LRU Replacement Policy

Cache replacement policies are used to determine the block to be evicted from a set in a conflict. LRU (Least Recently Used) policy is one of the cache replacement policies. In the LRU replacement policy, least recently used block in a set is selected to be replaced.

Question: How many bits are needed to implement LRU replacement policy in caches?

Answer: $\lceil \log_2 N \rceil$ bits are needed for each block, where N is the associativity of cache.

Proof: To implement the LRU policy, each block in a set is assigned with a unique index determining the order of usage (i.e., age). The ages are updated in each memory reference and the oldest block is replaced in a set conflict. The age section must be large enough to provide a unique age to all N blocks in a set. k bits can represent 2^k different numbers. Hence, the number of age bits = a must be selected so that $2^a \geq N$. Taking logarithm yields $a \geq log_2 N$. Hence, at least $\lceil \log_2 N \rceil$ age bits are required. Note that the age of each block must be tracked, so $\lceil \log_2 N \rceil$ bits are required for each block.

Example: For this example, a 4-word fully associative cache is used, with block size of 1 word. Number of age bits are $\lceil \log_2 4 \rceil = 2$. Bigger age means less recently used block (i.e., 00 is the most recently used and 11 is the least recently used). The main memory is 16 words and is word addressable. The cache uses LRU replacement policy. If there are multiple empty blocks, the data must be stored in the left-most block.

Assuming that the cache is initially empty, for the reference string 1, 2, 3, 4, 2, 5, 3, 6, 2, 7, Give the contents of all Age and Tag fields after each reference. The Age field should be given in binary, and the Tag field values should be given in hexadecimal.

Solution:

Reference to 1: Cache miss, data placed into left-most empty block.

Age	Tag		Age	Tag		Age	Tag		Age	Tag
00	1									

Reference to 2: Cache miss, data placed into left-most empty block.

Age	Tag		Age	Tag		Age	Tag		Age	Tag
01	1		00	2						

Reference to 3: Cache miss, data placed into left-most empty block.

Age	Tag		Age	Tag		Age	Tag		Age	Tag
10	1		01	2		00	3			

Reference to 4: Cache miss, data placed into left-most empty block.

Age	Tag	Age	Tag	Age	Tag	Age	Tag
11	1	10	2	01	3	00	4

Reference to 2: Cache hit, ages updated.

Age	Tag	Age	Tag	Age	Tag	Age	Tag
11	1	00	2	10	3	01	4

Reference to 5: Cache miss, block 1 is replaced. Ages updated.

Age	Tag	Age	Tag	Age	Tag	Age	Tag
00	5	01	2	11	3	10	4

Reference to 3: Cache hit, ages updated.

Age	Tag	Age	Tag	Age	Tag	Age	Tag
01	5	10	2	00	3	11	4

Reference to 6: Cache miss, block 4 is replaced. Ages updated.

Age	Tag	Age	Tag	Age	Tag	Age	Tag
10	5	11	2	01	3	00	6

Reference to 2: Cache hit, ages updated.

Age	Tag	Age	Tag	Age	Tag	Age	Tag
11	5	00	2	10	3	01	6

Reference to 7: Cache miss, block 1 is replaced. Ages updated.

Age	Tag	Age	Tag	Age	Tag	Age	Tag
00	7	01	2	11	3	10	6