ECSE 548: Introduction to VLSI Design
Project Definition

Fall 2013

Objective

The objective of this project is to build on the learning experience of the introductory labs with the development of a large-scale VLSI design. The project will include the specification, design, and layout, verification, and optimization of a large circuit, and therefore require careful planning and testing. Unlike the lab exercises, design evaluation will also make up an important part of the project.

Each project group (up to four people) will propose, execute, evaluate, and optimize its own project, submitting a variety of deliverables during this process.

Deliverables

The project deliverables consist of:

Deliverable	Weight	Due Date
Project proposal (PP)	5%	9-Oct-2013
Mid-project status report (SR)	20%	6-Nov-2013
Final demo (FD)	25%	20-Nov – 27-Nov-2013
Final presentation (FP)	25%	29-Nov, 3-Dec-2013
Final report (FR)	25%	3-Dec-2013

Please note that late deliverables will not be accepted and plan accordingly.

Project proposal (submitted electronically, due 9-Oct-2013, 23h59)

In one page or less:

- Indicate the members of your group, up to four people,
- Summarize the intended result of your project, specifying
  - the system to be designed (e.g., a cache, a multiplier, etc.),
  - the intended manner of evaluation (e.g., cost, performance, etc.), and
  - the intended optimization approach (e.g., stage effort optimization, logic minimization, circuit family selection, etc.)
- Briefly describe your design plan (e.g., microarchitecture, circuit, layout strategy), and
- Identify an appropriate deliverable for the mid-project status report, (e.g., microarchitecture, circuit schematic, standard cell library, test benches, etc.).

All submissions must be in PDF format.

Status report (submitted electronically, due 6-Nov-2013, 23h59)

In addition to providing the deliverable described in the proposal, in no more than one

page:

- Indicate whether or not the deliverable has been completed as planned,
- Describe any issues encountered in carrying out the initial implementation, evaluation, and optimization plan, and
- Describe any corresponding changes made in the implementation plan.

All submissions must be in PDF format.

Final demo (performed in person between 20-Nov and 27-Nov-2013)

Prepare a demonstration of:

- The design process and overall design,
- Noteworthy design details (things you are proud of),
- The design passing DRC, NCC, ERC,
- The test bench used to verify design functionality,
- The evaluation approach used to measure the quality of the initial and optimized designs, and
- The initial and optimized designs passing functional simulation.

Final presentation (performed in class on 29-Nov-2013 or 3-Dec-2013)

Present your design, evaluation, optimization, and conclusions, and describe any lessons learned in the process. The time allotted and corresponding schedule will depend on the number of project groups.

Final report (submitted electronically, due 3-Dec-2013 at 23h59)

All final reports will be in PDF format, prepared using LaTeX, and maintained using a version-control system. Submit by the due date above instructions for how to access your repository (e.g., SVN, GIT, mercurial), and build your report (e.g., using a Makefile or script).

Your report should, in no more than four pages (0.5" margins, 10 pt font, or similar conference or journal article format), present your design, evaluation, and conclusions, and describe any lessons learned in the process. Provide background (including adequate citations), motivate the problem, describe your solution (including optimization), and present the results of your evaluation. Special care must be taken with figures to ensure they are legible and relevant. No appendices are allowed.

Project Ideas

Propose to do something that your team is interested in doing. Evaluate the resulting area, performance and/or power relative to an initial or baseline implementation. Here are some ideas that are (hopefully) of the right scope.

Extend or revise the MIPS 8-bit processor:

- Pipeline the processor,
- Implement and integrate a new, complex, functional unit (e.g., multiply accumulate),

- Implement and integrate some other new, complex structure (e.g., a cache or branch predictor),
- Re-implement and optimize the data path using a different logic family,
- Re-implement and optimize the data path to use appropriate stage effort.

Build something interesting from scratch, for instance, using the provided standard cell library:

- Compare multiple implementations of a functional unit (e.g., multiplier or adder),
- Compare synthesized and full-custom implementations of something.

Students in the past have performed full-custom design using automatically generated RTL from the Spiral project (http://www.spiral.net).

Feel free to suggest something not mentioned above, provided it is of similar scope!