DECO Important Question by Bhavy Sharma

Q1. Write short notes on following:
(i) Virtual memory organization.
(ii) Random access memories.
Ans. (i) Virtual Memory Organization:

Virtual memory is a memory management technique used by operating systems to provide the illusion of a larger and contiguous memory space than is physically available in the system.

1. **Concept:** Virtual memory allows programs to use more memory than is physically available by utilizing disk space as an extension of RAM.

2. **Pages and Frames:** Memory is divided into fixed-size blocks called pages. Physical memory is divided into corresponding blocks called frames.

3. **Page Table:** Each process has its own page table, which maps virtual addresses to physical addresses. This table helps the CPU translate virtual addresses into physical addresses.

4. **Page Replacement:** When a program needs to access data that is not currently in physical memory, the operating system swaps out a page from RAM to disk, making room for the needed page.

5. **Benefits:** Allows for efficient memory management, enables multitasking, and prevents programs from crashing due to insufficient memory.

(ii) Random Access Memories (RAM):

RAM is a type of computer memory that allows data to be read and written in almost the same amount of time, regardless of the physical location of data inside the memory.

1. **Types:** There are mainly two types of RAM: Dynamic RAM (DRAM) and Static RAM (SRAM).

2. **Volatile Memory:** RAM is volatile, meaning it loses its data when power is turned off. This is in contrast to non-volatile memory like ROM.

3. **Speed:** RAM is much faster than secondary storage devices like hard drives and SSDs, making it ideal for storing data that the CPU needs to access quickly.

4. **Access Time:** RAM provides random access, meaning any storage location can be accessed directly and in a constant time, regardless of the address that was accessed previously.

5. **Capacity:** RAM capacity determines how much data a computer can handle at once. More RAM allows for smoother multitasking and faster program execution.

Q2. What do you mean by cache memory? How is the performance of a memory system improved by using cache?
Ans. Cache memory is a small, high-speed memory unit located between the CPU and main memory (RAM) in a computer system. Its purpose is to temporarily store copies of frequently accessed data and instructions from main memory, allowing the CPU to access them more quickly than if it had to retrieve them from RAM every time they are needed.

1. **Faster Access Time:** Cache memory is much faster than main memory (RAM) in terms of access time. Since it is closer to the CPU, accessing data from the cache takes significantly less time compared to accessing it from RAM.

2. **Reduced Access Latency:** By storing frequently accessed data and instructions in the cache, the CPU can retrieve them more quickly, reducing the overall access latency. This leads to faster program execution and improved system responsiveness.

3. **Higher Hit Rate:** When the CPU requests data, the cache checks if the requested data is already stored in its memory. If it is, it's called a cache hit, and the data can be quickly retrieved. Cache hits result in faster data access and execution.

4. **Lower Memory Traffic:** With cache memory, the CPU can reduce the number of requests it makes to main memory. This reduces the overall memory traffic on the system bus, freeing up bandwidth for other tasks and improving overall system performance.

5. **Improved CPU Utilization:** Faster access to data means the CPU spends less time waiting for data to be fetched from main memory. As a result, the CPU can spend more time executing instructions, leading to better utilization of its processing power.

Q3. Write De Morgan's Laws?

Ans. De Morgan's Laws are a pair of fundamental rules in Boolean algebra that describe the relationship between logical operations (AND, OR, NOT) when they are negated.

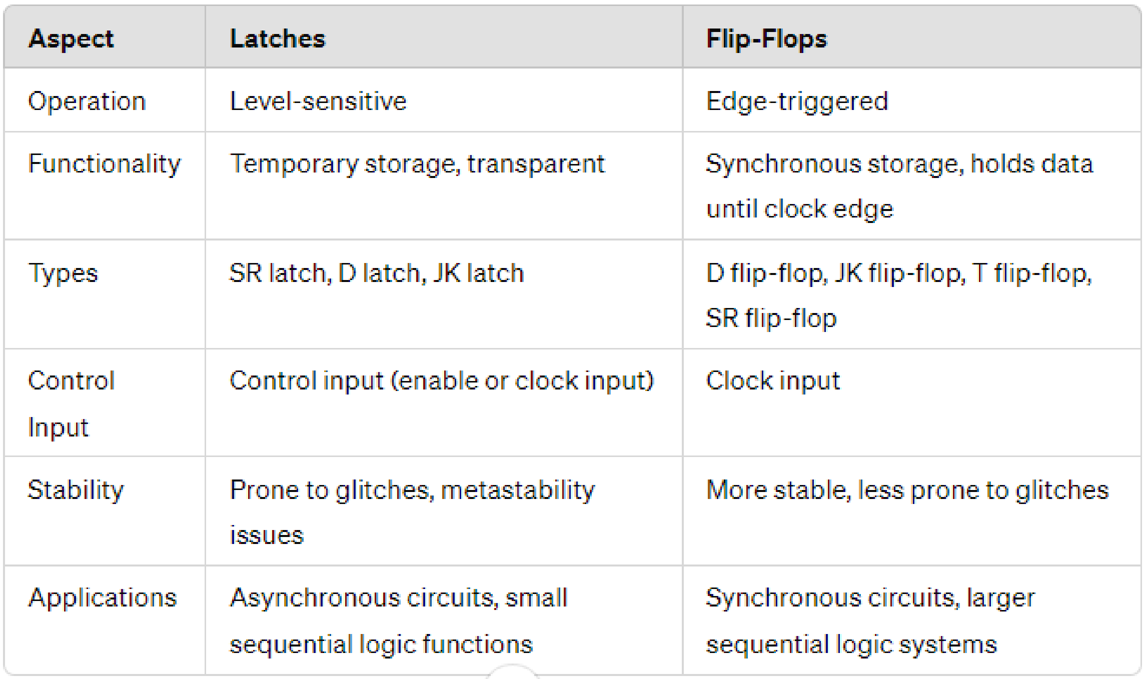
There are two laws:

De Morgan's First Law :- $(AUB)' = A'\cap B'$

De Morgan's Second Law :- $A'\cap B' = (AUB)'$

Q4. Differentiate between flip flop and latches.?

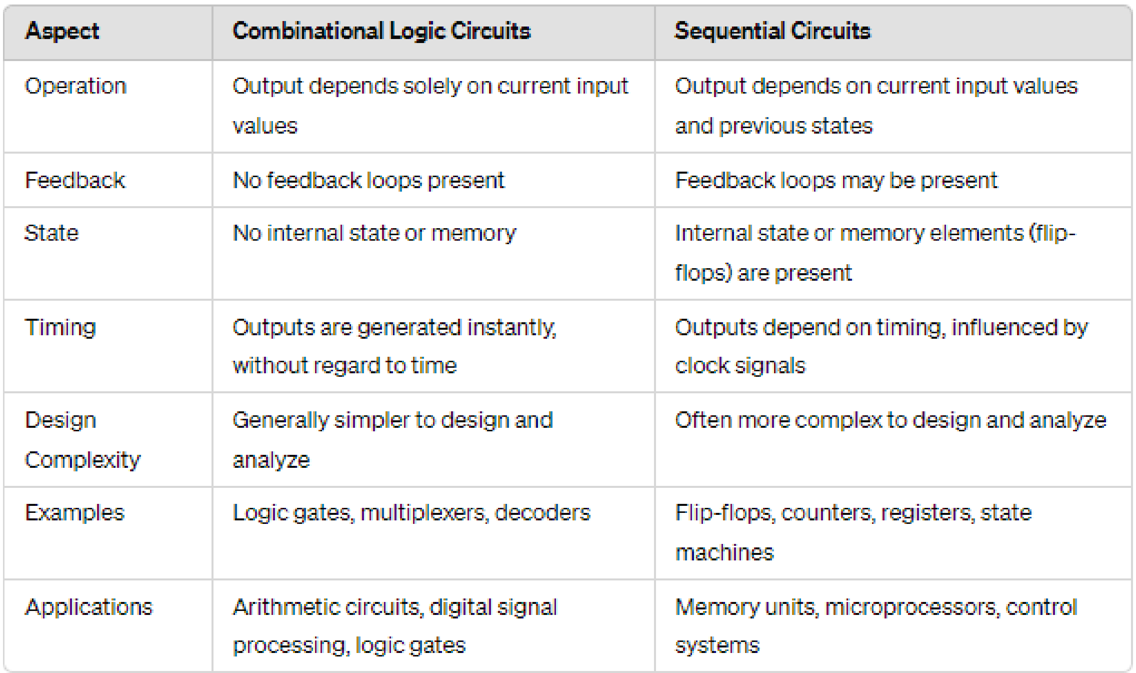
Ans.



Aspect	Latches	Flip-Flops
Operation	Level-sensitive	Edge-triggered
Functionality	Temporary storage, transparent	Synchronous storage, holds data until clock edge
Types	SR latch, D latch, JK latch	D flip-flop, JK flip-flop, T flip-flop, SR flip-flop
Control Input	Control input (enable or clock input)	Clock input
Stability	Prone to glitches, metastability issues	More stable, less prone to glitches
Applications	Asynchronous circuits, small sequential logic functions	Synchronous circuits, larger sequential logic systems

Q5. Differentiate between combinational logic circuit and sequential circuits?

Ans.



Aspect	Combinational Logic Circuits	Sequential Circuits
Operation	Output depends solely on current input values	Output depends on current input values and previous states
Feedback	No feedback loops present	Feedback loops may be present
State	No internal state or memory	Internal state or memory elements (flip-flops) are present
Timing	Outputs are generated instantly, without regard to time	Outputs depend on timing, influenced by clock signals
Design Complexity	Generally simpler to design and analyze	Often more complex to design and analyze
Examples	Logic gates, multiplexers, decoders	Flip-flops, counters, registers, state machines
Applications	Arithmetic circuits, digital signal processing, logic gates	Memory units, microprocessors, control systems

Q6. Which gates are called universal gates and why?

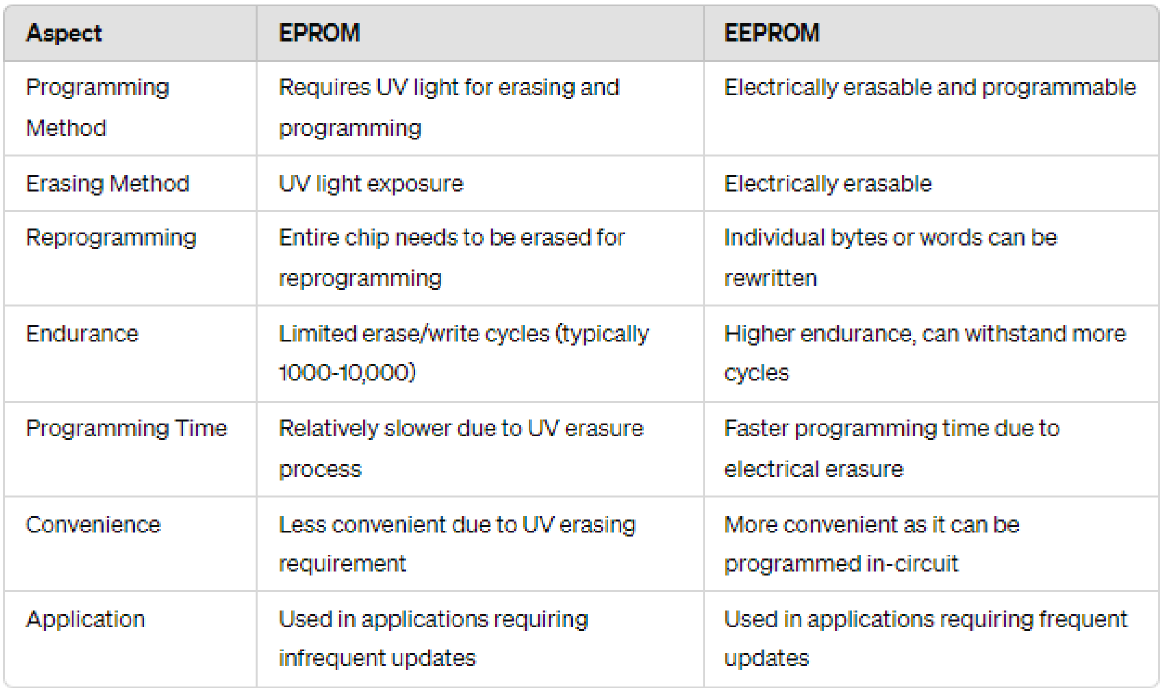
Ans. Universal gates are those logic gates from which any other type of logic gate can be derived. There are two types of universal gates: NOR gate and NAND gate. Here's why they are called universal gates:

1. **NOR Gate:**
    - **A NOR gate is called a universal gate because it can perform all the basic logic operations, including AND, OR, and NOT.**
    - **The output of a NOR gate is only low (0) when both of its inputs are high (1), making it the negation of an OR gate. Thus, it can be used to implement an OR gate by negating its inputs.**
    - **By combining multiple NOR gates, one can implement any other logic function. For example, by inverting the output of a NOR gate, it becomes an AND gate.**
    - **Therefore, with NOR gates alone, it is possible to construct any logic circuit, making it a universal gate.**
2. **NAND Gate:**
    - **Similarly, a NAND gate is called a universal gate because it can perform all the basic logic operations, including AND, OR, and NOT.**
    - **The output of a NAND gate is only high (1) when both of its inputs are low (0), making it the negation of an AND gate. Thus, it can be used to implement an AND gate by negating its inputs.**
    - **By combining multiple NAND gates, one can implement any other logic function. For example, by inverting the output of a NAND gate, it becomes an OR gate.**
    - **Therefore, with NAND gates alone, it is possible to construct any logic circuit, making it a universal gate.**

Q7. What do you mean by shift register? What is the need of the shift register?
Ans. A shift register is a type of sequential logic circuit that stores and shifts data serially. It consists of a chain of flip-flops connected in series, with the output of one flip-flop connected to the input of the next. Shift registers can shift data either to the left or to the right, depending on the application.

The primary function of a shift register is to temporarily store and manipulate data in a digital system. It is commonly used for various purposes such as data storage, data transfer, parallel-to-serial and serial-to-parallel conversion, and implementing digital counters and shifters.

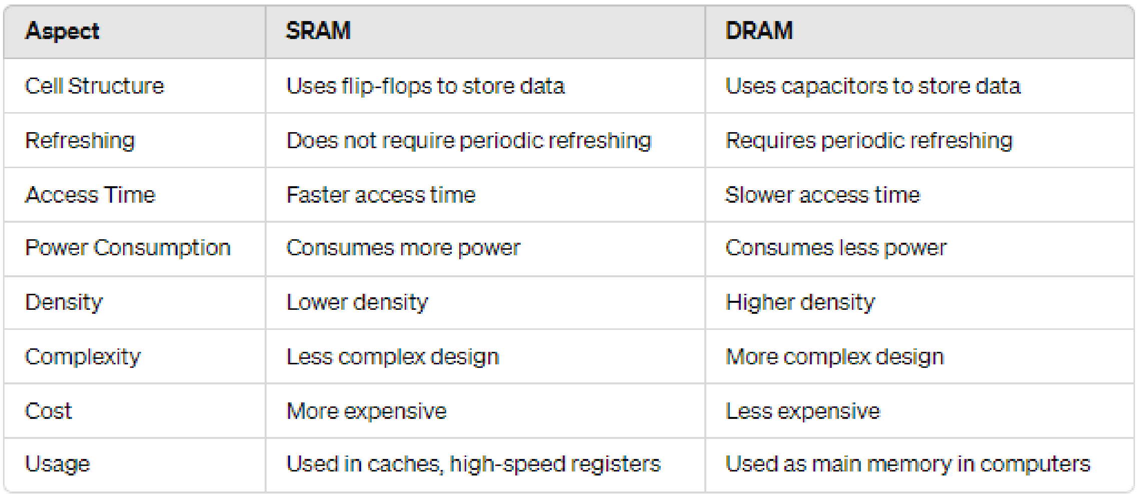
Q8. Difference between EPROM & EEPROM?
Ans.



Aspect	EPROM	EEPROM
Programming Method	Requires UV light for erasing and programming	Electrically erasable and programmable
Erasing Method	UV light exposure	Electrically erasable
Reprogramming	Entire chip needs to be erased for reprogramming	Individual bytes or words can be rewritten
Endurance	Limited erase/write cycles (typically 1000-10,000)	Higher endurance, can withstand more cycles
Programming Time	Relatively slower due to UV erasure process	Faster programming time due to electrical erasure
Convenience	Less convenient due to UV erasing requirement	More convenient as it can be programmed in-circuit
Application	Used in applications requiring infrequent updates	Used in applications requiring frequent updates

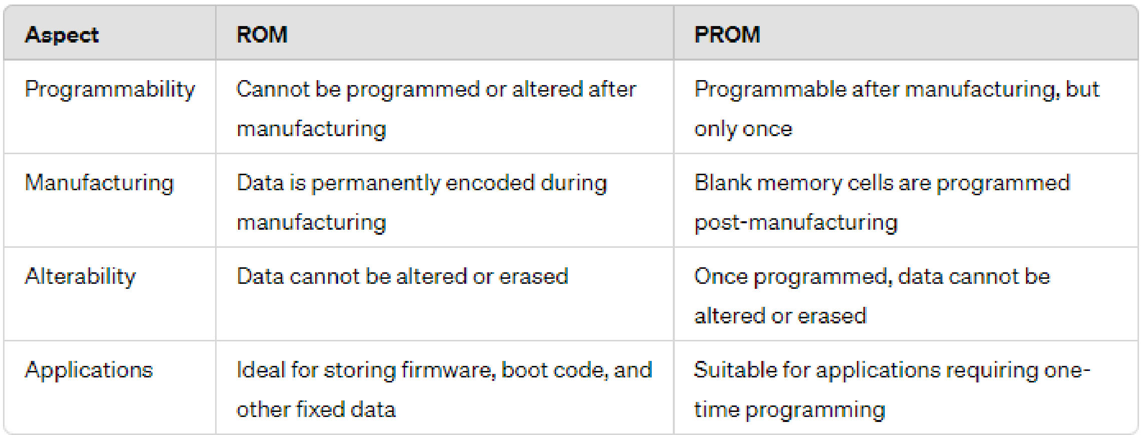
Q9. Difference between SRAM & DRAM?

Ans.



Aspect	SRAM	DRAM
Cell Structure	Uses flip-flops to store data	Uses capacitors to store data
Refreshing	Does not require periodic refreshing	Requires periodic refreshing
Access Time	Faster access time	Slower access time
Power Consumption	Consumes more power	Consumes less power
Density	Lower density	Higher density
Complexity	Less complex design	More complex design
Cost	More expensive	Less expensive
Usage	Used in caches, high-speed registers	Used as main memory in computers

Q10. Differentiate between ROM & PROM?
Ans.

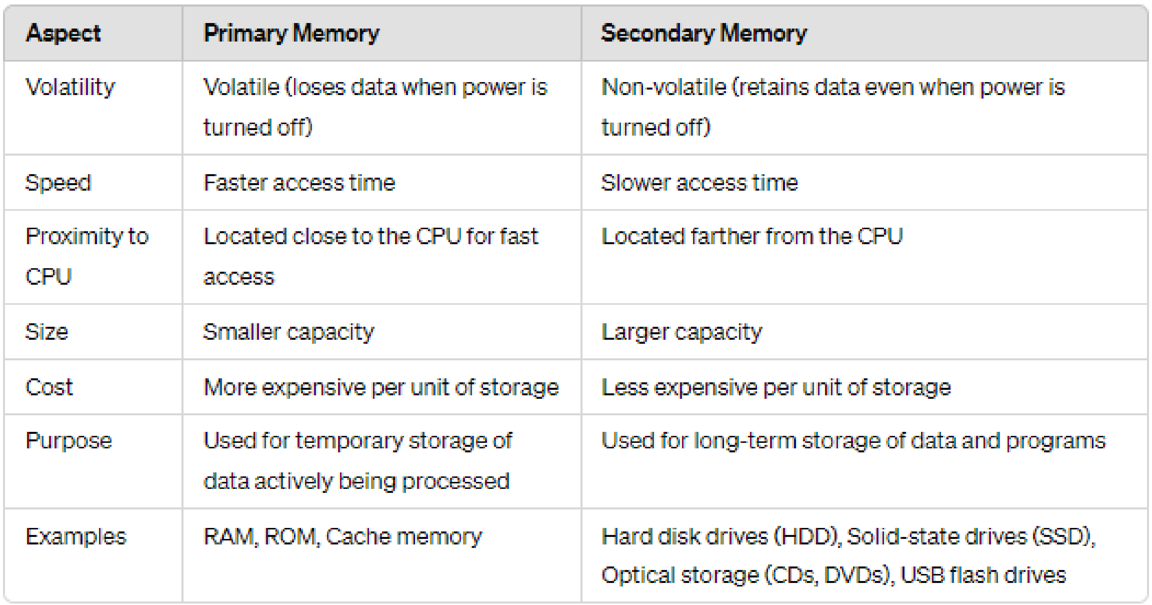


Aspect	ROM	PROM
Programmability	Cannot be programmed or altered after manufacturing	Programmable after manufacturing, but only once
Manufacturing	Data is permanently encoded during manufacturing	Blank memory cells are programmed post-manufacturing
Alterability	Data cannot be altered or erased	Once programmed, data cannot be altered or erased
Applications	Ideal for storing firmware, boot code, and other fixed data	Suitable for applications requiring one-time programming

Q11. Describe USB.
Ans. USB (Universal Serial Bus) is a widely used standard for connecting peripherals and external devices to computers and other digital devices. It provides a simple, standardized interface for data transfer, power supply, and communication between devices.

Q12. Differentiate between primary memory and secondary memory. Also list the examples of primary memory and secondary memory.

Ans.



Aspect	Primary Memory	Secondary Memory
Volatility	Volatile (loses data when power is turned off)	Non-volatile (retains data even when power is turned off)
Speed	Faster access time	Slower access time
Proximity to CPU	Located close to the CPU for fast access	Located farther from the CPU
Size	Smaller capacity	Larger capacity
Cost	More expensive per unit of storage	Less expensive per unit of storage
Purpose	Used for temporary storage of data actively being processed	Used for long-term storage of data and programs
Examples	RAM, ROM, Cache memory	Hard disk drives (HDD), Solid-state drives (SSD), Optical storage (CDs, DVDs), USB flash drives

Q13. What are multiplexer and Demultiplexer?
Ans. <u>Multiplexer (MUX):</u>
Think of a multiplexer like a data traffic controller. It's a device that takes multiple input signals and combines them into one output signal. Just like how a traffic controller merges several lanes of traffic into one lane, a multiplexer merges several input signals into one output signal.

- A multiplexer has multiple input ports labeled 0, 1, 2, ..., n.
- It also has a control input that determines which input signal gets routed to the output.
- Depending on the control input, the multiplexer selects one of the input signals and forwards it to the output.
- So, if you have, say, 8 input signals and you want to send only one of them to the output at a time, you can use an 8-to-1 multiplexer.

<u>Demultiplexer (DEMUX):</u>
Now, think of a demultiplexer as the opposite of a multiplexer. It's like a signal splitter. A demultiplexer takes one input signal and routes it to one of several output ports based on a control signal.

- A demultiplexer has one input port and multiple output ports labeled 0, 1, 2, ..., n.
- It also has a control input that determines which output port receives the input signal.
- Depending on the control input, the demultiplexer selects one of the output ports and forwards the input signal to that port.

- So, if you have one input signal and you want to send it to one of, say, 8 output ports based on a control signal, you can use a 1-to-8 demultiplexer.