CMPEN/EE 331 – Computer Organization and Design, Chapter 5 Review Questions

1. Caches are important to providing a high-performance memory hierarchy to processors. Below is a list of 32-bit memory address references, given as word addresses.

   3, 180, 43, 2, 191, 88, 190, 14, 181, 44, 186, 253

a. For each of these references, identify the binary address, the tag, and the index given a direct-mapped cache with 16 one-word blocks. Also list if each reference is a hit or a miss, assuming the cache is initially empty.

b. For each of these references, identify the binary address, the tag, and the index given a direct-mapped cache with two-word blocks and a total size of 8 blocks. Also list if each reference is a hit or a miss, assuming the cache is initially empty.

c. You are asked to optimize a cache design for the given references. There are three direct-mapped cache designs possible, all with a total of 8 words of data: C1 has 1-word blocks, C2 has 2-word blocks, and C3 has 4-word blocks. In terms of miss rate, which cache design is the best? If the miss stall time is 25 cycles, and C1 has an access time of 2 cycles, C2 takes 3 cycles, and C3 takes 5 cycles, which is the best cache design?

d. There are many different design parameters that are important to a cache's overall performance. Below are listed parameters for different direct-mapped cache designs.
   Cache Data Size: 32 KiB
   Cache Block Size: 2 words
   Cache Access Time: 1 cycle

   Calculate the total number of bits required for the cache listed above, assuming a 32-bit address.

1. Caches are important to providing a high-performance memory hierarchy to processors. Below is a list of 32-bit memory address references, given as word addresses.

   3, 180, 43, 2, 191, 88, 190, 14, 181, 44, 186, 253

a. For each of these references, identify the binary address, the tag, and the index given a direct-mapped cache with 16 one-word blocks. Also list if each reference is a hit or a miss, assuming the cache is initially empty.
This is 16 blocks cache, it will need 4 bits for index and the rest will be for the tag.

Word Address	Binary Address	Tag	Index	Hit/Miss
3	0000 0011	0	3	M
180	1011 0100	11	4	M
43	0010 1011	2	11	M
2	0000 0010	0	2	M
191	1011 1111	11	15	M
88	0101 1000	5	8	M
190	1011 1110	11	14	M
14	0000 1110	0	14	M
181	1011 0101	11	5	M
44	0010 1100	2	12	M
186	1011 1010	11	10	M
253	1111 1101	15	13	M

b. For each of these references, identify the binary address, the tag, and the index given a direct-mapped cache with two-word blocks and a total size of 8 blocks. Also list if each reference is a hit or a miss, assuming the cache is initially empty.
Since there are two words per block and it is word addressable, we will need 1 bit for the word offset and 3 bits for the index since we have 8 blocks. This means that the 1st bit to the right is responsible for the word offset (to choose if it is the word to the right or the word to the left). Then the next 3 bits will be responsible for the index. Then, to know if it is hit or miss – you need always to check the previous indices to see if it was called earlier or not. If it is not called then it will be miss but if it is called, then it is hit). Take as an example the word address 190 with index 111, if you check all previous indices you will find that a previous word address that has the same index (191) has been called, then we go to check if it is in the same exact spot inside that block or not by comparing the first bit. Since they are not the same, it will have a spot available inside the block and it will hit if the tag is the same.

Word Address	Binary Address	Tag	Index	Hit/Miss
3	0000 001 1	0	1	M
180	1011 010 0	11	2	M
43	0010 101 1	2	5	M
2	0000 001 0	0	1	H
191	1011 111 1	11	7	M
88	0101 100 0	5	4	M
190	1011 111 0	11	7	H
14	0000 111 0	0	7	M
181	1011 010 1	11	2	H
44	0010 110 0	2	6	M
186	1011 101 0	11	5	M
253	1111 110 1	15	6	M

c. You are asked to optimize a cache design for the given references. There are three direct-mapped cache designs possible, all with a total of 8 words of data: C1 has 1-word blocks, C2 has 2-word blocks, and C3 has 4-word blocks. In terms of miss rate, which cache design is the best? If the miss stall time is 25 cycles, and C1 has an access time of 2 cycles, C2 takes 3 cycles, and C3 takes 5 cycles, which is the best cache design?

The total number of blocks is 8 as stated in item c, then

Cache 1 has 3-bits index (because we have 8-lines 'blocks') and no-bit to choose which word in the block because we have only one word per line

1st word
2nd word
3rd word
4th word
5th word
6th word
7th word
8th word

Cache 2 has 2-bits index (because we have 4-lines 'blocks') and 1-bit to choose which word in the block

1st word	2nd word
3rd word	4th word
5th word	6th word
7th word	8th word

Cache 3 has 1-bit index (because we have 2-lines 'blocks') and 2-bits to choose which word in the block

1st word	2nd word	3rd word	4th word
5th word	6th word	7th word	8th word

All of them will have 5-bits for Tag, here is the table for the three caches.

Word Address	Binary Address	Tag	Cache 1		Cache 2		Cache 3	
			index	hit/miss	index	hit/miss	index	hit/miss
3	00000 0 11	0	3	M	1	M	0	M
180	10110 1 00	22	4	M	2	M	1	M
43	00101 0 11	5	3	M	1	M	0	M
2	00000 0 10	0	2	M	1	M	0	M
191	10111 1 11	23	7	M	3	M	1	M
88	01011 0 00	11	0	M	0	M	0	M
190	10111 1 10	23	6	M	3	H	1	H
14	00001 1 10	1	6	M	3	M	1	M
181	10110 1 01	22	5	M	2	H	1	M
44	00101 1 00	5	4	M	2	M	1	M
186	10111 0 10	23	2	M	1	M	0	M
253	11111 1 01	31	5	M	2	M	1	M

Cache 1 miss rate = 100%
Cache 1 total cycles = 12 x 25 + 12 x 2 = 324
Cache 2 miss rate = 10 /12 = 83%
Cache 2 total cycles = 10 x 25 + 12 x 3 = 286
Cache 3 miss rate = 11/12 = 92%
Cache 3 total cycles = 11 x 25 + 12 x 5 = 335

Cache 2 provides the best performance.

d. There are many different design parameters that are important to a cache's overall performance. Below are listed parameters for different direct-mapped cache designs.
Cache Data Size: 32 KiB
Cache Block Size: 2 words
Cache Access Time: 1 cycle

Calculate the total number of bits required for the cache listed above, assuming a 32-bit address.

First we must compute the number of cache blocks in the initial cache configuration. For this, we divide 32 KiB (32768) by 4 (for the number of bytes per word) and again by 2 (for the number of words per block). This gives us 4096 blocks and a resulting index field width of 12 bits as $2^{12} = 4096$. We also have a word offset size of 1 bit and a byte offset size of 2 bits. This gives us a tag field size of 32 - 15 = 17 bits. These tag bits, along with one valid bit per block, will require 18 x 4096 = 73728 bits or 9216 bytes. The total cache size is thus 9216 + 32768 = 41984 bytes.
To summarize the previous explanation:

The total cache size can be generalized to

Total size = datasize + (validbitsize + tagsize) x blocks
Total size = 32768 + (1 + 17) x 4096
Total size = 41984
datasize = blocks x blocksize x wordsize
wordsize = 4
tagsize = 32 - log2(blocks) - log2(blocksize) - log2(wordsize)
validbitsize = 1