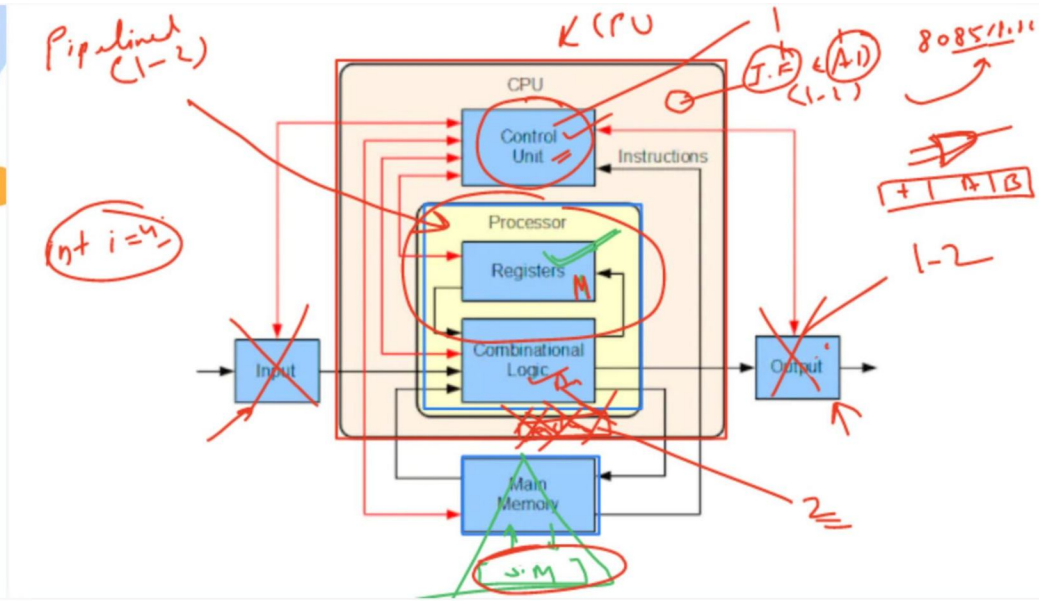
Memory Hierarchy

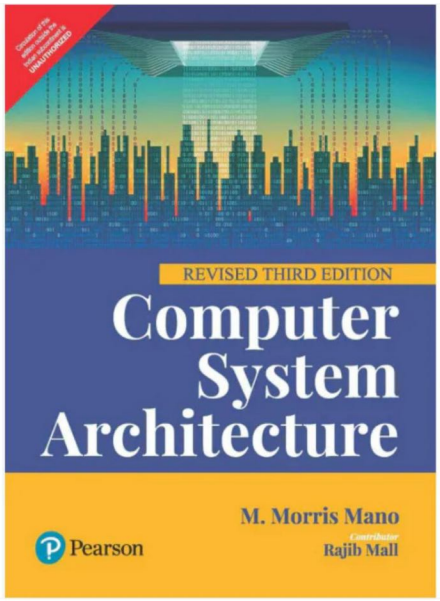
Complete Course on Computer Architecture for GATE 2021

Sanchit Jain • Lesson 1 • Nov 21, 2020

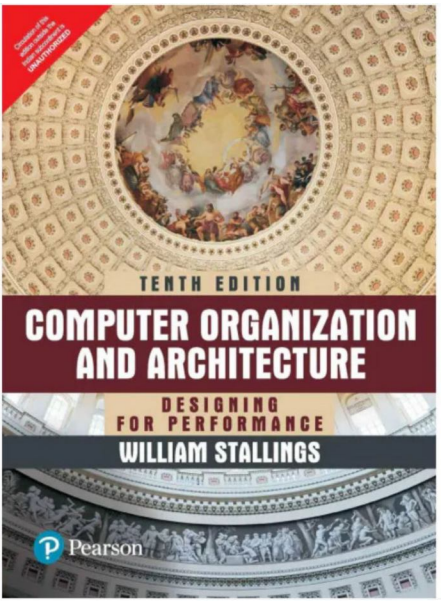


Computer Organization and Architecture

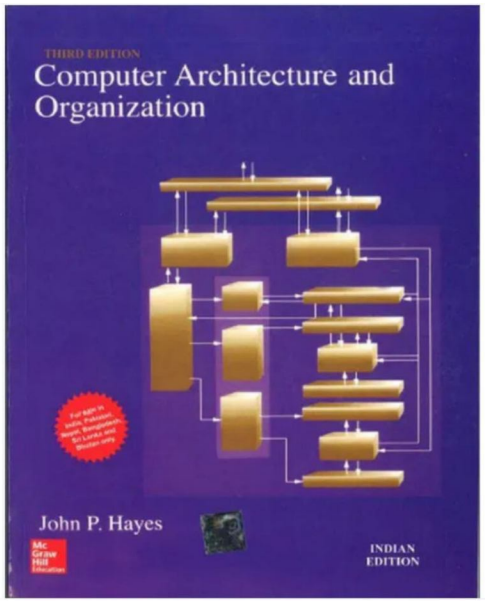
- Core subjects for CS/IT Students
- In GATE 7-8 Marks out of 100 Marks, and 5-6 questions on an average
- In NET 20-22 Marks out of 200 marks and 10-12 questions
- Both parts are important Theory and Numerical
- Needs a little time, have to fight to scoring
- Not required in Industry



- **Book Name**
  - Computer System Architecture
- **Writers**
  - M. Morris Mano
- **Publisher** – Pearson
- **Edition** – 3th



- **Book Name**
  - Computer Organization and Architecture
- **Writers** – William Stallings
- **Publisher** – Pearson
- **Edition** – 10th



- Book Name
  - Computer Architecture and Organization
- Writers -
  - John P. Hayes
- Publisher – McGraw Hill
- Edition – 3th

Syllabus

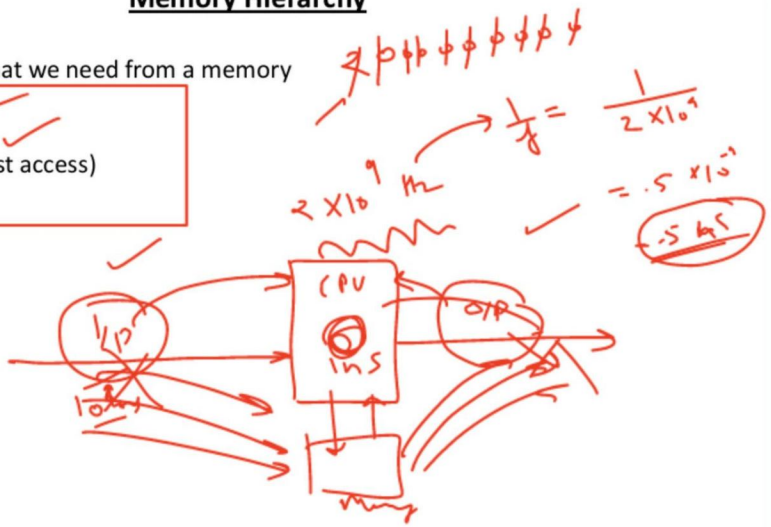
- Machine instructions and addressing modes.
- ALU, data path and control unit.
- Instruction pipelining.
- Memory hierarchy: cache, main memory and secondary storage.
- I/O interface (interrupt and DMA mode).

Syllabus

- **Memory organization**
  - Memory Hierarchy Basics
  - Cache memory and Mapping
  - principle of virtual memory
- **I/O**
  - addressing,
  - Data transfer technique,
  - program driven,
  - Interrupt driven and DMA
  - Disk, Tape memories, reliability & RAID
- **Pipeline & Control unit design**
  - Micro operations
  - Control unit design
- **Instructions**
  - Types of instructions
  - Addressing mode
  - Hardware, Micro-Programmed
- **Control unit design**
- **Miscellaneous**

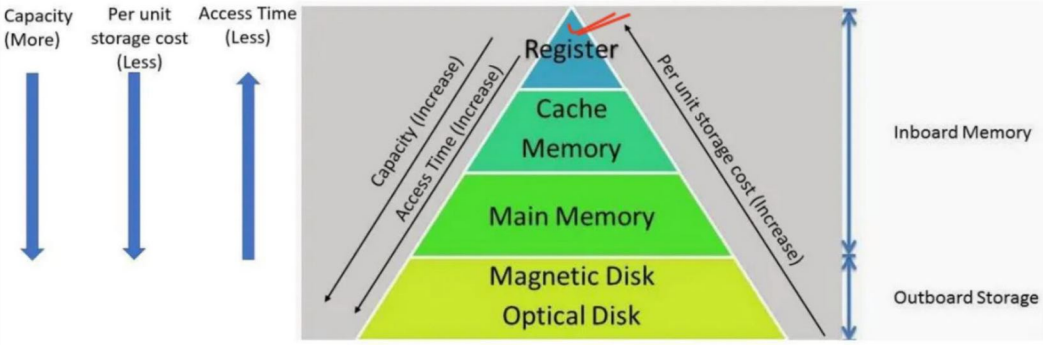
Memory Hierarchy

- Let first understand what we need from a memory
  - Large capacity
  - Less per unit cost
  - Less access time(fast access)



Memory Hierarchy

- The memory hierarchy system consists of all storage devices employed in a computer system from the slow but high-capacity auxiliary memory to a relatively faster main memory, to an even smaller and faster cache memory accessible to the high-speed



Cycle

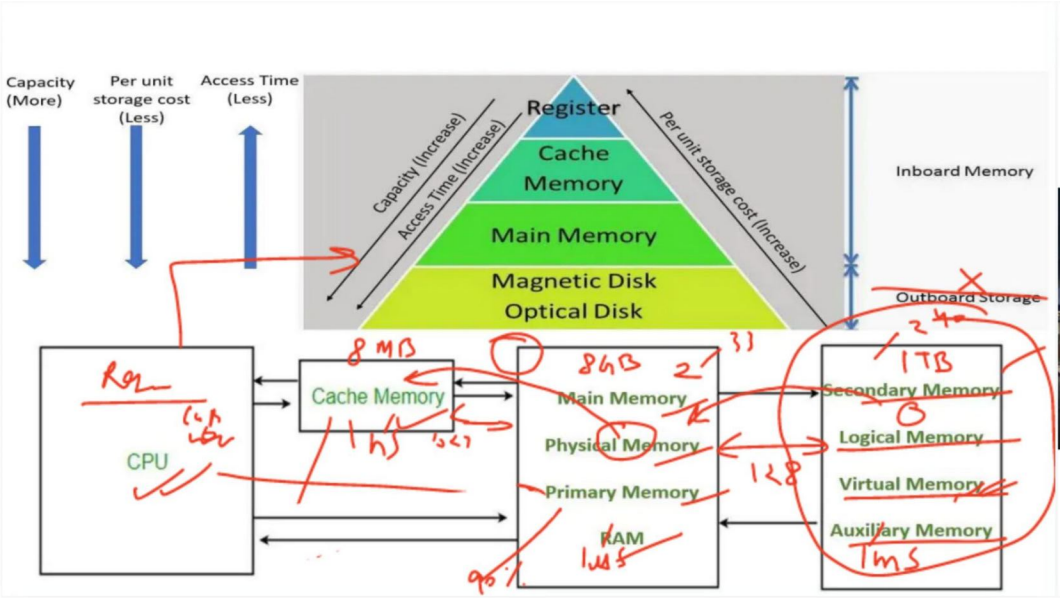
Car

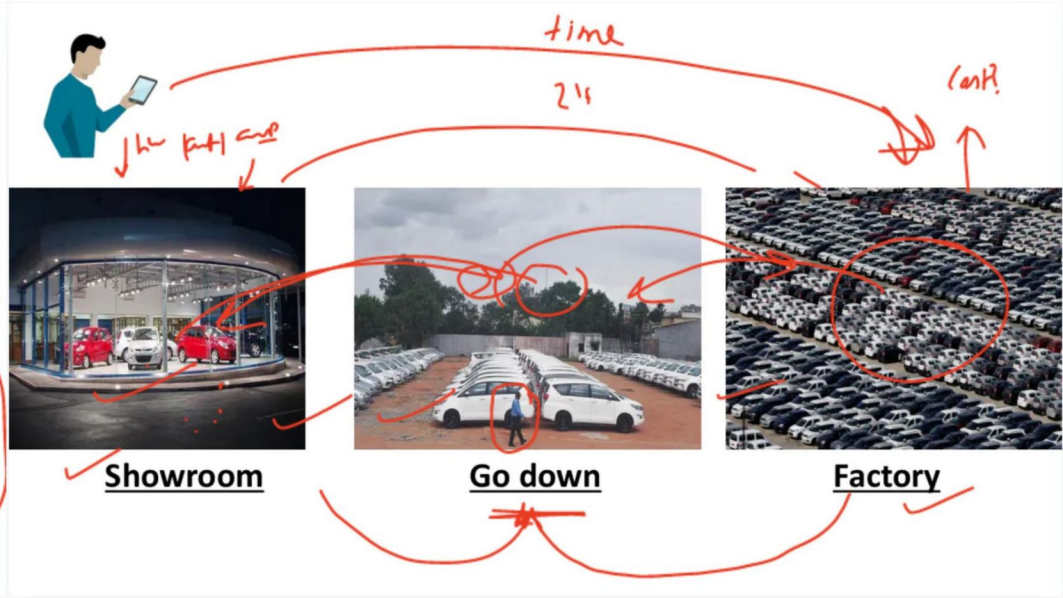
Airbus

Lathi

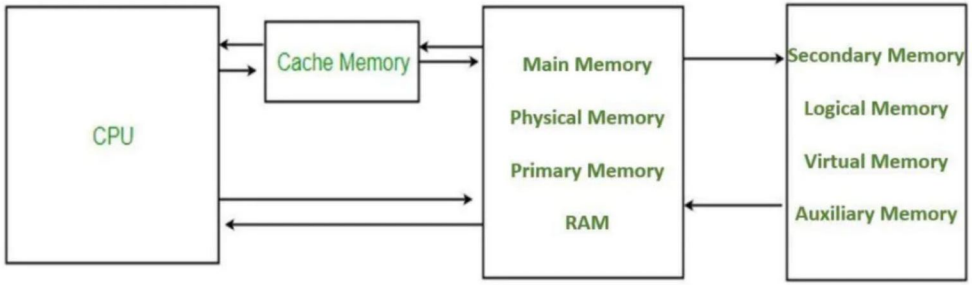
303

AK-47

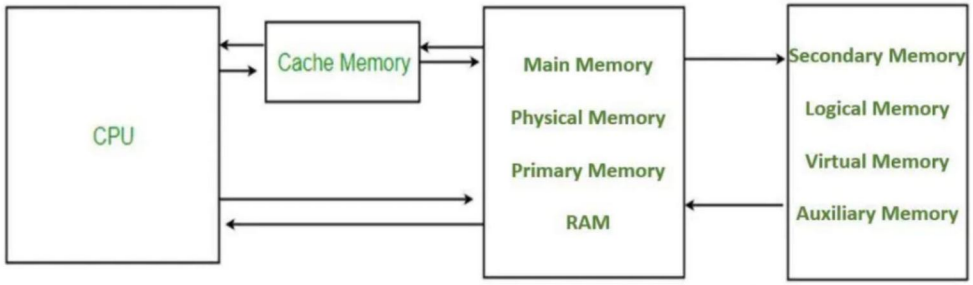




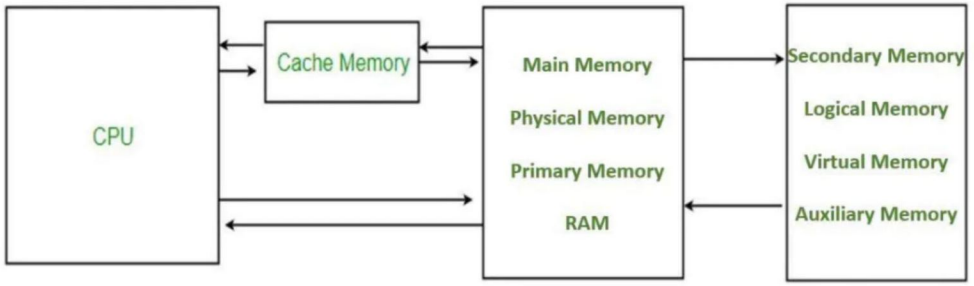
- If we want this hierarchy to work fine, then most of the time when CPU needs a data it must be present in Cache, if not possible main memory, worst case Secondary memory.
- But this is a difficult task to do as my computer has, 8 TB of Secondary Memory, 32 GB of Main Memory, but only 768KB, 4MB, 16 MB of L_1, L_2, L_3 cache respectively.
- If somehow we can estimate what data CPU will require if future we can prefetch in Cache and Main Memory.
- Locality of reference Helps we to perform this estimation.



- The main memory occupies a central position by being able to communicate directly with the CPU.
- When programs not residing in main memory are needed by the CPU, they are brought in from auxiliary memory.
- Programs not currently needed in main memory are transferred into auxiliary memory to provide space for currently used programs and data.

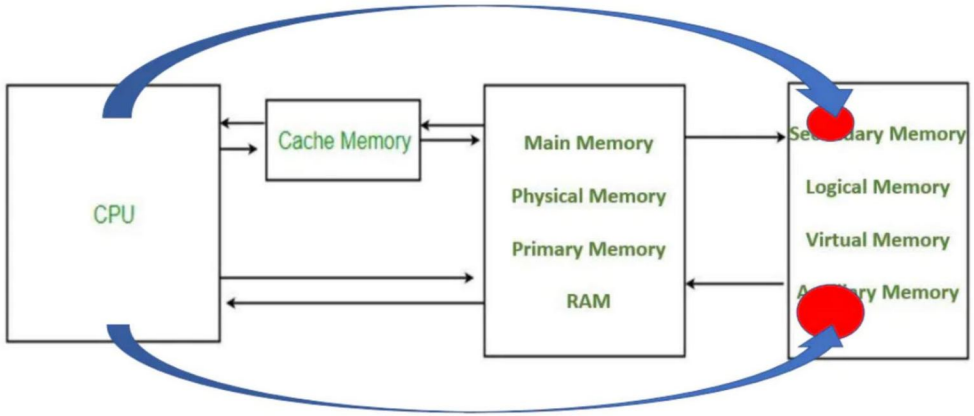


- A special very-high speed memory called a Cache is used to increase the speed of processing by making current programs and data available to the CPU at a rapid rate.
- The cache memory is employed in computer systems to compensate for the speed differential between main memory access time and processor logic.

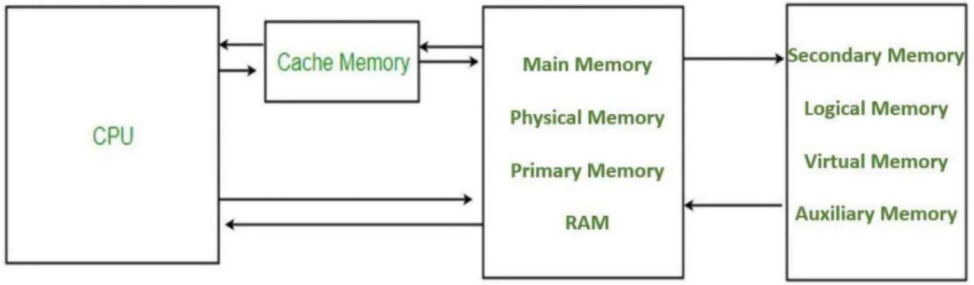


Locality of Reference

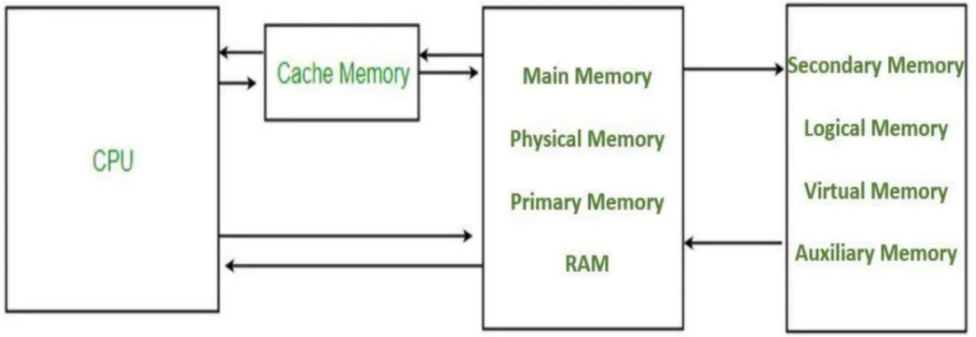
- The references to memory at any given interval of time tend to be confined within a few localized areas in memory. This phenomenon is known as the property of locality of reference. There are two types of locality of reference



- **Spatial Locality:** Spatial locality refers to the use of data elements in the nearby locations.



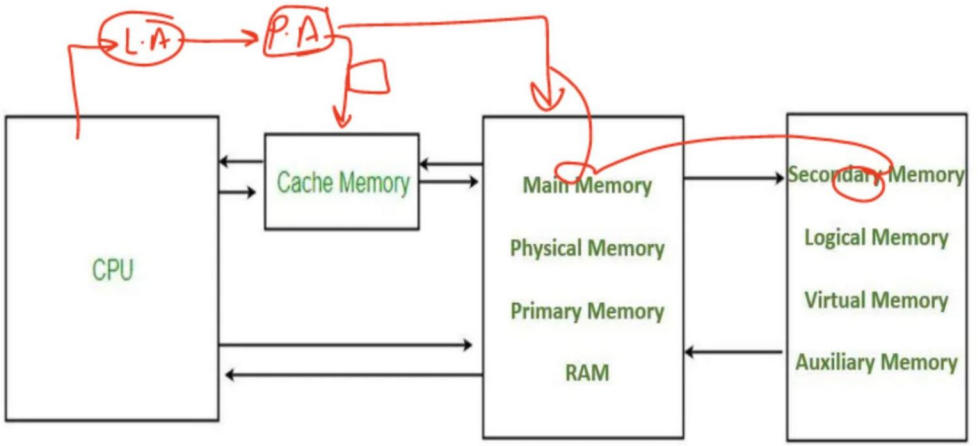
- **Temporal Locality:** Temporal locality refers to the reuse of specific data, and/or resources, within a relatively small-time duration. Or, the most frequently used items will be needed soon. (LRU is used for temporal locality)



Showroom

Go down

Factory



Q The principle of locality justifies the use of: **(Gate-1995) (1 Marks)**

a) Interrupts

b) DMA

c) Polling

d) Cache Memory

Q More than one word are put in one cache block to **(Gate-2001) (1 Marks)**

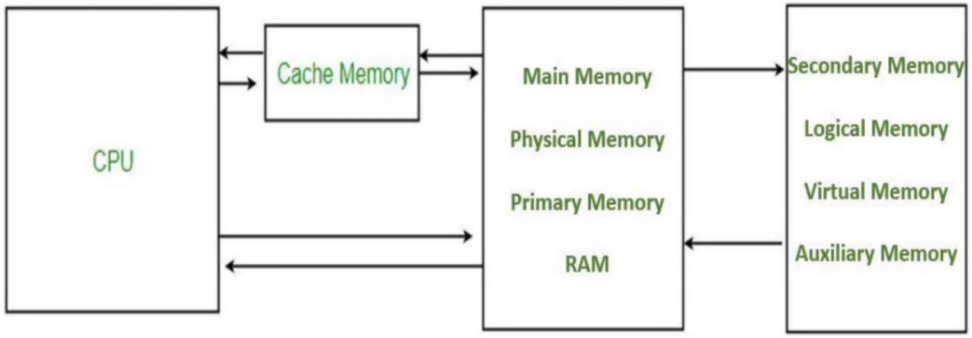
(A) exploit the temporal locality of reference in a program

(B) exploit the spatial locality of reference in a program

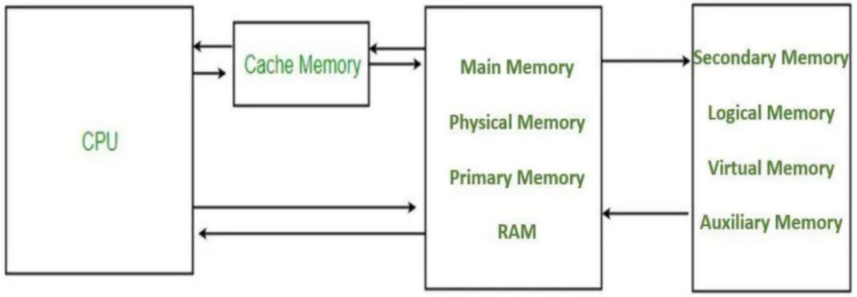
(C) reduce the miss penalty

(D) none of the above

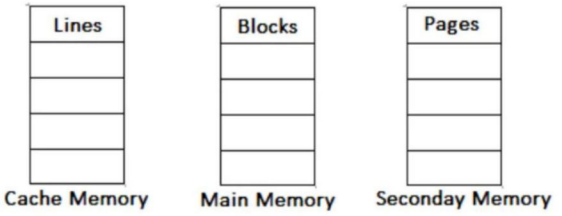
- **Cache Hit:** The performance of cache memory is frequently measured in terms of a quantity called hit ratio.
- When the CPU refers to memory and finds the word in cache, it is said to produce a hit.
- **Hit Latency:** The time required to match the data into cache memory is known as hit latency.

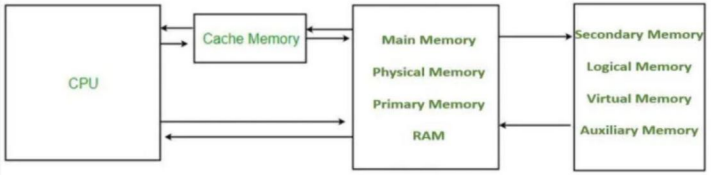


- **Cache Miss:** If the word is not found in cache, it is in main memory and it counts as a miss.
- **Miss Latency:** When cache miss occurs, the time required to get the needed data into cache memory from main memory. **Or,** The time required to recover from a cache miss.
- If the hit ratio is high enough so that most of the time the CPU accesses the cache instead of main memory, the average access time is closer to the access time of the fast cache memory.



Q In designing a computer's cache system, the cache block (or cache line) size is an important parameter. Which one of the following statements is correct in this context? **(Gate-2014) (1 Marks)**
(A) A smaller block size implies better spatial locality
(B) A smaller block size implies a smaller cache tag and hence lower cache tag overhead
(C) A smaller block size implies a larger cache tag and hence lower cache hit time
(D) A smaller block size incurs a lower cache miss penalty





- The transformation of data from main memory to cache memory is referred to as a mapping process.
- Three types of mapping procedures:
  - ***Direct mapping***
  - ***Associative mapping***
  - ***Set-associative mapping***

Q Which of the following mapping is not used for mapping process in cache memory? **(NET-JULY-2018)**

(a) Associative mapping **(b)** Direct mapping

(c) Set-Associative mapping **(d)** Segmented - page mapping

10^3	1 Thousand
10^6	1 Million
10^9	1 Billion
10^{12}	1 Trillion

10^3	1 kilo
10^6	1 Mega
10^9	1 Giga
10^{12}	1 Tera
10^{15}	1 Peta

2^{10}	1 kilo
2^{20}	1 Mega
2^{30}	1 Giga
2^{40}	1 Tera
2^{50}	1 Peta

Address Length in bits	n
No of Locations	2^n