ECE 571 Final Project -- Verify a pipelined picoblaze model and then find the defects we injected into the design using your verification environment

---

This assignment is worth 100 points. Final project presentations will be conducted live on either Mon, 06-Dec or Tue, 07-Dec . Presentation slots can be reserved via a Doodle calendar. Deliverables are due on Wed, 08-Dec by 10:00 PM. NO LATE SUBMISSIONS WITHOUT PERMISSION

We will be using GitHub classroom for this assignment. You should submit your assignment before the deadline by pushing your final code and other deliverables to your team's GitHub repository for the assignment.

After completing this assignment students should have:

- Gained experience w/ unit level and system level design verification
- Gained experience w/ SystemVerilog verification techniques using checkers, assertions, randomization, etc
- Gained experience making a technical presentation
- (optional) Gained experience w/ hardware emulation using the Mentor Graphics Veloce emulation system

Final Project write-up:

Final Project Assignment

Summary

The final project is a chance to put what you learned this term into practice. Each team will perform unit-level and full-core testing on a pipelined picoblaze model. You will implement a testbench (or testbenches) for simulation. Emulation on Veloce or exploring Formal Verification techniques is optional. Teams pursuing this additional work may add a 4th team member. Please submit a proposal (no more than 2 pages) for the additional work. The 4th team member should make a significant contribution to the project

The pipelined picoblaze (also called Rojoblaze) is a pipelined version of the Xilinx Picoblaze, an 8-bit microcontroller for Spartan-3, Virtex-II and Virtex-II Pro and Series 7 FPGAs. Rojoblaze was created by John D. Lynch and Roy Kravitz from Pablo Bleyer Kocik's Pacoblaze code base. PacoBlaze is a from-scratch synthesizable & behavioral Verilog clone of Ken Chapman's popular Picoblaze embedded microcontroller We are providing a SystemVerilog implementation of the Rojoblaze that was written by SethR, MilesS, ShubhankaSPM, and Supraj Vastrad for their ECE 571 Winter 2020 final project.

We envision that the key to verifying this design will be writing and executing a suite of test programs using the KCASM (written in Java) or kcpsm3 (Windows) Assembler. Assertions will be the primary mechanism for checking that the pipelined picoblaze works. The verification strategy and implementation, however, is up to the team.

The final project will culminate in a technical presentation for the instructor and T/A. The teams will present their verification environment, verification strategy, verification approach, test results, and lessons learned during this presentation. Each team will have ~20 minutes to make their presentation.

We will be using the group project support in GitHub classroom for this assignment. This means that your team will share a private repository on GitHub that can also be accessed by the instructor and T/A for the course. You will submit your work via that repository and we would also like you to submit a .zip version of your repository to the team's Final Project dropbox. We will

review your work and provide feedback based on what your final submission was. The process for creating a team in GitHub classroom is not much different than that of creating a new team in D2L. One of the team members creates a new group in GitHub classroom, accepts the assignment, and tell his/her partners what group to join. The other members join the group and accept the assignment. Please only join your team; joining another team either accidentally or deliberately could prevent the actual team member(s) for that group to join.

Where to submit your deliverables for the project:

- Verification Plan: Submit your Verification Plan to the appropriate dropbox on D2L. Include a .pdf of the Verification Plan in your GitHub repository
- Verification Report: Submit your verification report with your final deliverables. The Verification Report should be a completed (test results included) version of your Verification plan. The Verification report should include the coverage statistics that can be provided by QuestaSim, conclusions, lessons learned, next steps, and a work breakdown (which team member did what?)
- Demo presentation slides - Include a .pdf of your final project presentation in your final deliverables.
- Source code, makefiles, etc.: Include all of the source code you wrote for the project. This code should naturally be in your GitHub repository since you are committing your changes, merging your source code, etc. to GitHub.

Grading Rubric

This project is worth 100 points. There is the possibility of extra credit for projects and project presentations that stand out (in a good way)

- 25 pts: Verification Plan
- 40 pts: Final project presentation
- 25 pts: Quality and contents of your Design Report
- 10 pts: Quality/readability of your source code
- (up to) 5 pts: Extra Credit. Extra credit is just that - extra. Your documentation and presentation must be prepared, well-written, and well presented for the project to be eligibile for extra credit

Broken model

We will provide a pipelined picoblaze model that we've injected "subtle" bugs into as we get closer to the project due date. This code in this model will be encrypted so that you cannot perform a diff or code review between the working model and the broken model. How do you find the bugs? If you have a good verification strategy and good test cases they should identify the problems. If, not, you may need to add additional test cases. It is our intent to give each team doing the Pipelined Picoblaze project their own "broken" model.

Important Dates

- Mon, 03-Nov: Final project assigned during class
- Mon, 08-Nov: Pipelined Picoblaze project released to D2L and GitHub Classroom
- Mon, 29-Nov: "Broken" model(s) released. The model(s) will be encrypted.
- Mon, 06-Dec: Final project presentations - day 1
- Tue, 07-Dec: Final project presentations - day 2
- Wed, 08-Dec: Final project deliverables due to D2L and GitHub by 10:00pm