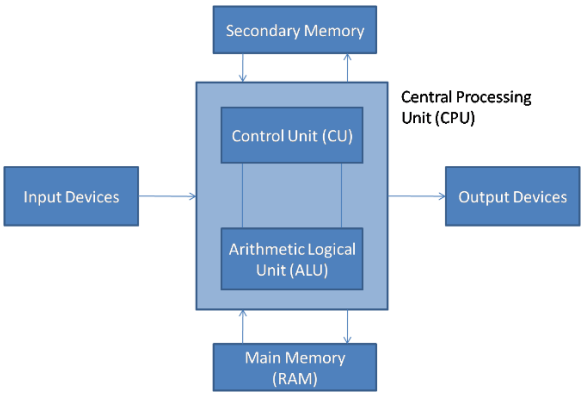
Marks-3

1. Define computer organization.

→ Computer organization:- Computer organization refers to the way components like CPU, memory, and input/output (I/O) devices are structured and connected within a computer system. It involves designing how data flows between these components to perform tasks efficiently. In simple words we can say, computer organization is the nuts and bolts of a computer system.

→An example of computer organization is the Central Processing Unit (CPU) fetching instructions from memory and sending them to the control unit for execution.



2. Explain the purpose & the function of the instruction "MOV A,M" in the 8085 microprocessor.

→ Purpose: The instruction MOV A, M in the 8085 microprocessor is used to copy the data from a memory location to the accumulator register.

Function:-

MOV: This is the mnemonic for the move operation.

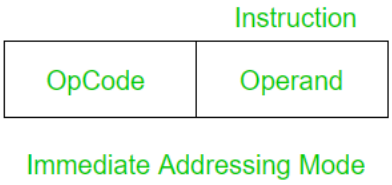
A: This represents the accumulator register.

M: This represents the memory location.

This instruction takes the data value stored at the memory address specified by the HL register pair. It then copies that data value into the accumulator register (A) of the 8085 microprocessor.

3. Describe immediate addressing mode. Give an example & explain how it operates using immediate addressing mode.

→Immediate addressing mode:- The immediate addressing mode is a method where the operand is specified directly within the instruction itself. It's like giving a command with all the information needed right there, without needing to fetch it from memory or another location. In immediate addressing mode, the operand is a part of the instruction.

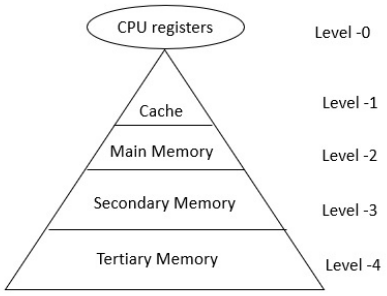


→ Example: MOV A, #42

In this example, MOV stands for “move”. A is the accumulator register. #42 is the immediate value. The operation is straightforward. The value 42 is loaded directly into the accumulator register (A). No memory access or additional calculations are involved.

4. Describe memory hierarchy.

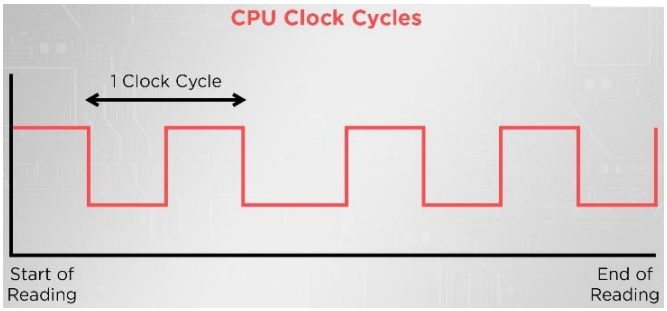
→ The memory hierarchy is a concept in computer architecture that illustrates the organization of different types of memory in terms of their speed, capacity, and cost. At the top of the hierarchy are registers and cache memory, which are small but extremely fast. Next are main memory and secondary storage, which are slower but have larger capacities. Finally, tertiary storage, such as external hard drives or cloud storage, offers even greater capacity but slower access speeds.



5. Describe a clock cycle in computer architecture.

→ A clock cycle, also known as a tick, is a basic time unit used in computer architecture. It is the time that elapses between two successive pulses of a system clock. In modern computer systems, the clock cycle is usually synonymous with a CPU cycle. In a clock cycle happens-

→During each clock cycle, the CPU fetches instructions from memory, decodes them to understand what to do, executes the instruction, and stores any resulting data back into memory or registers.



6. Difference between RAM and ROM.

→RAM (Random Access Memory) is volatile memory used for temporary data storage, while ROM (Read-Only Memory) is non-volatile memory used for permanent data storage.

→The differences:-

RAM	ROM
RAM is a volatile memory.	ROM is a non-volatile memory.
Read and write operations are supported.	Only read operations are supported.
It is a high-speed memory.	It is slower than RAM.
RAM is used in CPU, primary memory, etc.	ROM is used in micro-controllers, etc.
RAM is costlier than ROM.	ROM is not costly like RAM.

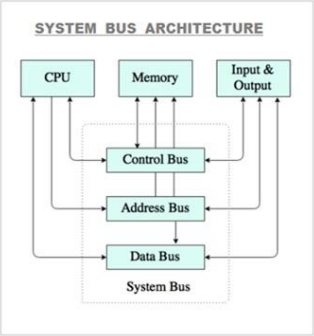
7. State the purpose of bus of a computer system.

→ In computer architecture, a bus or data bus is a communication system that transfers data between components inside a computer, or between computers. Here are the purposes of a bus in computer system:-

Data Transfer: The bus enables the transfer of data between the CPU, memory, and other peripheral devices.

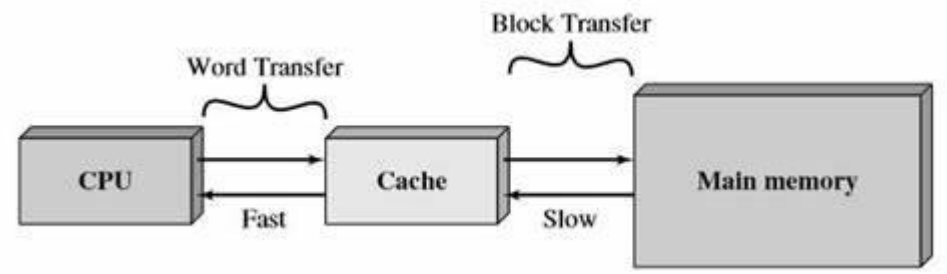
Addressing: The bus carries addressing information that indicates where data should be read from or written to.

Power: A bus supplies power to various peripherals connected to it.



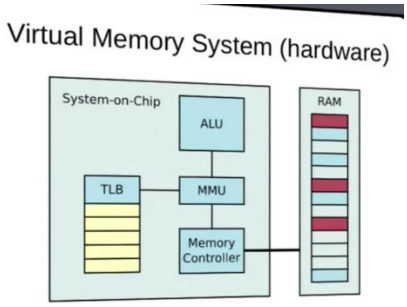
8. Explain cache memory.

→Cache Memory:- Cache memory is a small, high-speed memory located between the CPU and main memory in a computer system. Its purpose is to store frequently accessed data and instructions, reducing the time it takes for the CPU to access them. This improves overall system performance because accessing cache is significantly quicker than accessing main memory. The size of cache memory varies depending on the specific processor, but it typically ranges from a few kilobytes (KB) to a few megabytes (MB).



9. Role of TLB in virtual memory.

→ The full form of TLB is **Translation Lookaside Buffer**. TLB is a memory cache that speeds up virtual memory by storing recently used virtual-to-physical address translations.



The role:-

Speed: It accelerates memory access by providing quick translations, improving overall system performance.

Hardware vs. Software Management: TLBs can be managed by hardware for speed or by software for flexibility.

Security: The TLB helps enforce memory access permissions, contributing to system security.

10. Write in brief about the branch predictor in computer system.

→ Branch prediction is a technique used in CPU (central processing unit) design that attempts to guess the outcome of a conditional operation and prepare for the most likely result.

→A branch predictor is a clever trick CPUs use to speed things up. The branch predictor tries to guess which way the jump will go based on past behaviour. There are different types of branch predictors, but they all aim to improve the efficiency of the processor.

→ Example:- Imagine a loop: for (int i = 0; i < 10; i++) { ... }

The processor predicts the loop will continue and fetches instructions ahead. If the prediction is correct, the processor keeps running smoothly. If not, the processor discards fetched instructions and adjusts its prediction for next time.

11. Single issue vs multiple issue pipeline.

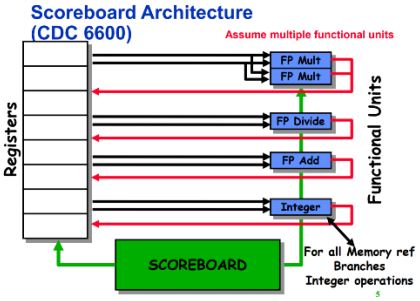
→ Single issue pipeline processes one instruction per clock cycle, while multiple issue can handle multiple instructions simultaneously.

The differences:-

Single issue pipeline	Multiple issue pipeline
It has 1 instruction per cycle.	It has 2 or more instructions per cycle.
It's hardware complexity is lower.	It's hardware complexity is higher.
It's power consumption is lower.	It's power consumption is higher.
It's performance is slower.	It's performance is faster.
It's overall dependency management is simple.	It's overall dependency management is more complex.

12. Purpose of score board in computer architecture.

→ In computer architecture, a scoreboard is a hardware component used in multiple issue pipelines to improve efficiency.



The purposes:-

Detecting hazards: It identifies conflicts between instructions (like using the same data) to prevent errors.

Scheduling instructions: It determines when instructions are ready to execute based on their dependencies and available resources.

Status Monitoring: Tracks the status of instructions and resources.

13. Pipeline in computer architecture.

→ A pipeline in computer architecture is a technique for improving the performance of a processor. It works by breaking down a complex instruction into smaller, simpler steps.

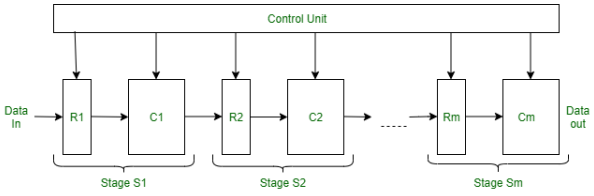


Figure - Structure of a Pipeline Processor

→ How it works:

Instruction Fetch: The pipeline fetches the instruction from memory.

Decode: The instruction is decoded to understand what operation needs to be performed.

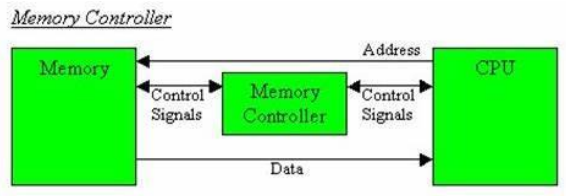
Execute: The operation specified in the instruction is executed (e.g., addition, multiplication).

Memory Access (if needed): If the instruction requires data from memory, it's accessed during this stage.

Write Back (if needed): The result of the operation is written back to a register or memory.

14. Define memory controller in computer system.

→ A memory controller, also known as a memory chip controller (MCC) or memory control unit (MCU), is a crucial component in a computer system.



→How it works:-

Manages Data Flow: The memory controller regulates the flow of data going to and from the RAM. It receives data requests from the CPU, locates the data in memory, and ensures it's delivered accurately.

Communication Hub: It acts as a central point of communication between the CPU, RAM, and other devices that need to access memory.

Optimizes Performance: The memory controller plays a significant role in overall system performance.

15. Explain the workflow of CPU.

→ CPU: The full form of CPU is Central Processing Unit. A CPU is a hardware that performs data input/output, processing and storage functions for a computer system.

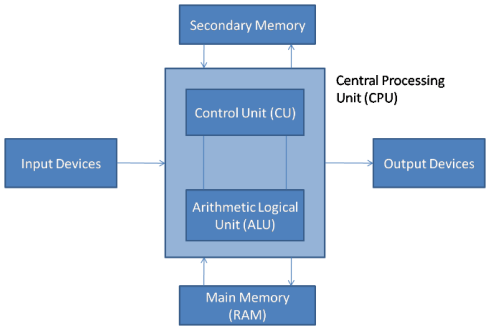
→ The workflow:-

-The CPU fetches instructions from memory.

-It decodes these instructions to understand them.

-The CPU then executes the decoded instructions.

-Finally, it may store the result back into memory or registers.



Here is the diagram of CPU.

16. Explain the significant of timing & control in digital system.

→ Timing and control are fundamental concepts in any digital system, from simple circuits to complex computer processors.

→The significant:-

- Digital systems rely on timing signals for synchronization, to avoid errors.
- Together, they guarantee reliable operation and minimize malfunctions.
- They work together to ensure the system operates correctly and efficiently.
- Optimizing timing and control can significantly improve performance.

17. Explain the effect of instruction set on the performance of a computer system.

→ The Instruction Set Architecture(ISAs) of a computer system plays a significant role in determining the performance of the system. It is a set of commands that a processor can execute, and acts as an interface between software and hardware. Here's how the instruction set affects performance:

Simple ISAs: These instruction sets focus on providing basic operations that the processor can execute very quickly.

Complex ISA: These instruction sets offer a wider range of more complex instructions that can perform multiple operations in a single instruction.

18. What is the purpose of a control unit?

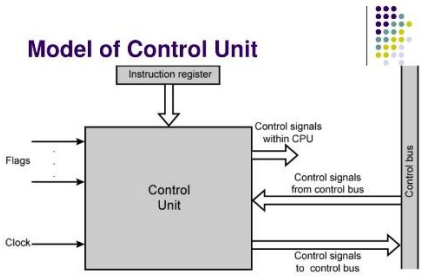
→ The Control Unit (CU) directs the CPU's operations by generating control signals to manage the execution of instructions and data flow.

→The purposes:-

Fetches instructions: It retrieves instructions from memory one at a time.

Decodes instructions: Once fetched, the control unit deciphers the instruction to understand what operation needs to be performed.

Executes instructions: The control unit doesn't directly perform the operations itself. Instead, it sends control signals to other parts of the CPU (like the ALU) telling them what to do with the data based on the decoded instruction.



19. Explain the workflow of a virtual memory.

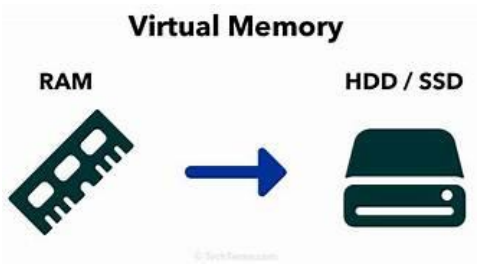
→ Virtual Memory is a storage allocation scheme. Virtual memory creates the illusion of a much larger main memory (RAM) than physically exists on your computer.

→The workflow:-

Divided Memory Spaces: Virtual memory creates two separate address spaces: virtual and physical. Programs use virtual addresses, while RAM uses physical addresses.

Increase Physical RAM: The more RAM you have, the fewer page faults you'll experience.

Optimize Applications: Software developers can write programs that utilize memory efficiently, reducing the need for virtual memory.



20. Explain the role of DMA in computer system.

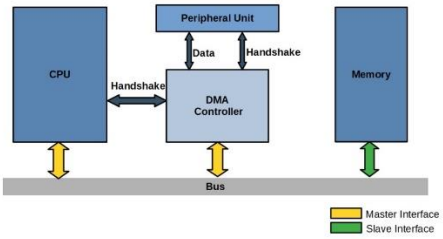
→ DMA (Direct memory address) is the special feature within the computer system that transfers the data between memory and peripheral devices (like hard drives) without the intervention of the CPU.

→Role of DMA in Computer Systems:-

Efficient Data Transfer: DMA enables high-speed data transfer between peripherals and memory. DMA can transfer data in large blocks, making it more efficient.

Memory-to-Memory Transfer: DMA can be used for memory-to-memory transfers, allowing data to be moved directly within memory.

Improved System Performance: DMA can handle data transfers more quickly than CPU-driven transfers, improving the responsiveness.



21. Compare between synchronous and asynchronous D-RAM.

→ DRAM:- Dynamic Random Access Memory (DRAM) is a type of volatile memory used in computers for storing data and program instructions temporarily.

→Comparison:-

i. Performance:-

-SDRAM is faster and more efficient due to synchronization.

-Async DRAM is slower and less efficient, can lead to wait times.

ii. Communication Style:-

-In SDRAM, CPU and memory communicate based on clock cycles.

-Async DRAM, CPU and memory communicate with independent signals.

iii. Usage:-

-SDRAM is used in Modern high-speed PCs.

-Async DRAM is used in older low-speed PCs.

22. Explain the concept of general register organization in computer architecture.

→ General Register Organization is the processor architecture that stores and manipulates data for computations. The main components of a register organization include registers, memory, and instructions. The registers act as memory within the processor and are used to process instructions as they are executed.

→For example, adding two numbers might involve moving them from registers to the ALU, performing the addition, and storing the result back in a register.

23. Illustrate how does a beach predictor work in a pipeline processor.

→ A branch predictor is a clever trick CPUs use to speed things up. The branch predictor tries to guess which way the jump will go based on past behaviour.

How it works:-

Branch Prediction Types: Modern processors use dynamic branch prediction techniques. These accumulate data during runtime, utilizing, make accurate predictions.

Pipeline Efficiency: By predicting branches, the pipeline processor avoids stalls caused by control hazards & increased efficiency.

Multi-Tiered Predictors: Some processors employ multi-tiered branch predictors, combining different prediction techniques to enhance accuracy.

24. Explain the role of a memory management unit(MMU) in a virtual memory system.

→ The memory management unit (MMU) translates virtual addresses used by programs into physical addresses in memory.

Role:-

Address Translation: The MMU converts virtual addresses generated by the CPU during program execution into physical addresses.

Virtual Memory Management: The MMU enables efficient utilization of secondary storage & manages the virtual memory.

Multitasking Capacity: In modern operating systems, the MMU does seamless multitasking by managing memory allocation and address translation.

25. What are the advantages of using cache memory in computer system?

→ Cache memory is a small, high-speed memory located between the CPU and main memory in a computer system.

Advantages:-

Increased CPU Utilization: Cache memory allows the CPU to operate at a higher effective speed, improving overall system efficiency.

Reduced Memory Latency: It minimizes the latency for accessing data, enhancing overall system performance.

Faster Access: Cache memory is faster than the main memory, reducing the time needed to retrieve data.

26. Why the size of cache memory is much lesser than the main memory?

→ Cache memory is a small, high-speed memory located between the CPU and main memory in a computer system. Its purpose is to store frequently accessed data and instructions, reducing the time it takes for the CPU to access them. This improves overall system performance because accessing cache is significantly quicker than accessing main memory. The size of cache memory varies depending on the specific processor, but it typically ranges from a few kilobytes (KB) to a few megabytes (MB).

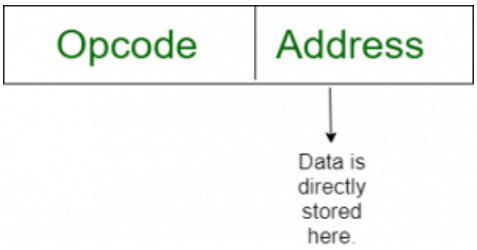
→ Cache memory is much smaller than main memory due to its primary function of providing faster access to used data and instructions. This smaller size allows cache memory to be faster and more expensive per byte compared to main memory, which is larger but slower.

27. Recall how many addressing modes are there and explain with an example.

→Addressing modes in computer architecture refer to the techniques and rules used by processors to calculate the effective memory address for data operations. Here are some of the common addressing modes:-

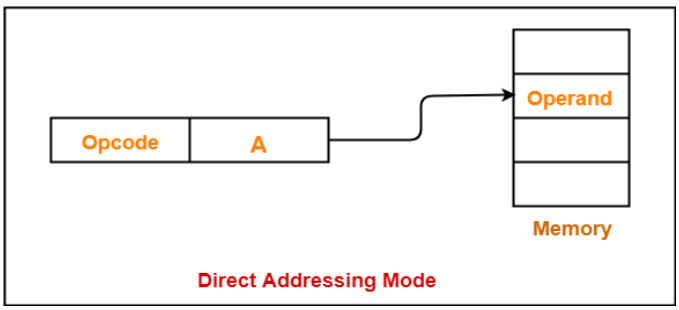
Immediate addressing mode:- This addressing mode is a method where the operand is specified directly within the instruction itself.

For example, in the instruction "ADD B, #10", the value 10 is immediately available for addition to the contents of register B.



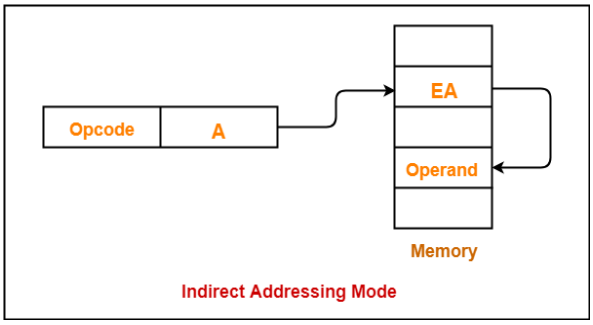
Direct Addressing Mode: **This** address of the operand is explicitly provided in the instructions.

For example, in this instructions “ADD AL,[0301]”, refers to add the contents of offset address 0301 to AL.



Indirect addressing: In this mode instruction holds address to memory location containing operand's address.

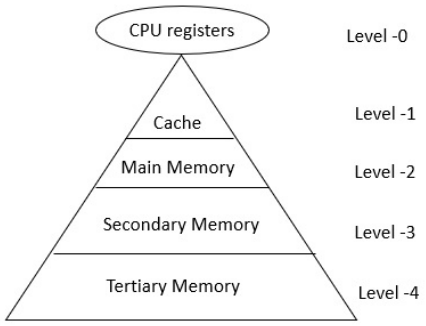
For example, an instruction that loads the value from a memory location whose address is stored in a register.



28. Illustrate memory hierarchy with diagram.

→ The memory hierarchy is a concept in computer architecture that illustrates the organization of different types of memory in terms of their speed, capacity, and cost. At the top of the hierarchy are

registers and cache memory, which are small but extremely fast. Next are main memory and secondary storage, which are slower but have larger capacities. Finally, tertiary storage, such as external hard drives or cloud storage, offers even greater capacity but slower access speeds.



29. Distinguish between PROM & EPROM.

→ PROM (Programmable Read-Only Memory) is a type of ROM that is written only once. While EPROM (Erasable Programmable Read-Only Memory) is a type of ROM that is read and written optically.

→The differences:-

PROM	EPROM
PROM stands for Programmable Read-Only Memory.	EPROM stands for Erasable Programmable Read-Only Memory.
PROM is not reusable.	EPROM is reusable multiple times.
PROM is not expensive.	EPROM is costlier than PROM.
PROM is the older version of EPROM.	EPROM is the modern version of PROM.
It is more flexible.	It is less flexible.

30. Define why virtual memory is needed in a computer system.

→Virtual Memory: Virtual memory is a memory management technique that uses secondary storage as an extension of RAM.

Why needed:-

Memory Shortages: Computers often run multiple programs simultaneously, each requiring memory. Virtual memory extends available memory by using secondary storage as an extension of RAM.

Program Size: Virtual memory allows loading large programs that exceed physical RAM. It creates an illusion of unlimited memory for users.

Memory Protection: Virtual memory provides security and prevents unauthorized access.

31. Difference between virtual memory & cache memory.

→Virtual memory provides programs into having more RAM, while cache memory speeds up access to frequently used data.

→The differences:-

Virtual Memory	Cache Memory
Virtual memory increases the capacity of main memory.	While cache memory increase the accessing speed of CPU.
Operating System manages the virtual memory.	Hardware manages the cache memory.
It is slower than cache memory.	It is faster.
Virtual memory is not a memory unit, its a technique.	It is a memory unit.
The size of virtual memory is greater than the cache memory.	It has the less size.

32. Briefly explain parallel processing.

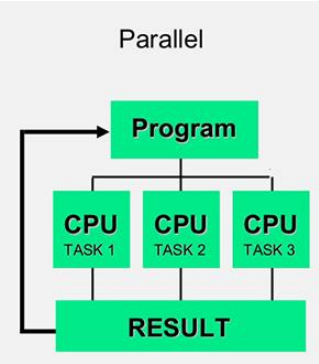
→ Parallel processing is used to increase the computational speed of computer systems by performing multiple data-processing operations simultaneously.

→It does some tasks:-

Task division:- Parallel processing solves large problems by dividing them into smaller. These are then solved simultaneously by multiple processors.

Speedup: Parallel processing speeds up computation, improving overall performance.

Applications: Parallel processing is used in fields like scientific simulations, graphics rendering, and machine learning.



33. Briefly explain maskable & non maskable interrupts.

→**Maskable Interrupts**: Maskable interrupts are those interrupts that can be turned off or ignored by the instructions of the microprocessor.

Usage: Commonly used for less critical tasks, such as I/O operations.

Non-Maskable Interrupts: Non-maskable interrupts are those that cannot be turned off or can't be ignored by the microprocessor.

Usage: Typically used for power failure, system crash, or other catastrophic events.

In summary, Maskable interrupts can be turned off, while non-maskable interrupts always demand immediate attention. Both types of interrupts are crucial for efficient and safe system operation.

34. Determine/ define instruction hazards.

→ Instruction hazards arise in pipelined computer architectures. Pipelining breaks down instruction execution into stages like fetching, decoding, and execution.

There are three main types of instruction hazards:

Structural Hazards: These occur when multiple instructions compete for the same hardware resource within a pipeline stage.

Data Hazards: These hazards happen due to dependencies between instructions that use the same data.

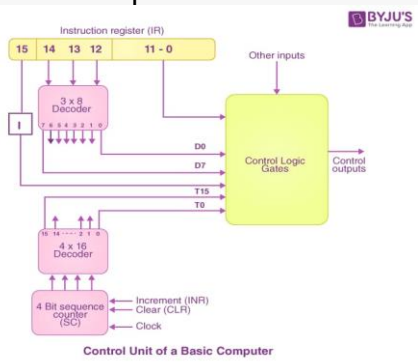
Control Hazards: Control hazards, also known as branch hazards, occur when the pipeline makes wrong decisions on branch prediction.

35. Extend briefly the design of a hardware control unit.

→ A hardware control unit is responsible for directing the flow of data and instructions within the CPU. Here's a brief extension of the design:

Hardwired Control Unit: Uses a fixed set of logic gates and circuits to execute instructions. It typically includes decoders, counter, and logic gates to interpret and execute instructions.

Micro-programmed Control Unit: Utilizes a set of microinstructions stored in memory to execution of instructions.
Overall, the design choice depends on factors like speed, complexity, and flexibility required for the CPU's operations.



Control Unit of a Basic Computer

36. Discuss the influence of a pipelining on instruction set design.

→ In computer architecture, pipelining is a technique to speed up instruction execution by breaking it down into smaller stages. Pipelining can significantly influence the design in some ways:

Simplicity of Instructions: Pipelining favours a simple set of instructions that can be easily decoded and executed in a pipeline.

Minimizing Pipeline Hazards: The instruction set should be designed to minimize pipeline hazards such as data hazards.

Reduced Instruction Set: A reduced instruction set (RISC) can make pipelining more efficient by minimizing the complexity of operations.

37.Define clocked sequential circuits.

→In computer architecture, a clock sequential circuit is a digital circuit that relies on a synchronized clock signal to control its operation.

→ The key points about clocked sequential circuits:-

Memory Elements: Clocked sequential circuits include memory elements (such as flip-flops) that store binary information. These memory elements maintain their state.

Clock Signal: A periodic clock signal is connected to the clock inputs of all memory elements in the circuit. The clock synchronizes all internal state changes.

38. Explain cache write through & cache write back.

→Write-Through and Write-Back are two strategies for handling write operations in a cache memory system. Here's the breakdown of each:-

Cache Write-Through: In the Write-Through method, when the processor writes data to the cache, it also writes that data to the main memory simultaneously.

Cache Write-Back: In the Write-Back method, when the processor writes data to the cache, it does not immediately write that data to the main memory.

In summary Write-Through ensures data consistency but can be slower, while Write-Back can be faster but requires more complexity to maintain data consistency.

39. Describe the method for dealing with instruction hazards.

→ same as 34

40. Illustrate different types of array processions.

→ Array processing refers to operations that manipulate arrays. In computing, there are several types of array processing operations, each serving different purposes. Here are some common types:-

Element-wise Processing: Each element in the array is processed individually. For example, multiplying each element in an array by a constant.

Vectorized Operation: This involves applying simultaneously to multiple elements using vectorized instructions.

Reduction: This involves applying a binary operation (like addition, multiplication) to all the elements of an array to produce a single output.

41. Discuss the function of ALE & READY pin in 8085 microprocessor.

→The 8085 is an 8-bit microprocessor that simplifies computer design by requiring less supporting hardware.

→Function of ALE:- ALE is an important control signal in the 8085 microprocessor. Its primary function is to enable the lower 8 bits of the address bus. When ALE is active, the lower 8 bits of the address are available on the address bus. If ALE is not active, the data bus is activated instead.

→Function of READY:- The READY pin is used for interfacing with peripheral devices. If the READY pin is high (1), it indicates that the peripheral is ready. If the READY pin is low (0), the microprocessor waits until it goes high. READY is particularly useful for interfacing with low-speed devices.

42. Explain reduced instruction set computer (RISC).

→RISC:- A RISC (Reduced Instruction Set Computer) is a type of computer architecture designed for efficiency and speed. RISC is a processor that focuses on doing simple things very quickly.

→Characteristics:-

-Simpler instruction.

-More general-purpose registers.

-Simple Addressing Modes.

-Fewer Data types.

43. Discuss the address capacity of 8085 microprocessor in 64 kb.

→ The 8085 is an 8-bit microprocessor that simplifies computer design by requiring less supporting hardware.

→ The 8085 microprocessor has a 16-bit address bus (A0-A15), allowing it to address a total of 65,536 memory locations. Each unique address corresponds to an 8-bit or 1-byte memory space. Therefore, the maximum addressable memory capacity is 64 KB (2^16 bytes). This is because 1 Kb equals 2^10 bytes.

44. Justify the various registers of 8085 microprocessor.

→ The 8085 is an 8-bit microprocessor that simplifies computer design by requiring less supporting hardware.

→ The various registers in the 8085 microprocessor:

Accumulator (A) Register: The accumulator is an 8-bit register. It plays a central role in arithmetic and logical operations.

General-Purpose Registers (B, C, D, E, H, L): These six 8-bit registers are used for temporary data storage.

Flag Register (F): The flag register contains status flags (bits) that reflect the outcome of arithmetic and logical operations.

Program Counter (PC): The program counter keeps track of the memory address of the next instruction to be executed.

Stack Pointer (SP): The stack pointer points to the top of the stack in memory.

45. Identify the various hardware interrupts supported by the 8085 microprocessor.

→ The 8085 is an 8-bit microprocessor that simplifies computer design by requiring less supporting hardware. The 8085 microprocessor supports hardware interrupts:-

TRAP: A non-maskable interrupt generated by external devices. It has the highest priority and cannot be disabled.

RST 7.5, RST 6.5, RST 5.5 (Maskable): Can be turned on or off by the program, used for various external device requests.

INTR: A maskable interrupt from external devices like keyboards or mice. It has the lowest priority and can be disabled.

46. Distinguish HLDA & HOLD pin in 8085 microprocessor.

→ The 8085 is an 8-bit microprocessor that simplifies computer design by requiring less supporting hardware. The differences:-

HOLD	HLDA
The HOLD pin is an input signal.	The HLDA pin is an output signal.
Its function is to request control of buses.	Its function is to acknowledges HOLD request.
It stays high until HOLD removed.	It goes low after HOLD removed.
It has lower priority than internal operations.	It has lower priority than HOLD.

47. Identify the triggering of flipflop. (Triggering method).

→ The triggering of a flip-flop refers to the event that causes it to change its output state. There are two main types of triggering:-

→ Asynchronous Triggering (Level Triggering): In this type, the flip-flop responds directly to the changes in its input signals. It relies on the actual level of the clock signal. It is commonly used in asynchronous circuits.

→ Synchronous Triggering (Edge Triggering):- This is the most common type used in modern digital circuits. In edge triggering, a flip-flop responds to a specific transition on its clock input. Edge-triggered flip-flops are activated by a change in the clock signal from one logic level to another.

48. Evaluate the performance improvement of a computer system by using cache memory.

→ Cache memory is a small, high-speed memory located between the CPU and main memory in a computer system. Cache memory plays a crucial role in enhancing the performance of a computer system. How it achieves this improvement:-

Reduced Access Time: Cache memory is significantly faster than main memory (RAM) because it's built with faster SRAM technology and resides closer to the processor.

Effective CPU Utilization:- Cache memory allows the CPU to operate at a higher effective speed. As a result, the CPU utilization improves.

Reduce Bus Traffic:- This reduces bus traffic and minimizes contention for memory access.

49. Organise the performance ALU 8085 microprocessor.

→ In the 8085 microprocessor, the ALU (Arithmetic Logic Unit) is the heart of its computational power. The performances:-

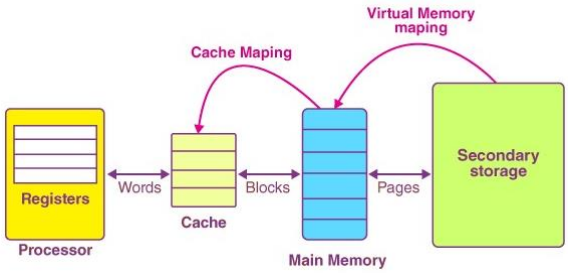
Arithmetic Operations: The ALU performs fundamental arithmetic operations, like addition and subtraction. It operates on 8-bit variables.

Logical Operations: The ALU handles logical operations such as complement, AND, OR, and EX-OR. Logical operations are crucial for bitwise manipulation.

Bit-Shifting Operations: The ALU supports bit-shifting operations like left shift and right shift. Bit shifts are used for data, multiplication, and division by powers of 2.

50. Explain the different types of mapping techniques in cache memory.

→ Cache mapping techniques determine how data is organized in cache memory. There are 3 types:- direct mapping, associative mapping, set-associative mapping.



Direct-Mapped: Simplest mapping technique with straightforward hardware implementation. It is fast, no search needed. It has lower cost due to simplier design.

Associative: Most complex mapping technique. It is slow and needs to search all entities. It has higher cost due to complex search logic.

Set-Associative: Moderately complex, combining aspects of both direct-mapped and fully associative caches. It is moderate, depends on number of sets to search. It has moderate cost.

5 MARKS

1. Explain arithmatic shift right, circular shift right, logical shift left operation. Suppose register 'A' holds the 8 bit number 11011001. Determine the sequence of binary value in 'A' after and arithmatic shift right followed by a circular shift right and followed by a logical shift left.

→ Arithmetic Shift Right:- Shifts all bits to the right. The leftmost is replicated, while the rightmost bit is discarded.

Circular Shift Right (Rotate Right): Shifts all bits to the right. The bit that is shifted out of the rightmost position is placed into the leftmost position.

Logical Shift Left: Shifts all bits to the left, and discards the leftmost bit & fills with (x).

→ Initial Value of Register A: 11011001

i. Arithmetic Shift Right

- Initial: 11011001
- After Arithmetic Shift Right: 1 1 0 1 1 0 0

ii. Circular Shift Right

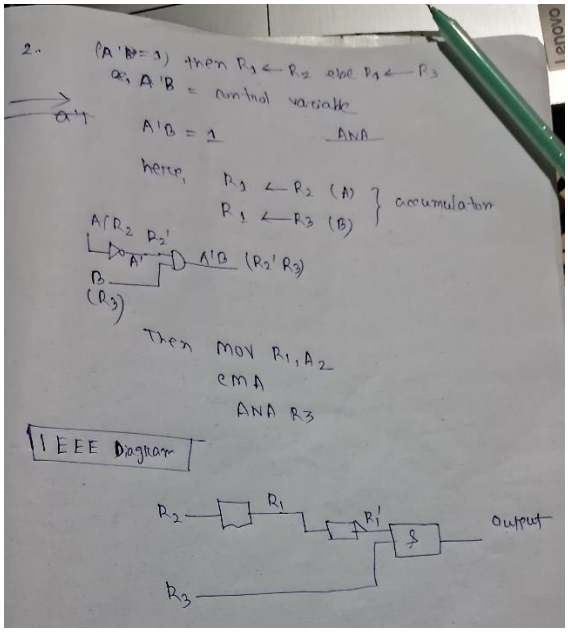
- Initial: 11011001
- After Circular Shift Right: 11101100

iii. Logical Shift Left:

- Initial: 11011001
- After **Logical Shift Left: 1 0 1 1 0 0 1 x**

2. Explain the circuit diagram : if(a'b=1) then R1← R2 else R1←R3, a'b=control variable.

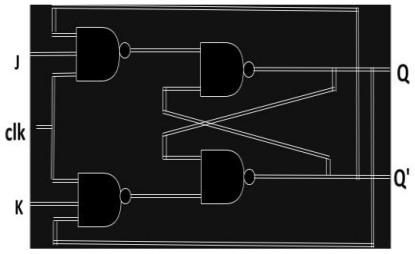
→



4. Explain the race around condition in details.

→The race-around condition is a specific issue in JK flip-flops, a digital circuit. It occurs when both J and K inputs are set to 1 and the clock pulse stays high for too long.

This creates a feedback loop where the output influences the inputs, causing the flip-flop's output (Q) to unpredictably toggle as long as the clock remains high.



J	K	CLK	Q	Q’
0	0	1	Unchanged	State
0	1	1	0	1
1	0	1	1	0
1	1	1	Toggle	State

For J-K flip-flop, if J=K=1, and if clk=1 for a long period of time, then output Q will toggle as long as CLK remains high which makes the output unstable or uncertain.

This is called a race around condition in J-K flip-flop.

We can overcome this problem by making the clock =1 for very less duration.

The circuit used to overcome race around conditions is called the Master Slave JK flip flop.

5. Illustrate logic and shfit micro operations.

→Logic micro-operations are like tiny switches in the CPU. They perform bit-wise AND, OR, and NOT operations directly on data in registers.

Basic building blocks: Logic micro-operations are fundamental operations performed on data within the CPU.

Bit-wise manipulation: They work directly on individual bits (0s and 1s) in registers.

Controlling data flow: These operations determine if a bit is turned on (1) or off (0) based on the logic applied.

→Shift micro-operations are those micro-operations that are used for the serial transfer of information.

Logical Shift: It transfers the 0 zero through the serial input. We use the symbols '<<' for the logical left shift and '>>' for the logical right shift.

Arithmetic Shif:- The arithmetic shift micro-operation moves the signed binary number either to the left or to the right position.

i. Arithmetic Left Shift

ii. Arithmetic Right Shift

Circular Shift:-The circular shift circulates the bits in the sequence of the register around both ends without any loss of information. The two ways to perform the circular shift:-

Circular Shift Left

Circular Shift Right

6. Illustrate the instructions of 8085 microprocessor with example:- SPHL, SCHG, MOV, MD, DAA, RAR.

→SPHL (Stack Pointer Load): This instruction loads the stack pointer (SP) with the contents of the HL register pair.

Eg.:- If H = 0xCA, after execution, SP becomes 0xCA00

XCHG (Exchange): This instruction exchanges the contents of HL register pair with DE register pair.

Example:

Before: HL = 0xABCD, DE = 0x1234

After: HL = 0x1234, DE = 0xABCD

MOV (Move): Copies data from one register or memory location to another.

Example: MOV A, B (Copy the content of register B to register A).

MD (Memory Data): This instruction moves data between the accumulator and a memory location.

Eg:- MD 2050H

DAA (Decimal Adjust Accumulator): This instruction adjusts the contents of the accumulator to form two packed decimal digits after addition or subtraction.

Example: Let, add two BCD numbers: MOV A, 0x39 ; ADD A, 0x26 ; DAA.

Before: A = 0x39

After BCD addition: A = 0x65

After DAA: A = 0x65

RAR (Rotate Accumulator Right): This instruction rotates the contents of the accumulator to the right through the carry flag.

Example: MOV A, 0xCA ; RAR

Before: A = 0xCA (binary: 11001010)

After: A = 0x5D (binary: 01011001)

7. Assemnly laguage prog to add two 8 bits nos.

→

8. Explain the timing diagram for STA 526 AH.

→ STA means Store Accumulator. The content of the accumulator is stored in the specified address (526A).

-The op-code of the STA instruction is said to be 32H.

- Then the lower order memory address is read (6A). – Memory Read Machine Cycle

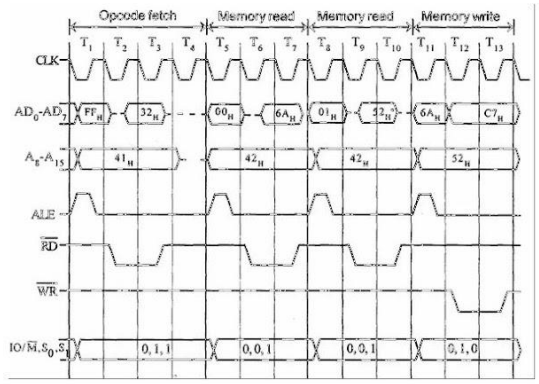
-Read the higher order memory address (52).- Memory Read Machine Cycle.

- The combination of both the addresses is considered and the content from accumulator is written in 526A.

-Assume the memory address for the instruction and let the content of accumulator is C7H. So, C7H from accumulator is now stored in 526A.

Address	Memories	Hex-code
41FFH	STA	32H
4200H	6AH	6AH
4201H	52H	52H

526AH	C7H	C7H



9. Write an assembly language program to sort the numbers in accending order.

→

10. What are the registers reference instructions? Explain.

→Registers reference instructions in the context of the 8085 microprocessor are instructions that perform operations directly on the various registers available within the processor. Some common examples of register reference instructions in the 8085 microprocessor include:-

ADD (Addition): Adds the contents of a specified register to the accumulator. For example, ADD B adds the contents of register B to the accumulator.

SUB (Subtraction): Subtracts the contents of a specified register from the accumulator. For example, SUB C subtracts the contents of register C from the accumulator.

INR (Increment): Increments the contents of a specified register by one. For example, INR D increments the contents of register D by one.

DCR (Decrement): Decrements the contents of a specified register by one. For example, DCR E decrements the contents of register E by one.

MOV (Move): Copies data from one register to another. For example, MOV A, B copies the contents of register B to register A.

11. Define the types of interrupts and model the interruptcycle.

→The interrupt is a signal emitted by hardware or software when a process or an event needs immediate attention. The 8085 microprocessor supports two main types of interrupts:-

Hardware Interrupts: These are initiated by external devices connected to the microprocessor. The devices signal the processor through dedicated interrupt lines, requesting service.

Software interrupts: Software interrupts are generated by executing a specific instruction in the program. These are often used for system calls within the software. In the 8085 microprocessor, software interrupts are called using the RST (Restart) instructions.

→Interrupt Cycle Model (8085)

Interrupt Recognition: Processor checks for interrupts after completing the current instruction. If an interrupt is detected, the processor acknowledges it.

Saving the Current State: Save the Program Counter (PC) on the stack. Save other necessary registers.

Fetch Vector: Get ISR address (vectored) or determine it (non-vectored).

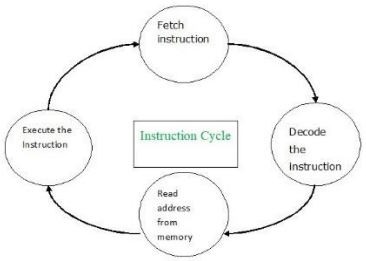
Executing the ISR: Execute the instructions in the ISR to handle the interrupt.

Restoring the State: Restore the PC and other registers from the stack.

Resuming Execution: Resume normal program execution from where it was interrupted.

12. Draw and explain the flowchart of instruction cycle.

→ Instruction Cycle: The structure of the instruction cycle defines the processing of a single instruction. It's a continuous loop that fetches, decodes, and executes instructions one by one. Here's a flowchart explaining the steps:-



Start

Fetch:- The Program Counter (PC) register holds the memory address of the next instruction to be executed. The CPU reads the instruction from memory at the address pointed to by the PC. The fetched instruction is loaded into the Instruction Register (IR).

Decode: The Control Unit (CU) analyzes the instruction in the IR to determine its operation (opcode) and any operands it may require. The CU decodes the instruction and generates control signals.

Execute: Based on the decoded instruction, the Arithmetic Logic Unit (ALU) performs calculations, the data transfer unit moves data between registers and memory, or the control unit handles control flow operations.

Repeat: Once the execution is complete, the cycle starts again, fetching the next instruction from memory and repeating the process.

End

13. Explain memory interfacing with help of an example.

→Memory interfacing refers to the process of connecting memory devices (like RAM) to the CPU so they can exchange data.

→Components Involved:

Memory: This is the actual storage device (e.g., RAM module) that holds the data.

Memory Address Register (MAR): This CPU register stores the memory address of the data being accessed.

Address Bus: This set of wires carries the memory address from the MAR to the memory device.

Data Bus: This set of wires transfers data between the MAR and memory.

Control Lines: These signal lines manage the data transfer process, indicating read/write operations.

Example:- Imagine the CPU wants to read a value from a specific memory location. The CPU places the address (1010) on the address bus. The CPU asserts a control signal indicating a "read" operation. The memory chip locates the data at address 1010 and places it on the data bus. The CPU reads the data from the data bus and stores.

14. Explain the storage structure of a ROM storage cell and the read and write operation.

→ Storage Structure of a ROM Cell:

- A ROM (Read-Only Memory) is a type of non-volatile memory used to store important information permanently.
- Unlike RAM (Random Access Memory), which allows both reading and writing, ROM is read-only.
- ROM cells are organized into a matrix structure.
- The internal structure of a ROM includes two main components:-
  - **Decoder**: Converts encoded input into a decimal form.
  - **OR gates**: These gates provide the output data based on the selected address.

To read data from a ROM:-

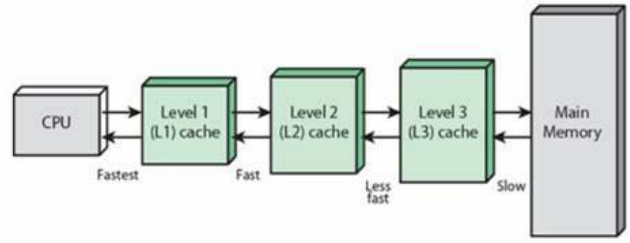
- An address is provided via input lines (usually binary).
- The decoder translates this address into a specific location within the ROM.

Write Operation:-

- Unlike RAM, ROM does not support writing or altering data.
- Once programmed, the data remains fixed, making ROM ideal.

15. Explain the concept of cache levels.

--> Cache Memory: A smaller, faster type of volatile computer memory that provides high-speed data access to the processor.



L1 Cache (Level 1): The smallest and fastest cache, typically located directly on the CPU core itself. It holds a very limited amount of data (few kilobytes) but offers the quickest access time. L1 cache is often divided into separate caches for instructions and data.

L2 Cache (Level 2): Larger than L1 cache (tens of kilobytes) and slightly slower. It acts as a buffer between L1 and main memory. Frequently accessed data not found in L1 might be found in L2.

L3 Cache (Level 3) (optional): The largest and slowest cache on the CPU hierarchy (typically hundreds of kilobytes to a few megabytes). It further reduces the need to access main memory by storing frequently used data from both L1 and L2 misses. Not all CPUs have L3 cache.

16. Compare between primary, secondary & tertiary memory.

→ Primary (RAM) holds active data super fast, but forgets it when off. Secondary (hard drives) stores data permanently, slower than RAM. Tertiary (tape libraries) are massive, archives for rarely used data.

→ Compare between primary, secondary & tirtiary memory:-

Primary Memory	Secondary Memory	Tertiary Memory
It is also known as Main memory (RAM).	It is also known as external storage, mass storage.	It is also known as Offline storage, Nearline storage.
It is volatile.	It is non-volatile.	It is non-volatile.
It is the fastest.	It is slower than primary memory.	It is slowest.
It is more costly.	It is less costlier than primary memory.	It is less costlier.
It is used for store data and programs.	It is used for store data and programs for long-term storage.	It is used for archive rarely accessed data.

17. Estimate the average access time & efficiency of the memory system with a hierarchical cache main memory sub-system processing cache access time of 160 ns main memory access time of 960 ns & hit ratio of cache is 0.9.

→
- Cache hit ratio: 0.9
- Cache access time: 160 ns
- Main memory access time: 960 ns

Average Access Time:-

Cache Hit Time = Hit Ratio * Cache Access Time = 0.9 * 160 ns = 144 ns

Cache Miss Time = (1 - Hit Ratio) * Main Memory Access Time = (1 - 0.9) * 960 ns = 96 ns

Average Access Time = (Hit Ratio * Hit Time) + ((1 - Hit Ratio) * Miss Time) = (0.9 * 144 ns) + ((1 - 0.9) * 96 ns)

= 129.6 ns + 9.6 ns = 240.0 ns (approximately)

Efficiency = Cache Hit Ratio * 100% = 0.9 * 100% = 90.0%

18. How many addressing lines are needed to access RAM chips arranged in a 4 x 6 arrays, where each chip is 8k x 4 bits if each address space represent 1 byte of storage space?

→ Each chip is 8K x 4 bits.

- 8K bits = 8 * 1024 bits = 8192 bits.
  - 8192 bits = 1024 bytes (since 1 byte = 8 bits).
- Total number of chips = 4 * 6 = 24 chips.
- Total storage = Number of chips * Storage per chip
  - Total storage = 24 chips * 1024 bytes/chip = 24,576 bytes.
  - Number of address lines = $\log_2$(Total storage)
  - Number of address lines = $\log_2(24{,}576) \approx 14.96$ (rounded up to 15).

Therefore, the number of address lines needed to access the RAM chips arranged in a 4 x 6 array, where each chip is 8K x 4 bits, is **15**

19. Distinguish between RISC & CISC.

→ **RISC (Reduced Instruction Set Computer):** A CPU design strategy with a small set of simple instructions aiming for higher performance.

CISC (Complex Instruction Set Computer): A CPU design strategy with a large set of complex instructions to accomplish tasks in fewer lines of assembly code.

Differences:-

RISC	CISC
It has smaller set of simpler instructions.	It has larger set of more complex instructions.
It has Fixed-length instructions.	It has Variable-length instructions.
Simpler hardware design.	Complex hardware design.
It is used in Mobile devices, embedded systems.	It is used in Desktop computers, legacy systems.
Generally lower power consumption.	Generally higher power consumption.

20 / 29 / 47. Discuss the advantages & disadvantages of types of addressing modes.

→ Addressing modes are a key concept in computer architecture, defining how the operand of an instruction is chosen or specified. There are more addressing mode in the computer system, the important are:-

Immediate Addressing: This addressing mode is a method where the operand is specified directly within the instruction itself.

Example: MOV R1, #5 (Move the immediate value 5 into register R1)

- **Advantages:**
  - No memory reference needed.

Disadvantages:- Not suitable for accessing large data structures.

Register Addressing: The operand is located in a register.

Example: ADD R1, R2 (Add the contents of register R2 to register R1)

- **Advantages:** Fastest access due to registers.

Disadvantages: Requires careful register management to avoid conflicts.

Direct Addressing: The operand's address is given directly in the instruction.

Example: MOV R1, [1000] (Move the contents of memory location 1000 to register R1)

Advantages: Easy to understand and use.

Disadvantages:- May not be suitable for large memory spaces.

Indirect Addressing: In this mode, instruction holds address to memory location containing operand's address.

Example: MOV R1, [R2]

- **Advantages:** It is more Flexible.

Disadvantages:- Can be slower than direct addressing.

Stack Addressing Mode: Operands are implied to be on top of the stack.

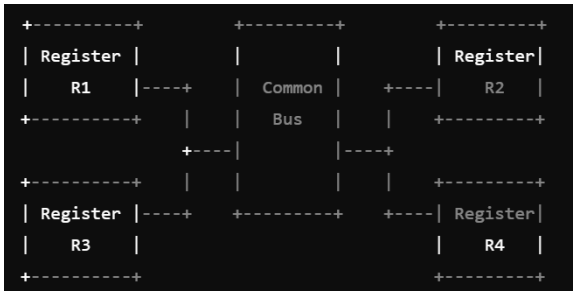
Example: PUSH R1 (Push the contents of register R1 onto the stack).

Advantages: Typically it is very fast.

Disadvantages: It has overflow risk.

21. Draw & explain the data movement among the registers using the common bus.

→ Data Movement among Registers using a Common Bus:-



Common Bus: A shared pathway used to transfer data among multiple registers.

Data Movement: Data can be transferred from one register to another via the common bus.

Control Signals: Multiplexers and control signals determine which register's data is placed onto the bus.

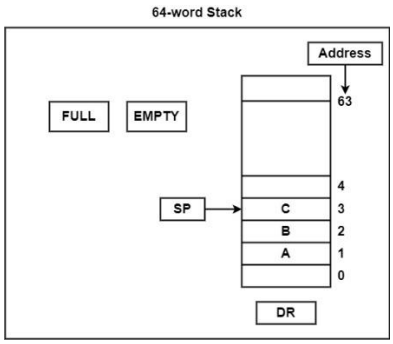
Transfer Example: To move data from R1 to R2.

Multiplexers: Used to select which register's data is placed on the bus.

Sequential Operation: Only one register can send data to the bus at a time.

22. Discuss the stack organisation of the CPU.

→ The stack organization in the CPU uses a Last-In, First-Out (LIFO) structure with a stack pointer to manage function calls, local variables, and temporary data efficiently.



Stack Organization of the CPU:-

Stack Pointer (SP): A special register that keeps track of the top of the stack.

Push Operation: Adds data to the top of the stack, decreasing the stack pointer or increasing it.

Pop Operation: Removes data from the top of the stack.

LIFO Structure: Operates on a Last-In, First-Out principle, which is efficient for recursive function calls.

Temporary Storage: Frequently used for temporary data storage, such as intermediate results of arithmetic operations.

23. Discuss the general register organization of the CPU.

→ The general register organization of a CPU refers to the way the CPU allocates and utilizes a set of high-speed memory locations called registers.

General Register Organization of the CPU:-

Multiple Registers: The CPU contains a set of general-purpose registers that can be used for various functions such as holding data.

Instruction Flexibility: Many instructions can operate directly on registers, allowing for more versatile.

Special-Purpose Registers: In addition to general-purpose registers, there are often special-purpose registers like the Program Counter (PC).

Context Switching: During context switching, the state of all registers must be saved and restored.

Efficient Data Access: Registers provide faster data access compared to memory.

24. (3*4)+(5*6) convert into RAM and show its stack operation.

→**Push 3**: Push the value 3 onto the stack. Stack: [3]

Push 4: Push the value 4 onto the stack. Stack: [3, 4]

Multiply: Pop the top two values (4 and 3), multiply them, and push the result (12) onto the stack. Stack: [12]

Push 5: Push the value 5 onto the stack. Stack: [12, 5]

Push 6: Push the value 6 onto the stack. Stack: [12, 5, 6]

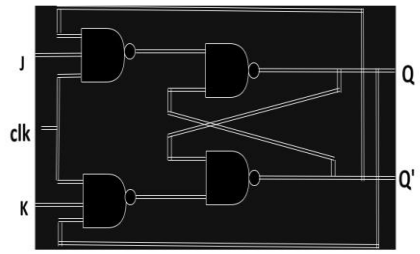
Multiply: Pop the top two values (6 and 5), multiply them, and push the result (30) onto the stack. Stack: [12, 30]

Add: Pop the top two values (30 and 12), add them, and push the result (42) onto the stack. Stack: [42]

28 / 25. Develop the JK flipflop with the help of NAND gate.

→ JK flip-flop is a sequential logic circuit that stores binary information and can toggle with two inputs.

→ The logic circuit using NAND gate, and also the truth table:-



Here inputs atr S,R and clk. Outputs are Q and Q'.

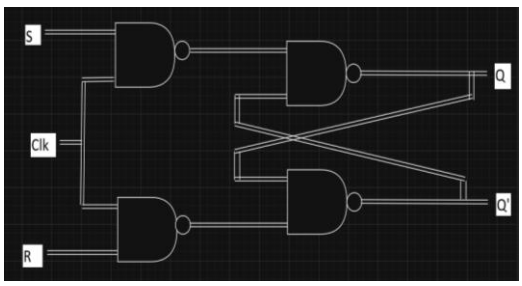
J	K	CLK	Q	Q'
0	0	1	Unchanged	State
0	1	1	0	1
1	0	1	1	0
1	1	1	Toggle	State

In this truth table, clock is always positive (when press the clock). When J,K are 0 & clk is 1, output will be unchanged state. If J,K are 0 & 1, and clk is 1, output Q ia 0 & Q' is 1. If J,K are 1 & 0, and clk is 1, output Q ia 1 & Q' is 0. When J,K are 1 & clk is 1, output will be toggle state.

26. Develop SR flipflop by using NAND GATE.

→ SR flip-flop is a flip-flop with two inputs, one is S and the other is R. S here stands for Set and R here stands for Reset.

The logic circuit using NAND gate, and also the truth table:-



Here inputs atr S,R and clk. Outputs are Q and Q'.

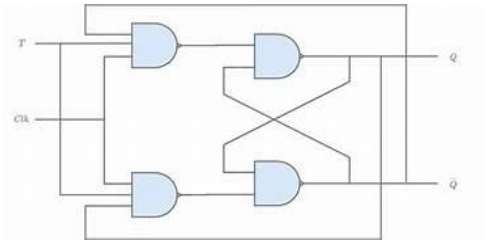
S	R	CLK	Q	Q'

0	0	1	Unchanged	State
0	1	1	0	1
1	0	1	1	0
1	1	1	Forbidden	State

In this truth table, clock is always positive (when press the clock). When S,R are 0 & clk is 1, output will be unchanged state. If S,R are 0 & 1, and clk is 1, output Q ia 0 & Q' is 1. If S,R are 1 & 0, and clk is 1, output Q ia 1 & Q' is 0. When S,R are 1 & clk is 1, output will be forbidden state.

27. Develop T flipflop.

→ A T flip flop is known as a toggle flip flop because of its toggling operation. It is a modified form of the JK flip flop.



T	Clk	Output
0	1	Unchanged
1	1	Toggle

In this truth table, clock is always positive (when press the clock). When T is 0 & clk is 1, output will be unchanged state. When T is 1 & clk is 1, output will be Toggle.

30. Caputizies the accumulator.

→ Accumulator: A register in the CPU used for arithmetic and logic operations.

Function: Temporarily holds data being processed and the results of operations.

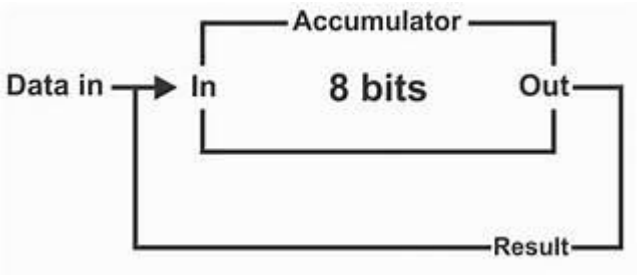
Usage: Central to performing calculations, comparisons, and data transfers within the CPU.

Efficiency: Reduces the need to access slower main memory for intermediate results.

Operations: Supports addition, subtraction, and other arithmetic functions directly.

Optimization: Enhances computational speed by minimizing the number of memory accesses.

Role: Crucial for executing instructions efficiently in a CPU's instruction set.



31. Explain the function of direct memory address.

→ DMA (Direct memory address) is the special feature within the computer system that transfers the data between memory and peripheral devices. The function:-

Efficient Data Transfer: DMA enables high-speed data transfer between peripherals and memory. DMA can transfer data in large blocks, making it more efficient.

Memory-to-Memory Transfer: DMA can be used for memory-to-memory transfers, allowing data to be moved directly within memory.

Improved System Performance: DMA can handle data transfers more quickly than CPU-driven transfers, improving the responsiveness.

How DMA Works:-

DMA Controller: A dedicated DMA controller chip manages DMA operations. It coordinates the data transfer and ensures that the CPU and the I/O device do not interfere with each other.

Bus Control: The DMA controller takes control of the system bus from the CPU to initiate the data transfer. This process is known as "cycle stealing".

32. Discuss the pin function of 8085 microprocessor.

→ The 8085 is an 8-bit microprocessor that simplifies computer design by requiring less supporting hardware.

Address Bus (A0 - A15): This 16-bit bus carries memory addresses or I/O port addresses during data transfer operations. The most significant 8 bits (A8-A15) specify the upper portion of the memory address. The least significant 8 bits (A0-A7) are used for lower memory address.

Data Bus (AD0 - AD7): This 8-bit bus is used for transferring data between the microprocessor and memory or I/O devices. It's a multiplexed bus.

Control and Status Signals: These signals manage communication and program flow:-

Read (RD): Indicates the microprocessor is reading data from memory or an I/O device.

Write (WR): Indicates the microprocessor is writing data to memory or an I/O device.

TRAP, Status flag and more.

Power Supply and Frequency: Some of are:-

Vcc (+5V): Provides power to the microprocessor.

Ground (GND): Connects to the system ground for proper electrical reference.

Reset Signals: RESET: An active-low signal that resets the microprocessor to its initial state.

And also some Serial I/O Ports, DMA Signals (optional).

33. Design the arcitecture of 8085.

→ The 8085 is an 8-bit microprocessor that simplifies computer design by requiring less supporting hardware.

Registers:-

Accumulator (A) Register: The accumulator is an 8-bit register. It plays a central role in arithmetic and logical operations.

General-Purpose Registers (B, C, D, E, H, L): These six 8-bit registers are used for temporary data storage.

Flag Register (F): The flag register contains status flags (bits) that reflect the outcome of arithmetic and logical operations.

Program Counter (PC): The program counter keeps track of the memory address of the next instruction to be executed.

Stack Pointer (SP): The stack pointer points to the top of the stack in memory.

Arithmetic Logic Unit (ALU): Performs arithmetic (addition, subtraction, etc.) and logical (AND, OR, NOT, etc.) operations on 8-bit binary data.

Address Bus (A0 - A15): This 16-bit bus carries memory addresses or I/O port addresses during data transfer operations.

Control and Status Signals: These signals manage communication and program flow:-

Read (RD): Indicates the microprocessor is reading data from memory or an I/O device.

Write (WR): Indicates the microprocessor is writing data to memory or an I/O device.

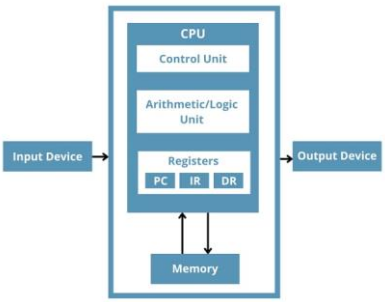
And many more.

34. Apply the do while loop & for loop in assembly language.

→

35. Explain the computer register with help of the structural diagram.

→ Computer registers are a fundamental component of a CPU (Central Processing Unit). They act as temporary storage locations within the CPU that hold data and instructions during program execution.



Registers: Represented as rectangular boxes in the diagram. They are typically much faster to access than main memory (RAM).

Data and Instructions: Registers can store both data (numbers, characters) and instructions (program code) during processing.

Arithmetic Logic Unit (ALU): Performs arithmetic (addition, subtraction) and logical (AND, OR, NOT) operations on data fetched from registers.

Main Memory (RAM): Slower but larger storage for data and programs.

Control Unit (CU): Decodes instructions from memory or registers, generates control signals, and manages data flow between registers, ALU, and memory.

Types of registers:-

Accumulator (A) Register: The accumulator is an 8-bit register. It plays a central role in arithmetic and logical operations.

General-Purpose Registers (B, C, D, E, H, L): These six 8-bit registers are used for temporary data storage.

Flag Register (F): The flag register contains status flags (bits) that reflect the outcome of arithmetic and logical operations.

Program Counter (PC): The program counter keeps track of the memory address of the next instruction to be executed.

Stack Pointer (SP): The stack pointer points to the top of the stack in memory.

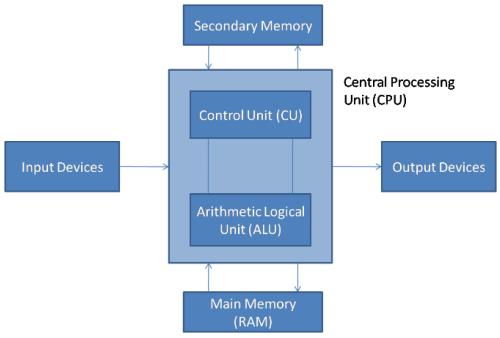
36. Classify different types of computer register.

→ Answer of 35.

37. Answer 12.

38. Design & discuss the components of computer system.

→ A computer system can be broadly divided into two main categories: hardware and software. Hardware refers to the physical components of the computer, while software is the set of instructions that tells the hardware what to do.



→ Hardware Components:

Central Processing Unit (CPU): Also known as the brain of the computer, the CPU is responsible for executing instructions, performing calculations, and managing the overall flow of information within the system.

Main Memory (RAM): This volatile memory stores data and programs that the CPU needs for immediate use. It's fast to access but temporary.

Secondary Storage Devices: These devices provide non-volatile storage for data and programs that need to be preserved even after the computer is turned off. Eg. DVD, HDD.

Input Devices: These components allow users to interact with the computer and provide data or instructions. Common examples include keyboards.

Output Devices: These components display information or results generated by the computer. Eg. Monitors.

→ Software Components:

Operating System (OS): The core software that manages hardware resources, provides a user interface for interaction, and allows other software applications to run.

Application Software: Programs designed for specific tasks like word processing, web browsing.

System Software: Low-level software that interacts directly with the hardware and provides core functionalities like device drivers & utility programs.

39. Construct the Von- Neumann model with all its components.

→ The von Neumann architecture consists of the processing, control, memory, input, and output units.

→Components:

Input Devices: Keyboards, mice, scanners, webcams, microphones, etc. They provide data and instructions to the computer.

Output Devices: Monitors, printers, speakers, projectors, etc. They display information or results generated by the computer.

Main Memory (RAM): Stores data and programs that the CPU needs for immediate use. It's fast to access but temporary.

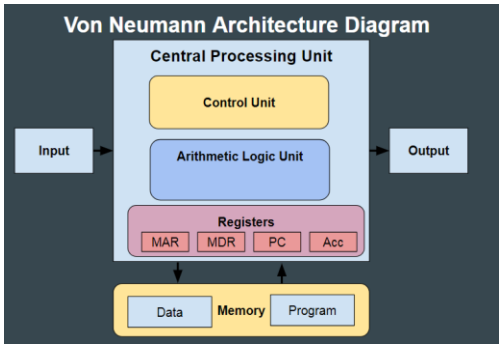
Secondary Storage: Hard disks, SSDs, etc., provide non-volatile storage for programs and data that need to be preserved.

Central Processing Unit (CPU): The brain of the computer, responsible for executing instructions, performing calculations, and managing data flow. It consists of two main subunits:

- **Control Unit (CU):** Decodes instructions, manages data flow, and controls communication with other components.
- **Arithmetic Logic Unit (ALU):** Performs arithmetic (addition, subtraction) and logical (AND, OR, NOT) operations on data.

➔ **Data Flow:**

- Data enters the system from input devices.
- The CPU fetches instructions and data from RAM as needed.
- The CU decodes instructions and controls the ALU to perform operations.



40. Distinguish the importance of accumulator unit of the computer system.

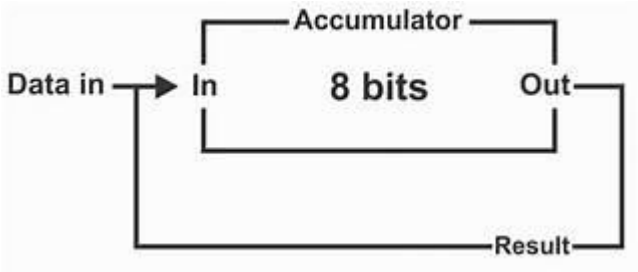
→ Accumulator: A register in the CPU used for arithmetic and logic operations. Here are three key reasons highlighting the importance of the accumulator unit:-

Central Role in Arithmetic and Logic Operations:-

- The accumulator serves as the primary register for performing arithmetic (addition, subtraction, multiplication, division) and logical (AND, OR, NOT) operations within the CPU.
- It interacts directly with the Arithmetic Logic Unit (ALU), where these operations take place.

Temporary Storage: The accumulator acts as a high-speed scratchpad memory within the CPU. It temporarily stores operands (data elements involved in an operation) and results generated by the Arithmetic Logic Unit (ALU).

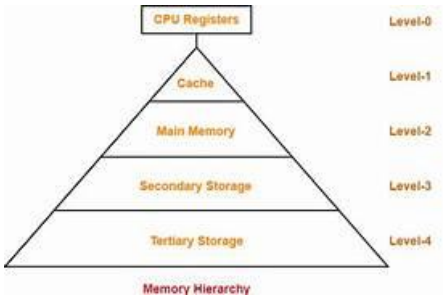
Reduced Memory Access: By holding operands and results, the accumulator reduces the need for the CPU to constantly access slower main memory (RAM) for data transfer.



41. Explain the memory hierarchy of the computer system.

→ The computer system relies on a memory hierarchy, which is a tiered structure with different types of memory organized based on speed and capacity. Here's a breakdown of the key points.

Levels: The hierarchy consists of several levels, typically five:-



Registers: These are the fastest and smallest memory units located within the CPU. They hold very specific data.

Cache Memory: This high-speed memory acts as a buffer between the CPU and main memory, storing frequently accessed data.

Main Memory (RAM): This is the primary memory (usually volatile) that holds the programs and store the data temporarily. It's slower than registers and cache but faster than secondary storage.

Secondary Storage: They offer much larger capacities than main memory but have slower access times. They store data persistently even when the computer is powered off.

Tertiary Storage: This is the slowest and has the highest capacity but is rarely used for everyday tasks due to its extremely slow access times. Magnetic tapes are mainly used for backup.

42. Outline the features of primary & secondary memory.

→ Primary memory is fast, temporary computer memory; while secondary memory is slower, permanent storage.

→Features of Primary Memory:-

High-Speed Access: Primary memory, such as RAM (Random Access Memory), provides very fast data access, crucial for system performance.

Volatile: Primary memory is volatile, meaning it loses its data when the power is turned off. It is used for temporary storage.

Capacity: Primary memory generally has a smaller capacity compared to secondary memory.

Cost: More expensive per unit of storage compared to secondary memory.

→Features of Secondary Memory:-

Non-volatile: Secondary memory stores data even when the power is turned off, making it suitable for long-term storage.

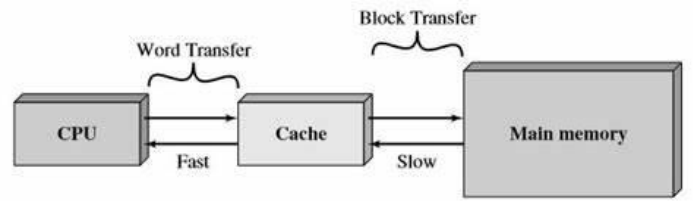
Large Storage Space: Secondary memory offers much larger storage capacities, from hundreds of gigabytes (GB) to several terabytes (TB).

Slower Access Times: Access times are slower than primary memory, typically in milliseconds for hard disk drives.

Cost: Less expensive per unit of storage compared to primary memory.

43. Relate the term cache memory & cache mapping.

→Cache memory is a small, high-speed memory storage component located between the CPU and the main memory (RAM).



Features of Cache Memory:-

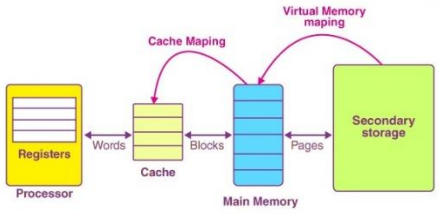
High-Speed Access: Cache memory operates at a much higher speed than main memory (RAM), providing rapid data access to the CPU.

Multi-Level Cache: Cache memory is typically organized into multiple levels:- **L1 Cache (Level 1)**, **L2 Cache (Level 2)**, **L3 Cache (Level 3)**.

Volatile Memory: Like RAM, cache memory is volatile, meaning it loses its contents when the power is turned off.

Cost: Cache memory is more expensive per byte than main memory due to its high speed and advanced technology.

→Cache mapping is a technique used in computer systems to manage how data from main memory is stored in cache memory.



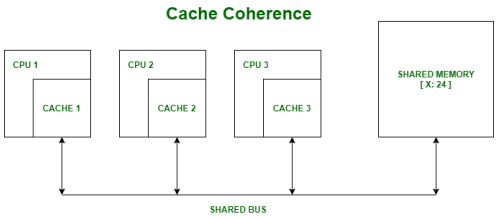
Direct-Mapped Cache: A direct-mapped cache is a simple scheme where each memory block maps to a single specific cache line. **It o**ffers fast access because of the simplicity of the mapping. It has lower cost due to simplier design.

Fully Associative Cache: Any block of main memory can be loaded into any line of the cache. Most complex mapping technique. It is slow & i t has higher cost due to complex search logic.

Set-Associative Cache: A set-associative cache divides the cache into groups (sets) where each set can hold multiple copies of data blocks. It is moderately complex & has moderate cost.

44. Interpret the term cache coherence & its importance.

→ Cache coherence refers to the consistency and uniformity of data stored in multiple cache memories that are part of a shared memory multiprocessor system. In such systems, each processor has its own cache, and these caches can store copies of the same memory locations. Cache coherence ensures that any changes made to a data value in one cache are immediately reflected in all other caches.



Importance of Cache Coherence:-

Data Integrity: Ensures that all processors have the same up-to-date version of the data, preventing errors and corruption.

Performance Optimization: By maintaining cache coherence, processors can utilize their caches effectively, improving overall system performance.

Programmer Simplicity: Coherence provides a simpler programming model for developers. They can assume a consistent memory view across all processors, making it easier.

45. Construct the architecture of RAM.

→The architecture of RAM (Random Access Memory) is designed to provide fast and efficient access to data.

Memory Cells: These are the fundamental units that store individual bits of data (0 or 1). There are two main types:-

- **Static RAM (SRAM):** Faster and more expensive.
- **Dynamic RAM (DRAM):** More common and cheaper.

Control Logic: Manages the operations of reading, writing, and refreshing (in DRAM).

Power Distribution: Ensures that all components receive stable power.

Address Decoding:-

- **Row Decoder**: Selects the row in the memory array corresponding to the address.
- **Column Decoder**: Selects the column in the memory array corresponding to the address.

Data Input/Output (I/O) Buffers:-

- **Data Input Buffer**: Temporarily holds the data to be written to memory.
- **Data Output Buffer**: Temporarily holds the data read from memory before sending it to the CPU

Key Points of RAM Architecture:-

- **Speed**: Designed for high-speed access to support the CPU's operations.
- **Volatility**: RAM is volatile memory, meaning it loses data when power is turned off.

46. Explain the structure of instruction format.

→ The structure of an instruction format in a computer system is crucial for defining how instructions are encoded and executed by the CPU.

Instruction Format

→ Key Components of an Instruction Format:-

Operation **Code (Opcode)**: Specifies the operation that the instruction performs, such as addition, subtraction, load, store, etc. Typically a fixed number of bits (e.g., 4 to 8 bits).

Operands: These are the inputs for the operation. They can be registers, memory addresses, or immediate values. **Types of Operands**:-

- **Register Operand**: Specifies a register in the CPU.
- **Immediate Operand**: A constant value embedded within the instruction.

Addressing Mode: An addressing mode specifies how the CPU determines the memory location or register that holds the data for an instruction.

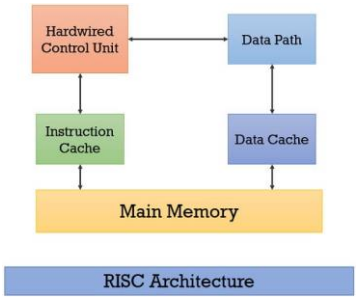
Instruction Length:-

Fixed-Length Instructions: All instructions have the same length.

Variable-Length Instructions: Instructions have different lengths.

48. Determine the importance of RISC.

→ RISC (Reduced Instruction Set Computer) is a processor design focused on simpler instructions for faster execution and potentially lower power consumption. Here are outlining the importance of RISC:-



Performance: RISC processors focus on simpler instructions that can be executed faster. This streamlined approach can lead to quicker processing time.

Efficiency: RISC processors often require less complex hardware due to the simpler instruction set. This translates to potentially lower power consumption, making them ideal for battery-powered devices.

Instruction Set Flexibility: RISC architectures can be extended with additional instructions as needed, providing a balance between simplicity and functionality.

Compiler Optimization: Simpler instructions are easier for compilers to optimize, potentially leading to more efficient machine code.

Overall, RISC offers a compelling approach to processor design, prioritizing efficiency, & performance.

49. Examine the data transfer and data manipulation perform by the central processing unit.

→ The Central Processing Unit (CPU) is the brain of the computer and plays a crucial role in both data transfer and manipulation.

→Data Transfer Operations:-

i. Load and Store Instructions:-

•**Load (LD)**: Moves data from memory to a register.

- **Example**: `LOAD R1, 1000`

•**Store (ST)**: Moves data from a register to memory.

- **Example**: `STORE R1, 1000`

ii. Register Transfer

- **Move (MOV)**: Transfers data between registers.
  - **Example**: `MOV R1, R2`

iii. **Input/Output Operations**

- **Input (IN)**: Reads data from an I/O device into a register.
  - **Example**: `IN R1, PORT1`
- **Output (OUT)**: Writes data from a register to an I/O device.
  - **Example**: `OUT PORT1, R1`

→ Data Manipulation Operations:-

i. Arithmetic Operations

Addition (ADD): Adds two operands.

a. **Example**: `ADD R1`

Subtraction (SUB): Subtracts one operand from another.

b. **Example**: `SUB R1`

ii. **Logical Operations**

- **AND**: Performs a logical AND operation.
  - **Example**: `AND R1`
- **OR**: Performs a logical OR operation.
  - **Example**: `OR R1`

iii. **Shift and Rotate Operations**

- **Shift Left (SHL)**: Shifts bits to the left.

Shift Right (SHR): Shifts bits to the right.

50. Assembly language prog to multiply & divide two 8 bits numbers.