Faculty of Media Engineering and Technology
Dept. of Computer Science and Engineering
Assoc. Prof. Milad Ghantous

CSEN 702: Microprocessors
Winter 2024

Project requirements
(Groups of 6)

You are required develop a GUI simulator in any language you're comfortable with, for the Tomasulo algorithm. Preferably, you can use Java FX and represent everything as tables. You're not required to develop a complex GUI, just enough to represent the status of the system and the content each cycle.

- Your simulator should accept inputs (MIPS instructions in assembly format, either by constructing them graphically using select lists or by loading a text file containing the code) and show **step by step** (cycle by cycle) how these instructions are executed as well as the content of each reservation station/buffer, the register file, the cache and the queue.)
- Your simulator should handle all types of codes, with and without a loop, and that might contain a combination of RAW, WAR and WAW hazards.
- Your simulator should include ALU ops (FP adds, subs, multiply, divide), (integer ADDI or SUBI needed for loops), loads and stores and branches. You can only implement LW, LD and L.S and L.D (and same for stores)
- You can take any assumptions about the cache hit latency and penalty or allow the user to enter it.
- Cache misses need to be incorporated by allowing the user to choose the block size and the cache size. Only consider misses for the data and not the instructions. Also always consider the effective address is calculated. You can write the instructions as such L.D F2,100 where 100 is the effective address directly. Address clashes can be ignored.
- The user should be able to input the latency of each type of instruction before we start simulating. All instructions (integer and floating) enter the architecture. You can create new reservation stations types for integer instructions.
- No branch prediction is used.
- The user should be able to select the size of all stations and buffers.
- The register file can be pre-loaded with some values or allow the user to load them.

- When two instructions wish to publish their result on the bus in the same cycle, you need to handle that and explain how you did so.

Deliverables:

- A report explaining the steps you did to develop and run and test your code. The report should explain your approach, code structure, and test cases.
- A demo (while evaluating the project with your TA).

Warning:

- **Copying a project (or sharing codes) from the internet or from each other will result in a zero for both parties, the copying and the copied.**
- **It's better to submit a non-complete project than to submit a stolen one.**

Notes and deadlines:

The work should be divided equally among all group members according to the complexity of the project and its content. An equal grade for all group members is not guaranteed. In the evaluation, be prepared for any question. All group members should know how to answer.

Submission and evaluations: December 6, 2024, midnight.

Project submission by Google form which will be shared later.

Evaluations will start on December 7, 2024 as well. A reservation system will be provided to you. While evaluating, we should be able to try out any combination of instructions, and your simulator should work.