Cache Memory:

Locality of Reference - Analysis of large number of typical programs has shown that the references to memory at any given interval of time tend to be confined with in a few localized areas in memory. This phenomena is refer as locality of reference.

Cache Memory - If the active portion of the program and data are placed in fast small memory, the average memory access time can be reduced, thus reducing the total execution time of the program. Such a fast small memory is refered as cache memory.

Hit Ratio - The performance of cache memory is frequently measured in terms of quantity called hit ratio.

- When the CPU refers to memory and finds the word in cache, it is said to produce a hit.
- If the word is not found in cache, it is in main memory and it counts as a miss.

The ratio of the number of hits divided by the total CPU references to memory is the hit ratio.

Types of Mapping

- The transformation of data from main memory to cache memory is refered as mapping process.
- There types of mapping.
  1) Associative Mapping
  2) Direct mapping
  3) Set - Associative Mapping

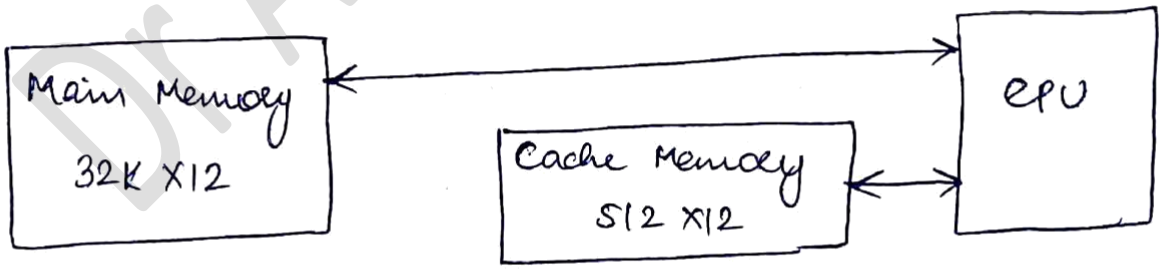
~~Associative Mapping~~ :-

To understand mapping procedure, consider an example

→ The main memory can store 32 K words of 12 bits each.

→ The cache is capable of storing 512 of these words at any given time.

$32\ K = 2^5 \cdot 2^{10} \rightarrow 2^{15}$



→ For every word stored in cache, there is a duplicate copy in main memory.

→ The CPU communicates with both memories. It sends 15 bit address to cache.

Hit - CPU accepts data from cache.

Miss - CPU read the word from main memory.

1) Associative Mapping

CPU Address (15 bits)

↓

Argument Register

Address	Data
01000	3450
02777	6710
22345	1234

5×3 = 15 → Address
4×3 = 12 bit data

→ It stores both address and content of memory word.

→ The address value of 15 bit is shown as five-digit octal number and its corresponding 12 bit word as four digit octal number.

→ A CPU address of 15 bit is placed in the argument register and associate memory is search for matching process.

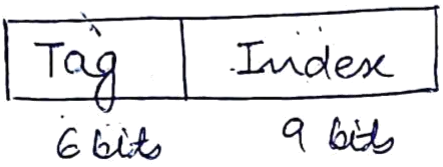
If the address is found, the 12 bit data is read and sent to CPU.

If no match occur, the main memory is accessed for the word.

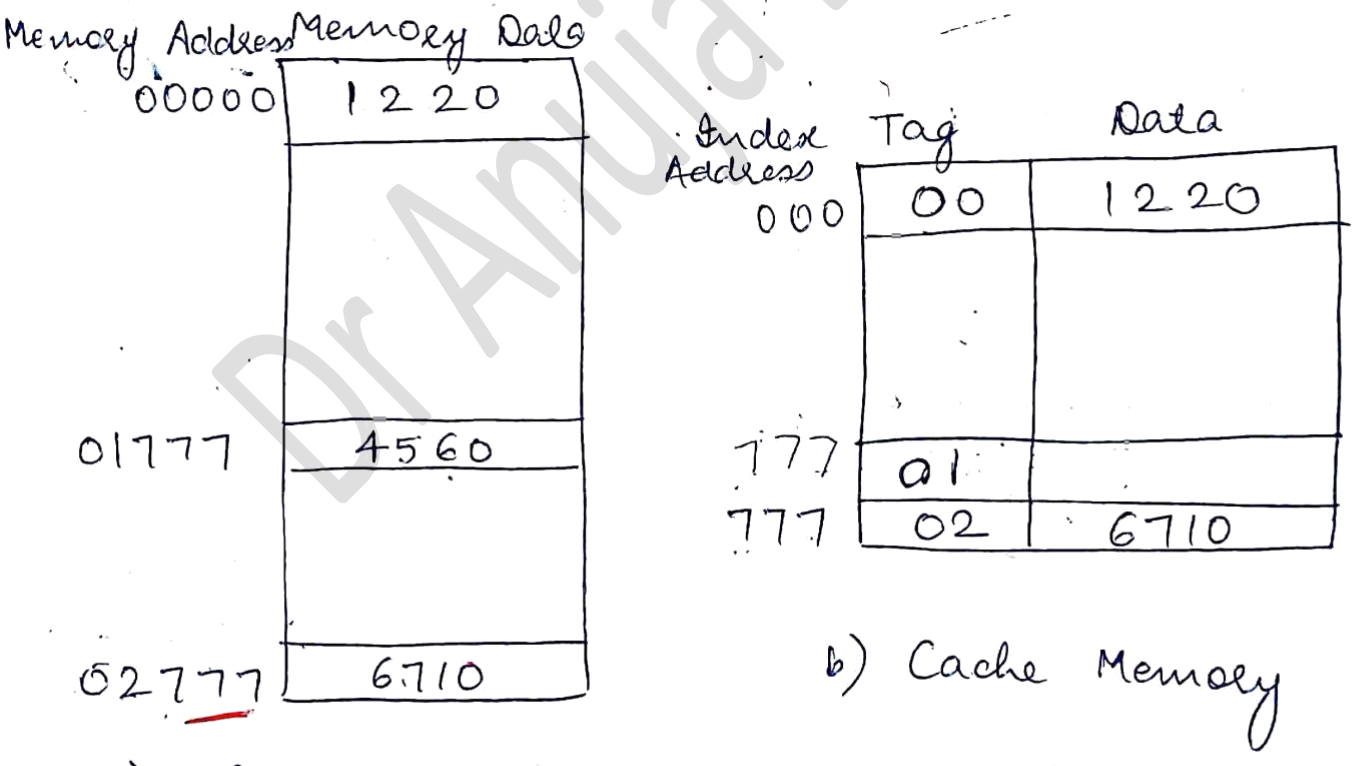
The address data pair is then transfered to the associative cache memory.

2) Direct Mapping

- Associative mapping is expensive compared to RAM because of added logic associated with each cell.
- The CPU address of 15 bits is divided into two fields. The nine least significant bits constitute the index field and the remaining six bits form tag field.



- The internal organization of the words in cache memory is as shown



a) Main Memory

b) Cache Memory

Fig :- Direct Mapping Cache Organization

- Each word in cache consist of data word and its associated tag.
- When a new word is associated with cache, the tag bit are stored along the data bits.
- When the CPU generates memory request, the index field is used for the address to access the cache
- The tag field of CPU address is compared with the tag in the word read from cache. If there is no match, there is a miss and required word is read from main memory.
- Disadvantage- Hit ratio drops, if two or more address have same index but different tags are accessed repeatly.

3) Set Associative Mapping

It is an improvement over direct mapping organization in that each word of cache can store two or more words of memory under the same index address.

Each data word is stored together with its tag and the number of tag data items in one word of cache is said to form a set.

Index	Tag	Data	Tag	Data
000	01	3450	02	5670
777	02	6710	00	2340

Fig :- Two way Set - Associative mapping cache

- Each index address refers to two data words and their associated tags.
- Each tag requires six bits and each data word has 12 bits, so the

word length = 2 (6 + 12)

= 36 bits

- An index address can accomodate 512 words. Thus size of cache memory is 512 X 36
- It can accommodate 1024 words of main memory since each word of cache contains two data words.
- In fig, the word stored at addresses 01000, & 02000 of main memory are stored in cache memory at index 000.

- When the CPU generates a memory request, the index value of the address is used to access the cache.
- The tag field of the CPU address is then compared with both tags in the cache to determine if a match occurs.
- The hit ratio will improve as the set size increases because more words with the same index but different tags can reside in a cache.

Note :- When a miss occurs in set associative cache, it is necessary to replace one of the tag data items with a new value.

Q) A two way associate cache memory uses block of four words. The cache can accommodate a total of 2048 words from main memory. The main memory size is 128 K X 32. formulate all pertinent information required to construct the cache memory. What is the size of cache memory.

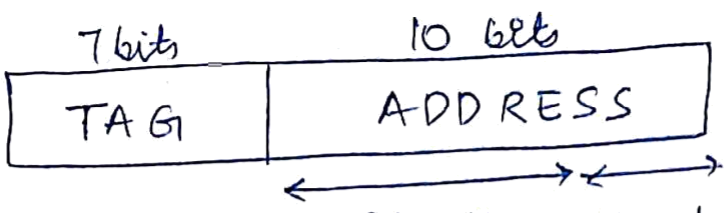
Soln:- ∴ Size of Main Memory = 128 K

∴ 128 K = 2^{17}

2048 ~~(from main memory~~ (

i.e ~~2048/2 = 1024 words of cache.~~

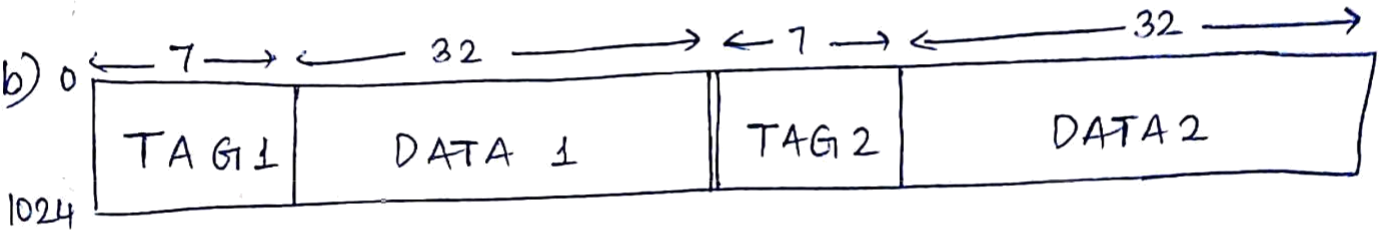
(a)



Memory Address (17 bit)		
Block Address (15)		Block Offset 2
7	8	2

1 Block of Cache = 4 words of RAM
$= 2^{2 \cdot 80}$

(b)



Size of Cache Memory = $1024 \times 2 (7+32)$

$= 1024 \times 78$

Q) The access time of a cache memory is 100 ns and that of main memory is 1000 ns. It is estimated that 80% of the memory request are for read and 20% for write. The hit ratio for read access only is 0.9. A write through procedure is used.

a) What is the average access time of the system considering only memory read cycles. b) What is the average access time of the system for both read & write requests.

c) What is the hit ratio considering the write request also

Solv.-

(a) Average Access time for Cache to read = 0.9 × 100

= 90 ns

Average Access time for cache & Memory to write = 0.1 × (100 + 1000)

= 110 ns

Total time taken = 90 + 110

= 200 ns

(b) Average access time for read = 0.8 × 200

= 160

Average access time for write = 0.2 × 1000

= 200

Total time taken = 160 + 200

= 360 ns

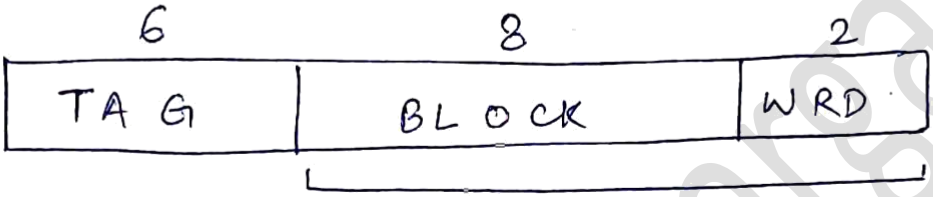
(c) Hit ratio = 0.8 × 0.9

= 0.72

Q. A computer has memory unit of 64K X 16 and cache memory of 1K words. The cache uses direct mapping with block size of four words (a) How many bits are there in the tag, index, block and words fields of the address format. (b) How many words are there in each word of cache, how are they divided into functions. Include a valid bit. (c) How many blocks can cache accomodate.

Soln.

(a)



Index = 10 Bit Cache Address

(b)

23 Bits in each word of cache

(c) 2^8 = 256 block of 4 words each.

$1K \cong \frac{1024}{4} = 256$ Block