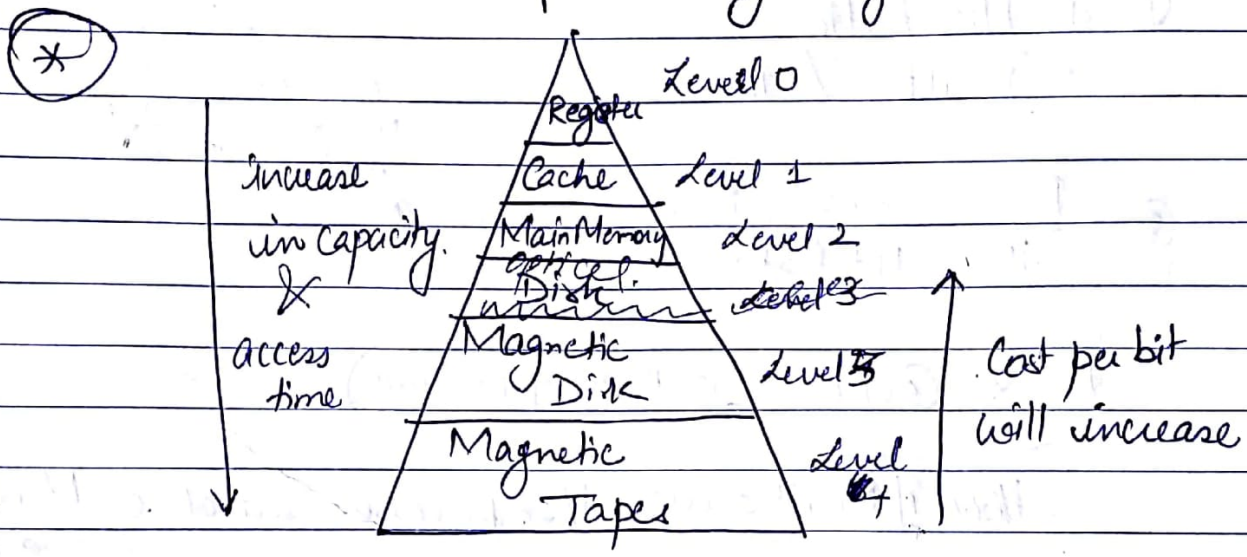
Memory Organization — how memory is organized in a system.

Memory Hierarchy :—

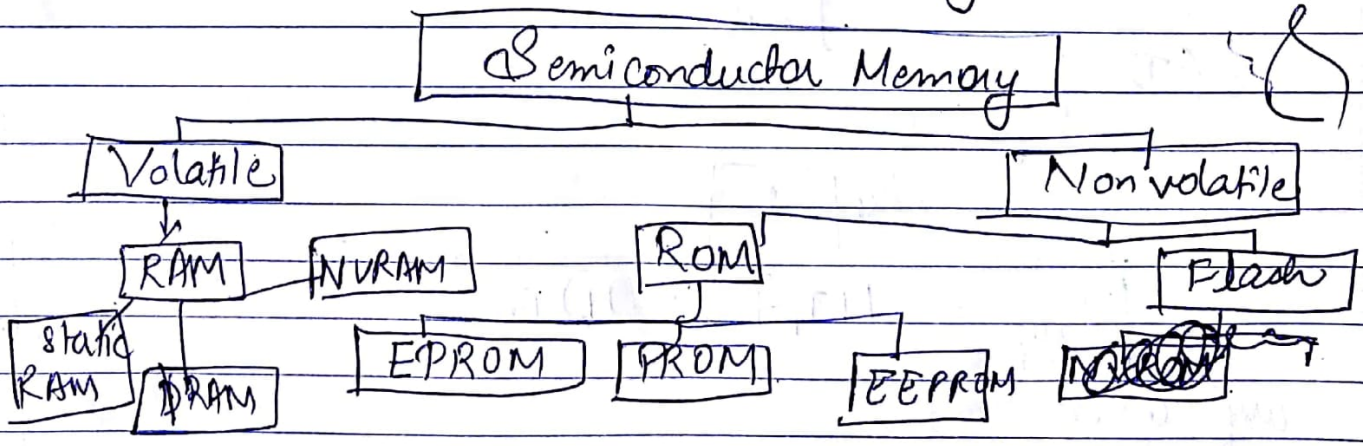
→ Computer memory should be fast, large & inexpensive.

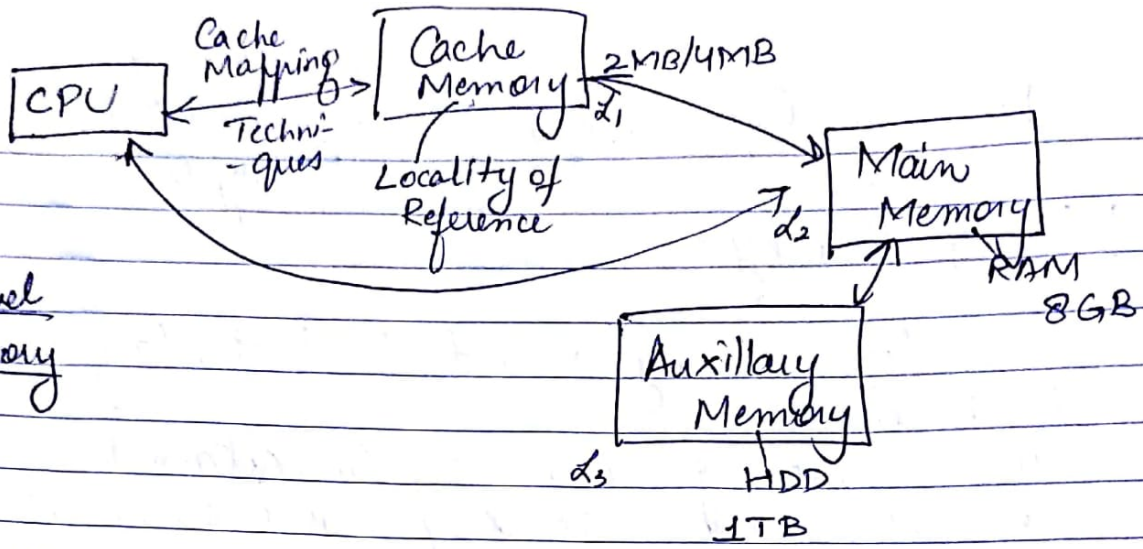
→ The memory hierarchy is to obtain the highest possible access speed while minimizing the total cost of memory system.



Main Memory :— It stores the data & program during computer operations.

⇒ It uses semiconductor technology hence known as semiconductor memory.

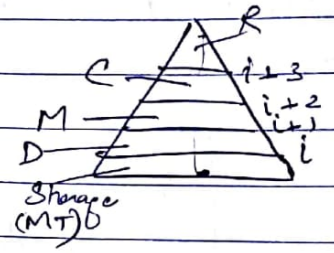




2 Level Memory

Memory :-

1. Hit Ratio
2. Fault / Miss

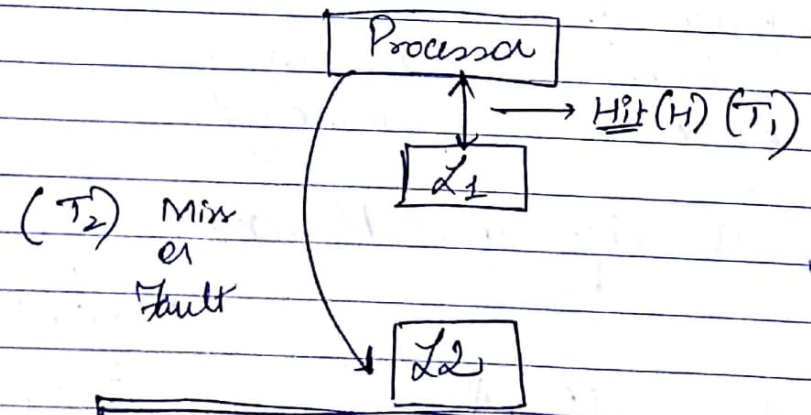


if (Processor) P → Cache (Found)
then Hit Ratio

if P → Cache → Main Memory (Found)

if Fault occurs then processor searches Main Memory

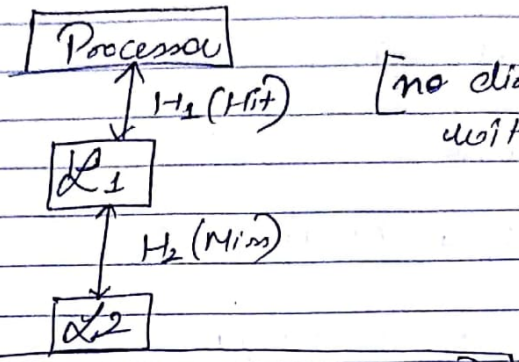
Case 1 Default Approach :-



$$T_{avg} = HT_1 + (1-H_2)T_2$$

avg access time

Case 2. Strict Approach



$$T_{avg} = H_1 T_1 + (1 - T_1)(T_1 + T_2)$$

Ques 1 Consider a 2 Level Memory system in which access time of Level 1 & Level 2 memories are 10 nsec & 150 nsec respectively. What is average access time when Level 1 Hit ratio = 90% ?

Sol 1

$$T_{avg} = HT_1 + (1 - H)T_2$$

$$\Rightarrow 0.9 \times 10 + (0.1)\,150$$

$$\Rightarrow 9 + 15 = 24 \text{ nsec.}$$

Ques 2 In a 2 Level memory system, the average access time without Level 1 is 150 nsec & with Level 1 is 30 nsec.

If Level 1 access time is 20 nsec, then

(a) What is Hit Ratio ?

(b) If Hit Ratio is made to 100%, What will be the access time of Level 1 & Level 2 memory

(c) If average access time is increased by 10% then what will be % change in the Hit Ratio?

8.12:

(a) $T_2 = 150$ nsec

$T_{avg} = 30$ nsec.

$T_1 = 20$ nsec.

$$T_{avg} = HT_1 + (1-H)T_2$$

$$30 = H(20) + (1-H)150$$

$$30 = 20H + 150 - 150H$$

$$150H - 20H = 150 - 30$$

$$130H = 120$$

$$H = 120/130 = 92.3\%$$

(b.) Access time of individual level 1 & level 2 is independent of Hit Ratio.

so no change.

(c.) $\uparrow T_{avg} = HT_1 + (1-H)T_2$

increased by 10%

$\Rightarrow$ 10% of 30 nsec. $\Rightarrow$ 33 nsec.

$$33 \text{ sec} = 20 \times H + (1-H)150$$

$$H = 90\%$$

Percentage change $\Rightarrow$ 92.3 − 90

$\Rightarrow$ 2.3%.

Q.3 Assuming Hit = 0.8 in Level 1 memory in a 2 Level memory system, the average access time is 60nsec. Moreover, the Level 1 memory is 5 times faster than Level 2.

If average access time is increased by 20%. What is the % change in Hit Ratio?

a) 10% increase
b) 20% increase
c) 10% decrease
d) 20% decrease.

Sol 3 $\boxed{T_1 = \frac{T_2}{5}}$

$T_{avg} = 60nsec = 72nsec$

increased by 20%

$60 \times \frac{20}{160} = 12$

$60 + 12$

$60 = 0.8 T_1 + 0.2 (5T_1)$

$T_2 = 166.66$

$T_1 = 33.33$

$72 = HT_1 + (1-H) T_2$

$72 \Rightarrow H \times 33.33 + (1-H) \times 166.66$

$H = 70\%$

Diff = 80 - 70 = 10% decrease

Principle — ? Principle of Locality.

A cache has used a word from a memory block ranged b/w 0-63. If the same word is required soon then it will exploit

Ques 4 A cache is having 60% hit ratio for read operation. Cache access time is 30 ns and main memory access time is 100 ns. What will be average access time for read operation?

Sol 4

$T_1 = 30ns \quad T_2 = 100ns \quad H = 0.6$

$\Rightarrow 0.6 \times 30 + (0.4) \times 100$

Avg access Time $\Rightarrow 18 + 40 = 58$ nsec

Ques 5 Consider a system with 80% hit ratio, 50 nano-seconds time to search the associative registers, 750 nano-seconds time to access memory. Find the ~~time to access~~ exp. effective memory access time.

Ques 6 In a two-level memory hierarchy, if the cache has an access time of 40 ns and main memory has an access time of 60 ns. What is the hit ratio in cache required to give an avg access time of 40 ns?

$40 = H_1 40 + (1 - H_1) 60$

$40 = H_1 40 + 60 - H_1 60$

$-5\phi = -2\phi H$

$H = 5/2$ ⇒ 250%

72

Direct Mapping & Associative Mapping in cache memory

Cache Mapping

What its represents?

→ It tells that which word of main memory will be placed at which location of cache memory.

→ Mapping b/w the cache addresses & main memory addresses refering to same unit of info.

Types of Mapping

1. Direct Mapping
2. ~~Indirect~~ Associative Mapping
3. Set Associative Mapping.

Direct Mapping

If the i^{th} block of main memory has to be placed in j^{th} block of cache memory, then

$$j = i \bmod (\text{No. of blocks in cache})$$

Example

Let us say that there are 128 blocks in cache block in cache memory. Then the scenario is like:—

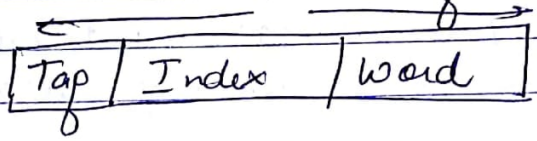
MM Block 0^{th} block $\xrightarrow{\text{mapped to}}$ 0 mod 128 = 0^{th} block of cache

1^{st} block $\xrightarrow{\text{mapped to}}$ 1 mod 128 = 1^{st} block of cache.

Direct Mapping Address space

In direct mapping we divide the main memory addressing space into 3 parts:—

I word
II index
III tag



Word → No. of bits required to identify a particular word within the block.

$= \log_2$ (No. of words/block)

Index → No. of bits required to identify the block no. of cache memory where a MM block will be placed.

$= \log_2$ (No. of blocks in cache)

Example:- Let there be given a 16 word 64 k w main memory & 128 line cache. Then find the corresponding Tag, Index & word bits in Direct Mapping.

Sol: 128 Line Cache $\xrightarrow{\text{implies}}$ 128 blocks in cache

Similarly, no. of blocks in M.M = $\frac{64\,KW}{16\,W}$

$= 4K = 4 \times 2^{10} = 2^{12}$

Also, since MM size = 64 KW

$= 2^6 \times 2^{10} W$

$= 2^{16} W$

⇒ 16 bits are required to represent the address space of MM.

⇒ Now Index = $\log_2$ (No. of Blocks in cache)

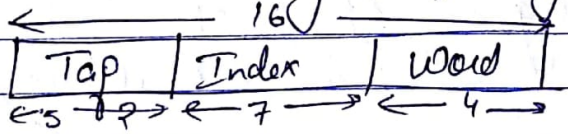
$= \log_2 (128) = \log_2 (2^7)$

$= 7 \log_2 2 = 7 \times 1 = 7$

& Word = $\log_2$ (No. of words / block)

$= \log_2$ (block size in words)

$= \log_2 (16) = \log_2 (2^4) = 4$



Tag bits = Total − Index bits − word bits

$= 16 - 7 - 4 = 16 - 11 = 5$

Numerical

Consider a machine with a byte addressable main memory of 2^{20} bytes, block size of 16 bytes & a direct mapping cache having 2^{12} cache lines. Let the address of two consecutive bytes in main memory are $(E201F)_{16}$ & $(E2020)_{16}$. What are the tag & cache line addresses (in hex) for main memory address $(E201F)_{16}$

Given

Sol: MM size = 2^{20} bytes

⇒ Total no of bits required to represent the addressing space = $\log_2$ (size of MM)
$= \log_2(2^{20}) = 20 \log_2 2 = 20$

Block size = 16 bytes.

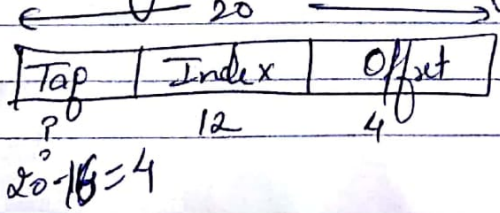
⇒ Offset bits = $\log_2$ (block size) = $\log_2$ (16 bytes) = $\log(2^4)$

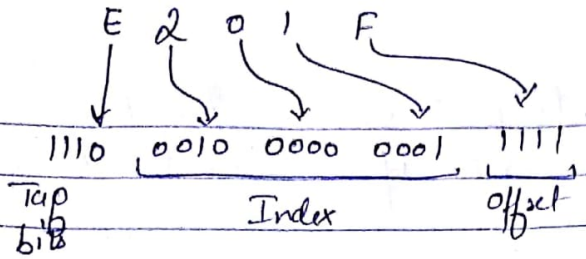
$4 \log_2 2 = 4 \times 1 = 4$

⇒ Block in cache = 2^{12} (2^{12} cache lines)

⇒ Index bits = $\log_2$ (No. of block lines)

$= \log_2(2^{12}) = 12 \log_2 2 = 12 * 1 = 12$





E, 201

The demerit of direct mapping is that hit ratio drops considerably if two or more words having same index & diff. tags are accessed consecutively one after the other.

Principle of Locality :- It is used in cache memory to help the program access small amounts of address space at any instant.

How to manage mapping from main memory to cache?

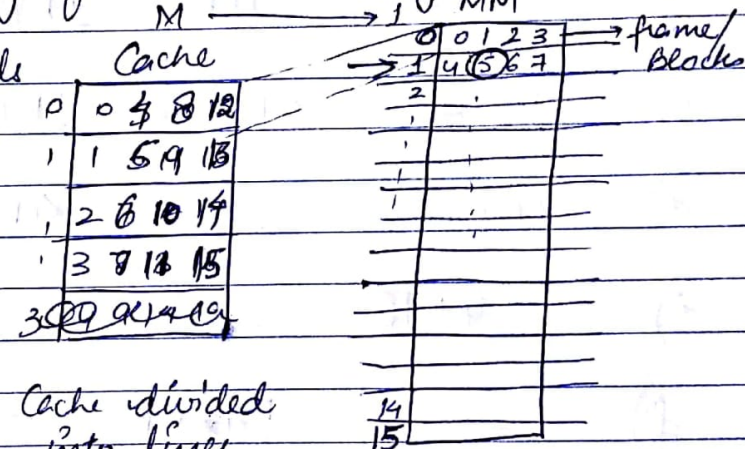
Suppose MM is 64 words & Cache is 16 words

Suppose Block size = 4 words

How many blocks in MM

$\Rightarrow \frac{64}{4}$

$\Rightarrow$ 16 blocks



No. of lines in Cache = $\frac{16}{4}$ = 4 lines

No. of bits in address space = $2^6 \Rightarrow$ 6 bits

1w = 1B

eg:- 0001 | 01

Block | block offset

20 5 10

Associative Mapping

Q. A direct mapped cache is of size 64kB with block size 32B. Logical address generated is 32bit. What is the bits required for block field?

Q. A digital computer has a memory unit of 64K*16 & a cache memory of 1K words. The cache uses direct mapping with a block size of 4words.

a) How many bits are there is tag, index & to block or word fields of address format.
b) How many blocks can cache accomodate?

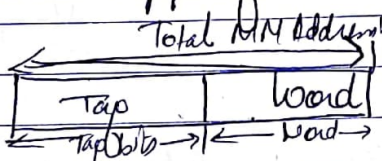
Associative Mapping

Drawback of Direct Mapping

↳ High Conflict Miss → We had to replace a cache memory block even when other blocks in the cache memory were present as empty.

Idea: To avoid high conflict miss, any block of MM can be placed anywhere in cache.

Associative Mapping Addressing scheme



Tag bits = used to distinguish any 2 blocks inside the cache
= $\log_2$ (No. of Blocks in MM)

word = $\log_2$ (No of words / block)

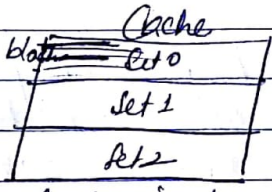
Advantage:— 0% of conflict miss occurs

Disadv:— Very high Tag comparisons are required.

No. of Tag Comparisons = No. of blocks inside the cache (Entire cache to be searched)

Set Associative Mapping

→ To overcome the high conflict miss in direct Mapping &
→ Large Tag comparisons in case of Associative Mapping.



Concept:-

1.) Cache blocks are divided into sets.
2.) Set size is always in the power of 2.
i.e if cache has 2^{way} blocks/set, then it is called 2^{n} set associativity.

Eg:- we need to map B_0 → set 0 → map to first available

4 way set associative → 4 blocks/set
2 way set " → 2 blocks/set.

Tag	set	word

word → no. of bits required to identify particular word within block. log (No. of words/block)
Tag → no. of bits req. to compare two blocks which belong to same set.

If i^{th} block of MM have to be placed in j^{th} set of cache-memory, then $j = i \bmod$ (no. of blocks in cache)

Set = No. of bits required to identify corresponding set no. inside cache where a MM block will be placed = $\log_2$ (No. of sets in cache)

Example: size of MM = 64 KW = $2^6 \times 2^{10}$ W = 2^{16} W
= 16 bit address

Cache memory size = 128
with each block size = 16 W
1 word = 4 bit
2 way set
word bits = $\log_2(16) = 4$

2-way set associative = sets = $\frac{128}{2} = 64$

⇒ set bits = $\log_2(64) = 6$

Tag bits = 16 − 4 − 6 = 6

6	6	4
Tag	set	word

MM Block	Cache memory sets
0th block of MM	0 mod 64 = 0th set of cache
1st	1 mod 64 = 1st set
1	
63	63 mod 64 = 63rd
64	64 mod = 0th set of cache.

Numerical — Set associative

A 4-way set associative cache memory unit with a capacity of 16KB is built using a block size of 8 words. The word length is 32 bits. The size of physical address space is 4GB. The no. of bits of tag field is?

Sol:-

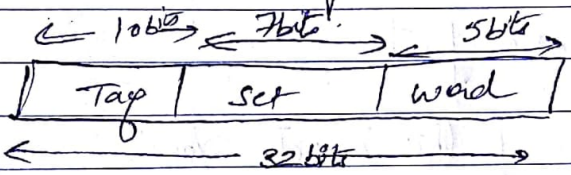
4 way set associative ⇒ 4 blocks/set.

Cache size = 16KB = 2^{14}B

Block size = 8 words.

Word length = 32 bits

size of address space = 4GB = $2^2 \times 2^{30}$ = 2^{32}



8×4 or 2^{3+2}

word ⇒ no. of words/block = 2^5

set = no. of sets in cache

Tag ⇒ no. of blocks in cache = $\frac{\text{size of cache}}{\text{size of 1 block}}$

$\Rightarrow \frac{16KB}{8 \times 32} = \frac{2^{14} \times 2^3}{2^8} \Rightarrow 2^9$

$= 512$

No. of sets = $\frac{512}{4} = 128$

set = $\log_2 128 = 2^7 \Rightarrow 7$

Q. A 2 way set associative cache is 256 K bytes in ~~cache~~ size. What is no. of sets if block size is 16 bytes?

2 way set = 2 blocks / set

Cache size = 256×2^{10}

$\Rightarrow 2^{18}$

Block size = 16 bytes

no. of blocks in cache = $\frac{\text{size of cache}}{\text{size of 1 block}}$

$= \frac{2^{18} \times 2^{3}}{2^{4} \times 2^{3}} \Rightarrow 2^{14}$

no. of sets $\Rightarrow \frac{2^{14}}{2^{1}} \Rightarrow 2^{13}$

no. of bits in sets $= \log_2 2^{13}$

$= 13 \log_2 2$ ($\log_2 2 = 1$)

$= 13$ bits