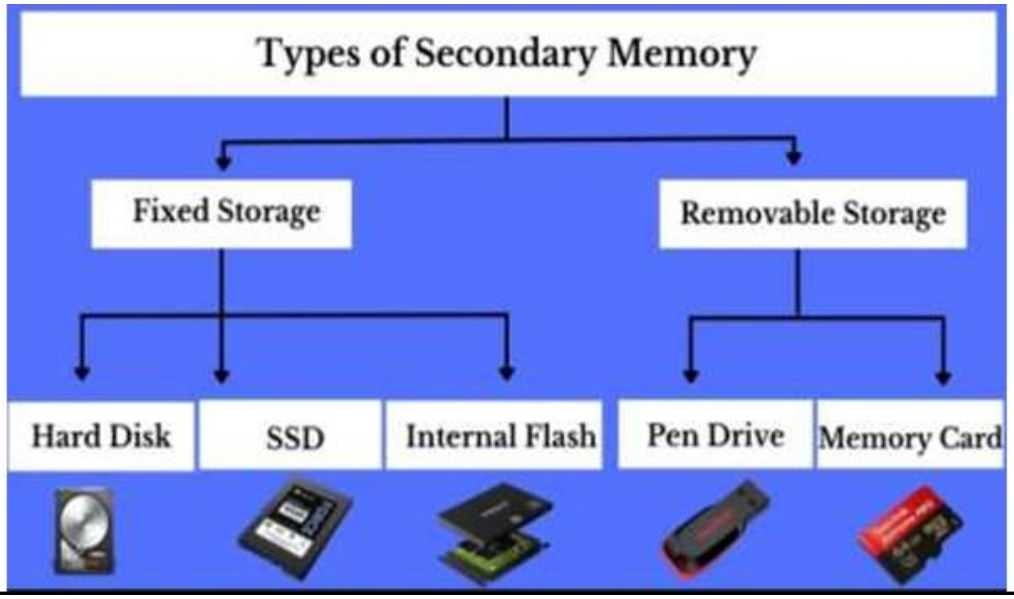
CO-4 SHORT ANSWERS

1. Identify and list some of the secondary storage devices.

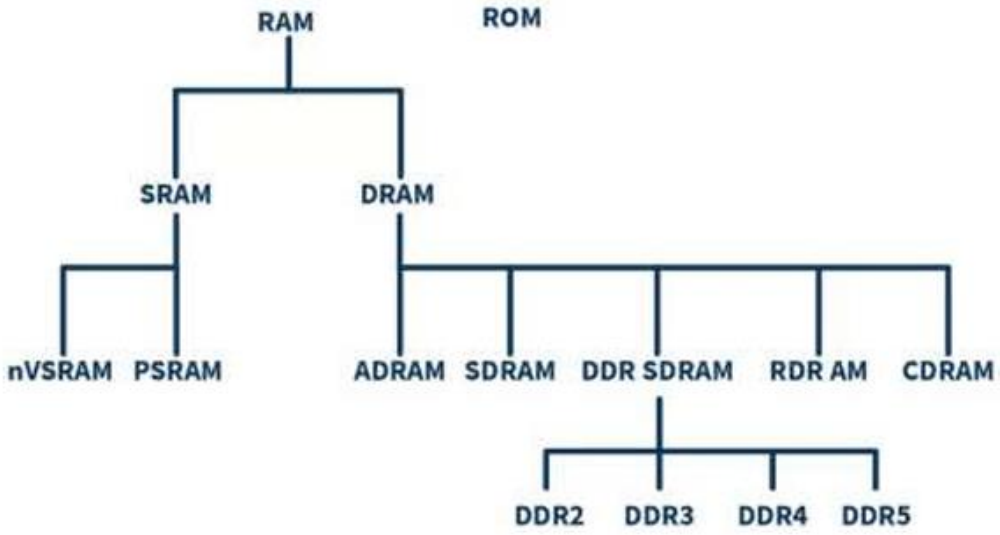
Ans:



---

2. Identify and categorize different types of RAM according to their characteristics.

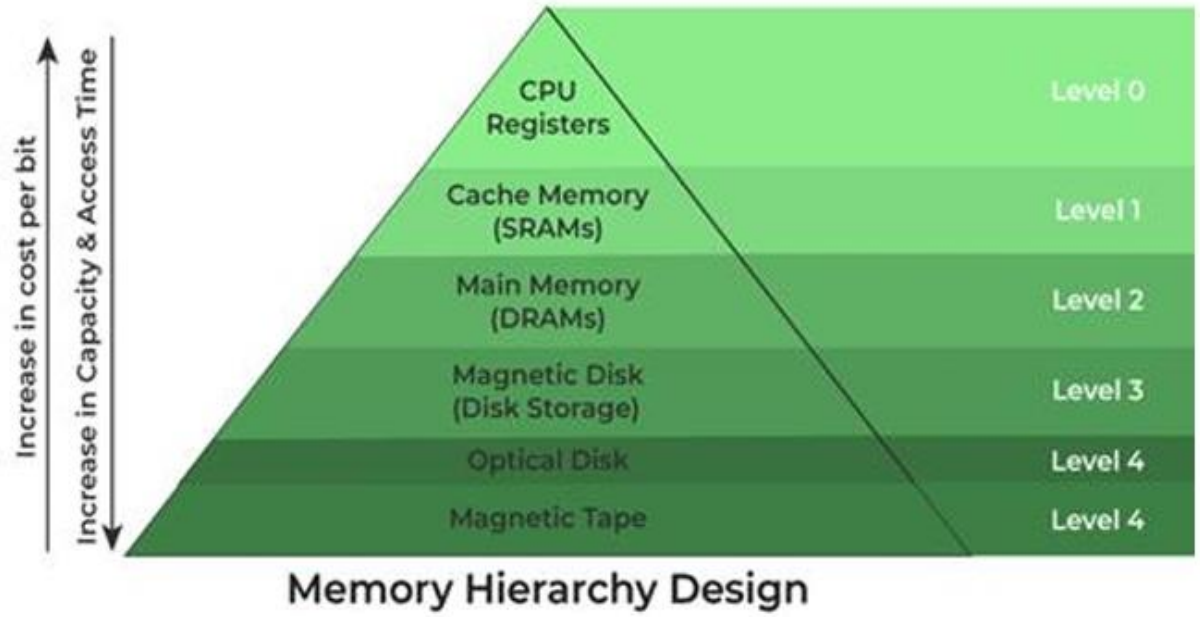
Ans:



- SRAM (Static RAM)
- DRAM (Dynamic RAM)
- DDR SDRAM (Double Data Rate Synchronous DRAM)
- LPDDR (Low Power DDR SDRAM)
- GDDR (Graphics Double Data Rate SDRAM)

3. Draw the memory hierarchy diagram with increasing distance from CPU.

Ans:



Memory Hierarchy Design

4. Specify the role of memory cell in the context of memory organization.

Ans:

• Basic Storage Unit: It's the fundamental unit that stores a single bit of data (0 or 1).

• Building Block: Groups of memory cells are combined to form words (e.g., 4 bits, 8 bits) which are the basic units of data transfer and addressing within the memory.

5. Formulate cache performance and its purpose.

Ans:

1. Cache Performance is,

Hit Ratio = Number of cache/ hits Number of searches

2. Cache memory speeds up computer programs by storing frequently accessed data in a faster location closer to the CPU.

6. Highlight various policies of cache data replacement.

Ans:

1. Least Recently Used (LRU)
2. First-In, First-Out (FIFO)
3. Random Replacement
4. Least Frequently Used (LFU)
5. Most Recently Used (MRU)
6. Optimal Replacement

7. Represent the use of virtual memory in computer system.

Ans:

Virtual memory acts like a larger, contiguous memory space for programs even though physical RAM is limited. This allows:

• **Running larger programs**: By utilizing disk space as an extension of RAM, programs exceeding physical RAM capacity can still run.

• **Multitasking**: Multiple programs can be loaded partially into RAM, swapped with data on disk as needed, enabling efficient multitasking.

- **Memory Extension**
- **Memory Sharing**
- **Memory Protection**

8. List various external storage devices.

Ans:

Examples include external hard drives, USB flash drives, memory cards, optical discs (CDs/DVDs/Blu-rays), and network-attached storage (NAS).

9. Summarize various Asynchronous Data Transfer methods.

Ans:

1. **Strobe Control**: Syncs data transfer with one signal, indicating when to exchange data.
2. **Handshaking:** Sender says "ready," receiver confirms, ensuring reliable communication.

10. List potential reasons for buffering in I/O operations.

Ans:

(i) Mismatched speeds between devices

(ii) Temporarily storing data during slow transfer rates

(iii) Managing data flow between fast and slow devices

(iv) Network and File System Factors

11. List the various data transfer methods in IO communication.

Ans:

(i) Programmed I/O

(ii) Interrupt-driven I/O

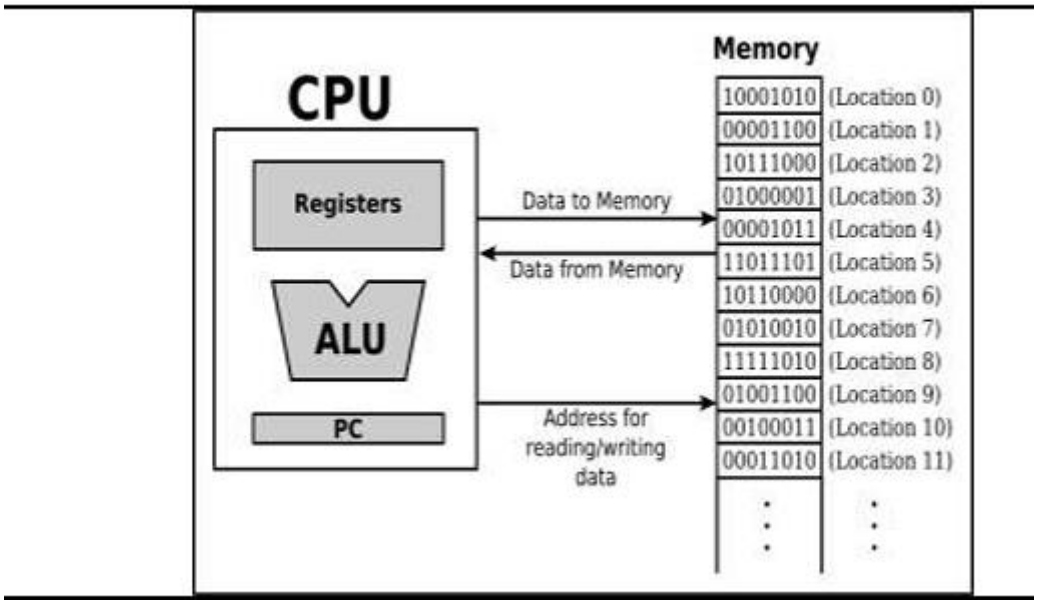
(iii) Direct Memory Access (DMA)

(iv) Memory Mapped I/O

12. List various types of buffering techniques.

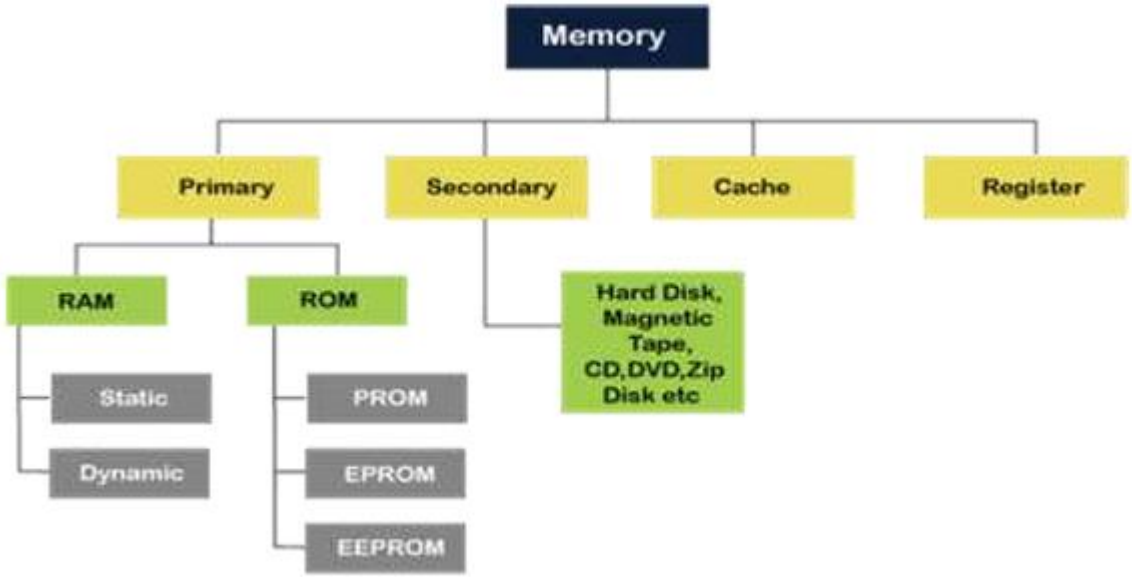
- Single Buffering
- Double Buffering
- Circular buffering

13. Sketch the memory addressing diagram.



CO-4 LONG ANWERS

1. Identify the types and functionality of primary memory devices exploring their role in data storage.



Primary memory, directly accessible by the CPU, offers fast speeds but limited capacity. Here's a breakdown:

1. **RAM (Random Access Memory):** Volatile (loses data on power off). Stores:

- Running programs
- Operating system
- Temporary data

Benefits: Fast access, random access

2. **ROM (Read-Only Memory):** Non-volatile (data persists). It Stores essential startup programs.

ROM Classification:

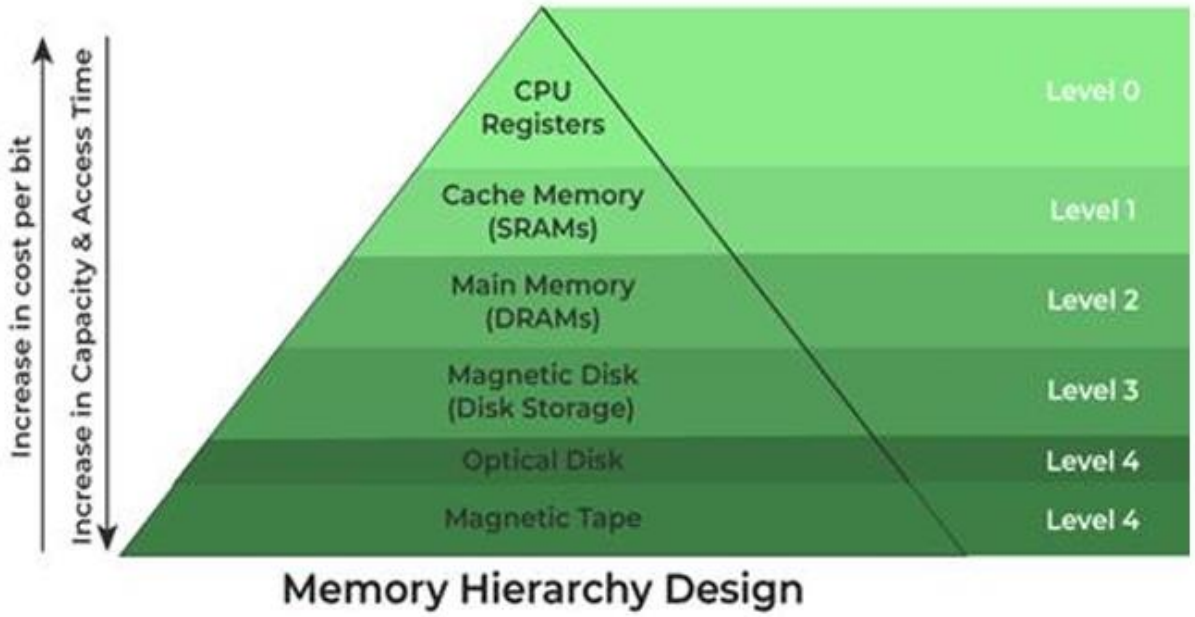
• **Masked ROM**: Data permanently programmed during manufacturing.

• **PROM (Programmable ROM)**: One-time programming with a special device.

• **EPROM (Erasable Programmable ROM)**: Erasable with ultraviolet light, then reprogrammable.

• **EEPROM (Electrically Erasable Programmable ROM):** Electrically erasable and reprogrammable.

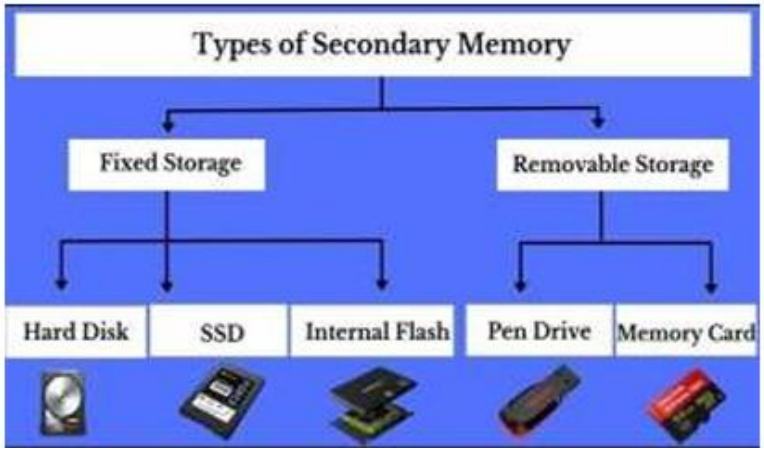
2. Investigate the hierarchical organization of memory in computing systems with examples.



Memory Hierarchy Design

- **Memory hierarchy in computing**: Memory in computing systems is organized hierarchically into multiple levels, with each level offering different characteristics in terms of speed, capacity, and cost.
- **Cache Memory**: Fast, small memory integrated into CPU, stores frequently accessed data for rapid retrieval.
- **Main Memory (RAM):** Larger but slower than cache, holds executing programs and data temporarily.
- **Secondary Storage:** Hard drives, SSDs, etc., offer large storage but slower access than RAM, used for long-term data storage.

3. Evaluate the types and functionality of secondary memory devices exploring their role in data storage.



Hard Disk Drives (HDDs):

- Store data magnetically on rotating platters.
- Commonly used for long-term storage in computers and servers.

Solid State Drives (SSDs):

- Store data using flash memory chips.
- Provide faster access times and higher data transfer rates compared to HDDs.

USB Flash Drives:

- Portable storage devices that use flash memory.
- Used for transferring files between computers and carrying data on the go.

External Hard Drives:

- Similar to internal HDDs but housed in external enclosures.
- Provide additional storage capacity for backups and large data sets.

4. Compare and contrast Volatile and Non-volatile memory types.

Volatile Memory	Non-Volatile Memory
It is a type of computer memory that stores the data temporarily.	It is also a type of computer memory that stores the data permanently.
It requires a continuous electric current to maintain its saved data.	It retains the data in the system even when the power is gone.
It has less storage capacity, more expensive and provides easy data transfer.	It has less more capacity, less expensive and data transfer is complex.
Ex: RAM, Cache Memory	Ex: CD, HDD, Pen drive, SD Card

5. Differentiate between the concepts of temporal and spatial locality in memory access patterns.

Temporal	Spatial
A recently executed instruction is likely to be executed again very soon.	Nearby instructions to recently executed instruction are likely to be executed soon.
It refers to the tendency of execution where memory location that have been used recently have a access.	It refers to the tendency of execution which involve a number of memory locations .
It is also known as locality in time.	It is also known as locality in space.
It repeatedly refers to same data in short time span.	It only refers to data item which are closed together in memory.

6. Describe the importance of temporal and spatial locality with respect to memory.

1. **Temporal locality:**

- Recent access predicts future access, a principle vital for caching mechanisms.
- Caching stores recently accessed data in faster memory layers like CPU caches.
- Enhances performance by reducing access times for frequently accessed data.

2. **Spatial locality:**

- Access of one storage location indicates likely access to nearby addresses.
- Programs typically access data sequentially, benefiting from spatial locality.
- Fetching adjacent data into faster storage layers optimizes data retrieval by exploiting this principle.

7. Formulate the concept of gaming development in which memory addressing and access requests play a role.

Memory Management in gaming:

Game performance relies on efficiently using memory. Here's the key concept:

• **Memory Addresses**: Like house addresses, these tell the game engine where to find game assets (textures, models) in memory.

• **Access Requests**: Retrieving data (assets) from memory takes time. Faster memory (cache) is better, but space is limited.

The Challenge:

• Balancing complex graphics with smooth gameplay requires optimizing how the game engine finds and retrieves assets.

Developers use tricks:

• **Smart Placement**: Putting frequently used assets close together in memory reduces retrieval time.

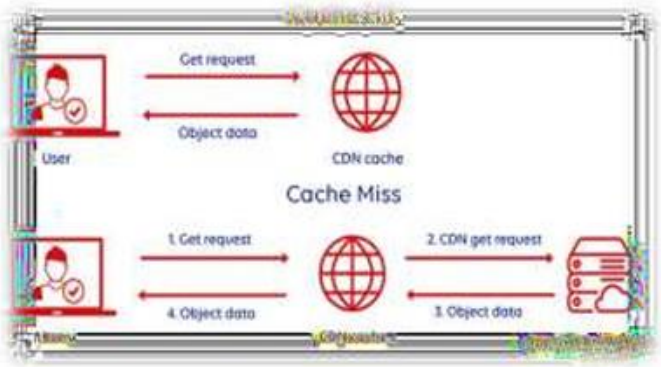
• **Data Locality**: Reusing recently accessed data minimizes memory access.

• **Prefetching**: Anticipating future needs and loading data in advance.

• **Level Streaming**: Loading game sections on demand (open-world games)

The Result: By optimizing memory access, developers create immersive games that run smoothly without sacrificing visual quality.

8. Investigate the significance and impact of "Hit" and "Miss" events in cache memory, detailing how theseoccurrences influence system performance.

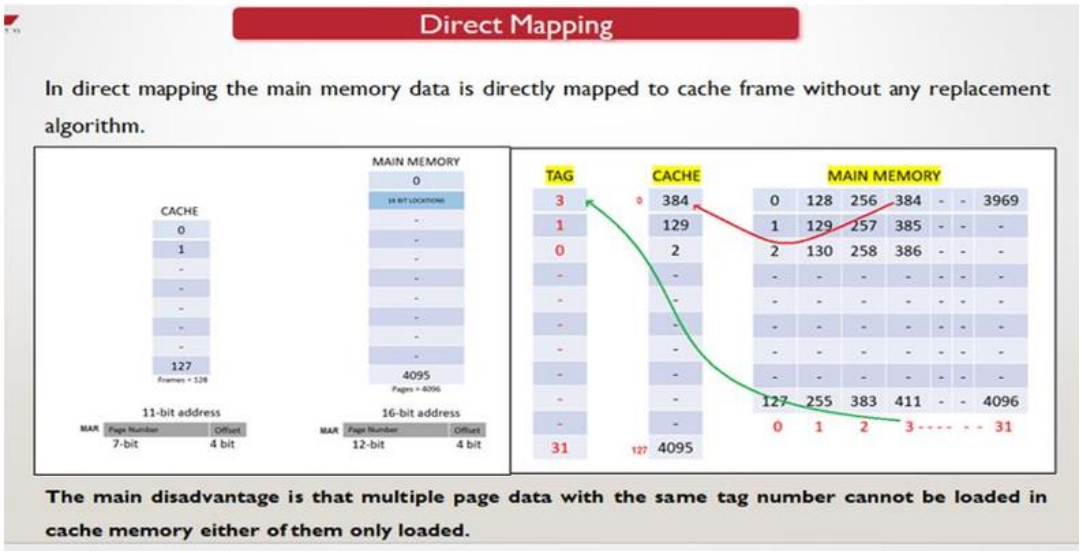


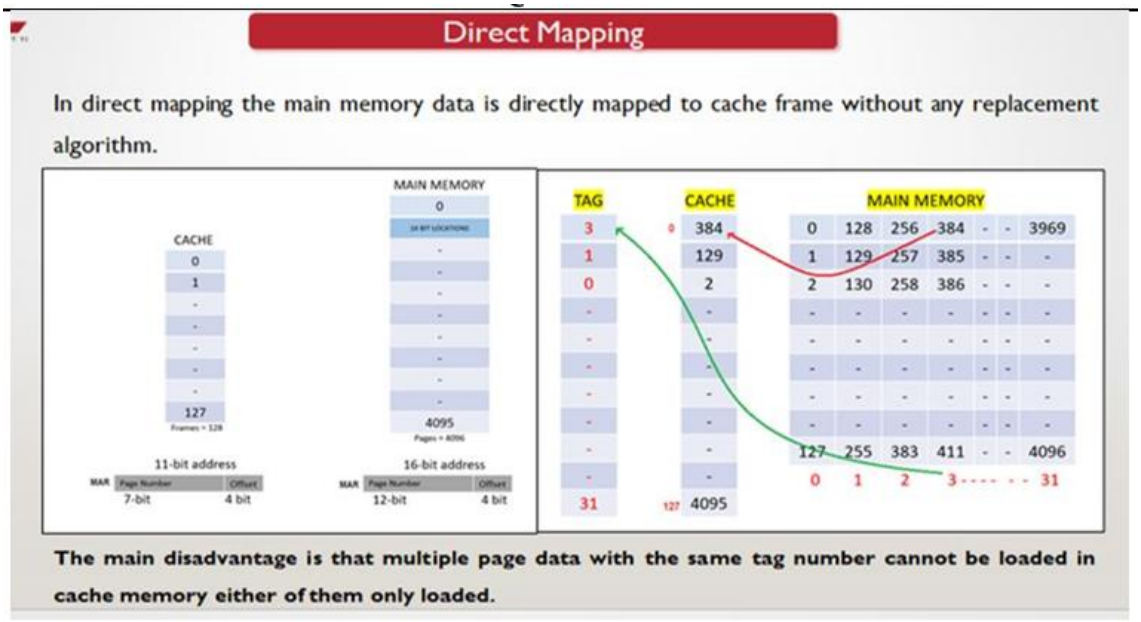
In cache memory, "Hit" and "Miss" events refer to the outcome of the CPU's attempt to access data:

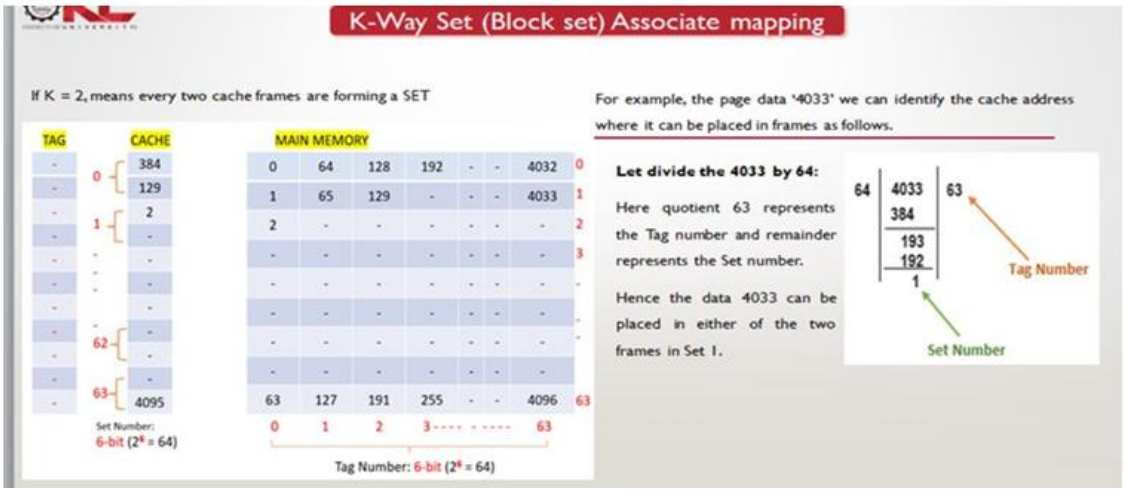
• In cache memory, a "hit" occurs when the CPU finds the needed data in the cache, enabling quick retrieval without accessing main memory.

• A "miss" happens when the data is not found in the cache, requiring the CPU to fetch it from slower main memory, resulting in slower execution.

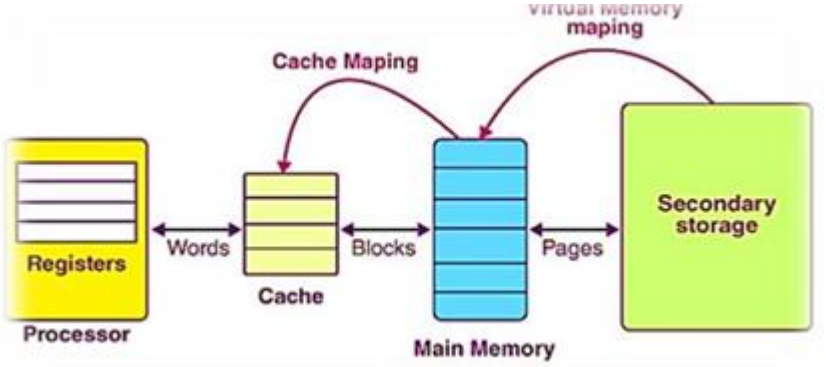
9. Designing a gaming console with a focus on cache memory efficiency, outline and exemplify three mapping procedures, each tailored to specific scenarios to optimize performance in gaming console architecture.







10. Interpret the operation of cache memory contribute to improving the performance of a processor, and what advantages does it offer in terms of speed and efficiency?



• Cache memory is a speedier, smaller section of memory with an access time that is comparable to registers.

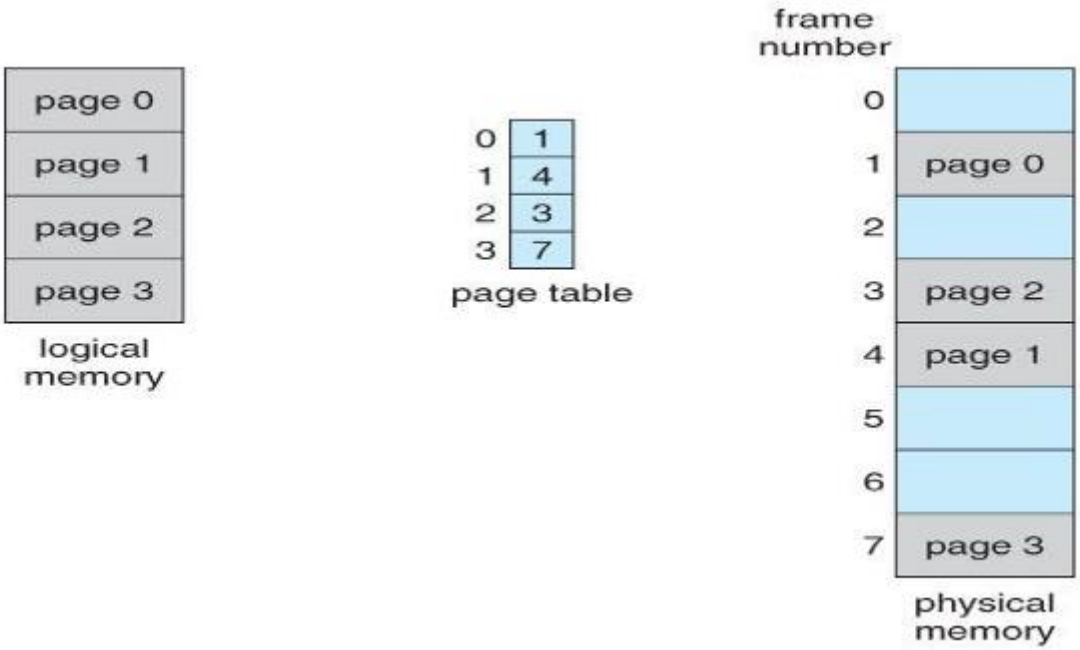
• Cache memory has a shorter access time than primary memory in a memory hierarchy. Since cache memory is typically relatively little, it serves as a buffer.

• Cache provides faster access.

• It acts as buffer between CPU and main memory (RAM).

• Cache primary role is to reduce the average time taken to access data, thereby improving overall system performance.

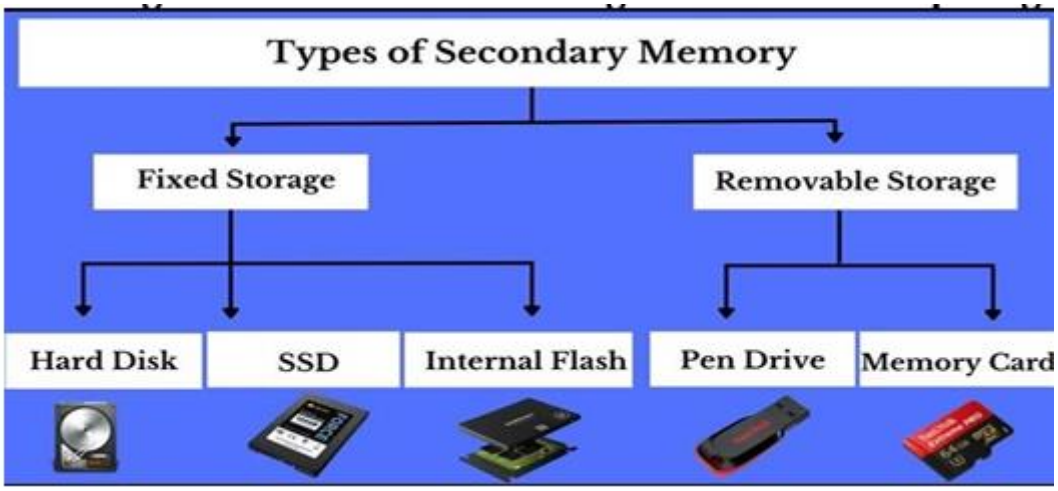
11. Describe the virtual memory system, focusing on the role of page tables.



CO-4 LONG ANWERS

1. Virtual memory provides alternate memory addresses for programs, which are converted to real addresses during execution.
2. Page Table, a data structure in the operating system, maps virtual to physical addresses.
3. It provides frame numbers, indicating where each page is stored in main memory, facilitating memory management.

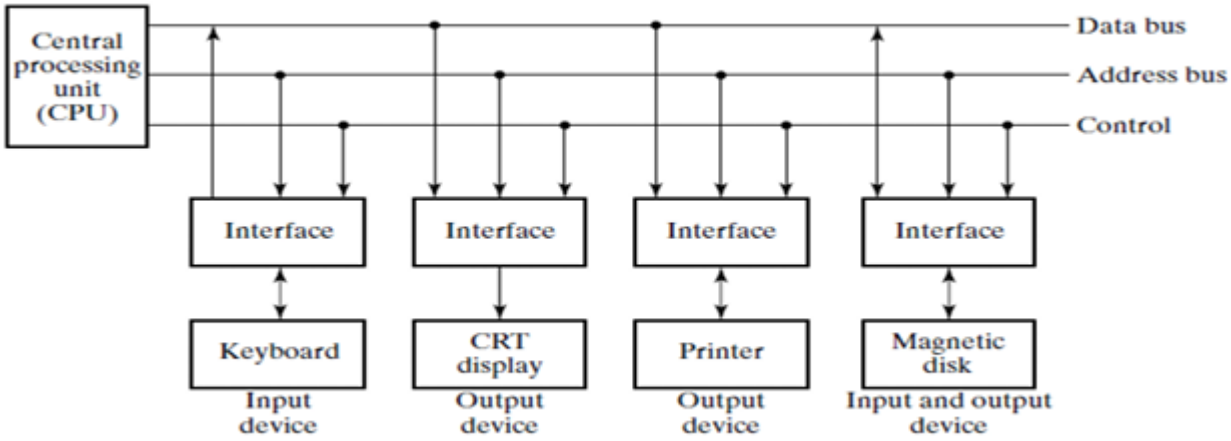
12. Analyze the role and significance of external storage solutions in computing systems.



External storage devices come in various forms, each with a specific role:

- **HDDs:** High-capacity workhorses for bulk data (movies, archives) at a lower cost.
- **SSDs**: Blazing-fast for frequently accessed data (applications, games) but pricier per gigabyte.
- **USB Drives**: Compact and portable for data transfer and booting certain systems (limited capacity).
- **Portable SSDs**: Speedy and portable for data transfer on the go (more expensive than USB drives).
- **NAS**: Centralized network storage for sharing files across devices or backing up small businesses.
- **Cloud Storage**: Virtual storage accessed from anywhere for off-site backups and file synchronization (requires internet).

13. Elaborate role and architecture of IO buses and interface modules with diagram.



The bus typically consists of three main types of lines:

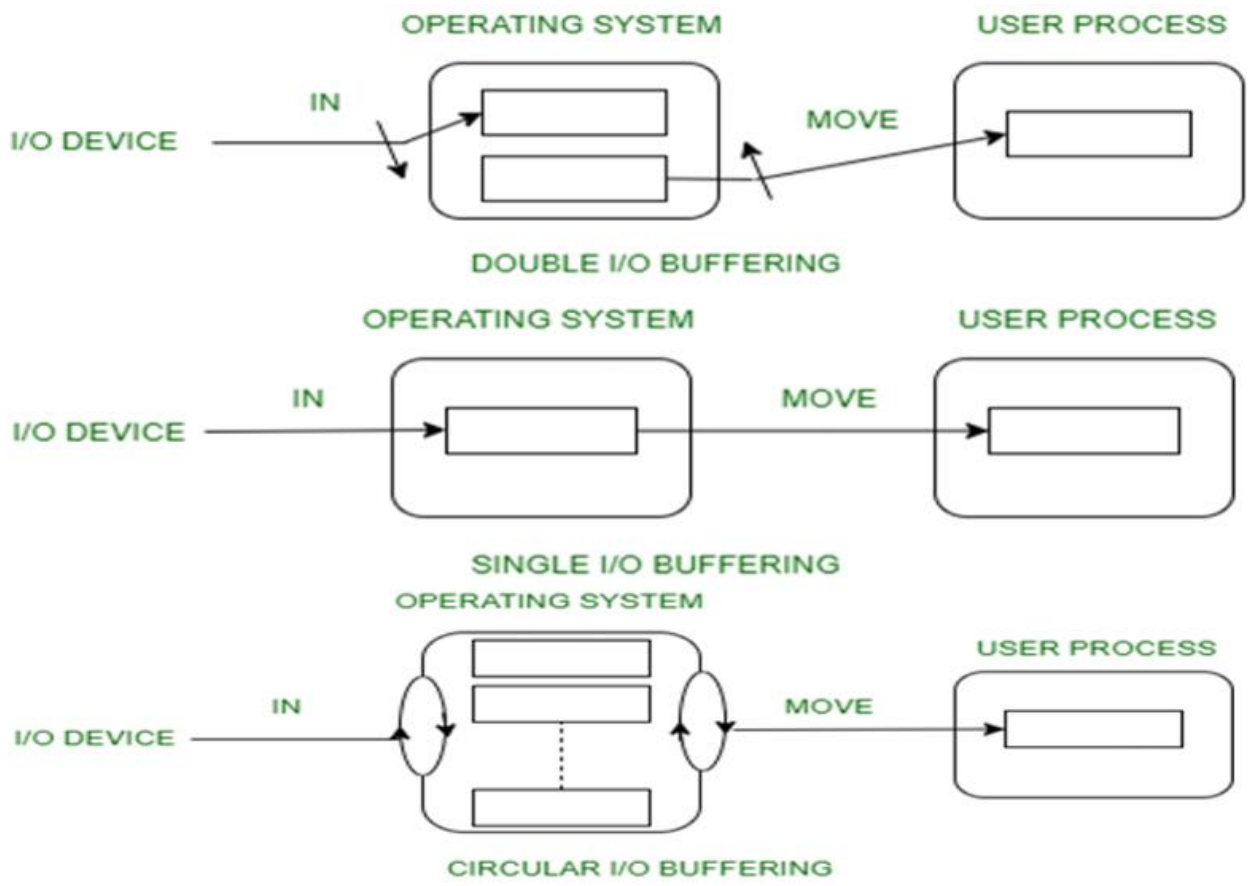
1. **Data bus**: These carry the actual data being transferred between the CPU and devices.
2. **Address bus**: These specify the location (address) of the data on the device or in memory.
3. **Control bus**: These carry control signals for synchronization and coordination (e.g., read/write, start/stop).

14. Analyze buffering with its types in the context of IO operations.

Ans:

• Buffering creates a synchronization between two devices having different processing speed. For example, if a hard disc (supplier of data) has high speed and a printer (accepter of data) has low speed, then buffering is required.

• Buffering is also required in cases where two devices have different data block sizes.



• **Single Buffering**: Simple, but CPU stalls if receiver isn't ready.
• **Double Buffering**: Smoother data flow but requires more memory and can overflow.
• **Circular Buffer**: Efficient memory usage, avoids overflow, but adds complexity.

15. Analyze handshaking with its types in the context of IO communication.

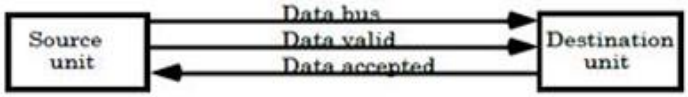
Ans:

Handshaking Signals:
• These are dedicated control lines that carry information about the data transfer status.
• Common handshaking signals include:

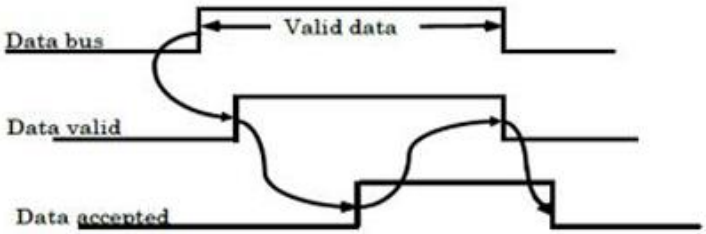
1. Request: Sent by the receiver to indicate it's ready to receive data.
2. Acknowledge: Sent by the receiver after successfully receiving data.
3. Grant: Sent by the sender to inform the receiver that data is being sent.
4. Busy: Sent by the receiver to indicate it's not ready to receive data.

Source Initiated Handshaking:

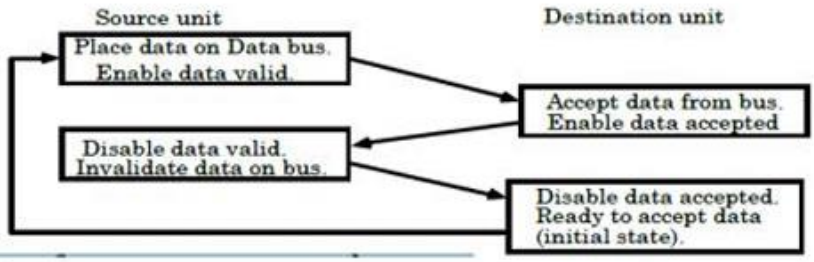
Block Diagram



Timing Diagram

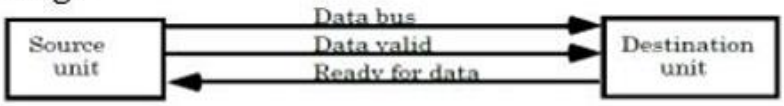


Sequence of Events

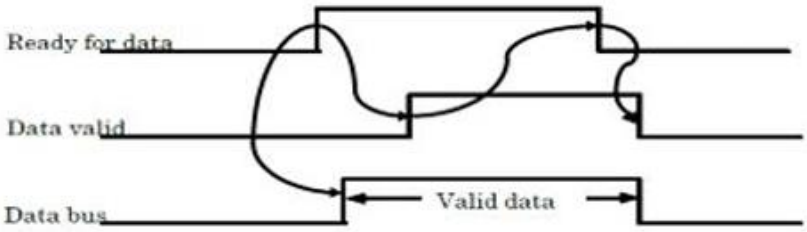


Destination Initiated handshaking:

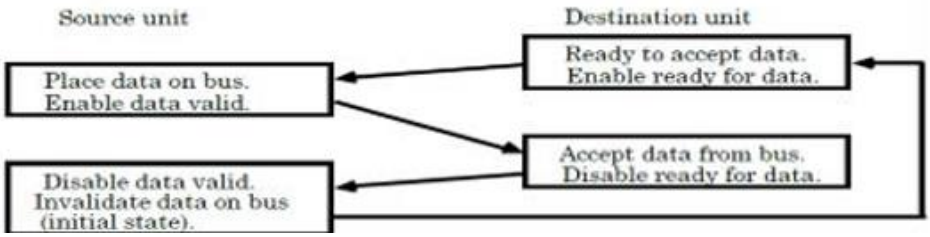
Block Diagram



Timing Diagram



Sequence of Events



16. Construct a flowchart for the utilization of Programmed I/O in a traffic light control system.

Ans:

Flow: Traffic Light Control with Programmed I/O

Start

1. **System Initialization**:

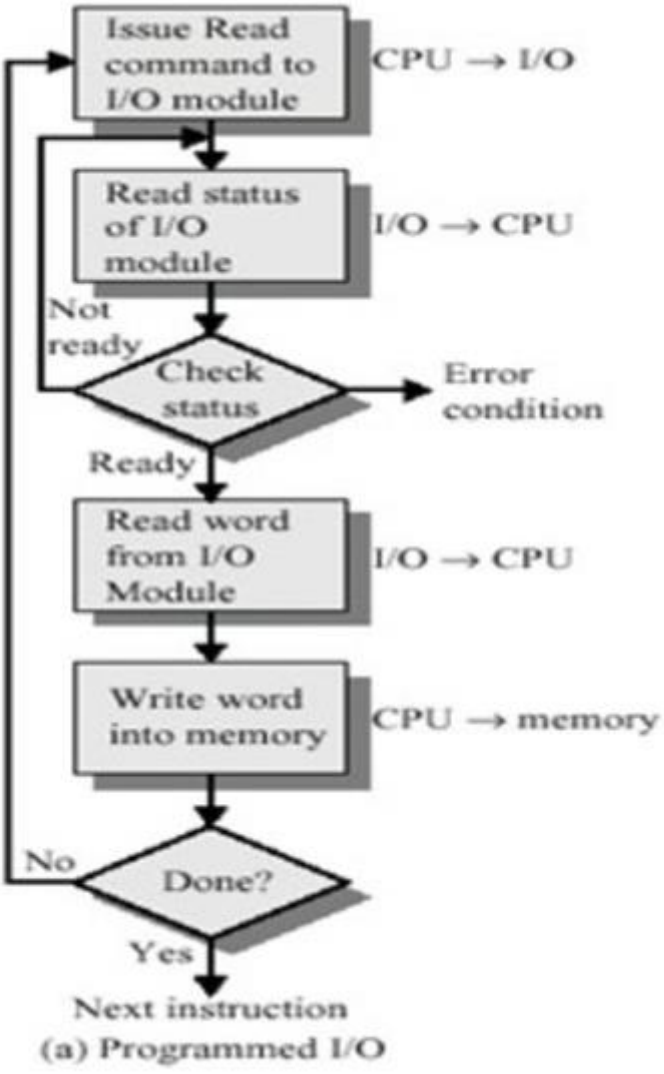
- Set up timers for each traffic light phase (red, yellow, green).
- Initialize variables for storing traffic light states.

2**. Main Loop**:

- Check timer for current phase (red, yellow, green).
- Red Phase: Turn on red lights for all directions.
- Yellow Phase (if timer for red phase expires): Turn on yellow lights for all directions.
- Green Phase (if timer for yellow phase expires): Turn on green lights for designated direction(s).

3. **Loop**: Return to step 2 and continue the cycle, switching between phases based on timer expirations.

End



(a) Programmed I/O

17. In designing a real-time system for an autonomous drone, evaluate the pros and cons of Programmed IO versus Interrupt-driven IO, and recommend the most suitable approach for ensuring timely data processing and control.

Ans:

Programmed IO:

- **Pros:**

1. Simpler to implement, requires less hardware overhead.
2. Deterministic behavior, the CPU has complete control over the timing of I/O operations.

- Cons:

1. **Poor real-time performance**: The CPU constantly needs to check the status of the I/O devices, wasting valuable processing cycles. This can lead to missed deadlines and delays in processing sensor data critical for real-time control.
2. **Limited scalability**: As the number of I/O devices increases, the amount of code needed to manage them also grows, making the program complex and difficult to maintain.

Interrupt-driven IO:

- **Pros:**

1. **Improved real-time performance**: The CPU is only interrupted when an I/O device needs attention, allowing it to focus on critical tasks most of the time. This minimizes processing delays and ensures timely responses to sensor data.
2. **Scalability**: Interrupt-driven IO is more scalable as additional devices can be added without significantly impacting the main program structure. The interrupt handler for each device can be written independently.

- **Cons:**

1. **More complex to implement**: Requires setting up interrupt handlers for each device and managing interrupt priorities to ensure proper handling of multiple devices requesting attention simultaneously.

2. **Less predictable timing**: The timing of program execution can be influenced by the arrival of interrupts, making it slightly less deterministic compared to programmed IO.

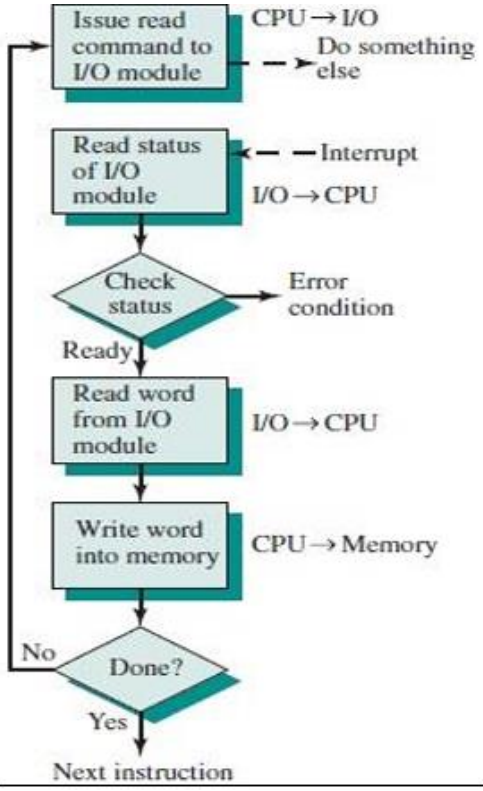
Recommendation for Autonomous Drone:

For an autonomous drone, interrupt-driven IO is the most suitable approach. Timely processing and control are critical for safe and reliable flight.

18. Utilizing interrupt driven I/O, build a flow diagram for the purpose of sensing data from external devices in a data acquisition system

Ans:

1. System Setup: Configure I/O and enable interrupts for the sensor device.
2. Main Loop: Perform non-critical tasks (wait for interrupts).
3. Interrupt (Sensor Data Ready): Read sensor data, store, set data ready flag.
4. Main Loop (continued): Check data ready flag. If data ready: process data (calculations, display).
5. Loop: Repeat steps 2-4.

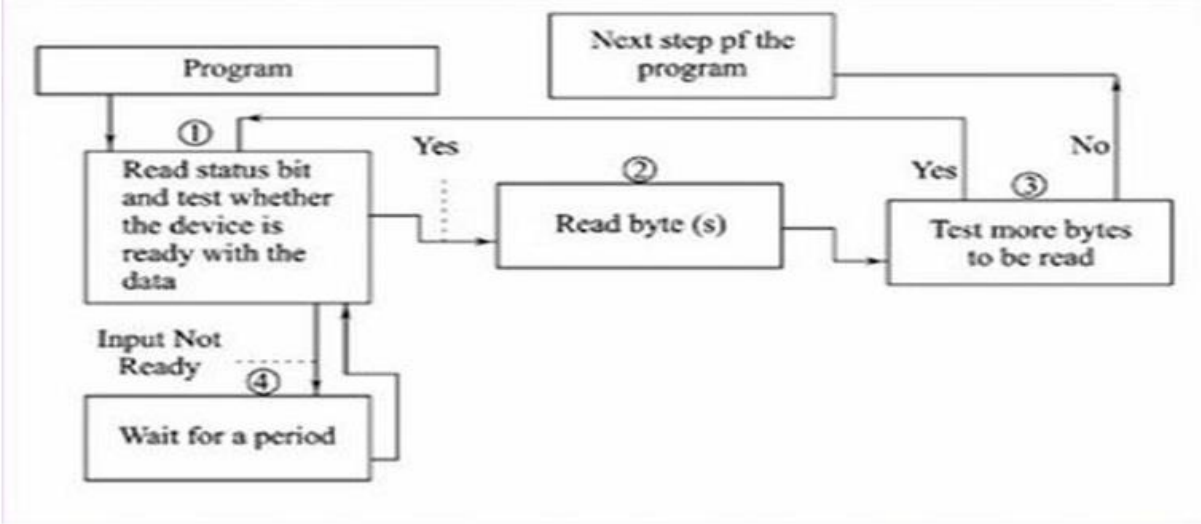


CO-4 LONG ANWERS

19. Examine the process of Programmed IO through the use of a block diagram and workflow.

Ans:

Block Diagram:



Workflow:

(i) CPU sends an I/O request to the I/O device.

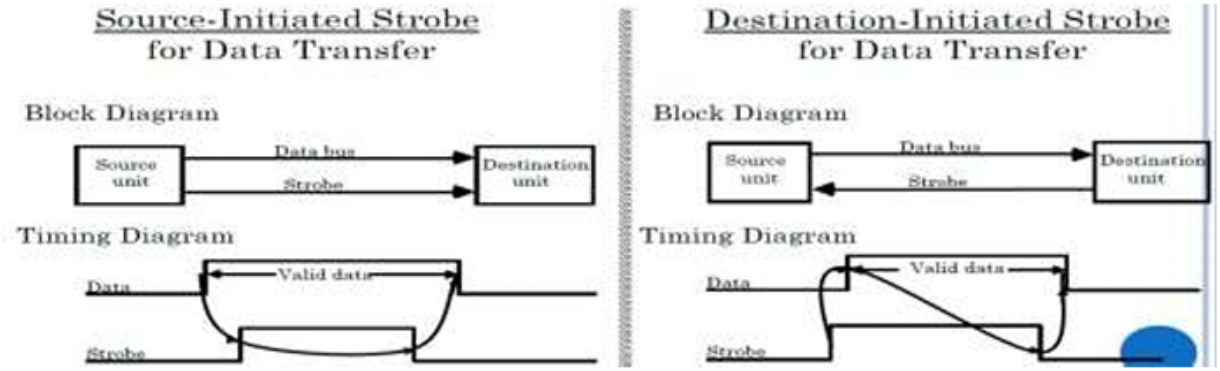
(ii) I/O device performs the requested operation.

(iii) I/O device signals the CPU upon completion.

(iv) CPU checks the status of the operation.

(v) If the operation is complete, CPU proceeds with the next instruction.

20.Illustrate the concepts of source-initiated and destination-initiated strobe pulses in data transmission.



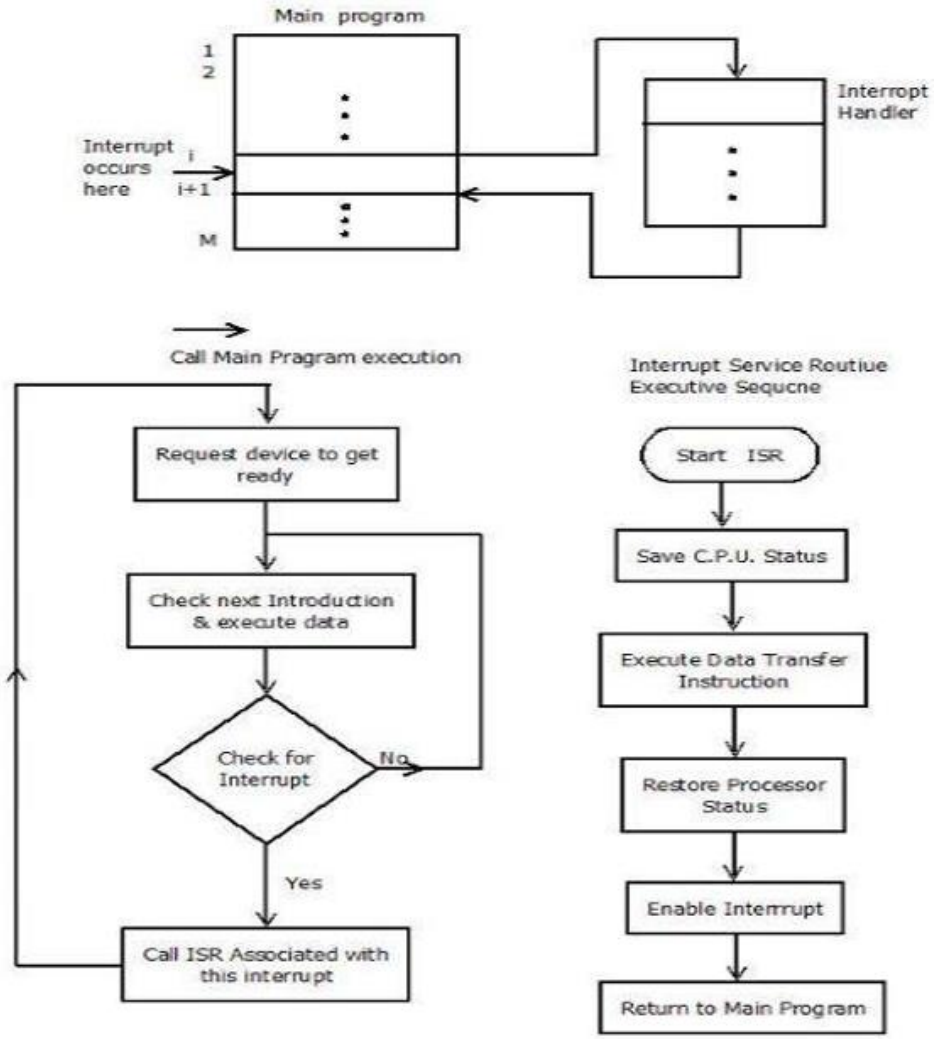
Source-initiated strobe pulses are signals sent by the transmitting device to indicate the beginning or end of data transmission, helping synchronize communication between devices.

Destination-initiated strobe pulses are signals generated by the receiving device to acknowledge the receipt of data or to request more data, facilitating reliable and efficient data transmission in digital communication systems.

21. Examine the process of interrupt driven IO through the use of a block diagram and workflow.

Ans:

Block Diagram:



Workflow:

(i) CPU sends an I/O request to the I/O device.

(ii) CPU continues executing other instructions.

(iii) I/O device performs the requested operation independently.

(iv) Upon completion, I/O device sends an interrupt signal to the CPU.

(v) CPU suspends its current operation and handles the interrupt.

(vi) CPU services the I/O request.

(vii) CPU resumes its previous operation.

22. Identify various cache replacement policies that are useful to manage the cache memory.

- **First-in-first-out (FIFO) policy:** The earliest inserted item in the cache will be evicted when a new item needs to be inserted.
- **Last-in-first-out (LIFO) policy:** The last item inserted in the cache will be evicted first.
- **Least-recently used (LRU) policy:** The item which is least recently used will be evicted first. This is one of the most simple and common cache replacement policies.
- **Least-frequently-used (LFU) policy:** The cache algorithm maintains a counter on the number of times an item in the cache is accessed. It will evict the least frequently accessed item to add a new item.
- **Most-recently used (MRU) policy:** The item which is most recently used will be evicted first. This policy is useful, when the chance of repeating the same request soon is unlikely (like scrolling through a social media feed or flipping through a photo album).
- **Time-to-live (TTL) policy:** If an item remains in the cache beyond a given period without being accessed, the cache algorithm would discard it, to make room for a new item.
- **Random replacement (RR) policy:** The cache algorithm randomly selects an item to evict.