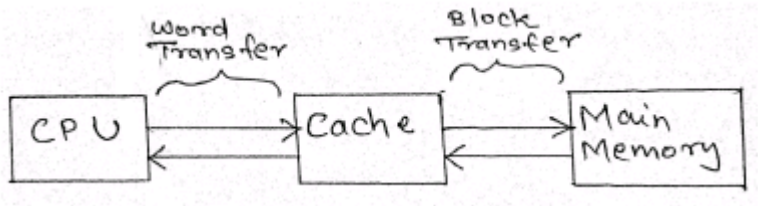
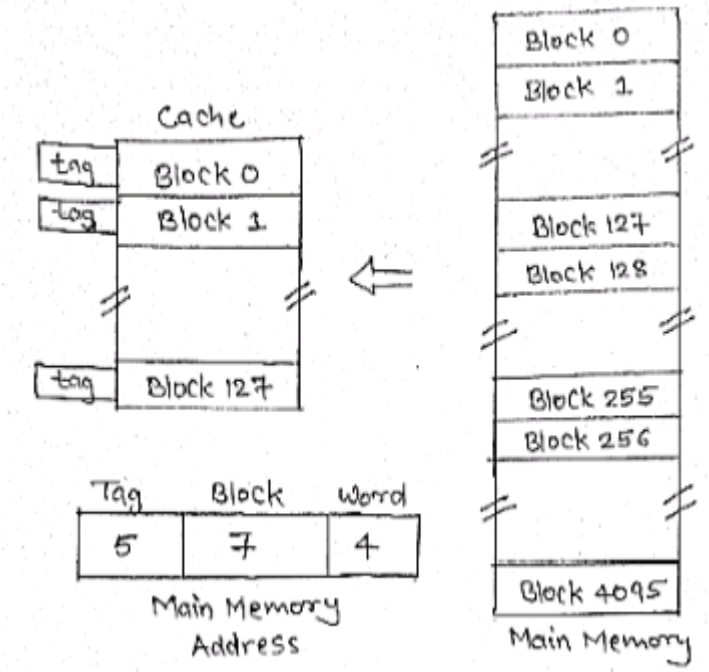
Explain Cache Memory and describe Cache mapping technique.

Cache Memory:-
1) Cache Memory is very high speed memory used to increase the speed of program by making current program & data available to the CPU at a rapid rate.
2) Access time to cache memory is less compared to main memory. It contains a copy of potions of the main memory.
3) When CPU attempts to read a word from main memory, check is made to determine if the word is in cache. It so, then word is delivered form cache.
4) If word is not there in cache then a block of main memory consisting some word along with that word, is read into cache and the required word is delivered to CPU. This is called “Principle of Locality of Reference”.
5) During a miss if there are no empty blocks in the cache, then some replacement policies such as FIFO, LRU, LFU, etc. are used.



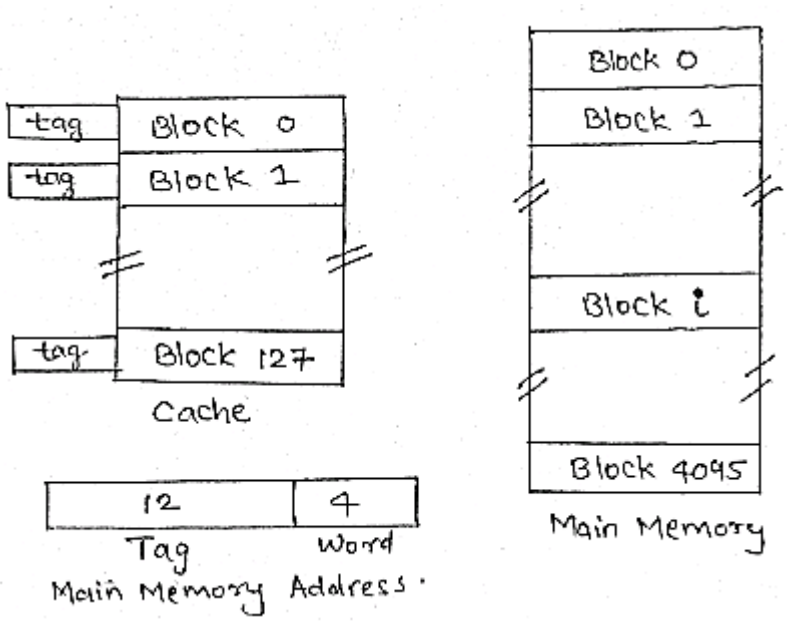
Cache Mapping Technique:-
The different Cache mapping technique are as follows:-
1) Direct Mapping
2) Associative Mapping
3) Set Associative Mapping
Consider a cache consisting of 128 blocks of 16 words each, for total of 2048(2K) works and assume that the main memory is addressable by 16 bit address. Main memory is 64K which will be viewed as 4K blocks of 16 works each.

(1) Direct Mapping:-



1) The simplest way to determine cache locations in which store Memory blocks is direct Mapping technique.
2) In this block J of the main memory maps on to block J modulo 128 of the cache. Thus main memory blocks 0,128,256,….is loaded into cache is stored at block 0. Block 1,129,257,….are stored at block 1 and so on.
3) Placement of a block in the cache is determined from memory address. Memory address is divided into 3 fields, the lower 4-bits selects one of the 16 words in a block.
4) When new block enters the cache, the 7-bit cache block field determines the cache positions in which this block must be stored.
5) The higher order 5-bits of the memory address of the block are stored in 5 tag bits associated with its location in cache. They identify which of the 32 blocks that are mapped into this cache position are currently resident in the cache.
6) It is easy to implement, but not Flexible

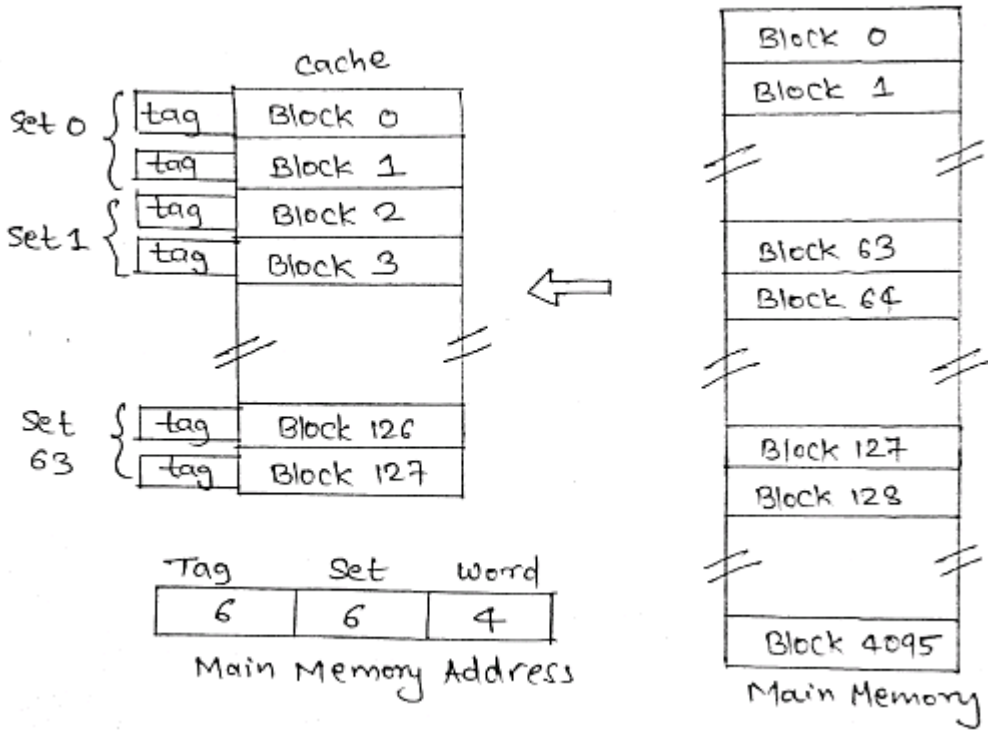
(2) Associative Mapping:-



1) This is more flexible mapping method, in which main memory block can be placed into any cache block position.
2) In this, 12 tag bits are required to identify a memory block when it is resident in the cache.
3) The tag bits of an address recevied from the processor are compared to the tag bits of each block of the cache to see, if the desired block is present. This is known as Associative Mapping technique.
4) Cost of an associated mapped cache is higher than the cost of direct-mapped because of the need to search all 128 tag patterns to determine whether a block is in cache. This is known as associative search.

(3) Set-Associated Mapping:-

1) It is the combination of direct and associative mapping technique.
2) Cache blocks are grouped into sets and mapping allow block of main memory reside into any block of a specific set. Hence contention problem of direct mapping is eased , at the same time , hardware cost is reduced by decreasing the size of associative search.
3) For a cache with two blocks per set. In this case, memory block 0, 64, 128,…..,4032 map into cache set 0 and they can occupy any two block within this set.
4) Having 64 sets means that the 6 bit set field of the address determines which set of the cache might contain the desired block. The tag bits of address must be associatively compared to the tags of the two blocks of the set to check if desired block is present. This is two way associative search.


Cache
Set 0
tag
Block 0
tag
Block 1
Set 1
tag
Block 2
tag
Block 3
Set 63
tag
Block 126
tag
Block 127
Tag
Set
Word
6
6
4
Main Memory Address
Block 0
Block 1
Block 63
Block 64
Block 127
Block 128
Block 4095
Main Memory