Lab session 10

MIPS R2000 CACHE MEMORY
CODE CACHE

1. How many elements have the vectors of the program? How many bytes occupy every element?

First we will determine the size occupied by program variables in the data segment and the size occupied by the program's instructions in the code segment.

2. Complete the following information in the data segment. Use the hexadecimal system to express memory addresses (do the same along the entire lab guide).

Starting address of vector A	
Bytes occupied by vector A	
Initial direction of vector B	
Bytes occupied by vector B	
Address of variable k	
Address of variable dim	

3. Complete the following information about the code segment. In this case do not forget to consider translating the program pseudoinstructions into machine instructions, since the latter are the only ones to consider. In this case it will be beneficial to load the program in the simulator (no need to run it) to see the address where the last instruction of the program is allocated.

Address of the first instruction	
Address of the last instruction	
Number of program instructions	
Bytes occupied by the program code (instructions)	

4. Determine the number of accesses to memory done by the program. These values are very important because they will help us later to know what is the total number of accesses served by the cache, that is, we can distinguish between accesses that are hits and accesses that are faults.

Accesses to data segment	
Accesses to code segment	

5. Considering the above features, indicate how many lines there are in the cache memory.

6. Indicate what will be the interpretation that this cache will do about the receiving addresses (tag fields, line and offset).

7. The program instruction `jal sax` is stored in the address `0x0040001C` data segment. Indicate what line of the cache will allocate it and its tag.

8. Calculate in this case, how many control bits are stored per line. Similarly, calculate the volume of the directory, that is, the total number of control bits contained in the code cache.

Control bits per line	
Directory volume (bytes)	

9. Load the *original program* and run it by selecting F10 (step by step) to follow in detail the effect on the code cache. Complete the following table:

Accesses to code segment	
Hits	
Faults	
Hit rate (H)	

10. Confirm that the `sax jal` instruction is stored in the expected line with the tag calculated above.

11. Obtain the average access time to the code segment experienced by the program.

12. Use the simulator and set the code cache with a block size of 16 bytes keeping all other parameters as before. Load and run now the original program and complete the following table:

Accesses to code segment	
Hits	
Faults	
Hit rate (H)	

13. As shown, the number of failures has been reduced considerably. What is the reason why?