CS 3844 Computer Organization Term Project Part 3 - Cache Simulator– Spring 2021 – 100%

A. Part A Deliverables - Running the Data Cache Simulator tool and observe the runtime results and document your answers for all qns. (40%)

1. From **Tools** menu of MARS, select **Data Cache Simulator tool**

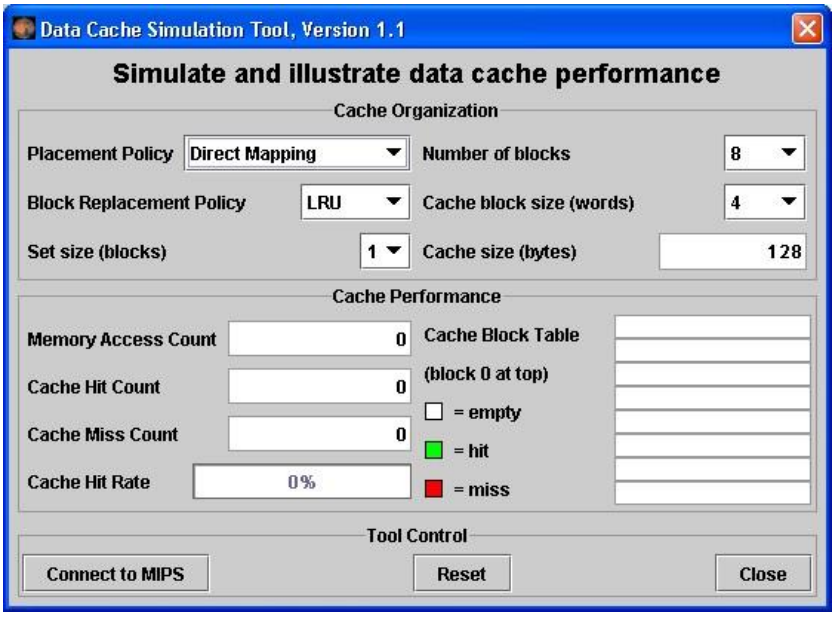
2. Close any MIPS programs you are currently using.

3. Open the program **`row-major.asm`** from the `Examples` folder. This program will traverse a 16 by 16 element integer matrix in row-major order, assigning elements the values 0 through 255 in order. It performs the following algorithm:

```
  for (row = 0; row < 16; row++)
for (col = 0; col < 16; col++)
      data[row][col] = value++;
```

4. Assemble the program.

5. From the **Tools** menu, select **Data Cache Simulator**. A new frame will appear in the middle of the screen.



6. This is a MARS Tool that will simulate the use and performance of cache memory when the underlying MIPS program executes. Notice its three major sections:

a. *Cache Organization:* You can use the combo boxes to specify how the cache will be configured for this run. Feel free to explore the different settings, but the default is fine for now.
b. *Cache Performance:* With each memory access during program execution, the simulator will determine whether or not that access can be satisfied from cache and update the performance display accordingly.
c. *Tool Control:* These buttons perform generic control functions as described by their labels.

7. Click the tool's **Connect to MIPS** button. This causes the tool to register as an observer of MIPS memory and thus respond during program execution.

8. Back in MARS, adjust the **Run Speed slider** to 30 instructions per second. It is located at the right side of the toolbar. This slows execution so you can watch the Cache Performance animation.

9. In MARS, run the program using the **Run** toolbar button  , the menu item or keyboard shortcut. Watch the Cache Performance animate as it is updated with every access to MIPS memory.

10. *What was the final cache hit rate?* _____. 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.

11. Given that explanation, *what do you predict the hit rate will be if the block size is increased from 4 words to 8 words?* _____. *Decreased from 4 words to 2 words?* _____.

12. Verify your predictions by modifying the block size and re-running the program from step 7.
NOTE: when you modify the Cache Organization, the performance values are automatically reset (you can always use the tool's **Reset** button).

NOTE: You have to **reset**  the MIPS program before you can re-run it.
NOTE: Feel free to adjust the **Run Speed slider** to maximum speed anytime you want.

13. Repeat steps 2 through 12 for program **`column-major.asm`** from the `Examples` folder. This program will traverse a 16 by 16 element integer matrix in column-major order, assigning elements the values 0 through 255 in order. It performs the following algorithm:

```
  for (col = 0; col < 16; col++)
for (row = 0; row < 16; row++)
      data[row][col] = value++;
```

NOTE: You can leave the Cache Simulator in place, move it out of the way, or close it. It will not interfere with the actions needed to open, assemble, or run this new program and will remain connected to MIPS memory. If you do not close the tool, then skip steps 4 and 5.

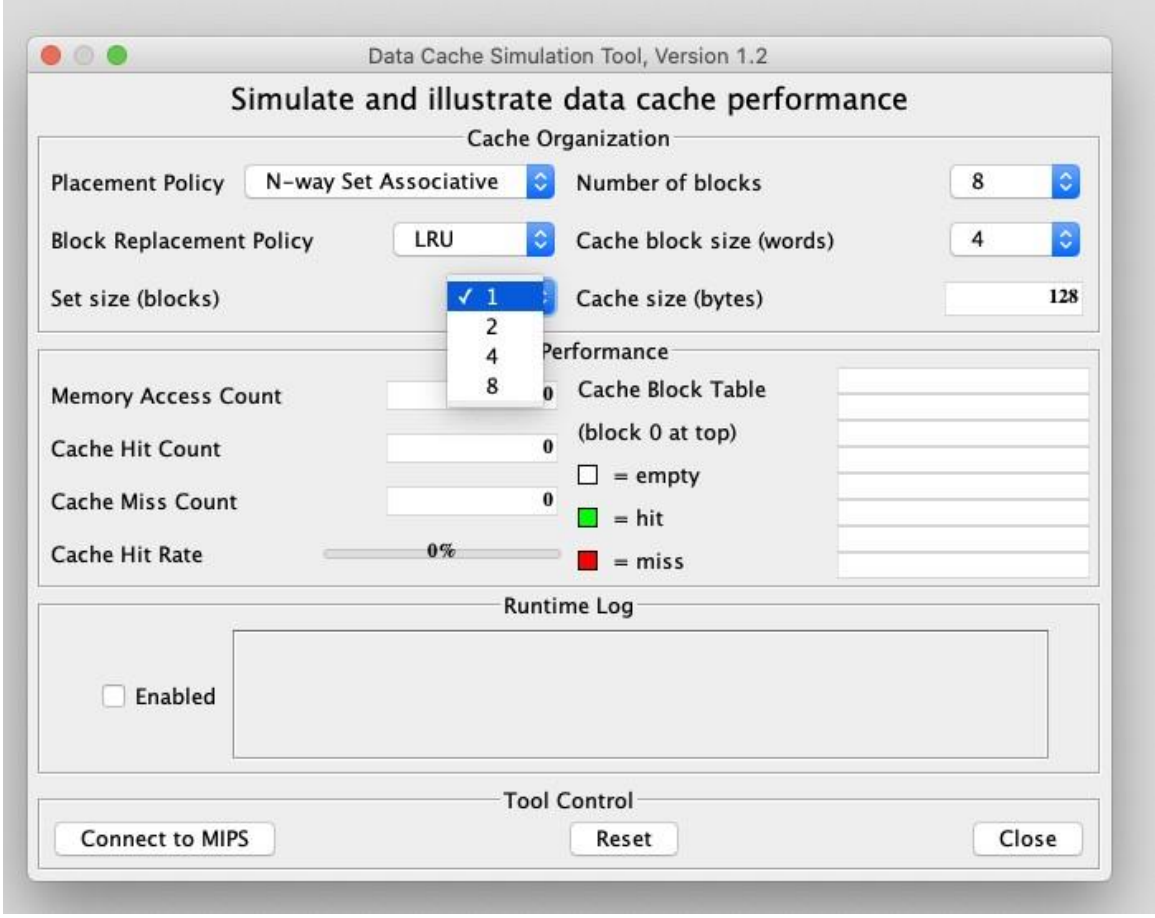
14. *What was the cache performance for this program?* _____. The problem is the memory locations are now accessed not sequentially as before, but each access is 16 words beyond the previous one (circularly). With the settings we've used, no two consecutive memory accesses occur in the same block so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

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

16. 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.

17. Re-run the program. *What is the cache performance of the original tool instance?* _____. 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?* _____. 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.

B. Part B Deliverables (60%)



For the two .asm fles, alter to the different cache configurations (# of blocks, Cache Block Size, Cache Size – see below), collect:

(1) cache hit rates,
(2) screen shots of cache hit rates for each configuration
(3) Plot the performance chart (Y axis – Cache Hit Rates, X axis – Cache Configuration) of each .asm program in an Excel spreadsheet.

- row_major.asm
  - Direct-mapping (# of blocks = 8 and16, block size = 1, cache size = 64 to 16384)
  - 4 way Set Associative mapping (# of blocks = 8 and16, block size = 4, cache size = 64 to 16384)
  - 16 way Set Associative mapping (# of blocks = 8 and16, block size = 16, cache size = 64 to 16384)

- column_major.asm

    - Direct-mapping (# of blocks = 8 and16, block size = 1, cache size = 64 to 16384)
    - 4 way Set Associative mapping (# of blocks = 8 and16, block size = 4, cache size = 64 to 16384)
    - 16 way Set Associative mapping (# of blocks = 8 and16, block size = 16, cache size = 64 to 16384)

- Zip all Part A and B files as the deliverable and submit as Part3.zip to BB. Please be sure to write down names of your team members and their roles and responsibilities.