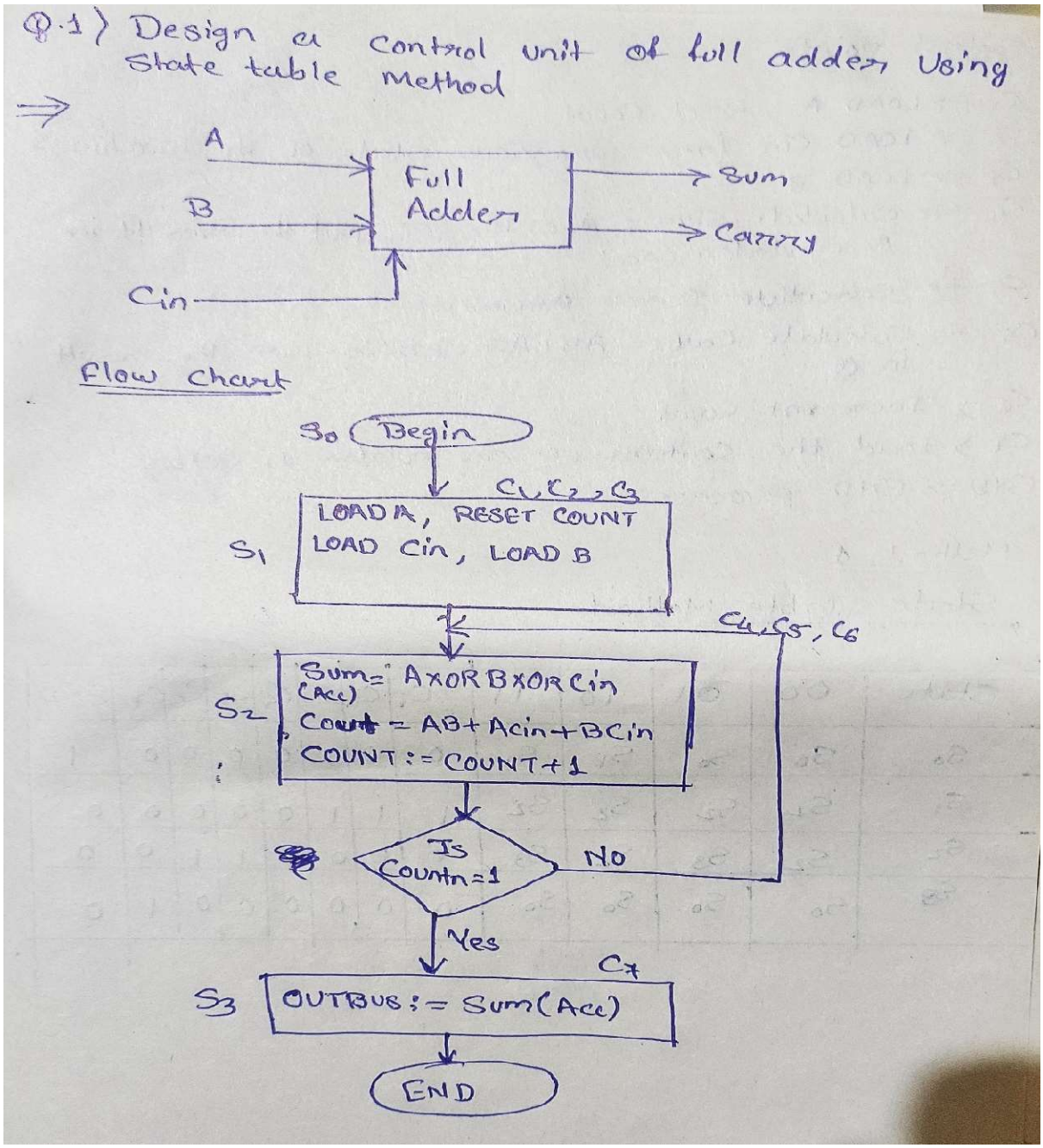
NAME: GANESH NARAYAN MOROLIYA

ROLL NO: 102

SECTION : A

ASSIGNMENT NO 2

Q.1) Design the control unit of full adder using state table method.



Control points

$C_1 \Rightarrow$ LOAD A , Reset Count

$C_2 \Rightarrow$ LOAD Cin from Previous State of the machines

$C_3 \Rightarrow$ LOAD B

$C_4 \Rightarrow$ Calculate Sum = A xor B xor Cin put the result in Accumulator (Acc)

~~$C_5 \rightarrow$ Calculate Sum = A xor B x.~~

$C_5 \Rightarrow$ Calculate Cout = $AB + ACin + BCin$ put the result in Cy

$C_6 \Rightarrow$ Increment Count

$C_7 \Rightarrow$ Load the contents of accumulator in outbus

END $\Rightarrow$ END Process

Method 1

State Table Method

State	00	01	10	11	C_1	C_2	C_3	C_4	C_5	C_6	C_7	END
S_0	S_0	S_0	S_1	S_1	0	0	0	0	0	0	0	1
S_1	S_2	S_2	S_2	S_2	1	1	1	0	0	0	0	0
S_2	S_2	S_3	S_2	S_3	0	0	0	1	1	1	0	0
S_3	S_0	S_0	S_0	S_0	0	0	0	0	0	0	1	0

Q.2) Explain various characteristics of memory devices and state the design objectives of memory system.

Ans=> Memory devices come in various forms, each with its own set of characteristics and design objectives

Capacity: Capacity refers to the amount of data a memory device can store. It's typically measured in bytes (e.g., kilobytes, megabytes, gigabytes).

Speed: Speed refers to how quickly data can be read from or written to the memory device. It's usually measured in nanoseconds (ns) or megahertz (MHz).

Volatility: Volatility determines whether the data stored in the memory is retained when power is removed. Volatile memory loses its data when power is turned off, while non-volatile memory retains data even without power.

Access Method: Memory devices can have different access methods, such as random access or sequential access. Random access allows data to be accessed directly, while sequential access requires accessing data in a specific order.

Cost: Cost is a crucial factor in memory device selection. Different memory technologies have varying costs per unit of storage.

Durability: Durability refers to the ability of a memory device to withstand read and write operations over time without failure.

Power Consumption: Power consumption is an important consideration, especially in portable devices. Low-power memory devices help conserve battery life.

Design objectives of memory systems include:

Reliability: Memory systems must reliably store and retrieve data without errors. This involves using error detection and correction techniques to ensure data integrity.

Performance: Memory systems should provide fast access to data to minimize processing delays. This involves optimizing access times and throughput.

Scalability: Memory systems should be scalable to accommodate increasing storage demands. This may involve using modular designs that allow for easy expansion.

Compatibility: Memory systems should be compatible with existing hardware and software standards to ensure interoperability.

Cost-effectiveness: Memory systems should provide the required capacity and performance at a reasonable cost. This involves balancing performance requirements with budget constraints.

Power Efficiency: Memory systems should minimize power consumption to extend battery life in portable devices and reduce energy costs in data centers.

Security: Memory systems should implement security features to protect sensitive data from unauthorized access or tampering. This may include encryption, access controls, and secure erase capabilities.

Q.3) Explain in brief the RAM structure and working. Also explain the various addressing schemes used in RAM.

Ans=> RAM, or Random Access Memory, is a type of computer memory that allows data to be stored and retrieved randomly, meaning any storage location can be accessed directly.

Structure: RAM is typically organized into memory cells, each containing a unique address. These memory cells are grouped into bytes, with each byte consisting of 8 bits. RAM chips are organized into arrays of rows and columns, with each intersection of a row and column representing a memory cell.

Working: When the computer needs to read or write data to RAM, it sends a memory address along with the data to the RAM module. The RAM then activates the appropriate row and column based on the address to access the desired memory cell. Data is then either read from or written to that cell.

Addressing Schemes:

Flat Addressing: In this scheme, each memory location is assigned a unique address, starting from 0 and incrementing sequentially. This is the simplest and most commonly used addressing scheme.

Bank Addressing: RAM is divided into multiple banks, each with its own set of addresses. This scheme allows for parallel access to multiple memory banks, increasing memory throughput.

Segmented Addressing: Memory is divided into segments of variable lengths, and each segment is assigned a unique address. This scheme was commonly used in older computer architectures but has largely been replaced by flat addressing.

Paging: Memory is divided into fixed-size pages, and each page is assigned a unique address. This allows for efficient memory management, virtual memory systems, and memory protection.

Q.4) Explain in short ferrite core memories.

Ans=> Ferrite core memory, also known as core memory or magnetic core memory, was a type of computer memory used in the mid-20th century.

Structure: Ferrite core memory consisted of a grid of tiny ferrite rings, typically arranged in a two-dimensional array. Each ferrite core represented a single bit of data storage, with the ability to store either a 0 or a 1.

Working: To write data to a ferrite core memory, a current was passed through selected wires called the X and Y lines, creating a magnetic field that would magnetize the ferrite core in a specific direction, representing either a 0 or a 1. Reading data from the memory involved inducing a voltage in the sense wire of the selected core, which would detect the magnetic orientation of the core and determine the stored bit.

Advantages:

Non-Volatile: Ferrite core memory was non-volatile, meaning it retained data even when power was turned off.

Reliability: It was highly reliable and durable, with a long lifespan compared to other memory technologies of its time.

Speed: Ferrite core memory was relatively fast compared to alternative technologies of its era.

Disadvantages:

Cost: It was expensive to manufacture due to the intricate wiring required for the core matrix.

Density: Its density was limited compared to later semiconductor-based memory technologies like RAM chips.

Power Consumption: It consumed more power compared to modern memory technologies.

Q.5) Write a short note on stack replacement policies.

Ans=> Stack replacement policies are used in computer systems, particularly in memory management and cache management, to determine which items should be removed (evicted) from the stack when it becomes full and a new item needs to be pushed onto it. The stack replacement policy helps decide which item is the least useful or the least likely to be accessed again in the near future.

There are several common stack replacement policies:

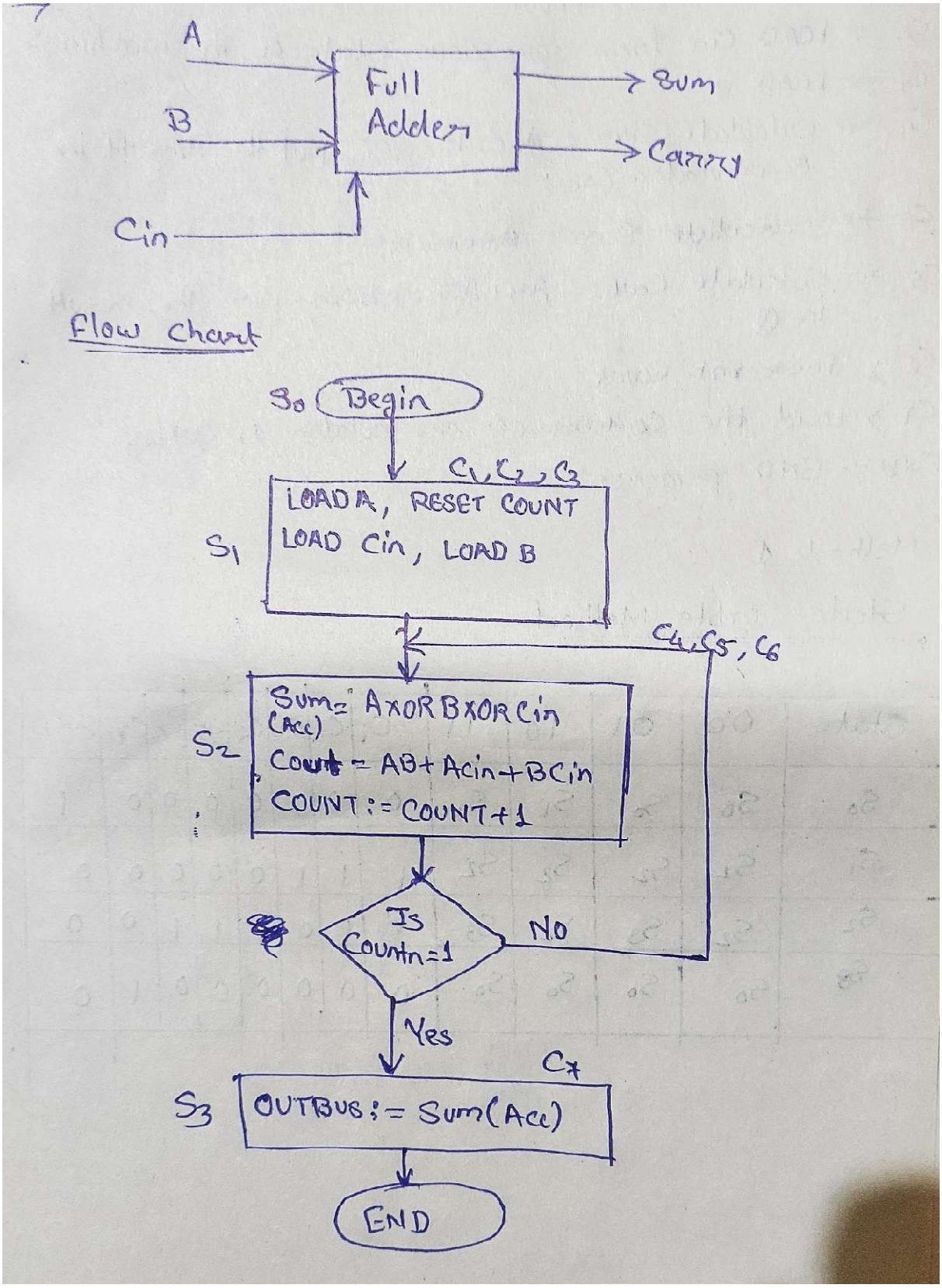
FIFO (First-In, First-Out): In FIFO policy, the least recently accessed item is evicted first. This policy follows the principle of removing items in the order they were added. While FIFO can be fair, it may not consider the access patterns of the items, leading to potentially inefficient cache or memory usage.

Optimal: The Optimal replacement policy, also known as Belady's optimal algorithm, replaces the item that will not be used for the longest period of time in the future. It is considered the theoretical best strategy as it minimizes the number of page faults. However, it requires knowledge of future memory accesses, which is usually not feasible. As a result, it is often used as a benchmark to compare other replacement policies' performance. Despite its impracticality, it provides insight into the potential performance ceiling of replacement policies.

LRU (Least Recently Used): In LRU policy, the least recently accessed item is evicted first. This policy assumes that items that have not been accessed for a long time are less likely to be accessed in the near future. LRU is widely used due to its effectiveness in capturing temporal locality, but it may require complex data structures or tracking mechanisms to implement efficiently.

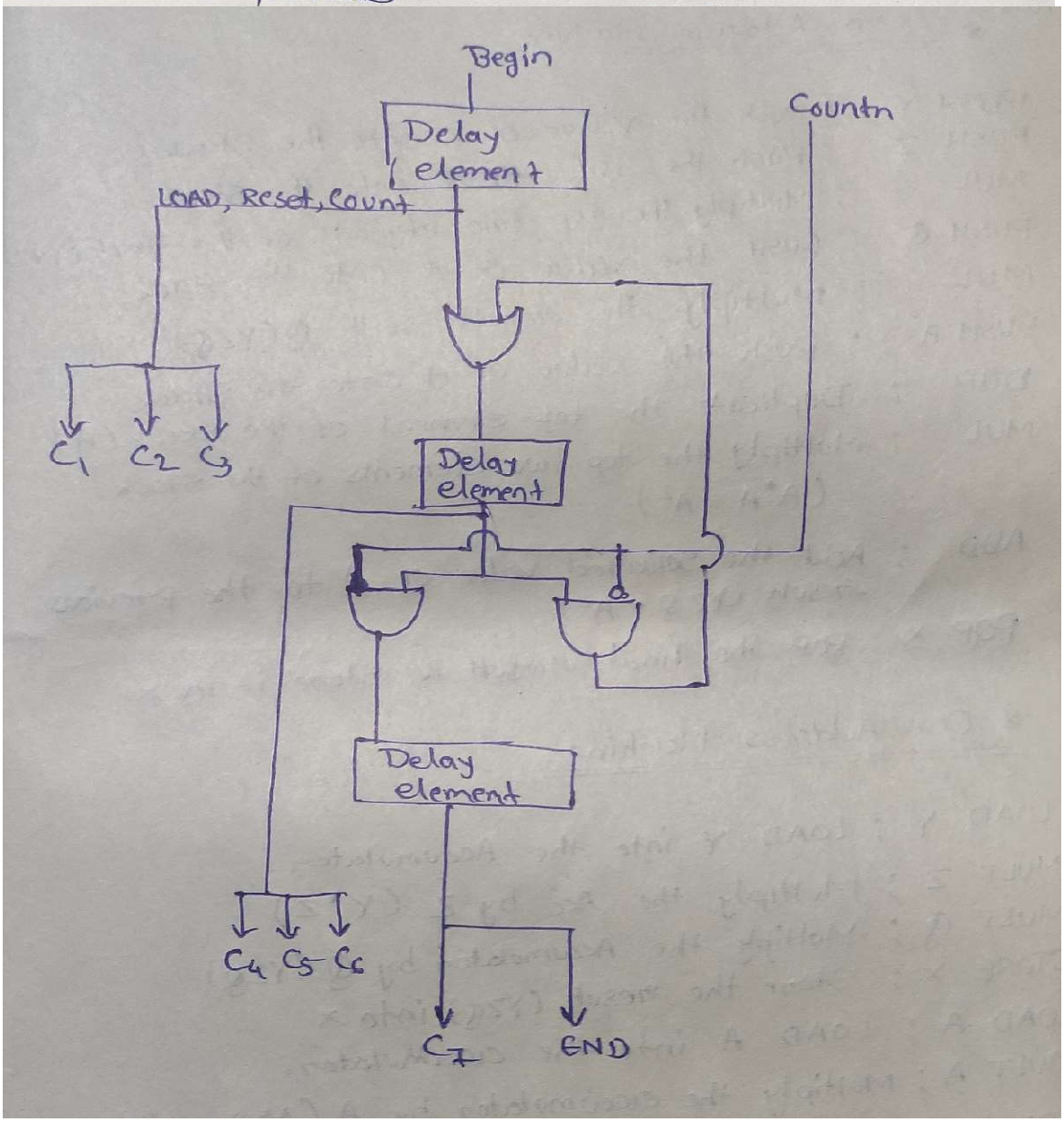
MRU (Most Recently Used): In MRU policy, the most recently accessed item is evicted first. This policy assumes that the most recently accessed item is likely to be accessed again soon. MRU can be useful in certain scenarios but may not always be optimal for capturing access patterns.

Q.6) Design the control unit of full adder using delay element method.



Control Points

C_1 ⇒ LOAD A , Reset Count
C_2 ⇒ LOAD Cin from Previous State of the machine
C_3 ⇒ LOAD B
C_4 → Calculate Sum = A xor B xor Cin put the result in Accumulator (Acc)
~~C_5 → Calculate Sum = A xor B.~~
C_5 ⇒ Calculate Cout = AB + ACin + BCin put the result in Cy
C_6 ⇒ Increment Count
C_7 ⇒ Load the contents of accumulator in Outbus
END ⇒ END Process



Q.7) Explain a static bipolar RAM cell and a dynamic MOS Ram cell.

Ans=> Static Bipolar RAM Cell:

In a static bipolar RAM cell, each memory cell is typically constructed using several bipolar transistors. The basic structure of a static RAM (SRAM) cell consists of two cross-coupled inverters made up of bipolar transistors (typically BJTs - Bipolar Junction Transistors) and two access transistors.

explanation of how it works:

The cross-coupled inverters store the data as long as power is supplied to the RAM.

The access transistors control the read and write operations to the cell. When a read or write operation is required, the access transistors enable the data to flow in or out of the cross-coupled inverters.

The data is maintained as long as power is supplied to the memory cell, making it a type of volatile memory.

Static RAM is known for its fast access times and low power consumption but requires more space compared to dynamic RAM due to the additional transistors needed for each memory cell.

Dynamic MOS RAM Cell:

In dynamic RAM (DRAM) cells, each memory cell consists of a single transistor and a capacitor. The basic structure of a DRAM cell is simpler compared to SRAM but requires additional circuitry to refresh the data periodically due to charge leakage from the capacitor.

explanation of how it works:

The transistor acts as a switch, controlling the flow of charge between the capacitor and the bit line.

The capacitor stores the data in the form of electric charge. When the transistor is turned on, the capacitor either charges to represent a logic 1 or discharges to represent a logic 0.

To read the data, the transistor is activated, and the charge on the capacitor is sensed by the sense amplifier connected to the bit line.

Due to charge leakage, the capacitor loses its charge over time. Therefore, DRAM requires periodic refresh operations to rewrite the data back into the cells, which adds complexity and overhead.

Dynamic RAM is known for its high density and lower cost compared to SRAM but has slower access times and higher power consumption due to the need for refreshing.

Q.8) What is seek time and latency time?

Ans=> Seek time and latency time are both measures of the time it takes for a storage device, typically a hard disk drive (HDD), to access data. They are essential metrics for understanding the performance of storage systems.

Seek Time:

Seek time refers to the time it takes for the read/write heads of a disk drive to move to the specific location (track or cylinder) where the data is stored.

When a request is made to access data stored on a disk, the read/write heads need to move to the correct track or cylinder to read or write the data. The time it takes for this movement to occur is the seek time.

Seek time is typically measured in milliseconds (ms).

Seek time is influenced by factors such as the distance the heads need to travel, the speed of the actuator mechanism, and the design of the disk.

Latency Time:

Latency time, also known as rotational latency or rotational delay, refers to the time it takes for the desired data to rotate under the read/write heads after the heads are positioned over the correct track or cylinder.

Once the read/write heads have reached the correct track, they may need to wait for the desired sector of the disk to rotate under them before the data can be read or written.

Latency time is dependent on the rotational speed of the disk (measured in revolutions per minute or RPM) and the position of the desired sector relative to the current position of the read/write heads.

Latency time is also measured in milliseconds (ms).

For example, in a disk rotating at 7200 RPM, the average latency time would be half the time it takes for one complete revolution, which is around 8.33 milliseconds.

Together, seek time and latency time contribute to the total access time of a disk drive. Lower seek time and latency time generally result in faster data access and better overall disk performance.

Q.9) Write a short note on cache read operation.

Ans=> A cache read operation is a fundamental aspect of a cache memory system, which is used to speed up access to frequently accessed data in a computer system. Here's a brief overview of how a cache read operation typically works:

Cache Hierarchy:

In modern computer architectures, the cache memory is organized into a hierarchy, typically consisting of multiple levels such as L1, L2, and sometimes L3 caches.

Each level of cache is smaller and faster than the next level, with the goal of minimizing the time it takes to access data.

Cache Structure:

Cache memory is organized into cache lines or blocks, each capable of storing a fixed amount of data.

When data is read from main memory, it is loaded into the cache in its entirety or in blocks.

Cache Read Operation:

When the CPU needs to read data from memory, it first checks the cache hierarchy to see if the data is already present in the cache.

The cache is typically indexed using the memory address of the data being accessed. The index is used to determine the location of the data within the cache.

If the data is found in the cache (cache hit), it is quickly retrieved, and the read operation is completed without accessing main memory. This results in a faster access time since cache memory is much faster than main memory.

If the data is not found in the cache (cache miss), the CPU must fetch the data from main memory and load it into the appropriate cache level. This process is known as a cache miss penalty and typically takes longer than a cache hit.

In some cases, particularly with multi-level caches, if the data is not found in the L1 cache, the CPU may check the L2 or L3 cache levels sequentially before accessing main memory.

Cache Replacement Policies:

In the event of a cache miss, where the cache is full and there is no available space to store the requested data, a cache replacement policy is used to determine which cache line to evict to make room for the new data.

Common cache replacement policies include Least Recently Used (LRU), First-In, First-Out (FIFO), and Random Replacement.

Q.10) What are static and dynamic RAMS?

Ans=> Static RAM (SRAM) and Dynamic RAM (DRAM) are two common types of semiconductor-based memory used in computer systems. Here's a brief overview of each:

1)Static RAM (SRAM):

Structure: SRAM stores data using a flip-flop circuit made of cross-coupled inverters. Each SRAM cell typically consists of 6 transistors.

Operation: SRAM does not need to be periodically refreshed to maintain data, hence the name "static." Data is retained as long as power is supplied to the memory.

Access Time: SRAM offers faster access times and lower latency compared to DRAM because it does not require refreshing.

Power Consumption: SRAM consumes more power compared to DRAM due to its more complex structure and constant power requirement.

Density: SRAM has lower storage density compared to DRAM because each memory cell requires more transistors.

2)Dynamic RAM (DRAM):

Structure: DRAM stores data using a capacitor to store charge, and a single access transistor to control the flow of charge in and out of the capacitor. Each DRAM cell typically consists of a single capacitor and transistor.

Operation: DRAM requires periodic refreshing to maintain data, hence the name "dynamic." Data stored in the capacitors leaks over time, necessitating periodic refreshing to rewrite the data back into the cells.

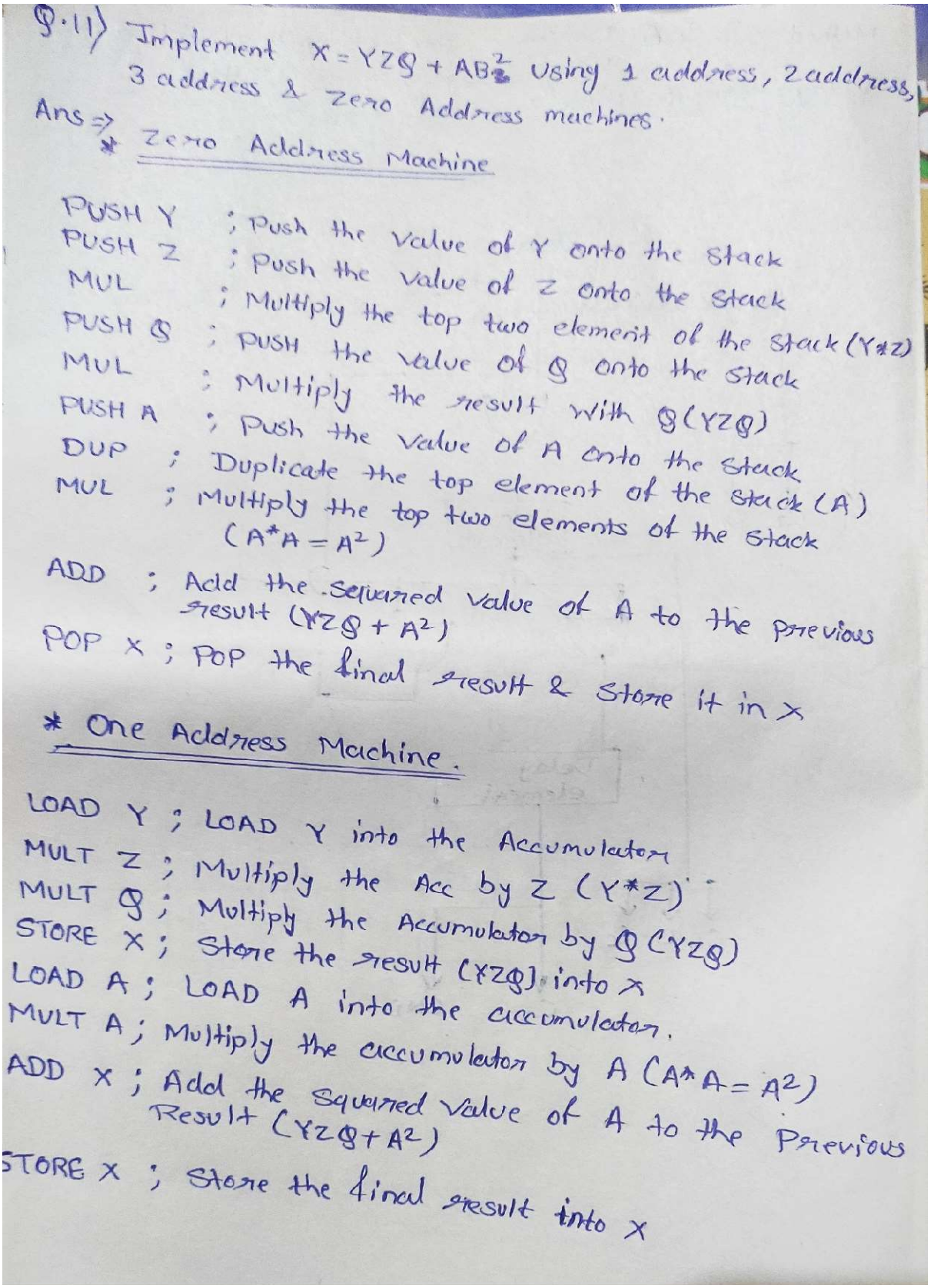
Access Time: DRAM typically has slower access times and higher latency compared to SRAM due to the need for refreshing and the additional overhead of accessing each cell.

Power Consumption: DRAM consumes less power compared to SRAM because of its simpler structure and lower constant power requirement. However, it consumes additional power during the refresh cycles.

Density: DRAM offers higher storage density compared to SRAM because each memory cell requires fewer components, allowing for more cells to be packed into the same area**.**

Q.11) Implement X = YZQ +AB2 using 1 address, 2 address, 3 address and zero address machines.

Ans=>



Q.11) Implement $X = YZQ + AB^2$ using 1 address, 2 address, 3 address & zero Address machines.

Ans ⇒

* Zero Address Machine

```
PUSH Y   ; Push the value of Y onto the stack
PUSH Z   ; Push the value of Z onto the stack
MUL      ; Multiply the top two element of the stack (Y*Z)
PUSH Q   ; PUSH the value of Q onto the stack
MUL      ; Multiply the result with Q (YZQ)
PUSH A   ; Push the value of A onto the stack
DUP      ; Duplicate the top element of the stack (A)
MUL      ; Multiply the top two elements of the stack
           (A*A = A^2)
ADD      ; Add the Squared value of A to the previous
           result (YZQ + A^2)
POP X    ; POP the final result & Store it in X
```

* One Address Machine.

```
LOAD Y   ; LOAD Y into the Accumulator
MULT Z   ; Multiply the Acc by Z (Y*Z)
MULT Q   ; Multiply the Accumulator by Q (YZQ)
STORE X  ; Store the result (YZQ) into X
LOAD A   ; LOAD A into the accumulator.
MULT A   ; Multiply the accumulator by A (A*A = A^2)
ADD X    ; Add the Squared value of A to the Previous
           Result (YZQ + A^2)
STORE X  ; Store the final result into X
```

* Two Address Machine

```
LOAD X, Y  ; LOAD Y into register X
MUL X, Z   ; Multiply register X by Z (X=X*Z)
MUL X, G   ; Multiply register X by G (X=X*G)
LOAD Y, A  ; LOAD A into register Y
MULT Y, A  ; Multiply register Y by A (Y=Y*A=A^2)
ADD X, Y   ; Add register Y to X (X=X+Y=YZG+A^2)
```

* Three Address Machine :-

```
MULT X, Y, Z ; Multiply Y by Z & Store the result in
               X (X=Y*Z)
MUL X, X, G  ; Multiply X by G & Store the result in
               X (X=X*G=YZG)
MUL Y, A, A  ; Multiply A by A & Store the result
               in Y (Y=A*A=A^2)
ADD X, X, Y  ; Add Y to X & Store the result in
               X (X=X+Y=YZG+A^2)
```

Q.12) What is locality of reference? Explain in details.

Ans=> Locality of reference is a fundamental principle in computer architecture and operating systems, describing the tendency of programs to access certain memory locations more frequently than others over a period of time. It refers to the observation that programs often exhibit patterns of memory access that exhibit spatial and temporal locality.

Two types of locality:

Spatial Locality: Spatial locality refers to the tendency of programs to access memory locations that are near each other. When a program accesses a particular memory location, it is likely to access nearby memory locations in the near future. This is because data and instructions tend to be stored in contiguous memory locations and are accessed sequentially during program execution. Spatial locality is exploited by caching mechanisms such as cache memory and TLBs (Translation Lookaside Buffers), which store recently accessed memory locations along with their neighboring locations, anticipating future accesses.

Temporal Locality: Temporal locality refers to the tendency of programs to access the same memory locations repeatedly over a short period of time. When a program accesses a particular memory location, it is likely to access the same location again in the near future. This occurs due to looping constructs, subroutine calls, and repeated data accesses. Temporal locality is exploited by caching mechanisms to keep frequently accessed data and instructions in fast memory buffers such as cache, reducing the need to fetch them from slower main memory repeatedly.

Why Locality of Reference Matters: Locality of reference is a critical concept in computer architecture for several reasons:

Performance Optimization: Exploiting spatial and temporal locality can significantly improve system performance by reducing the latency of memory accesses. Caching mechanisms such as CPU caches and TLBs are designed to leverage locality of reference to speed up memory accesses.

Cache Design: Cache design is heavily influenced by the principle of locality of reference. Cache hierarchies are organized to capture and exploit both spatial and temporal locality effectively, with different cache levels optimized for different access patterns.

Memory Management: Virtual memory systems leverage the principle of locality to optimize memory management. Page replacement algorithms such as LRU (Least Recently Used) exploit temporal locality by favoring pages that have been accessed recently, while spatial locality is used to optimize page placement and prefetching strategies.

Optimizing Algorithms: Knowledge of locality of reference can inform algorithm design and optimization strategies, enabling developers to design more efficient algorithms that minimize memory access times and improve overall system performance.

Q.13) What are the advantages and disadvantages of segmented and paged memories?

Ans=> Segmented and paged memories are both techniques used in memory management to provide virtual memory and improve system performance. Each approach has its own set of advantages and disadvantages:

Segmented Memory:

Advantages:

Protection: Segmented memory allows for memory protection by assigning different access permissions (such as read-only, read-write, execute-only) to each segment. This helps prevent unauthorized access to memory regions and enhances system security.

Flexibility: Segmented memory provides flexibility in memory allocation by allowing each segment to grow or shrink dynamically as needed. This allows for efficient memory management in variable-size data structures and programs.

Logical Organization: Segmented memory reflects the logical organization of programs, making it easier to manage and understand memory layouts. Each segment can represent a different aspect of program functionality (e.g., code segment, data segment, stack segment).

Disadvantages:

Fragmentation: Segmented memory can suffer from fragmentation, both internal and external. Internal fragmentation occurs when segments are allocated more memory than they actually need, leading to wasted space within segments. External fragmentation occurs when free memory is fragmented into small, non-contiguous blocks, making it difficult to allocate large segments.

Complexity: Segmented memory management is more complex compared to other memory management techniques, such as paging. It requires additional overhead for managing segment tables, segment descriptors, and segment registers.

Limited Address Space: Segmented memory may impose limitations on the maximum addressable memory space due to the finite number of segment descriptors or segment registers available in the system architecture.

Paged Memory:

Advantages:

Reduced Fragmentation: Paged memory reduces external fragmentation by dividing physical memory into fixed-size blocks (pages) and allowing these pages to be allocated and managed independently. This helps optimize memory utilization and reduce wasted space.

Simpler Management: Paged memory management is simpler compared to segmented memory, as it involves managing a single-level page table that maps virtual pages to physical pages. This simplifies memory management operations such as allocation, deallocation, and address translation.

Large Address Spaces: Paged memory supports large address spaces by allowing virtual memory to be larger than physical memory. This enables systems to run larger programs and handle more extensive datasets without running out of memory.

Disadvantages:

Fragmentation: Although paged memory reduces external fragmentation, it can still suffer from internal fragmentation if pages are not fully utilized. This occurs when the allocated memory space within a page is smaller than the page size, leading to wasted space.

Page Table Overhead: Paged memory requires maintaining a page table to map virtual addresses to physical addresses. As the size of the virtual address space increases, the size of the page table also increases, leading to additional memory overhead and potential performance implications.

Page Faults: Paged memory systems may experience page faults when a requested page is not present in physical memory and must be fetched from secondary storage. Page faults incur additional overhead and can lead to performance degradation, particularly if the page replacement algorithm is inefficient.

Q.14) Write a short note on interleaved memory and state its necessity.

Ans=> Interleaved memory is a memory organization technique used in computer systems to improve memory access performance by distributing memory addresses across multiple memory modules in a systematic manner. Here's a brief explanation along with its necessity:

Interleaved Memory:

In interleaved memory, the memory address space is divided into consecutive address ranges called interleaves. These interleaves are distributed across multiple memory modules, such as RAM chips or banks, in a round-robin fashion. Each memory module is responsible for storing data corresponding to a specific interleave.

For example, if we have two memory modules and a memory address space of 8 bytes, interleaved memory might divide the address space into two interleaves: interleaves 0 and 1. Memory module 1 would be responsible for storing data for interleaves 0, while memory module 2 would be responsible for interleaves 1.

When the CPU issues memory access requests, the memory controller interleaves the requests across the memory modules. This means that consecutive memory accesses are distributed evenly across the memory modules, reducing contention and improving memory access concurrency.

Necessity of Interleaved Memory:

Improved Memory Bandwidth: Interleaved memory improves memory bandwidth by allowing multiple memory modules to work in parallel. This reduces contention and bottlenecks, especially in systems with high memory access demands, such as multiprocessor systems or systems running memory-intensive applications.

Reduced Access Latency: By distributing memory accesses across multiple memory modules, interleaved memory reduces the average access latency experienced by the CPU. This is particularly important for systems where memory access time is a critical factor in overall system performance.

Balanced Access Patterns: Interleaved memory helps balance memory access patterns by spreading memory accesses evenly across memory modules. This prevents hotspots and uneven utilization of memory resources, ensuring efficient use of available memory bandwidth.

Scalability: Interleaved memory facilitates system scalability by allowing additional memory modules to be added to the system without significantly impacting memory access performance. As the system grows, interleaved memory can accommodate the increased memory capacity and access demands more effectively.

Q.15) What is memory mapping? Discuss in details.

Ans=> Memory mapping is a fundamental concept in computer systems, referring to the process of associating logical addresses generated by the CPU with physical addresses in memory. It enables the operating system to manage memory efficiently and provides programs with a consistent and abstracted view of memory, regardless of the underlying hardware architecture.

1. Address Spaces:

Memory mapping operates within the context of address spaces. An address space is the range of memory addresses that a CPU or a process can access.

CPUs generate logical addresses, which are also referred to as virtual addresses. These addresses are used by programs during execution.

2. Physical Memory:

Physical memory refers to the actual hardware components where data and instructions are stored. This includes RAM (Random Access Memory) modules and other storage devices.

Physical memory is divided into memory cells, each with a unique address. These addresses are used by the hardware to access data stored in memory.

3. Mapping Process:

Memory mapping involves associating logical addresses generated by the CPU with physical addresses in memory.

The mapping process is managed by the operating system, which maintains data structures called page tables or translation tables to store the mappings.

4. Types of Memory Mapping:

a. Static Memory Mapping: - In static memory mapping, the mapping between logical addresses and physical addresses is fixed and predetermined. - Static memory mapping is commonly used for system memory, where specific memory regions are reserved for different purposes such as code, data, and stack segments.

b. Dynamic Memory Mapping: - In dynamic memory mapping, the mapping between logical addresses and physical addresses can change dynamically during program execution. - Dynamic memory mapping is used for memory-mapped I/O devices, where logical addresses are mapped to control and data registers of I/O devices.

5. Virtual Memory:

Virtual memory is an abstraction provided by the operating system that extends the available address space beyond physical memory.

Virtual memory uses techniques such as demand paging and page replacement to manage memory efficiently and transparently to programs.

6. Importance of Memory Mapping:

Memory mapping is essential for efficient memory management and resource allocation in computer systems.

It provides programs with a uniform and abstracted view of memory, simplifying program development and portability across different hardware architectures.

Memory mapping enables features such as virtual memory, memory protection, and multitasking, which are essential for modern operating systems and applications.