

- Q] Explain characteristics & performance parameters of memory devices.

→ Memory devices has several characteristics & performance parameters that are crucial for their operation. Here are some:

- Capacity:

It refers to the amount of data it can store. It is typically measured in bytes, & determines how much information the device can hold.

- Access Time:

It is the time taken by the memory device to retrieve or store data. It is usually measured in nanoseconds.

- Latency:

It is the delay between a request for data & the start of data retrieval or storage. It includes factors like access time, data transfer time & any other overhead involved.

- Bandwidth:

It refers to the rate at which data can be transferred to and from the memory device. It measured in bytes per seconds.

- Volatility:

It describes whether the memory device can retain its stored data when the power supply is disconnected. Non-volatile memory retains data even without power, while volatile memory loses data when powered off.

### - Reliability:

It indicates the stability & dependability of the memory device. It includes factors like error rates, data integrity, & mean time between failure.

- Cost:

It includes the manufacturing, maintenance & operational expenses associated with the device. Different memory technologies vary in cost, & the choice depends on factors such as performance needs, capacity requirements, & budget constraints.

- Q] Write Difference between SRAM & DRAM.

SRAM

- Static Random Access Memory.

- It consists of D flip flop

- It having complicated design

- It is faster than DRAM

- No refreshing is required.

- They are expensive

DRAM

- Dynamic Random Access Memory.

- It consists of capacitor.

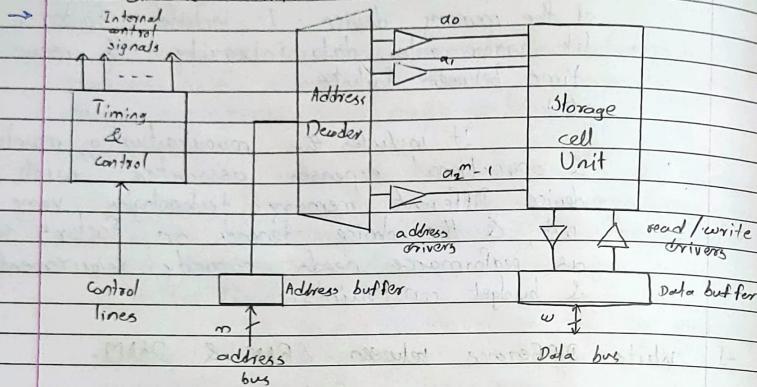
- It having simple design

- It is slower than SRAM

- Refreshing is required.

- They are cheaper.

3) With neat diagram explain working of one dimensional RAM.



One dimensional RAM is often represented as an array of memory cells. Each memory cell stores a binary value & it is uniquely addressed by its location or index within the array.

- Array Structure:

The RAM is organized as a linear array of memory cells. Each cell is assigned a unique address, typically represented by binary numbers.

- Read operation:

When a read operation is registered the CPU sends the address of the desired memory cells to the RAM. RAM uses this address to locate the corresponding memory cell & retrieves the stored value.

- Write operations:

During write operation the CPU provides the address of the memory cell & the new data to be stored. The RAM locates the specified memory cell & updates its contents with the new data.

- Address Decoding:

It is performed to select the appropriate memory cell based on the provided address. This process typically involves decoder circuits that activate the specific memory cell based on the address lines received.

- Data Bus:

The RAM is connected to the data bus which allows the transfer of data between the CPU & the memory cells. The data bus is used for both read & write operations, enabling the CPU to retrieve data from or store data into the RAM.

Q] Explain Memory allocation techniques:

→ Memory allocation techniques are strategies used by computer systems to allocate & manage memory resources efficiently. Here's a brief explanation of common memory allocation techniques:

i) Contiguous Memory Allocation:

Memory is divided into fixed size partitions, and each process is assigned a contiguous block of memory. It is simple to implement but can lead to fragmentation.

### 2] Non-contiguous memory:

Memory is allocated in a non-contiguous manner allowing different parts of a process to be scattered throughout the memory. It helps overcome fragmentation but requires additional management overhead.

### 3] Paging:

Memory is divided into fixed sized blocks called pages, & processes are divided into frames.

### 4] Segmentation:

Processes are divided into variable sized segments such as code, data, & stack segments. Each segment is allocated separately, & they can be located anywhere in memory.

### 5] Dynamic Memory:

Memory is allocated and released during runtime as needed. It enables programs to request and release memory dynamically adapting to changing memory requirements.

## 5] Explain page replacement policies.

→ Page replacement policies are used in virtual memory system to decide which pages should be evicted from physical memory when a new page needs to be loaded. These policies aim to optimize memory utilization & minimize page faults. Here's brief explanation of common page replacement policies.

### Optimal:

The optimal page replacement policy also known as the MIN policy selects the page that will not be used for the longest period of time in the future.

### LRU:

This optimal policy selects the page that has not been accessed for the longest time to be replaced. It assumes that pages that have not been accessed recently are less likely to be used in the future.

### FIFO:

This policy evicts the page that has been in physical memory the longest. It follows a queue like structure, where the oldest page is selected for replacement.

~~These page replacement policies trade off between simplicity, overhead & performance. Each policy has its strengths & weaknesses, & the choice of policy depends on the specific characteristics of the workload & the system requirements.~~

### Q] Solve the given problem with Optimal, LRU & FIFO page replacement policy.

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7

Optimal

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	7
7	7	7	2	2	2	2	2	2	7	2	2	2	2	2	7	7
0	0	0	0	0	4	4	6	0	0	0	0	0	0	0	0	0
1	1	1	3	3	3	3	3	3	3	3	3	1	1	1	1	1
9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9

Page Page Page Page Page  
bit bit bit hit hit hit hit

IRU:

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	7
7	7	7	2	2	2	2	4	4	4	0	0	0	1	1	1	7
0	0	0	0	0	0	0	3	3	3	3	3	3	0	0	0	0
1	1	1	3	3	3	2	2	2	2	2	2	2	2	2	2	2
9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9

Page Page Page  
hit hit hit

IFO:

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	7
7	7	7	2	2	2	2	4	4	4	0	0	0	0	0	0	7
0	0	0	0	3	3	3	2	2	2	2	2	1	1	1	1	1
1	1	1	1	0	0	0	3	3	3	3	2	2	2	2	2	2
9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9

Page Page Page  
hit hit hit

iii) What is address translation? Explain dynamic address translation scheme.

Address translation is a process used in computer systems to convert logical addresses into physical addresses. This translation is necessary in systems with virtual memory, where the logical address space is larger than the physical memory available.

Dynamic address translation is a scheme that maps logical addresses to physical addresses during runtime. It involves the use of a hardware component called the memory management unit.

i) Page Table:

It is a data structure that maps logical page numbers to physical page frames.

ii) MMU:

It is a hardware component responsible for performing the address translation.

iii) Translation Lookaside Buffer:

It is a fast cache that stores frequently accessed page table entries.

iv) Address Translation process:

It is send by the CPU when the CPU generates a logical address. The MMU extracts the page number & offset from the logical address.

v) Page Table Look-up:

The MMU performs a look up in the page table using the page number to find the corresponding physical page frame number.

M T W T F S

Page No. : YOUVA

Date:

Dynamic address translation allows for efficient utilization of physical memory by allowing logical addresses to be mapped to different physical locations dynamically.

Q) What is address mapping? Explain set associative mapping.

→ Address mapping is the process of mapping memory addresses from a higher level representation to a lower level representation in computer system. It involves techniques that efficiently manage the mapping of addresses to physical memory locations.

Set associative mapping is a mapping technique used in cache memory systems. It combines the advantages of direct mapping & fully associative mapping.

- Cache Structure:

The cache memory is organized into a set of cache lines or slots, each of which can store a block of data.

- Address Structure:

Memory addresses are divided into three components: the tag, the index, & the offset. The tag bits uniquely identify a memory block. The index bits determine the set in the cache. & the offset bits specify the position within the cache lines.

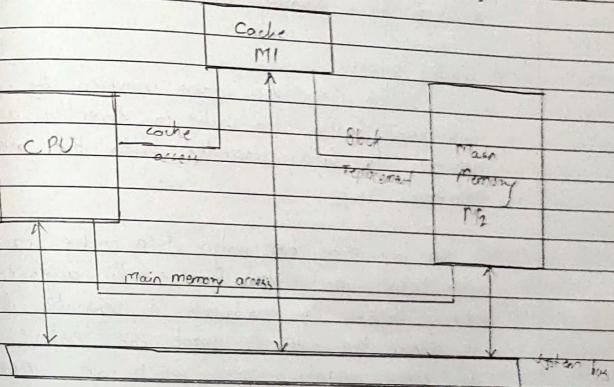
- Cache line selection:

In set associative mapping, multiple cache lines in the selected set are checked simultaneously. The tag bits of memory address are compared with the stored tags in the cache lines ~~are~~ of the set.

- Replacement Policy:

If a cache miss occurs & there is no empty cache line in the set a replacement policy is used to determine which cache line to evict.

Q) With neat diagram explain look aside design of cache.



The look aside design of a cache involves the use of a separate cache directory or lookup structure called the look aside buffer or cache

directory to accelerate cache accesses. It keeps the track of recently accessed memory blocks & their corresponding locations in the cache.

#### - Cache Structure:

The cache itself is organized in a conventional manner, consisting of cache lines or slots that stores blocks of data fetched from memory.

#### - Memory access process:

When memory access request is made, the address is first checked in the Look Aside Buffer to determine if it has been recently accessed & is present in the cache.

#### - Cache access:

With the cache index provided by the look aside buffer the cache is directly accessed to retrieve the data associated with the memory address.

The look aside ~~bad~~ design of a cache provides a faster lookup mechanism for recently accessed memory blocks. By maintaining a separate directory, it reduces the need to search the entire cache for every memory access which can improve cache access time.

Q) Differentiate between tightly coupled & loosely coupled.

#### → Loosely coupled

- There is distributed memory

- It has low data rate.

- They are cheaper.

- Memory conflicts don't take place.

- It has low degree interaction between tasks.

- They are distributed computing systems.

#### 2] Explain Computer Module:

A computer module consisting of a processor, local memory, local input / output devices and a channel with an arbiter switch is a common configuration.

#### - Processor:

It referred to as the central processing unit is the core component responsible for executing instructions & performing computations. It fetches instructions from memory, decodes them & executes the corresponding operations. The processor interacts with other

components through data & control signals.

#### - Local Memory:

The local memory typically referred to as cache memory resides closer to the processor & provides fast access to frequently used data & instructions. It acts as a buffer between the processor & the main memory.

#### - Local I/O devices:

It facilitates communication between the computer system & external devices. They enable data exchange with peripherals such as keyboards, mice, displays, printers or storage devices.

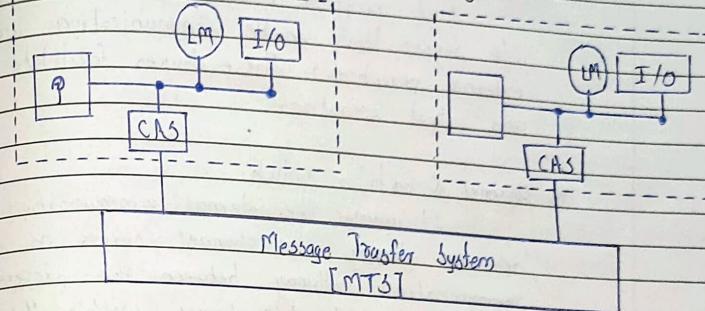
#### - Channel:

It is a communication pathway or medium that enables the transfer of data between different modules or components within the computer system. It facilitates the reliable transmission of data, commands & signals between the sender & the receiver.

#### - Arbiter Switch:

It manages access to shared resources or communication channels among various modules. It determines which module is granted access to the shared resource at any given time, resolving conflicts or contention for resources.

Q) Explain CM\* architecture with diagram.



In a Computer Module the components such as the processor, local memory, local I/O devices, channel and arbiter switch are interconnected and if the multiple computer modules are connected & using MTS [Message Transfer System] then that architecture called as computer module \* architecture.

#### i) Processor:

It is responsible for executing instructions & performing computations within that module. The processors handle tasks specific to their respective modules and participate in communication through the message transfer system.

#### ii) Local Memory:

The local memory in each module provides fast access to frequently used data and instructions within that specific module.

### 3] Local I/O Devices:

Each computer module has its own local I/O devices that enable communication with external peripherals. These devices facilitate input and output operations.

### 4] Channel & Arbiter Switch:

It manages the internal communication & resource sharing. The channel serves as a communication pathway between the processor, local memory & I/O devices within the module.

### 5] Message Transfer System:

The central message transfer system connects multiple computer modules together. It provides the means for inter-module communication, allowing data, commands, & signals to be transferred between different modules.

### 4] Write a Note on role of CAS.

#### → Channel and Arbiter Switch [CAS]:

It is a crucial component in computer architecture that facilitates communication and resource sharing among different modules or components within a computer system. It comprises the channel which serves as a communication pathway, and the arbiter switch, which manages access to shared resources.

#### i) Channel:

It provides a medium for the transfer of data, commands & signals between different modules or components within the

computer system. It ensures reliable and efficient communication by adhering to specific protocols and standards.

### 2] Arbiter Switch:

It is responsible for managing access to shared resources such as the channel itself or other system resources, among multiple modules that contend for access simultaneously.

### 6] Write a note on MTS.

#### → Message Transfer System [MTS]:

It is the central component that connects multiple computer modules within a computer system. It provides a framework for inter-module communication, allowing data, commands & signals to be transferred between different modules.

The central message transfer system handles the routing of messages between modules, manages data synchronization, and ensures efficient utilization of the overall system resources. The communication between modules is facilitated by the channels and arbiter switches within each module as well as the message transfer system.

### 6] Explain working of communication memory with neat diagram.

→ The working of communication memory involves the interaction of various components such as processor input ports, common memory containing communication ports, & process

input ports.

### 1 Processor input ports:

Each processor in the multiprocessor system has input ports that serve as communication interfaces. These input ports allow the processor to send & receive messages or data to and from other processes or components in the system. The processor's input ports handle the sending & receiving of communication requests.

### 2 Common Memory with communication ports:

It contains communication ports that facilitate communication between processes. These communication ports act as shared memory locations or buffer where processes can exchange data or messages.

### Process Input Ports:

Each process running on a processor also has input ports dedicated to communication. These input ports allow the process to interact with other processes by sending or receiving messages. When a process wants to send message, it writes the message into its output port. The target process can then read the message from its input port to receive & process the communication.

