CS3339 Semester Project

Spring 2018

Dr Ashfaq Hossain

Due Date for Submission: April 24th, midnight, 2018

Class Presentation: April 26th, 2018 (Tentative)

Project Description

1. This is a Team-based Project. There are 3 Teams in the Class. I have placed the students in each Team. **TEAM LEADERS MUST SUBMIT EACH TEAM MEMBER'S SUB-TASK ASSIGNMENT.** Leaders: John Carruth (Team RED), Grant Slape (Team WHITE), Michael Volling (Team BLUE). Please contact your Team Leads (as per the earlier email).

2. The Project simulates a MIPS-like 5 Stage Pipeline with the following Stages: IF, ID, EX, MEM, WB. 32 Registers, 32-bits wide.

3. The Pipeline implements the following:
- Stalls (as needed by the Pipe)
- FORWARDING is implemented in the machine as follows:
  - From MEM Stage to EX Stage (from ***Clock Cycle X to Cycle X+1***)
  - From EX Stage to EX Stage (from ***Clock Cycle X to Cycle X+1*** and from ***Cycle X to Cycle X+2***)
- If an Instruction (Ii) needs Register Source Operand(s) to become available from some previous instruction (say, Ig), then Ii can read it during the 2nd half of the ID phase, if Ig is producing it during its WB stage in the same cycle.
- The Branch Target Address (BTA) and Branch Condition Evaluation (taken/not taken) are known at the end of the EX phase.
- Two memory ports are available so that IF and MEM accesses (from different Instructions) can occur on the same cycle (No structural hazard).

4. For simplicity of your Simulation, consider the following:
- You will have the other needed registers such as Program Counter (PC).
- Exceptions and Interrupts are assumed not to occur in the Pipe during the simulation.
- The following Instructions is a minimal set for your Simulation. Inclusion of other Instructions from MIPS is optional (not a requirement).
  - **lw, sw, add, sub, xor, addi, subi, beq, j Lbl (jump), jal (jump and link), j $reg (e.g. j $ra)**
- Branch range is: PC-128 to PC+128

5. **Two models need to be developed: (a) Icarus Verilog, (b) C or Python. Each model is driven by a single clock signal, on the same clock-edge.**

6. Co-simulation between the Verilog and C/Python models. Co-simulation pointers will be disseminated by the Team Leads to the Members.

7. Passing Criteria for the Project: (a) For Every WB stage of the Pipeline, the Register Files MUST match the values in the 2 models, (b) Stall in a certain Pipeline Stage in Verilog MUST be modeled in C/Python and compared in the correct cycle, (c) Same (Instruction) input MUST be provided to both the models on the same cycle (IF Stage).

8. Any MISMATCH between the 2 models MUST BE reported. Mismatch must be eliminated by correcting the offending model. A Mismatch is an Error!!

9. Input to the simulations is a simple text file where a sequence of instructions is provided.

10. **FINAL submission is midnight, April 24th, 2018.**