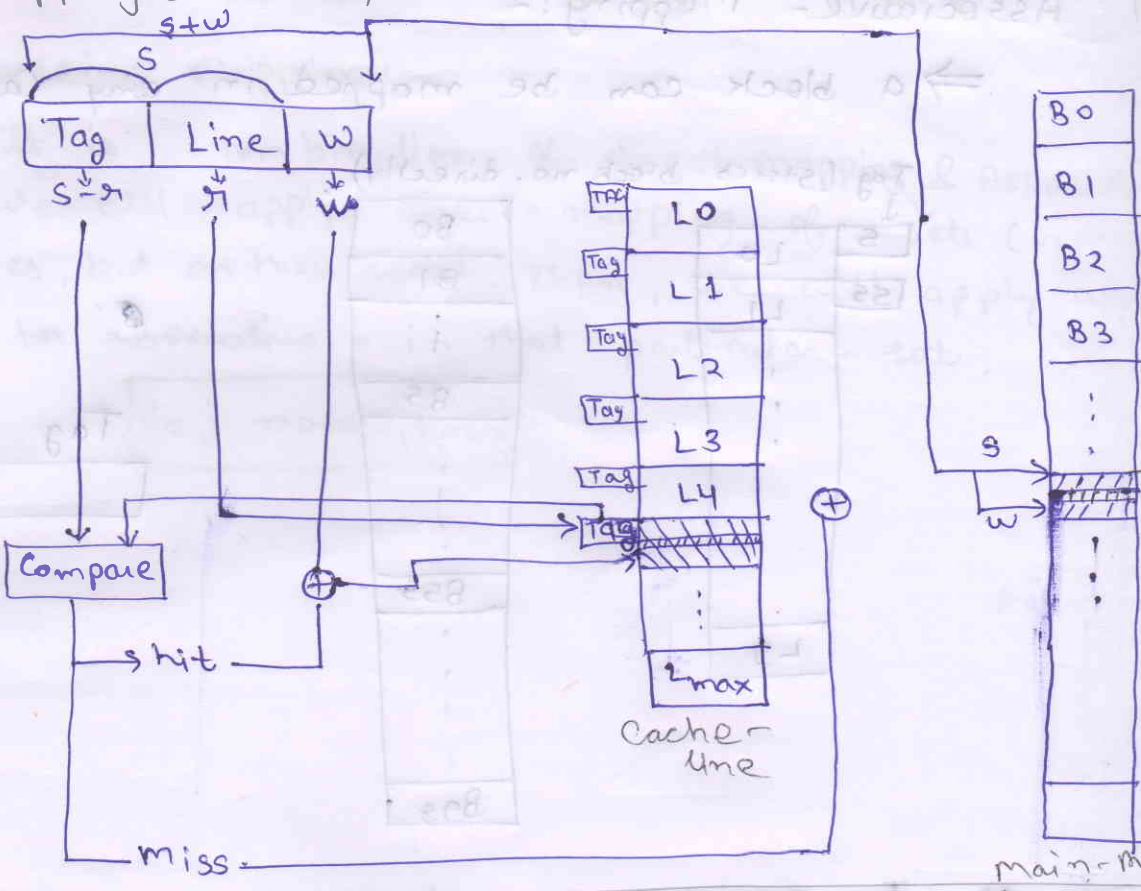
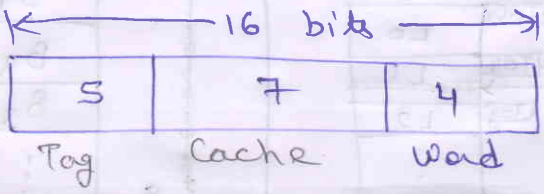
March 5th, 2020

① Direct - Mapping (Continued) :-



Eg :- Given
- Size of m.m. = 64K words = $2^6 \cdot 2^{10} = 2^{16}$
- # blocks in cache = 128 = 2^7 [= # Cache-lines]
- # Size of a block = 2^4 words.

How do you divide main-memory address into Tag, Cache & Word?



Disadvantage :-

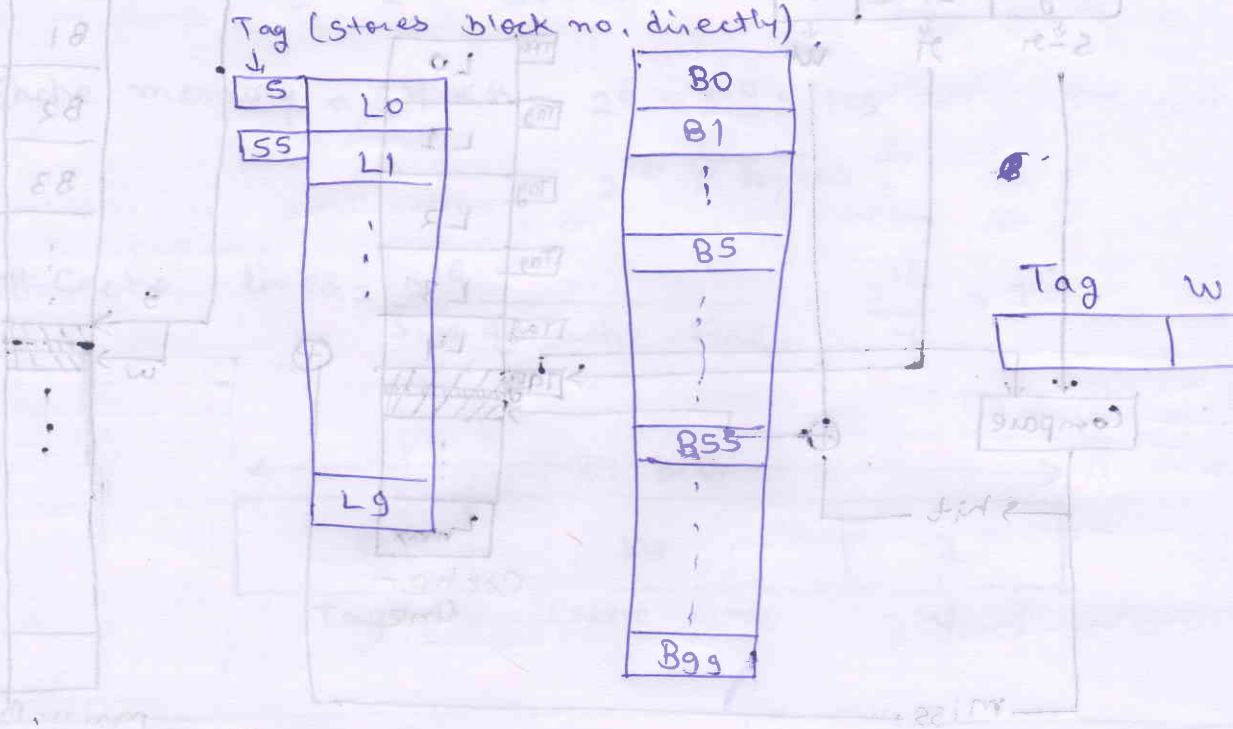
Each block in main-memory is mapped into a fixed cache-line.

Eg :- $C = A + B$ (5 → A, 55 → B) → Both A & B in same cache-line.

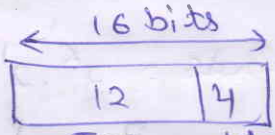
Here, Initially Block 5 is referred & brought in Cache-memory (having 10 cache-lines say). Then, similarly, B SS is referred for B, then brought in cache-memory. Init brought 5 will now be replaced by SS in cache-m Now, if we have access to B in Cache-memory, but not to A. So, These will be ~~mandatorily~~ miss

(2) Associative Mapping :-

⇒ a block can be mapped in any cache

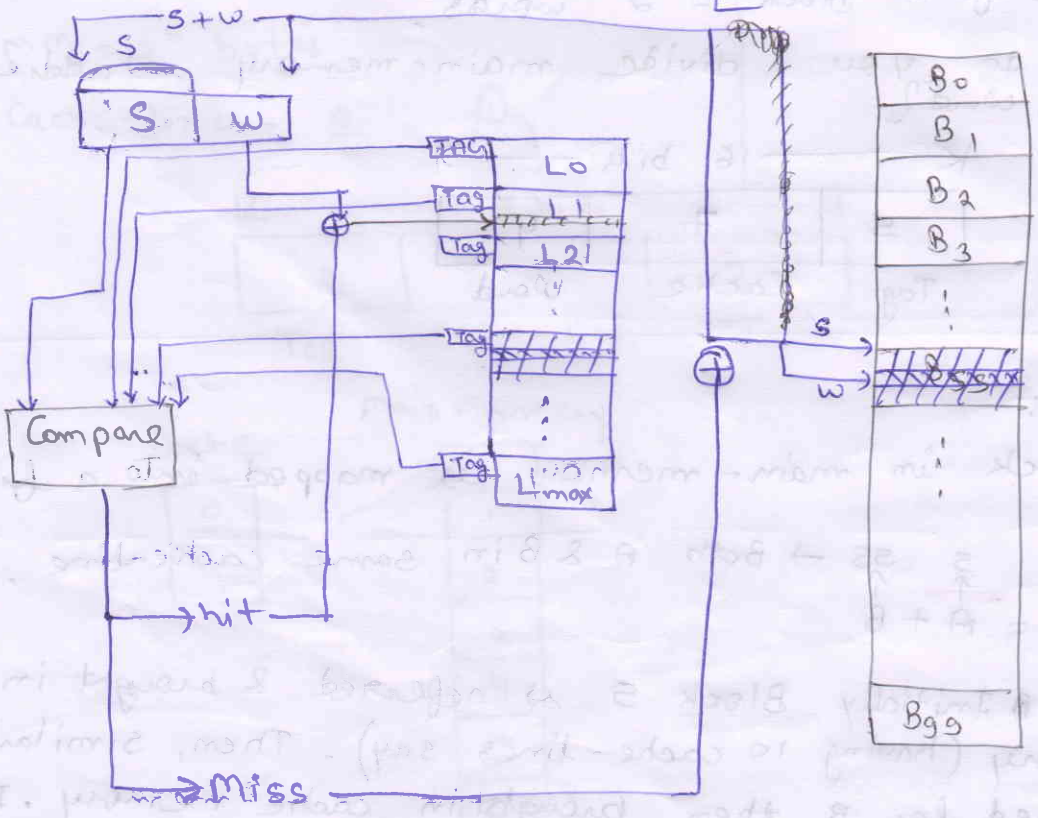


Eg:- In previous example, Answer will be:



∴ No. of Blocks = $\frac{m.m}{\text{Size}}$

$= \frac{2^{16}}{2^4}$



DRAWBACK : Complex circuit for tag comparision.

(3) Set-Associative Mapping :-

It is combination of direct-mapping & Associative Mapping. We will apply direct-mapping of sets (not on cache-lines, but on their sets). Then, we will apply associ mapping ~~in~~ ~~associative~~ in that particular-set.

$$S = j \text{ modulo } (\#set)$$