**Thực hành kiến trúc máy tính tuần 39**

**Full name: Lê Quang Khải**

**Student ID: 20225638**

Assignment 1

8. What was the final cache hit rate? \_\_\_\_\_\_75%\_\_\_\_\_\_. With each miss, a block of 4 words are written into the cache. In a row-major traversal, matrix elements are accessed in the same order they are stored in memory. Thus each cache miss is followed by 3 hits as the next 3 elements are found in the same cache block. This is followed by another miss when Direct Mapping maps to the next cache block, and the patterns repeats itself. So 3 of every 4 memory accesses will be resolved in cache.

A computer screen shot of a computer program

Description automatically generated

9. Given that explanation, what do you predict the hit rate will be if the block size is increased from 4 words to 8 words? \_\_\_\_\_\_\_88%\_\_\_\_\_\_. Decreased from 4 words to 2 words? \_\_\_\_50%\_\_\_\_\_\_

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

12. What was the cache performance for this program? \_\_\_\_\_0%\_\_\_\_\_\_. The problem is the memory locations are now accessed not sequentially as before, but each access is 16 words beyond the previous one (circularly). With the settings we've used, no two consecutive memory accesses occur in the same block so every access is a miss.

A screenshot of a computer

Description automatically generated

13. Change the block size to 16. Note this will reset the tool.

A screenshot of a computer

Description automatically generated

14. Create a second instance of the Cache Simulator by once again selecting Data Cache Simulator from the Tools menu. Adjust the two frames so you can view both at the same time. Connect the new tool instance to MIPS, change its block size to 16 and change its number of blocks to 16.

A screenshot of a computer

Description automatically generated

15. Re-run the program. What is the cache performance of the original tool instance? \_\_\_\_\_\_0%\_\_\_\_\_. Block size 16 didn't help because there was still only one access to each block, the initial miss, before that block was replaced with a new one. What is the cache performance of the second tool instance? \_\_\_\_\_94%\_\_\_\_\_\_. At this point, the entire matrix will fit into cache and so once a block is read in it is never replaced. Only the first access to a block results in a miss.

**The Memory Reference Visualization tool**

**Row -major**

**A screenshot of a computer

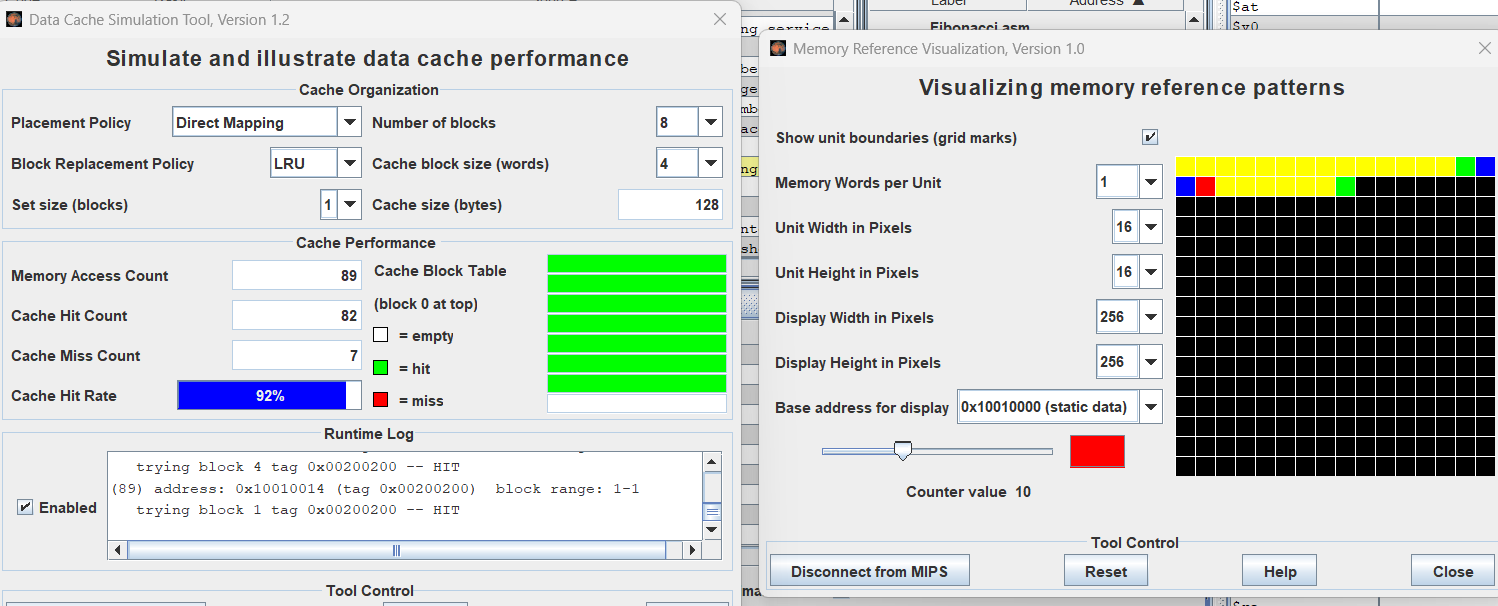
Description automatically generated**

**Column-major**

**A screenshot of a computer

Description automatically generated**

**Fibonacci**

****