

# 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 Tue, 06-Dec or Wed, 07-Dec . Presentation slots can be reserved via an online calendar. Deliverables are due on Thu, 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

## 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.

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 one of the keys to verifying this design will be writing and executing a suite of test programs using the KCASM (written in Java) or kcpasm3 (Windows) Assembler. Assertions will be the primary mechanism for checking that the pipelined Picoblaze works. Teams that have done this project in the past have made heavy use (often over 100 assertions) of assertion-based checking to flag defects in the implementation. The verification strategy and its 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. Depending on the size of the conference room for the final project presentations it may be possible for other teams to attend a presentation - we hope so.

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 on Canvas. We will review your work and provide feedback based on what your final submission was.

## Where to submit your deliverables for the project:

- Verification Plan: Submit your Verification Plan to the appropriate dropbox on Canvas. 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)

- 30 pts: Verification Plan
- 40 pts: Final project presentation
- 25 pts: Quality and contents of your Design Report
- 5 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 eligible 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.

## Important Dates

- Wed, 09-Nov: Final project assigned during class
- Tue, 15-Nov: Pipelined Picoblaze project released to Canvas and GitHub Classroom
- Sat, 26-Nov: Verification plan submitted to Canvas by 10:00 PM
- Tue, 29-Nov: "Broken" model(s) released. The model(s) will be encrypted.
- Tue, 06-Dec: Final project presentations - day 1
- Wed, 07-Dec: Final project presentations - day 2
- Thu, 08-Dec: Final project deliverables due to Canvas and GitHub by 10:00pm