Computer Architecture & Organization

Numerical Problems

Cache Memory

Question 1:

A block-set associative cache memory consists of 128 blocks divided into four block sets. The main memory consists of 16,384 blocks and each block contains 256 eight bit words.

1. How many bits are required for addressing the main memory?
2. How many bits are needed to represent the TAG, SET and WORD fields?

Solution:
Given-

- Number of blocks in cache memory = 128
- Number of blocks in each set of cache = 4
- Main memory size = 16384 blocks
- Block size = 256 bytes
- 1 word = 8 bits = 1 byte
    - **Main Memory Size-**
    -
    - We have-
    - Size of main memory
    - = 16384 blocks
    - = 16384 x 256 bytes
    - = 2^{22} bytes
    - Thus, Number of bits required to address main memory = 22 bits

Number of Bits in Block Offset-

We have-

Block size

= 256 bytes

= 2^8 bytes

Thus, Number of bits in block offset or word = 8 bits

Number of Bits in Set Number-

Number of sets in cache

= Number of lines in cache / Set size

= 128 blocks / 4 blocks

= 32 sets

= 2^5 sets

Thus, Number of bits in set number = 5 bits

Number of Bits in Tag Number-

Number of bits in tag

= Number of bits in physical address – (Number of bits in set number + Number of bits in word)

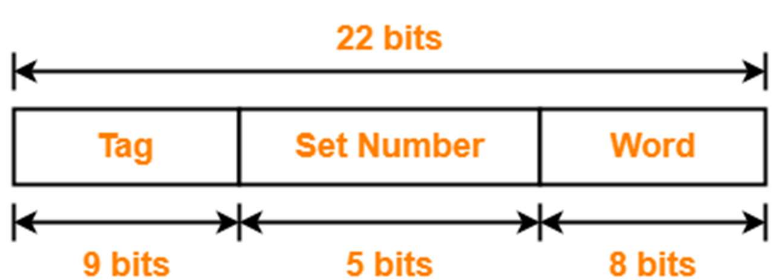
= 22 bits – (5 bits + 8 bits)

= 22 bits – 13 bits

= 9 bits

Thus, Number of bits in tag = 9 bits

Thus, physical address is-



Question 2:

A 4-way set associative cache memory unit with a capacity of 16 KB is built using a block size of 8 words. The word length is 32 bits. The size of the physical address space is 4 GB. The number of bits for the TAG field is _____.

Solution-

Given-

- Set size = 4 lines
- Cache memory size = 16 KB
- Block size = 8 words
- 1 word = 32 bits = 4 bytes
- Main memory size = 4 GB

Number of Bits in Physical Address-

We have,

Main memory size

= 4 GB

= 2^{32} bytes

Thus, Number of bits in physical address = 32 bits

Number of Bits in Block Offset-

We have,

Block size

= 8 words

= 8 x 4 bytes

= 32 bytes

= 2^5 bytes

Thus, Number of bits in block offset = 5 bits

Number of Lines in Cache-

Number of lines in cache

= Cache size / Line size

= 16 KB / 32 bytes

= 2^{14} bytes / 2^5 bytes

= 2^9 lines

= 512 lines

Thus, Number of lines in cache = 512 lines

<u>Number of Sets in Cache-</u>

Number of sets in cache
= Number of lines in cache / Set size
= 512 lines / 4 lines
= 2^9 lines / 2^2 lines
= 2^7 sets
Thus, Number of bits in set number = 7 bits

<u>Number of Bits in Tag-</u>

Number of bits in tag
= Number of bits in physical address – (Number of bits in set number + Number of bits in block offset)
= 32 bits – (7 bits + 5 bits)
= 32 bits – 12 bits
= 20 bits
Thus, number of bits in tag = 20 bits

Question 3:

A 4-way set associative cache memory unit with a capacity of 16 KB is built using a block size of 8 words. The word length is 32 bits. The size of the physical address space is 4 GB. The number of bits for the TAG field is _____.

Solution-

Given-

- Set size = 4 lines
- Cache memory size = 16 KB
- Block size = 8 words
- 1 word = 32 bits = 4 bytes
- Main memory size = 4 GB

Number of Bits in Physical Address-

We have,

Main memory size

= 4 GB

= 2^{32} bytes

Thus, Number of bits in physical address = 32 bits

Number of Bits in Block Offset-

We have,

Block size

= 8 words

= 8 x 4 bytes

= 32 bytes

= 2^5 bytes

Thus, Number of bits in block offset = 5 bits

Number of Lines in Cache-

Number of lines in cache

= Cache size / Line size

= 16 KB / 32 bytes

= 2^{14} bytes / 2^5 bytes

= 2^9 lines

= 512 lines

Thus, Number of lines in cache = 512 lines

Number of Sets in Cache-

Number of sets in cache

= Number of lines in cache / Set size

= 512 lines / 4 lines

= 2^9 lines / 2^2 lines

= 2^7 sets

Thus, Number of bits in set number = 7 bits

Number of Bits in Tag-

Number of bits in tag

= Number of bits in physical address – (Number of bits in set number + Number of bits in block offset)

= 32 bits – (7 bits + 5 bits)

= 32 bits – 12 bits

= 20 bits

Thus, number of bits in tag = 20 bits

Ques 4: Consider a direct mapped cache with 8 cache blocks (0-7). If the memory block requests are in the order-

3, 5, 2, 8, 0, 6, 3, 9, 16, 20, 17, 25, 18, 30, 24, 2, 63, 5, 82, 17, 24

Which of the following memory blocks will not be in the cache at the end of the sequence?

1. 3
2. 18
3. 20
4. 30

Also, calculate the hit ratio and miss ratio.

Solution-

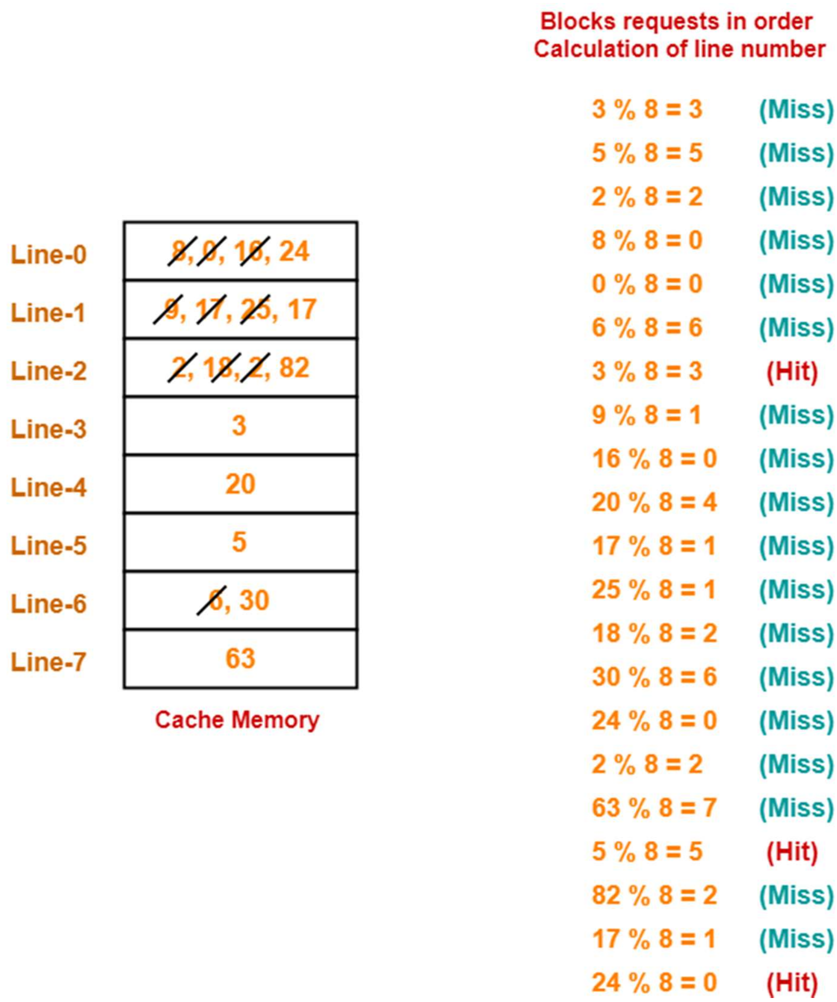
We have,

- There are 8 blocks in cache memory numbered from 0 to 7.
- In direct mapping, a particular block of main memory is mapped to a particular line of cache memory.
- The line number is given by-

Cache line number = Block address modulo Number of lines in cache

For the given sequence-

- Requests for memory blocks are generated one by one.
- The line number of the block is calculated using the above relation.
- Then, the block is placed in that particular line.
- If already there exists another block in that line, then it is replaced.



Thus,

- Out of given options, only block-18 is not present in the main memory.
- Option-(B) is correct.
- Hit ratio = 3 / 21
- Miss ratio = 17 / 21

Ques 5.

Consider a fully associative cache with 8 cache blocks (0-7). The memory block requests are in the order-

4, 3, 25, 8, 19, 6, 25, 8, 16, 35, 45, 22, 8, 3, 16, 25, 7

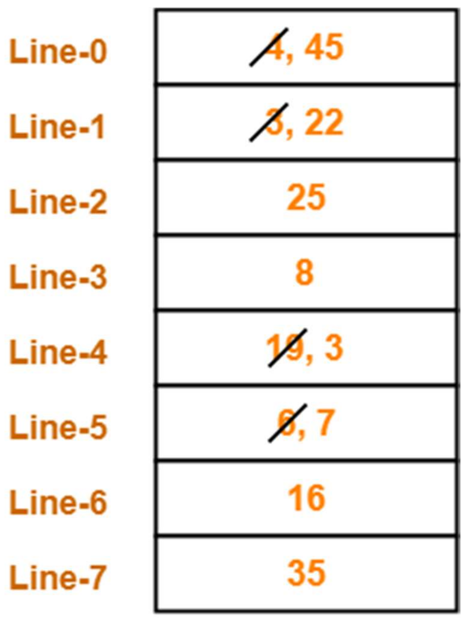
If LRU replacement policy is used, which cache block will have memory block 7?

Also, calculate the hit ratio and miss ratio.

<u>Solution-</u>

We have,

- There are 8 blocks in cache memory numbered from 0 to 7.
- In fully associative mapping, any block of main memory can be mapped to any line of the cache that is freely available.
- If all the cache lines are already occupied, then a block is replaced in accordance with the replacement policy.



Cache Memory

Thus,

- Line-5 contains the block-7.
- Hit ratio = 5 / 17
- Miss ratio = 12 / 17

Unsolved question:

Consider the cache has 4 blocks. For the memory references-

5, 12, 13, 17, 4, 12, 13, 17, 2, 13, 19, 13, 43, 61, 19

What is the hit ratio for the following cache replacement algorithms-

1. FIFO

2. LRU
3. Direct mapping
4. 2-way set associative mapping using LRU

Disk Structure & Organization

Q. Consider a (very old) disk with the following characteristics:
block size B=512 bytes
gap size G=128 bytes
20 sectors per track
400 tracks per surface
15 double-sided platters.

Solution:
1. What is the total capacity of a track?
2. What is its useful capacity of a track (excluding interblock gaps)
3. How many cylinders are there?
4. What is the total capacity of a cylinder?
5. What is the useful capacity of a disk?

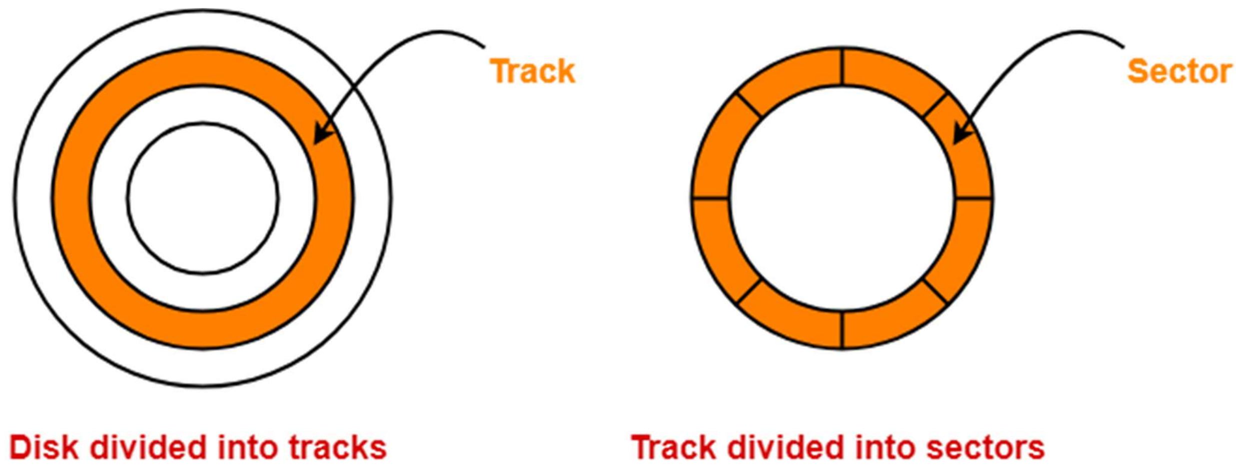
1. 20 * (512+128) = 12800 bytes = 12.5 KB
2. Useful = 20 * 512 = 10240 bytes = 10 KB
3. 400
4. Total = 15 * 2 * 20 * (512+128) = 384000 B = 375 KB
5. Useful = 15 * 2 * 400 * 20 * 512 = 117.18 MB

Theory on Disk structure

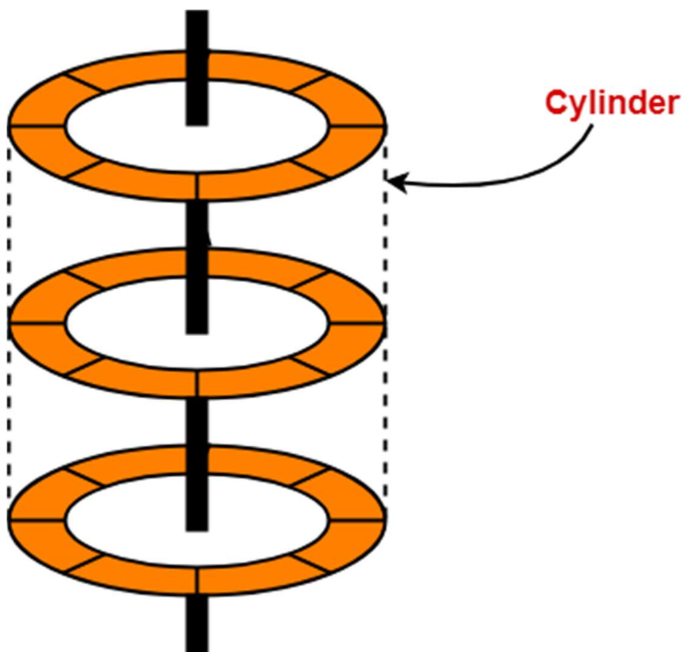
- Magnetic disk is a storage device that is used to write, rewrite and access data.
- It uses a magnetization process.

<u>Architecture-</u>

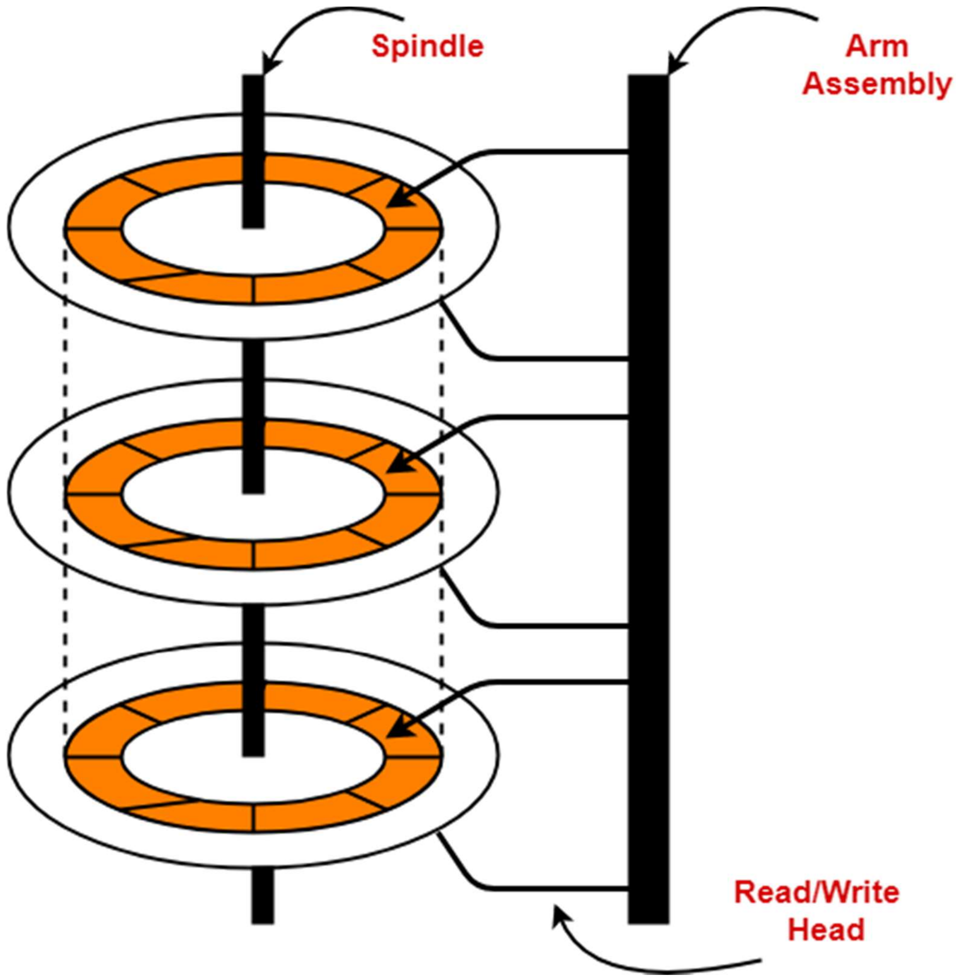
- The entire disk is divided into **platters**.
- Each platter consists of concentric circles called as **tracks**.
- These tracks are further divided into **sectors** which are the smallest divisions in the disk.



- A **cylinder** is formed by combining the tracks at a given radius of a disk pack.



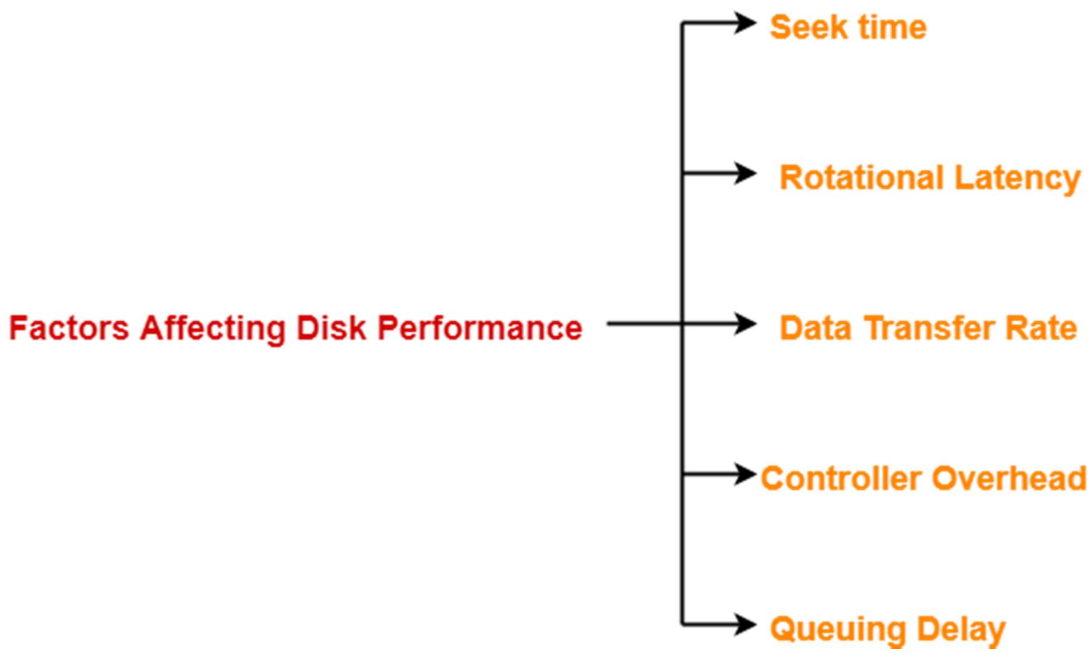
- There exists a mechanical arm called as **Read / Write head**.
- It is used to read from and write to the disk.
- Head has to reach at a particular track and then wait for the rotation of the platter.
- The rotation causes the required sector of the track to come under the head.
- Each platter has 2 surfaces- top and bottom and both the surfaces are used to store the data.
- Each surface has its own read / write head.



Disk Performance Parameters-

The time taken by the disk to complete an I/O request is called as **disk service time** or **disk access time**.

Components that contribute to the service time are:



1. Seek time
2. Rotational latency
3. Data transfer rate
4. Controller overhead
5. Queuing delay

1. Seek Time-

- The time taken by the read / write head to reach the desired track is called as seek time.
- It is the component which contributes the largest percentage of the disk service time.
- The lower the seek time, the faster the I/O operation.

2. Rotational Latency-

The time taken by the desired sector to come under the read / write head is called as rotational latency.

It depends on the rotation speed of the spindle.

Average rotational latency = 1 / 2 x Time taken for full rotation

3. Data Transfer Rate-

- The amount of data that passes under the read / write head in a given amount of time is called as **data transfer rate**.
- The time taken to transfer the data is called as **transfer time**.

It depends on the following factors-

1. Number of bytes to be transferred
2. Rotation speed of the disk
3. Density of the track
4. Speed of the electronics that connects the disk to the computer

4. Controller Overhead-

- The overhead imposed by the disk controller is called as **controller overhead**.
- Disk controller is a device that manages the disk.

5. Queuing Delay-

- The time spent waiting for the disk to become free is called as **queuing delay**.

Storage Density-

- All the tracks of a disk have the same storage capacity.
- This is because each track has different storage density.
- Storage density decreases as we from one track to another track away from the center.

Thus,

- Innermost track has maximum storage density.
- Outermost track has minimum storage density.

1. Disk Access Time-

Disk access time is calculated as-

Disk access time

= Seek time + Rotational delay + Transfer time + Controller overhead + Queuing delay

3. Average Seek Time-

Average seek time is calculated as-

Average seek time

= 1 / 3 x Time taken for one full stroke

4. Average Rotational Latency-

Average rotational latency is calculated as-

Average rotational latency = 1 / 2 x Time taken for one full rotation

Average rotational latency may also be referred as-

- Average rotational delay
- Average latency
- Average delay

5. Capacity Of Disk Pack-

Capacity of a disk pack is calculated as-

Capacity of a disk pack

= Total number of surfaces x Number of tracks per surface x Number of sectors per track x Storage capacity of one sector

Points to Remember-

- The entire disk space is not usable for storage because some space is wasted in formatting.
- When rotational latency is not given, use average rotational latency for solving numerical problems.
- When seek time is not given, use average seek time for solving numerical problems.
- It is wrong to say that as we move from one track to another away from the center, the capacity increases.
- All the tracks have same storage capacity.