CS224 – Fall 2019 - Lab #5

Implementing the MIPS Processor with Pipelined Microarchitecture

Dates: Section 1, Wednesday, Dec 4th 2019, 13:40-17:30
Section 2, Friday, Dec 6th 2019, 13:40-17:30
Section 3, Wednesday, Dec 4th 2019, 8:40-12:30
Section 4, Thursday, Dec 5th 2019, 13:40-17:30
Section 1, 2, 3, 4 Preliminary Report Deadline: Wednesday, Dec 4th 2019, 10:40
Lab Location: EA-Z04

Purpose: In this lab you will implement and test a pipelined MIPS processor using the digital design engineering tools (System Verilog HDL, Xilinx Vivado). To do this, you will need to add pipeline registers, forwarding MUXes, a hazard unit, etc. to your datapath, and of course make the control pipelined as well. You will be provided a skeleton System Verilog code for the Pipelined MIPS processor and fill the necessary parts to make it work. Then, you will synthesize them and demonstrate.

Summary

Part 1 (45 points): Preliminary Report/Preliminary Design Report: Pipeline hazards evaluation and preparing test modules in MIPS (Due date of this part is the same for all).

Part 2 (55 points): Implementation and simulation of the MIPS-lite pipelined processor.

DUE DATE/TIME OF PART 1 --SAME FOR ALL SECTIONS

a. Please drop your written Preliminary Design Report into the box provided in front of the lab by 10:40 am on Dec 4th. No late submission!

DUE TIME OF PART 2—DIFFERENT FOR EACH SECTION:

a. You have to demonstrate your Part 2 lab work to the TA for grade by **12:15** in the morning lab and by **17:15** in the afternoon lab. Your TAs may give further instructions on this. If you wait idly and show your work last minute, 20 points may be taken off from your grade.

b. At the conclusion of the demo for getting your grade, you will **upload your lab work** to the Unilica Assignment, for similarity testing by MOSS. Please see the related section below for further instructions on MOSS submission.

If we suspect that there is cheating, we will send the work with the names of the students to the university disciplinary committee.

Part 1. Preliminary Work / Preliminary Design Report (45 points)

At the end of this lab, you will have implemented the pipelined MIPS architecture that can be seen in the file that is provided as ***PipelineDatapath.PNG*** (Notice that there is no early branch prediction in this pipeline. Hence, the branch resolution is done in the ~~Execute~~ Decode stage.). Note also that there is no jump instruction implemented as well. Be sure to have a printout of the pipelined processor with you, to use during the lab. Your PDR should contain the following items:

a) Cover page, with university name, department name, and course name and number at the top, “Preliminary Design Report”, Lab # (e.g. 5), Section #, and your name and ID# in the middle, and the date of your lab at the bottom.

b) **[10 points]** The list of all hazards that can occur in this pipeline. For each hazard, give its type (data or control), its specific name (“compute-use” “load-use”, “load-store” ~~“J-type jump”~~, “branch” etc.), the pipeline stages that are affected.

c) **[10 points]** For each hazard, give the solution (forwarding, stalling, flushing, combination of these), and explanation of what, when, how.

d) **[10 points]** The logic equations for each signal output by the hazard unit, as a function of the input signals that come to the hazard unit. This hazard unit should handle all the data and control hazards that can occur in your pipeline (listed in b) so that your pipelined processor computes correctly.

e) **[15 points]** Write small test programs, in MIPS assembly, that will show whether the pipelined processor is working or not. Each of your test programs should be designed to catch problems, if there are any, in the execution of MIPS instructions in your pipelined machine. Write:

- A test program with no hazards (to verify that there are no problems with the connections in your pipeline etc.)
- A test program that has one type of hazard, and another, and another...

In the end, have at least 4 test programs (testing at least 3 hazards) with their machine code (in hex).

You can use the student-written assembler tool available online to help you quickly implement your test programs[1]*. Remember that the goal of testing is to verify that all the instructions are fully working, and that all the instructions are working even in the presence of hazards.*

[1] https://github.com/bilkentCraps/mips

Part 2: Implementation and Simulation (55 points)

a) You are given a skeleton System Verilog code for your pipelined MIPS processor in the file ***PipelinedMIPSProcessorToFill.txt***. The places in the code that needs to be modified are shown with comment blocks above them. Fill them to implement Pipelined Processor. You don't need to follow the skeleton code point by point. If you think your design is better, you are welcome to try it in your code, as long as your version of the code works, too.

b) Now make a System Verilog testbench file and using Xilinx Vivado, simulate your Pipeline Processor by executing the test programs you wrote at Prelim e). Implement a new top module in order to see memwrite, regwrite, writedata, pc, instruction and resultw signals. Study the results given in the simulation window. Find each instruction, and understand its values. Do this step for each of the test programs you wrote.

c) When you have integrated all the System Verilog modules together and your whole pipelined MIPS is working in simulation with the test programs you wrote, call the TA and show it for grade. To get full points from this part, you must know and understand everything about what you have done.

Part 3. Submit your code for MOSS similarity testing

Combine all the new and modified Verilog codes into a file called **StudentID_FirstName_LastName_SecNo_LabNo_LAB.txt**. You will then upload this file to the Unilica > CS224 > Assignment for your section. While the TA or Tutor is watching, you will upload this file. Be sure that this .txt file contains exactly and only the codes which are specifically new or modified. Check the specifications! *Even if you didn't completely finish, or get the Verilog codes working, you must submit the YourName_YourSurname_Lab5.txt file to the Unilica Assignment for similarity checking.* If you don't submit your code, your grade for the lab will be 0 (see Lab Policies section, below NOTES.) Your codes will be compared against all the other codes in the class, by the MOSS program, to determine how similar it is (as an indication of plagiarism). So be sure that you only submit code that you actually wrote yourself! All students must upload their code to Unilica > Assignment while the TA or Tutor is watching, and before the deadline. NOTE: you are allowed to upload only ONE file to Unilica, so be sure it contains exactly and only the codes required.

Part 4. Cleanup

1) After saving any files that you might want to have in the future to your own storage device, erase all the files you created from the computer in the lab.
2) When applicable put back all the hardware, boards, wires, tools, etc. where they came from.
3) Clean up your lab desk, to leave it completely clean and ready for the next group who will come.

LAB POLICIES

1. You can do the lab only in your section. Missing your section time and doing in another day is not allowed.
2. Students will earn their own individual lab grade. The questions asked by the TA will have an effect on your individual lab score.
3. Lab score will be reduced to 0 if the code is not submitted for similarity testing, or if it is plagiarized. MOSS-testing will be done, to determine similarity rates. Trivial changes to code will not hide plagiarism from MOSS—the algorithm is quite sophisticated and powerful. Please also note that obviously you should not use any program available on the web, or in a book, etc. since MOSS will find it. The use of the ideas we discussed in the classroom is not a problem.
4. You must be in lab, working on the lab, from the time lab starts until your work is finished and you leave.
5. No cell phone usage during lab.
6. Internet usage is permitted only to lab-related technical sites.
7. For labs that involve hardware for design you will always use the same board provided to you by the lab engineer.