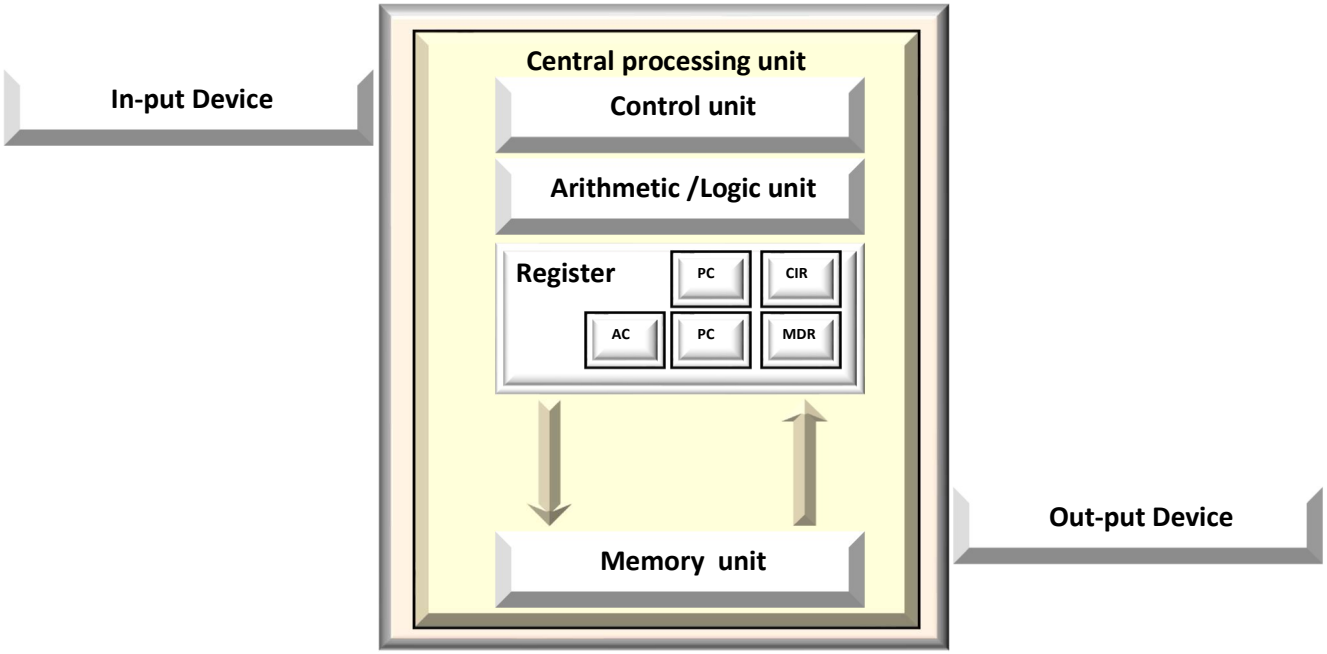
BCA 3RD SEMSESTER

DCA2103, Computer Organization

(set:- I)

Question:- Explain von NNeumann'sArchitecture in detail.

Answer:- Von Neumann's architecture was first published by John von Neumann in 1945. His computer architecture design consists of a Control Unit, Arithmetic and Logic Unit (ALU), Memory Unit, Registers, and Inputs/Outputs. Von Neumann's architecture is based on the stored-program computer concept, where instruction data and program data are stored in the same memory. This design is still used in most computers produced today.



➔ Central Processing Unit (CPU)

The Central Processing Unit (CPU) is the electronic circuit responsible for executing the instructions of a computer program. It is sometimes referred to as the microprocessor or processor. The CPU contains the ALU, CU, and a variety of registers.

➔ Registers

Registers are high-speed storage areas in the CPU. All data must be stored in a register before it can be processed.

MAR	Memory Address Register	Holds the memory location of data that needs to be accessed
MDR	Memory Data Register	Holds data that is being transferred to or from memory

AC	Accumulator	Where intermediate arithmetic and logic results are stored
PC	Program Counter	Contains the address of the next instruction to be executed
CIR	Current Instruction Register	Contains the current instruction during processing

➔ Arithmetic and Logic Unit (ALU)

The ALU allows arithmetic (add, subtract, etc) and logic (AND, OR, NOT, etc) operations to be carried out.

➔ Control Unit (CU)

The control unit controls the operation of the computer's ALU, memory, and input/output devices, telling them how to respond to the program instructions it has just read and interpreted from the memory unit.

The control unit also provides the timing and control signals required by other computer components.

➔ Buses

Buses are how data is transmitted from one part of a computer to another, connecting all major internal components to the CPU and memory.

A standard CPU system bus is comprised of a control bus, a data bus, and an address bus.

Address Bus *Carries the addresses of data (but not the data) between the processor and memory*

Data Bus	Carries data between the processor, the memory unit, and the input/output devices
Control Bus	Carries control signals/commands from the CPU (and status signals from other devices) to control and coordinate all the activities within the compute

➔ Memory Unit

The memory unit consists of RAM, sometimes referred to as primary memory. Unlike a hard drive (secondary memory), this memory is fast and also directly accessible by the CPU.

RAM is split into partitions. Each partition consists of an address and its contents (both in binary form).

The address will uniquely identify every location in the memory.

Loading data from permanent memory (hard drive), into the faster and directly accessible temporary memory (RAM), allows the CPU to operate much quicker.

2.) Question:- Explain in detail the different instruction formats with examples.

Answer:- An instruction format defines the layout of bits of instruction in terms of constituent parts. There are various instruction formats depending upon the architecture of the computer. The types of commonly used instructions are:-

- Three-Address Instruction.
- Two-Address Instruction.
- One-Address Instruction.
- Zero-Address Instruction.

➔ Three-Address Instruction:

A three-address instruction consists of the following parts: Operation code.

Addresses of two operands called address 1 and address 2.

Address of the memory location where the result of the operation is to be stored i.e., address of the destination. The number of bits (field length) allocated to each of the three parts depends upon the computer. A typical three-address instruction is shown below.

OP code **Address 1** **Address 2** **Address of Destination**

A register called program counter (PC) is used to compute the address of the next instruction and the process of the execution of the instruction continues.

➔ **Two Address Instruction:**

In this type of instruction one operand is placed in a specified register such as an accumulator and the address of the next instruction is obtained from another register called program counter (PC). This implies that such an instruction should have the following parts.

- Operation code.
- Address of one of the operands, say address 1.
- Address of the storage location where the result is to be stored. This address is denoted by address 2.

The general form of a two-address instruction is

OP code Address 1 Address 2

➔ **One Address Instruction:**

As the name suggests, this instruction has the address of one operand only, the other operand is stored in an accumulator. The result of the operation they is left in the accumulator itself, from where these can be moved to main memory by another instruction. The address of the next instruction is obtained from the program counter. The general form of a one-address instruction is

OP code Address 1

No doubt with one address instruction more bits of the instruction word could be allocated to OP code and the address of the operand.

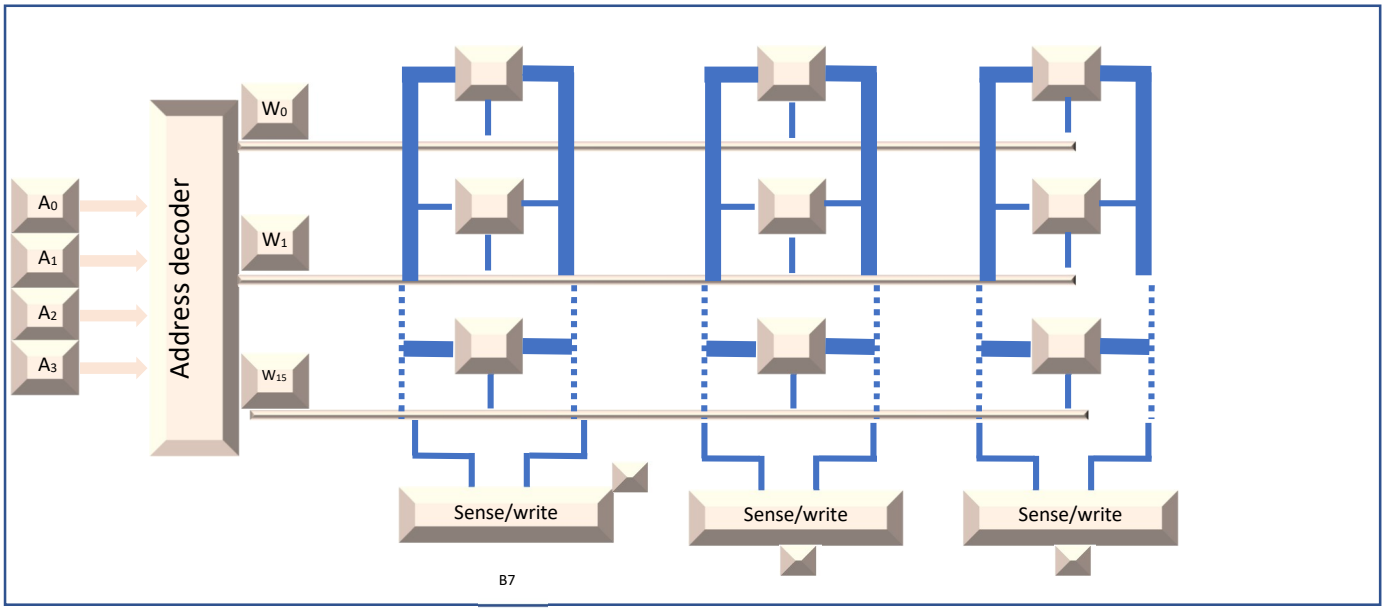
➔ **Zero Address Instruction:**

The zero address instructions are also called stack instructions and consist of OP code only. The address of operand and destination is implied. The general form of zero address instruction is.

OP code

3.) Question:- Discuss the organization of the main memory.

Answer:- Organizational memory is the collective ability to accumulate, store, and retrieve knowledge and data. It is also sometimes called institutional or corporate memory.



The memory unit that communicates directly within the CPU, Auxiliary memory, and Cache memory, is called main memory. It is the central storage unit of the computer system. It is a large and fast memory used to store data during computer operations. Main memory is made up of **RAM** and **ROM**, with RAM integrated circuit chips holding the major share.

- RAM: Random Access Memory
    - **DRAM**: Dynamic RAM, is made of capacitors and transistors, and must be refreshed every 10~100 ms. It is slower and cheaper than SRAM.
    - **SRAM**: Static RAM, has a six transistor circuit in each cell and retains data until powered off.
    - **NVRAM**: Non-Volatile RAM, retains its data, even when turned off. Example: Flash memory.
- ROM: Read Only Memory, is non-volatile and is more like permanent storage for information. It also stores the **bootstrap loader** program, to load and start the operating system when the computer is turned on. **PROM**(Programmable ROM), **EPROM**(Erasable PROM), and **EEPROM**(Electrically Erasable PROM) are some commonly used ROMs

SET:- II

4.) Question:- List and explain the mapping functions.
Answer:-

A function is a special type of relation in which each element of the domain is paired with another element in the range. This pairing can be shown on a Mapping Diagram.
It is similar to a flow chart for a function, showing the input and output values.

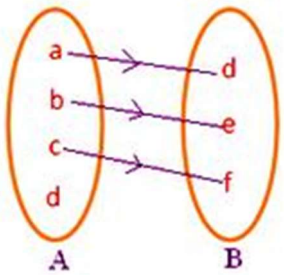
➔Mapping Function

Let us see the mapping diagram function and function mapping in a little more depth now.
If A and B are two non-empty sets, then a relation from set A to set B is said to be a function or mapping, or mapping function.

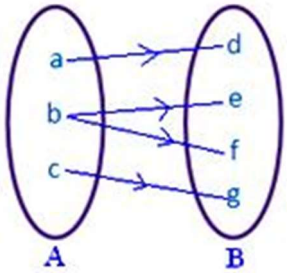
- If every element of set A is associated with a unique element of set B.
- Function " f " from A to B is denoted by f: A → B.
- If " f " is a function from A to B and x ∈ A, then f (x) ∈ B and f (x) is called the image of x under f, and x is called the preimage of f (x) under "F".

Note:

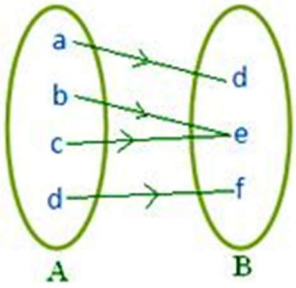
- Every element of A must have an image in B. Adjoining figure does not constitute a mapping since the element d in set A is not associated with any element of set B.



- No one element of A must have more than one image in the mapping function.



Different elements of A have the same image in B.
The adjoining figure represents a mapping function.



A and B are two non-empty sets, then a rule ' f ' that associates each element of A with a different element of B is called a function or a mapping from A to B.
Whether ' f ' is a mapping from A to B,
we express it as f: A → B
We can read it as ' f ' is a function from A to B.
If " f " is a function from A to B and x ∈ A and y ∈ B, then we can say that y is the image of element x under the function ' f ' and denote it by f (x).
Accordingly, we write it as y = f (x)
Here, element x is called the preimage of y.
Thus, for a function from A to B.

- A and B should be non-empty.
- Each element of A should have an image in B.
- No one element of ' A ' should have more than one image in ' B '.

5.) Question :- Wtrre4rhat is an interrupt? Discuss the hardware actions in interrupt handling.

Answer :- Alternatively referred to as a **maskable interrupt** or **trap**, an **interrupt** is sometimes abbreviated as **INTR** and is a command or request. It tells the processor to stop what it's doing and wait for further instructions.

If there is an interrupt present then it will trigger the interrupt handler, the handler will stop the present instruction which is processing and saving its configuration in a register, and load the program counter of the interrupt from a location which is given by the interrupt vector table. After processing the interrupt by the processor interrupt handler will load the instruction and its configuration from the saved register, process will start its processing where it's left. This saving the old

instruction processing configuration and loading the new interrupt configuration is also called context switching.

The interrupt handler is also called an Interrupt service routine (ISR). There are different types of an interrupt handlers that will handle different interrupts. For example, the clock in a system will have its interrupt handler, the keyboard it will have its interrupt handler for every device will have its interrupt handler. The main features of the ISR are

- Interrupts can occur at any time they are asynchronous. ISRs can call for asynchronous interrupts.
- Interrupt service mechanisms can call the ISRs from multiple sources.
- ISRs can handle both maskable and nonmaskable interrupts. An instruction in a program can disable or enable an interrupt handler call.
- ISR at beginning of execution will disable other devices and interrupt services. After completion of the ISR execution, it will re-initialize the interrupt services.
- The nested interrupts are allowed in ISR for diversion to other ISR.

6.) Question:- Explain the characteristics of RISC and CISC architectures.

Answer:-

➔ RISC

A reduced instruction set computer is a computer that only uses simple commands that can be divided into several instructions that achieve low-level operation within a single CLK cycle, as its name proposes “Reduced Instruction Set”. The RISC is a Reduced Instruction Set Computer microprocessor and its architecture includes a set of instructions that are highly customized. The main function of this is to reduce the time of instruction execution by limiting as well as optimizing the number of commands. So each command cycle uses a single clock cycle where every clock cycle includes three parameters namely fetch, decode & execute.

⇨ **RISC Architecture**:- The term RISC stands for ‘’Reduced Instruction Set Computer’’. It is a CPU design plan based on simple orders and acts fast. This is a small or reduced set of instructions. Here, every instruction is expected to attain very small jobs. In this machine, the instruction sets are modest and simple, which helps in comprising more complex commands. Each instruction is of a similar length; these are wound together to get compound tasks done in a single operation. Most commands are completed in one machine cycle. This pipelining is a crucial technique used to speed up RISC machines.

⇨ **Characteristics of RISC**

- Pipeline architecture
- The number of instructions is restricted as well as decrease
- The instructions like load as well as store have right of entry to memory
- Addressing modes are less
- Instruction is uniform and its format can be simplified

➔CISCOSC

It was developed by the Intel Corporation and it is Complex Instruction Set Computer. This processor includes a huge collection of simple to complex instructions. These instructions are specified at the level of assembly language level and the execution of these instructions takes more time.

A complex instruction set computer is a computer where single instructions can perform numerous low-level operations like a load from memory, an arithmetic operation, and a memory store or are accomplished by multi-step processes or addressing modes in single instructions, as its name proposes “Complex Instruction Set ”.

So, this processor moves to decrease the number of instructions on every program & ignores the number of cycles for each instruction. It highlights to assembling complex instructions openly within the hardware as the hardware is always as compared with software. However, CISC chips are relatively slower as compared to RISC chips but utilize small instructions as compared to RISC. The best examples of the CISC processor include AMD, VAX, System/360 & Intel x86.

⇨ CISC Architecture

The term CISC stands for ‘’Complex Instruction Set Computer’’. It is a CPU design plan based on single commands skilled in executing multi-step operations.

CISC computers have small programs. It has a huge number of compound instructions, which take a long time to perform. Here, a single set of instructions is protected in several steps; each instruction set has additional than 300 separate instructions. Maximum instructions are finished in two to ten machine cycles. In CISC, instruction pipelining is not easily implemented.

⇨ Characteristics *of CISC*

The main characteristics of the RISC processor include the following.

- CISC may take more time to execute the code as compared with an only clock cycle.
- CISC supports high-level languages for simple compilation and complex data structure.
- It is collected with more addressing nodes and fewer registers normally from 5 to 20.
- For writing an application, less instruction is required
- The code length is very short, so it needs extremely small RAM.

- It highlights the instruction on hardware while designing as it is faster to design than the software.
- Instructions are larger as compared with a single word.
- It gives simple programming within assembly language.