ECE627: Register-transfer-level Digital Systems

Instructor: Nachiket Kapre (nachiket@uwaterloo.ca)

Motivation

The design and engineering of computing chips is an exciting, and valuable skill that opens the door to careers in the highly-paid, and richly rewarding chip design industry. With the impending demise of Moore's Law, now more than ever, careful bottom-up rethinking of the hardware stacks are necessary. Many chip startups are developing new hardware for machine learning/AI, database acceleration, virtualized network functions, and other domains.

Content

What are the key concepts we must know to design correct, high-performance digital chips? How do we capture the architecture of these chips in a programming language, how do we verify correctness of these designs, and how do we implement these descriptions down to silicon? In ECE627, you will learn RTL design and analysis concepts that are positioned above ECE637: Design of VLSI MOS Integrated Circuits, and below ECE621: Computer Organization. We will cover the entire design flow and implementation on FPGAs as part of this course. We will learn about open-source tools such as yosys and verilator that will help speedup the design process as well as industry-standard FPGA compilation tools such as Quartus and Vivado. There will be a significant project component (project+oral), and a final exam.

Objectives

1. The design of hardware components requires reasoning about a unique form of parallelism and concurrency and operations at a cycle-by-cycle level of abstraction. Hardware description languages such as Verilog, and VHDL provide a programming environment that makes it possible to express this form of parallelism. We will understand the fundamental building blocks of the programming languages necessary to describe hardware.
2. There are commonly used paradigms, and design patterns that we must known when organizing the RTL code. Most digital designs must describe interaction protocols with the outside world, manage state, and capture datapath operation behavior of the digital design. We will learn how to break down a complex design into these components, and learn how to reason about correctness, and understand their translation to hardware primitives. We will use tools such as verilator and Modelsim (verification) along with yosys and Quartus/Vivado (synthesis).

3. It is important to analyze the design for goodness metrics such as resource usage, performance, and power consumption. Most digital design are constrained to fit specific cost targets, and the analysis tools help guide the design process with appropriate feedback. We will learn how to interpret the results of the FPGA CAD tools and incorporate that feedback to redesign our RTL. These compilation iterations are fundamental to digital design and a good designer knows how to direct the tools and generate the desired hardware.
4. A class project will be used to tie together all the key design lessons we learn in this course. The project will be broken down into phases and milestones over the course of the term to guide student progress. In particular, we will look at hardware for machine learning as the theme for the course project in 2018.

Assessment Criteria

Midterm (20% credit): This will be a short written exam to evaluate your understanding of the core concepts of RTL design.

Project (30% credit): Students will work on a prescribed class project and submit a design + report (4 pages) + oral presentation based on the project work.

Final exam (50% credit): Students will have a final exam for the course that will cover RTL design and analysis topics.

Details

Workload: 3 hours of lecture/week.

Project: The class projects infrastructure will be distributed by the end of the third week. Students are expected to work in teams. Student teams who have a well-developed idea may propose their own project that may be allowed at the discretion of the instructor.

Learning Outcomes

At the end of the course, students should be able to: 1. Describe hardware components at the RTL-level, verify these designs, and map these design to an FPGA fabric. 2. Perform engineering tradeoffs in design for area, performance, and power. 3. Operate an FPGA board, and debug designs in real hardware. 4. Document the class project in the language of a technical paper, and deliver a high-quality group presentation.

Pre-Requisites

Undergraduate course on digital hardware design with VHDL or Verilog.

Co-Requisites (not mandatory)

If students are interested in mastering hardware design at all levels of abstraction beyond RTL design, they are encouraged to take **ECE637**, and **ECE621**. If they want to know about FPGA design, they should take **ECE720T6**.

Syllabus

Topic
Introduction to RTL Design
Simulation of RTL Designs
Synthesizing RTL Designs
FPGA Architecture Fundamentals
State Machine Design
Datapath Design
Managing Memories
Timing Analysis
High-Level Synthesis
Formal Verification