

# EE4363 / CSci 4203 – Computer Architecture

## Machine Problem 4

### Description

- The file `mipspipe_mp4.v` contains an incomplete behavioral Verilog description of a MIPS processor.
- Study the Verilog code to determine how it works. Next, modify the hardware description to add two features.
  - Stalling to avoid data hazards on load instructions
  - Static branch prediction to speculate past control hazards – The given code statically predicts that a branch is not taken. Create another version of the code that statically predicts that a branch is taken and computes the branch target with a dedicated adder in the decode stage.
- Compile the Verilog code for your MIPS processor. Suggestions for free Verilog compilation and simulation tools can be found in the text file `Verilog_tools.rtf` on Canvas.
- Use the testbench `test_mipspipe_mp4.v` to simulate operation of the MIPS pipeline Verilog module.
- Compare the average CPI for the pre-loaded code with the two different branching policies.
- Summarize your observations in a report.

### Report Format

- Provide a brief summary of your code modifications.
- Provide a brief summary of your experiments.
- The instruction memory (IMemory) is pre-loaded with instructions. Decode and list the pre-loaded instructions.
- Provide a comparison of performance for the two different branching policies.
- Attach your Verilog code for the processor and testbench with your submission.
- Submit a tarball or zip file of your files through Canvas.

### Reference Materials (available on Canvas)

- [Appendix-C-Computer-Organization.pdf](#) – The Basics of Hardware Design
- [Appendix-D-Computer-Organization.pdf](#) – Mapping control to Hardware
- [Section-4.12-Computer-Organization.pdf](#) – An introduction to digital design using a hardware language to describe and model a pipeline and more pipelining illustrations