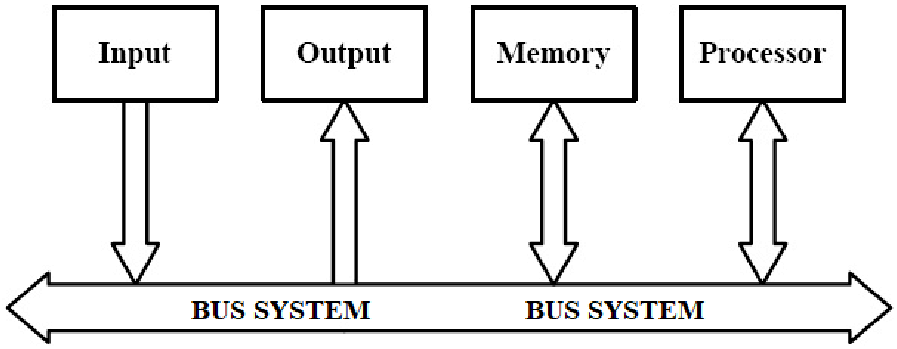
Questions and answers: Lecture-1

1.	Show a simple block diagram of a computer and identify its main functional units.  Input Output Memory Processor BUS SYSTEM BUS SYSTEM
2.	What technology is used to design main functional units of a computer? Semiconductor Technology
3.	What do you understand by Digital technology? In digital technology, systems are designed to work with two-states logic; called High/low or 1/0 or true/false, although in electronic implementation two different voltage levels or ranges are used (example: +5v for High, 0v for Low) to represent two logic states. Logic gates, such as AND, OR, NOT, and Flip-flops, decoders, encoders, multiplexers etc are used to design systems in digital technology.
4.	What do you understand by digital computer? Digital computers are designed using digital technology. Programs and data are represented by two-states logic, means using 1's and 0's. Memory and storage devices are designed to store information and programs as a string of 1's and 0's. CPU is designed to process data/information (arithmetic, logical operations) in binary. Information, programs and commands also flow/transferred among different functions units in binary forms as well. Logic gates, flip-flops and other digital logic systems are used to design all functional units of a digital computer.
5.	What is the basic element in CPU design? Transistor, a three terminal solid-state electronic device, is used as the basic building of CPU. Logic gates, flip-flops and other digital logic systems are designed using transistors. ON and OFF states of transistors represent two logic states.
6.	How arithmetic operations are performed in digital computers. Numbers are encoded in binary and arithmetic operations are performed/processed on binary representation of data.
7.	What are the different programming levels? High level language ----- low level (assembly) language ----- machine code. Programming in High level language does not require any knowledge on hardware rather it is independent of CPU. English words, arithmetic notations and variables are used to write programs following the syntax of the language. Programming in Low level language requires detailed information/knowledge on hardware: CPU architecture and instruction set. In Machine code, programs and data are represented in Binary strings of 1's and 0's and it also requires detailed information/knowledge on hardware
8.	What is an Instruction? An instruction is a command to CPU for a basic operation. For example: add two numbers and store result to memory/another storage.

An instruction in low level language must contain information on

- Operation to be performed
- Data of sources of data
- Result field

Other information (size of data, source of data etc) for the CPU

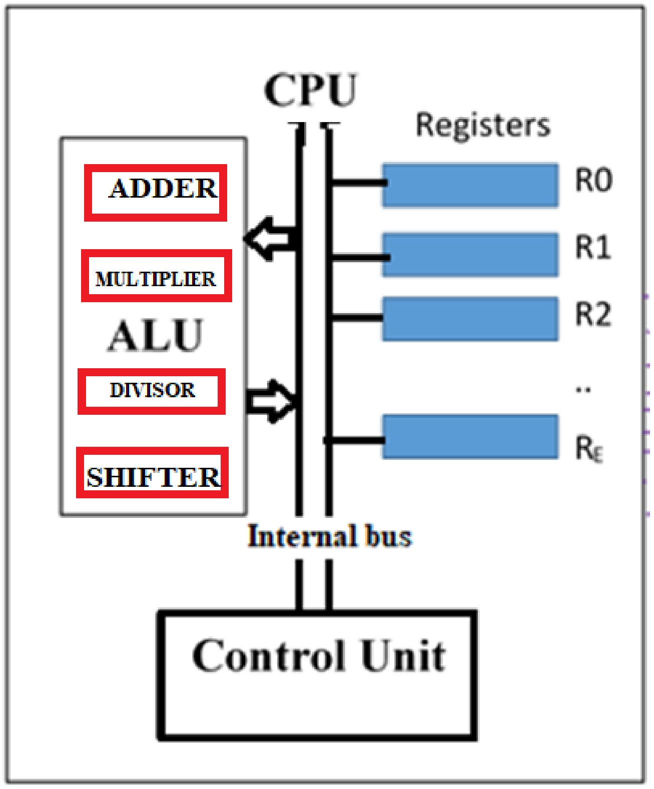
9. Show instruction format in low level language.

Operation field	Result field	Data -1 Source of data-1	Data -2 Source of data-2

10. What is machine code?
Machine code is binary representation of an instruction in low level language.
A machine code must contain all information of an instruction written in low level language but represented in binary. Moreover, machine code may contain more information required for the CPU.
A typical machine code: 0011010101111000011100

11. What is Instruction set?
A CPU is designed to perform only a limited numbers of basic operations and separate commands or instructions are used for basic operations. Instruction set refers to all instructions of a CPU.

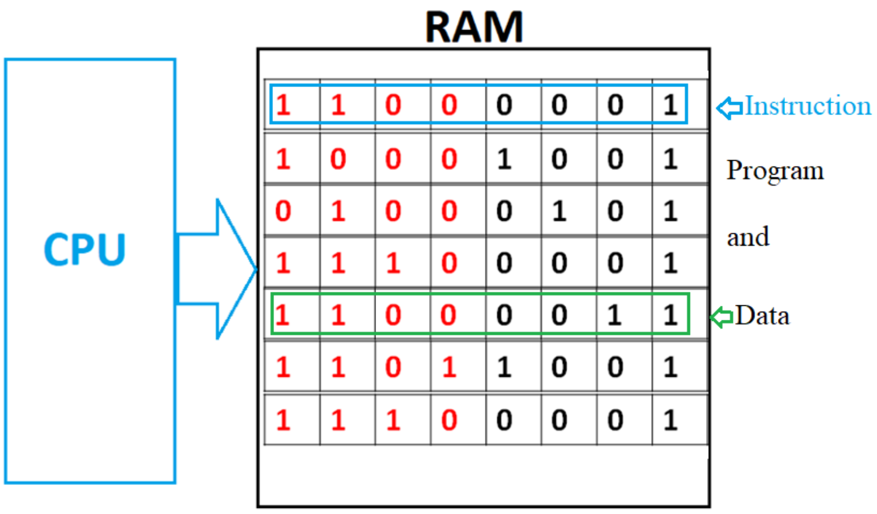
12. Show internal architecture of a CPU.

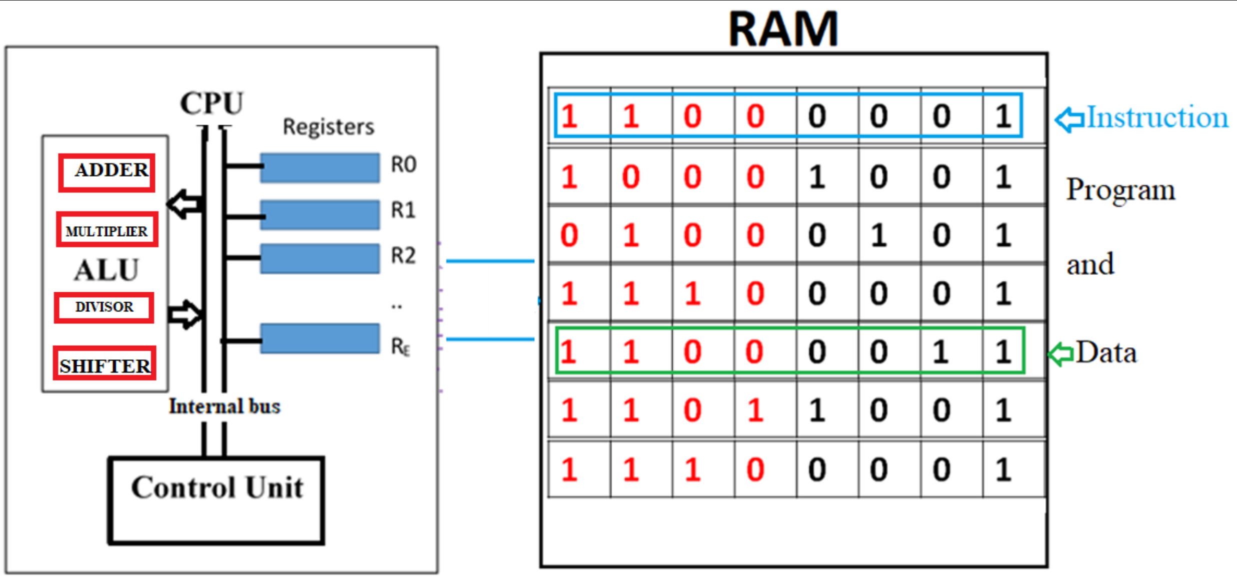


13. What is a register?
A register is a high speed electronic storage within the CPU? There are a number of registers in a CPU.

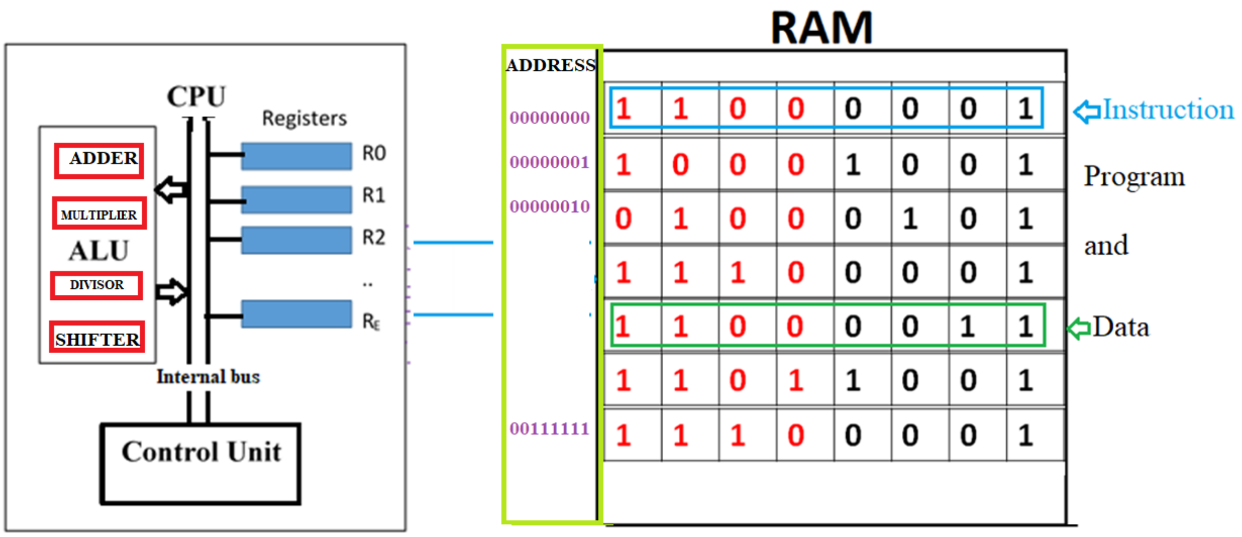
14. What is the use of register?
CPU uses registers to hold data for arithmetic/logical operations. Registers are also used to store results.

15. How registers are used in low level language?

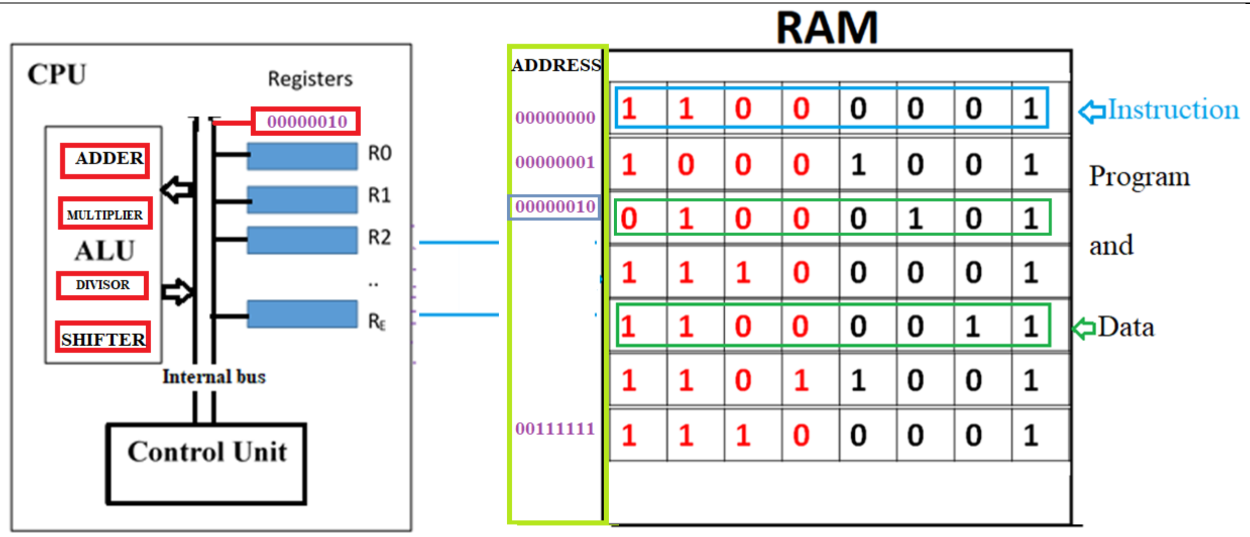
	Registers are only few in numbers, may be 8, 16, 32 so. Registers are addressed by names in low level language as assigned by CPU designers. Usually English letters and subscripts are used, for example R0, R1….. or S0, S1,….. AX, BX….. etc.
16.	How registers are used in machine code? By specific codes, containing 1's and 0's.
17.	What is control unit? Control unit is designed to decode machine codes of instructions and control all other sections of CPU and computer in electronic form.
18.	What is ALU? ALU is a functional unit of a CPU that is designed to perform all arithmetic and logical operations. ALU contains digital circuits to perform basic operations.
19.	How programs and data are stored in a digital computer? Program and Data are converted into a binary strings of 1's and 0's and stored in storage (hard disk/secondary storage). When a program is intended to run, that program and data are loaded to RAM. CPU reads binary coded instructions and data from RAM and process.  RAM CPU 1 1 0 0 0 0 0 1 1 0 0 0 1 0 0 1 0 1 0 0 0 1 0 1 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 1 0 1 1 0 0 1 1 1 1 0 0 0 0 1 Instruction Program and Data
20.	Explain the basic computer function Machine codes of instructions of a program are loaded to RAM in consecutive locations, starting from an address which is either set by user or by the operating system.



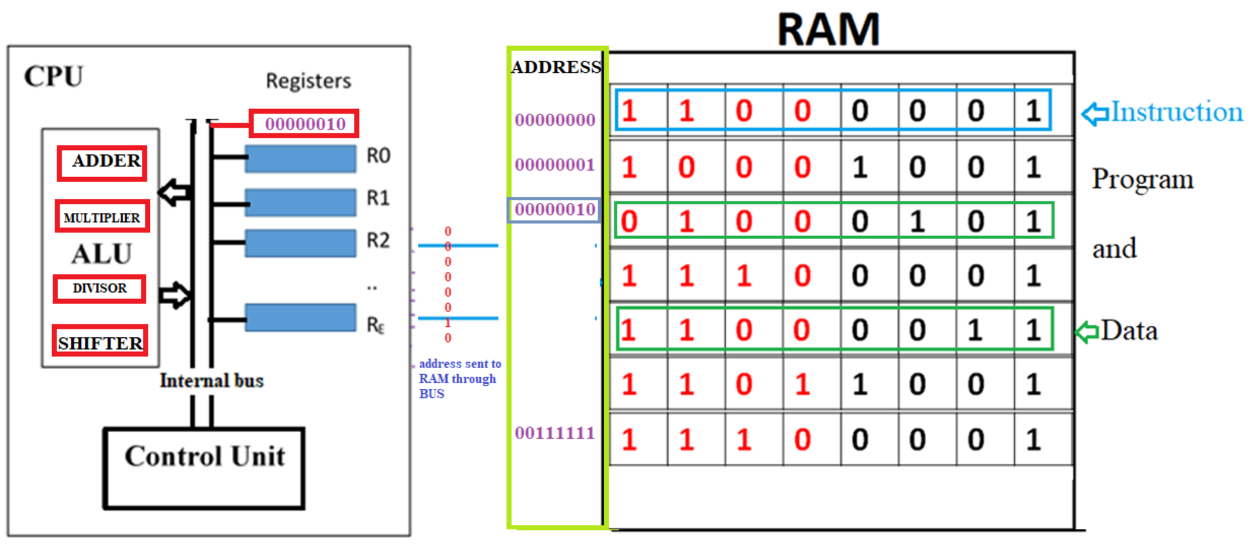
CPU will access RAM and read machine code and data using memory address



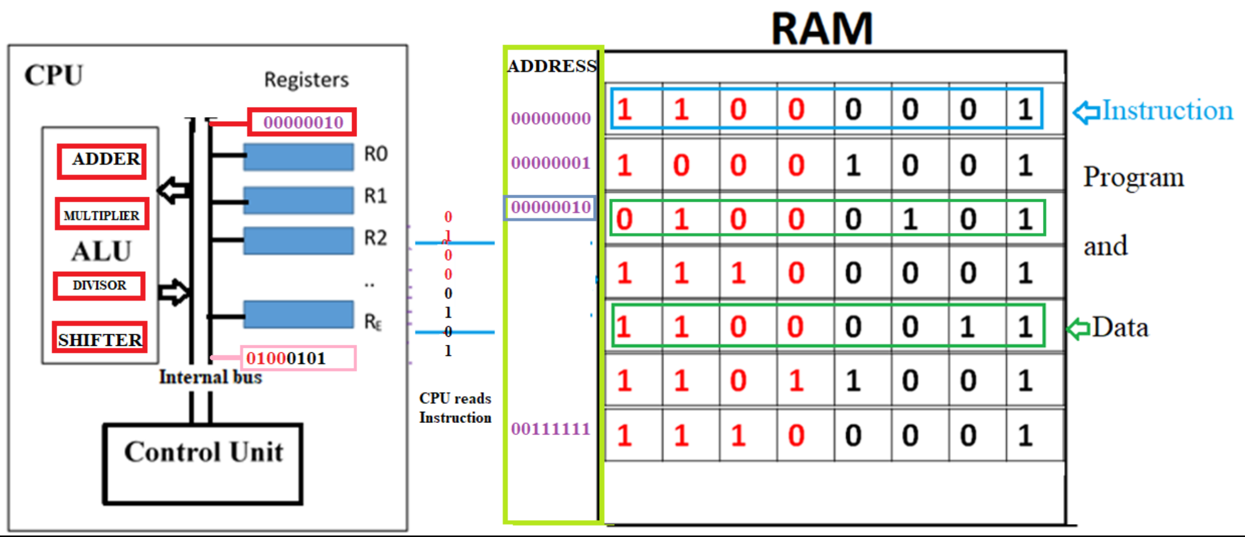
CPU has a special register called Instruction pointer or program counter that points instruction in RAM to be read



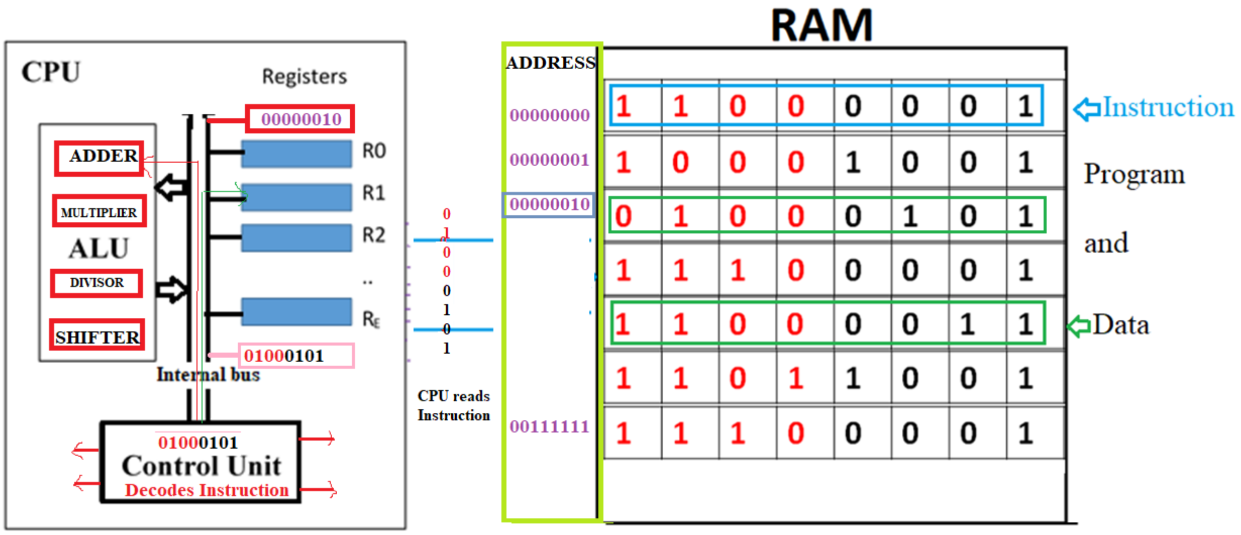
CPU sends address of instruction to RAM



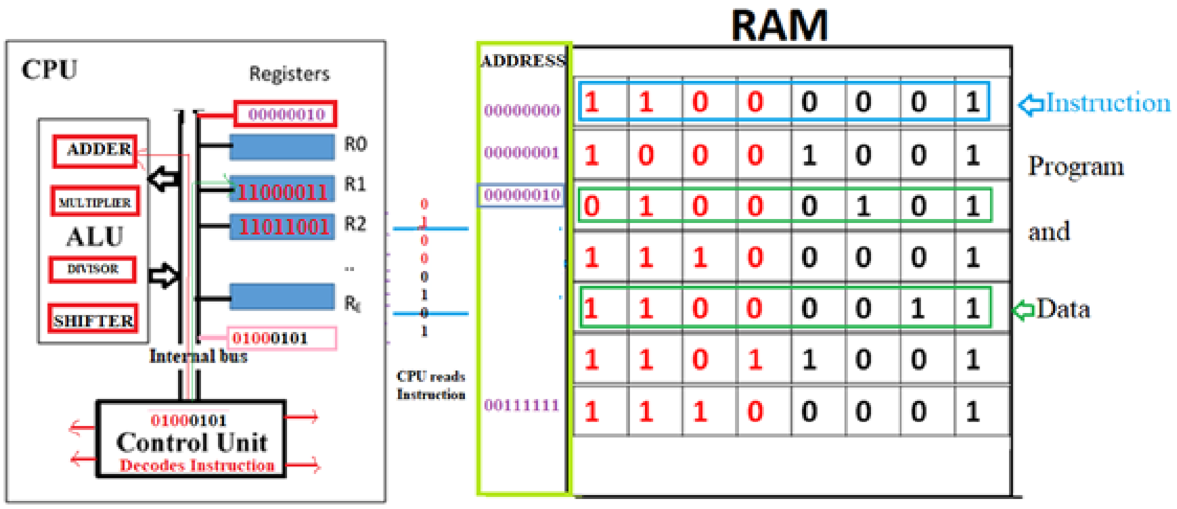
CPU reads Instructions from RAM, it means the machine code is copied to a register



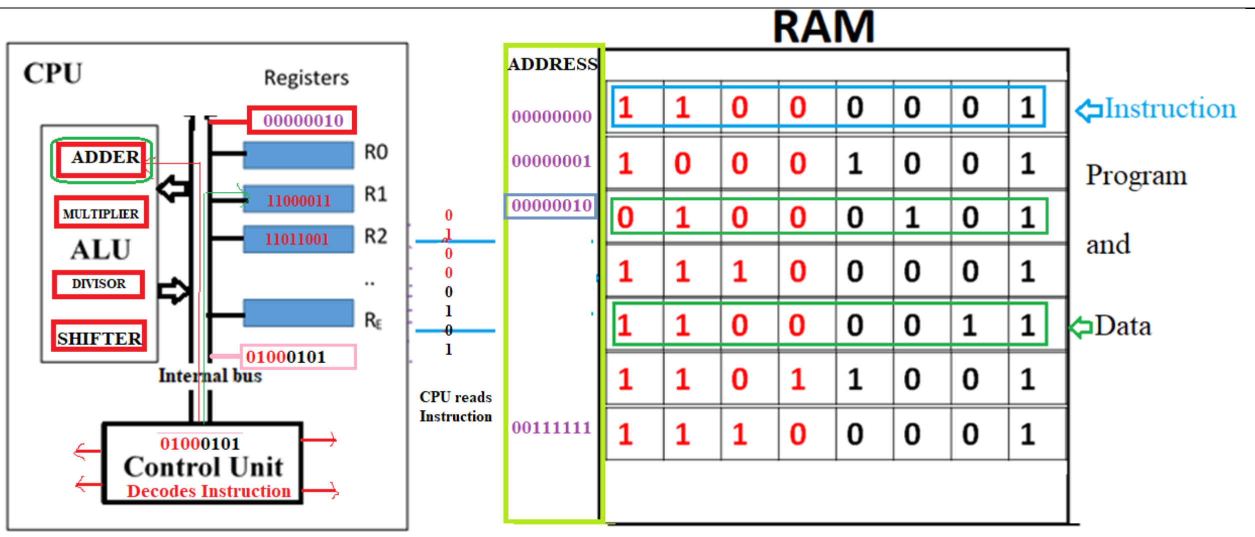
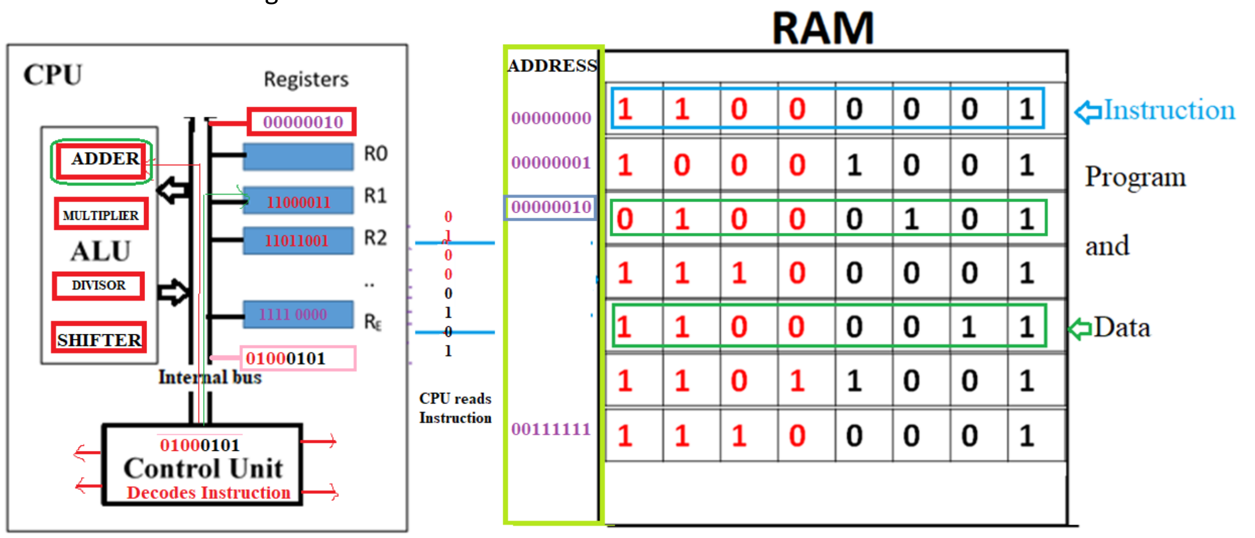
The machine code is decoded at the control unit. Decode means, the control unit will generate a sequence of electronic signals. These signals will select circuit within ALU, activate registers so that contents of registers (data) are transferred to ALU, read data from RAM if required etc.

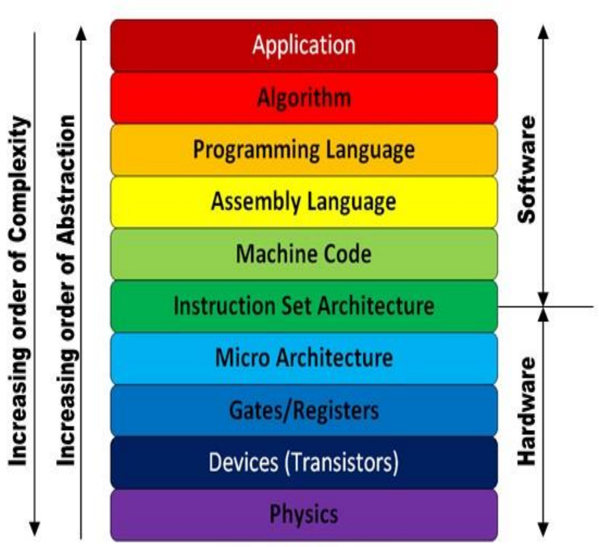


CPU reads data and stores in registers (if data are initially stored in RAM)



CPU performs arithmetic/logical operations as indicated in the instruction

	 RAM CPU Registers 00000010 R0 11000011 R1 11011001 R2 RE 01000101 ADDER MULTIPLIER ALU DIVISOR SHIFTER Internal bus 01000101 Control Unit Decodes Instruction CPU reads Instruction ADDRESS 00000000 00000001 00000010 00111111 1 1 0 0 0 0 0 1 1 0 0 0 1 0 0 1 0 1 0 0 0 1 0 1 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 1 0 1 1 0 0 1 1 1 1 0 0 0 0 1 Instruction Program and Data CPU stores result to register  RAM CPU Registers 00000010 R0 11000011 R1 11011001 R2 1111 0000 RE 01000101 ADDER MULTIPLIER ALU DIVISOR SHIFTER Internal bus 01000101 Control Unit Decodes Instruction CPU reads Instruction ADDRESS 00000000 00000001 00000010 00111111 1 1 0 0 0 0 0 1 1 0 0 0 1 0 0 1 0 1 0 0 0 1 0 1 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 1 0 1 1 0 0 1 1 1 1 0 0 0 0 1 Instruction Program and Data Instruction pointer or program counter is incremented to point next instruction in RAM and preceding steps are repeated.
21.	What is Instruction cycle? A program, written in high level language, is compiled. The compiler will generate a list of instructions in a sequence. An instruction is a command to CPU for a basic arithmetic, logical or data transfer operation. When a program is intended to run, the instructions are loaded to RAM in consecutive locations. The CPU is designed to perform a list of tasks in a sequence to process each instruction and these steps are repeated to process each instruction. The tasks are listed sequentially below: 1. Instruction Fetch: CPU reads machine code of instruction from RAM 2. Decode: Instruction is decoded at control unit. As a result electronic signals are generated in a sequence to activate other functional parts/units of CPU and computer, as required. 3. Execution: Operation is performed electronically with ALU. For data transfer operation, data is transferred from source location to destination as specified in instruction. 4. Store the result to register (in case of ALU operation).

	To process next instruction, these tasks are repeated. Instruction cycle refers to this repetitive/cyclic process, indicated in tasks 1 to 4.
22.	What do you represent computer in **Layers of abstraction? What is the benefit of this?** **Design stages of computer hardware and levels of software from user to machine are shown by a number of layers for ease of representation, called layers of abstraction.**  Application Algorithm Programming Language Assembly Language Machine Code Instruction Set Architecture Micro Architecture Gates/Registers Devices (Transistors) Physics Increasing order of Complexity Increasing order of Abstraction Software Hardware Layers of Abstraction 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. transistors gates circuits (adders, multiplexors …) central processing units (ALU, registers …) computer hardware 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.