ECE 3544: Digital Design I
Project 1 (Part B): Simulation in the ModelSim Environment; Continuous Assignment Models

Objectives
This project will improve your familiarity with the ModelSim tools, timing models, and with the use of test benches. Additionally, it will allow you to compare the functionality and coding structure of Verilog models that use primitive gates and those that use continuous assignment.

Requirements
You must have ModelSim ALTERA STARTER EDITION 10.3c (or higher) installed on your computer. The instructions in this assignment are consistent with that version of ModelSim. Even if these instructions are consistent with other versions of ModelSim, you should use the version indicated here.

The DE1-SoC Board is not required for this project. This project involves only simulation.

Project Description
As in this project your first step will be to simulate two pre-designed circuits:

- The 74138 3-to-8 decoder, and
- A 4-to-16 decoder that uses two of the 3-to-8 decoders as structural blocks.

This time, however, the basis for describing the 74138 chip is a continuous assignment model using dataflow operators, instead of the primitive gate model you used in Project 1A. In your lab report, you should comment on the differences that you perceive in continuous assignment models that use dataflow operators, and structural models that use primitive gates. Your commentary should include discussion on design differences and implementation differences.

After using test benches to simulate the decoders and verify that the changes in the models do not cause differences in behavior, you will repeat the design and implementation process of Project 1A, except that your target model will be continuous assignment. You should be able to simulate your design and verify that its behavior matches that of the primitive gate circuit you designed in Project 1A.

Instructions

Step 1: Make a new project.

Follow the instructions in Project 1A for making a project for the Project 1B Starter Files. Study the example files to make sure that you understand how each model works in Verilog. Start with the “new” sn74138.v. This is a Verilog behavioral model – specifically, one that uses continuous assignment and dataflow operators to describe the same 74138 chip that you saw in Project 1A.

Next, look at decoder4to16.v. This is the Verilog model for a 4-to-16 decoder made from two 3-to-8 decoders. Even though this model does not contain the same explicit inverter that you saw in the model from Project 1A, you should be able to determine the manner in which it performs the same operation.

You should recognize the two test benches as being the same ones that you saw in Project 1A. This should make sense, since you are seeking to simulate the behavior of the same circuits. Review the two test benches to ensure that you still understand how they function as stimulus files.

Step 2: Compile and simulate the starter files.

Follow the instructions in Project 1A to compile the files in the project and to simulate the two decoders. Include screenshots of your results showing proper operation of the two decoders. Comment on whether the change in the modeling technique affects the behavior of the two decoders.

Step 3: Create your own models and test bench

For this step, you must create the "judge" circuit for the game "Rock, Paper, Scissors" that you designed and implemented in Project 1A, except that you must implement it with a *continuous assignment model using dataflow operators*.

The circuit accepts same two 2-bit inputs (player_a[1:0], player_b[1:0]) and provides the same three outputs (player_a_wins, player_b_wins, and tie_game). The 2-bit values represent the same moves that they did in Project 1A:

Player's Move	Rock	Paper	Scissors
Input Value	11	10	00

As before, the outputs player_a_wins and player_b_wins should equal one if the associated player wins, and should equal 0 otherwise. The output tie_game should equal 1 if the players tie, and should equal 0 otherwise.

Follow the instructions in Project 1A for using the starter files, naming your module and files accordingly, and creating a new project and add your modules to it. The test benches that you created in Project 1A should suffice for this implementation. Make sure to change the names to match the convention that you follow in this project. You should include waveforms from the test-bench in your report that show the proper operation of your design. Comment on whether the change in the modeling technique affects the behavior of this model, as compared to the one you derived in Project 1A.

Your modules and test bench should include proper header information, contain reasonable comments, and be neatly formatted. The starter files provide examples of header information; make sure that you fill in the header comments.

Project Submission
Your submission should include the following files:

- A project report. Your report should be in Word or PDF format. Include your PID in your report filename, *e.g.*, project1report_jthweatt.pdf.

  Your report should contain the following elements:
  - A restatement of the assignment's objectives.
  - A section that describes the process by which you derived the design of your circuit and the manner in which you implemented it in Verilog. In particular, you should comment on how applicable you believe your design process and your method for implementing it in Verilog are to performing more general and larger-scale designs and implementations. Compare your responses here to the ones you gave in Project 1A for primitive gate models.
  - A section that shows the simulation of the decoders and the results of your game circuit testing. Discuss how you formulated the tests contained in the test bench that you had to generate, and how you verified the correctness of your implementation.
  - A section where you discuss your conclusions and the lessons you learned from the assignment.

- The Verilog files for your game module and test-bench. Name your files according to the convention described in the starter files.

All of the files should be zipped together and submitted on the Canvas assignment page along with your report. Include your PID in the name of your archive file, *e.g.*, project1bfiles_jthweatt.zip.