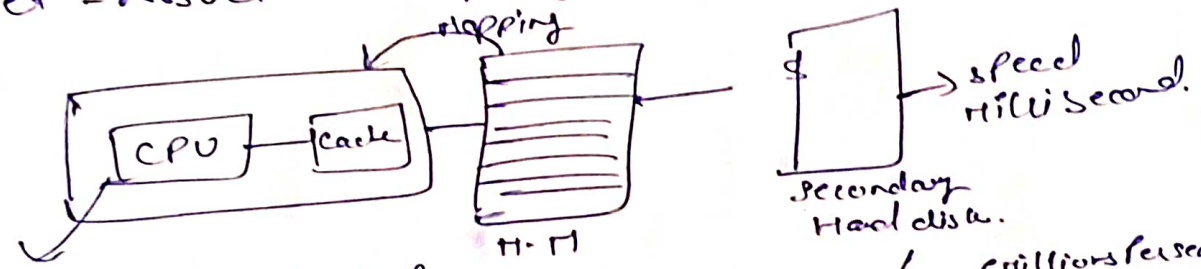
Cache mapping :-

is a technique by which content of main memory is brought into the cache memory.

→ It is a transformation of data from main memory to cache memory.

3 types :-

1) Associative mapping
2) Direct mapping
3) Set - Associative Mapping.



Work MIPS (Millions inst per second) its work IPS means (10^6 millions per second)

IPS = (10^9 inst per second)

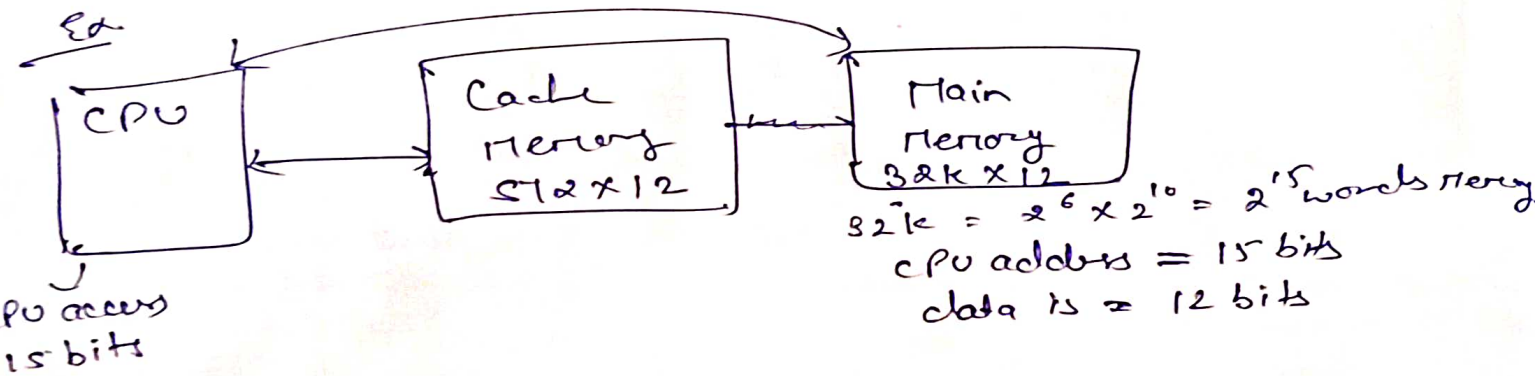
inst speed.

so Speed mismatch are both that why we create the diff levels of memory. the memory which speeds match with CPU speed its cache memory. CPU directly interact with cache memory. but cache memory get the data from main memory. it all called mapping.

cl one more process is Paging.

Paging is the process of transfer the data from secondary memory to main memory. called Paging.

M.M are divide diff no of blocks. and the blocks store 16 words. and these blocks transfer to cache. and CPU directly access to cache.

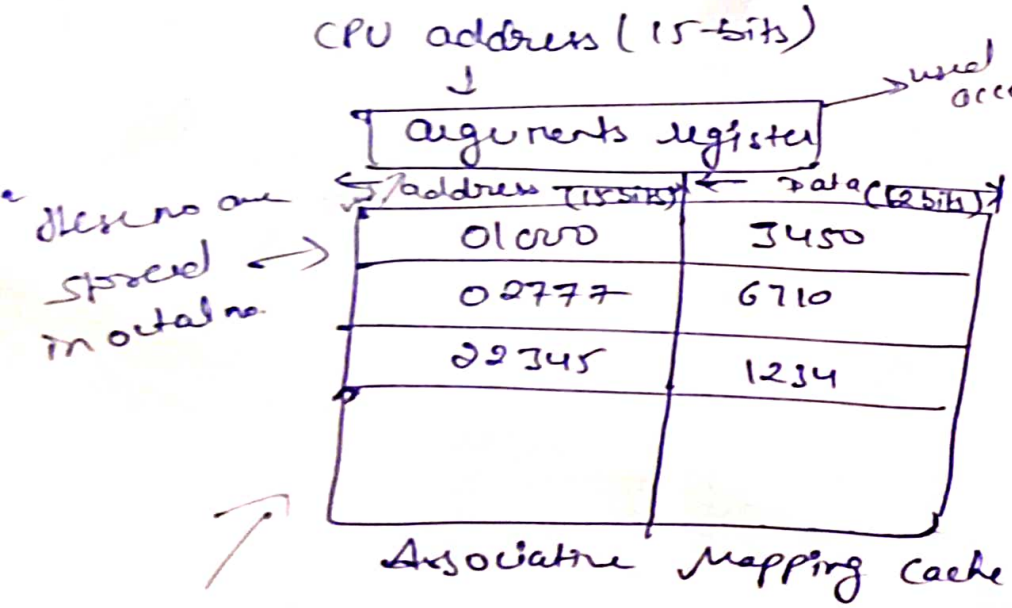


Associative Mapping :-

→ fastest & most flexible cache organization uses Associative Memory.

→ In associative memory mapping - cache are needs up of Associative Memory Associative Memory is used to other this address and ~~data~~ content (data) of the memory word.

→ It permits any location in cache to store any word from main memory it enables any word from main memory at any place is the cache memory (which does not happens in other mappings).



Associative Mapping Cache.

If place any address or data in address field, then address field must be empty. If this field is not empty then address paid replace with new address

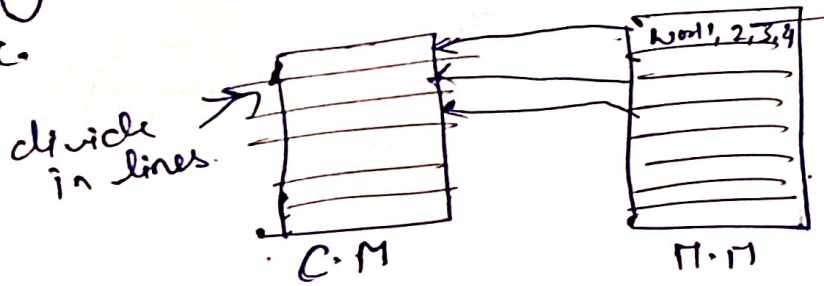
pair. ~~when~~ if the address field is already full. then it is used Replacement Policy for replace address pair with round Robin fashion. FIFO mostly used.

(2) ~~Set~~ & Direct Mapping :-

associative Memories are expensive compared to Random Access Memory. (bcoz of added matching logic ~~to~~ attached with each cell).

→ for Random Access memory Direct Mapping can be used.

→ Simplest technique: Direct Mapping → it maps each blocks of main memory into only one possible cache line. or it assigns each memory block to a specific line in the cache.

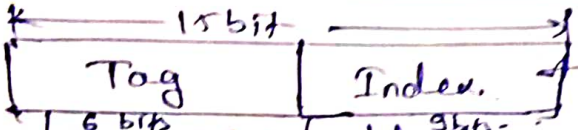


1 Block may contain 1 word to 16 words

128 words

→ CPU address or address space divided into

two Parts. CPU address divided into two parts

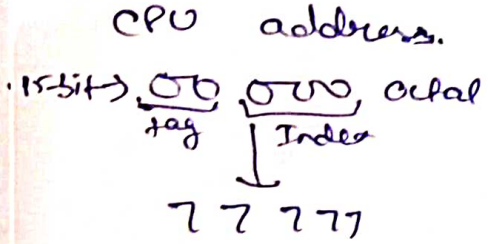


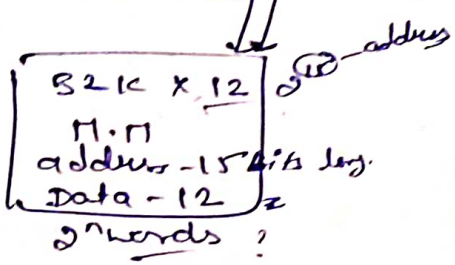
MBB

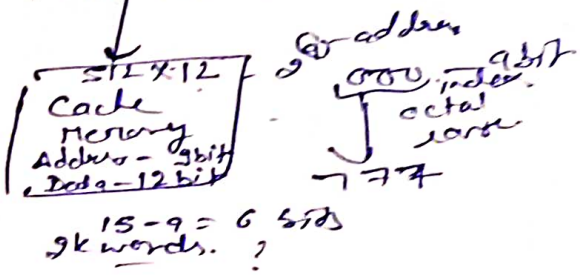
CPU address

access

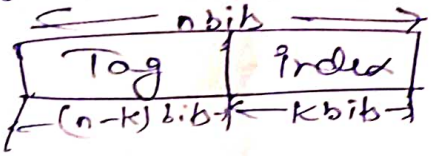
MM access CPU address then? MM access complete CPU address.





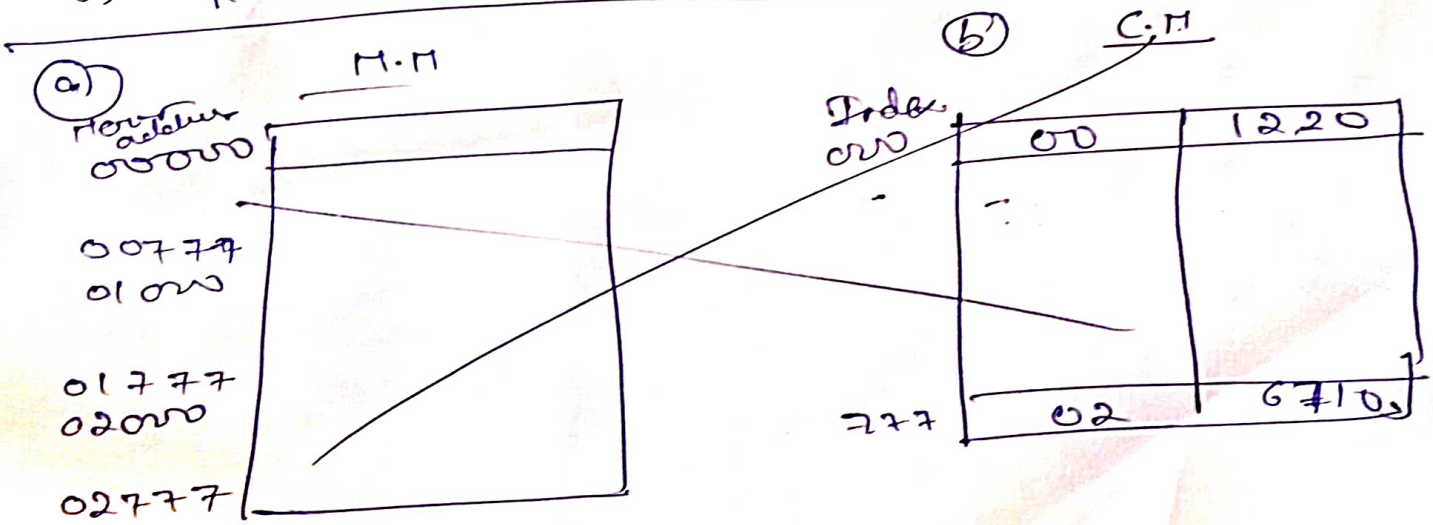


If M.M is 2^n words & CM is of 2^k words.



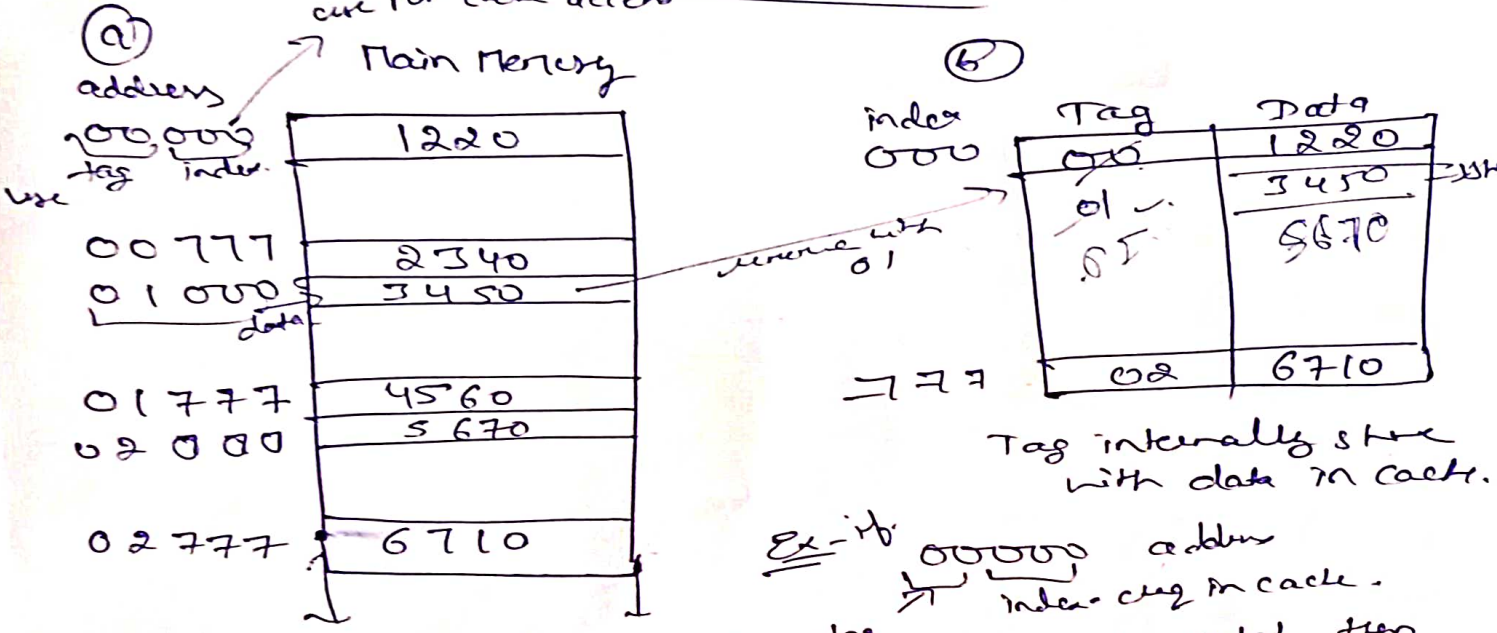
1) n bits address is used to access the M.M.

2) k-bits index is used to access the cache memory



[Bhawna_CSE@ngfcet.com
Manoj_cse@ngfcet.com]

Direct mapping internal organization :-



Ex- if 000000 address
index chg in cache.

① tag match if it is match then hit & read the data. Bcoz data in cache.

② if 01000 CPU access index go to cache. tag match but it is not present cache miss occur. in this case the CPU go to MM.

③ if 02000 CPU access. again miss again go to MM. then read MM data and transfer.

But Problem is ~~there~~ there. the Hit ratio is drop. bcoz tag is diff. and again & again Cache miss is occur and transfer the data from Main to Cache.

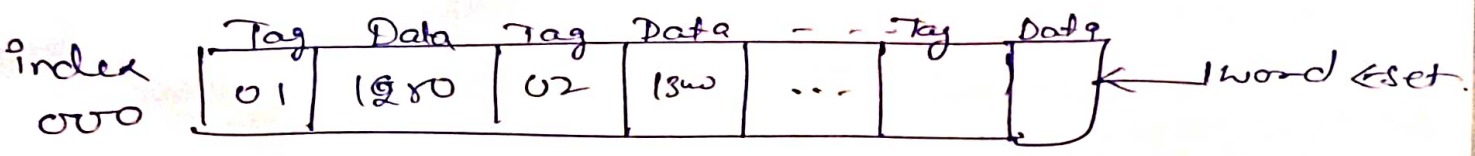
Set associative Memory :

→ Each word of a cache can store two or more words of memory under the same index address, create a set.

→ Each data word is stored together with its tag.

→ The no of tag - data item in one word is said to form a set.

→ set associative Mapping Combines Direct Mapping & associative Mapping.



→ Improved form of direct Mapping, where drawbacks of direct mapping is removed.

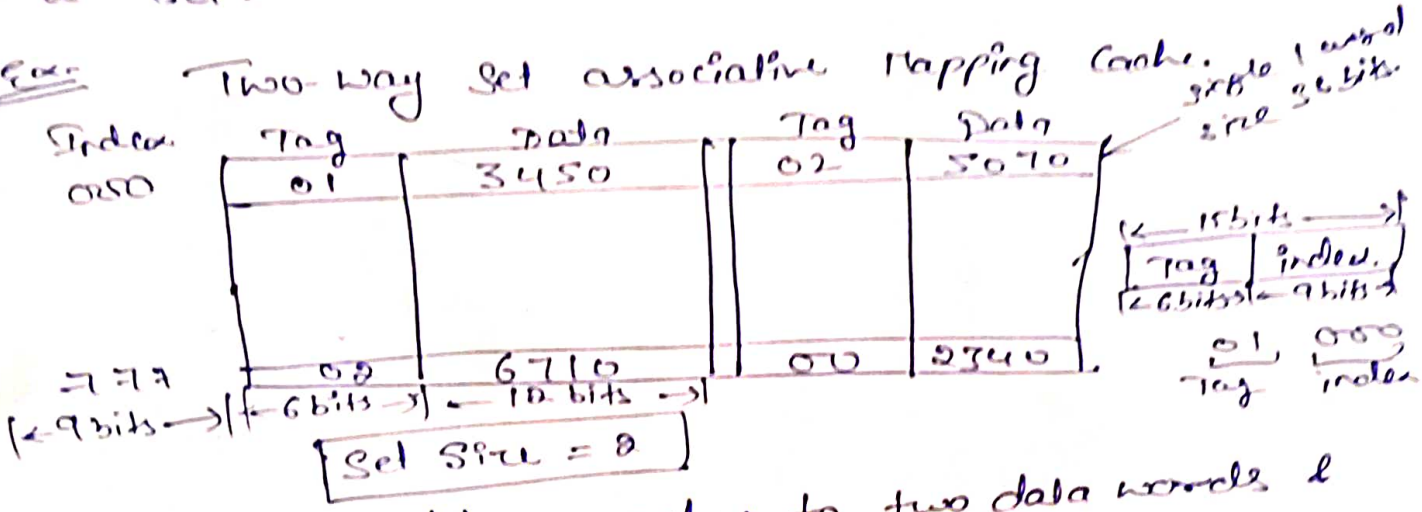
→ Drawback of Direct Mapping! Two words with the same index in their address but with different tag values cannot reside in cache memory at the same time.

Ex- 01 000 index 000
02 000

tag	data
01	1200
02	8500

In Set associative Mapping → each word of a cache can store two or more words of memory under the same index address. Creating a Set.

Ex: Two-way Set associative Mapping Cache.



Each index address refers to two data words & their associated tag.

✓ Tag = 6 bits & Data = 12 bits

Word length = 2 (6 + 12) = 36 bits

✓ Index 9 bits

2^9 = 512 words. → cache memory.

✓ Cache memory size = 512 × 36 = 18432 ✓

It can accomodate [1024] words of main memory. Since each words of cache contains two data words (512 × 2) = 1024 words.

In General of Set associative Cache will accomodate K words of RM in each words of cache.