UNIT 1

WHAT IS A COMPUTER?

- A computer is a programmable electronic device that accepts raw data as an input and processes it with a set of instructions (a program) to produce the result as output.
- It produces output just after performing mathematical and logical operations and can save the output for future use.
- It can process numerical as well as non-numerical calculations.
- A computer is designed to execute applications and provides a variety of solutions through integrated hardware and software components.
- It works with the help of programs and represents the decimal numbers through a string of binary digits

- It also has a memory that stores the data, programs, and result of processing.
- The components of a computer such as machinery that includes wires, transistors, circuits, hard disk are called hardware.
- Whereas, the programs and data are called software.
- It is believed that the analytical engine was the first computer which was invented by Charles Babbage in 1837.
- Charles Babbage is known as the father of the computer.

FUNCTIONALITIES OF A COMPUTER

If we look at it in a very broad sense, any digital computer carries out the following five functions −

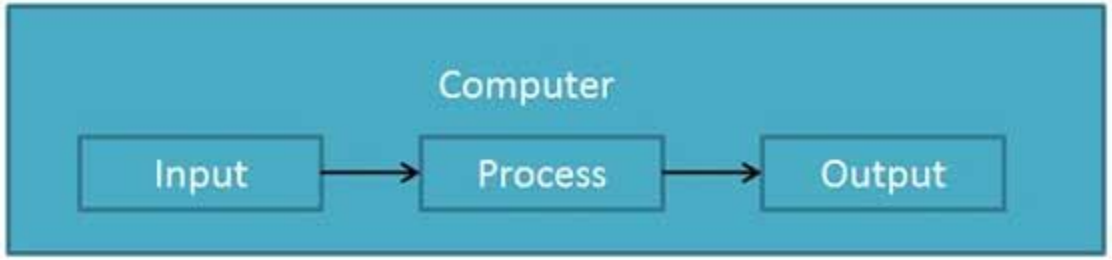
Step 1 − Takes data as input.

Step 2 − Stores the data/instructions in its memory and uses them as required.

Step 3 − Processes the data and converts it into useful information.

Step 4 − Generates the output.

Step 5 − Controls all the above four steps.



CHARACTERISTICS AND ADVANTAGES OF COMPUTERS

Following are certain advantages and characteristics of computers.

1) High speed

- Computer is a very fast device.
- It is capable of performing calculation of very large amount of data.
- The speed rate of the computer depends on its processor.
- The computer has units of speed in microsecond, nanosecond, and even in picosecond.

2) Accuracy

- In addition to being very fast, computers are very accurate.
- Computers perform all jobs with 100% accuracy provided that the input is correct.
- The computers do not get tired; they can do the same task repetitively with the same accuracy.

3) Storage capability

- Memory is a very important characteristic of computers.
- A computer has much more storage capacity than human beings.
- It can store large amount of data.
- It can store any type of data such as images, videos, text, audio, etc.

4) Diligence

- The computer can work continuously without any break and creating error, and it never gets irritated with the work.
- The computer can reply again and again of similar things; still, it can produce an accurate result.
- It can perform repeated tasks with the same speed and accuracy.

5) Versatility

- A computer is a very versatile machine.
- A computer is very flexible in performing the jobs to be done.
- This machine can be used to solve the problems related to various fields.
- At one instance, it may be solving a complex scientific problem and the very next moment it may be playing a card game.

6) Reliability

- A computer is a reliable machine.
- Modern electronic components have long lives.
- Computers are designed to make maintenance easy.

7) Automation

- Computer is an automatic machine.
- Automation is the ability to perform a given task automatically.
- Once the computer receives a program, the program is stored in the computer memory, then the program and instruction can control the program execution without human interaction.

8) Reduction in paper work and cost

- The use of computers for data processing in an organization leads to reduction in paper work and results in speeding up the process.
- As data in electronic files can be retrieved as and when required, the problem of maintenance of large number of paper files gets reduced.
- Though the initial investment for installing a computer is high, it substantially reduces the cost of each of its transaction.

DISADVANTAGES OF COMPUTERS

Following are certain disadvantages of computers.

1) No I.Q.

- A computer is a machine that has no intelligence to perform any task.
- Each instruction has to be given to the computer.
- A computer cannot take any decision on its own.

2) Dependency

- It functions as per the user’s instruction, thus it is fully dependent on humans.

3) Environment

- The operating environment of the computer should be dust free and suitable.

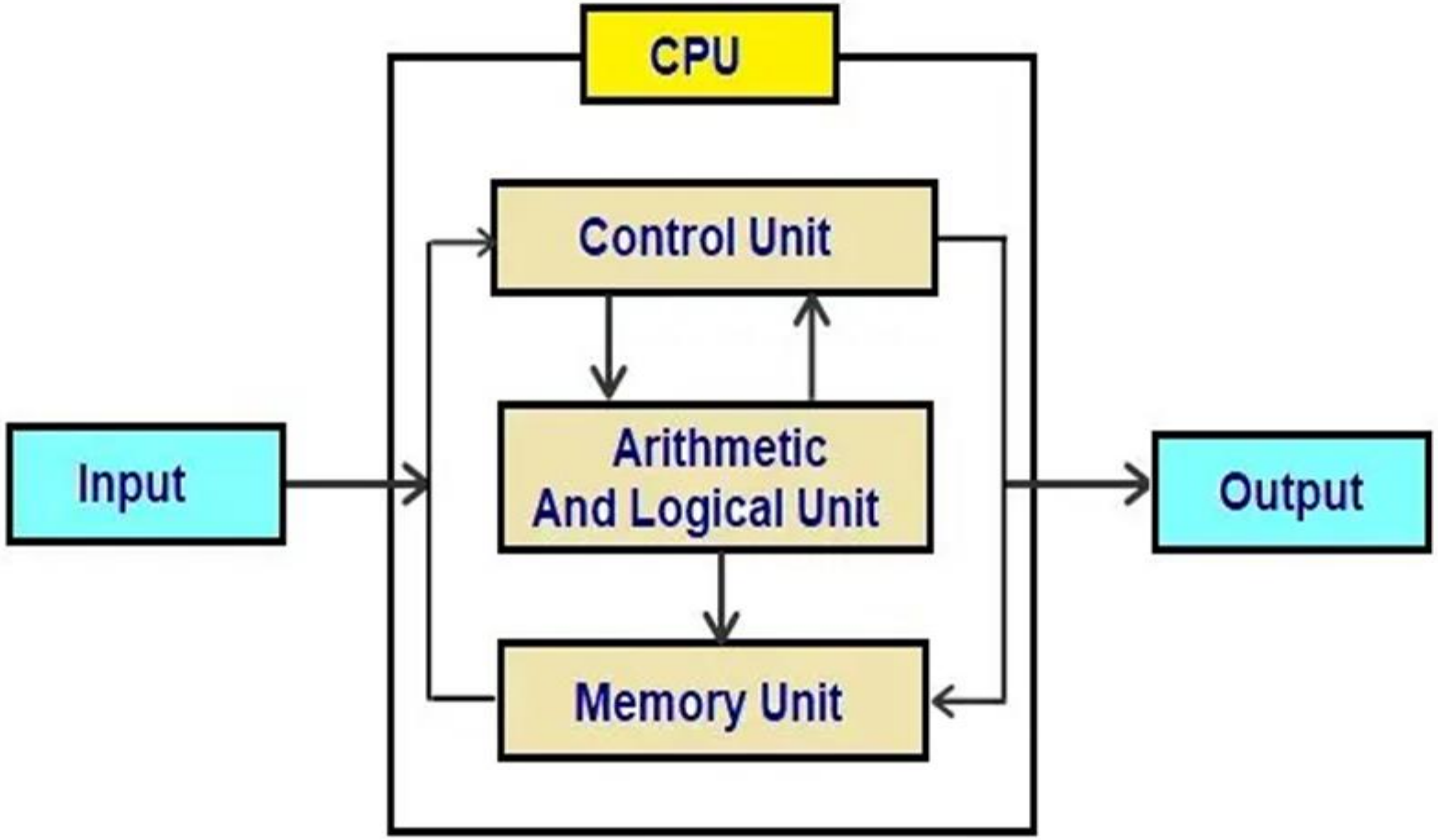
4) No knowledge

- It cannot make judgment based on feeling, taste, experience, and knowledge unlike humans.

Components of computer system

- The hardware and software exist on the computer. The information which is stored through the device is known as computer software.
- The hardware components of the computer system are related to electronic and mechanical parts, and the software component is related to data and computer programs.
- Many elements are connected to the main circuit board of the computer system called a "motherboard."
- Mainly computer system consists of three parts, that are central processing unit (CPU), input devices, and output devices.
- The central processing unit (CPU) is divided into two parts again: Arithmetic and Logical Unit (ALU) and the Control unit (CU).

BLOCK DIAGRAM OF COMPUTER



Block Diagram of a Computer

1) Input

- All the data received by the computer goes through the input unit.
- The input unit comprises different devices like a mouse, keyboard, scanner, etc.
- In other words, each of these devices acts as a mediator between the users and the computer.
- The data that is to be processed is put through the input unit.
- The computer accepts the raw data in binary form.
- It then processes the data, and produces the desired output.
- The 3 major functions of the input unit are-

➢ Take the data to be processed by the user.

➢ Convert the given data into machine-readable form.

➢ Transmit the converted data into the main memory of the computer.

➢ This creates easy communication between the user and computer.

2) CPU – Central Processing Unit

- Central Processing Unit or the CPU, is the brain of the computer.
- It works the same way a human brain works.
- As the brain controls all human activities, the CPU too controls all tasks.
- Moreover, the CPU conducts all the arithmetical and logical operations in the computer.
- Now the CPU comprises of two units, namely – ALU (Arithmetic Logic Unit) and CU (Control Unit).
- Both of these units work in sync.
- It also consists of memory unit.
- The CPU processes the data as a whole.

Let us see what particular tasks are assigned to both units.

a) ALU – Arithmetic Logic Unit

The Arithmetic Logic Unit is made of two terms, arithmetic and logic. There are two major functions that this unit performs.

- Data inserted through the input unit into the primary memory. Performs the basic arithmetical operation on it. Like addition, subtraction, multiplication, and division. It performs all sorts of calculations required on the data. Then sends back data to the storage.
- The unit is also responsible for performing logical operations like, AND, OR, Equal to, Less than, etc. In addition to this it conducts merging, sorting, and selection of the given data.

b) CU – Control Unit

- The control unit as the name suggests is the controller of all the activities/tasks and operations.
- All this is performed inside the computer.
- The memory unit sends a set of instructions to the control unit.
- The control unit converts instructions into control signals.
- These control signals help in prioritizing and scheduling the activities.
- Thus, the control unit coordinates the tasks inside the computer in sync with the input and output units.

c) Memory Unit

- All the data that has to be processed or has been processed is stored in the memory unit.
- The memory unit acts as a hub of all the data.
- It transmits it to the required part of the computer whenever necessary.
- The memory unit works in synchronization with the CPU.
- This helps in faster accessing and processing of the data.
- Thus, making tasks easier and faster.

There are two types of computer memory-

Primary memory – This type of memory cannot store a vast amount of data. Therefore, it is only used to store recent data. The data stored in this is temporary. It can get erased once the power is switched off. Therefore, is also called temporary memory.

RAM and ROM are the examples of primary memory. This memory is directly accessible by the CPU. RAM is used for reading and writing purposes. For data to be processed, it has to be first transferred to the RAM and then to the CPU.

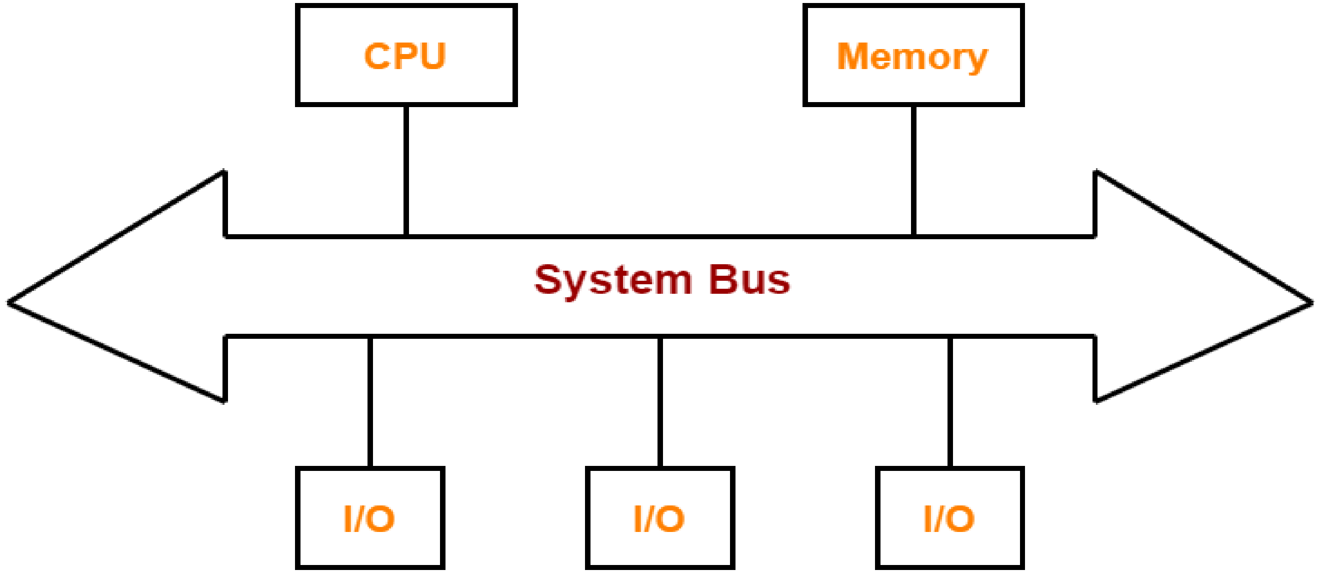
Secondary memory – As explained above, the primary memory stores temporary data. For permanent storage purposes, secondary memory is used. It is also called as permanent memory or the auxiliary memory. The hard disk is an example of secondary memory. Even in a power failure data does not get erased easily.

3) Output

- All the information sent to the computer once processed is received by the user through the output unit.
- Devices like printers, monitors, projector, etc. all comes under the output unit.
- The output unit displays the data either in the form of a soft copy or hard copy.
- The printer is used to get the hard copy and the monitor displays the output.
- The output unit accepts the data in binary form from the computer.
- It then converts it into a readable form for the user.

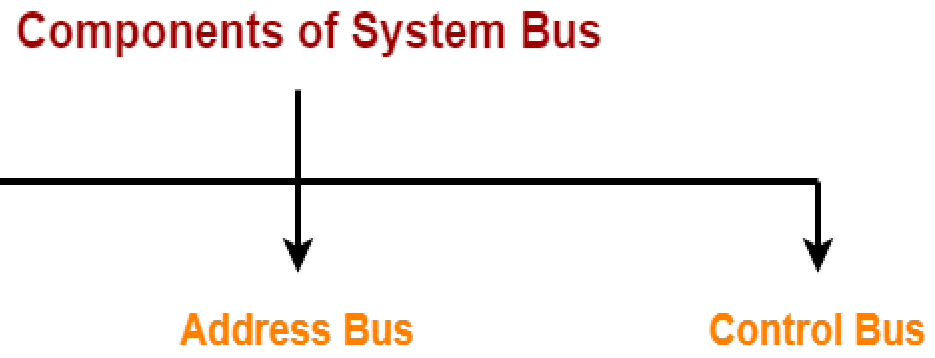
What Is A System Bus?

- A bus is a set of electrical wires (lines) that connects the various hardware components of a computer system.
- It works as a communication pathway through which information flows from one hardware component to the other hardware component.
- A bus that connects major components (CPU, memory and I/O devices) of a computer system is called as a **System Bus**.



Components Of A System Bus-

The system bus consists of three major components-



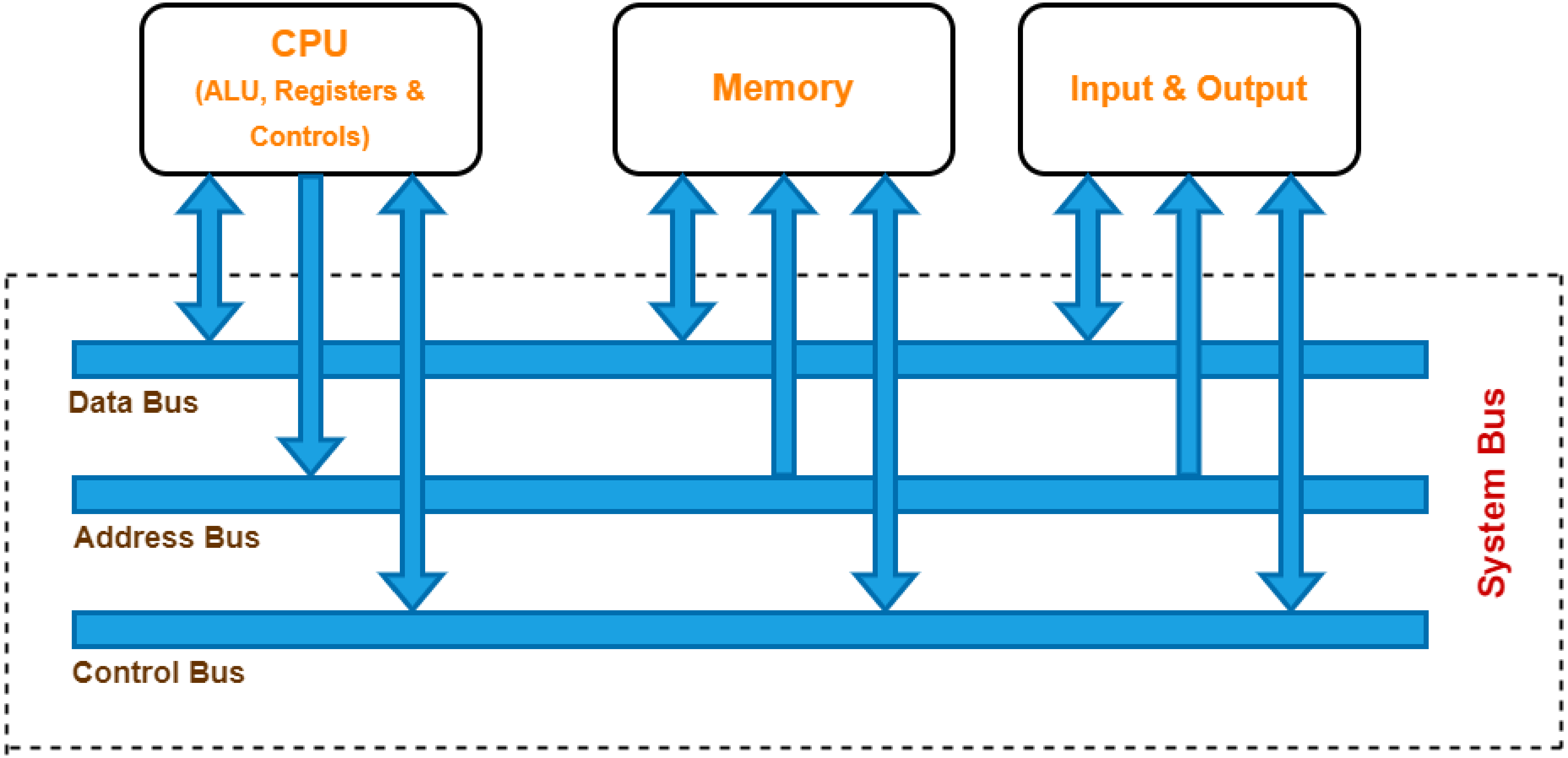
Components of System Bus

Data Bus

Address Bus

Control Bus

- Data Bus
- Address Bus
- Control Bus


CPU
(ALU, Registers & Controls)
Memory
Input & Output
Data Bus
Address Bus
Control Bus
System Bus

1) Data Bus-

- Data bus is used for transmitting the data / instruction from CPU to memory/IO and vice-versa.
- It is bi-directional.
- The width of a data bus refers to the number of bits (electrical wires) that the bus can carry at a time.
- Each line carries 1 bit at a time. So, the number of lines in data bus determine how many bits can be transferred.
- The width of data bus is an important parameter because it determines how much data can be transmitted at one time.
- The wider the bus width, faster would be the data flow on the data bus and thus better would be the system performance.

2) Control Bus-

- Control bus is used to transfer the control and timing signals from one component to the other component.
- The CPU uses control bus to communicate with the devices that are connected to the computer system.
- The CPU transmits different types of control signals to the system components.
- It is bi-directional.
- Control signals are generated in the control unit of CPU.
- Timing signals are used to synchronize the memory and I/O operations with a CPU clock.

Typical control signals hold by control bus-

- **Memory read –** Data from memory address location to be placed on data bus.
- **Memory write –** Data from data bus to be placed on memory address location.
- **I/O Read –** Data from I/O address location to be placed on data bus.
- **I/O Write –** Data from data bus to be placed on I/O address location.

- Other control signals hold by control bus are interrupt, interrupt acknowledge, bus request, bus grant and several others.
- The type of action taking place on the system bus is indicated by these control signals.
- When CPU wants to read or write data, it sends the memory read or memory write control signal on the control bus to perform the memory read or write operation from the main memory.
- Similarly, when the processor wants to read from an I/O device, it generates the I/O read signal.

3) Address Bus-

- Address bus is used to carry address from CPU to memory/IO devices.
- It is used to identify the particular location in memory.
- It carries the source or destination address of data i.e. where to store or from where to retrieve the data.
- It is uni-directional.
- When CPU wants to read or write data, it sends the memory read or memory write control signal on the control bus to perform the memory read or write operation from the main memory and the address of the memory location is sent on the address bus.

- The width of address bus determines the amount of physical memory addressable by the processor.
- In other words, it determines the size of the memory that the computer can use.
- The wider is the address bus, the more memory a computer will be able to use.
- The addressing capacity of the system can be increased by adding more address lines.
- The number of memory locations that can be accessed is dependent on the size of the address bus.
- An ‘n’ bit address bus can access 2^n memory locations.
- Example:

If the size of an address bus = 4,

The no. of memory locations that can be accessed = 2^4 = 16

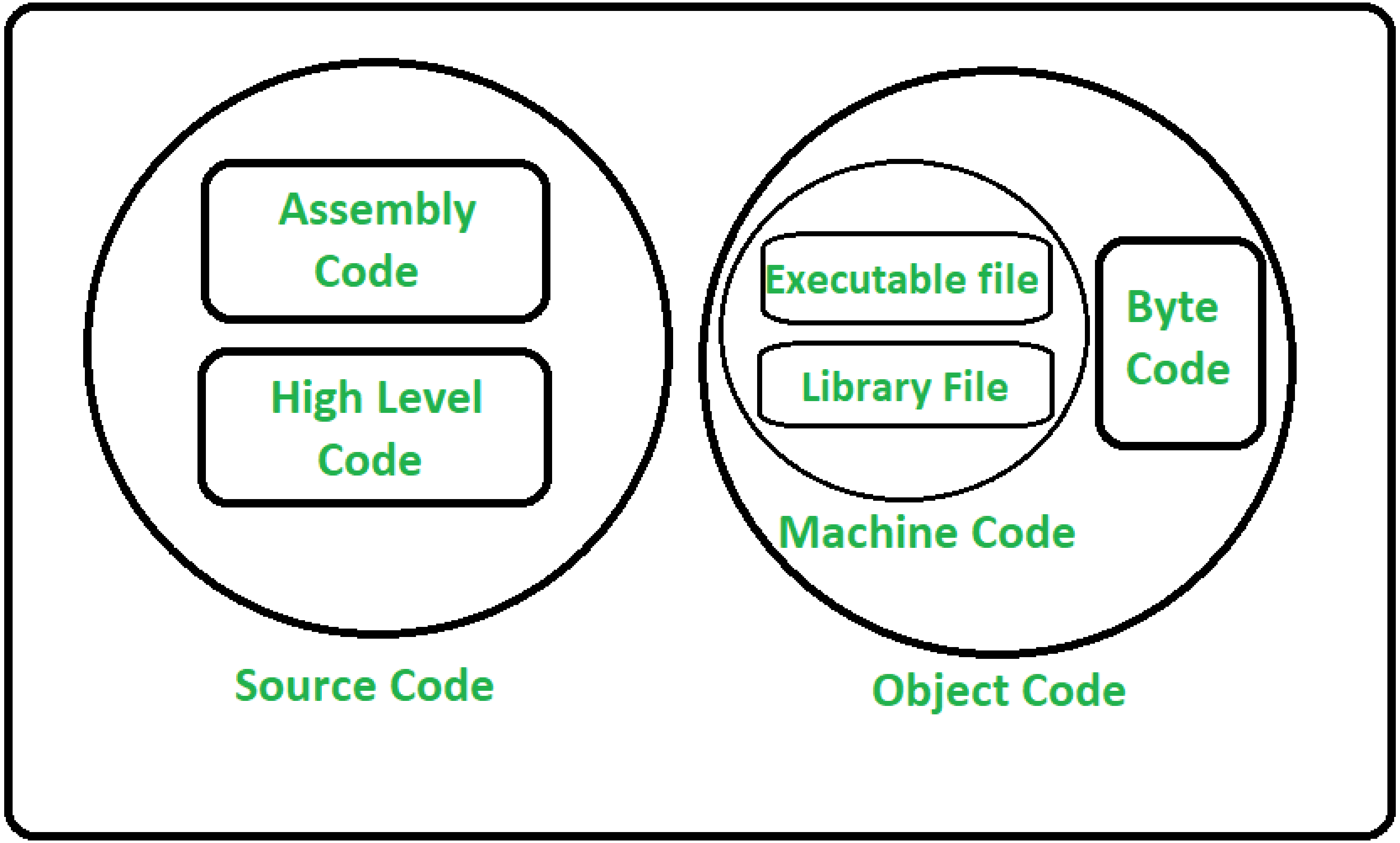
Therefore, 16 memory locations can be accessed using a 4-bit address bus.

Source Code:

- Source code refers to high level code or assembly code which is generated by human/programmer.
- Source code is easy to read and modify.
- It is written by programmer by using any High Level Language or Intermediate language which is human-readable.
- Source code contains comments that programmer puts for better understanding.
- Source code is provided to language translator which converts it into machine understandable code which is called machine code or object code.
- Computer can not understand direct source code, computer understands machine code and executes it.
- It is considered as fundamental component of computer.

Object Code:

- Object code refers to low level code which is understandable by machine.
- Object code is generated from source code after going through compiler or other translator.
- It is in executable machine code format.
- Object code contains a sequence of machine understandable instructions to which Central Processing Unit understands and executes.
- Object file contains object code.
- It consists of one or more machine code(s).


Assembly Code
High Level Code
Source Code
Executable file
Library File
Machine Code
Byte Code
Object Code

Sr. No.	SOURCE CODE	OBJECT CODE
01.	Source code is generated by humans or programmer.	Object code is generated by compiler or other translator.
02.	Source code is high level code.	Object code is low level code.
03.	Source code is written in plain text by using some high level programming language.	Object code is translated code of source code. It is in binary format.
04.	Source code is human understandable.	Object code is not human understandable.
05.	Source code is not directly understandable by machine.	Object code is machine understandable and executable.
06.	It is written in a high-level language like C, C++, Java, Python, etc., or assembly language.	It is written in machine language through compiler or assembler or other translator.

07.	It can be easily modified.	It can not be modified.
08.	It contains comments for better understanding by programmer.	It does not contain comments for understanding by machine.
09.	It contains less number of statements than object code.	It contains more number of statements than source code.
10.	It is less close towards machine.	It is more close towards machine.
11.	Source code is input to compiler or any other translator.	Object code is output of compiler or any other translator.
12.	Source code is not system specific.	Object code is system specific.

14.	It can be changed over time.	Source code needs to be compiled or translated by any other translator to get modified object code.
15.	Language translators like compiler, assembler, interpreter are used to translate source code to object code.	Object code is machine code so it does not require any translation.
16.	The source lines of code gives the readability and understandability to the user. Use of fewer lines of code gives better performance by giving same results in most cases.	This is not the case with object code.

Computer Language Translator and its Types

- Computers only understands machine language.
- A computer understands instructions in machine code, i.e. in the form of 0s and 1s.
- It is a tedious task to write a computer program directly in machine code.
- The programs are written mostly in high-level languages like Java, C++, Python etc. and are called source code.
- These source code cannot be executed directly by the computer and must be converted into machine language to be executed.
- Hence, a special translator system software is used to translate the program written in a high-level language into machine code is called Language Processor and the program translated into machine code is called an object program/object code.
- There are two types of languages, low level language and high level language.
- Low level language is machine understandable.
- High level language is understandable by humans/users.

- Translator is a computer program that translates program written in a given programming language into a functionally equivalent program in a different language.
- It converts the source code into machine code.
- It provides an interface to the computer to read high level languages.
- Depending on the translator, this may involve changing or simplifying the program flow without losing the essence of the program, thereby producing a functionally equivalent program.

Types of Language Translator

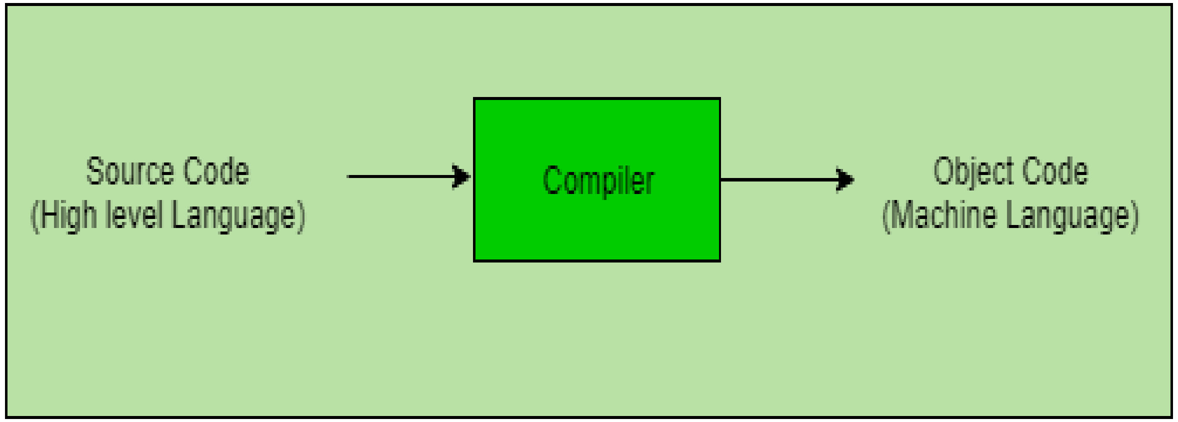
There are mainly three Types of translators which are used to translate different programming languages into machine equivalent code:

- Assembler
- Compiler
- Interpreter

- Compilers, interpreters, translate programs written in high-level languages into machine code that a computer understands.
- And assemblers translate programs written in low-level or assembly language into machine code.
- In the compilation process, there are several stages, to help programmers write error-free code, tools are available.
- Assembly language is machine-dependent, yet mnemonics used to represent instructions in it are not directly understandable by machine and high-Level language is machine-independent.

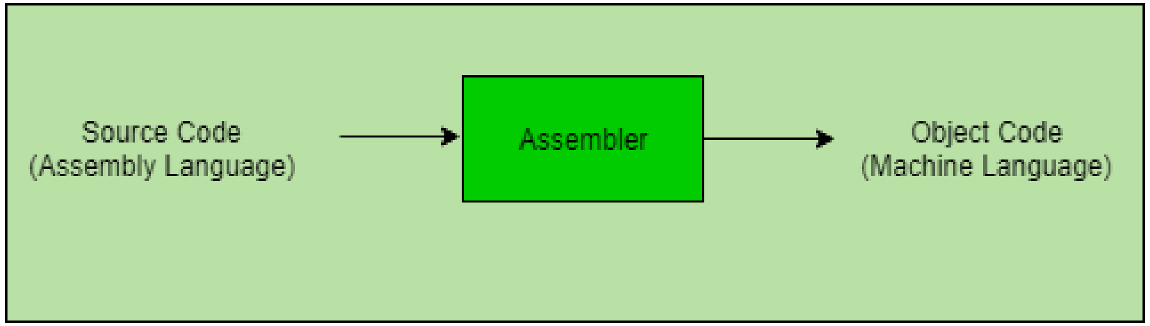
1. Compiler :

- A compiler is a translator used to convert high level programming language to low-level programming language.
- Example: C, C++, Java.
- It converts the whole program in one session and reports errors detected after the conversion.
- An error report is often produced after the full program has been translated.
- The source code is translated to object code successfully if it is free of errors.
- The compiler specifies the errors at the end of the compilation with line numbers when there are any errors in the source code.
- The errors can only be fixed by changing the original source code and compiling the program again.
- Compiler takes time to do its works as it translates the code all at once and then saves it to memory.



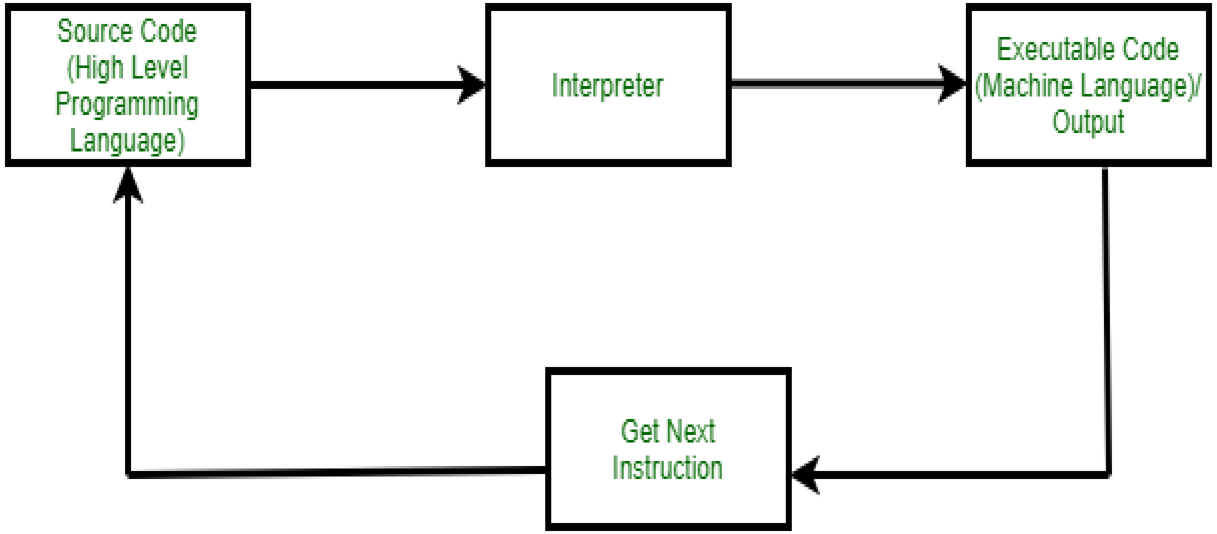
2. Assembler :

- The Assembler is used to translate the program written in Assembly language into machine code.
- The source program is an input of an assembler that contains assembly language instructions.
- The output generated by the assembler is the object code or machine code understandable by the computer.
- Code written in assembly language is some sort of mnemonics(instructions) like ADD, MUL, MUX, SUB, DIV, MOV and so on.
- The assembler is basically able to convert these mnemonics in Binary code.
- Here, these mnemonics depends upon the architecture of the machine.
- For example, the architecture of 8085 and 8086 microprocessors are different, hence the instruction set will also be different.



3. Interpreter :

- An interpreter translates a single statement of the source program into machine code and executes it immediately before moving on to the next line.
- If there is an error in the statement, the interpreter terminates its translating process at that statement and displays an error message.
- The interpreter moves on to the next line for execution only after the removal of the error.
- An Interpreter directly executes instructions written in a programming language without previously converting them to an object code or machine code.
- Example: Perl, Python and Matlab.


Source Code (High Level Programming Language)
Interpreter
Executable Code (Machine Language)/ Output
Get Next Instruction

Difference between Compiler and Interpreter –

Compiler	Interpreter
A compiler is a program that converts the entire source code of a programming language into executable machine code for a CPU.	An interpreter takes a source program and runs it line by line, translating each line as it comes to it.
The compiler takes a large amount of time to analyze the entire source code but the overall execution time of the program is comparatively faster.	An interpreter takes less amount of time to analyze the source code but the overall execution time of the program is slower.
The compiler generates the error message only after scanning the whole program, so debugging is comparatively hard as the error can be present anywhere in the program.	Its Debugging is easier as it continues translating the program until the error is met.

The compiler requires a lot of memory for generating object codes.	It requires less memory than a compiler because no object code is generated.
Generates intermediate object code.	No intermediate object code is generated.
For Security purpose compiler is more useful.	The interpreter is a little vulnerable in case of security.
Examples: C, C++, Java	Examples: Python, Perl, JavaScript, Ruby

Difference between Compiler and Assembler:

Compiler	Assembler
Compiler converts the source code written by the programmer to a machine level language.	Assembler converts the assembly code into the machine code.
Compiler input source code.	Assembler input assembly language code.
It converts the whole code into machine language at a time.	The Assembler does not do this at once.
It takes less execution time compared to an assembler.	It takes more time than a compiler.
It shows the whole program error after the whole program is scanned.	It detects errors in the first phase, fixes them, and then the second phase is start to execute.