I/O module.

## INSTRUCTION CYCLE

In Section 3.2, we described the processor's instruction cycle (Figure 3.9). To recall, an instruction cycle includes the following stages:

- Fetch: Read the next instruction from memory into the processor.
- Execute: Interpret the opcode and perform the indicated operation.
- Interrupt: If interrupts are enabled and an interrupt has occurred, save the current process state and service the interrupt.

## Logic Design and Switching Circuit

### What is analog and what is digital system?

Analog: Analog system is an electronic system where the variable or signal are continuously. The term Analog describes the proportional relationship between a signal and a voltage or current that represents the signal.

Digital System: A digital signal has two or more distinguishable waveforms.

### Make differences between analog and digital signal

Difference between Digital and Analog signal:

Digital	Analog
---------	--------



Digital signals are discrete time generated by digital modulation.	Analog signals are continuous signal which represents physical measurements.
Waves denoted by square waves.	Waves denoted by sine wave.
Use discrete discontinuous values to represent information.	Use continuous range of values to represent information.
Samples analog waveforms into a limited set of numbers and records them.	Analog technology records waveforms as they are.
Can be noise-immune without deterioration during transmission and write/read cycle.	Subjected to deterioration by noise during transmission and write/read cycle.
Digital hardware is flexible in implementation.	Analog hardware is not flexible.
There is no guarantee that digital signal processing can be done in real time and consumes more bandwidth to carry out the same information.	Analog signal processing can be done in real time and consumes less bandwidth.
Digital instrument draws only negligible power.	Analog instrument draws large power.
Digital instruments are free from observational errors like parallax and approximation errors.	Analog instruments usually have a scale which is cramped at lower end and give considerable observational errors.

## **Describe Advantages and Disadvantages of Digital and Analog Techniques.**

Advantages of Digital Technique:

- I. Easier to design.
- II. Information storage is easy.
- III. Accuracy and precision is greater than other.
- IV. Less affected by noise. V. Fabricated by IC chips.
- VI. Operation can be programmed
- VII. Digital circuits are less affected by noise. As long as the noise is not large enough to prevent us from distinguishing a HIGH from a LOW.

Disadvantages or limitation of Digital Technique:

There is really only one major drawback when using digital techniques:

The real world is mainly analog.

- I. Convert the real-world analog inputs to digital form. II. Process (operate on) the digital information.
- III. Convert the digital outputs back to real-world analog form.

Advantages of Analog:

- I. No quantization errors, requires less bandwidth, low cost and can be easily constructed because of less pre-processing requirements.
- II. Produces a more 'faithful' reproduction of the physical quantity.
- III. Usually simple

Disadvantages of Analog:

- I. The effects of random noise can make signal loss and distortion impossible to recover.

## Classification of IC

IC: An integrated circuit is a small silicon semiconductor crystal called a chip. Example: Transistor, Diodes, Register, Capacitors and so on.

- I. SSI: Small scale integrated. Used for basic gates.
- II. MSI: Medium scale integrated. Used for 10 to 100 gates.
- III. LSI: Large scale integrated. Used for 100 to 1000 gates.
- IV. VLSI: Very large scale integrated. Used for above 1000 gates.

## What is Gate?

**Gate:** A logic gate is an elementary building block of a digital circuit.

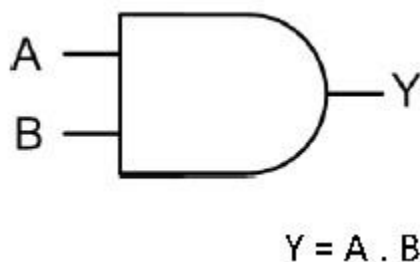
**Types of Gate:** Mainly there are two types of logic gate. Those are:

- I. Basic Logic Gate
- II. Universal Logic Gate

## Describe Basic gates

**There are three basic gates:**

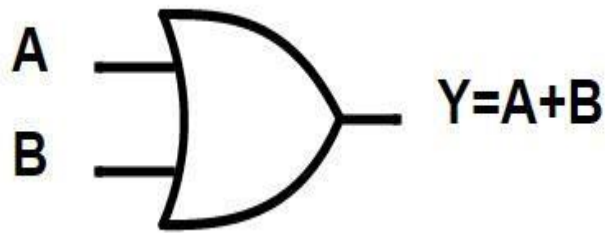
**AND Gate:** A Boolean operator that gives the value one if and only if all the operands are one, and otherwise has a value of zero.



A	B	Y=AB
0	0	0
0	1	0
1	0	0
1	1	1

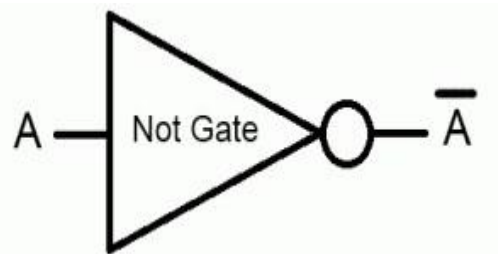
**OR Gate:** A Boolean operator that gives the value one if at least one operand (or input) has a value of one, and otherwise has a value of zero.

A	B	Y=A+B
0	0	0



0	1	1
1	0	1
1	1	1

NOT Gate: A Boolean operator with only one variable that has the value one when the variable is zero and vice versa.



A	$A = \bar{\square}$
0	1
0	1
1	0
1	0

## Describe Universal Gates

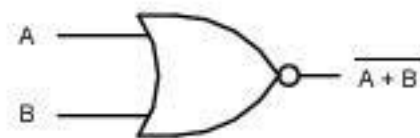
### Universal gates:

NAND Gate: A Boolean operator that gives the value zero if and only if all the operands have a value of one, and otherwise has a value of one (equivalent to NOT AND).



A	B	$Y = \bar{\square} \bar{\square}$
0	0	1
0	1	1
1	0	1
1	1	0

NOR Gate: The NOR gate is a digital logic gate that implements logical NOR - it behaves according to the truth table to the right. A HIGH output (1) results if both the inputs to the gate are LOW (0); if one or both input is HIGH (1), a LOW output (0) results. NOR is the result of the negation of the OR operator.



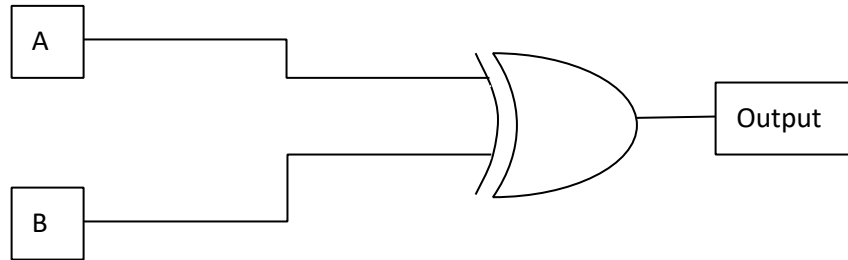
A	B	$Y = \bar{\square} + \bar{\square}$
0	0	1
0	1	0
1	0	0
1	1	0

## Describe Exclusive Gate

### Exclusive Gate:

#### Exclusive OR (XOR):

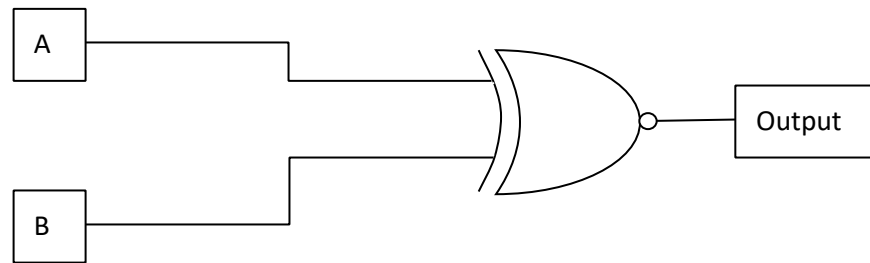
The XOR gate (sometimes EOR gate, or EXOR gate and pronounced as Exclusive OR gate) is a digital logic gate that gives a true (1/HIGH) output when the number of true inputs is odd. An XOR gate implements an exclusive or; that is, a true output results if one, and only one, of the inputs to the gate is true.



INPUT		OUTPUT
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

#### Exclusive NOR (XNOR):

The XNOR gate (sometimes, EXNOR, ENOR, and, rarely, NXOR, XAND) is a digital logic gate whose function is the logical complement of the exclusive OR (XOR) gate. The two-input version implements logical equality, behaving according to the truth table to the right. A high output (1) results if both of the inputs to the gate are the same. If one but not both inputs are high (1), a low output (0) results.



INPUT		OUTPUT
A	B	A XNOR B
0	0	1
0	1	0
1	0	0
1	1	1

## What is K-Map? Advantages and Disadvantages of K-Map.

K-Map: A diagram consisting of a rectangular array of squares, each representing a different combination of the variables of a Boolean function.

Advantages of K-Map:

- Time consuming is less.
- Space not much occupied.
- This method is very easy to follow.
- This is a systematic process. It's always leads to a single minimal solution.

Disadvantages of K-Map:

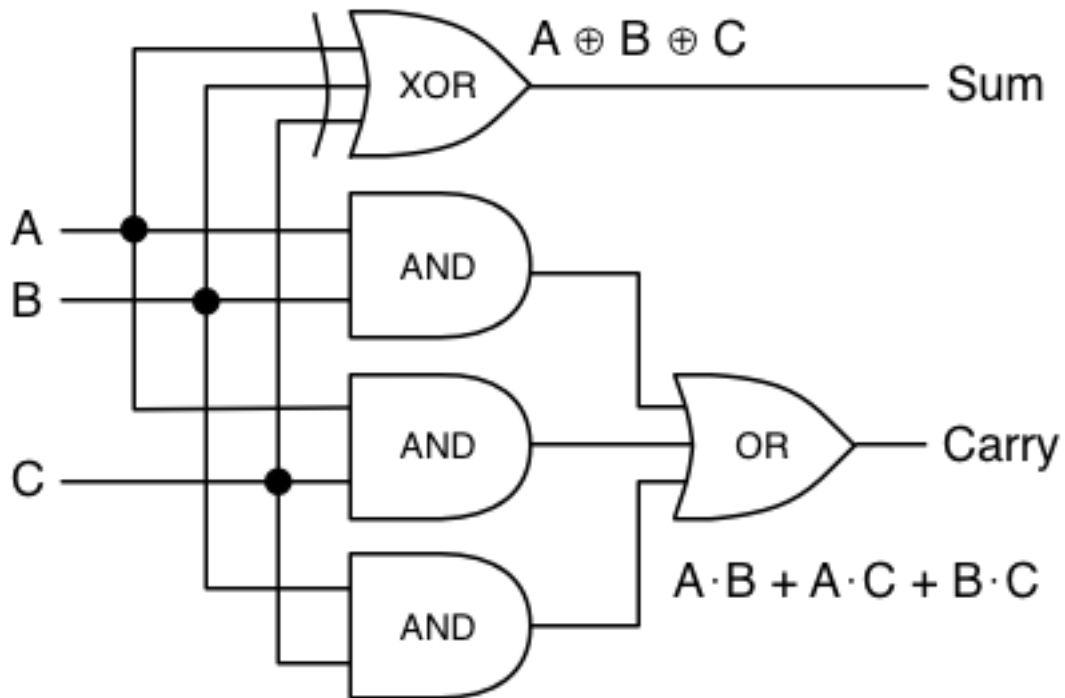
- The K map does not necessarily "fail" for higher dimensions.
- It is so difficult to visualize for more than five variables.
- A 4 variable K-map is 2 dimensional and easy to visualize. A 5 variable is three dimensional, but it is still manageable from a visualization standpoint, because the 2
- States of the 5th variable only require visually moving from one plane to the next, without moving in the x or y directions of either plane.
- Just getting equations correct with more than 5 variables is difficult enough using the K map, much less considering an optimum set of terms.

**What is hamming code? Write down the advantages of hamming code.**

It is an error detecting and correction code. That is used sign bit error and correction.

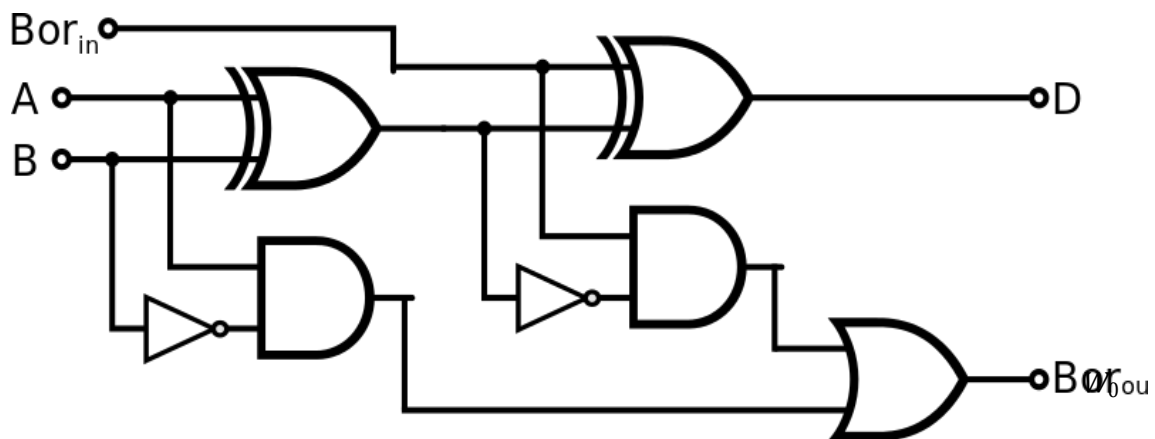
### Full Adder

Full adder provides addition of 3 bits logic circuit of full adder.



### Full Subtraction:

Full subtraction of two bit and takes borrow from previous position



**What is half adder? Draw the logic circuit of half adder with truth table.**

The half adder is an example of a simple, functional digital circuit built from two logic gates. The half adder adds two one-bit binary numbers (AB). The output is the sum of the two bits (S) and the carry (C).

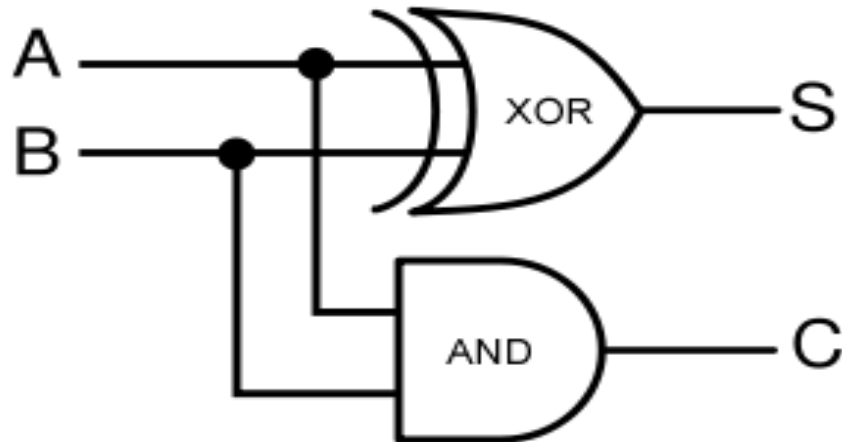


Fig: Logic Circuit

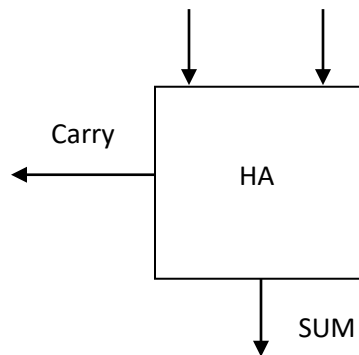


Fig: Block Diagram

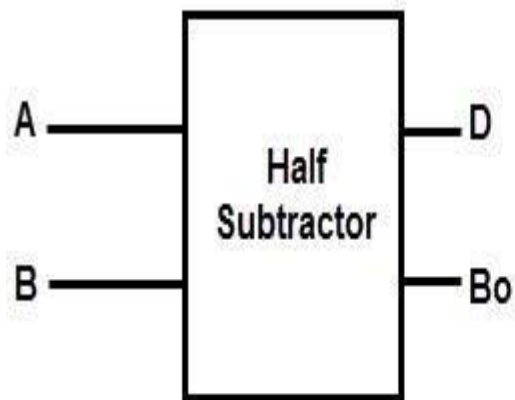
Input		Output	
<i>A</i>	<i>B</i>	<i>C</i>	<i>S</i>
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Truth table

**What is half subtractor? Describe half adder and half subtractor.**

The half subtractor is a combinational circuit which is used to perform subtraction of two bits.

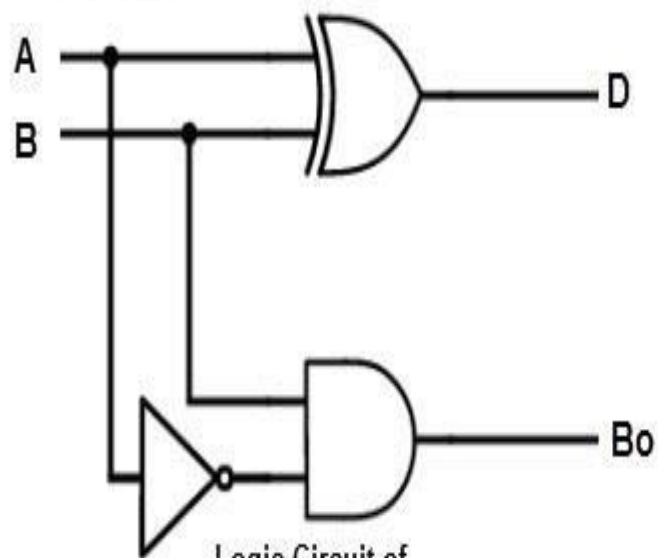




Block Diagram

A	B	D	B <sub>0</sub>
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Truth Table



Logic Circuit of  
Half Subtractor

# Fundamental of Electronics and Digital Systems

## What is Electronics?

The branch of engineering which deals with current conduction through a vacuum or gas or semiconductor is known as electronics.

## Importance of Electronics

- **Rectification: Conversion of AC into DC.**
- **Amplification: Process of raising the strength of a weak signal.**
- **Control: Automatic control of devices such as speed of motor, voltage across refrigerator etc.**
- **Conversion of light into electricity: The conversion of light into electricity is known as photo-electricity.**
- **Conversion of electricity into light: Electronic devices can convert electricity into light.**

## Conductor, Insulator and Semiconductor

### Conductors

- Have low resistance allowing electrical current flow
- A material is usually a metal and a conductor if the number of valence electrons is less than 4

### Insulators

- Have very high resistance suppressing electrical current flow
- A material is usually a non-metal and an insulator if the number of valence electrons is more than 4

### Semiconductors

- Has both metal and non-metal properties
- A material is usually a semiconductor if the number of valence electrons is 4

## What is Semiconductor?

A semiconductor is a substance which has resistivity in between conductors and insulators.

- It has negative temperature co-efficient of resistance which means the resistance of a semiconductor decreases with the increase in temperature and vice-versa.
- When a suitable metallic impurity is added to a semiconductor, its current conducting property change appreciably.

## **Intrinsic and Extrinsic Semiconductor**

### **Intrinsic semiconductor**

- An intrinsic semiconductor is a pure semiconductor which does not have any doping agent (an impurity added to crystal lattice).
- In an intrinsic semiconductors, the number of charge carrier depends upon property of the material not on impurity present.
- Has little current conduction capability at room temperature.

### **Extrinsic Semiconductors**

- An extrinsic semiconductor is a semiconductor whose electrical conductivity can be increased by adding the trace amounts of other elements such as impurities in the material.
- There are two types of impurities added to Ge and Si Crystal.
  - Pentavalent material - it is made up of atoms which have five valence electrons.
  - Trivalent material – it is materials has three valence electrons.

## **What is Doping?**

### **Doping**

- To make the semiconductor conduct electricity, other atoms called impurities must be added.
- “Impurities” are different elements.
- This process is called doping.

### **Depletion Layer**

- The free electrons of n region begin to diffuse across the junction into the p region where they combine with the holes near the junction.
- The p region also loses holes as the electrons and holes combine.
- The result is that there are layers of positive charge (pentavalent ions) and negative charge (trivalent ions).
- These two layers form depletion layer or depletion region
- This layer is free of charge carriers

### **Barrier Potential**

The potential difference across the depletion layer is called barrier potential

### **Use of semiconductor in computers**

- Computer memory is made up of semiconductor materials because it provides reliability
- CPU contains number of integrated circuits, which are made up of semiconductor materials.

## **Breakdown voltage and Knee voltage**

### Breakdown Voltage

- The minimum reverse voltage at which p-n junction breaks down with sudden rise in reverse current

### Knee voltage

- The forward voltage at which the current through the junction starts to increase rapidly

## **Types of Diodes**

### PN Junction Diode

These are standard types of diodes. They are formed by combining a P-type and an N-type semiconductor. These diodes are used in radio frequency applications.

### Zener Diode

Zener diodes have a low pre-determined Reverse Breakdown Voltage called Zener Voltage ( $V_{zk}$ ).

### Light Emitting Diode

The LED diodes are used in television sets to indicate whether the television is on or is on a standby mode.

### Photodiode

Photodiodes are used for detecting light in optical fibers and other light-sensitive applications.

### Schottky Diodes

It is used in radio frequency applications.

### PIN Diodes

Used in power rectifiers and RF protection circuits.

## **What is Transistor**

- Transistor is a three terminal electronic device
- Transistor Amplifies the weak input signal
- Transistor is used as switch and amplifier with various electronic devices.
- Transistors consume less power hence they have great efficiency.

## **Types of Transistor**

Bipolar Junction Transistors (BJT) – Three layer sandwich of doped semiconductor materials.

Field Effect Transistor (FET) – Two layers of semiconductor material.

- Junction Field Effect Transistor (JFET)
- Metal Oxide Semiconductor Field Effect Transistors (MOSFET)
- Phototransistors – Light detectors that possess internal gain

## **Layers of Transistor**

A transistor consists of three layers of semiconductors:

- Base
- Emitter
- Collector

## **Use of Transistor in Computer**

- Transistor is used in memory chips as a switch
- Number of transistors in computer helps to store large amount of binary numbers to represent ordinary numbers and alphabets
- Transistors used in microprocessors and printer logic board
- Switching transistor is used in SMPS
- Thin film transistor is used in monitor.

## **Transformer and Bridge Rectifier**

### **Transformer**

- Used to transfer alternating signal from one circuit to another
- The principle on which transformer action is based on is called as electromagnetic mutual inductance.
- Can be used to step-up or step-down the signals

### **Bridge Rectifier**

- Converts the transformer secondary AC voltage into pulsating DC voltage

- Consists of four diodes arranged so that current can flow in only one direction through the load
- To get a smooth and constant supply voltage, the rectifier circuit is followed immediately by a filter

### **What is Filter?**

Used to convert the pulsating DC provided by the rectifier into the steady DC voltage required for electronic circuits.

### **What is Regulator**

Maintains the output voltage of an ordinary power supply constant irrespective of load variations or changes in input AC voltage

### **Types of Digital Integrated Circuits (ICs)**

The following are the different types of digital integrated circuits:

- SSI (Small-Scale Integration)
- MSI (Medium-Scale Integration)
- LSI (Large-Scale Integration)
- VLSI (Very Large-Scale Integration)

### **Basic Logic Gates**

- NOT
- AND
- OR

### **Applications of Logic Gates**

The following are some of the applications of Logic gates:

- Build complex systems that can be used to different fields such as
  - Genetic engineering,
  - Nanotechnology,
  - Industrial Fermentation,
  - Metabolic engineering and
  - Medicine
- Construct multiplexers, adders and multipliers.
- Perform several parallel logical operations
- Used for a simple house alarm or fire alarm or in the circuit of automated machine manufacturing industry

### **What is Adder?**

- A logic circuit that performs the function of binary addition is called **electronic adder** or **adder**.
- The adder circuit consists of properly connected logic gates.

### **What is Multiplexer?**

Multiplex means many into one. A multiplexer is a circuit used to select and route any one of several input signals to a single output.

### **Applications of Multiplexer**

Multiplexers are used in various fields where multiple data needs to be transmitted using a single line

Following are some of the applications of multiplexers:

- Communications system
- Telephone network
- Computer memory
- Transmission from the computer system of a satellite

### **What is DE multiplexer?**

DE multiplex means one into many. A DE multiplexer is a circuit with one input and many outputs.

### **Applications of De Multiplexer**

DE multiplexer is used to connect a single source to multiple destinations

Following are some of the applications of multiplexers:

- Communications systems
- ALU (Arithmetic Logic Unit) circuits
- Serial to parallel converter

### **What is Encoder?**

An encoder is a circuit with multiple inputs which generates a unique address at its output that is, it converts an active input signal to a coded output signal

If only one input is active at a time, then the encoding is straightforward and can be accomplished with gates.

### **Applications of Encoder.**

An encoder is a circuit with multiple inputs which generates a unique address at its output that is, it converts an active input signal to a coded output signal

If only one input is active at a time, then the encoding is straightforward and can be accomplished with gates.

### **What is Decoder?**

A decoder is a circuit with no data input, but accepts only the control inputs and generates many outputs

The basic function of a decoder is conversion of coded data into original data

### **Applications of Decoder**

A decoder functions just the reverse of an encoder. Generally, decoders are used to convert the data coded by encoders into its original form.

This process of retrieving original data is used for various applications, which are as follows:

- Memory address decoding
- Seven segment display

## **Microprocessor**

### **What is microprocessor ?**

Ans: Microprocessor is a program-controlled device, which fetches the instructions from memory, decodes and executes the instructions. Most Micro Processor are single- chip devices

### **What are the flags in 8086 ?**

Ans: In 8086 Carry flag, Parity flag, Auxiliary carry flag, Zero flag, Overflow flag, Trace flag, Interrupt flag, Direction flag, and Sign flag

### **What are the variuos Registers in 8085 ?**

Ans: Accumulator register, Temporary register, Instruction register, Stack Pointer, Program Counter are the various registers in 8085.



**What are the various flags in 8085 ?**

Ans: Sign flag, Zero flag, Auxiliary flag, Parity flag, Carry flag.

**What is Stack pointer ?**

Ans: Stack pointer is a special purpose 16-bit register in the Microprocessor, which holds the address of the top of the stack.

**What is Program Counter ?**

Ans: Program counter holds the address of either the first byte of the next instruction to be fetched for execution or the address of the next byte of a multi byte instruction, which has not been completely fetched.

In both the cases it gets incremented automatically one by one as the instruction bytes get fetched. Also Program register keeps the address of the next instruction.

**What are the Hardware interrupts ?**

Ans:

TRAP, RST7.5, RST6.5, RST5.5, INTR

**What are the software interrupts ?**

Ans:

RST0, RST1, RST2, RST3, RST4, RST5, RST6, RST7.

## **Compiler**

**What is Compiler ?**

Ans: A compiler is a program that reads a program written in one language –the source language and translates it into an equivalent program in another language-the target language. The compiler reports to its user the presence of errors in the source program.

### **What are the two parts of compiler ?**

Ans : Analysis and Synthesis are the two parts of compilation.

The analysis part breaks up the source program into constituent pieces and creates an intermediate representation of the source program.

The synthesis part constructs the desired target program from the intermediate representation.

### **What is code motion ?**

Ans: Code motion is an optimization technique in which amount of code in a loop is decreased. This transformation is applicable to the expression that yields the same result independent of the number of times the loop is executed. Such an expression is placed before the loop.

### **What are the properties of optimizing compiler ?**

Ans:

The source code should be such that it should produce minimum amount of target code.

There should not be any unreachable code.

Dead code should be completely removed from source language.

The optimizing compilers should apply following code improving transformations on source language.

- common subexpression elimination
- dead code elimination
- code movement
- strength reduction

### **Define symbol table ?**

Ans: Symbol table is a data structure used by the compiler to keep track of semantics of the variables. It stores information about scope and binding information about names.

**List various phases of a compiler ?**

- Lexical Analyzer
- Syntax Analyzer
- Semantic Analyzer
- Intermediate code generator
- Code optimizer
- Code generator

## **Fundamental Computer**

**What is operating system ?**

Ans: An operating system is a program that acts as an intermediary between the user and the computer hardware. The purpose of an OS is to provide a convenient environment in which user can execute programs in a convenient and efficient manner.

**What is data communication ?**

Ans: Data communications (DC) is the process of using computing and communication technologies to transfer data from one place to another, and vice versa. It enables the movement of electronic or digital data between two or more nodes, regardless of geographical location, technological medium or data contents.

**What is Computer network ?**

Ans: A computer network, or data network, is a digital telecommunications network which allows nodes to share resources. In computer networks, computing devices exchange data with each other using connections between nodes

**Types of Computer network ?**

Ans: There are 3 types of computer Network

Local Area Network (LAN) –

LAN or Local Area Network connects network devices in such a way that personal computer and workstations can share data, tools and programs. The group of computers and devices are

connected together by a switch, or stack of switches, using a private addressing scheme as defined by the TCP/IP protocol. Private addresses are unique in relation to other computers on the local network. Routers are found at the boundary of a LAN, connecting them to the larger WAN.

**Metropolitan Area Network (MAN) –**

MAN or Metropolitan area Network covers a larger area than that of a LAN and smaller area as compared to WAN. It connects two or more computers that are apart but resides in the same or different cities. It covers a large geographical area and may serve as an ISP (Internet Service Provider). MAN is designed for customers who need a high-speed connectivity. Speeds of MAN ranges in terms of Mbps. It's hard to design and maintain a Metropolitan Area Network.

**Wide Area Network (WAN) –**

WAN or Wide Area Network is a computer network that extends over a large geographical area, although it might be confined within the bounds of a state or country. A WAN could be a connection of LAN connecting to other LAN's via telephone lines and radio waves and may be limited to an enterprise (a corporation or an organization) or accessible to the public. The technology is high speed and relatively expensive.

### **Give short description of Router, Switch, Bridge, Hub and Gateway.**

**Ans: Router-** A router is a device like a switch that routes data packets based on their IP addresses. Router is mainly a Network Layer device. Routers normally connect LANs and WANs together and have a dynamically updating routing table based on which they make decisions on routing the data packets. Router divide broadcast domains of hosts connected through it.

**Switch-** switch is a multi port bridge with a buffer and a design that can boost its efficiency (large number of ports imply less traffic) and performance. Switch is data link layer device.

**Gateway –** A gateway, as the name suggests, is a passage to connect two networks together that may work upon different networking models. They basically work as the messenger agents that take data from one system, interpret it, and transfer it to another system. Gateways are also called protocol converters and can operate at any network layer. Gateways are generally more complex than switch or router.

**Bridge** – A bridge operates at data link layer. A bridge is a repeater, with add on functionality of filtering content by reading the MAC addresses of source and destination. It is also used for interconnecting two LANs working on the same protocol. It has a single input and single output port, thus making it a 2 port device.

**Hub** – A hub is basically a multiport repeater. A hub connects multiple wires coming from different branches, for example, the connector in star topology which connects different stations. Hubs cannot filter data, so data packets are sent to all connected devices

**Explain the main purpose of an operating system ?**

Ans: Operating systems exist for two main purposes. One is that it is designed to make sure a computer system performs well by managing its computational activities. Another is that it provides an environment for the development and execution of programs.

**What is thread ?**

Ans: A thread is a program line under execution. Thread sometimes called a light-weight process, is a basic unit of CPU utilization; it comprises a thread id, a program counter, a register set, and a stack

**What is virtual memory ?**

Ans: Virtual memory is a memory management technique for letting processes execute outside of memory. This is very useful especially is an executing program cannot fit in the physical memory.

**What is kernel ?**

Ans: Kernel is the core and essential part of computer operating system that provides basic services for all parts of OS.

**What is real-time systems ?**

Ans: Real-time systems are used when rigid time requirements have been placed on the operation of a processor. It has well defined and fixed time constraints.

### What is SMP ?

Ans: SMP is a short form of Symmetric Multi-Processing. It is the most common type of multiple-processor systems. In this system, each processor runs an identical copy of the operating system, and these copies communicate with one another as needed.

# Operating System

## 1. What is Operating System?

A program that, after being initially loaded into the computer by a boot program, manages all software that communicates with the hardware and allows other programs to run. e.g: Windows, Mac OS X and Linux.

1. What are the basic functions of an operating system?
  - a) **File and folder management:** An operating system creates a file structure on the computer hard drive where the data can be stored and retrieved
  - b) **Applications management:** Whenever a program is requested the operating system locates it and loads into the primary memory or RAM.
  - c) **Support for built-in utility programs:** The operating system comes with tools for maintenance and repairs. They identify the problem, they find lost files, repair the damaged ones and do backups for your data.
  - d) **Computer hardware control:** Operating systems facilitates the access of programs to the computer hardware through the BIOS and through device drivers.
2. **The main parts of the operating system can be divided into 4 parts:**
  - i) The Kernel
  - ii) Device Drivers
  - iii) User Interface
  - iv) System Utilities
3. **Operating System can be classified as follows:**
  - a) Multi-user: Allows two or more users to run programs at the same time. Some operating systems permit hundreds or even thousands of concurrent users.
  - b) Multiprocessing: Supports running a program on more than one CPU.
  - c) Multitasking: Allows more than one program to run concurrently.

- d) Multithreading: Allows different parts of a single program to run concurrently.
- e) Real time: Responds to input instantly. General-purpose operating systems, such as DOS and UNIX are not real-time.

#### 4. Describe about Central Processing Unit (CPU)

CPU is responsible for activating and controlling the operations of other units of a computer system. It is also referred to as the brain of computer. CPU consists of two main parts:

**Control Unit (CU):** Control Unit is responsible for coordinating various operations of a computer. It uses time signals to do that. The control unit determines the sequence in which computer programs and instructions are executed. It also acts as a switch board operator when several users access the computer simultaneously. In simple words, it acts like the supervisor seeing that things are done in proper fashion.

**Arithmetic and Logic Unit:** The actual processing of the data and instruction are performed by Arithmetic Logical Unit. ALU takes care of various calculations. The major operations performed by the ALU are addition, subtraction, multiplication, division, logic and comparison. Data is transferred to ALU from storage unit when required. After processing the output is returned back to storage unit for further processing or getting stored.

#### 5. What are the different types of kernel:

##### a) Kernel

Kernel is a computer program that manages input/output requests from software and translates them into data processing instructions for the central processing unit and other electronic components of a computer.

##### b) Micro-Kernel

A microkernel is the near-minimum amount of software that can provide the mechanisms needed to implement an operating system.

##### c) Nanokernel

A nanokernel delegates virtually all services including even the most basic ones like interrupt controllers or the timer to device drivers to make the kernel memory requirement even smaller than a traditional microkernel.

#### 6. What is Thread, Kernel Thread and Multithread Model?

Threads have some of the properties of processes, they are sometimes called lightweight processes. Threads share its code section and data section and other operating systems resources with other threads belonging to the same process.

a) Kernel Thread: Supported directly by the operating system, slower to create and manage than user threads.

b) Multithread Model: Creating user thread requires creating corresponding kernel thread. Creating kernel thread is expensive. Multiplexes many user level threads to a smaller or equal number of kernel threads.

#### 7. What is Paging?

OS performs an operation for storing and retrieving data from secondary storage devices for use in main memory. Paging is one of such memory management schemes. Data is retrieved from storage media by OS, in the same sized blocks called as pages.

#### 8. Describe About CPU Scheduling:

The objective of multiprogramming is to run processes all time, keeping the CPU busy. Uniprocessing system is different. Only one process can run at a time-means processes have to wait until the CPU is free.

CPU Scheduling Algorithm:

- First Come First Served (FCFS)
- Shortest Job First (SJF)
- SRTF
- Priority Scheduling
- Priority Scheduling (Starvation)
- Priority Scheduling (Aging)

#### 9. What is Physical Address and Logical Address?

**Logical Address:** The address which is generated by the CPU which is added with base address to form the physical address is called as logical address.

**Physical Address:** Physical Address is the actual memory address which denotes a memory area in a storage device. It is generated by the MMU (Management Memory Unit)

#### 10. What is NTFS and FAT?

NTFS: New Technology File System (NTFS) is the file system that the Windows NT operating system uses for storing and retrieving files on a hard disk.

FAT: File Allocation Table (FAT) is a table that the operating system uses to locate files on a disk. Versions of the FAT file system:

FAT-12, FAT-16, FAT-32.

#### 11. What do you understand by system clock? What is System Unit?

The system clock is used to produce a specific pulse at a fixed rate of time. The machine cycle of a system can be completed in a single or multiple clock pulse. A single program instruction could be multiple instructions for the CPU. Any central processing unit has a predefined set of instructions also known as the instruction set. These are the instructions that it can process and understand.

The clock speeds are nowadays measuring in Ghz  $1\text{ghz} = 1000\text{ mhz}$ .

#### 12. What is Fragmentation and types of Fragmentation?

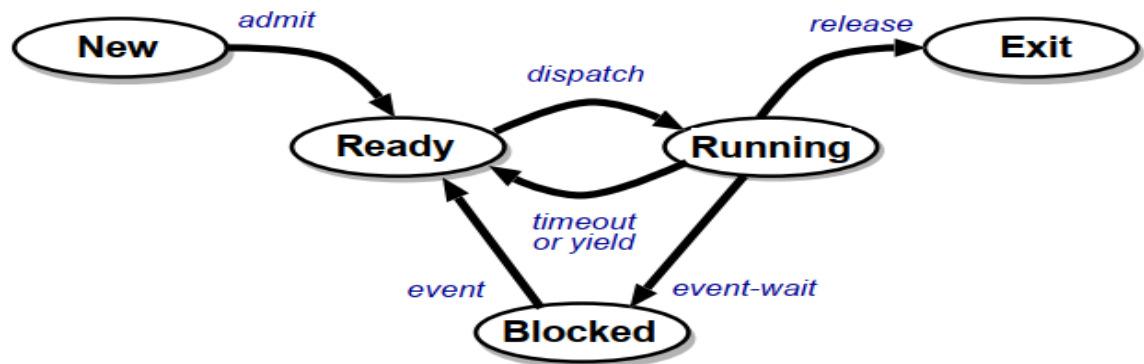
Fragmentation, in the context of a hard disk, is a condition in which the contents of a single file are stored in different locations on the disk rather than in a contiguous space. This results in inefficient use of storage space as well as occasional performance degradation. Users frequently create, modify, delete and save files. Back-end operating systems (OS) continuously store these files on hard drives, which inevitably creates scattered files.

- Internal fragmentation:** Unusable allocated space that reduces performance. Files are stored in clusters, which are mini-allocated hard drive storage areas. Each file is automatically written at the beginning of a cluster, which creates a gap potential between the first and last file bytes, that is, slack space. Internal fragmentation also occurs when additional specific bytes are allocated to each file for alignment.
- External fragmentation:** Unusable allocated storage space. Applications divide and allocate available space as drive data are read and written. Leftover space becomes fragmented due to allocating algorithm weakness, which renders available storage unusable. External fragmentation also occurs when a large number of files are created, modified and deleted, that is, deleted files are divided into smaller stored chunks.



- c) **Data fragmentation:** Occurs when large memory files are divided into smaller pieces and/or an OS tries to allocate large files to an external fragmented storage. For example, when a user creates new files and performs different operations (e.g., renaming, modifying and deletion), extremely small spaces are expected to hold new data files. However, if these new files require more overhead, the OS must seek storage that is greater than average.

### 13. Process States



- As a process executes, it changes state:
  - New: the process is being created
  - Running: instructions are being executed
  - Ready: the process is waiting for the CPU (and is prepared to run at any time)
  - Blocked: the process is waiting for some event to occur (and cannot run until it does)
  - Exit: the process has finished execution.
- The operating system is responsible for maintaining the state of each process.

## Computer Hardware and Maintenance

### What is Hardware?

Hardware is the physical artifacts of a technology

1. What is the writer name of Hardware Book?

Mike Meteyers  
Emmett Dulaney  
Toby Skandier

2. **Describe About Memory.**

A Computer basically has two types of memories:

- a) **Primary memory or Main memory:** The primary memory or the main memory is part of the main computer system. The processor or the CPU directly stores and retrieves

information from it. This memory is accessed by CPU, in random fashion. There are two types of them namely: **RAM** (Read/Write) & **ROM** (Read only Memory)

- b) **Secondary Memory:** the secondary memory is much slower and also less costly. It stores the data permanently unless it is erased. The secondary memory is usually available in the form of floppy disk storage media, hard disk, CD, DVD, Pen drive etc.

3. **What are the different ways in which data can be represented?**

There are three ways in which data can be represented namely Bit, Byte and Word:

**Bit:** This is also the short name for binary digits. A characteristic of BIT is that it can never be empty. Zero implies a power off state whereas one means on state.

**Byte:** A byte is a collection or group of 8 bits. A byte can store a single character which can either be an alphabet, a number or a special character.

**Word:** The number of bits that a CPU possess indicates the power of the computer. It also indicates how many numbers of bytes are present. In today's date most computers can handle 32- or 64-bit length.

4. **What are the steps involved in following a particular instruction given by the CPU?**

- i. The instruction pointer tells the instruction fetch where in the memory the instruction is.
- ii. The fetch takes the instruction and gives it to the decoder, which determines the steps that are necessary to fulfill the instructions.
- iii. The information is then sent to the ALU, which performs the instructions that need to be performed. This includes adding, subtracting, or manipulating the data further.
- iv. Finally, the instructions are sent out into the computer where they are needed.

5. **Describe About RAM.**

Random-access memory is a form of computer data storage that stores data and machine code currently being used.

There are two main types of RAM:

- a) **DRAM (Dynamic Random-Access Memory):** The term dynamic indicates that the memory must be constantly refreshed or it will lose its contents. DRAM is typically used for the main memory in computing devices. If a PC or smartphone is advertised as having 4-GB RAM or 16-GB RAM, those numbers refer to the DRAM, or main memory, in the device.

**There are mainly 5 types of DRAM:**

**Asynchronous DRAM (ADRAM):** The DRAM described above is the asynchronous type DRAM. The timing of the memory device is controlled asynchronously. A specialized memory controller circuit generates the necessary control signals to control the timing. The CPU must take into account the delay in the response of the memory.

**Synchronous DRAM (SDRAM):** These RAM chips' access speed is directly synchronized with the CPU's clock. For this, the memory chips remain ready for operation when the CPU expects them to be ready. These memories operate at the CPU-memory bus without imposing wait states. SDRAM is commercially available as modules incorporating multiple SDRAM chips and forming the required capacity for the modules.

**Double-Data-Rate SDRAM (DDR SDRAM):** This faster version of SDRAM performs its operations on both edges of the clock signal; whereas a standard SDRAM performs its operations on the rising edge of the clock signal. Since they transfer data on both edges of the clock, the data transfer rate is doubled. To access the data at high rate, the memory cells are organized into two groups. Each group is accessed separately.

**Rambus DRAM (RDRAM):** The RDRAM provides a very high data transfer rate over a narrow CPU-memory bus. It uses various speedup mechanisms, like synchronous memory interface, caching inside the DRAM chips and very fast signal timing. The Rambus data bus width is 8 or 9 bits.

**Cache DRAM (CDRAM):** This memory is a special type DRAM memory with an on-chip cache memory (SRAM) that acts as a high-speed buffer for the main DRAM.

- b) **SRAM (Static Random-Access Memory):** While DRAM is typically used for main memory, today SRAM is more often used for system cache. SRAM is said to be static because it doesn't need to be refreshed, unlike dynamic RAM, which needs to be refreshed thousands of times per second. As a result, SRAM is faster than DRAM. However, both types of RAM are volatile, meaning that they lose their contents when the power is turned off.

## 6. Describe about ROM.

Read-only memory (ROM) is a type of non-volatile memory used in computers and other electronic devices. The most common ROM are:

**Programmable Read-Only Memory (PROM):** This type of ROM can be re-programmed by using a special device called a PROM programmer. Generally, a PROM can only be changed/updated once.

**Erasable Programmable Read-Only Memory (EPROM) :** This type of ROM can have its contents erased by ultraviolet light and then reprogrammed by an EPROM programmer. This procedure can be carried out many times; however, the constant erasing and rewriting will eventually render the chip useless.

**Electrically Erasable Programmable Read-Only Memory (EEPROM) :** This type of ROM works in a similar way to Flash memory in that its contents can be 'flashed' for erasure and then written to without having to remove the chip from its environment. EEPROMs are used to store a computer system's BIOS, and can be updated without returning the unit to the factory. In many cases, BIOS updates can be carried out by computer users wishing a BIOS update.

## 7. What is cache memory?

Cache memory, also called CPU memory, is random access memory (RAM) that a computer microprocessor can access more quickly than it can access regular RAM. This memory is typically integrated directly with the CPU chip or placed on a separate chip that has a separate bus interconnect with the CPU.

## 8. What is a flash memory?

Flash memory is a type of nonvolatile memory that erases data in units called blocks. A block stored on a flash memory chip must be erased before data can be written, or programmed, to the microchip. Flash memory retains data for an extended period of time whether a flash-equipped device is powered on or off.

## 9. What is Modem?

Modulation: Modulation – Computer digital signals converted to analog

Demodulation – Analog signal converted back to digital

## 10. What is System Bus? Explain a few related terms.

System Buses are parallel electrical paths that transport data between the CPU and Memory.

Bus Width: The number of electrical paths that to carry the data. It is measured in Bits. With larger Bus Widths, a CPU can transfer more data at a time.

Bus Speed: The speed of the bus is measured in megahertz (MHz), refers to how much data can move across the bus simultaneously. Personal computers have a bus speed of 400 MHz or 533 MHz

## 11. Explain what do you understand by registers, briefly explain the various types of registers.

CPU registers perform a variety of functions, a primary one of which is to offer temporary storage for the CPU to access information stored on the hard drive. Every CPU register has a distinct function and theregisters are essential components of CPU commonly recognized for memory allocation purposes.

**Instruction Register (IR):** The instruction register holds the instruction currently being executed.

**Memory Data Register (MDR):** The memory data register (also known as the memory buffer register or data buffer) holds the piece of data that has been fetched from memory.

**Memory Address Register (MAR):** The memory address register holds the address of the next piece of memory to be fetched.

**Program Counter (PC):** The program counter holds the location of the next instruction to be fetched from memory. It is automatically incremented between supplying the address of the next instruction and the instruction being executed.

**Accumulator:** The accumulator is an internal CPU register used as the default location to store any calculations performed by the arithmetic and logic unit

## 12. What is RISC technology?

RISC (reduced instruction set computer) is a microprocessor that is designed to perform a smaller number of types of computer instructions so that it can operate at a higher speed (perform more millions of instructions per second, or MIPS). Since each instruction type that a computer must perform requires additional transistors and circuitry, a larger list or set of computer instructions tends to make the microprocessor more complicated and slower in operation.

## 13. What do you understand by the term ‘Transistors’? –

Transistors are electronic switches that may or may not allow the flow of current in a current path. When it allows current to flow, the switch is ON. This represents 1 bit. When it does not allow current to flow, the switch is OFF. This represents 0 bit.

Transistors are placed into chips also known as IC. They measure in mm's and are known to contain millions of transistors.

Microprocessors these days are created using microns as their measuring size.

#### 14. What is bursting?

The process of increasing data throughput while RAM transmits data repeatedly.

#### 15. What is buffered RAM?

Buffered memory is a type of computer memory. It is designed to control the amount of electrical current which goes to and from the memory chips at any one time. This makes for more stable memory, but increases the cost and slows the speed at which it works. In a buffered memory system, a hardware register is located between the part of the computer which controls memory and the memory chips themselves. This is a device which can hold a certain amount of information at once. The register will fill up completely and then pass on all this information at once.

#### 16. Some Storage Devices:

**Floppy Disk Drive (FDD):** A Floppy Disk is a magnetic storage medium that uses a diskette made of thin, flexible plastic enclosed in a protective casing.

**Hard Disk Drive (HDD):** HDD systems are used for permanent storage and quick access.

**Solid-State Drive (SSD):** They are metal or glass discs with a magnetic coating on their surface.

#### 17. How Motherboard Works?

- A. Three primary characteristics
  - 1. Form factor determines size of the motherboard and general location of the parts.
  - 2. Chipset defines the type of processor and RAM supported.
  - 3. Built-in components determine functionality and expandability of the system.
- B. The motherboard contains traces or wires through which the data and commands flow. Motherboards typically have traces on multiple layers of the board.

#### 18. Motherboard Northbridge and Southbridge:

##### Northbridge:

- ▶ The North Bridge and South Bridge together form the computer's **chip set**, secondary only to the processor in determining the performance and capabilities of a PC.
- ▶ The North Bridge chip either provides or controls the computer's graphics, RAM, and the **front side bus**, the main highway for data connecting graphics and memory to the CPU.

- ▶ A crucial mechanism in the North Bridge is the memory controller, which constantly renews the memory modules (RAM). Each memory cell with an electrical charge represents a 1 bit. Because the charge begins to dissipate as soon as it's created, the bridge's memory controller endlessly, thousands of times a second, reads each of the millions of cells and writes back the values it read.
- ▶ When the CPU needs data from RAM, it sends a request to the North Bridge memory controller. The controller, in turn, sends the request along to memory and tells the CPU how long the processor must wait to read the memory over a speedy connection called the **front side bus (FSB)**.
- ▶ The North Bridge is also the liaison with the other component for which speed is crucial: the video card. (Some chip sets have video, sound, or other functions built into them, but on chip video is not as fast as a dedicated expansion card).

#### **Southbridge:**

- ▶ The other half of the PC **chip set**, the South Bridge is in charge of input/output with the disk drives, audio, networking and universal serial port.
- ▶ The South Bridge primarily handles the routing of traffic between the various input/output (I/O) devices on the system for which speed is not vital to the total performance, such as the disk drives, optical drives, PCI-Express devices, the older PCI bus, and the USB, Ethernet, and audio ports.
- ▶ It is also responsible for less prominent input/output, such as the real-time clock, interrupt controller, and power management.
- ▶ Some South Bridge chips incorporate audio capabilities good enough to support Dolby Digital and THX multimedia audio.

### **19. Bios(Basic Input / Output System) Functions:**

**Power-on Self Test( POST )** - Takes place right after you power on. It will test computer hardware, ensuring hardware is properly functioning before starting process of loading operating system. If the POST is successful, the BIOS calls INT 19 (Interrupt 19) and then proceeds to look for devices attached to the motherboard.

**Bootstrap Loader** - Process of locating the operating system. If capable operating system located, BIOS will pass the control to it.

### **20. Bootstrapping:**

- a. Tests the system and prepares the computer for operation based on CMOS settings (installed hardware and the configuration settings from the manufacturer and user)
- b. Load the interrupt handlers and device drivers
 

Interrupt handlers - Small pieces of software that act as a translator between the hardware components and the operating system.

Device drivers – other pieces of software that identify the base hardware components such as keyboard, mouse, hard drive or floppy drive.
- c. Initialize registers and power management

### **21. Chipset:**

The exact function of the chip set is constantly changing. But in all cases, the bridges determine what kinds of memory, processors, and other components can work with that particular motherboard.

## Data Communications and Networking

### 1. What is the writer Name of Data Communications and Networking?

Behrouz A. Forouzan

### 2. What is Data Communication?

Data communication is the exchange of data between two devices via some form of transmission medium such as a wire cable.

### 3. Some concepts for Data communication:

#### 1- Message

The message is the information or data that is to be communicated. It may consist of text, numbers, pictures, sounds, videos or any combination of these.

#### 2- Sender

A device that is used for sending messages (or data) is called *sender*. It is also called *transmitter* or *source*. The sender can be a computer, telephone, or a video camera etc. Usually, a computer is used as sender in data communication system.

#### 3- Receiver

A device that is used for receiving messages is called *receiver*. It is also known as *sink*. The receiver can be a computer, telephone set, printer, or a fax machine etc. Usually, a computer is also used as receiver in data communication system.

#### 4- Medium

The path through which data is transmitted (or sent) from one location to another is called transmission medium. It is also called communication channel. It may be a wire, or fiber optic cable, or telephone line etc. If the sender and receiver are within a building, a wire is used as the medium. If they are located at different locations, the medium may be telephone line, fiber optics, microwave or satellite system.

**5-Encoder:** The encoder is an electronic device. It receives data from sender in the form of

digital signals. It converts digital signals into a form that can be transmitted through transmission medium.

**6-Decoder:** The decoder is an electronic device. It receives data from transmission medium. It converts encoded signals (i.e. analog signals) into digital form.

**7-Multiplexing:** Multiplexing is the set of techniques that allows the simultaneous transmission of multiple signals across a single data link. As data and telecommunications use increases, so does traffic.

#### 4. Data Flow:

a) **Simplex:** In *simplex mode*, the communication is unidirectional, as on a one-way street. Only one of the two devices on a link can transmit; the other can only receive. Keyboards and traditional monitors are examples of simplex devices. The keyboard can only introduce input; the monitor can only accept output.

b) **Half-Duplex:** *half-duplex mode*, each station can both transmit and receive, but not at the same time. When one device is sending, the other can only receive, and vice versa. Walkie-talkies and CB (citizens band) radios are both half-duplex systems.

c) **Full Duplex:** In *full-duplex mode*, both stations can transmit and receive simultaneously the full-duplex mode is like a two-way street with traffic flowing in both directions at the same time. One common example of full-duplex communication is the telephone network. When two people are communicating by a telephone line, both can talk and listen at the same time.

#### 5. Describe About Connecting Devices:

**Hub:** A hub connects multiple wires coming from different branches, for example, the connector in star topology which connects different stations.

**Bridge** A bridge operates at data link layer. A bridge is a repeater, with add on functionality of filtering content by reading the MAC addresses of source and destination. It is also used for interconnecting two LANs working on the same protocol. It has a single input and single output port, thus making it a 2-port device.

**Switch:** A switch is a multi-port bridge with a buffer and a design that can boost its efficiency (large number of ports imply less traffic) and performance. Switch is data link layer device. Switch can perform error checking before forwarding data, that makes it very efficient as it does not forward packets that have errors and forward good packets selectively to correct port only.



**Router:** A router is a device like a switch that routes data packets based on their IP addresses. Router is mainly a Network Layer device. Routers normally connect LANs and WANs together and have a dynamically updating routing table based on which they make decisions on routing the data packets. Router divide broadcast domains of hosts connected through it.

**6. What is Gateway?**

A gateway is a passage to connect two networks together that may work upon different networking models. They basically work as the messenger agents that take data from one system, interpret it, and transfer it to another system. Gateways are also called protocol converters and can operate at any network layer. Gateways are generally more complex than switch or router.

**7. What is NIC?**

Network Interface Card(NIC) receives data and convert it into electrical signals. Determine if the data received for a particular computer. Control the flow of data through the cable.

**8. What is Firewall?**

A firewall is a hardware or software system that prevents unauthorized access to or from a network. All data entering or leaving the intranet pass through the firewall, which examines each packet and blocks those that do not meet the specified security criteria.

**9. What is Bandwidth?**

The bandwidth of a composite signal is the difference between the highest and the lowest frequencies contained in that signal.

**10. What is Protocol?**

A network protocol defines rules and convention for communication between network devices.

**11. What is a network?**

A network is a collection of computers, servers, mainframes, network devices, peripherals, or other devices connected to one another to allow the sharing of data. An excellent example of a network is the Internet, which connects millions of people all over the world.

**12. Describe About Network Topology:**

The layout pattern of the interconnections between computers in a network is called network topology.

- a) **Bus Topology:** Uses one main cable to which all nodes are directly connected. The main cable acts as a backbone for the network. One of the computers in the network typically acts as the computer server.
- b) **Star Topology:** Each computer is connected to a central hub using a point-to-point connection. The central hub can be a computer server that manages the network, or it can be a much simpler device that only makes the connections between computers over the network possible.
- c) **Ring Topology:** The computers in the network are connected in a circular fashion, and the data travels in one direction. Each computer is directly connected to the next computer, forming a single pathway for signals through the network.
- d) **Mesh Topology:** Every node has a direct point-to-point connection to every other node. Because all connections are direct, the network can handle very high-volume traffic.

- e) Tree Topology: This topology imitates as extended Star topology and inherits properties of Bus topology. This topology divides the network into multiple levels/layers of network.

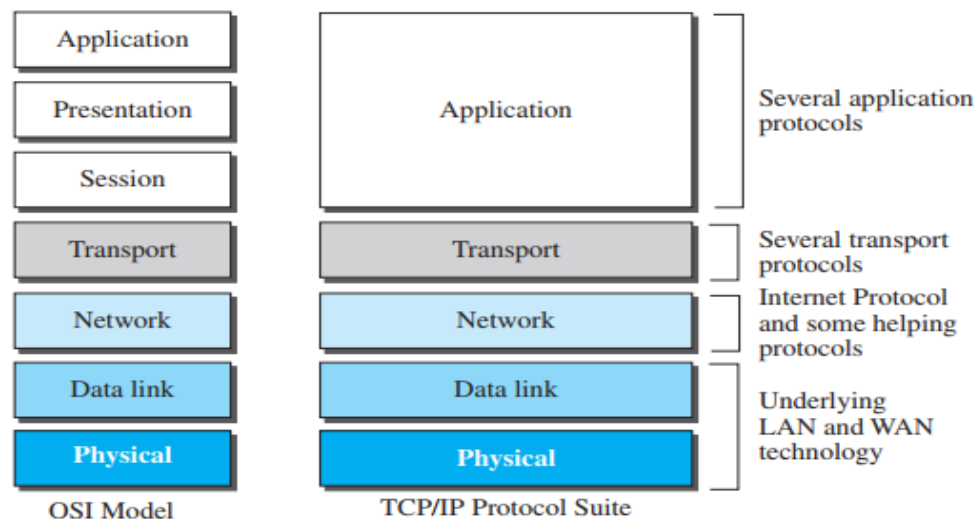
13. Types of Network:

**Local Area Network:** A computer network spanned inside a building and operated under single administrative system is generally termed as Local Area Network (LAN). Usually, LAN covers an organization offices, schools, colleges or universities.

**Metropolitan Area Network:** The Metropolitan Area Network (MAN) generally expands throughout a city such as cable TV network. It can be in the form of Ethernet, Token-ring, ATM, or Fiber Distributed Data Interface (FDDI). This service enables its users to expand their Local Area Networks.

**Wide Area Network:** Wide Area Network (WAN) covers a wide area which may span across provinces and even a whole country. Generally, telecommunication networks are Wide Area Network.

14. OSI Model And TCP/Model:



15. What is IPV4 and IPV6 Address?

**IPV4 Address:** An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a host or a router to the Internet. IPv4 addresses are unique in the sense that each address defines one, and only one, connection to the Internet.

**IPV6 Address:** An IPv6 address consists of eight groups of four hexadecimal digits.

16. IPV4 address classes:

Class A: 1.x.x.x – 126.x.x.x

Class B : 128.0.x.x – 191.255.x.x

Class C: 192.0.0.x – 223.255.255.x

Class D: 224.0.0.0 – 239.255.255.255

Class E: 240.0.0.0 – 255.255.255.254

Private IP Address:

Class A: 10.0.0.0 – 10.255.255.255

Class B: 172.16.0.0 – 172.31.255.255

Class C: 192.168.0.0 – 192.168.255.255

#### **17. Classless IPV4:**

Subnetting: Subnetting allows to create multiple logical networks that exist within a single Class A, B, or C network.

#### **Some values calculated in subnetting:**

- i. Number of subnets: (Given bits for mask – No. of bits in default mask)
- ii. Subnet address: AND result of subnet mask and the given IP address
- iii. Broadcast address: By putting the host bits as 1 and retaining the network bits as in the IP address
- iv. Number of hosts per subnet:  $2^{(32 - \text{Given bits for mask})} - 2$
- v. First Host ID: Subnet address + 1 (adding one to the binary representation of the subnet address)
- vi. Last Host ID: Subnet address + Number of Hosts

## **Graphics-455**

**Question:** What is Computer Graphics?

**Ans:**

Computer graphics is an art of drawing pictures, lines, charts, etc using computers with the help of programming. Computer graphics is made up of number of pixels. Pixel is the smallest graphical picture or unit represented on the computer screen.

**Question:** What is Picture elements (Pixels)?

**Ans:**

The monitor can be thought as a piece of graph paper, split into squares with the help of row and column lines, these squares are called picture element or pixels.

**Question:** Types of Computer Graphics?

Basically, There are two types of computer graphics:

1. **Raster Graphics**, where each pixel is separately defined (as in a digital photograph).
2. **Vector Graphics**, where mathematical formulas are used to draw lines and shapes, which are the interpreted at the viewer's end to produce the graphic.

On the other way, There are Two types of computer graphics:

1. *Interactive Graphics*
2. *Non Interactive Graphics or Passive*

**1. Interactive Graphics:** Interactive computer graphics user have some control over the picture i.e user can make any change in the produced image.

**Description:**

Interactive Computer Graphics involves a two way communication between computer and user. Here the observer is given some control over the image by providing him with an input device for example the video game controller of the ping pong game. This helps him to signal his request to the computer. The computer on receiving signals from the input device can modify the displayed picture appropriately. To the user it appears that the picture is changing instantaneously in response to his commands. He can give a series of commands, each one generating a graphical response from the computer. In this way he maintains a conversation, or dialogue, with the computer.

**For Example:** Its helps to train the pilots of our airplanes.

**1. Non Interactive Graphics:** In non interactive computer graphics images can not be changed. That's why it is also known as passive computer graphics. it is the computer graphics in which user does not have any kind of control over the image

**Description:**

In Passive Computer Graphics, Image is merely the product of static stored program and will work according to the instructions given in the program linearly. The image is totally under the control of program instructions not under the user.**For Example :** Screen Savers

**Question:** Applications of Computer Graphics?

**Ans:**

1. Computer Aided Design (CAD)
2. Presentation Graphics
3. Computer Art
4. Entertainment (animation, games)
5. Education & Training
6. Visualization (scientific & business)
7. Image Processing

**Question:** What is Morphing?

**Ans: Morphing** is a special effect in motion Pictures and animations that changes (or morphs) one image or shape into another through a seamless transition.

**Question:** What is Resolution?

**Ans:** The maximum number of points (pixels) that can be displayed without overlap on a screen is referred to as the resolution, and determines the detail that can be seen in an image.

**Question:** Refresh Rate?

**Ans:** The number of times per second the image is redrawn.

**Question:** What is **Polarization**?

**Ans: Polarization** is a property of waves, such as light and other electromagnetic radiation.

**Question:** What is Rendering?

**Ans: Rendering** is the process of generating an image from a 2D or 3D model (or models in what collectively could be called a *scene* file), by means of computer programs. Also, the results of such a model can be called a rendering.

**Question:** What is Shading?

**Ans:** In computer graphics, shading refers to the process of altering the color of an object/surface/polygon in the 3D scene, based on its angle to lights and its distance from lights to create a photorealistic effect.

**Question:** What is **Compositing**?

**Ans: Compositing** is the combining of visual elements from separate sources into single images, often to create the illusion that all those elements are parts of the same scene.

**Question:** What is OpenGL?

**Ans:** Graphics rendering API (Low Level) High-quality color images composed of geometric and image primitives Window system independent Operating system independent.

# Programming language structure

- ☐ A language is a set of strings of characters from some alphabet
- ☐ The strings of a language are called sentences or statements
- ☐ These small units are called lexemes
- ☐ The lexemes of a programming language include its numeric literals, operators, and special words
- ☐ Lexemes are partitioned into groups—for example, the names of variables, methods, classes

# MIPS instruction set:

**MIPS** (originally an acronym for **Microprocessor without Interlocked Pipeline Stages**) is a reduced instruction set computer (RISC) instruction set (ISA) developed by MIPS Technologies (formerly MIPS Computer Systems, Inc.).

The early MIPS architectures were 32-bit, with 64-bit versions added later. Multiple revisions of the MIPS instruction set exist, including MIPS I, MIPS II, MIPS III, MIPS IV, MIPS V, MIPS32, and MIPS64. The current revisions are MIPS32 (for 32-bit implementations) and MIPS64 (for 64-bit implementations). MIPS32 and MIPS64 define a control register set as well as the instruction set.

# LISP:

- ☐ LISt Processing language

- ☐ Syntax:

- ☒ A study of rule for formation of grammatical sentence in language

The **syntax of a** programming language is the form of its expressions, statements, and program units.

☐ Semantic: is the meaning of those expressions, statements, and program units.

☐ **Backus-Naur Form (BNF):**

A meta language is a language that is used to describe another language.

BNF is a meta language for programming languages.

BNF uses abstractions for syntactic structures.

### ❖ **Management Information System**

1. What is the concept of MIS?

Answer: Management Information System, MIS is a collection of systems, hardware, procedures and people that all work together to process, store, and produce information that is useful to the organization.

2. What are the types of Information Systems?

Answer:

- Transaction Processing Systems (TPS)
- Management Information Systems (MIS)
- Decision Support Systems (DSS)

3. What is Management?

Answer:

- Effective utilization of human and material resources to achieve the enterprise objective
- It is a process consisting of the five basic functions:

- ❖ Planning
- ❖ Organizing
- ❖ Staffing
- ❖ Directing
- ❖ Controlling



#### 4. What are the Functional Areas of Management?

Answer:

- ❖ Production management
- ❖ Marketing management
- ❖ Financial management
- ❖ Personnel or human resource management

#### 5. What Is A Technology Manager?

Answer: Technology management can also be defined as the integrated planning, design, optimization, operation and control of technological products, processes and services, a better definition would be the management of the use of technology for human advantage.

6. Give the concept of data and information?

Answer: **Data** can be defined as a representation of facts, concepts, or instructions in a formalized manner, which should be suitable for communication, inter.

**Information** is organized or classified data, which has some meaningful values for the receiver. Information is the processed data on which decisions and actions are based. Pretention, or processing, by human or electronic machine.

7. What is system and subsystem?

Answer: **Systems** involves basically the pattern or a way in which one thinks about managing optimally. In management Information Systems, it acts as the framework for the visualization and the analyzation of the internal as well as the external environments and the factors affect these particular environments in a very integrated way or pattern.

Subsystem is a relational database management system that has pre-formatted and structured tables for storage of data. These structures are arranged in a way that helps in faster storage and retrieval of such data with adequate security.

8. What are the features of MIS?

Answer:

- ❖ Data Collection
- ❖ Report Generation
- ❖ Accessibility and Integration

9. What is Information Technology?

Answer: Information technology (IT) is the use of any computers, storage, networking and other physical devices, infrastructure and processes to create, process, store, secure and exchange all forms of electronic data.

10. What is the Quality for a Manager?

Answer:

- ❖ Leadership

- ❖ Experience
- ❖ Communication
- ❖ Knowledge
- ❖ Reliability

11. What is the difference between data and information?

- ❖ Data is the input language for a computer and information is the output language for human.
- ❖ Data is unprocessed facts or mere figures but information is processed data which has been made sense of.
- ❖ Data does not depend on information but information depends on data and without it, information cannot be processed.
- ❖ Data is not specific but information is specific enough to generate meaning.
- ❖ Data is the raw material that is collected but information is a detailed meaning generated from the data.

12. What are characteristics of useful information?

Answer:

- ❖ Accuracy
- ❖ Completeness
- ❖ Consistency
- ❖ Uniqueness
- ❖ Timeliness.

❖ **System Analysis and Design**

### 1. What is System Analysis & Design?

Answer: **System Analysis** - Process of gathering and interpreting facts, diagnosing problems, and using the facts to improve the system.

**System Design** - Process of planning a new system to replace or complement the old. Analysis specifies what the system should do and design states how to achieve the objective.

### 2. What is System development life cycle (SDLC)?

**Answer:** A System development life cycle (SDLC) is a process by which systems analysts, software engineers, programmers, and end users build information systems and computer applications. It consists of 5 stages.

- ❖ Problem Identification
- ❖ System Analysis
- ❖ System Design
- ❖ System Implementation
- ❖ System Support & Maintenance

### 3. What does a Systems Analyst do?

Answer:

- ❖ Conduct a study of the feasibility of the proposed system.
- ❖ Liaise with users of the system and determine their requirements.
- ❖ Find out the facts relevant to the design of the proposed system.
- ❖ Determine the human and computer procedures that will make up the system, designing forms, files, reports.
- ❖ Write program specifications.
- ❖ Test the programs and the system.
- ❖ Participate in the implementation of the system.
- ❖ Document the system.
- ❖ Do anything else that will produce an efficient and effective system.

### 4. What skills does a system analyst need?

Answer:

- ❖ The ability to communicate verbally and in writing.
- ❖ To extract relevant facts - a Detective.
- ❖ To obtain information in a reasonable way - a Diplomat.
- ❖ To interpret a jumble of facts and convert them into a logical form.
- ❖ To understand and have a broad knowledge of modern computer hardware & software.
- ❖ To keep up-to-date on System Methodologies.
- ❖ To be creative.

5. What is a DFD?

Answer: A hierarchical set of diagrams which is used to define:

- ❖ The boundary of the system to be developed
- ❖ The information flow to and from the system
- ❖ Data flows within the system
- ❖ The functions used by the system.

6. How DFD is developed?

Answer:

1. Identify inputs & outputs.
2. Label all data flows.
3. Label all processes.
4. Identify data stores.
5. Label all External Entities.
6. Start again.

7. What is Black box testing?

Answer: Different values can be input for variables to determine whether the program can cope with them. These values should include typical values,

borderline values and values which are not acceptable. For example, if a program is written which uses marks out of 100 from a math's examination as input, the test data would include typical data like 27, 73.., borderline data which would be 0 and 100, and unacceptable data like -34, 123, 16.345 This type of testing is called black box testing.

8. What is White box testing?

Answer: White box testing is testing the program to determine whether all the possible paths through the program produce the desired results. As a large program can have a very large number of routes, when you take into account the different condition statements and loops, white box testing is rarely carried out exhaustively.

Think of black box as a test where you cannot see into the box (program) all you see is what comes out at the end. White box testing means that you are able to see what is happening as the data goes through the box because it is transparent.

## ❖ **Software Engineering**

1. What is software engineering?

Answer: Software engineering is an engineering branch associated with software system development.

2. What is computer software?

Answer: Computer software is a complete package, which includes software program, its documentation and user guide on how to use the software.

3. Differentiate computer software and computer program?

Answer: A computer program is piece of programming code which performs a well-defined task whereas software includes programming code, its documentation and user guide.

4. What is software process or Software Development Life Cycle (SDLC)?

Answer: Software Development Life Cycle or software process is the systematic development of software by following every stage in the development process

namely, Requirement Gathering, System Analysis, Design, Coding, Testing, Maintenance and Documentation in that order.

5. What are SDLC models available?

Answer: There are several SDLC models available such as Waterfall Model, Iterative Model, Spiral model, V-model and Big-bang Model etc.

6. What are various phases of SDLC?

Answer: The generic phases of SDLC are: Requirement Gathering, System Analysis and Design, Coding, Testing and implementation. The phases depend upon the model we choose to develop software.

7. What is feasibility study?

Answer: It is a measure to assess how practical and beneficial the software project development will be for an organization. The software analyzer conducts a thorough study to understand economic, technical and operational feasibility of the project.

- ❖ Economic - Resource transportation, cost for training, cost of additional utilities and tools and overall estimation of costs and benefits of the project.
- ❖ Technical - Is it possible to develop this system? Assessing suitability of machine(s) and operating system(s) on which software will execute, existing developers' knowledge and skills, training, utilities or tools for project.
- ❖ Operational - Can the organization adjust smoothly to the changes done as per the demand of project?

8. What are CASE tools?

Answer: CASE stands for Computer Aided Software Engineering. CASE tools are set of automated software application programs, which are used to support, accelerate and smoothen the SDLC activities.

9. What are the categories of software?

Answer:

- ❖ System software

- ❖ Application software
- ❖ Embedded software
- ❖ Web Applications
- ❖ Artificial Intelligence software
- ❖ Scientific software.

10. What is verification and validation?

Answer: **Verification** refers to the set of activities that ensure that software correctly implements a specific function.

**Validation** refers to the set of activities that ensure that the software that has been built is traceable to customer requirements.

11. What is debugging?

Answer: Debugging is the process that results in the removal of error. It occurs as a consequence of successful testing.

12. What is error tracking?

Answer: Error tracking is an activity that provides a means for assessing the status of a current project.

13. What is data design?

Answer: Data design transforms the information domain model created during analysis into the data structures that will be required to implement the software.

14. Define process?

Answer: A series of steps involving activities, constraints, and resources that produce an intended output of some kind is known as process.

15. What is software requirements definition?



Answer: A software requirements definition is an abstract description of the services which the system should provide and the constraints under which the system must operate.

16. What are the advantages and disadvantages of white box testing?

Answer: Advantages: Software's structure logic can be tested.

Disadvantages: Doesn't ensure that user requirements are met.

Its test may not mimic real world situations

17. What are function points?

Answer: Function points are the features which are provided by the software product. It is considered as a most important measurement for software size.

18. What are functional and non-functional requirements?

Answer: Functional requirements are functional features which are expected by users from the proposed software product.

Non-functional requirements are related to security, performance, look, and feel of the user interface.

19. What is the difference between stack and queue?

Answer: Queue is always First In, First Out

Stack is always Last In, First Out

20. Describe the difference between Interface-oriented, Object-oriented and Aspect-oriented programming?

Answer:

❖ Interface programming is contract based.

- ❖ Object-oriented is a way to write granular objects which have a single purpose.
- ❖ Aspect Oriented Programming is to segregate the code in such a manner that various objects carry the main tasks, and the subsidiary tasks are carried by independent objects.

# Database Management System

## 1. What is Data?

Computer **data** is information processed or stored by a computer. This information may be in the form of text documents, images, audio clips, software programs, or other types of **data**.

## 2. What is database?

A **database** is a collection of information that is organized so that it can be easily accessed, managed and updated. Data is organized into rows, columns and tables, and it is indexed to make it easier to find relevant information.

## 3. What is Entities?

An entity is an object that exists. In database administration, an **entity** can be a single thing, person, place, or object. Data can be stored about such entities.

## 4. What is attributes?

An **attribute** defines the information about the entity that needs to be stored. If the entity is an employee, attributes could include name, employee ID, Salary, and Address.

## 5. What is Tuple?

A table has rows and columns, where rows represent records and columns represent the attributes. **Tuple** is a single row of a table, which contains a single record for that relation is called a **tuple**.

## 6. What is Primary Key?

A **primary key** is a special relational database table column designated to uniquely identify all table records. A **primary key's** main features are: It must contain a unique value for each row of data. It cannot contain null values.

## 7. What is Foreign Key?

A **foreign key** is a column or group of columns in a relational database table that provides a link between data in two tables. It acts as a cross-reference between tables because it references the primary **key** of another table, thereby establishing a link between them.

## 8. What are the main categories of SQL queries?

The main categories are-

- DDL (Data Definition Language)
- DML (Data Manipulation Language)
- DQL (Data Query Language)
- DCL (Data Control Language)
- Data administration commands
- Transactional control commands

### **DDL (Data Definition Language):**

Data Definition Language, DDL, is the part of SQL that allows a database user to create and restructure database objects, such as the creation or the deletion of a table.

Some of the most fundamental DDL commands discussed during following hours include the following:

```
CREATE TABLE
ALTER TABLE
DROP TABLE
CREATE INDEX
ALTER INDEX
DROP INDEX
CREATE VIEW
DROP VIEW
```

### **DML (Data Manipulation Language):**

Data Manipulation Language, DML, is the part of SQL used to manipulate data within objects of a relational database.

There are three basic DML commands:

```
INSERT  
UPDATE  
DELETE
```

### **DQL (Data Query Language):**

Data Query Language (DQL) is the most concentrated focus of SQL for modern relational database users. The base command is as follows:

```
SELECT
```

### **Data Control Language**

Data control commands in SQL allow you to control access to data within the database. These DCL commands are normally used to create objects related to user access and also control the distribution of privileges among users. Some data control commands are as follows:

```
ALTER PASSWORD  
GRANT  
REVOKE  
CREATE SYNONYM
```

### **Data Administration Commands**

Data administration commands allow the user to perform audits and perform analyses on operations within the database. They can also be used to help analyze system performance. Two general data administration commands are as follows:

```
START AUDIT  
STOP AUDIT
```

### **Transactional Control Command**

In addition to the previously introduced categories of commands, there are commands that allow the user to manage database transactions.

- COMMIT Saves database transactions
- ROLLBACK Undoes database transactions
- SAVEPOINT Creates points within groups of transactions in which to ROLLBACK
- SET TRANSACTION Places a name on a transaction

## 9. How many join in SQL?

There are four basic types of SQL joins:

- Inner join,
- Left Join
- Right Join
- Full Join

## 10. What is ER Diagram?

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases.

# Data Structure and Algorithm

## 1. What is Data Structure?

Data may be organized in different ways; the logical or mathematical model of a particular organization of data is called a data structure.

## 2. What is Algorithm?

An Algorithm is a well defined list of steps for solving a particular problem. One major purpose of this text is to develop efficient algorithms for processing of data.

## 3. What is Complexity of Algorithm?

Algorithmic complexity is concerned about how fast or slow particular algorithm performs. We define complexity as a numerical function  $T(n)$  - time versus the input size  $n$ .

#### **4. What is Stack?**

A stack is a container of objects that are inserted and removed according to the last-in first-out (LIFO) principle. In the pushdown stacks only two operations are allowed. These are Push and Pop.

#### **5. What is Queue?**

A queue is an ordered collection of items where the addition of new items happens at one end, called the “rear,” and the removal of existing items occurs at the other end, commonly called the “front.” As an element enters the queue it starts at the rear and makes its way toward the front, waiting until that time when it is the next element to be removed.\

#### **6. What is priority Queue?**

A priority queue is a collection of elements such that each element has been assigned a priority and such that the order in which elements are deleted and processed comes from the following rules.

- i) An element of higher priority is processed before any element of lower priority.
- ii) Two elements are processed according to the order in which they were added to the queue.

#### **7. What is the main data structure Operations?**

There are four major operations-

- 2. Traversing: Accessing each record exactly once so that certain item in the record may be processed.
- 3. Searching: finding the location of the records with a given key values.
- 4. Inserting: Adding a New record to the structure.
- 5. Deleting: Removing a record from the structure.

## **Visual Programming and C#**

### **1. What is Visual Programming Language?**

In computing, a visual programming language (VPL) is any programming language that lets users create programs by manipulating program elements graphically rather than by specifying them textually. A VPL allows programming with visual expressions, spatial arrangements of text and graphic symbols, used either as elements of syntax or secondary notation.

## 2. Parts of C# programming:

A C# program consists of the following parts –

- Namespace declaration
- A class
- Class methods
- Class attributes
- A Main method
- Statements and Expressions
- Comments

## 3. Specifying an Attribute

Name of the attribute and its values are specified within the square brackets, before the element to which the attribute is applied. Positional parameters specify the essential information and the name parameters specify the optional information.

```
[attribute(positional_parameters, name_parameter = value, ...)]  
Element
```

## 4. How Many Data types in C#

The variables in C#, are categorized into the following types –

- Value types
- Reference types
- Pointer types

### **Value Type**

Value type variables can be assigned a value directly. They are derived from the class **System.ValueType**.

### **Reference Type**

The reference types do not contain the actual data stored in a variable, but they contain a reference to the variables.

### **Pointer Type**

Pointer type variables store the memory address of another type. Pointers in C# have the same capabilities as the pointers in C or C++.

Syntax for declaring a pointer type is –

## **Theory of Computation**

### **What is Automata?**

The term "Automata" is derived from the Greek word "αὐτόματα" which means "self-acting". An automaton (Automata in plural) is an abstract self-propelled computing device which follows a predetermined sequence of operations automatically.

### **Finite automata**

An automaton with a finite number of states is called a **Finite Automaton** (FA) or **Finite State Machine** (FSM).

### **Formal definition of a Finite Automaton**

An automaton can be represented by a 5-tuple  $(Q, \Sigma, \delta, q_0, F)$ , where –



- **Q** is a finite set of states.
- $\Sigma$  is a finite set of symbols, called the **alphabet** of the automaton.
- $\delta$  is the transition function.
- **q<sub>0</sub>** is the initial state from where any input is processed ( $q_0 \in Q$ ).
- **F** is a set of final state/states of Q ( $F \subseteq Q$ ).

Finite Automaton can be classified into two types –

- Deterministic Finite Automaton (DFA)
- Non-deterministic Finite Automaton (NFA / NFA)

### **Deterministic Finite Automaton (DFA)**

In DFA, for each input symbol, one can determine the state to which the machine will move. Hence, it is called **Deterministic Automaton**. As it has a finite number of states, the machine is called **Deterministic Finite Machine** or **Deterministic Finite Automaton**.

#### **Formal Definition of a DFA**

A DFA can be represented by a 5-tuple  $(Q, \Sigma, \delta, q_0, F)$  where –

- **Q** is a finite set of states.
- $\Sigma$  is a finite set of symbols called the alphabet.
- $\delta$  is the transition function where  $\delta: Q \times \Sigma \rightarrow Q$
- **q<sub>0</sub>** is the initial state from where any input is processed ( $q_0 \in Q$ ).
- **F** is a set of final state/states of Q ( $F \subseteq Q$ ).

### **Non-deterministic finite automata**

In NFA, for a particular input symbol, the machine can move to any combination of the states in the machine. In other words, the exact state to which the machine moves cannot be determined. Hence, it is called **Non-deterministic Automaton**. As it has finite number of states, the machine is called **Non-deterministic Finite Machine** or **Non-deterministic Finite Automaton**.

#### **Formal Definition of an NFA**

An NFA can be represented by a 5-tuple  $(Q, \Sigma, \delta, q_0, F)$  where –

- **Q** is a finite set of states.
- $\Sigma$  is a finite set of symbols called the alphabets.
- $\delta$  is the transition function where  $\delta: Q \times \Sigma \rightarrow 2^Q$

(Here the power set of Q ( $2^Q$ ) has been taken because in case of NFA, from a state, transition can occur to any combination of Q states)

- **q<sub>0</sub>** is the initial state from where any input is processed ( $q_0 \in Q$ ).
- **F** is a set of final state/states of  $Q$  ( $F \subseteq Q$ ).

## DFA vs NDFA

The following table lists the differences between DFA and NFA.

DFA	NFA
The transition from a state is to a single particular next state for each input symbol. Hence it is called <i>deterministic</i> .	The transition from a state can be to multiple next states for each input symbol. Hence it is called <i>non-deterministic</i> .
Empty string transitions are not seen in DFA.	NDFA permits empty string transitions.
Backtracking is allowed in DFA	In NDFA, backtracking is not always possible.
Requires more space.	Requires less space.
A string is accepted by a DFA, if it transits to a final state.	A string is accepted by a NDFA, if at least one of all possible transitions ends in a final state.

## Acceptors, Classifiers, and Transducers

### Acceptor (Recognizer)

An automaton that computes a Boolean function is called an **acceptor**. All the states of an acceptor is either accepting or rejecting the inputs given to it.

### Classifier

A **classifier** has more than two final states and it gives a single output when it terminates.

### Transducer

An automaton that produces outputs based on current input and/or previous state is called a **transducer**. Transducers can be of two types –

- **Mealy Machine** – the output depends both on the current state and the current input.
- **Moore Machine** – the output depends only on the current state.

## Regular Expressions

A **Regular Expression** can be recursively defined as follows –

- $\epsilon$  is a Regular Expression indicates the language containing an empty string. ( $L(\epsilon) = \{\epsilon\}$ )
- $\phi$  is a Regular Expression denoting an empty language. ( $L(\phi) = \{\}$ )
- $x$  is a Regular Expression where  $L = \{x\}$
- If  $X$  is a Regular Expression denoting the language  $L(X)$  and  $Y$  is a Regular Expression denoting the language  $L(Y)$ , then
  - $X + Y$  is a Regular Expression corresponding to the language  $L(X) \cup L(Y)$  where  $L(X+Y) = L(X) \cup L(Y)$ .
  - $X \cdot Y$  is a Regular Expression corresponding to the language  $L(X) \cdot L(Y)$  where  $L(X \cdot Y) = L(X) \cdot L(Y)$
  - $R^*$  is a Regular Expression corresponding to the language  $L(R^*)$  where  $L(R^*) = (L(R))^*$

## Regular Sets

Any set that represents the value of the Regular Expression is called a **Regular Set**.

## Applications of Pumping Lemma

Pumping Lemma is to be applied to show that certain languages are not regular. It should never be used to show a language is regular.

- If  $L$  is regular, it satisfies Pumping Lemma.
- If  $L$  does not satisfy Pumping Lemma, it is non-regular.

## Method to prove that a language $L$ is not regular

- At first, we have to assume that  $L$  is regular.
- So, the pumping lemma should hold for  $L$ .
- Use the pumping lemma to obtain a contradiction –
  - Select  $w$  such that  $|w| \geq c$
  - Select  $y$  such that  $|y| \geq 1$
  - Select  $x$  such that  $|xy| \leq c$
  - Assign the remaining string to  $z$ .
  - Select  $k$  such that the resulting string is not in  $L$ .

## Context-Free Grammar Introduction

**Definition** – A context-free grammar (CFG) consisting of a finite set of grammar rules is a quadruple  $(N, T, P, S)$  where

- $N$  is a set of non-terminal symbols.
- $T$  is a set of terminals where  $N \cap T = \text{NULL}$ .
- $P$  is a set of rules,  $P: N \rightarrow (N \cup T)^*$ , i.e., the left-hand side of the production rule  $P$  does have any right context or left context.
- $S$  is the start symbol.

## Generation of Derivation Tree

A derivation tree or parse tree is an ordered rooted tree that graphically represents the semantic information a string derived from a context-free grammar.

Representation Technique

- **Root vertex** – Must be labeled by the start symbol.
- **Vertex** – Labeled by a non-terminal symbol.
- **Leaves** – Labeled by a terminal symbol or  $\epsilon$ .

## Ambiguity in Context-Free Grammars

If a context free grammar  $G$  has more than one derivation tree for some string  $w \in L(G)$ , it is called an **ambiguous grammar**. There exist multiple right-most or left-most derivations for some string generated from that grammar.

## Basic Structure of PDA

A pushdown automaton is a way to implement a context-free grammar in a similar way we design DFA for a regular grammar. A DFA can remember a finite amount of information, but a PDA can remember an infinite amount of information.

A pushdown automaton has three components –

- an input tape,
- a control unit, and
- a stack with infinite size.

The stack head scans the top symbol of the stack.

A stack does two operations –

- **Push** – a new symbol is added at the top.
- **Pop** – the top symbol is read and removed.

## Pushdown Automata Acceptance

There are two different ways to define PDA acceptability.

### Final State Acceptability

In final state acceptability, a PDA accepts a string when, after reading the entire string, the PDA is in a final state. From the starting state, we can make moves that end up in a final state with any stack values. The stack values are irrelevant as long as we end up in a final state.

For a PDA  $(Q, \Sigma, S, \delta, q_0, I, F)$ , the language accepted by the set of final states  $F$  is –

$$L(\text{PDA}) = \{w \mid (q_0, w, I) \vdash^* (q, \varepsilon, x), q \in F\}$$

for any input stack string  $x$ .

### Empty Stack Acceptability

Here a PDA accepts a string when, after reading the entire string, the PDA has emptied its stack.

For a PDA  $(Q, \Sigma, S, \delta, q_0, I, F)$ , the language accepted by the empty stack is –

$$L(\text{PDA}) = \{w \mid (q_0, w, I) \vdash^* (q, \varepsilon, \varepsilon), q \in Q\}$$

## Parsing

Parsing is used to derive a string using the production rules of a grammar. It is used to check the acceptability of a string. Compiler is used to check whether or not a string is syntactically correct. A parser takes the inputs and builds a parse tree.

A parser can be of two types –

- **Top-Down Parser** – Top-down parsing starts from the top with the start-symbol and derives a string using a parse tree.
- **Bottom-Up Parser** – Bottom-up parsing starts from the bottom with the string and comes to the start symbol using a parse tree.

## **Turing Machine Introduction**

A Turing Machine (TM) is a mathematical model which consists of an infinite length tape divided into cells on which input is given. It consists of a head which reads the input tape. A state register stores the state of the Turing machine. After reading an input symbol, it is replaced with another symbol, its internal state is changed, and it moves from one cell to the right or left. If the TM reaches the final state, the input string is accepted, otherwise rejected.

## **Time and Space Complexity of a Turing Machine**

For a Turing machine, the time complexity refers to the measure of the number of times the tape moves when the machine is initialized for some input symbols and the space complexity is the number of cells of the tape written.

Time complexity all reasonable functions –

$$\mathbf{T(n) = O(n \log n)}$$

TM's space complexity –

$$\mathbf{S(n) = O(n)}$$

