CSCI 4335
Dr. Abraham

Problem 1:
Given a RAM size of 512KB with a block size of 128 bytes each, and a cache size of 8KB, 4-way set associate cache. Calculate Tag, Index (**Sets** of lines) and offset.

So 512 kb of ram is 524288 bytes of ram, if we use the conversion of 1 kb = 1024 bytes. So 2^19 gives us 524288 bytes of ram, so to address it we need 19 bits.
So we have a block size of 128 bytes each. That means we need 7 bits to indicate byte offset, because 2^7 or 2 to the seventh power gives us 128.
So we have a cache size of 8 kb. So 2^13, which is our cache size, divided by 2^7, which was our block size, gives us 2^6. 2^6 is 64, which is the amount of lines we have.
However, since it is a 4 way set associate cache, and 2^2=4, our final index is 2^6-2^2=2^4
Finally we can just subtract like so: 19-7-4=8. So our tag is 8 bits . This is because we used 7 bits for the offset, 4 bits for the cache, and we need 19 to address the entire thing.

Problem 2:
Given a RAM size of 64MB, and a 2-way associative 8KB cache with 16 byte block size, calculate Tag, Set and Offset

So we have 64 mb of ram, which is 33554432 bytes of ram, since 1 kb = 1024 bytes and 1 mb is 1024 kb. Since 2^25 is 33554432, we need 26 bits to address this.
So we have a block size of 16 bytes. That means we need 4 bytes to indicate byte offset, because 2^4 = 16.
So we have a cache size of 8 kb. So 2^13, which is our cache size, divided by 2^4, which was our block size, gives us 2^9. 2^9 is 512, which is the amount of lines we have. However, we have a 2 way associative cache. So our final index is 2^9-2^1=2^8.
Finally, we subtract. 26-4-8=14. So our tag is 14 bits. This is because we need 26 bits to address, we used 4 bits for the offset, 8 bits for the index, and we are left with 14 for the tag.

Problem 3:

Given RAM of 128KB and fully associative cache of 16KB with a block size of 256 bytes, find the Tag and the Offset.

So we have 128 kb of ram, which is 131072 bytes of ram. 2^17 equals 131072, which means we need 17 bits to address this.

So our block size is 256 bytes, which means that we have 8 bytes for our offset, since 2^8 gives us 256.

We subtract like this: 17-8=9. So our tag is 9 bits.

Problem 4

Memory for this problem is small so that it could be written out in binary. Consider a machine with a byte addressable main memory of 64KB s and block size of 8 bytes. Assume that a direct mapped cache consisting of 32 lines is used with this machine. How is a 16-bit memory address divided into tag, line number, and byte number? Into what line would bytes with each of the following addresses be stored? A. 0001 0001 0001 1011 B. 1100 0011 0011 0100

A.

So 2^3 gives us 8 bytes, which is the block size. So our offset value is 3 numbers.

We are also told we have 32 lines. So 2^5 = 32, so our index is 2^5.

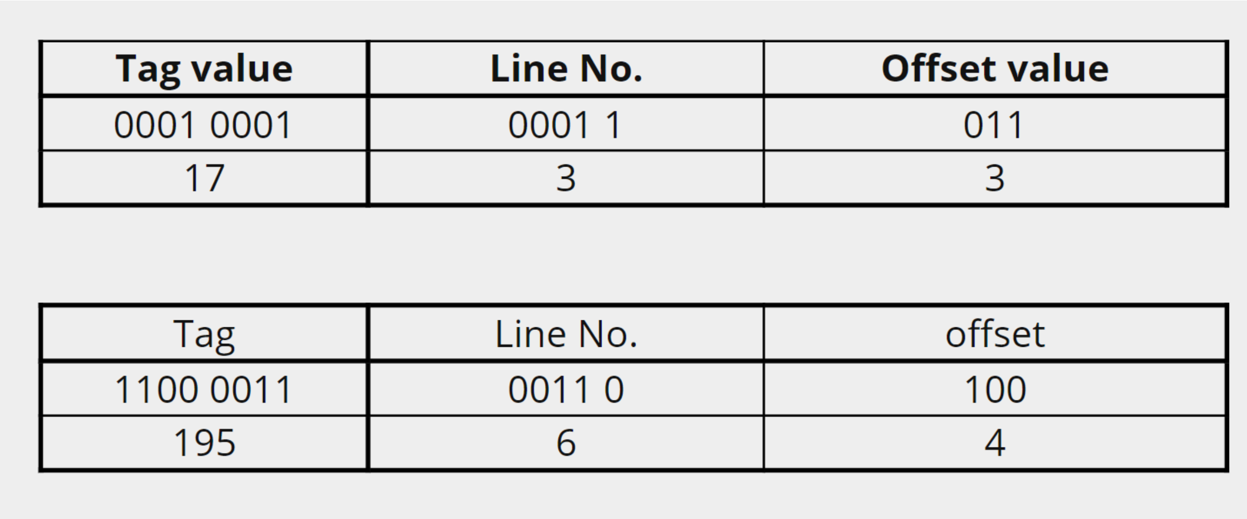
We subtract cache and offset to give our line number. So 2^2

Remaining bits form the tag.

Tag value	Line No.	Offset value
00010001000	11	011

B.

Tag value	Line No.	Offset value
11000011001	10	100



Tag value	**Line No.**	**Offset value**
0001 0001	0001 1	011
17	3	3

Tag	Line No.	offset
1100 0011	0011 0	100
195	6	4