Dept. of Computer Science and Engineering
IIT Delhi
COL216 : Assignment 4
II Semester 2020-2021

Release date: 24 March 2021
Submission deadline: 10 April 2021 (Saturday), 11:59 PM

General Instructions

1. This assignment can be done in groups (same group as earlier).
2. Demos (online/phone) would be held for all the lab assignments.
3. Adopting any unfair means may lead to -MAX marks and a report to the Institute Disciplinary Committee.
4. Maximum Marks = 20 Marks. This assignment has a total of (20/70 × 40%) = 11.43% overall weight.
5. You can reuse the MIPS simulator you have developed so far, including the DRAM addressing feature used in the Minor (the non-blocking feature is optional for this assignment).

Submission instructions

- Prepare a small write-up (1-2 pages) on the approach taken to solve the problem along with test cases you have considered. The write-up can consist of handwritten notes.
- Explain the approach along with its strengths and weaknesses.
- Explain the testing strategy.
- Zip the document along with the C++ file and test cases and submit at the Moodle submission link.

Problem Statement: Memory Request Ordering

Consider the following memory READ address sequence with a DRAM Row size = 1024 bytes: 1000, 2500, 1004, 2504.
If we service the requests in the above order, the DRAM will change rows ON EVERY ACCESS, which results in poor performance. Can we do better?

1. Sometimes there is an opportunity to change the order in which DRAM requests are serviced. When does this opportunity arise? Assume that the order of instructions and address values of memory instructions cannot be changed.
2. Design and implement a strategy for efficient ordering of DRAM requests at runtime. Remember that the program's semantics cannot be violated (its output cannot change).

Use the same DRAM size/row-size/other architectural assumptions used in the Minor exam. Sample test cases are provided. If you did not handle some of these instruction formats in earlier assignments, please do so now for this assignment.

Input:

1. MIPS assembly language program (as text file, NOT machine instructions). Your interpreter should handle all the instructions: mentioned in Assignment-3: add, sub, mul, beq, bne, slt, j, lw, sw, addi.

2. DRAM timing values ROW_ACCESS_DELAY and COL_ACCESS_DELAY in cycles (as command line arguments). Typical values could be 10 cycles and 2 cycles respectively.

Output:

1. At every clock cycle, print the clock cycle number and all activity in that cycle, such as:
    a. Address of Completed instruction, if any
    b. Modified registers, if any (register number and new value)
    c. Modified memory locations, if any (memory location and new value)
    d. Activity on the DRAM, if any (memory location, row buffer updates)
2. After execution completes, print the relevant statistics such as:
    a. Total execution time in clock cycles
    b. Number of row buffer updates