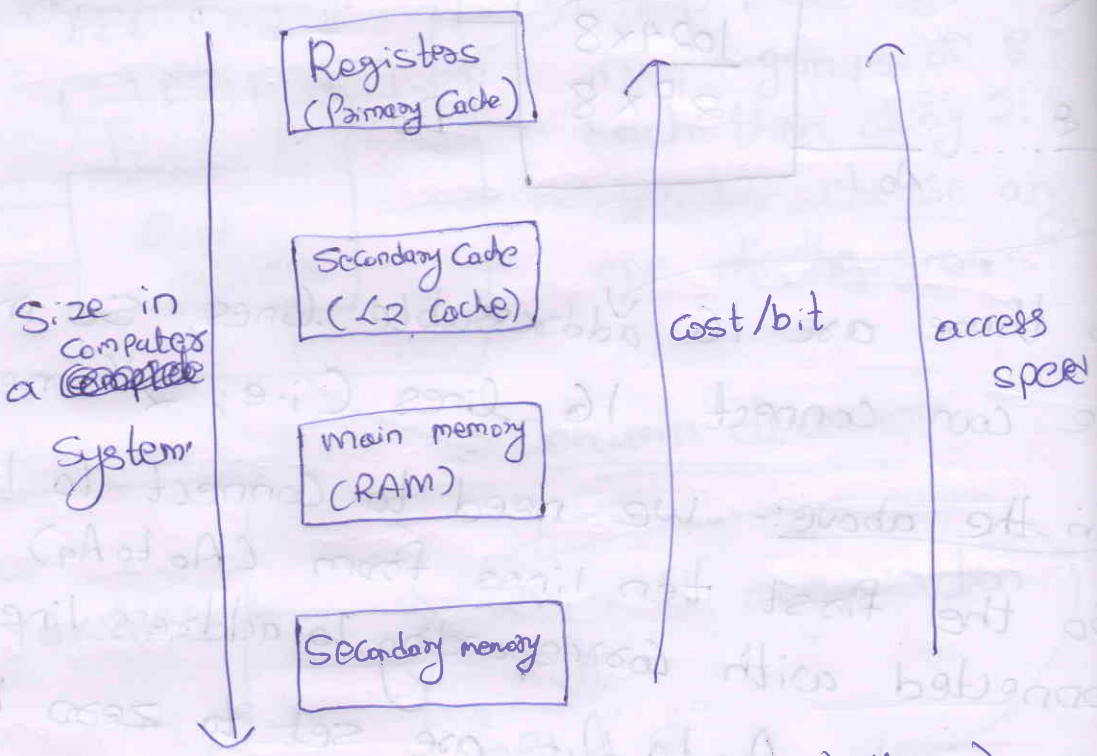
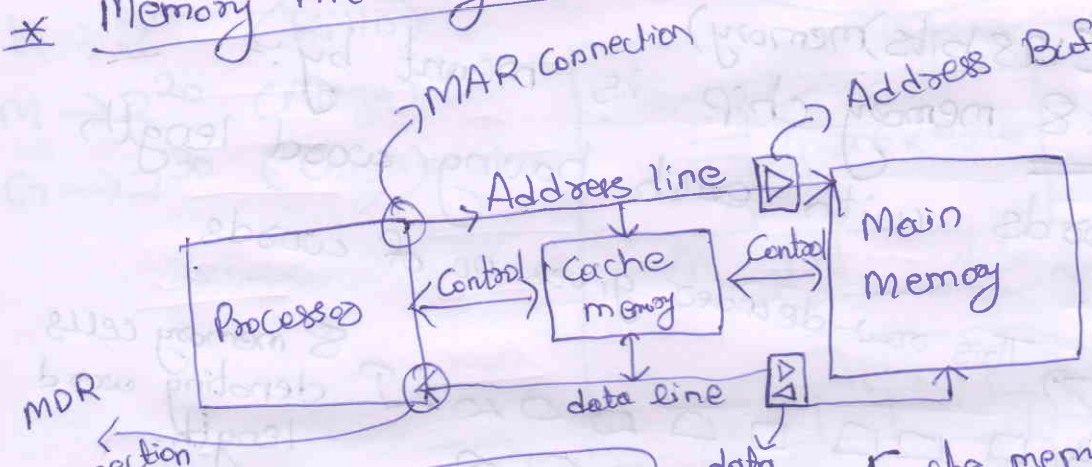
Memory System

* Memory Hierarchy



* Memory hierarchy and Mapping (Hamacher/Stallings)



$|C| << |M|$

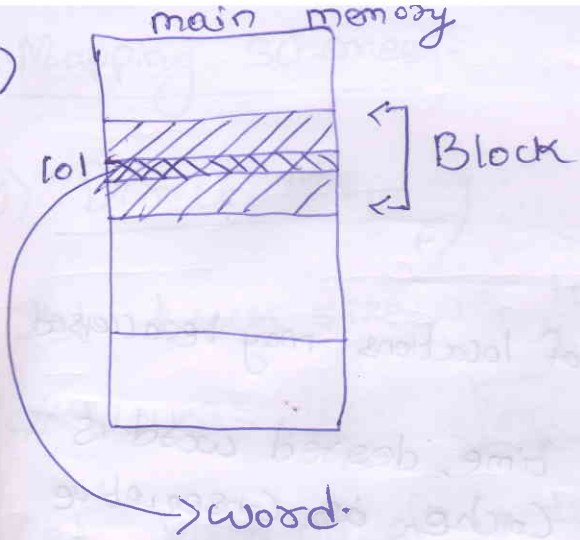
Size of Cache memory

Size of main memory

Cache memory is faster than main memory but less than processor

Typical Cache organization
(Stallings)

•)

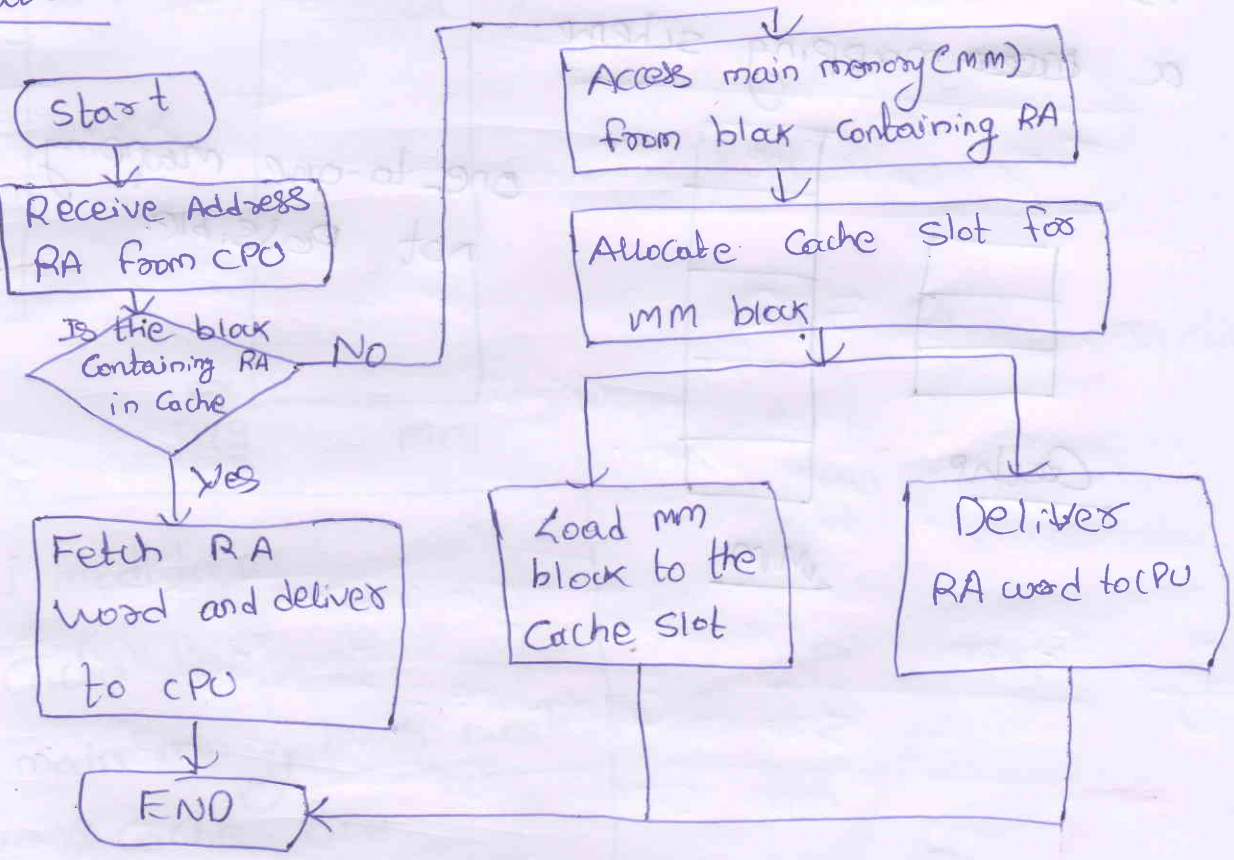


.) Processor wants to read a particular word. Address of word (101) is generated. eg:

.) Block is sent from main memory to cache memory and word is sent to the processor.

.) otherwise both the buffers are blocked in case of failure to search the word. and the word is in Cache memory

Cache operation :-



Important:-

We know that $|C| << |M|$

•) Locality of Reference :-

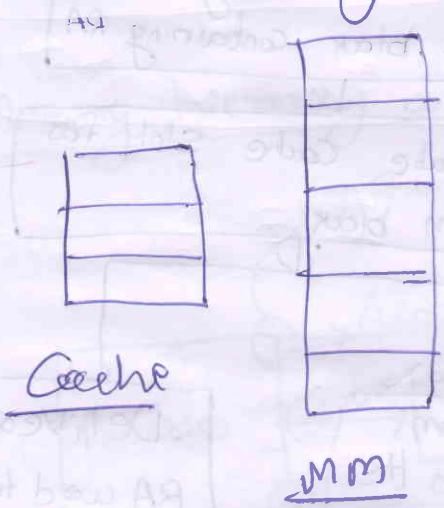
→ Temporal : Same type of locations may be accessed in looping

→ Spatial : 70 % of time, desired word is found in Cache as consecutive words are stored.

↳ If

Note: Block size in MM is same as Cache memory.

•) As there is no one-to-one mapping of blocks from MM to Cache memory, as Cache memory is smaller than MM therefore we should have a mapping scheme.



one-to-one mapping not possible.

1) Direct Mapping:

Let block size is 10 and no of words in MM = 1000

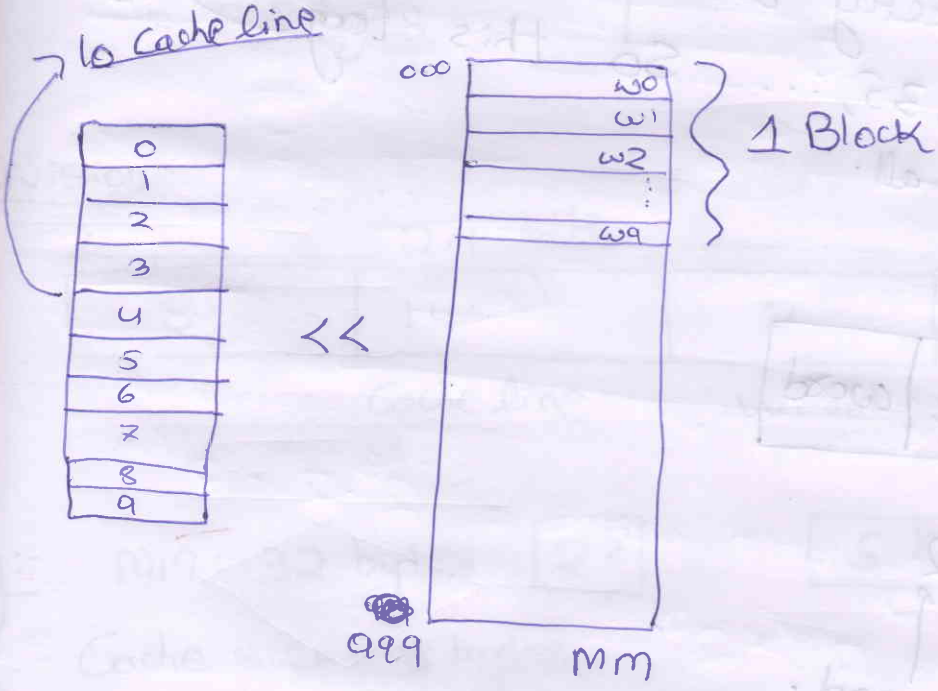
$$\# \text{ blocks} = \frac{1000}{10} = \boxed{100}$$

Let no of words in Cache = 100

$$\# \text{ Blocks in Cache Memory} = \frac{100}{10} = \boxed{10}$$

or Known as (to avoid confusion with that of MM)

Cache-line



☆)

i = j modulo m

i = Cache line

j: main memory block number

m = # Cache line

Let the Block no (in MM) = 99

Cache line selected: 99 % 10 = 9

Mapping schemes:-

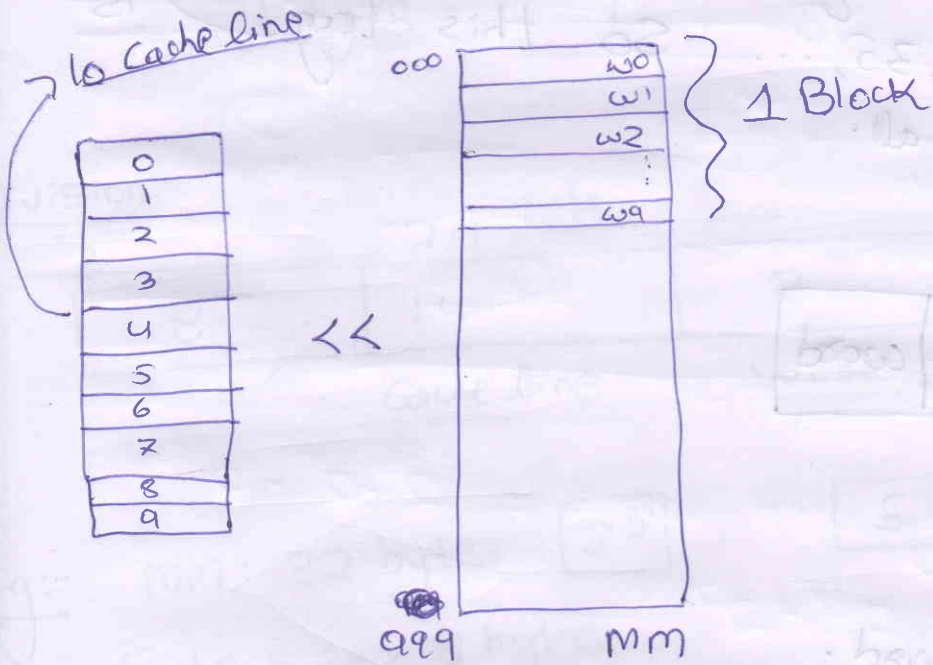
1) Direct Mapping:

Let block size is 10 and no of words in mm = 1000

blocks = $\frac{1000}{10}$ = 100

Let no of words in cache = 100

Blocks in Cache Memory (to avoid confusion with that of mm)
or
Known as Cache-line = $\frac{100}{10}$ = 10



*) i = j modulo m
i = Cache line
j = main memory block number
m = # Cache line

Let the Block no (in mm) = 99
Cache line selected: 99%10 = 9

Block in MM = 59

Cache line = 59 % 10 = 9

Now:- 5, 15, 25, 35, 45, 55, 65, 75, 85, 95

↓ All are mapped with Cache line = 5.

CPU generates address 652

Block : 65

Cache line : 5

But to know exactly we have to diff. from 65 is to other 25, 45, 35, ... So this tag is the difference in all.

Tag	Cache line	word

Above: 6 5 2

↑ ↑ ↑

tag, cache line, word.

Eg:- Given Cache line : 64 Kbytes (KB)

Size of Cache line : 4 bytes (4B)

MM size : 16 M bytes (MB)

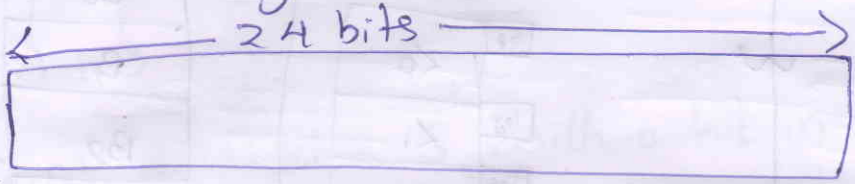
How do you divide main memory address if direct mapping used?

Sol:) Considering byte-addressable memory

16M bytes

$= 2^4 \cdot 2^{20}$ bytes

$= 2^{24}$ bytes (24 bits to



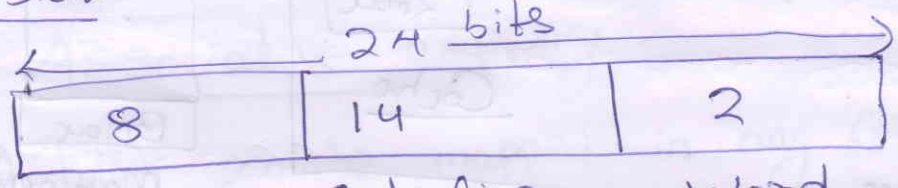
Cache = 64 Kbytes

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

$= 2^{16}$ bytes

Cache lines $= \frac{2^{16}}{2^2} = 2^{14}$

Division:-



Eg:- MM: 32 bytes : 2^5 — 5 bits memory

Cache size: 8 bytes

block = 4 bytes = 2^2 — 2 bits for block/word

No of cache lines: $\frac{8}{4} = \frac{2^3}{2^2} = 2^1$ — 1 bit for cache lin