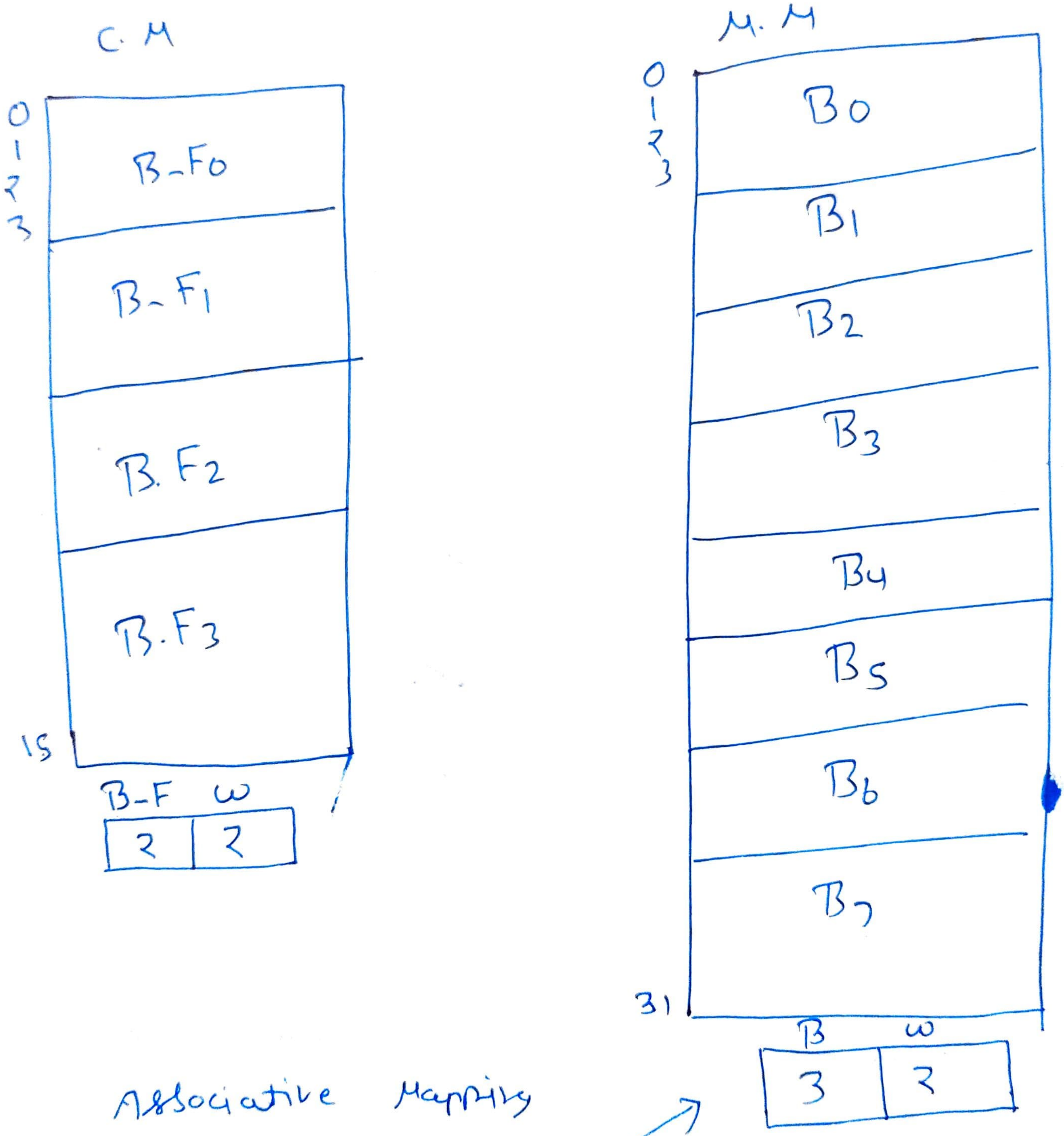
5 (Cache Memory).

Associative Mapping

→ Main memory blocks can be placed into any block_Frame.

→

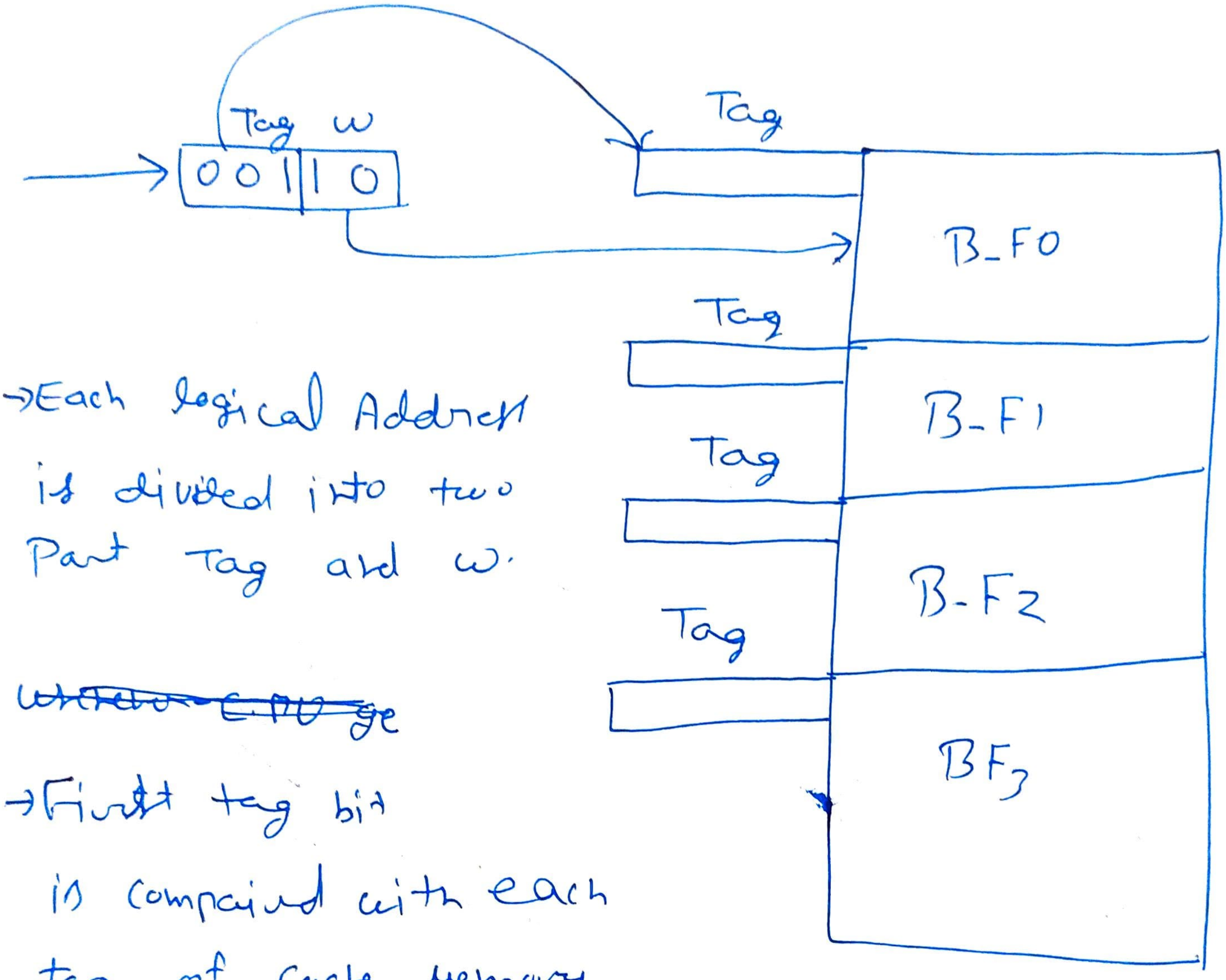


Associative Mapping

Tag	w
3	2

7 (Cache Memory)

6. C.M



→Each logical Address is divided into two Part Tag and w.

~~Whenever CPU~~ ge

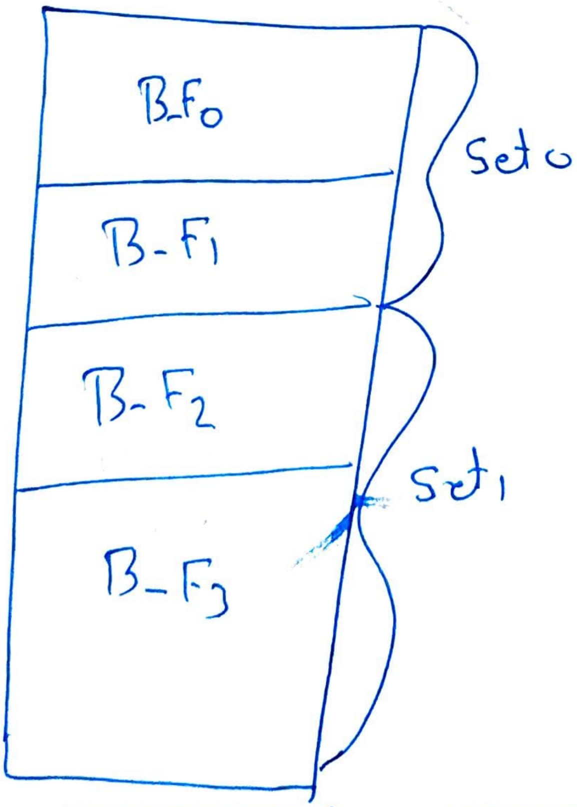
→First tag bit is compaired with each tag of cache memory. in parallal.

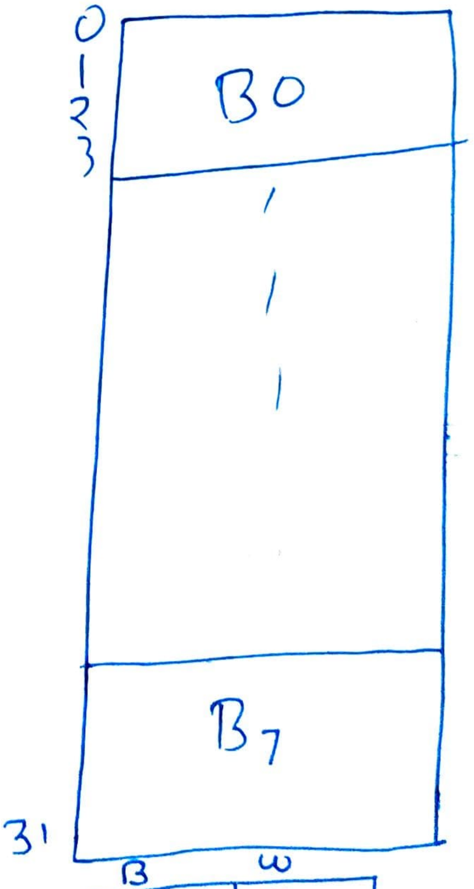
→ If there is a match, desired black (or address) is available in cache. Otherwise a cache miss is occure.

7 (Cache Memory)

Set associative

- → Combination of associative and direct mapping.
- → ~~Blocks~~ Block_Farms are grouped together ~~&~~ to farm sets.
- → It allows the mapping of a block to reside in any block_frame of a specific set.
- → Therefore, sets implement the direct mapping and within the set associative mapping is performed.





Set	BF	W

→

1	1	2

B	W
3	2

8- CC-MI

Mapping function

~~B~~ ~~i~~ ~~Block % #sets~~

$\boxed{\begin{array}{l} \text{Block}_j \longrightarrow \text{Set}_i \\ \text{iff} \\ \text{Block}_J \ \% \ \# \text{Sets} = i \end{array}}$

Tag	Set	Word
1	1	2

Word → 4 word

Set → 2 sets

Tag → Two group of main memory

~~B~~ Set_0 → 0, 2, 4, 6

0	0%2	= 0
2	2%2	= 0
4	4%2	= 0
6	6%2	= 0

Set_1 → 1, 3, 5, 7

1	1%2	= 1
3	3%2	= 1
5	5%2	= 1
7	7%2	= ɸ

9. (C. M)

Direct Mapping

$\text{Block}_j \longrightarrow \text{Block\_Frame}_i$

iff

$\text{Block}_j \ \% \ \#\ \text{Block\_Frames} = i$

Tag	B_F	w

Associative Mapping

$\text{Block}_J \xrightarrow{\text{Any}} \text{Block\_Frame}\ i$

Tag	w

Set Associative Mapping

$\text{Block}_j \longrightarrow$ B. Set i [Any available Block_Frame of Set i]

iff

$\text{Block}_j \ \% \ \#\ \text{Sets} = i$

10 (C-M)

Block_Frame Replacement

→ In any types of mapping, often we need to replace one block by another block.

→ To perform this task some specific algorithm is used such as LRU (Least recently used)

→ In this algorithm, the ~~blocks~~ ~~bloc~~ oldest block need to replaced by newest block.

→ Along with this information M.M.U also maintain one bit corresponding to each Block_Fram, dirty or modify bit. This bit indicates the modification of a block.

→ If this bit set to 1, two main memory operations are required

1- Transfer data from C.M to M.M

2- " " " M.M to C.M

→ If it is 0 only one memory operation is required

1- Transfer data from M.M to C.M