6. (a) (5 points) Consider a 2-way set associative cache (N) with a memory word size of 4 bytes, and a block size of 64 words (b). What is the capacity (C) of this cache in bytes, if you want to have 128 sets (S) in the cache?

Solution: A cache of Capacity C contains B = C/b blocks where b is the block size. In our example b is 64 x 4bytes = 256 Bytes. An N-way set associative cache has S= B/N sets. In our case for S=128, and N=2, we need 256 blocks. So the capacity is 256 x 256 Bytes = 64 kBytes.

(b) (7 points) Explain what the function of the *tag* is in a cache. How many bits would a cache for a 32-bit word addressed MIPS architecture need to store the tags for the cache configuration described in part a)

Solution: The cache only stores a small part of the memory. Only one part of the address is used to address the data in the cache. Since many addresses can map to the same location, the rest of the address has to be stored with the data everytime a cache block is updated. This part of the address is called *tag* and is stored alongside the data in the cache.

A 32-bit MIPS processor uses 32 bits for addressing, and due to the word addressing the last of these address bits are always zero, leaving 30 effective address bits. There are 64 words in a block of our cache, so 6 bits ($2^6 = 64$) will be needed to select the words in a block. There are 128 sets in our cache, so 7 bits ($2^7 = 128$) of the address will be used to select the set. This leaves 17 bits (32 -2 -6 -7) for the tag in each way. There are two ways and 128 sets so 256 x 17 == 4352 bits will be needed to store the tags for this cache.