Replacement Algorithm (Hamacher)

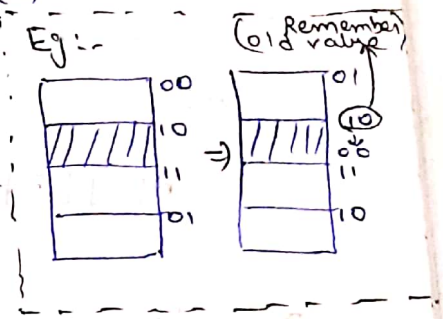
Least Recently Used (LRU):-

~~Replace a Cache-line (block)~~

Replace a Cache-line (block) which has not been referenced for longest period of time.

- Hit occurs, set the counter-value zero(0) for the newly referred block and increment counter-value for all those blocks which have counter-value less than the old value of the reffered page.

Eg:-



- Miss occurs, & Set is not full, counter-associated with just referred block is set to zero(0) and counters of all other blocks are incremented by 1.

- Miss Occcurs & set is full: Identify the block associated with highest counter-value (above 11 in 4-way Cache) & replace it, Subsequently, counter is set to zero(0) and Counters of all other blocks are incremented by one.

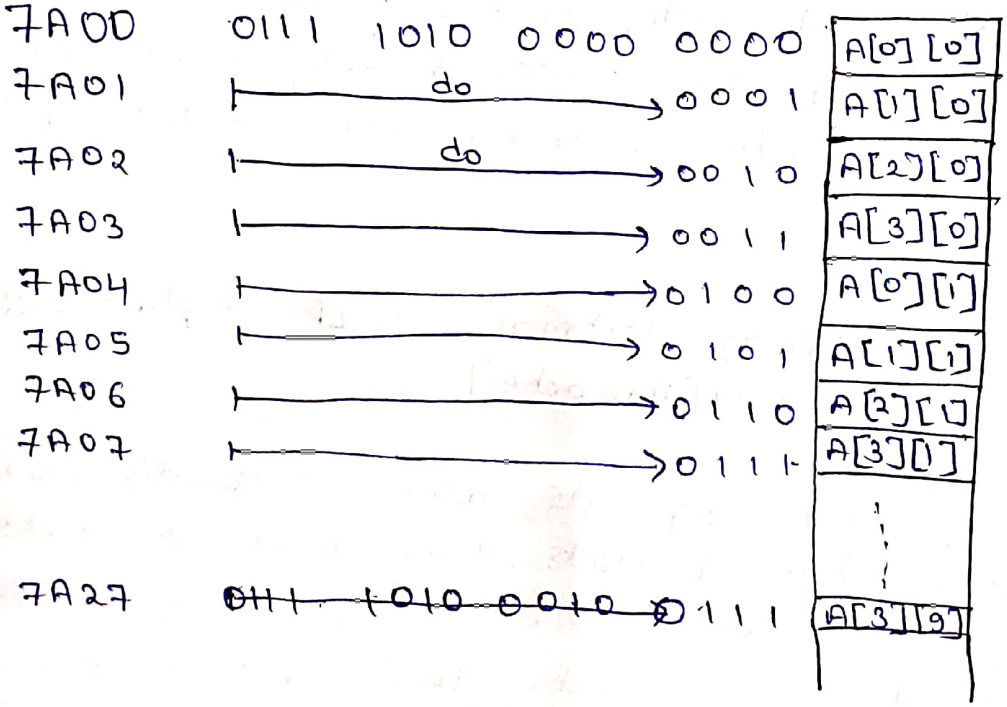
March 11, 2020

Example of Mapping Techniques & LRU Replacement Policy [Hamacher]

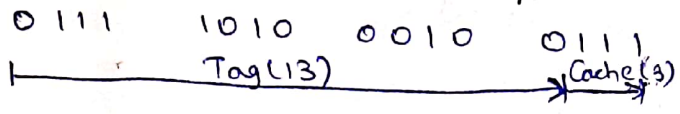
Problem:- A computer has separate data & instruction caches. The data cache has 8 blocks. Each block can hold a 1 word of 16 bit. Consider the program given in figure to identify the cache ~~test~~ content in various mapping schemes.

```
int A[4][10];
int SUM = 0;
for (j=0; j<10; j++)
    SUM = SUM + A[0][j];
AVG = SUM/10;
for (i=9; i>=0; i--)
    A[0][i] = A[0][i] / AVG;
```

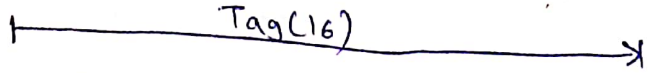
- Consider the Array A is organized in column order and first element A[0][0] is at location 7A00.
  Variables SUM, i, j are in processor's registers
  It is a set-associative mapping.



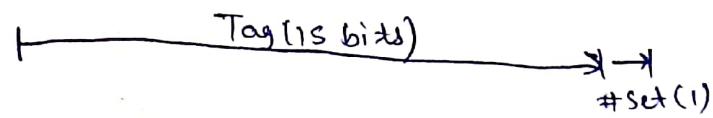
Direct Mapping :-



Associative Mapping :-



Set Associative Mapping :-



Direct-Mapping: Content of Cache after pass

Cache Block	j=1	j=3	j=5	j=7	i=8 j=9	i=6	i=4	i=2	i=0
0	A[0][0]	A[0][2]	A[0][4]	A[0][6]	A[0][8]	A[0][6]	A[0][4]	A[0][2]	A[0][0]
1									
2									
3									
4	A[0][1]	A[0][3]	A[0][5]	A[0][7]	A[0][9]	A[0][5]	A[0][5]	A[0][3]	A[0][1]
5									
6									
7									

Associative-Mapping ;-

Content of Cache after each pass

Cache Block	j=7	i=2 j=9	i=0
0	A[0][0]	A[0][8]	A[0][0]
1	A[0][1]	A[0][9]	A[0][1]
2	A[0][2]	A[0][2]	A[0][2]
3	A[0][3]	A[0][3]	A[0][3]
4	A[0][4]	A[0][4]	A[0][4]
5	A[0][5]	A[0][5]	A[0][5]
6	A[0][6]	A[0][6]	A[0][6]
7	A[0][7]	A[0][7]	A[0][7]

Set Associative Mapping:-

∵ Last-bit is zero in all addresses.

∴ All will go to set 0,

Cache Block		j = 3	j = 7	i = 7 j = 9	i = 2	i = 0
Set 0 ←	0	A[0][0]	A[0][4]	A[0][8]	A[0][4]	A[0][0]
	1	A[0][1]	A[0][5]	A[0][9]	A[0][5]	A[0][1]
	2	A[0][2]	A[0][6]	A[0][6]	A[0][2]	
	3	A[0][3]	A[0][7]	A[0][7]	A[0][3]	
Set 1 ←	4					
	5					
	6					
	7					