Problem 3 (30 points)

A byte-addressable, write-back cache of *fixed total size* and *fixed cache line (a.k.a., block) size* is implemented as both a direct mapped cache and also as an N-way set-associative cache. In both cases, we will assume the cache is initially empty.

First, consider the cache organized as a direct mapped cache. The following sequence of 11 accesses generates the hits/misses shown. Some miss/hit entries are intentionally left blank.

Address	Read/ Write	Direct Mapped (Hit/Miss)
0100001010	R	
1100100111	R	Miss
1110101000	R	Miss
0011000101	R	
0110111100	R	
1010110101	R	Miss
1100100000	R	Miss
0100001111	R	Hit
0101111111	W	Miss
0110110100	R	
0110100101	R	Miss

Note high order bits are the same but still miss.

Hit because of this miss

* The hit must be because of the first access because only the first address has the same high order bits as the hit address.
* Comparing the hit address and the first address, we know that offset is at least 3 bits.
* From the last two misses we that the offset is at most 4 bits.
* The miss immediately before the hit and the second miss have the same first 7 bits. So there must be an access in-between that evicts the line brought in by the second miss.
* There also could not be a miss that evicts the line brought in by the first miss for the hit to be a hit.
* All these could only be possible if offset is 4, index is 3.

Part a (4 points): How many cache lines does each set have in a direct mapped cache?

1

Part b (2 points): What is the cache line (a.k.a., block) size?

16 Bytes

Part c (2 points): What are the number of index bits for the direct mapped cache?

3

Now consider the cache organized as a N-way set-associative cache, with the same total size and same cache line size as before. The total size of "overhead" for this N-way set associative cache is 112 bits. Assume that in this particular cache, overhead in *each* cache line includes tag bits and 10 additional bits for bookkeeping (e.g., the valid bit, modified bit, LRU bits) that do not affect this problem. We have expanded the table to show the hit/misses for the same sequence of accesses when the cache is organized as an N-way set-associative cache.

Address	Read/ Write	Direct Mapped (Hit/Miss)	N-way associative (Miss/Hit)
0100001010	R	**Miss**	**Miss**
1100100111	R	Miss	Miss
1110101000	R	Miss	**Miss**
0011000101	R	**Miss**	Miss
0110111100	R	**Miss**	Miss
1010110101	R	Miss	**Miss**
1100100000	R	Miss	**Hit**
0100001111	R	Hit	**Hit**
0101111111	W	Miss	Miss
0110110100	R	**Miss**	Hit
0110100101	R	Miss	**Miss**

* T bits for tag, I bits for set index; we have (10+T) * 2^I * N = 112
* Since the cache line size is the same as before, the offset must be 4, and so we have T + I = 10 - 4 = 6.
* 10<=10+T<=16. only 14 and 16 are divisible by 112. So T is either 4 or 6.
* If T is 6, then 2^I * N = 7, so I = 0 and N = 7. If T is 4 then 2^I * N = 8, so I = 2 and N = 2 or I = 1 and N = 4.
* Since the total size is fixed and the cache line size is fixed, the total number of cache lines 2^I * N must be 8, same as before.
* So I must be 2 and N must be 2.

Part d (4 points): What is N?

2

Part e (4 points): What is the number of index bits for the N-Way set associative cache?

2

Part f (4 points): Is this a write-allocate cache?

No

Part g (10 points; 1 point per blank): Please complete the second table above by filling in "Hit" or "Miss" for each of the blank entries. "I Don't Know" is accepted on a per blank basis.