Verilog Project

Project Overview:

During the course of this project you will be required to design a limited version MIPS processor. The processor will support a limited number of instructions and functionalities. The processor needs to be tested for the supported functionalities. All design and test aspects should be well documented.

Deadline & Submission Process:

- The project along with its report are due by: **Noon 27th Dec 2019**.
- All submissions are via email to: a.almousa@psut.edu.jo
- The subject of the email should include your team #.
- It is required that your code should be available on GitHub. (Create an account with your personal Email, to maintain a portfolio of your projects)
- Delivery past the deadline will result in **ZERO** credit.
- The Project Report should be in docx format (~10 Pages. Not more than 15, single spaced, Font: Times New Roman size11)
- Submission should be a **ONE** zipped file names (Group_#.zip) that include the following:
  1. Honor Statement, signed by ALL members of the team. "We, signing below, swear that the work submitted in this report is done entirely by our team. And that we have not copied anything from any other resource"
  2. Project Report that included the following: (See template)
     - Introduction: describe the overall components and design methodology.
     - Processor Design: Detailed description of each module and its functionality.
     - **Design Block Diagram:** Preferably, standalone image or PDF file. Showing names and wire name of all components – needs to match the Verilog names.
     - Design Analysis: Discuss issues and decisions you made during the design.
     - Given Tests runs and discussion: Show the output of the given test cases and discuss their output.
     - Additional Test runs: Discuss additional crucial test cases (Maximum 10). These are to be discussed in the report. But you can create other tests and submit them with the project along with their results.
     - Conclusion
  3. Design & Test Verilog files: use the attached files as templates.
  4. Run logs: the run result for each test case: tb_1.txt ..etc.

Tools:

Use the following Verilog simulator (or any one of your choice)

- Download Verilog from: http://bleyer.org/icarus/

Grading Policy

The distribution of the grade will be as follows:

Category	Sub-Category	Max Grade
Report (25%)	Block Diagram	5
	Language and Organization	5
	Report Completeness & Quality	15
Code (Design Modules) (5%)	Code organization & Commenting	5
Design Functionality (70%)	Integer Instructions	20
	Basic Pipeline	15
	Forwarding	10
	Control Hazards	10
	Floating Points Instructions	15
Penalties	Past Deadline	-100
	Honor statement	-100
	Correct email and subject	-5
	Single Zip file	-5
	Report size and Font	-5
	Missing GitHub files	-5

Requirements:

You are required to develop the following:

- A limited version of a 32 bit-MIPS processor. Instruction set is based on the attached excel sheet.
- The processor should have the support for the following instructions – at least:
    1. All the core instruction set.
    2. The highlighted Floating Point (in Yellow).
- The Design should include ALU-ALU and MEM to ALU-Forwarding.
- The processor prediction scheme assumes (Branch/Jump ..etc) not to be taken.
- Assume Branch outcome is determined in the EXE stage.
- Handling for Control Hazards is required. (Example: need to flush pipeline if there is a jump!)
- Two separate Instruction and Data Memories (IM size 16K x1byte, DM size 1K x 1byte).
- The processor execution should be triggered on positive edge.
- To allow for Reading/Writing to memory and Reg file during the same cycle, use the negative edge of the clock to trigger reading)

The following are detailed info about JAL:

==========JAL========

This a bit detailed...

Instead of stalling the processor would try to execute an independent instruction in the delay slot while it is trying to branch or not ... so when it comes back from the jump it would need to skip that slot.

For the sake of the project, you can implement it to be PC+4

https://chortle.ccsu.edu/AssemblyTutorial/Chapter-26/ass26_4.html

http://www.jaist.ac.jp/iscenter-new/mpc/old-machines/altix3700/opt/toolworks/totalview.6.3.0-1/doc/html/ref_guide/MIPSDelaySlotInstructions.html