2)

We design a set-associative cache with parameters:

- Address size A: 32 bits
- Word size W: 32 bits
- Capacity C: 256 bytes
- Associativity N: 2
- Block size b: 4words
- Replacement policy: LRU
- Write policy: write back

a)

How many sets S does the cache have?

$$S = \frac{B}{N} = \frac{\frac{C}{b}}{N} = \frac{\frac{256\ bytes}{16\ bytes}}{2} = \frac{16\ blocks}{2} = 8$$

b)

Specify the bit ranges for the tag, set index, block offset, and byte offset of the address.

- Tag: 31:7
- Set index: 6:4
- Block offset: 3:2
- Byte offset: 1:0

c)

Determine the number of data bits and the number of directory bits (tag, valid, etc.) required to implement the cache.

Single set:

- Directory bits: Use (1) + 2*Dirty (1) + 2*Valid (1) + 2*Tag (25) = 55 bits
- Data bits: ways (2) * block size (4) * word size (32) = 256 bits

Total (8 sets):

- Directory bits: 55 bits * 8 sets = 440 bits
- Data bits: 256 bits * 8 sets = 2048 bits

d)

You have a supply of 4x8 SRAM arrays, 2:1 multiplexers and equality comparators with 32-bit inputs, and 2-input logic gates and inverters. Using these building blocks, design the read-part of the cache. How many building blocks of each type do you need?

- we need 64 words for data; thus 64 SRAMs for data
- we need at least 440bits = 55 bytes for directory info; take 16 SRAMs for directory info (theoretically we need only 13.75 SRAMs)
- We need one 2:1 MUX for the two ways and a 4:1 MUX for each way, thus $1 + 2 \cdot (2^2 - 1) = 7$ 2:1 MUX
- 2 equality comparators
- 2 AND-gates
- 1 OR-gate