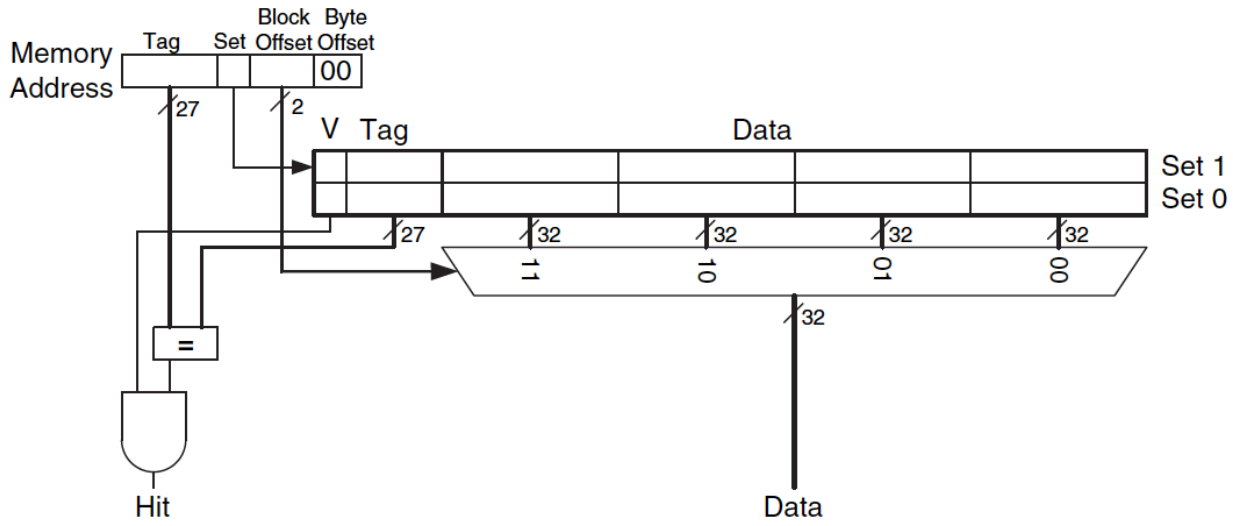

Assume we are working with the direct-mapped cache setup above.

1. Where is the address 0x0001309C mapped? (Give both the set and block). What other addresses are read in at the same time?

2. Give two addresses that will conflict with the address from (1). Choose at least one of them to have a *different* block offset.

3. Assume the cache is initially empty, but it updates after each read. Are each of the following memory lookups hits or misses?

Memory address	Hit/miss?
0x00001308	
0x00001304	
0x0000130C	
0x00001310	
0x00001304	
0x00001320	
0x00001308	
0x00001318	
0x00001338	
0x00001300	

V	Tag	Data
1	00…010	X
1	00…000	X

4. Given the cache above, what memory addresses are currently being stored?

5. If the size of the cache from (1) were doubled, how many bits would be needed for the set field? The tag?

6. If a direct-mapped cache holds four words per block and the tag field is 17 bits long, how many sets are there in the cache? How much data can be stored in the cache?

7. Order the following cache setups from the one that will result in the *most conflicts* to the one that will result in the *fewest conflicts* for a cache of a fixed size. (This is true in general – you can come up with unusual examples where the order is not the same.)
    a. Direct-mapped cache with a block size of 1 word
    b. Direct-mapped cache with a block size of 4 words
    c. 2-way set associative cache with a block size of 1 word