In class, you have studied the organization of cache. Recall from lecture that set-associativity is a compromise between a direct-mapped cache and a fully-associative cache where each memory location maps to a set of cache locations. So, an *N-way set associative* cache has N cache locations in each set. If the address has m bits and the cache has 2^s sets with blocks of 2^n bytes, then the address splits up as follows:

Tag = m-s-n	Index = s	Block Offset = n

Another issue is what to do when each set fills up. Temporal Locality tells us that the data which has not been accessed for longest is least likely to be needed, so we use the replacement policy called *least recently used*, or LRU, that is we assume the data we should will replace is the least recently used data.

Now we will practice these concepts by doing a few exercises.

Problems

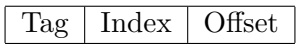
1. (a) The L1 data cache of the AMD Barcelona is a 64KB, 2-way set-associative cache with 64-byte blocks. The Barcelona supports 40-bit physical addresses (i.e., can address 1TB of memory).

   Compute the number of sets and the size of the tag, index, and block offset fields.

(b) The L2 data cache of the Intel Core 2 Duo is a 2MB, 8-way set-associative cache with 64-byte blocks. The Core 2 Duo supports 36- bit physical addresses (i.e., can address 64GB of memory).

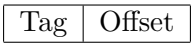
Compute the number of sets and the size of the tag, index, and block offset fields.

2. (a) **Cache 1:** Given a *direct-mapped* cache with 2 blocks of 2 bytes each, where the address is broken as follows: If the cache was initially empty and the addresses below



were accessed in that order, which of the following would be cache misses?
0110 0010 0110 1100 0111 1001 0110 1100 1101

(b) **Cache 2:** Given a fully associative cache (using a least-recently-used replacement policy) with 3 blocks of 2 bytes each, where the address is broken as follows:



If the cache was initially empty and the addresses below were accessed in that order, which of the following would be cache misses?
0110 1101 0111 1001 1110 1100 0110 1111 1101

3. Give an access sequence for which Cache 1 has strictly more hits than Cache 2, or argue that such a sequence cannot exist.

4. For this question, you are given a 16-byte cache (initially empty) and the sequence of memory accesses (byte loads) shown in the table. For each of the following cache configurations explain whether it can produce this sequence.

address	hit/miss
2	miss
14	miss
0	hit
17	miss
12	hit
3	hit

(a) 4-byte blocks, 2-way set-associative (e.g., 2 sets of 2 blocks each)

(b) 4-byte blocks, direct-mapped (4 blocks)

(c) 8-byte blocks, direct-mapped (2 blocks)

(d) 4-byte blocks, fully associative (4 blocks)