1)

We design a direct-mapped cache with parameters:

- Address size A: 32 bits
- Word size W: 32 bits
- Capacity C: 256 bytes
- Block size b: 8 words
- Write policy: write back

a)

How many blocks B does the cache have?

$$B = \frac{C}{b} = \frac{256\ bytes}{8\ words \cdot 32\,\frac{Bits}{word} \cdot 8} = \frac{2^8\ bytes}{4 \cdot 2^3\ bytes} = 2^3 = 8$$

b)

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

Tag:	31:8
Block index:	7:5
Block offset:	4:2
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 block/set:

- Directory bits: Dirty (1) + Valid (1) + Tag (24) = 26 bits
- Data bits: block size (8) * word size (32) = 256 bits

Total (8 blocks):

- Directory bits: 26 bits * 8 blocks = 208 bits
- Data bits: 256 bits * 8 blocks = 2048 bits

d)

You have a supply of 4 x 8 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?

4x8 SRAM: 4 x 8bits = 32 bits = 1 word

- we need 64 words for data; thus 64 SRAMs for data

- we need at least 200bits = 25 bytes for directory info; take 8 SRAMs for directory info (theoretically we need only 6.25 SRAMs)
- We need a 8:1 MUX, thus $2^3 - 1 = 7$ 2:1 MUX
- 1 equality comparator
- 1 AND-gate