ECE 593 Fundamentals of Pre-Silicon Validation
Project Code Drop 2 Report

Team Members:
Deeksha Kamath, **Ritvik Tiwari**, **Tanvi Patil**

Design Changes - We made some changes in our design

1. Loading instructions into the instruction memory module.
2. Converted local memory modules to globally visible memory arrays.

Stimulus/Constraints- We have added the following constraints in our testbench code.

1. Opcode – To ensure that the randomized opcode generated after randomization of the instruction word is valid.

2. Immediate field – To ensure that after the instruction word is randomized, the immediate field bits hence obtained don't amount to value should not cause segmentation fault.

Coverage- We have added the following covergroups and respective coverpoints in our testbench.

1. Hazard – To check if there are any hazards with a coverpoint called hazard detection.
2. Forward Unit – To check if forwarding has been enabled or not with coverpoint forward_EN.
3. Mux – To observe the multiplexer output with respect to Immediate Field and Memory Read Enable with coverpoints Is_Imm and MEM_R_EN.
4. ALU – To check for all the operations as per varying opcodes with the help of coverpoint called EXE_CMD and respective bins, to check the values of the val1, val2 signals.

Testbench structure- We have created the following components as of now:

1. Package – It contains a typedef enum for opcodes that are valid. Also, we have added typedef enum for ALU opcodes and included all .sv files of testbench components.

2. Transaction – Declared inputs for randomization and added rand type variables on different segments of instructions and added constraints.

3. Coverage – Added covergroups and coverpoints on internal signals of design.

4. Generator – To provide constrained randomized set of instruction to the mailbox.

5. Driver- To drive the inputs to DUT via interface.

6. Environment- Encapsulate all the testbench components.