1. Suppose a computer using a direct mapped cache has 2^{32} words of word-addressable main memory, and a cache of 1024 blocks, where each cache block contains 32 words.

a. How many blocks of main memory are there?

Ans

To calculate the number of blocks of main memory, we need to divide the total number of words of main memory by the number of words in each block.

$2^{32} / 2^{5} = 2^{32-5} = 2^{27}$

Therefore, there are 2^{27} blocks of main memory.

b. What is the format of a memory address as seen by the cache, i.e., what are the sizes of the tag, block, and offset fields?

Ans

Block: 1024 blocks in the cache, so the size of the block field is 10 bits(2^{10}).

Offset: Each cache block contains 32 words, so the size of the offset field is 5 bits(2^{5}).

Tag: 32 - 10 - 5 = 17 bits.

c. To which cache block will the memory address $000063FA_{16}$ map?

Ans

$(000063FA)_{16} = (00000000000000000\ 1100011111\ 11010)_2$

From block: $1100011111_2 = 799_{10}$

The memory address $000063FA_{16}$ will map at 799 cache block.

2. Suppose a word-addressable computer using a set associative cache has 216 words of main memory and a cache of 32 blocks, and each cache block contains 8 words.

a. If this cache is 2-way set associative, what is the format of a memory address as seen by the cache, that is, what are the sizes of the tag, set, and offset fields?
Ans
The cache used is an associative cache consisting of 2^{16} words, which means that the address used for this cache is 16 bits long. The cache is divided into 2^5 blocks, and each set requires 2 blocks, resulting in 2^4 sets. As a result, the 16-bit address can be split into 9 bits for the tag, 4 bits for the set, and 3 bits for the word.

tag (9 bits)	set (4 bits)	offset (3 bits)

b. If this cache is 4-way set associative, what is the format of a memory address as seen by the cache?
Ans
A cache with 32 blocks that we want to organize into 8 sets, with each set containing 4 blocks. To achieve this, we will use a tag (16 - 3 - 3 = 10 bits), a set field with 3 bits, and a word field with 3 bits.

tag (10 bits)	set (3 bits)	offset (3 bits)

3. Consider a byte-addressable computer with 24-bit addresses, a cache capable of storing a total of 64K bytes of data and blocks of 32 bytes. Show the format of a 24-bit memory address for:

a. direct mapped

Ans

We need 11 bits for the index 2^{11} (2048 blocks) , 5 bits for the offset 2^5 (32 bytes per block) and 8 bits for tag (24 - 11 - 5 = 8 bits)

tag (8 bits)	block (11 bits)	offset (5 bits)

b. associative

Ans

We need to divide into two fields: tag and offset.

Offset is 2^5 then, tag is 2^{19}.

tag (19 bits)	offset (5 bits)

c. 4-way set associative

Ans

We need to divide into three fields: tag, block, and offset.

The cache has a total of $2^{16}/2^5 = 2^{11}$

Since the cache is 4-way set associative, there are $2^{11}/2^2 = 2^9$ sets.

Offset = 2^5.

tag (10 bits)	block (9 bits)	offset (5 bits)

4. A direct-mapped cache consists of eight blocks. A byte-addressable main memory contains 4K blocks of eight bytes each. Access time for the cache is 22 ns and the time required to fill a cache slot from main memory is 300 ns (this time will allow us to determine the block is missing and bring it into cache). Assume a request is always started in parallel to both cache and to main memory (so if it is not found in cache, we do not have to add this cache search time to the memory access). If a block is missing from cache, the entire block is brought into the cache and the access is restarted. Initially, the cache is empty.

a. Show the main memory address format that allows us to map addresses from main memory to cache. Be sure to include the fields as well as their sizes.
b. Compute the hit ratio for a program that loops 4 times from locations 0 to 6710 in memory.
c. Compute the effective access time for this program.