A page is divided into 4 blocks. A word is 4 bytes. We have 4-bit offset. The cache can fit two pages at a time.

- What is the size of the block? What is the size of the cache?
- Which block, in which set, and which word this address is referencing: **0x01b6**?
- What is the maximum number of blocks that this system can address?
- What happens when this block is required by the system?

Solutions: This is the information we already have:

- A page is divided into 4 blocks.
- A word is 4 bytes.
- We have 4-bit offset.
- The cache can fit two pages at a time.

1) Since we know that we have 4 bit offset, we conclude that the block contains 16 words. This is because the offset bits specify the location of a word inside a block, which means that a block contains the number of combination of offset bits (0000, 0001, 0010, …, 1111) which is equal to **2^4** which is 16. We also know that a word is 4 bytes. We can calculate the size of the block by multiplying the size of a word by how many words are in a block. 16 × 4 Bytes = 64 bytes which is the size of a block.

Since we know that page has 4 blocks, and the cache can fit two pages, we conclude that the cache can fit 8 blocks, and has 8 sets. To calculate the size of the cache, we multiply the size of a block by how many blocks we can fit in the cache 64 × 8 = **512 bytes** is the size of the cache.

2) We already know that we have 4 bit offset. We also know that the cache can fit 8 blocks, which means it has 8 sets. So we have 3 set bits. We then convert the address **0x01b6** into binary. Once that is done, we need to map the binary representation of the address into the cache address:

0				**1**				**b**				**6**			
0	0	0	0	0	0	0	1	1	0	1	1	0	1	1	0
Tag bit									Set bits			Offset bits			

Now that we know what the address looks like, we start finding the names of the word, set, and block.

The offset bits refer to the word. This means that the word we are looking for is **word0110**.

The set bits refer to the set. This means that the block we are looking for is located in **set011**.

The tag bits refer to the name of the block we are looking for. This means that the block we are looking for is **block000000011**.

3) The maximum number of addressable blocks is how many blocks you can tag using the 9 bit tag. This is equal to **2^9** which is **512 bytes**.

4) The following happens: A job is divided into pages. One page that contains the word that is in address **0x01b6** is moved to the RAM into a page frame. Once in the RAM, this page is divided into blocks. One of those blocks will be called **block000000011**. Since this block will contain the word we are looking for, it is moved into the cache. In the cache, **block000000011** is stored in set011. When **block000000011** is fully copied to **set011**, one word in that block called **word0110** is sent to the register file. The register file then sends **word0110** to the CPU when it wants it.