MINOR

COL 216, Spring 2021

# MIPS Simulator with DRAM timing model

Asha Ram Meena, 2019CS10337

March 21, 2021

## 1 Problem Statement

- **Develop a model for the Main Memory and integrate into the basic interpreter.**
- **The memory access should be non-blocking**

## 2 Design Decisions

◇ DRAM is represented using a 1-d array of length $2^{20}$ and $(i, j)$ memory can be accessed using $i * 1024 + j$.

◇ Memory operations and register operations can run simultaneously in single clock cycle but they should run in sequential order.

◇ Memory and Register operations are not queued.

◇ Whole syntax is based on MIPS convention - allows register access by name and number, jump can be performed using address or label.

◇ DRAM access is valid as long as 4 consecutive bytes are available in the same row and instruction memory is not accessed.

## 3 Strength

◇ Can reduce execution time by at most half when there are very few memory operations and DRAM access is time consuming.

◇ Works best when number of register operations between two consecutive memory accesses is at least equal to the access delay.

- ◇ Does not require any additional data structure for interaction between processor and DRAM.

## 4 Weakness

- ◇ Improvement in execution time will be none if most of the memory operation are grouped together. It can be solved by introducing a queue data structure between processor and DRAM which will store queue of memory accesses.
- ◇ Does not uses multi-threading and hence, loses the advantage of multiple cores in the processor.
- ◇ It can be improved further by performing operations in non-sequential order by maintaining priority queues.

## 5 Input and Output

### 5.1 Input Specifications:

- Input requires input file name, row access delay, column access delay and the mode in which program should run.

### 5.2 Output

- Summary of changes in every cycle and statements executed are shown.
- In the last lines, number of cycles, buffer updates and other relevant statistics are shown.

## 6 Testing

A few test cases showing clear advantage of non-blocking memory, syntax errors and invalid memory accesses were generated to test the Simulator. These have been stored in the directory test as text files containing the required portion of the asm code.