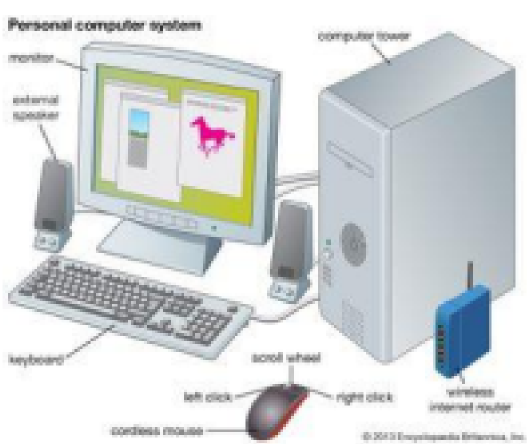
Computer: General: Descriptive questions

1. What is a computer? List the main parts of a general purpose computer.

A computer is a programmable electronic device for processing, storing, and displaying information. The first computers were used primarily for numerical calculations, and designed as a huge calculators. With the development of science, technology and software engineering, numbers, symbols (letters/alphabets/different types of symbols), audio, video and different types of physical parameters (temperature, pressure, flow, speed etc) are encoded and processed in a computer through a sequential list of arithmetic, logical and other commands, called programs. Arithmetic operations are performed to solve simple as well as complex mathematical, scientific and business computations. Different types of raw data; numbers, symbols, letters, audio, video, physical parameters are also processed through different types of programs and generate meaningful information. Computers are extensively used to control industrial process control. Computers are also used as one of the functional units of large and complex systems. Figures below show desktop computer as it appears and main electronic parts of a computer.



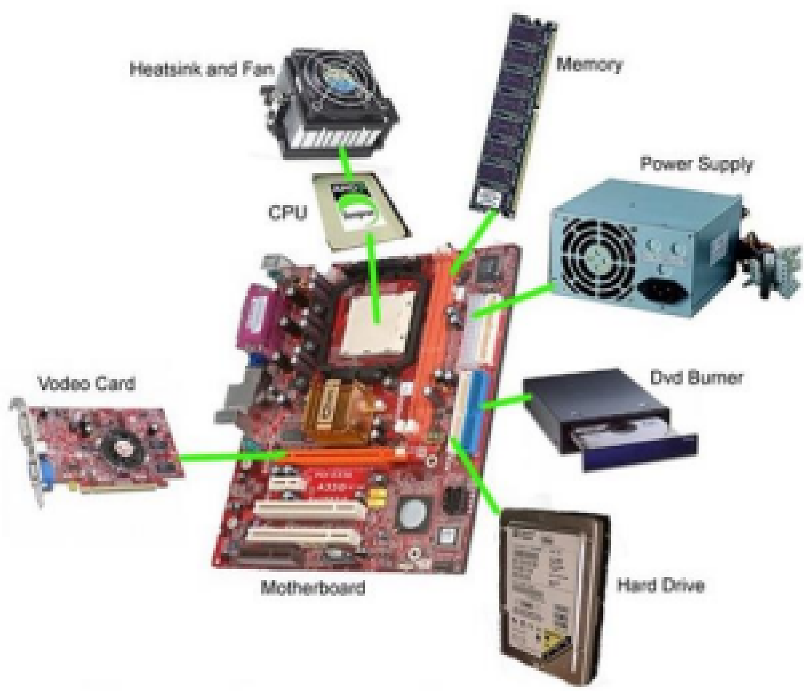
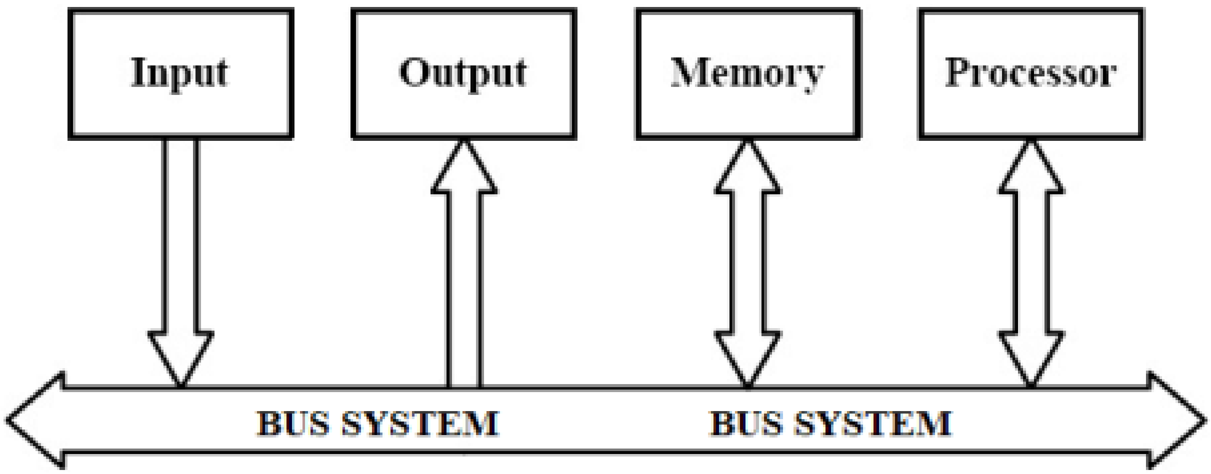
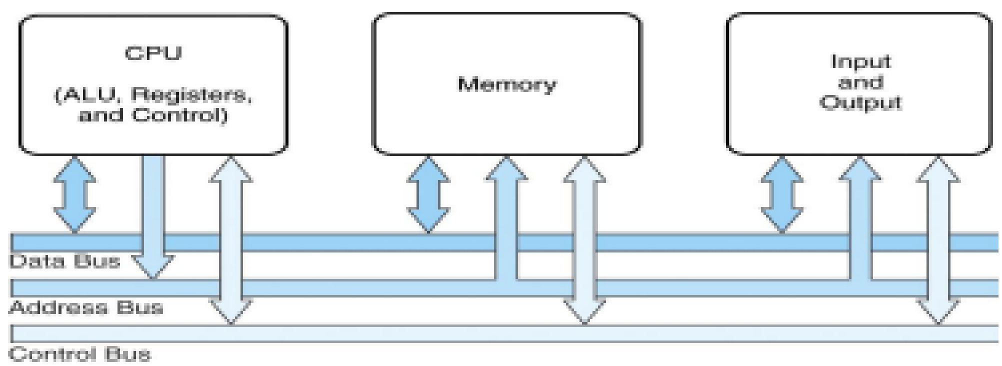


Figure: Desktop computer Figure: Main electronic parts of a computer Main electronic parts of a general purpose computer are: Processor (CPU), memory, motherboard, secondary storage (hard drive), video card, power supply, input & output devices and connectors.

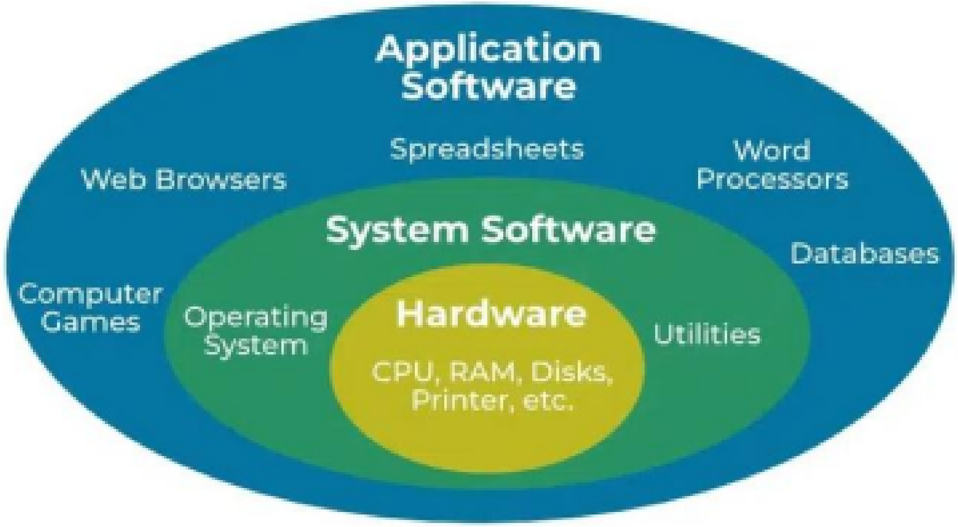
2. What types of diagrams are used to represent and study computer operation/organization & architecture? To understand and show the interconnections of different parts/functional units, oversimplified schematic/block diagrams are used as follows.



Main functional units are simply represented by blocks and interconnections are shown by pointed arrows and lines, called buses. Pointed arrows show the flow of signals (information and command) in the system. The bus system consists of three types of buses: data bus, address bus and control bus. To elaborate the bus system, following diagram can be used.



Layered diagrams are also used to show how users programs are run; implemented on electronic hardware. Applications software is used by users to carry out specific tasks but system software is needed to manage hardware resources: CPU, RAM, I/O devices etc to implement any task on a computer.

	 Moreover, layers of abstraction are also used to show organization and architecture of computer hardware and software.
3.	List the major components/functional units of a computer. A computer consists of following functional units • Central Processing Unit (CPU) • Main Memory Unit • Input and Output Devices • Secondary Storage • Bus System (a network of conducting wires connecting-interconnecting all functional units)

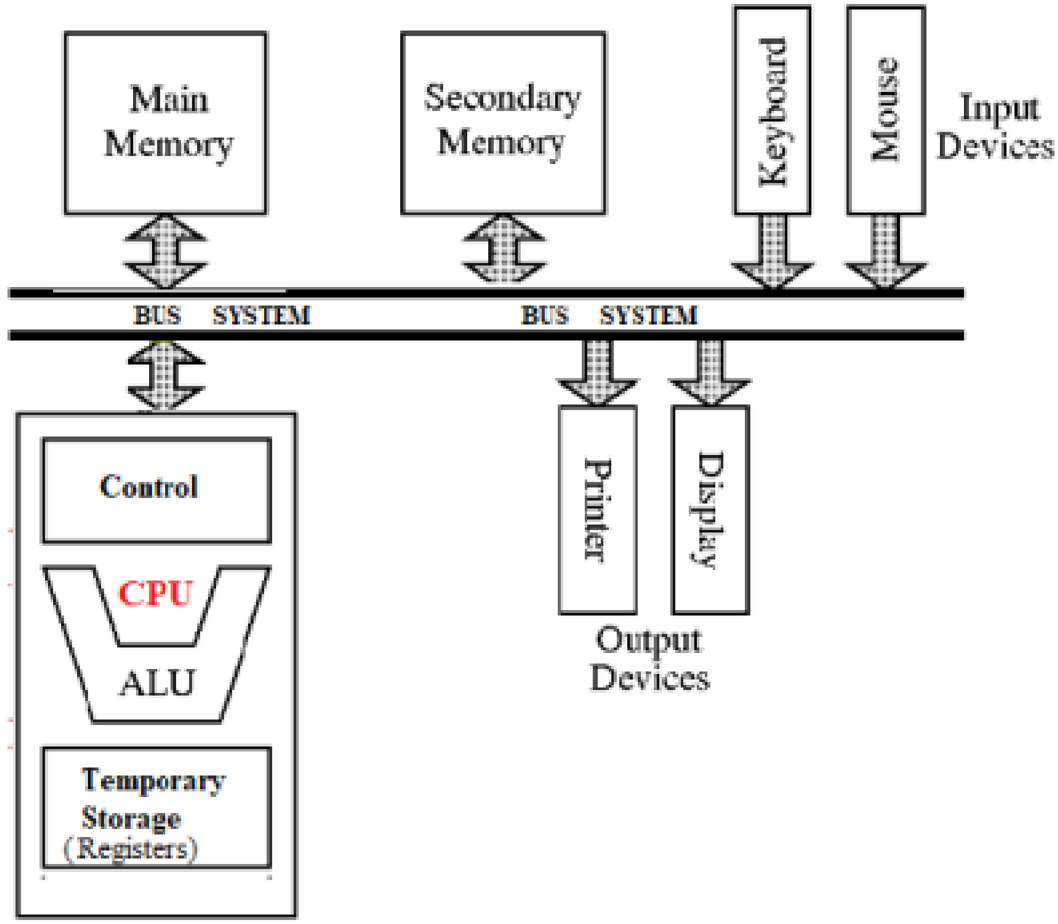


Figure: Block diagram of a general purpose computer

CPU: Part of a computer that controls all the other parts. It defines the performance of your computer. CPU is the electronic circuitry that executes instructions comprising a computer program. The CPU performs basic arithmetic (addition, subtraction, multiplication, division etc), logic (AND, OR, shift, rotate etc), controlling, and input/output operations.

CPU is the part of a computer which implements operations listed in programs.

Each CPU is designed to have:

electronic circuits for arithmetic and logical operations, called ALU,

a number of high speed temporary electronic storage, called registers

and a section that converts software programs to electronic signals and controls operations of all other functional units, called control unit.

Since 1970's, CPU is designed on a single Integrated Circuit (IC), called Microprocessor. Modern CPU contains multiple ALUs, high speed memory and many more functional units.

Memory: Most computers have two types of memory: Main memory and Secondary storage. Main Memory: which contains two kinds of memory: RAM and ROM.

RAM (Random Access Memory) is the most common form of Main Memory.

RAM is used to hold both program code and data. Programs are “loaded” into RAM from a disk prior to execution by the CPU.

When a program is run, both program code and data are transferred from RAM to temporary electronic storage (called registers) within the CPU. If any code or data is transferred from RAM to register, called READ operation.

When a program is run, results or partial results are temporarily stored in temporary electronic storage (called registers) within the CPU. Then results or partial results may be transferred from registers to RAM for further uses. If contents of any register is transferred (saved) to RAM, called WRITE operation. The time to read or write is referred to as the **access time** and is constant for a RAM. The contents of RAM are lost if the power is turned off and this characteristic is called volatile. ROM: Read only memory. It contains “firmware”: software permanently stored in hardware. This software is used to start the computer. Contents of ROM are not lost when the computer is turned off.

Secondary Storage: Used to save programs and data. Contents are not lost when computer is turned off.

Input/Output Devices: Keyboards, monitors, touchscreen, printer, mouse etc are used to input data to system and output processed data/information.

Computer I/O devices are often referred as peripherals. There are two general classes of I/O interface techniches to connect I/O devices.

Serial Interface port: transmits/receives data one bit at a time. Can transmit over long distance. Parallel interface port: Transmits/receives 8 bits or more bits at a time. High speed data transfer. Limited by cable length.

Bus System: A bus is a collection of electronic pathway through which data can be transferred. The CPU, memory, and I.O are separate electronic modules that are interconnected by buses. Buses play an essential role in modern computers by connecting all computer parts together with an efficient communication method. Buses enable the subsystems -such as I/O ports, RAM and ROM -to interact with one another by providing fast data transfer rates. Without these pathways – computers simply wouldn’t work properly. Any bus system contains three types of pathways, namely Data bus, Address bus and Control bus. Buses are named based on information/signal carried/transferred through these. Number of wires/conducting lines in different types of buses depend on CPU, capacity of RAM and I/O devices.

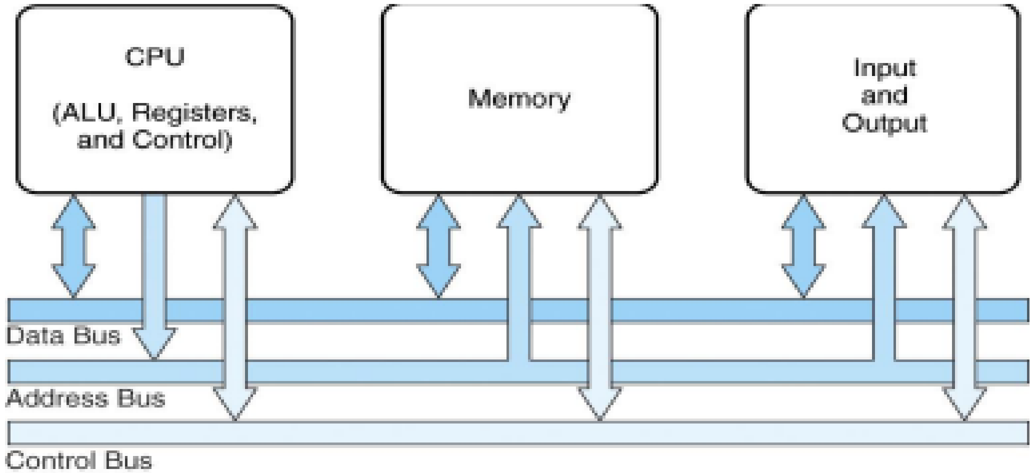


Figure: Block diagram of a general purpose computer

4. What is Software?

Software is a set of computer programs. A **program** is a sequence of command, called instruction that performs a basic/elementary arithmetic/logical operation/computation/data processing/data transfer. A **computer** is a combination of programs and hardware (electronic parts/functional units).

5. What are different levels of programming?

High level program: Contains English words, arithmetic signs, notations and symbols. This programming language is machine (CPU) independent. The programmers need to learn syntax of language and algorithm to write programs to solve any problem. Example: a program to calculate area of a triangle.

```
def main():
 print "Program calculates the area of a triangle."
 print
 a, b, c = input("Enter triangle's sides length: ")
 s = (a+b+c) / 2.0
 area = sqrt(s*(s-a)*(s-b)*(s-c))
 print "The area is %.2f" % area

main()
```

Low-level program: This programming language contains mnemonics (term or word used to define or specify a computing function), names of registers, memory addresses, numbers (binary, decimal, hexadecimal), characters etc. Each line is a command for CPU for a basic arithmetic/logical or data transfer operation, called Instruction. This programming language is machine (processor/hardware) dependent and instructions must be chosen from the list of instructions that a particular processor supports (designed for). The full list of instructions and its formats are found in the data sheet provided by the manufacturer. Here is an example for few instructions in Assembly language (low level program). Each line is called an instruction and it instructs the CPU for an elementary/basic operation the hardware is designed for. For example, MUL r1, r2 stands for multiplication of two numbers stored in CPU registers r1 and r2. The product will be saved into register r1.

```
LOAD r1,b
LOAD r2,h
MUL r1,r2
DIV r1,#2
RET
```

An **instruction** is a command to hardware for a single arithmetic/logical/data transfer operation. It is required to specify operation and data (numbers used in operation) and location to store result in an instruction. In instructions, locations of data to be used in operations are indicated instead. An instruction has following format, in general

Operation	Operand-1	Operand-2	Result field
(mnemonics for operation)	(data or register or memory address data)	(data or register or memory address data)	(register or memory address to store result)

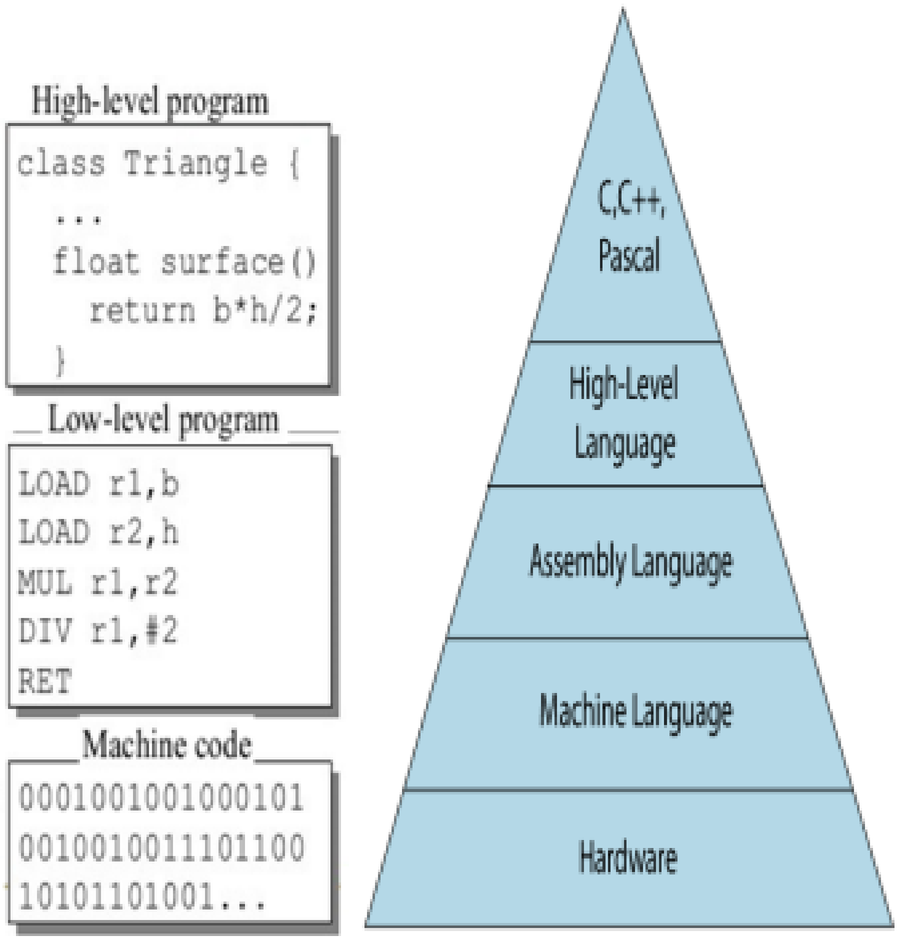
Machine code: The lowest level of programming where instructions are formed in binary code which is not readable at all and fully machine (processor/hardware) dependent. Each instruction in low level, as indicated above, is encoded in a pre-defined binary bits patterns followed by processors.

In early days, trained technicians and engineering used to form instructions in binary codes to write programs to solve any problems and it used to take long time and patience. Moreover, technicians used to follow the binary bit patterns assigned for different operations. It was extremely difficult to correct programs in case of mistakes.

Example of machine code:

	0001001001000101 0010010011101100 10101101001... Here each line represent an instruction for the CPU. In modern digital computers, Programs in high-level language and low level languages are converted into machine code and saved to memory/storage devices. Machine codes are loaded into RAM and then CPU reads it and implement/execute the operation electronically.
6.	What are the Levels of Program code? We usually write programs in high-level languages without thinking about on what machine our program will run. Electronics of the CPU and computer recognize binary representations of instructions. That means each CPU has its own machine language that it understands. Programs are typically written in higher-level languages and then translated into machine language (executable code).

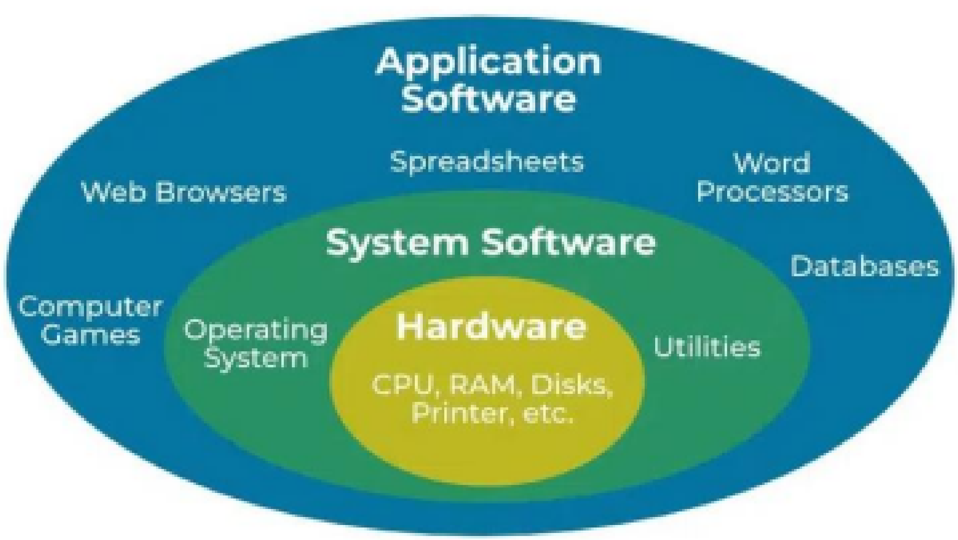
A **compiler** is a program that translates code written in one language into another language. An **interpreter** translates the instructions one line at a time into something that can be executed by the computer's hardware.



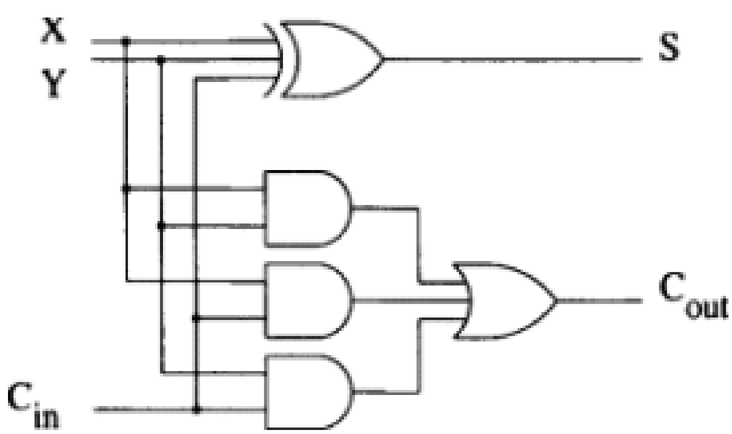
7.

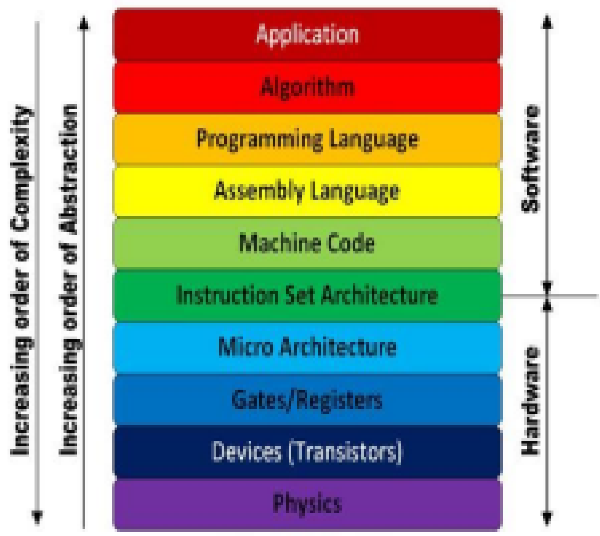
8.	What is system software? Systems software is software that controls the hardware, including any peripherals, allows other programs (applications) to run and provides an interface for the user to interact with the computer and maintains the system. There are two kinds of systems software to consider: operating systems and utilities. The "operating system" of a computer is like a first, supervisory program that begins running when the computer first starts up ("boots up"). The operating system plays an invisible administrative and bookkeeping role behind the scenes. When a desktop/laptop starts up, the operating system typically gets things organized and then launches a "file explorer" program which displays available programs and menus etc. that show the user what is available, allowing the user to navigate and run programs. The operating system keeps things organized in the background so that multiple programs can run at the same time, which is known as "multitasking". The operating system gives each program its own area of memory, so each program only accesses its own resources. Keeping the programs separate so it operates independently, without interfering with other programs or the system as a whole. Similarly, each program has some access to the screen through a window, but this output area is separated from the output of other programs.

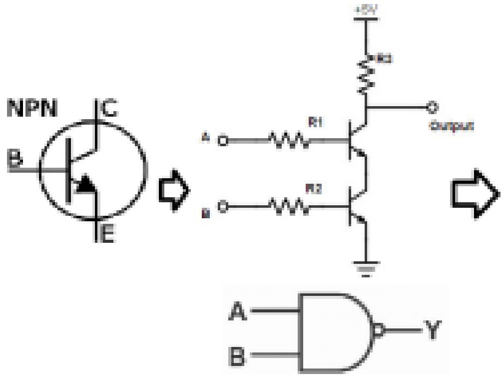
9.	What is Application software? An application program is a computer program designed to carry out a specific task other than one relating to the operation of the computer itself, typically to be used by end-users. Word processors, media players, and accounting software are examples.
10.	How application programs are run on a computer? Application software cannot run independently and need the presence of system software since systems software controls the hardware: CPU, RAM, disk and input/output devices. The following three layers of computer architecture explains an application software is run on computer hardware.

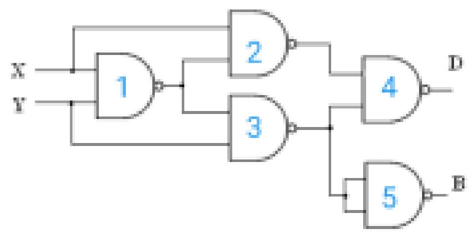


11.	What do you understand by Digital computer? In digital computers, information and data are represented, transferred, processed and saved in binary formats. Logic gates, flip-flops and digital/logic circuits are used to design different sections of computers, specially processing unit and main memory. In digital computers Programs(instructions) and data are converted into two logical forms: ones and zeros The primary memory and other storage devices are designed to hold two states only. The CPU is designed to process data in two states logic. All components are designed using logic gates and digital systems. All devices/components work on two discrete states of voltages/currents/charges etc.



12.	What are **Layers of abstraction?** Layers of abstraction is **a generalization of a conceptual model** of a computer; both hardware and software. For design, analysis, understanding and further development, computer organization, architecture and its operation; from user's application program to electronic signals at the circuits or components are represented by a number of layers in order, shown below  Layers of Abstraction This representation shows how application software is designed. It also shows how the computer hardware is designed from very basic step to user useable form as they appear commercial computers. Functions in application software or programs developed in high level language undergo a several level of conversion and are prepared compatible/useable for the underlying hardware for electronic implementations. Programs in application software or in any high level language is finally converted into binary codes, called machine code. For designing computer hardware, transistors are used as basic/fundamental building blocks.

<table>
<tr><td></td><td>




Half-Subtractor using NAND gates

Transistors are used to design different types of logic gates. Logic gates are used to design digital circuits: adders, multiplexers, flip-flops, decoders, multipliers etc. These are used to design ALU, registers and then the CPU.

Both computer architects and programmers had to invent techniques to make themselves more productive, for otherwise design time would lengthen as dramatically as resources grew. A major productivity technique for hardware and software is to use abstractions to represent the design at different levels of representation; lower level details are hidden to offer a simpler model at higher levels.

• Use layers of abstraction to hide details of the computer design.

• We can work in any layer, not needing to know how the lower layers work or how the current layer fits into the larger system.

Low level: transistors
gates
circuits (adders, multiplexors …)
central processing units (ALU, registers …)
computer

• A component at a higher abstraction layer uses components from a lower abstraction layer without having to know the details of how it is built.

• It only needs to know what it does.
</td></tr>
<tr><td>13.</td><td>
What is layer: instruction set architecture?

An important abstraction is the instruction set architecture or the interface between the hardware and the lowest level software. The abstraction provides a standardized interface between hardware and low-level software.
</td></tr>
<tr><td>14.</td><td>
What are the functional units of a CPU?

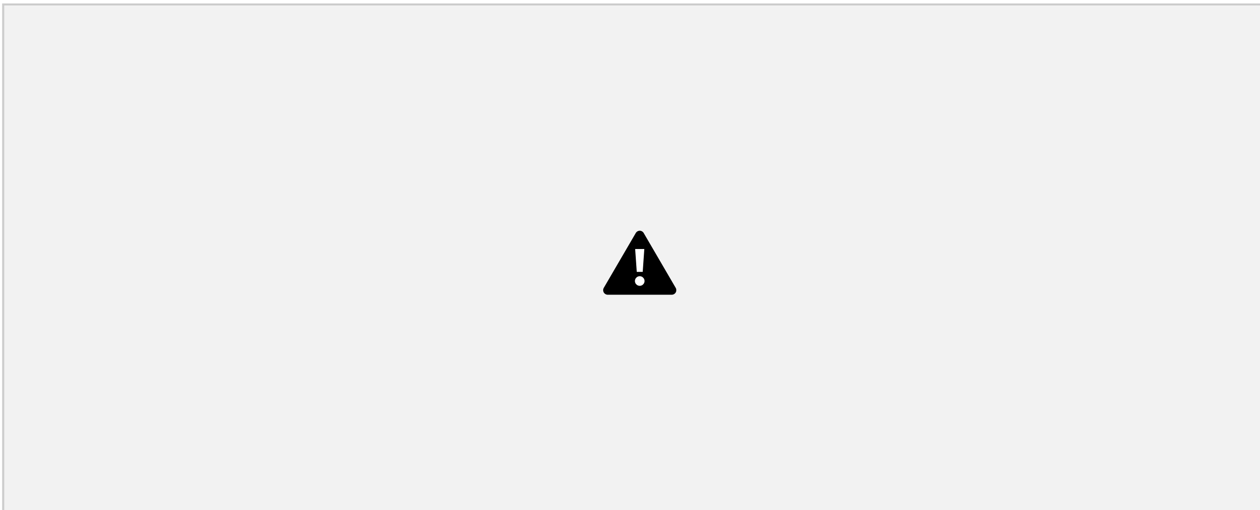
The main functional units of a CPU are:

Registers – high speed electronic storage to hold binary data currently being processed or to be processed next
</td></tr>
</table>

	Arithmetic Logic Unit (ALU) – The ALU can take the contents of two different registers and perform either an arithmetic operation on them (such as Add or Multiply) or a Boolean logic operation on them (such as And or Or). **Control Unit** – This contains the logic circuitry controlling the other parts of the CPU. These functional units are connected through conducting paths, called internal bus
15.	What is a Microprocessor? Microprocessor is an Integrated Circuit (IC) that contains all functional units of a CPU. Nowadays all CPUs are Microprocessors.

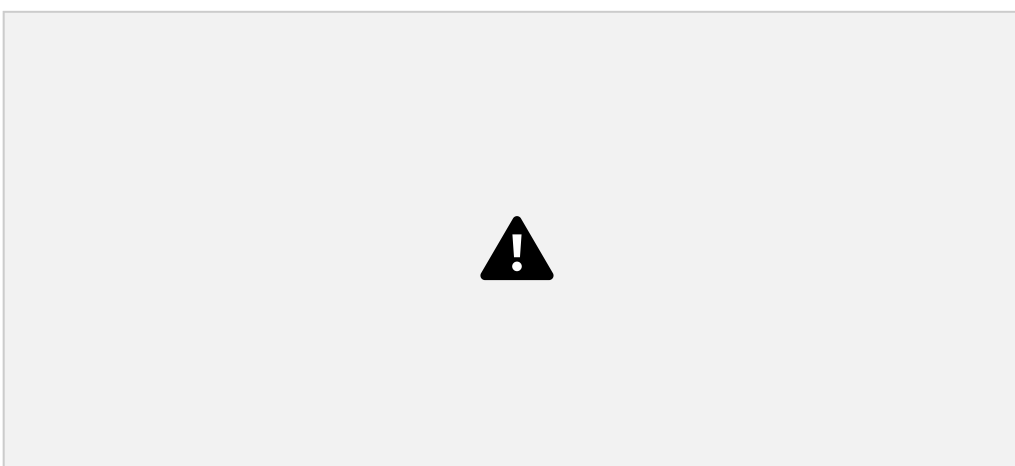
16.	What is Bus system? A bus is a collection of electronic pathway through which data can be transferred. The CPU, memory, and I.O are separate electronic modules that are interconnected by buses. Buses play an essential role in modern computers by connecting all computer parts together with an efficient communication method. Buses enable the subsystems -such as I/O ports, RAM and ROM -to interact with one another by providing fast data transfer rates. Without these pathways – computers simply wouldn't work properly. Buses contain several lines dedicated to communication. Each line carries a specific type of information like data being transmitted, for example, from one memory module to another, or from the CPU to RAM. The number of lines in a bus determines its bandwidth – which relates directly to its speed performance capability (with more lines usually resulting in higher performance). Single bus system: All functional units are connected through a common conducting pathway, as shown below. Although the design seems simpler but the data rate is usually low and it is set in such a way that the slowest device connected to this bus system can operate without loss of data. Slower devices dictate the maximum bus speed. CPU, the faster functional unit of the system, has to remain in long waiting state while

it communicates with RAM and other functional units through the bus system. As a result, utilization of faster functional units are usually very low in a single bus system. If processing speed of all functional units of a system is comparable, a single bus system is preferred.



To transfer different types of information/signals, three types of electronic pathways are provided in bus system and these are: Data bus, Address bus and Control bus. Each bus contains a number of conducting lines and the number of lines decide the size of bus. Sizes of different buses depend on CPU, capacity of RAM and types of I/O devices used in the system. For example, if the size of data bus is 8 bits, it means, the

CPU can transfer 8-bits data at a time. If the size of data bus is 16 bits, it means, the CPU can transfer 16-bits data at a time and the internal operations of that computer will be faster, in general, compared to a computer designed to have 8-bit data bus.

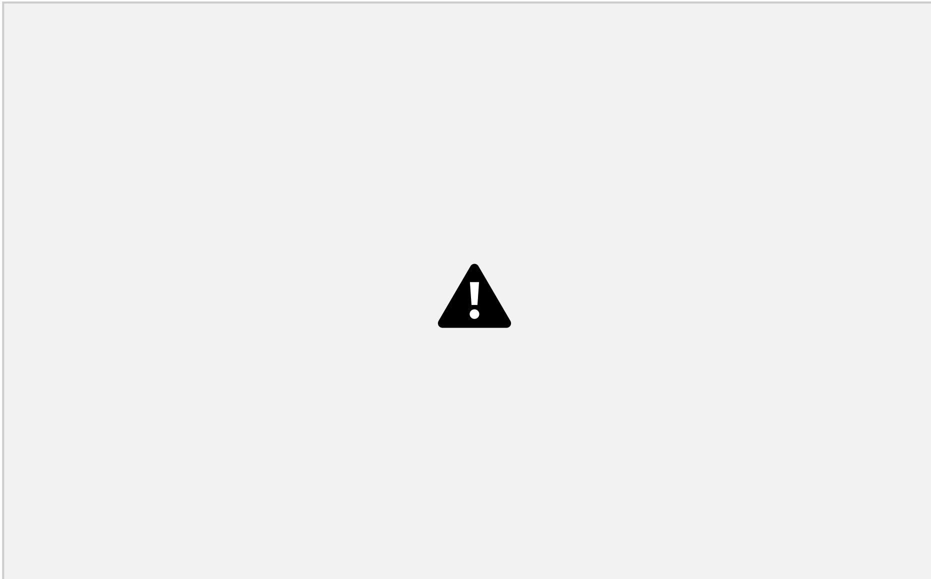


Multi-bus system:

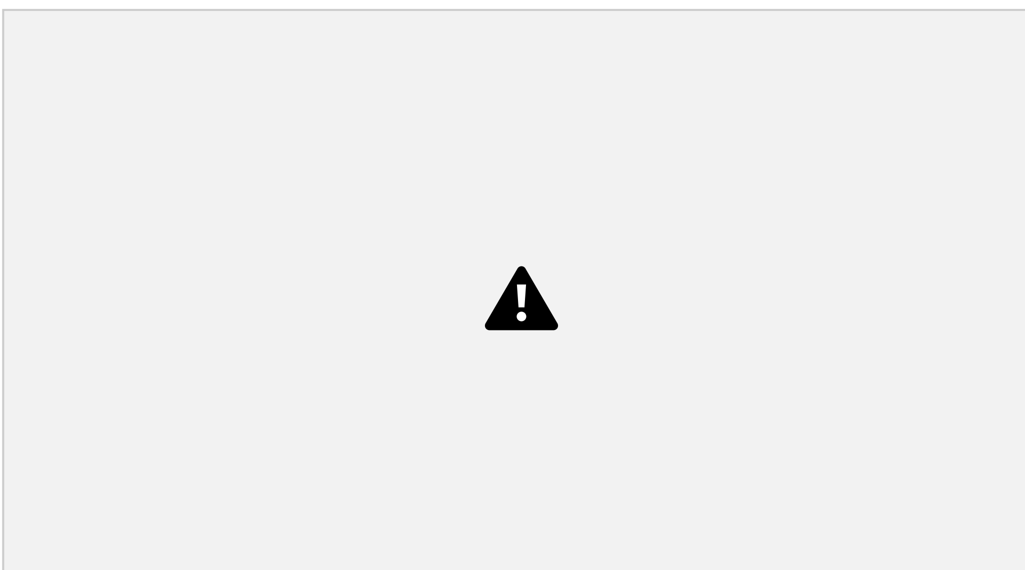
Since different functional units of a computer operate at different speeds due to their design, construction and technology used, multi-bus system allows faster units operate faster and relatively slower units operate at their usual speed without affective faster devices. The interface between CPU and RAM is very crucial

	and it largely decides overall performance of a computer. So in a multi-bus system, a high speed bus is used to

connect semiconductor RAM to CPU whereas a relatively slower I/O bus is used to connect slower I/O devices to the CPU, as shown below.



The most common types of buses include system buses, local buses, and expansion/peripheral buses. System buses are used to transfer data between the CPU and main memory (RAM). They also control the exchange of data between other components such as video cards or sound cards. Peripheral buses are used to connect external devices such as keyboards, mice, and monitors to your computer.

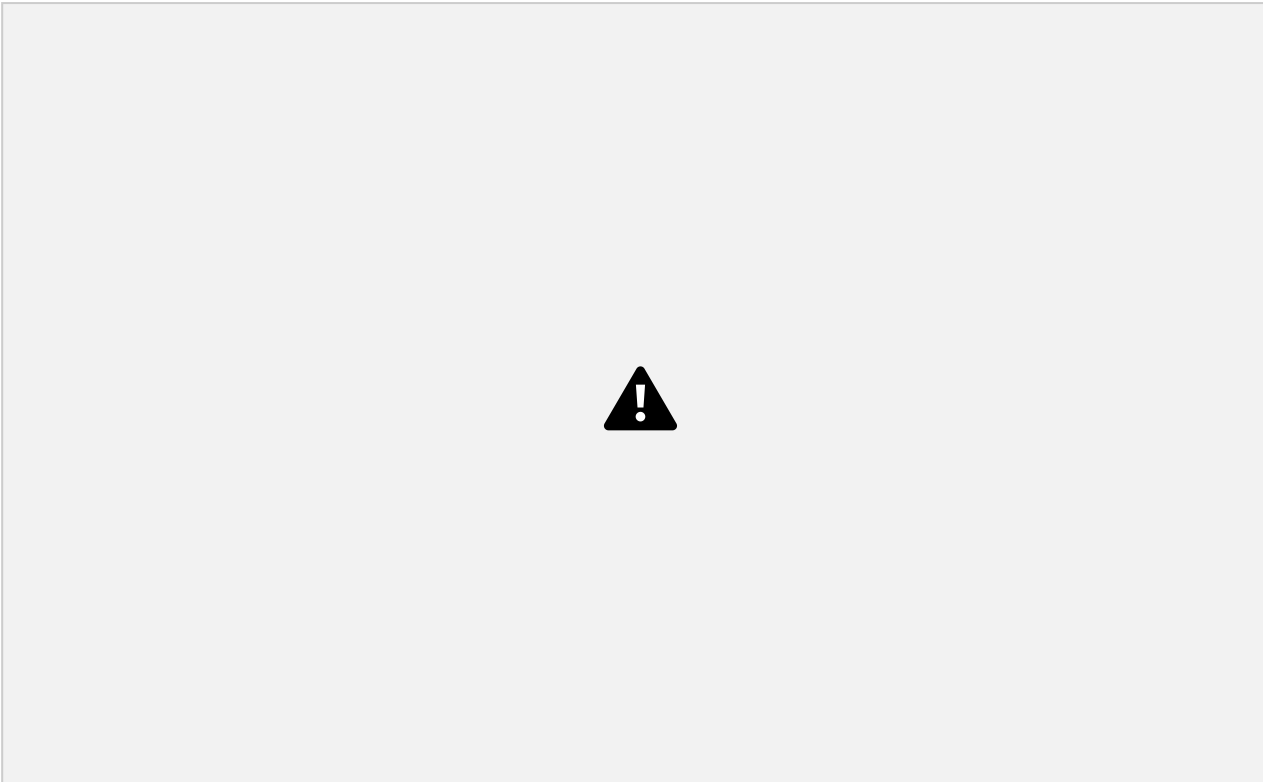


Isolate processor-to-memory traffic from I/O traffic

Support wider variety of interfaces

Processor has bus that connects as direct interface to chip, then an expansion bus interface interfaces it to external devices.

Another example of multi-bus system is shown below:

	 An additional high-speed bus is added to communicate with the faster devices and also the slower expansion bus Advantage is that high-speed devices are brought closer to processor
17.	What do you understand by stored-program computer? Programs (instructions) and data are converted into binary and stored in storage first. When the CPU runs and program, it is loaded into primary memory. The CPU reads instructions from primary memory, decodes it at control unit, reads data from register or memory as instructed and execute it at the ALU. The programs and data can be stored in the storage for long time even if the computer is turned off. The programs can be modified, amended and run anytime and then stored again for future use. This is contrary to earlier computers which were very much like huge trainer board or many trainers boards connected with cables/wires, switches and discrete electrical components. To solve a problem on a computer, skilled technicians and engineers had to work days/weeks to setup cables, wires, components, switches etc. Since problem solving was purely a hardware system design procedure and there was no memory or any technique to save the hardware design for future use unless preserving the whole setup.

18.	What is Control unit? The **control unit** (CU) is a functional unit of central processing **unit** (**CPU**) that decodes machine codes of instructions into electronic signals and executes/implements operations electronically. User programs are converted into binary and stored into storage and then loaded to RAM. CPU fetches machine codes of instructions and stores into Instruction register. The current machine code is decoded into electronic signals at

the control unit and these signals are used to activate different functional units within CPU and external to CPU, such as RAM, I/O devices etc as required by current machine code.

Example:

ADD R0, R1, R2; add the contents of registers R0 and R1 and save the result to R2

Let us assume the machine code of the instruction is as follows:

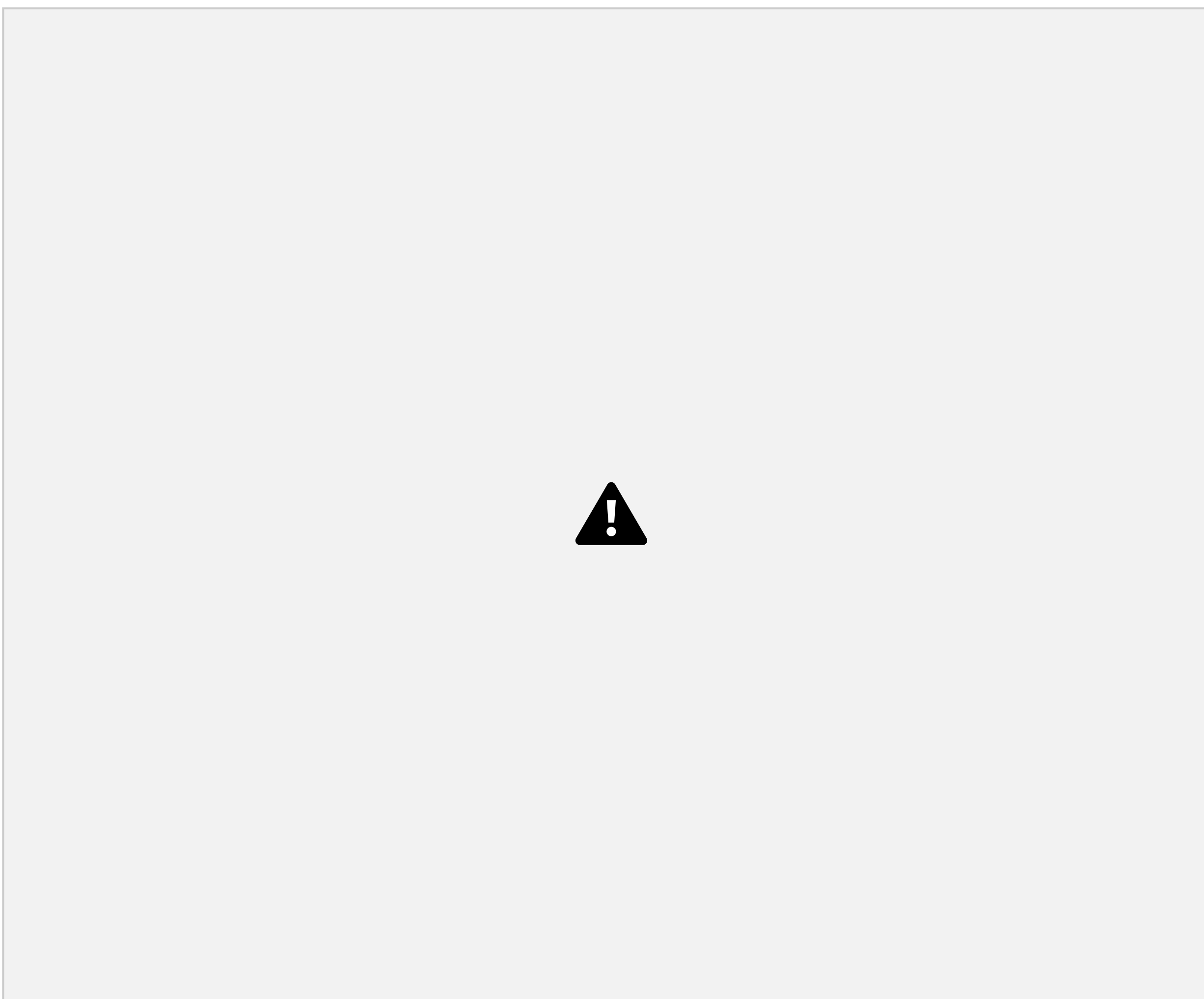
Low level Inst ADD R0 R1 R2 Machine code 0001

0000 0001 0010

(Opcode)

The CPU fetches machine code 0001 0000 0001 0010 and saves to instruction register for decoding.

It selects and retrieves instructions from the main memory in proper sequence and interprets them so as to activate the other functional elements of the system at the appropriate moment.



In the example CPU, there are 16 registers: R0 – R15; only one register is selected at a time using a demultiplexer select bits C6 – C9 and an active high signal at C0. Whenever any register is required to select, register code from machine code of the instruction is sent to C6 – C9 following the activation of C0. The ALU is designed to perform basic arithmetic and logical operations and it contains digital circuits for different types of operations, such as adder, multiplier, divisor, shifter etc. A demultiplexer and control signal C1 is used to select a particular circuit within ALU for an operation. Control signals C2 – C5 are used as select bits of the demultipler. The opcode part of the machine code is sent to C2 – C5 to select particular operation at the ALU following the activation of C1 control signal. In order to sequence the elementary operations (called micro-operations) required to execute an instruction, encoded in machine code, a timing signal, called CLK is used at the input to control unit and all other functional units of CPU. The micro-operations, sequencing and Control signals generated by control unit to implement micro-operation are listed in the following table.

CLK	Micro-operation	Control signals generated by control unit to implement micro-operation
	Select R0	C0 = 1
		C6 = 0
		C7 = 0
		C8 = 0
		C9 = 0
	Select R1	C0 = 1
		C6 = 0
		C7 = 0
		C8 = 0
		C9 = 1
	Select ADDER	C1 = 1
		C2 = 0
		C3 = 0
		C4 = 0
		C5 = 1

The design/format of machine codes and design of the control unit must be consistent. In fact, control unit is designed to activate appropriate control signals required to activate different functional units in a sequence and as required by the machine code.