EECS 470

Programming assignment #3

This assignment is worth 6% of your grade in the class and is graded out of 120 points.

VeriSimple4 is a simple pipelined implementation of a subset of the Alpha instruction set, written in synthesizable, behavioral Verilog. The structure of VeriSimple4 is very similar to the MIPS pipeline covered in the text. We have provided you with a base version of VeriSimple4 that simply stalls on each and every instruction for 4 cycles.

You are to extend the VeriSimple4 pipeline to support full hazard detection and bypassing. Your solution is subject to the following restrictions:

- Branches should resolve in the same stage they are currently resolved in.
- All forwarding must be to the EX stage, even if the data isn't needed until a later stage.
- Any stalling must occur in the decode stage. (That is, if stalling is required the dependent instruction should stall in the decode stage.) Obviously, instructions following the stalling instruction in the IF stage will have to stay in the IF stage. Put another way, if you need to insert a noop, it should be inserted in the EX stage. This is to be done by placing the noop in the ID/EX pipeline stage at the end of the cycle. *You should stall as little as possible given the above restrictions.*
- If you wish to insert a noop do so rather than simply invalidating the instruction.

Add control logic to detect all possible control and data hazards. Add bypass paths (forwarding) to eliminate hazards whenever possible (within the rules stated above), and stall instructions if (and only if) there is a hazard that cannot be resolved by bypassing. You should predict branches are not taken and squash if incorrect. Verify that the programs without no-ops produce the same results on your pipeline as the no-op padded versions.

Your synthesized version of the code (.vg files) as well as your behavior model (.v file) will be tested, so it does need to synthesize. You will also submit your Verilog code electronically. Details will be provided in advance of the due date, but will be similar to pervious submissions.

The grading will involve a number of test programs being run and us checking to be sure you've gotten the correct results, correct CPI, correct pipeline output, and are actually following the directions specified above. Additional files, including the VeriSimple4 files can be found on the website. The output files which are currently generated by VeriSimple4 (such as the pipeline output) should continue to reflect the state of your processor, and their format should not be changed at all.