Computer Architecture Lab Exam

Name: Ahmik Virani
Roll Number: ES22BTECH111001

Implementation details:
I followed the same pseudocode given in the question. I maintained 3 loops, and 4 counters

- Row_index_A
- Column_index_A
- Row_index_B
- Column_index_B

Now, to find the address of the required A[i][k] and B[k][j] as per the pseudocode, I did the following:

- We know that the values are given in row major format, which means that we could convert this 2d matrix to 1d array.
- To do so, we would need the indexing in 1D : array_A[current] = i*colA + k, where colA refers to the number of columns in A, and i, k are the current indexes.
- Similarly I have done for matrix B as well
- Now to access the value of A[i][k], I have basically added the base address of A with 8 * array_A[current]
- Similarly for B

To handle error cases:

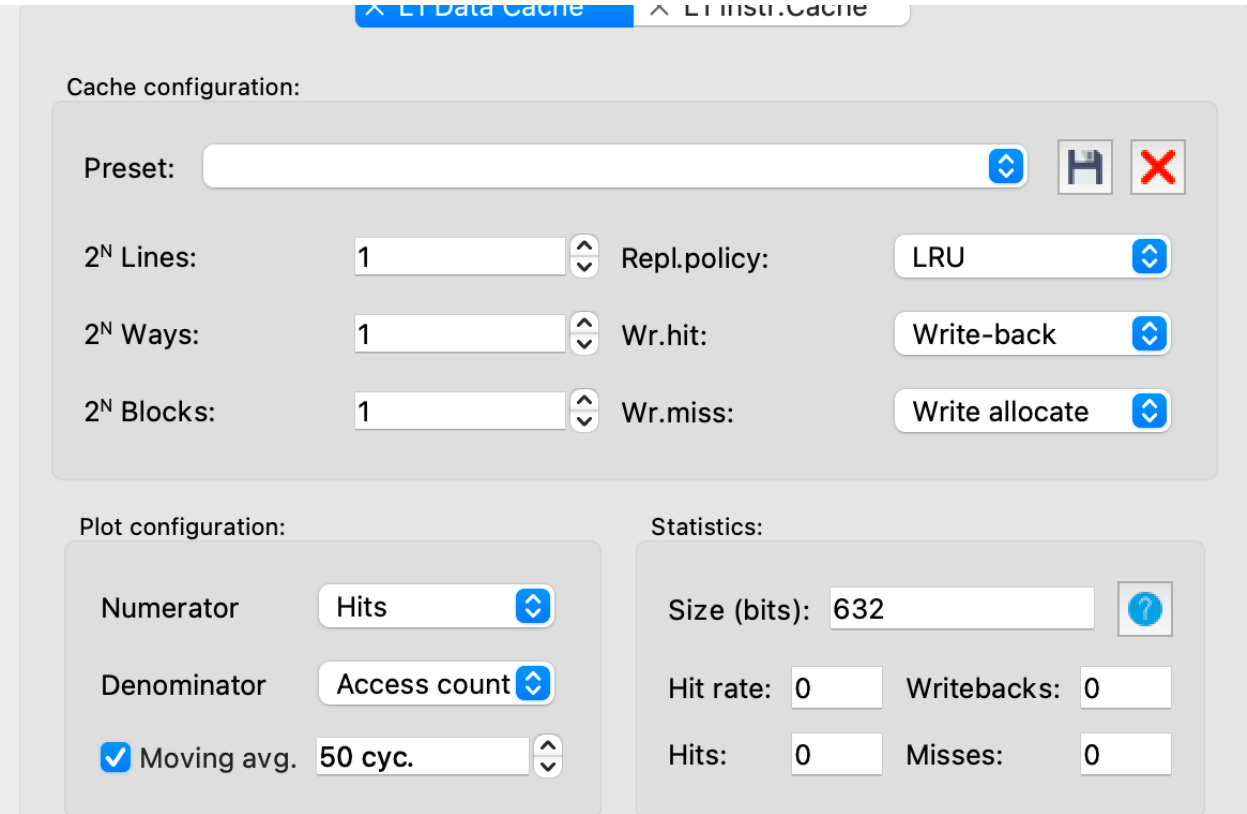
- If colA not equal rowB, then error
- If any of rows or columns are 0, then error

For testing I followed the following tests:

1. .dword 3, 4, 4, 2, 3, 1, 2, 4, 1, 1, 1, 2, 2, 3, 1, 1, 1, 2, 2, 2, 3, 4, 4, 1 -> Given in the question
2. .dword 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1
3. .dword 3, 3, 3, 3, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1 -> Unit matrix
4. .dword 3, 3, 4, 3, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1 -> Error case
5. .dword 0, 3, 3, 3, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1 -> Error case

Part-2

Below attached photo shows the configuration of the cache as specified by the question.



0x0000000010004000
0x0000000010004208

Original Code

Part	Cache Misses	Cache Hits	Total accesses	Hit Rate
A	803	1315		0.6209
B	12419	20611		0.624
C	6275	10371		0.623
D	25091	41475		0.6231

Note that I have tried doing 2nd part using transpose, but the multiplication was not coming correct, so I have submitted an incorrect implementation, but its not entirely incorrect, its partially incorrect.