COAA VIVA QUESTIONS: -

**Section 1:-**

**1. Define the term Computer Architecture.**

**Computer Architecture – is the abstract model & are those attributes which are visible to programmer like instruction sets, no of bits used for data addressing techniques.**

**i.e. abstract, programmer view.**

**¨Computer architecture refers to those attributes of a system visible to a programmer or put another way, those attributes that have direct impact on the logical execution of a program.**

**2. Define Multiprocessing.**

**Multiprocessing is the coordinated processing of** [**program**](https://searchsoftwarequality.techtarget.com/definition/program)**s by more than one computer processor. Multiprocessing is a general term that can mean the dynamic assignment of a program to one of two or more computers working in tandem or can involve multiple computers working on the same program at the same time (in parallel).**

**3. What is meant by instruction?**

**Computer instructions are a set of machine language instructions that a particular processor understands and executes. A computer performs tasks on the basis of the instruction provided.**

**An instruction comprises of groups called fields. These fields include:**

* **The Operation code (Opcode) field which specifies the operation to be performed.**
* **The Address field which contains the location of the operand, i.e., register or memory location.**
* **The Mode field which specifies how the operand will be located.**

**Computer Instructions**

**A basic computer has three instruction code formats which are:**

1. **Memory - reference instruction**
2. **Register - reference instruction**
3. **Input-Output instruction**

**4. What is Bus? Draw the single bus structure.**

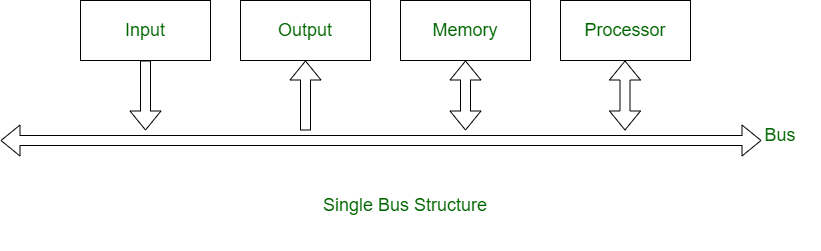
**• A bus is a group of lines that serves as a connecting path for several devices.**

**• A bus may be lines or wires.**

**• The lines carry data or address or control signal.**

**1. Single Bus Structure :**

**In single bus structure, one common bus used to communicate between peripherals and microprocessor. It has disadvantages due to use of one common bus.**

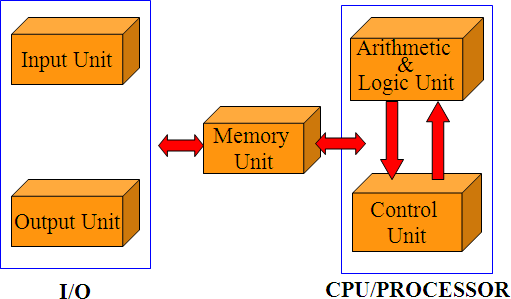
****

**5. Define Pipeline processing.**

**Pipelining is a technique where multiple instructions are overlapped during execution. Pipeline is divided into stages and these stages are connected with one another to form a pipe like structure. Instructions enter from one end and exit from another end.**

**Pipelining is the process of accumulating instruction from the processor through a pipeline. It allows storing and executing instructions in an orderly process. It is also known as pipeline processing.**

**6. Draw the basic functional units of a computer.**

****

**7. Briefly explain Primary storage and secondary storage.**

**Data storage is a common term for archiving data or information in a storage medium for use by a computer. It’s one of the basic yet fundamental functions performed by a computer.**

* **Primary storage is commonly referred to as simply “primary memory” which is volatile in nature such as the RAM which is a primary memory and tends to lose data as soon as the computer reboots or loses power.**
* **It’s a high-speed data storage medium which is directly connected to the processing unit via the memory bus, allowing active programs to interact with the processor.**
* **Simple speaking, primary storage refers to internal storage devices that provide fast and efficient access to data or information.**

* **Secondary storage is yet another ideal storage solution in the computer’s memory hierarchy that is used to store data or information on the long term basis, more like permanently.**
* **Unlike primary storage, they are non-volatile memory or commonly referred to as external memory that are not directly accessed by the central processing unit. They are also called as auxiliary storage which can be both internal and external, plus beyond the primary storage.**

|  |  |  |
| --- | --- | --- |
| Sr.No. | Primary memory | Secondary memory |
| **1.** | **Primary memory is temporary.** | **Secondary memory is permanent.** |
| **2.** | **Primary memory is directly accessible by Processor/CPU.** | **Secondary memory is not directly accessible by the CPU.** |
| **3.** | **Nature of Parts of Primary memory varies, RAM- volatile in nature. ROM- Non-volatile.** | **It’s always Non-volatile in nature.** |
| **4.** | **Primary memory devices are more expensive than secondary storage devices.** | **Secondary memory devices are less expensive when compared to primary memory devices.** |
| **5.** | **The memory devices used for primary memory are semiconductor memories.** | **The secondary memory devices are magnetic and optical memories.** |
| **6.** | **Primary memory is also known as Main memory or Internal memory.** | **Secondary memory is also known as External memory or Auxiliary memory.** |
| **7.** | **Examples: RAM, ROM, Cache memory, PROM, EPROM, Registers, etc.** | **Examples: Hard Disk, Floppy Disk, Magnetic Tapes, etc.** |

**8. What is register?**

**Registers are a type of computer memory used to quickly accept, store, and transfer data and instructions that are being used immediately by the CPU. The registers used by the CPU are often termed as Processor registers.**

**9. Define RAM.**

**Random access memory (RAM) is a computer's short-term memory, which it uses to handle all active tasks and apps. None of your programs, files, games, or streams would work without RAM.**

**10. Give short notes on system software.**

**Systems software includes the programs that are dedicated to managing the computer itself, such as the operating system, file management utilities, and disk operating system (or DOS).**

**System software is a software that provides platform to other softwares. Some examples can be operating systems, antivirus softwares, disk formatting softwares, Computer language translators etc. These are commonly prepared by the computer manufacturers. These softwares consists of programs written in low-level languages, used to interact with the hardware at a very basic level. System software serves as the interface between the hardware and the end users.**

**The most important features of system software include :**

**1. Closeness to the system**

**2. Fast speed**

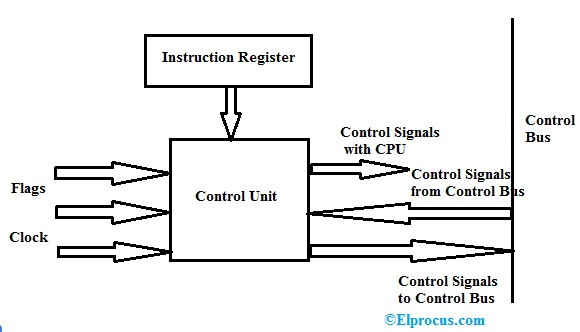
**3. Difficult to manipulate**

**4. Written in low level language**

**5. Difficult to design**

**11. Write down the operation of control unit?**

* **The control unit is the main component of a central processing unit (CPU) in computers that can direct the operations during the execution of a program by** [**the processor**](https://www.elprocus.com/microprocessor-generations-and-its-types/)**/computer.**
* **It is included as a part of Von Neumann architecture developed by John Neumann.**
* **The component which receives the input signal/information/instruction from the user and converts into control signals for the execution in the CPU.**
* **It controls and directs the main memory, arithmetic & logic unit (ALU), input and output devices, and also responsible for the instructions that are sent to the CPU of a computer.**
* **It fetches the instructions from the** [**main memory**](https://www.elprocus.com/memory-hierarchy-in-computer-architecture/) **of a processor and sent to the processor instruction register, which contains register contents.**

****

**12. Define Memory address register.**

**In a computer, the Memory Address Register is a CPU register that either stores the memory address from which data will be fetched to the CPU or the address to which data will be sent and stored. In other words, MAR holds the memory location of data that needs to be accessed.**

**13. What is stack & queue?**

**Stack is a container of objects that are inserted and removed according to the last-in first-out (LIFO) principle.**

**Queue is a container of objects (a linear collection) that are inserted and removed according to the first-in first-out (FIFO) principle.**

**14. Define Addressing modes.**

**Addressing Modes– The term addressing modes refers to the way in which the operand of an instruction is specified. The addressing mode specifies a rule for interpreting or modifying the address field of the instruction before the operand is actually executed.**

**Addressing modes for 8086 instructions are divided into two categories:**

**1) Addressing modes for data**

**2) Addressing modes for branch**

**15. Write the basic performance equation?**

* **CPU Time = I \* CPI \* T**
  + **I = number of instructions in program**
  + **CPI = average cycles per instruction**
  + **T = clock cycle time**
* **CPU Time = I \* CPI / R**
  + **R = 1/T the clock rate**
    - **T or R are usually published as performance measures for a processor**
    - **I requires special profiling software**
    - **CPI depends on many factors (including memory).**

**The performance equation analyzes execution time as a product of three factors that are relatively independent of each other.**

**16. Define clock rate.**

**The clock rate, or** [**clock speed**](https://www.easytechjunkie.com/what-is-clock-speed.htm)**, of a computer is the rate at which a** [**central processing unit**](https://www.easytechjunkie.com/what-is-a-central-processing-unit.htm) **(**[**CPU**](https://www.easytechjunkie.com/how-does-a-cpu-work.htm)**) is able to perform basic functions.**

**It is normally measured in megahertz, or millions of cycles per second, or even gigahertz, which are billions of cycles per second.**

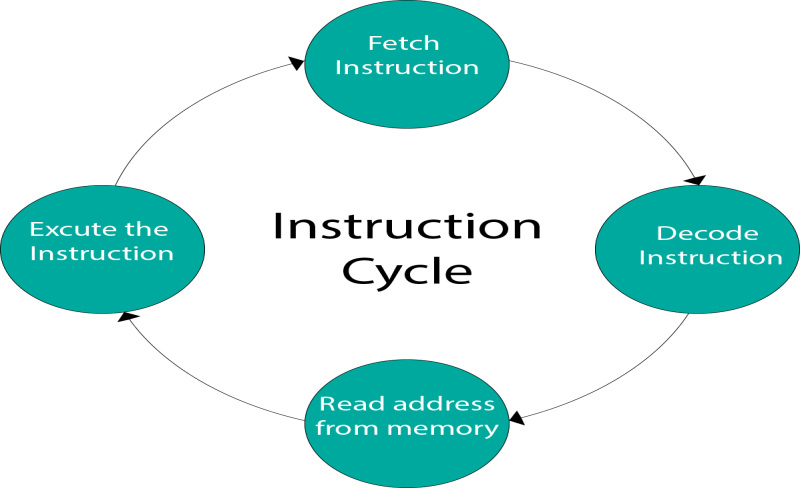
**17. List out the various addressing techniques.**

1. **Implied / Implicit Addressing Mode**
2. **Stack Addressing Mode**
3. **Immediate Addressing Mode**
4. **Direct Addressing Mode**
5. **Indirect Addressing Mode**
6. **Register Direct Addressing Mode**
7. **Register Indirect Addressing Mode**
8. **Relative Addressing Mode**
9. **Indexed Addressing Mode**
10. **Base Register Addressing Mode**
11. **Auto-Increment Addressing Mode**
12. **Auto-Decrement Addressing Mode**

**18. Draw the flow of Instruction cycle.**

**In a basic computer, each instruction cycle consists of the following phases:**

1. **Fetch instruction from memory.**
2. **Decode the instruction.**
3. **Read the effective address from memory.**
4. **Execute the instruction.**

****

**19. Suggest about Program counter.**

**A program counter (PC) is a CPU register in the computer processor which has the address of the next instruction to be executed from memory. It is a digital counter needed for faster execution of tasks as well as for tracking the current execution point.**

**A program counter is also known as an instruction counter, instruction pointer, instruction address register or sequence control register.**

**20. List out the types in displacement addressing.**

**In Displacement addressing, the instruction has two address fields, at least one of which is explicit. The value contained in one address field (value = A) is used directly. The other address field refers to a register whose contents are added to A to produce the effective address.**

**21. What is meant by stack addressing?**

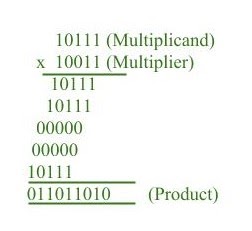
**In this mode, operand is at the top of the stack. For example: ADD, this instruction will POP top two items from the stack, add them, and will then PUSH the result to the top of the stack.**

**22. Define carry propagation delay.**

**Carry propagation is the propogation of the “carry”s of addition. When two numbers are summed, if they overflow their place, you carry a one to the next place. In decimal, the propagation does not typically last long, 999+1 for example, has a carry propagation of 3, the sum of 9 and 1 produces a carry in the tens, which propagates to the hundreds, and then the thousands.**

**23. Draw a diagram to implement manual multiplication algorithm.**

**Multiplication of two fixed point binary number in *signed magnitude representation* is done with process of *successive shift* and *add operation*.**

****

**In the multiplication process we are considering successive bits of the multiplier, least significant bit first.**

**If the multiplier bit is 1, the multiplicand is copied down else 0’s are copied down.**

**The numbers copied down in successive lines are shifted one position to the left from the previous number.**

**Finally numbers are added and their sum form the product.**

**The sign of the product is determined from the sign of the multiplicand and multiplier. If they are alike, sign of the product is positive else negative.**

**24. Perform the 2’s complement subtraction of smaller number(101011) from larger number(111001).**

Section 2:-

**1. Write briefly about computer fundamental system?**

* Input
* Output
* Processing
* Storage

***Input::*** The Computer receives its data from input devices in the form of raw data and later this data is processed in human-readable form with the help of other computer devices.

The primary input devices  are Keyboard,Mouse,Scanner,Trackball,Lightpen,Joystick

***Output ::*** The [output devices of computer](https://www.chtips.com/computer-fundamentals/output-devices-of-computer-system) receive data from the system and further process the data in human-readable form.eg are Printers,Monitors,Speakers,Headphones,Projectors

***Processing***:: This is the core function of the modern-day Personal Computer.

When the data is received from the memory it transfers the data or information for further processing.

***Storage::*** There are mainly two storage unit of the personal computer [PC]

***Primary Storage***:: Random Access Memory [RAM] is the primary storage unit of computers.

***Secondary Storage::*** [Hard Disk Drives](https://www.chtips.com/computer-peripherals/what-is-hard-disk) and Pen drives are called as secondary Storage units.

**2. Explain memory unit functions.**

This unit can store instructions, data, and intermediate results. This unit supplies information to other units of the computer when needed. It is also known as internal storage unit or the main memory or the primary storage or Random Access Memory (RAM).

Its size affects speed, power, and capability. Primary memory and secondary memory are two types of memories in the computer. Functions of the memory unit are −

·  It stores all the data and the instructions required for processing.

·  It stores intermediate results of processing.

·  It stores the final results of processing before these results are released to an output device.

·  All inputs and outputs are transmitted through the main memory.

**3. Explain memory locations and addresses.**

Memory addresses are fixed-length sequences of digits conventionally displayed and manipulated as unsigned integers. Memory location in a stored-program computer holds a binary number or decimal number of some sort.

Memory locations and addresses determine how the computer’s memory is organized so that the user can efficiently store or retrieve information from the computer. The computer’s memory is made of a silicon chip which has millions of storage cell, where each storage cell is capable to store a *bit* of information which value is either 0 or 1.

**4. Explain Software interface.**

Software interfaces (programming interfaces) are **the languages, codes and messages that programs use to communicate with each other and to the hardware**. Examples are the Windows, Mac and Linux operating systems, SMTP email, IP network protocols and the software drivers that activate the peripheral devices.

**5. Explain instruction set Architecture? Give examples.**

An Instruction Set Architecture (ISA) is **part of the abstract model of a computer that defines how the CPU is controlled by the software**. The ISA acts as an interface between the hardware and the software, specifying both what the processor is capable of doing as well as how it gets done.

Today's computers have thousands of instructions. An example of an instruction set is the x86 instruction set, which is common to find on computers today.

**6. What is bus explain it in detail?**

A bus is a subsystem that is used to connect computer components and transfer data between them. For example, an internal bus connects computer internals to the motherboard.A bus may be parallel or serial. Parallel buses transmit data across multiple wires. Serial buses transmit data in bit-serial format.

Computer bus types are as follows:

·        **System Bus:** A parallel bus that simultaneously transfers data in 8-, 16-, or 32-bit channels and is the primary pathway between the CPU and memory.

·        **Internal Bus:** Connects a local device, like internal CPU memory.

·        **External Bus:** Connects peripheral devices to the motherboard, such as scanners or disk drives.

·        **Expansion Bus:** Allows expansion boards to access the CPU and RAM.

·        **Frontside Bus:** Main computer bus that determines data transfer rate speed and is the primary data transfer path between the CPU, RAM and other motherboard devices.

·        **Backside Bus:** Transfers secondary cache (L2 cache) data at faster speeds, allowing more efficient CPU operations.

**7. Explain briefly about performance evaluation by using various bench marks. List out the types of bench marks and mention its advantage and disadvantage.**

In [computing](https://en.wikipedia.org/wiki/Computing), a **benchmark** is the act of running a [computer program](https://en.wikipedia.org/wiki/Computer_program), a set of programs, or other operations, in order to assess the relative [performance](https://en.wikipedia.org/wiki/Computer_performance) of an object, normally by running a number of standard [tests](https://en.wikipedia.org/wiki/Software_performance_testing) and trials against it.[[1]](https://en.wikipedia.org/wiki/Benchmark_(computing)#cite_note-1) The term *benchmark* is also commonly utilized for the purposes of elaborately designed benchmarking programs themselves.Types of benchmark-:

1. Real program

o   word processing software

o   tool software of CAD

o   user's application software (i.e.: MIS)

2. Component Benchmark / Microbenchmark

o   core routine consists of a relatively small and specific piece of code.

o   measure performance of a computer's basic components[[6]](https://en.wikipedia.org/wiki/Benchmark_(computing)#cite_note-6)

o   may be used for automatic detection of computer's hardware parameters like number of registers, [cache](https://en.wikipedia.org/wiki/Cache_(computing)) size, [memory latency](https://en.wikipedia.org/wiki/Memory_latency), etc.

3. Kernel

o   contains key codes

o   normally abstracted from actual program

o   popular kernel: Livermore loop

o   linpack benchmark (contains basic linear algebra subroutine written in FORTRAN language)

o   results are represented in Mflop/s.

4. Synthetic Benchmark

o   Procedure for programming synthetic benchmark:

§  take statistics of all types of operations from many application programs

§  get proportion of each operation

§  write program based on the proportion above

o   Types of Synthetic Benchmark are:

§ [Whetstone](https://en.wikipedia.org/wiki/Whetstone_(benchmark))

§ [Dhrystone](https://en.wikipedia.org/wiki/Dhrystone)

o   These were the first general purpose industry standard computer benchmarks. They do not necessarily obtain high scores on modern pipelined computers.

5. I/O benchmarks

6. Database benchmarks

o   measure the throughput and response times of database management systems (DBMS)

7. Parallel benchmarks

o   used on machines with multiple cores and/or processors, or systems consisting of multiple machines

**8. Explain the operations of stacks and queues.**

**In computer architecture, a stack is an abstract data type that serves as a collection of elements, with two main principal operations:**

·        Push, which adds an element to the collection, and.

·        Pop, which removes the most recently added element that was not yet removed..

In the queue only two operations are allowed **enqueue and dequeue**. Enqueue means to insert an item into the back of the queue, dequeue means removing the front item.

**9. Discuss about different types of addressing modes.**

There are various types of Addressing Modes which are as follows −

**Implied Mode** − In this mode, the operands are specified implicitly in the definition of the instruction. For example, the instruction "complement accumulator" is an implied-mode instruction because the operand in the accumulator register is implied in the definition of the instruction. All register reference instructions that use an accumulator are implied-mode instructions.

**Immediate Mode** − In this mode, the operand is specified in the instruction itself. In other words, an immediate-mode instruction has an operand field instead of an address field. The operand field includes the actual operand to be used in conjunction with the operation determined in the instruction. Immediate-mode instructions are beneficial for initializing registers to a constant value.

**Register Mode** − In this mode, the operands are in registers that reside within the CPU. The specific register is selected from a register field in the instruction. A k-bit field can determine any one of the 2k registers.

**Register Indirect Mode** − In this mode, the instruction defines a register in the CPU whose contents provide the address of the operand in memory. In other words, the selected register includes the address of the operand rather than the operand itself.

A reference to the register is then equivalent to specifying a memory address. The advantage of a register indirect mode instruction is that the address field of the instruction uses fewer bits to select a register than would have been required to specify a memory address directly.

**Autoincrement or Autodecrement Mode** &minuend; This is similar to the register indirect mode except that the register is incremented or decremented after (or before) its value is used to access memory. When the address stored in the register defines a table of data in memory, it is necessary to increment or decrement the register after every access to the table. This can be obtained by using the increment or decrement instruction.

**Direct Address Mode** − In this mode, the effective address is equal to the address part of the instruction. The operand resides in memory and its address is given directly by the address field of the instruction. In a branch-type instruction, the address field specifies the actual branch address.

**Indirect Address Mode** − In this mode, the address field of the instruction gives the address where the effective address is stored in memory. Control fetches the instruction from memory and uses its address part to access memory again to read the effective address.

**Indexed Addressing Mode** − In this mode, the content of an index register is added to the address part of the instruction to obtain the effective address. The index register is a special CPU register that contains an index value. The address field of the instruction defines the beginning address of a data array in memory.

**10. Explain in detail about different instruction types and instruction sequencing.**

Depending on operation they perform, all instructions are divided in several groups:

·      **Arithmetic Instructions**- Arithmetic instructions perform several basic operations such as addition, subtraction, division, multiplication etc. After execution, the result is stored in the first operand. For example: ADD A,R1 - The result of addition (A+R1) will be stored in the accumulator.

·        **Branch Instructions**- There are two kinds of branch instructions: Unconditional jump instructions: upon their execution a jump to a new location from where the program continues execution is executed. Conditional jump instructions: a jump to a new program location is executed only if a specified condition is met. Otherwise, the program normally proceeds with the next instruction.

·        **Data Transfer Instructions**- Data transfer instructions move the content of one register to another. The register the content of which is moved remains unchanged. If they have the suffix “X” (MOVX), the data is exchanged with external memory.

·        **Logic Instructions**- Logic instructions perform logic operations upon corresponding bits of two registers. After execution, the result is stored in the first operand.

·        Bit-oriented Instructions- Similar to logic instructions, bit-oriented instructions perform logic operations. The difference is that these are performed upon single bits.

instruction sequencing **The order in which the instructions in a program are carried out**. Normally the sequence proceeds in a linear fashion through the program, and the address of the instructions is obtained from the program counter in the control unit.

**11. Explain Fixed point representation.**

In computing, fixed-point number representation is a method of **representing fractional (non-integer) numbers by storing a fixed number of digits of their fractional part**.. With the help of fixed number representation, data is converted into binary form, and then data is processed, stored and used by the system. The fixed-point numbers in binary uses a sign bit and This representation has fixed number of bits for integer part and for fractional part.

**12. How floating point addition is implemented. Explain briefly with a neat diagram.**

To understand floating point addition, first we see addition of real numbers in decimal as same logic is applied in both cases.

**For example,** we have to add **1.1 \* 103** and **50.**

We cannot add these numbers directly. First, we need to align the exponent and then, we can add significant

After aligning exponent, we get **50 = 0.05 \* 103**

Now adding significant, **0.05 + 1.1 = 1.15**

So, finally we get **(1.1 \* 103 + 50) = 1.15 \* 103**

Here, notice that we shifted **50** and made it **0.05** to add these numbers.

**Now let us take example of floating point number addition**

We follow these steps to add two numbers:

1. Align the significant

2. Add the significant

3. Normalize the result

**Let the two numbers be**

x = 9.75

y = 0.5625

Converting them into 32-bit floating point representation,

**9.75**’s representation in 32-bit format = **0 10000010 00111000000000000000000**

**0.5625**’s representation in 32-bit format = **0 01111110 00100000000000000000000**

Now we get the difference of exponents to know how much shifting is required.

(**10000010 – 01111110**)2 = (**4**)10

Now, we shift the mantissa of lesser number right side by 4 units.

Mantissa of **0.5625 = 1.00100000000000000000000**

(note that 1 before decimal point is understood in 32-bit representation)

Shifting right by **4** units, we get **0.00010010000000000000000**

Mantissa of **9.75** = **1. 00111000000000000000000**

Adding mantissa of both

**0. 00010010000000000000000**

**+ 1. 00111000000000000000000**

————————————————-

**1. 01001010000000000000000**

In final answer, we take exponent of bigger number

So, final answer consist of :

Sign bit = **0**

Exponent of bigger number = **10000010**

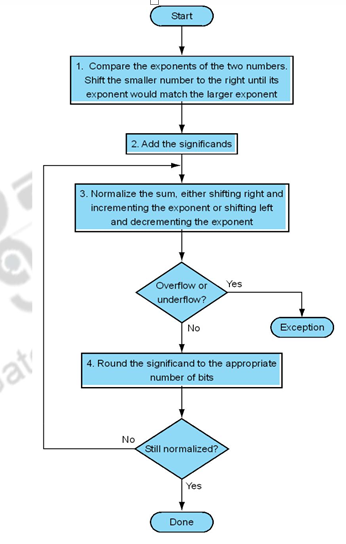
Mantissa = **01001010000000000000000**

32 bit representation of answer = **x + y** = **0 10000010 01001010000000000000000**

**13. Give the difference between RISC and CISC.**

|  |  |
| --- | --- |
| CISC | RISC |
| A large number of instructions are present in the architecture. | Very few instructions are present. The number of instructions is generally less than 100. |
| Some instructions with long execution times. These include instructions that copy an entire block from one part of memory to another and others that copy multiple registers to and from memory. | No instruction with a long execution time due to a very simple instruction set. Some early RISC machines did not even have an integer multiply instruction, requiring compilers to implement multiplication as a sequence of additions. |
| Variable-length encodings of the instructions.  **Example:** IA32 instruction size can range from 1 to 15 bytes. | Fixed-length encodings of the instructions are used.  **Example:** In IA32, generally all instructions are encoded as 4 bytes. |
| Multiple formats are supported for specifying operands. A memory operand specifier can have many different combinations of displacement, base, and index register. | Simple addressing formats are supported. Only base and displacement addressing is allowed. |
| CISC supports array. | RISC does not support an array. |
| Arithmetic and logical operations can be applied to both memory and register operands. | Arithmetic and logical operations only use register operands. Memory referencing is only allowed by loading and storing instructions, i.e. reading from memory into a register and writing from a register to memory respectively. |
| Implementation programs are hidden from machine-level programs. The ISA provides a clean abstraction between programs and how they get executed. | Implementation programs exposed to machine-level programs. Few RISC machines do not allow specific instruction sequences. |
| Condition codes are used. | No condition codes are used. |
| The stack is being used for procedure arguments and returns addresses. | Registers are being used for procedure arguments and return addresses. Memory references can be avoided by some procedures |

**14. Write an algorithm for the division of floating point number and illustrate with an example.**



Section 3: -

1. What are the basic operations performed by the processor?

The CPU processes instructions it receives in the process of decoding data. In processing this data, the CPU performs four basic steps:

1. **Fetch:** Each instruction is stored in memory and has its own address. The processor takes this address number from the program counter, which is responsible for tracking which instructions the CPU should execute next.
2. **Decode:** All programs to be executed are translated into Assembly instructions. Assembly code must be decoded into binary instructions, which are understandable to your CPU. This step is called decoding.
3. **Execute:** While executing instructions, the CPU can do one of three things: Do calculations with its ALU, move data from one memory location to another, or jump to a different address.
4. **Store:** The CPU must give feedback after executing an instruction, and the output data is written to the memory.

2. Define Data path.

A data path (also written as datapath) is a set of functional units that carry out data processing operations. Datapaths, with a control unit, make up the CPU (central processing unit) of a computer system. A larger data path can also be created by joining more than one together using multiplexers.

Currently, data paths can only be configured once. Researchers are trying to find ways to imprint data paths on fabrics and make them reconfigurable. This action would allow them to be configured at runtime, providing for improved efficiency and power savings.

3. Define Processor clock.

A computer's processor clock speed determines how quickly the central processing unit (CPU) can retrieve and interpret instructions. This helps your computer complete more tasks by getting them done faster. Clock speeds are measured in gigahertz (GHz), with a higher number equating to higher clock speed.

4. Define Latency and throughput.

**The time it takes for a packet to travel from the source to its destination is referred to as latency.  Latency indicates how long it takes for packets to reach their destination. Throughput is the term given to the number of packets that are processed within a specific period of time. Throughput and latency have a direct relationship in the way they work within a network.**

5. Discuss the principle operation of a micro-programmed control unit.

A control unit whose binary control values are saved as words in memory is called a microprogrammed control unit.

A controller results in the instructions to be implemented by constructing a definite collection of signals at each system clock beat. Each of these output signals generates one micro-operation including register transfer. Thus, the sets of control signals are generated definite micro-operations that can be saved in the memory.

6. What are the differences between hardwired and micro programmed control units?

|  |  |
| --- | --- |
| Hardwired Control Unit | Microprogrammed Control Unit |
| Hardwired control unit generates the control signals needed for the processor using logic circuits | Micrprogrammed control unit generates the control signals with the help of micro instructions stored in control memory |
| Hardwired control unit is faster when compared to microprogrammed control unit as the required control signals are generated with the help of hardwares | This is slower than the other as micro instructions are used for generating `signals here |
| Difficult to modify as the control signals that need to be generated are hard wired | Easy to modify as the modification need to be done only at the instruction level |
| More costlier as everything has to be realized in terms of logic gates | Less costlier than hardwired control as only micro instructions are used for generating control signals |
| It cannot handle complex instructions as the circuit design for it becomes complex | It can handle complex instructions |
| Only limited number of instructions are used due to the hardware implementation | Control signals for many instructions can be generated |
| Used in computer that makes use of Reduced Instruction Set Computers(RISC) | Used in computer that makes use of Complex Instruction Set Computers(CISC) |

7. Define nanoprogramming.

In microprogrammed processors, an instruction fetched from memory is interpreted by a micro program stored in a single control memory CM; whereas in other microprogrammed processors, the micro instructions are not directly used by the decoder to generate control signals.

This is achieved by the use of a second control memory called a Nano control memory (nCM).

So now there are two levels of control memories, a higher level control memory is known as micro control memory (µCM) and a lower level control memory is known as Nano control memory (nCM). This is shown in Figure 7.

Thus a microinstruction is in primary control-store memory, it then has the control signals generated for each microinstruction using a secondary control store memory The output word from the secondary memory is called Nano instruction.

The µCM stores micro instructions whereas nCM stores nano instructions.

The decoder uses Nano instructions from nCM to generate control signals.

Thus Nano programming gives an alternative strategy to generate control signals.

8. What is a control store?

A control store is the part of a CPU's control unit that stores the CPU's microprogram. It is usually accessed by a microsequencer. A control store implementation whose contents are unalterable is known as a Read Only Memory (ROM) or Read Only Storage (ROS); one whose contents are alterable is known as a Writable Control Store (WCS).

9. What are the advantages of multiple bus organization over a single bus organization?

In a single-bus architecture, all components including the central processing unit, memory and peripherals share a common bus. When many devices need the bus at the same time, this creates a state of conflict called bus contention; some wait for the bus while another has control of it. The waiting wastes time, slowing the computer down, as Engineering 360 explains. Multiple buses permit several devices to work simultaneously, reducing time spent waiting and improving the computer's speed. Performance improvements are the main reason for having multiple buses in a computer design.

10. **Write control sequencing for the executing the instruction. Add R4,R5,R6.**

11. What is nano control memory?

In microprogrammed processors, an instruction fetched from memory is interpreted by a micro program stored in a single control memory CM; whereas in other microprogrammed processors, the micro instructions are not directly used by the decoder to generate control signals.

This is achieved by the use of a second control memory called a Nano control memory (nCM).

12. **What is the nano instruction format of Qm-1?**

**13. What is the capacity of nano control memory?**

14. Define micro routine.

A sequence of control words corresponding to the control sequence of a machine instruction represents the micro routine for that instruction and the individual control words in this micro routine are referred to as microinstructions.

15. What is meant by hardwired control?

A hardwired control is a mechanism of producing control signals using Finite State Machines (FSM) appropriately. It is designed as a sequential logic circuit. The final circuit is constructed by physically connecting the components such as gates, flip flops, and drums. Hence, it is named a hardwired controller.

(https://www.tutorialspoint.com/what-is-hardwired-control-unit)

16. **What are the types of micro instruction?**

(https://www.tutorialspoint.com/what-is-the-format-of-microinstruction-in-computer-architecture)

17. Name the methods for generating the control signals.

(<https://www.geeksforgeeks.org/computer-organization-hardwired-vs-micro-programmed-control-unit/#:~:text=There%20are%20two%20approaches%20used,and%20Micro%2Dprogrammed%20control%20unit>.)

Section 4:-

1.What is Pipelining?

Pipelining is the process of accumulating and executing computer   instructions and tasks from the processor via a logical pipeline.It allows storing and executing instructions in an orderly process. Simultaneous execution of more than one instruction takes place in a pipelined processor. In a pipelined processor, a pipeline has two ends, the input end and the output end. Between these ends, there are multiple stages/segments such that output of one stage is connected to input of next stage and each stage performs a specific operation.Interface registers are used to hold the intermediate output between two stages.

Pipelining is the process of accumulating instruction from the processor through a pipeline. It allows storing and executing instructions in an orderly process. It is also known as pipeline processing.

Pipelining is a technique where multiple instructions are overlapped during execution. Pipeline is divided into stages and these stages are connected with one another to form a pipe like structure. Instructions enter from one end and exit from another end. Pipelining increases the overall instruction throughput.

2. What are the major characteristics of a Pipeline?

1. several computations can be in progress in distinct segments at the same time.
2. The overlapping of computation is made possible by associating a register with each segment in the pipeline.
3. Pipelining cannot be executed on a single task, as it works by splitting multiple tasks into a number of subtasks and operating on them concurrently.
4. The speedup or efficiency achieved by suing a pipeline depends on the number of pipe stages and the number of obtainable tasks that can be subdivided.

3. What are the various stages in a Pipeline execution?

1. **Instruction fetch,**
2. **Instruction Decode**
3. **Instruction execute**
4. **Memory access**
5. **write-back**

4. What are the types of pipeline hazards?

1. **Arithmetic Pipelining**
2. **Instruction Pipelining**
3. **Processor Pipelining**
4. **Unifunction Vs. Multifunction Pipelining**
5. **Static vs Dynamic Pipelining**
6. **Scalar vs Vector Pipelining**

5. Define structural, data, and control hazard.

i. **Data Hazards:**

A data hazard is any condition in which either the source or the destination operands of an instruction are not available at the time expected in the pipeline. As a result of which some operation has to be delayed and the pipeline stalls. Whenever there are two instructions one of which depends on the data obtained from the other.

A=3+A

B=A\*4

For the above sequence, the second instruction needs the value of ‘A’ computed in the first instruction.

ii. **Structural Hazards:**

This situation arises mainly when two instructions require a given hardware resource at the same time and hence for one of the instructions the pipeline needs to be stalled.

The most common case is when memory is accessed at the same time by two instructions. One instruction may need to access the memory as part of the Execute or Write back phase while other instruction is being fetched. In this case if both the instructions and data reside in the same memory. Both the instructions can’t proceed together and one of them needs to be stalled till the other is done with the memory access part.

iii. **Control hazards:**

The instruction fetch unit of the CPU is responsible for providing a stream of instructions to the execution unit. The instructions fetched by the fetch unit are in consecutive memory locations and they are executed.

Control hazard occurs when the pipeline makes wrong decisions on branch prediction and therefore brings instructions into the pipeline that must subsequently be discarded. The term branch hazard also refers to a control hazard.

6. List two conditions when processor can stall.

7. List the types of data hazards.

1) RAW (Read after Write) [Flow/True data dependency]

2) WAR (Write after Read) [Anti-Data dependency]

3) WAW (Write after Write) [Output data dependency]

8. List the techniques used for overcoming hazard.

[pipeline stalls](https://en.wikipedia.org/wiki/Pipeline_stall)/pipeline bubbling, [operand forwarding](https://en.wikipedia.org/wiki/Hazard_(computer_architecture)#Operand_forwarding), and in the case of [out-of-order execution](https://en.wikipedia.org/wiki/Out-of-order_execution), the [scoreboarding](https://en.wikipedia.org/wiki/Scoreboarding) method and the [Tomasulo algorithm](https://en.wikipedia.org/wiki/Tomasulo_algorithm).

9. What is instruction level parallelism?

Instruction Level Parallelism (ILP) is used to refer to the architecture in which multiple operations can be performed parallelly in a particular process, with its own set of resources – address space, registers, identifiers, state, program counters.

Instruction Level Parallelism is achieved when multiple operations are performed in single cycle, that is done by either executing them simultaneously or by utilizing gaps between two successive operations that is created due to the latencies.

10. What are the types of dependencies?

**data, name, and control**

11. What is delayed branching?

When branches are processed by a pipeline simply, after each taken branch, at least one cycle remains unutilized. This is because of the assembly line-like apathy of pipelining. Instruction slots following branches are known as branch delay slots. Delayed branch A conditional branch instruction found in some RISC architectures that include pipelining.

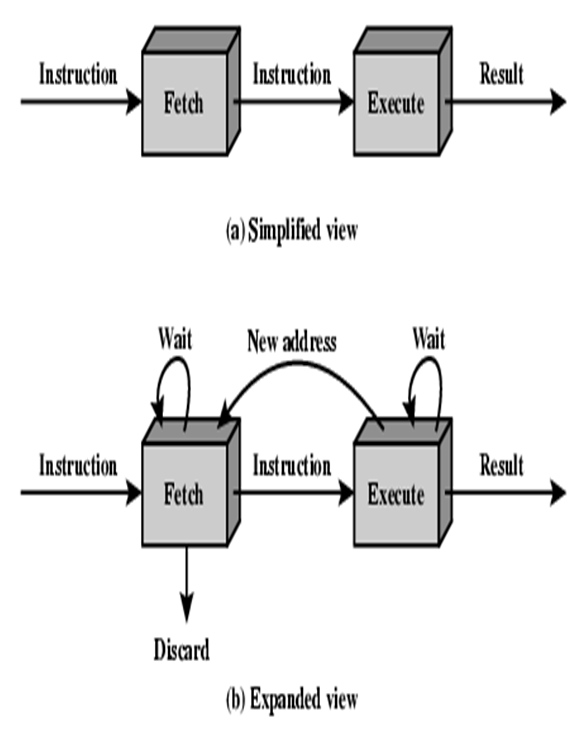
Delayed branching requires a redefinition of the architecture.

12. Define deadlock.

Deadlockis a situation where a set of processes are blocked because each process is holding a resource and waiting for another resource acquired by some other process.

A deadlock occurs when there is at least one process which is waiting for resources to be released by another process in order to finish a task correctly.

13. Draw the hardware organization of two stage pipeline.



14. What is branch prediction?

15. Give two examples for instruction hazard.

16. List the various pipelined processors.

17. Why we need an instruction buffer in a pipelined CPU?

18. What are the problems faced in instruction pipeline?

19. Write down the expression for speedup factor in a pipelined architecture.

   Speedup = **Pipeline Depth / 1 + Pipeline stall cycles per instruction**.

Explain different types of hazards that occur in a pipeline.

20. Explain various approaches used to deal with conditional branching.

When a conditional branched is recognized, the target of the branch is prefetched, in addition to the instruction following the branch. This target is then saved until the branch instruction is executed. If the branch is taken, the target has already been prefetched.

Multiple streams

Pre fetch branch target

Loop buffer

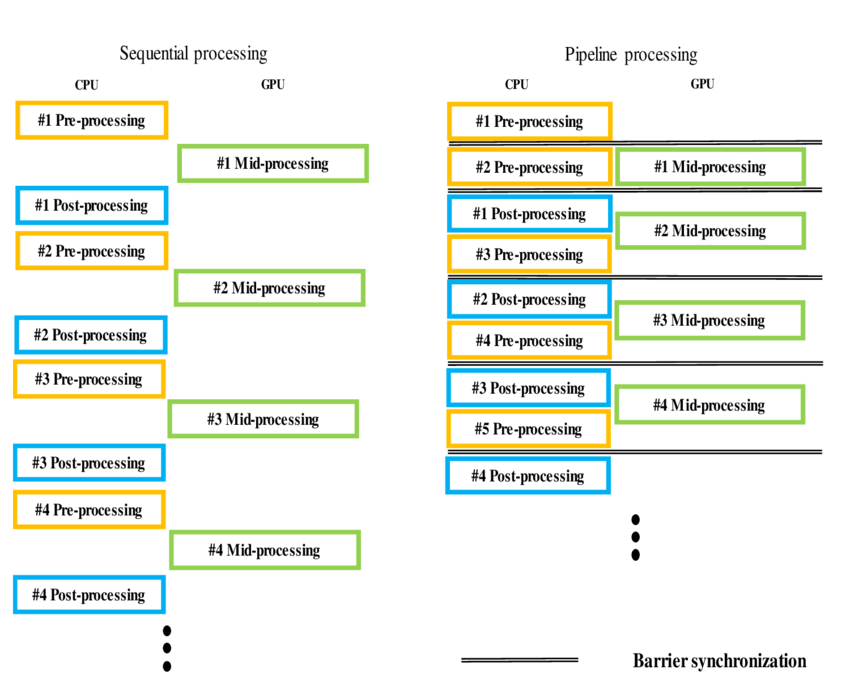
Branch prediction

Delayed branch

21. Explain the basic concepts of pipelining and compare it with sequence processing with a neat diagram.

Sequential mode executes instructions in the sequence in which they are written in the program. After a measurement is made, the result is converted to a value determined by processing arguments that are included in the measurement instruction, and then program execution proceeds to the next instruction. This line-by-line execution allows writing conditional measurements into the program.

Pipeline mode handles measurement, most digital, and processing tasks separately, and, in many cases, simultaneously. Measurements are scheduled to execute at exact times and with the highest priority, resulting in more precise timing of measurements, and usually more efficient processing and power consumption.



22. Explain instruction pipelining.

Pipeline processing can occur not only in the data stream but in the instruction stream as well. Most of the digital computers with complex instructions require instruction pipeline to carry out operations like fetch, decode and execute instructions. The organization of an instruction pipeline will be more efficient if the instruction cycle is divided into segments of equal duration.

23. What is branch hazard? Describe the method for dealing with the branch hazard?

A branch in a sequence of instructions causes a problem. An instruction must be fetched at every clock cycle to sustain the pipeline.

This delay in determining the proper instruction to fetch is called a *control* *hazard* or *branch hazard*, in contrast to the *data hazards* we examined in the previous modules.

1. Stall until the branch outcome is known or perform the fetch again
2. Predict the behavior of branchesuse
3. [branch prediction](https://en.wikipedia.org/wiki/Branch_prediction) and essentially make educated guesses about which instructions to insert, in which case a *pipeline bubble* will only be needed in the case of an incorrect prediction

24. What is data hazard? Explain the methods for dealing with data hazard?

25. Explain the function of six segment pipeline and draw a space diagram for six segment pipeline solving the time it takes to process eight tables.

26. Explain the influence of instruction sets.

27. Draw and explain data path modified for pipelined execution.

28. Explain about various exceptions.

**SECTION 5**

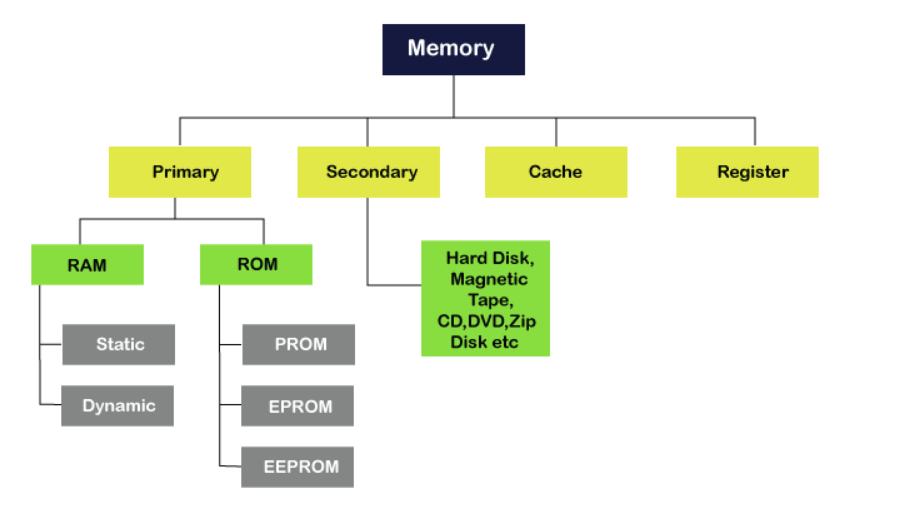
imp link: https://www.javatpoint.com/classification-of-memory

**1. What is Memory system?**

-> In a computer system, the memory system acts as the storehouse for information. The processor receives data from the memory system, performs computations on it, and then saves the results back to memory. The memory system is made up of a number of different storage sites. A numerical address is assigned to each storage location or memory word. An address space's set of storage locations

**2. Give classification of memory.**

->



**3. Define cache.**

->Cache memory, often known as cache, is a supplemental memory system that saves frequently used instructions and data in order for the central processing unit (CPU) of a computer to perform them more quickly.The cache stores just a copy of the most frequently used data or programme codes from the main memory. The cache's reduced size minimises the time it takes to identify data and send it to the CPU for processing.

**4. What is Read Access Time?**

**5. Define Random Access Memory.**

-> Random Access Memory (RAM) is a sort of primary memory that can be accessed directly by the CPU and is one of the fastest. It's the hardware in a computer device that stores data, programs, and program results temporarily. It's used to read and write data in memory until the machine is ready to use. It is volatile, which means that if the computer is shut off or if there is a power outage, the information contained in RAM will be gone. At any time, all data stored in computer memory can be read or accessed at random.

**6. What are PROMS?**

-> **(Programmable Read Only Memory)**

It's a sort of digital read-only memory in which the user can only write a single piece of data or programme. It refers to an empty PROM chip on which the user may only write the required content or programme once with a specific PROM programmer or PROM burner device; after that, the data or instruction cannot be modified or deleted**.**

**7. Define Memory refreshing.**

->Memory refresh is a procedure that specifies many of the features of dynamic random access memory (DRAM), the most common form of computer memory. The procedure entails reading information from a certain portion of the memory on a regular basis and immediately rewriting it to the same location without making any modifications. It is a background maintenance procedure that is required for DRAM functioning.

**8. What is SRAM and DRAM?**

->There are two types of RAM:

SRAM

DRAM

DRAM (Dynamic Random-Access Memory) is a form of RAM that allows data to be stored dynamically in RAM. Each cell in DRAM stores one bit of data. A capacitor and a transistor are the two components of the cell.

Characteristics of DRAM

* It requires continuously refreshed to retain the data.
* It is slower than SRAM
* It holds a large amount of data
* It is the combination of capacitor and transistor
* It is less expensive as compared to SRAM
* Less power consumption

Static Random-Access Memory (SRMA) is a form of RAM that is used to store static data. It means that data stored in SRAM is active for as long as the computer system is powered up. When there are power outages, however, data in SRAM is lost.

Characteristics of Static Ram

* It does not require to refresh.
* It is faster than DRAM
* It is expensive.
* High power consumption
* Longer life
* Large size
* Uses as a cache memory

**9. What is volatile memory?**

->In contrast to non-volatile memory, volatile memory is computer memory that requires power to keep its contents; it retains its contents while switched on, but the recorded data is quickly lost if the power is disconnected.

Volatile memory can be used for a variety of purposes, including main storage. Volatility can secure sensitive information by making it unreachable when the power is turned off, in addition to being quicker than other kinds of mass storage such as a hard disc drive. The majority of random-access memory (RAM) for general-purpose applications is volatile.

**10. Define data transfer or band width.**

->It is defined as the potential of the data that is to be transferred in a specific period of time. It is the data carrying capacity of the network/transmission medium.

**11. What is flash memory?**

->Flash memory is a non-volatile storage chip with a long life cycle that is frequently utilised in embedded devices. Even when the power is turned off, it may maintain saved data and information. It may be deleted and reprogrammed electronically. EEPROM was used to create flash memory (electronically erasable programmable read-only memory).

**12. What is multi level memories?**

**13. What is address translation page fault routine, page fault and demand paging?**

->Page fault dominates more like an error. It mainly occurs when any program tries to access the data or the code that is in the address space of the program, but that data is not currently located in the RAM of the system.

So basically when the page referenced by the CPU is not found in the main memory then the situation is termed as Page Fault.

Whenever any page fault occurs, then the required page has to be fetched from the secondary memory into the main memory.

In case if the required page is not loaded into the memory, then a page fault trap arises

(https://www.studytonight.com/operating-system/page-fault-in-operating-system)

**14. What is associate memory?**

->Associative memory is also known as content addressable memory (CAM) or associative storage or associative array. It is a special type of memory that is optimized for performing searches through data, as opposed to providing a simple direct access to the data based on the address.

Associative memory of conventional semiconductor memory (usually RAM) with added comparison circuity that enables a search operation to complete in a single clock cycle. It is a hardware search engine, a special type of computer memory used in certain very high searching applications.

Applications of Associative memory :-

* It can be only used in memory allocation format.
* It is widely used in the database management systems, etc.

**15. Define Seek time and latency time.**

->Seek Time:

A disk is divided into many circular tracks. Seek Time is defined as the time required by the read/write head to move from one track to another.

Rotational Latency:

The disk is divided into many circular tracks, and these tracks are further divided into blocks knows as sectors. The time required by the read/write head to rotate to the requested sector from the current position is called Rotational Latency.

(<https://www.geeksforgeeks.org/difference-between-seek-time-and-rotational-latency-in-disk-scheduling/>)

**16. What is TLB?**

->Translation Lookaside Buffer (TLB) in Paging

A translation lookaside buffer (TLB) is a memory cache that is used to reduce the time taken to access a user memory location.[1] It is a part of the chip's memory-management unit (MMU). The TLB stores the recent translations of virtual memory to physical memory and can be called an address-translation cache. A TLB may reside between the CPU and the CPU cache, between CPU cache and the main memory or between the different levels of the multi-level cache.

**17. Define Magneto Optical disk.**

->A magneto-optical disk is a rewritable disk that makes use of both magnetic disk and optical technologies. It is similar to a magnetic diskette except for its larger size. Magneto-optical disks are seldom manufactured and used due to the advent of flash drives and DVD/CD drives, which are less expensive and have better writing time and reliability.

Magneto-optical disks are also known as magneto-optical drives and MO drives.

One of the most well-known examples of a magneto-optical disk is the Sony MiniDisc.

* The magneto-optical disk is a special removable disk.
* The design of the drive allows the inserted disk to be exposed to the magnetic head on one side and to the laser on the other side.

* Its writing speed is faster than that of diskettes, but is slower than that of CD/DVD drives.

**18. Define Virtual memory.**

->Virtual Memory is a type of storage that gives the user the impression of having a large main memory. This is accomplished by reclassifying a portion of secondary memory as primary memory.

This approach allows the user to load programmes that are larger than the available main memory by creating the appearance that more memory is available.

Rather than loading a single large process into main memory, the Operating System loads the various elements of several processes into main memory.

The degree of multiprogramming will be enhanced as a result, and the CPU use will be increased as well.

**19. What are the enhancements used in the memory management?**

 ->The following 3 approaches are used to enhance memory performance

* Wide path memory access
* Memory interleaving
* Cache memory

Wide Path Memory Access

· This technique provides for wider data bus between the CPU and memory and large data register

· Many bytes can be transferred in parallel in a single operation

·  There is a limit to how wide the path can be

    ->  As the path size increases the complexity of directing data bytes to their location increases

     ->As the path size increases the probability that the extra bytes will never be used increases

· Many recent systems have found that 64-bit data path is an adequate compromise

Memory Interleaving

·  Memory interleaving is a method of dividing memory into several parts where each part can be accessed in parallel

·  Each part has its own MAR and MDR registers and each part is independently accessible

· n-way interleaving is a method of breaking memory so that successive locations are in different parts

Cache Memory

·  Cache memory is a small high-speed memory (e.g. made of SRAM) that stores most frequently used code and data

·  Cache memory is not user visible and sets in between the CPU and the main memory

·   As the level increases the cache size increases

**20. Define the term LRU and LFU.**

->LRU is a cache eviction algorithm called least recently used cache

LFU is a cache eviction algorithm called least frequently used cache.

(https://stackoverflow.com/questions/17759560/what-is-the-difference-between-lru-and-lfu#:~:text=LRU%20is%20a%20cache%20eviction%20algorithm%20called%20least%20recently%20used%20cache.&text=LFU%20is%20a%20cache%20eviction,entry%20at%20O(1).)

**21. Define memory cycle time.**

->It is the total time that is required to store next memory access

operation from the previous memory access operation.

Memory cycle time = access time plus transient time (any additional time required

before a second access can commence).

— Time may be required for the memory to “recover” before next access

— Cycle time is access + recovery

**22. What is static memories?**

Static memories are the Memories that consist of circuits capable of retaining their state as long as power is applied are known as static memories

(http://www.faadooengineers.com/online-study/post/cse/computer-organization-and-architecture/498/static-memories)

**23. What is locality of reference?**

In computer science, locality of reference, also known as the principle of locality, is the tendency of a processor to access the same set of memory locations repetitively over a short period of time.

 There are two basic types of reference locality – temporal and spatial locality.

Temporal locality refers to the reuse of specific data and/or resources within a relatively small time duration.

Spatial locality refers to the use of data elements within relatively close storage locations. Sequential locality, a special case of spatial locality, occurs when data elements are arranged and accessed linearly, such as traversing the elements in a one-dimensional array.

**24. Define set associative cache.**

->Set-associative cache is a trade-off between direct-mapped cache and fully associative cache.

A set-associative cache can be imagined as a (n\*m) matrix. The cache is divided into ‘n’ sets and each set contains ‘m’ cache lines. A memory block is first mapped onto a set and then placed into any cache line of the set.

**25. What is meant by block replacement?**

->Least Recently Used (LRU): replace the  block either never used of used long ago –It reduces the chances of throwing out information that may be needed soon –Here, the access time and number of times a block is accessed is recorded –The block replaced is one that has not been used for longest time

**26. List the advantages of write through cache.**

->In write-through, data is simultaneously updated to cache and memory. This process is simpler and more reliable. This is used when there are no frequent writes to the cache(The number of write operations is less).

It helps in data recovery (In case of a power outage or system failure). A data write will experience latency (delay) as we have to write to two locations (both Memory and Cache). It Solves the inconsistency problem. But it questions the advantage of having a cache in write operation (As the whole point of using a cache was to avoid multiple access to the main memory).

(<https://www.geeksforgeeks.org/write-through-and-write-back-in-cache/>)

**27. Give formula to calculate average memory access time.**

-> average memory access time = hit time0 + miss rate0 \* (hit time1 + miss rate1 \* miss penalty1 )

**28. Define conflict.**

-> A resource conflict is a situation when more than one instruction tries to access the same resource in the same cycle. A resource can be a register, memory, or ALU.

**29. What is memory interleaving?**

-> Memory Interleaving is less or More an Abstraction technique. Though it’s a bit different from Abstraction. It is a Technique that divides memory into a number of modules such that Successive words in the address space are placed in the Different modules.

(https://www.geeksforgeeks.org/memory-interleaving/)

**30. Parallel computing**

-> It is the use of multiple processing elements simultaneously for solving any problem. Problems are broken down into instructions and are solved concurrently as each resource that has been applied to work is working at the same time.

Advantages of Parallel Computing over Serial Computing are as follows:

It saves time and money as many resources working together will reduce the time and cut potential costs.

It can be impractical to solve larger problems on Serial Computing.

It can take advantage of non-local resources when the local resources are finite.

Serial Computing ‘wastes’ the potential computing power, thus Parallel Computing makes better work of the hardware.

(https://www.geeksforgeeks.org/introduction-to-parallel-computing/)

**31. Give the features of ROM cell.**

-> Non-volatile in nature

Cannot be accidentally changed

Cheaper than RAMs

Easy to test

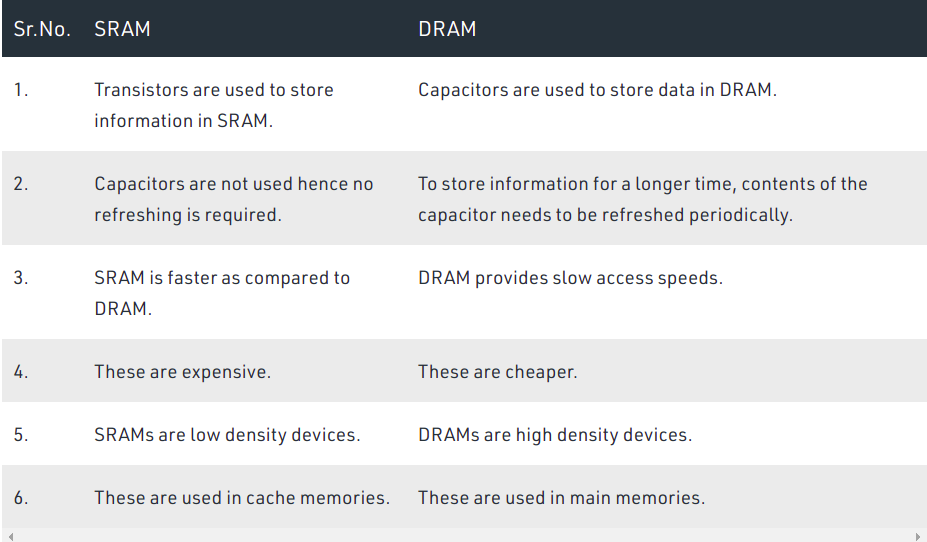
More reliable than RAMs

Static and do not require refreshing

Contents are always known and can be verified

**32. List the difference between static RAM and dynamic RAM.**

->



**33. What is disk controller?**

-> The disc controller is a circuit that allows the CPU to connect with a hard disc, floppy disc, or any other type of disc drive. It also serves as a link between the hard drive and the bus that connects it to the rest of the system.

**34. How a data is organized in the disk?**

->The mechanisms of disk drive technology are only half of the story; tile other half is the way data is structured on the disk. There is no way to plan for optimal storage configurations without understanding how data is structured on the surface of disk drive platters. This section discusses the following data structures used in disk drives:

Tracks, sectors, and cylinders

Disk partitions

Logical block addressing

Geometry of disk drives and zoned-bit recording

(https://www.snia.org/education/storage\_networking\_primer/stor\_devices/data\_structure#:~:text=Disk%20platters%20are%20formatted%20in,of%20data%E2%80%94typically%20512%20bytes.&text=Cylinders%20are%20the%20system%20of,multiple%20platters%20within%20the%20drive.)

**35. Flynns Classification**

-> Flynn's classification divides computers into four major groups that are:

Single instruction stream, single data stream (SISD)

Single instruction stream, multiple data stream (SIMD)

Multiple instruction stream, single data stream (MISD)

Multiple instruction stream, multiple data stream (MIMD)

(https://www.javatpoint.com/flynns-classification-of-computers)

**36. Performance evaluation on single processor and multi processor system.**

**37. Speed up and scale up**

-> Scaleup is the ability to keep the same performance levels (response time) when both workload (transactions) and resources (CPU, memory) increase proportionally.

**Scaleup = (Small system small problem elapsed time[single machine])/(large system large problem elapsed time[parallel machine])**

Speedup is the effect of applying an increasing number of resources to a fixed amount of work to achieve a proportional reduction in execution times:

**Speedup = (Small system elapsed time[single machine])/(large system elapsed time[parallel machine])**

**Section 6-**

1. **GPU?**

-> A graphics processing unit (GPU) is a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device. GPUs are used in embedded systems, mobile phones, personal computers, workstations, and game consoles.

1. **Thread?**

-> In computer science, a thread of execution is the smallest sequence of programmed instructions that can be managed independently by a scheduler, which is typically a part of the operating system.[1] The implementation of threads and processes differs between operating systems, but in most cases a thread is a component of a process. The multiple threads of a given process may be executed concurrently (via multithreading capabilities), sharing resources such as memory, while different processes do not share these resources. In particular, the threads of a process share its executable code and the values of its dynamically allocated variables and non-thread-local global variables at any given time

1. **CUDA Programming**
2. **GPU rendering**

-> GPU rendering uses a graphics card for rendering in place of a CPU, which can significantly speed up the rendering process as GPUs are primarily designed for quick image rendering. GPUs were introduced as a response to graphically intense applications that burdened CPUs and hindered computing performance.

GPU rendering takes a single set of instructions and runs them across multiple cores on multiple data, emphasizing parallel processing on one specific task while freeing up the CPU to focus on a variety of different sequential serial processing jobs. Rasterization, the rendering method used by all current graphics cards, geometrically projects objects in the scene to an image plane, which is an extremely fast process, but does not include advanced optical effects.

(<https://www.omnisci.com/technical-glossary/gpu-rendering>)

1. **GPU Performance evaluation**