COA-Module 4 Notes

Q1)What are the characteristics of Memory?

A:

1. Location:

CPU: Memory within the central processing unit.

Internal: Main memory or cache, directly accessible by the CPU.

External: Storage outside the main system, such as disks and tapes, used for backup and additional storage.

2. Capacity:

Word Size: The natural unit of data organization (e.g., 8-bit, 16-bit).

Number of Words or Bytes: Total number of words or bytes stored in the memory.

3. Unit of transfer:

Internal: Governed by data bus width, typically a word.

External: Generally a larger block, such as a disk sector.

Addressable Unit: The smallest unit uniquely addressable in the memory system, often a byte.

4. Access method:

Sequential: Data is accessed in a set sequence, starting from the beginning (e.g., tape storage). Access time depends on the location of data and the previous location

Direct: Individual blocks have unique addresses, and data access can be achieved by jumping to a specific block, followed by a sequential search (e.g., disk storage).Access is by jumping to vicinity plus sequential search. Access time depends on location and previous location

Random: Any location can be accessed independently and immediately (e.g., RAM).Access time is independent of location or previous access.

Associative: Data is accessed by comparing contents rather than by a specific address (e.g., cache memory). Access time is independent of location or previous access

5. Performance(SRAM,DRAM):

Access Time: Time between a request and data availability at the required location.

Memory Cycle Time: Minimum time between successive read requests.

Transfer Rate: Speed at which data can be transferred to/from memory.

6. Physical type:

Semiconductor: RAM types, such as SRAM and DRAM.

Magnetic: Disk drives and magnetic tapes.

Optical: Media such as CDs and DVDs.

7. Physical characteristics:

Decay: Rate at which stored charge decays, affecting data stability.

Volatility: Determines whether memory retains data without power (e.g., RAM is volatile, ROM is non-volatile).

Erasable: Some types can be erased and rewritten (e.g., Flash memory).

Power Consumption: Amount of power needed to maintain data, particularly in volatile memory.

8. Organisation- Direct Mapping, Associative Mapping:

Direct Mapping: A specific block in main memory maps directly to a specific line in cache.

Associative Mapping: A memory block can be loaded into any line of cache; often used in fully associative or set-associative caches for flexibility.

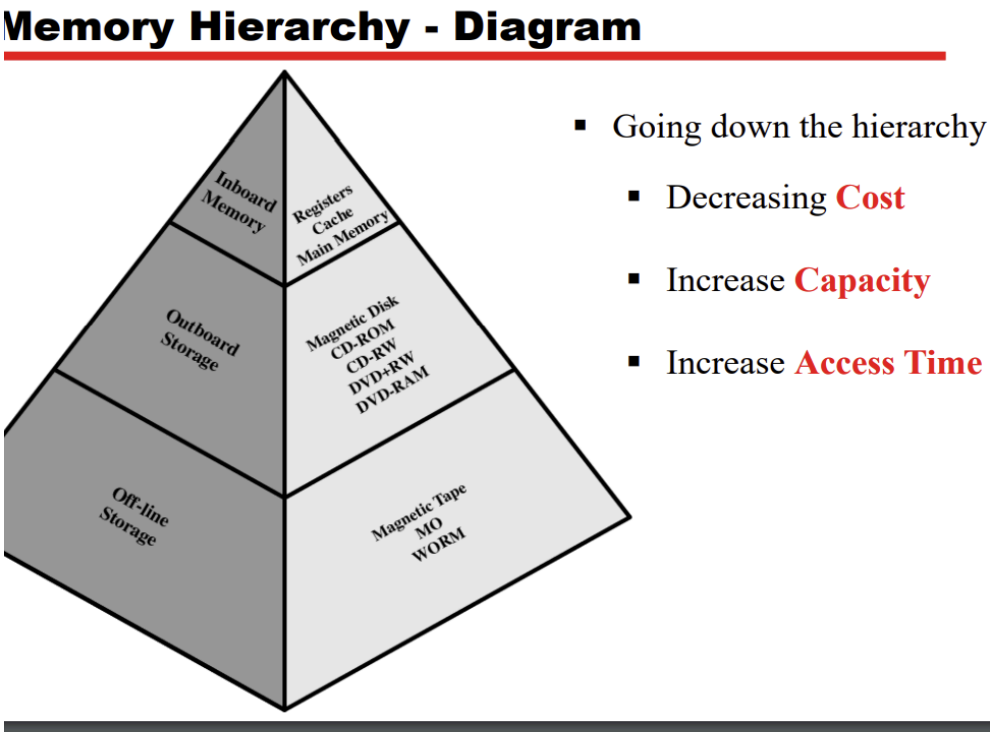
Q2)Explain Memory Hierarchy

A:

The memory hierarchy is designed to optimize the trade-offs between speed, cost, and storage capacity by layering various types of memory. The hierarchy includes the following main levels:

1. **Registers**:
    - **Description**: Registers are the smallest and fastest type of memory, located within the CPU.
    - **Function**: They hold data temporarily for immediate operations by the CPU, such as instruction operands or calculation results.
    - **Characteristics**: High-speed access but very limited capacity, typically storing only a few bytes of data.
2. **Internal or Main Memory (RAM)**:
    - **Description**: This layer, also called main memory, is made up of RAM, which may include several cache levels.
    - **Cache Memory**:
        - **Levels**: The cache is often split into multiple levels:
            - **L1 Cache**: The fastest and closest to the CPU core, with the smallest size.
            - **L2 Cache**: Slightly larger and slower than L1, but still within the CPU.
            - **L3 Cache**: The largest cache level, slower than L1 and L2, sometimes shared across CPU cores.
        - **Usage**: Cache stores frequently accessed data, reducing the time the CPU spends waiting for data from main memory.
        - **Key Operation**: Cache checks if the required data is present (a “hit”); if not, it fetches from main memory (a “miss”).
    - **Main Memory (DRAM)**:
        - **Description**: Main memory has a larger capacity than cache and holds data that applications are currently using.
        - **Characteristics**: Slower than cache but faster than secondary storage, and much larger in size than cache memory.

3. **External Memory (Secondary Storage)**:
    - **Description**: External memory, such as hard drives (HDDs) and solid-state drives (SSDs), offers larger storage capacity at lower cost compared to internal memory.
    - **Function**: Stores data and programs that aren't in active use, serving as a backup for main memory.
    - **Characteristics**: Non-volatile (retains data when power is off) but significantly slower than both cache and main memory.



Q3)What is RAM?

A:

RAM (Random Access Memory) is a type of volatile memory that allows data to be read from or written to any location directly (randomly), without needing to go through other memory locations sequentially. It's the primary memory in a computer system, playing a critical role in the performance of applications and processes.

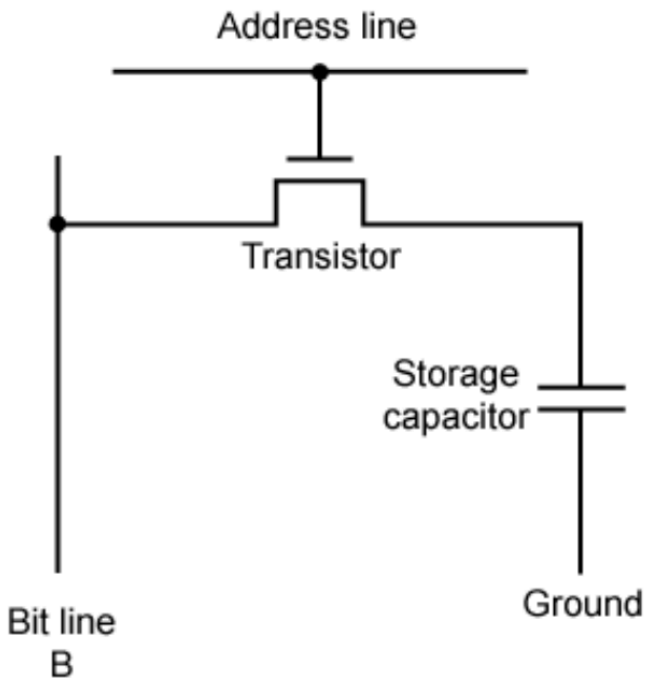
- **Random Access**: Allows data to be accessed directly from any location without sequential order.
- **Read/Write Memory**: Supports both reading and writing, enabling temporary storage of data and instructions for active processes.
- **Volatile**: Requires power to retain data; all information is lost when the system shuts down.
- **Temporary Storage**: Acts as a workspace for the CPU, holding data and applications in use for quick access.
- **Types**:
    - **Static RAM (SRAM)**: Faster, used in cache memory.
    - **Dynamic RAM (DRAM)**: Slower, forms the main system memory.

Q4)What is Dynamic RAM?

A:

- **Bits Stored as Charge in Capacitors**: DRAM stores each bit as a small charge in a capacitor.
- **Charge Leakage**: The charge in capacitors leaks over time, which means data cannot be stored permanently.
- **Need for Refreshing**: Even when powered, DRAM requires **periodic refreshing** to maintain data integrity due to charge leakage.
- **Simpler Construction**: DRAM cells are simpler than SRAM cells, making it **less expensive** to manufacture.
- **Refresh Circuits Required**: Additional **refresh circuits** are necessary to recharge the capacitors regularly, contributing to slower performance.
- **Speed**: DRAM is generally slower than SRAM because of the time required for refreshing and recharging.
- **Main Memory**: DRAM is widely used as **main memory** in computers due to its high density and cost-effectiveness.
- **Analog Nature**: DRAM is essentially analog; the **level of charge** in each capacitor determines the value (0 or 1) of the stored bit.

Dynamic RAM Structure



Q5)What is Static RAM?

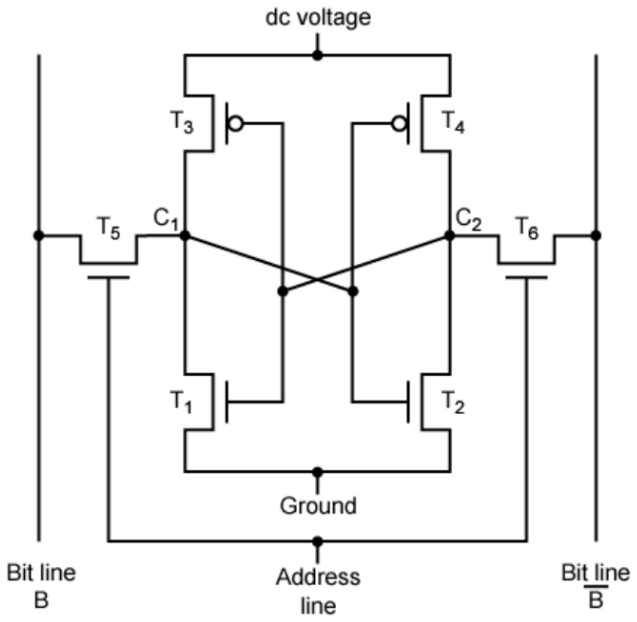
A:

- **Bits Stored as On/Off Switches**: SRAM stores each bit using a set of transistors configured as a flip-flop, which holds the data in a stable state without needing refreshing.
- **No Charge Leakage**: Unlike DRAM, SRAM does not rely on capacitors, so there is no charge to leak, and **data remains stable** as long as power is supplied.
- **No Refreshing Required**: Since SRAM does not lose charge, it does not need refreshing, which allows for **faster access times** compared to DRAM.
- **More Complex Construction**: SRAM cells use more transistors per bit (typically 4-6), making it more **complex and physically larger** than DRAM.

- **Higher Cost**: Due to its complex construction, SRAM is **more expensive** than DRAM per bit.
- **Faster Performance**: SRAM's lack of refresh cycles makes it significantly faster, which is why it is often used for **cache memory** close to the CPU.
- **Digital Nature**: Unlike the analog nature of DRAM, SRAM operates digitally, with each bit represented by a stable, discrete on/off state (flip-flop).

In summary, **SRAM** is a fast, stable type of memory often used for cache due to its **non-volatile behavior during power**, **higher speed**, and **reliability**, though it comes at a higher cost and with less density than DRAM.

Static RAM Structure



Q7)Differentiate between SRAM and DRAM.

A:

Feature	SRAM (Static RAM)	DRAM (Dynamic RAM)
Volatile	Yes	Yes
Power needed to preserve data	Requires constant power to retain data	Requires constant power to retain data
Dynamic cell	No, uses flip-flops for storing data	Yes, uses capacitors to store data
Simpler to build	More complex, requires more transistors	Simpler, uses fewer components
Smaller	Larger due to more complex structure	Smaller, more compact
Density	Less dense (fewer bits per unit area)	More dense (more bits per unit area)
Cost	More expensive	Less expensive
Needs refresh	No refresh needed	Requires periodic refreshing of data
Larger memory units	Typically used for smaller, faster caches	Used for larger memory systems (e.g., RAM)
Speed	Faster (due to lack of refresh cycles)	Slower (due to refresh cycles and simpler design)
Used in	Primarily used in cache memory	Used in main memory (system RAM)

Q8)What is ROM?

A:

ROM (Read-Only Memory) is a type of non-volatile memory that is primarily used for storing firmware or software that doesn't change frequently. Unlike RAM, which is volatile (loses data when power is turned off), ROM retains its contents even when power is removed. ROM typically stores the boot-up instructions and firmware for hardware devices, such as the computer's BIOS, embedded systems, or other permanent software that doesn't need to be modified often.

Characteristics of ROM:

- **Non-volatile**: Data is retained even when the device is powered off.

- **Read-only**: The data stored in ROM can be read, but typically cannot be modified (depending on the type).
- **Used for firmware**: Typically stores low-level system instructions, like the BIOS in a computer or firmware in devices like printers and routers.
- **Durable and reliable**: Since the data is etched permanently (in the case of traditional ROM), it is highly resistant to data corruption.

Q9)What is PROM?

A:

PROM (Programmable Read-Only Memory) is a type of non-volatile memory that is programmed during the manufacturing process, allowing it to retain data even when the power is turned off. PROM is often used to store firmware or other software that doesn't need to change once it's been written. Here's a more detailed explanation of the points you mentioned:

- **Written during manufacture**: PROM is programmed during its manufacture and cannot be altered afterward.
- **Programmable ("once")**: Data is written once and cannot be modified after programming.
- **Small amount of data to be written**: Typically used for storing small, critical data like firmware.
- **Less expensive**: PROM is cost-effective because it only needs a one-time programming process.
- **Non-volatile, written only once**: Retains data even without power and cannot be rewritten once programmed.
- **Writing performed electrically at the time of chip fabrication**: Data is written to the chip using electrical signals during manufacturing, with no possibility of modification afterward.

Q10) What is EPROM?

A:

- **Erasable Programmable (EPROM) – Erased by UV**: EPROM can be erased by exposing it to ultraviolet (UV) light, which resets the stored data.
- **Read and written electrically**: Data can be both read and written to an EPROM chip using electrical signals.
- **All storage cells should be erased electrically to initial state by exposure to UV radiation**: To erase EPROM, the chip must be exposed to UV light, which clears all the data, allowing it to be reprogrammed.
- **Can be altered multiple times and holds data virtually indefinitely**: EPROM can be rewritten many times and the data remains intact for a long period if not exposed to UV light.
- **More expensive than PROM**: EPROM is more expensive than PROM due to the additional equipment required for erasure and reprogramming.

Q11) What is EEPROM?

A:

- **Can be written anytime without erasing prior contents**: EEPROM allows individual bytes to be rewritten without affecting the other stored data.
- **Write operation takes longer than read**: Writing data to EEPROM takes more time than reading it, due to the more complex process involved.
- **More expensive than EPROM, less dense**: EEPROM is more costly than EPROM because it offers more flexibility but at the expense of lower storage density.

Q12)What are the types of ROM?

A:

1. **Programmable Read-Only Memory (PROM)**:
    - **Empty of data when manufactured**: PROM comes without any data preloaded and can be programmed by the user once.
    - **May be permanently programmed by the user**: After programming, the data is permanently written and cannot be changed.
2. **Erasable Programmable Read-Only Memory (EPROM)**:
    - **Can be programmed, erased, and reprogrammed**: EPROM can be written to, erased with UV light, and reprogrammed multiple times.
    - **The EPROM chip has a small window on top allowing it to be erased by shining ultra-violet light on it**: A transparent window allows UV light to erase the stored data for reprogramming.
    - **After reprogramming, the window is covered to prevent new contents from being erased**: Once reprogrammed, the window is covered to prevent accidental erasure.
    - **Access time is around 45–90 nanoseconds**: EPROM chips have relatively fast access speeds, with typical read times between 45 and 90 nanoseconds.
3. **Electrically Erasable Programmable Read-Only Memory (EEPROM)**:
    - **Reprogrammed electrically without using ultraviolet light**: EEPROM can be erased and rewritten electrically, making it more convenient than EPROM.
    - **Must be removed from the computer and placed in a special machine to do this**: Though EEPROM is electrically erasable, it still needs to be physically removed and placed in a programmer to update data.

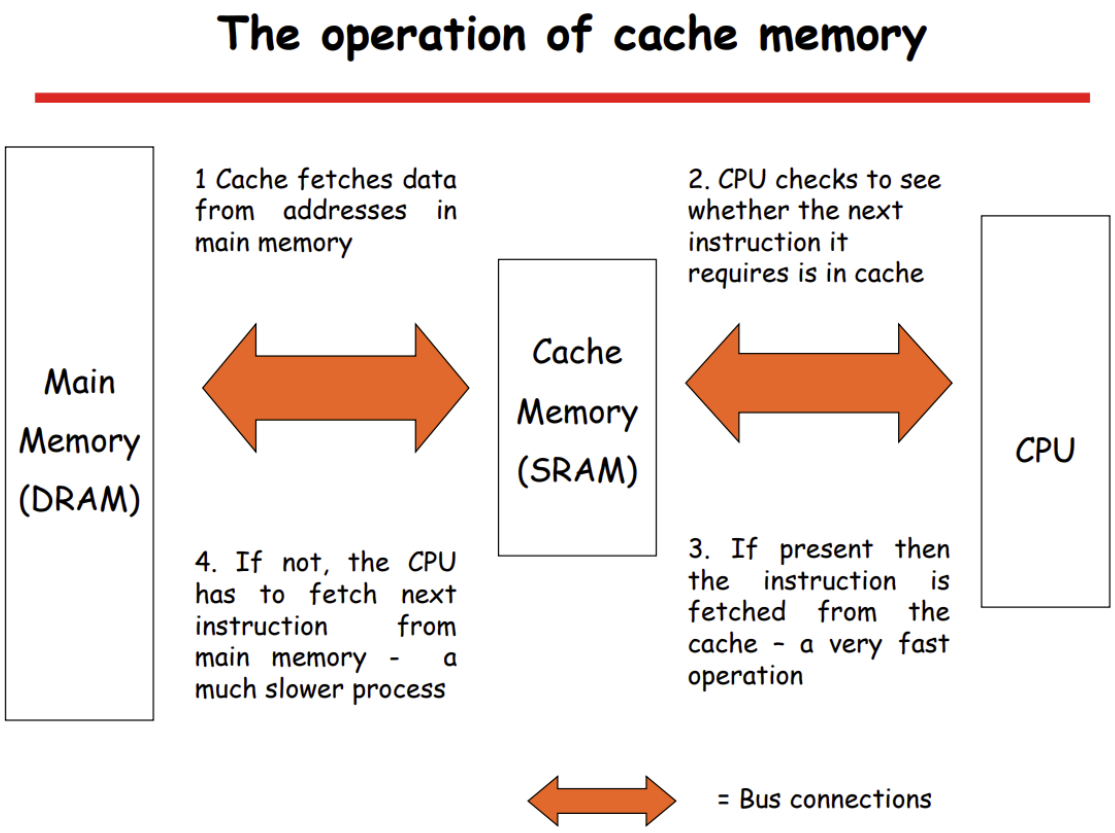
- **Access times between 45 and 200 nanoseconds**: EEPROM has slightly slower access times compared to EPROM, with read times ranging from 45 to 200 nanoseconds.

4. **Flash ROM**:
    - **Similar to EEPROM**: Flash ROM shares many similarities with EEPROM but has more advanced features for efficient reprogramming.
    - **However, can be reprogrammed while still in the computer**: Unlike EEPROM, Flash ROM can be updated without removing it from the device, making it more convenient for regular updates.
    - **Easier to upgrade programs stored in Flash ROM**: Flash ROM is commonly used in devices that require frequent updates, such as firmware upgrades in embedded systems.
    - **Used to store programs in devices e.g., modems**: Flash ROM is used in devices like modems, routers, and smartphones to store the system's firmware.
    - **Access time is around 45–90 nanoseconds**: Flash ROM has fast access times, similar to EPROM, typically between 45 and 90 nanoseconds.
5. **ROM Cartridges**:
    - **Commonly used in game machines**: ROM cartridges are used to store video games and other software in consoles, providing a portable and easily replaceable storage medium.
    - **Prevents software from being easily copied**: ROM cartridges offer some level of protection against software piracy because the content is usually read-only and not easily copied or modified.

Q13)What is Cache? Show the operation of cache memory.

A:

Cache is a small amount of fast memory- which sits between the main memory and CPU. It may be located on CPU chip or module. It is used to store data required in regular cases.

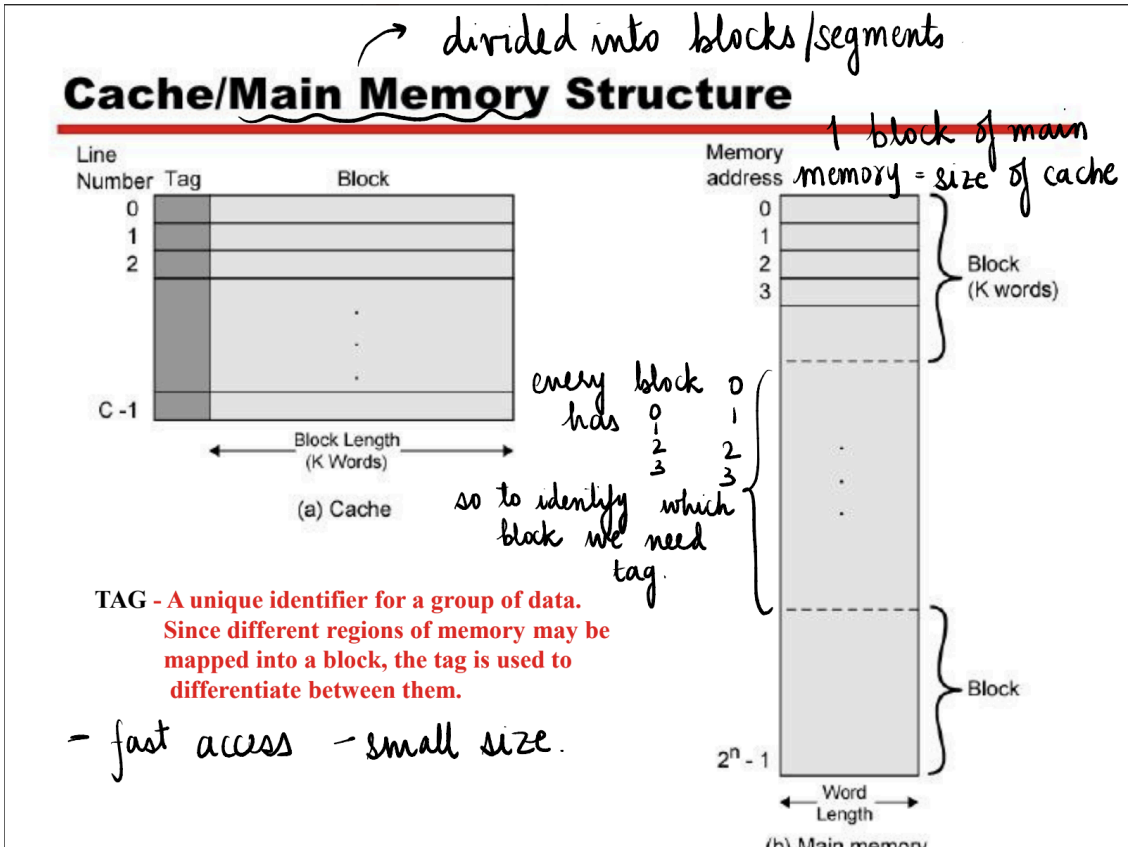


Q14)Give an overview of cache operation.

A:

- CPU requests contents of memory location.
- Check cache for this data.
- If present, get from cache (fast).
- If not present, read required block from main memory to cache.

- Then deliver from cache to CPU.
- Cache includes tags to identify which block of main memory is in each cache slot.



Q15)What is Tag?

A:

TAG - A unique identifier for a group of data. Since different regions of memory may be mapped into a block, the tag is used to differentiate between them.

Q16)Explain the differences between L1, L2, and L3 cache in terms of speed, size, and location.

A:

- **L1 Cache**: This is the smallest and fastest cache, located directly on the CPU chip. It has a very small capacity (usually 16KB to 128KB) and is used for quick access to frequently used data and instructions.
- **L2 Cache**: Larger than L1, with more capacity (typically 128KB to several MB), and located either on the CPU chip or on a separate chip near the CPU. It is slower than L1 but still faster than RAM and helps in improving CPU performance by storing data that L1 might not have.
- **L3 Cache**: This is the largest and slowest among the three, often shared by multiple CPU cores. It can be several MB to tens of MB in size. It helps improve the performance of both L1 and L2 by storing a larger pool of data, with access speeds roughly twice that of RAM.

Q17)Explain the key factors involved in cache design.

A:

- **Size**: The size of the cache determines how much data can be stored, affecting performance and cost; larger caches generally improve speed but increase cost.
- **Mapping Function**: The mapping function defines how data from main memory is assigned to cache locations, impacting cache efficiency and access time.
- **Replacement Algorithm**: This algorithm decides which data to evict when the cache is full, with common strategies like LRU (Least Recently Used) or FIFO (First In First Out).
- **Write Policy**: Determines how data is written to cache and main memory, with strategies like write-through (immediate write to memory) and write-back (write only when data is evicted).
- **Block Size**: Refers to the amount of data fetched from memory in a single cache line, affecting the trade-off between data locality and cache efficiency.

- **Number of Caches**: This refers to how many levels of cache exist (L1, L2, L3) and their respective sizes, impacting overall system performance.
- **Cost**: More cache increases the overall cost of the system due to the need for more physical memory and more complex cache management.
- **Speed**: Larger caches can improve system speed by reducing the need to access slower main memory, but there's a point where adding more cache yields diminishing returns.
- **Checking Cache for Data Takes Time**: While caches improve speed, checking multiple levels of cache for data introduces overhead, particularly in systems with deep cache hierarchies.

Q18)What does Mapping Technique mean?

A:

Mapping Technique means how data from main memory is stored in cache.

Q19)What is: i)Tag ii)Word iii)Line

A:

1. **Tag Field**: The tag field stores the higher-order bits of the memory address and is used to check if the requested data is in the cache.
2. **Line Field**: The line field selects which cache line the data might be stored in by using the middle portion of the memory address.
3. **Word Field**: The word field selects the specific word within the cache line when multiple words are stored in each line.

Q20)What is direct mapping?

A:

Direct Mapping is a cache mapping technique where each block of main memory is mapped to exactly one cache line. In other words, for a given memory address, there is only one specific location in the cache where the data can be stored, determined by a simple mapping function.

Key Points:

- Each block of memory is assigned to a specific cache line based on the address.
- The memory address is divided into three parts: **Tag**, **Line (or Index)**, and **Block Offset**.
- The **Line field** determines which cache line to look in, and the **Tag field** is compared with the stored tag in that cache line to check for a hit or miss.

Advantages of Direct-mapping

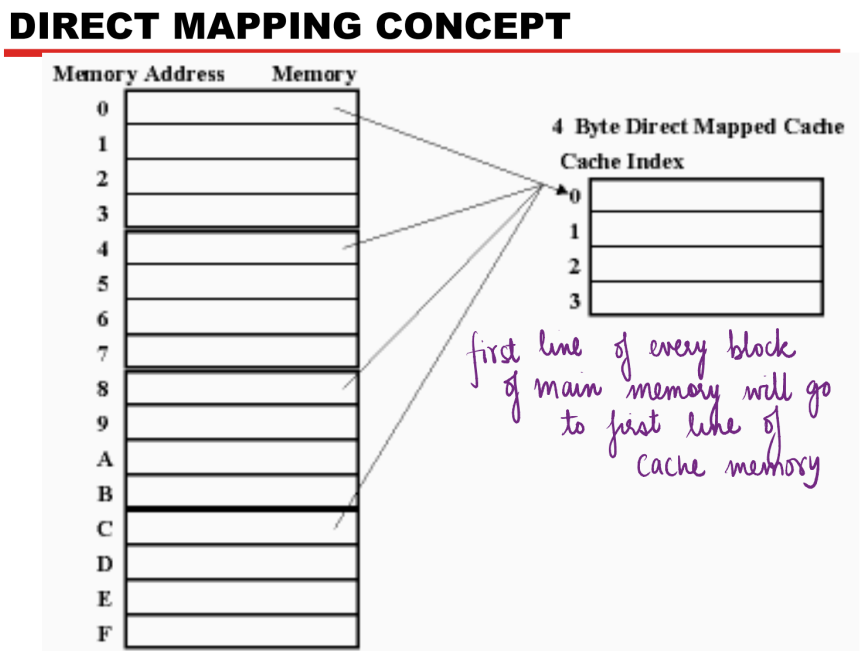
- It requires very less hardware complexity.
- It is a very simple method of implementation.
- Access time is very low.

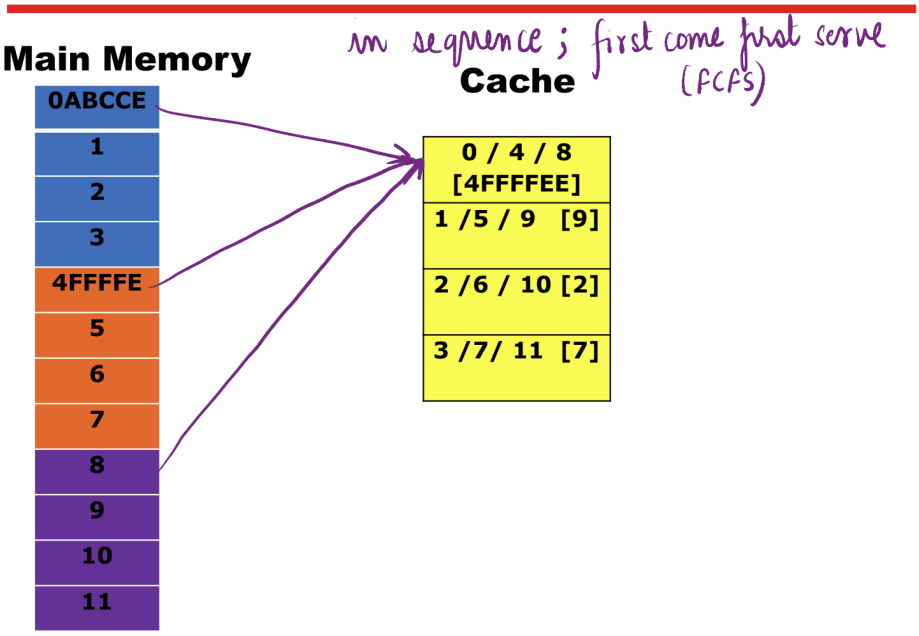
Disadvantages of Direct-mapping

- Cache performance is unpredictable in direct mapping.
- Handling of spatial locality is poor.
- Use of cache space is inefficient.

- Conflict misses are high.

Simple • Inexpensive • Fixed location for given block —If a program accesses 2 blocks that map to the same line repeatedly, cache misses are very high





Q21)What is Fully associative mapping?

A:

Associative Mapping (also known as **Fully Associative Mapping**) is a cache mapping technique where any block of memory can be stored in any cache line, allowing the cache to store data more flexibly.

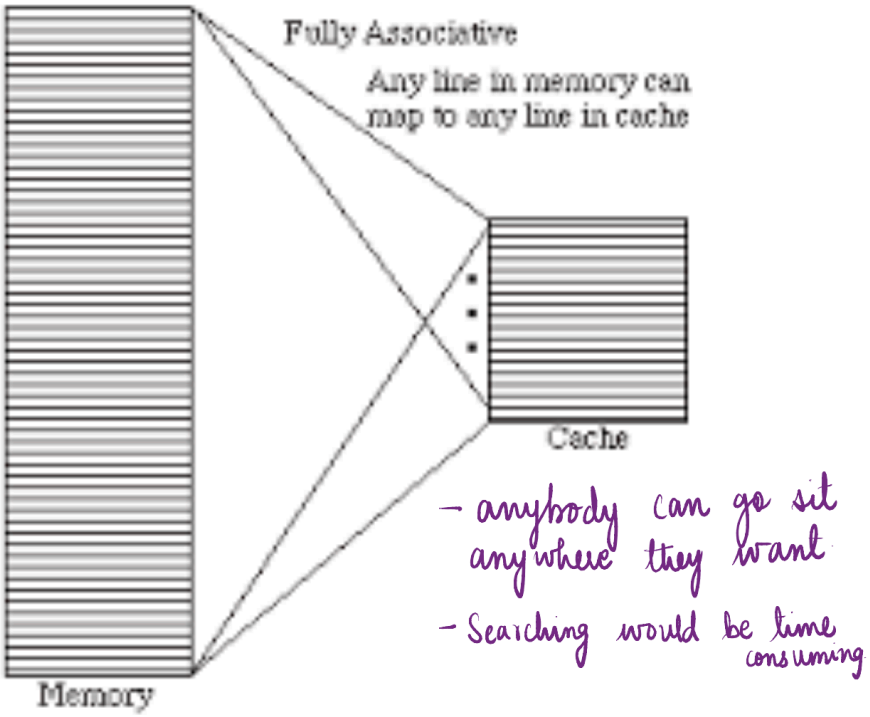
Key Points:

- In associative mapping, the **Tag** field holds the entire address (without any predefined index or line field), and every cache line can store any memory block.

- There is no fixed relationship between a memory block and a cache line, so the cache controller must search through all the cache lines to check if the required data is present (cache lookup).
- **Tag field** is compared with all cache lines to determine a hit or miss.

A main memory block can load into any line of cache • Memory address is interpreted as tag and word • Tag uniquely identifies block of memory • Every line's tag is examined for a match • Cache searching gets expensive.

FULLY ASSOCIATIVE MAPPING



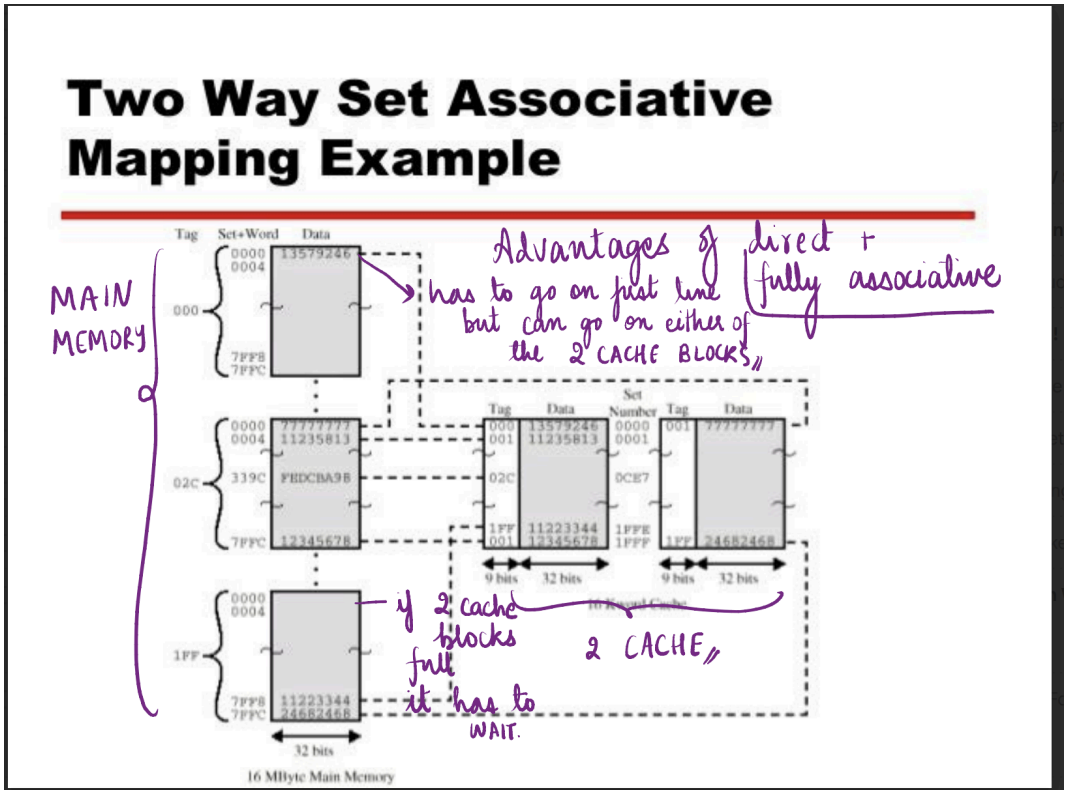
Q22)What is Set associative mapping?

A:

Set-Associative Mapping is a cache mapping technique that combines aspects of both **direct mapping** and **associative mapping**. In this method, the cache is divided into multiple sets, and each set contains a fixed number of lines (or slots). A given block of memory can be stored in any of the lines within a specific set.

Key Points:

1. **Cache divided into sets**: The cache is organized into multiple sets, each containing a small number of cache lines.
2. **Each set contains a number of lines**: Each set can store a predefined number of cache lines (e.g., 2, 4, etc.).
3. **Block maps to any line in a set**: A memory block is mapped to one of the lines in a specific set, but not to a particular line in the cache. The block can go into any available line within the set.
4. **Example (2-way set associative)**: If there are 2 lines per set, it's called **2-way associative mapping**, meaning that a given memory block can be placed in either of the 2 lines of one set.



Brushing up on Associative & Set Associative Mapping:

Associative Mapping:

- A main memory block can load into any line of cache
- Memory address is interpreted as tag and word
- Tag uniquely identifies block of memory
- Every line's tag is examined for a match
- Cache searching gets expensive

Set Associative Mapping:

- Cache is divided into a number of sets
- Each set contains a number of lines
- A given block maps to any line in a given set

- —e.g. Block B can be in any line of set i
- e.g. 2 lines per set
- —2 way associative mapping
- —A given block can be in one of 2 lines in only one set