CprE 381 Homework 9

[Note: This homework is meant to help you form insights regarding the principles behind caches.]

1. Principle of Locality
    a. Write a valid MIPS assembly program that executes at least 20 instructions and demonstrates spatial locality in instruction fetching, but not data accesses. Explain this locality in the assembly comments.
    b. Write a valid MIPS assembly program that executes at least 20 instructions and demonstrates temporal locality in data accesses, but not instruction fetching. Explain this locality in the assembly comments.
    c. Spend some time looking at open-source programs on Github.com. Find a piece of a C or C++ program on github that appears to display a significant amount of data locality. Provide the html browsable file URL and line numbers of the example. Justify why these lines demonstrate data locality. *[Note that since this is real code, you may need to reference multiple files to demonstrate locality even in a single example.]*

2. Breaking Locality
    Complete the following C function that scales each element of a 2D array by a scalar. First, complete it in row-major ordering (https://en.wikipedia.org/wiki/Row-_and_column-major_order). Second, complete it in column-major ordering. Which is faster on your computer (report the time each takes to execute 1000 calls to scale and what processor model you have)? Why is it faster? Please use the C template provided with this homework.

    ```
    void scale(int n, int m, int array[n][m], int scale);
    ```

3. Direct-Mapped Cache Configuration and Simulation
    a. ZyBooks 5.19.2.a
    b. ZyBooks 5.19.2.b
    c. ZyBooks 5.19.2.c
    d. (From P&H 5th Edition) 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

    The formula shown in Section 5.3 shows the typical method to index a direct-mapped cache, specifically (Block address) modulo (Number of blocks in the cache). Assuming a 32-bit address and 1024 blocks in the cache, consider a

different indexing function, specifically (Block address[31 :27] XOR Block address[26:22]). Is it possible to use this to index a direct-mapped cache? If so, explain why and discuss any changes that might need to be made to the cache. If it is not possible, explain why.