ECE 351 - Hardware Description Languages and Prototyping

This 300 level course introduces the student to a *hardware description language* (HDL) and explains its role in the electronic design automation (EDA) environment. Students will learn how to describe, simulate, synthesize, and test logic designs in an HDL. Several designs will be prototyped using Xilinx FPGAs, which are part of the Xilinx Digilab Circuit Boards[1].

This is a formal programming language course that teaches the Verilog HDL. Both structural and register-transfer level (RTL) descriptions[2] are covered. Specific course topics include

- Overview of EDA and the role of HDL
- Verilog data typing and operators
- Verilog RTL and structural descriptions of logic designs
- Simulating Verilog descriptions
- Synthesizing Verilog descriptions
- Constructing testbenches for functional testing
- Using the Programming Language Interface (PLI) to extract information from the simulation environment
- Prototyping with FPGAs

Prerequisite: ECE271 or equivalent, completion of a C or C++ programming course

Text: Samir Palnitkar, *Verilog HDL: A Guide to Digital Design and Synthesis*, Prentice-Hall, 1996 (with CDROM)[3]

Course objectives: Students must demonstrate the ability to

1. model combinational and sequential logic designs in the Verilog Hardware Description language.
2. design a simple testbench to verify the functionality of a Verilog description.
3. use the PLI to create tasks and functions for monitoring simulations
4. synthesize both combinational and sequential designs.
5. prototype a Verilog design in a FPGA.

[1] There are 12 Xilinx FPGA boards available for this class. They are currently secured in the FAB.
[2] RTL descriptions are comprised of both behavioral and dataflow representations.
[3] The CDROM contains a Windows based compiler and simulator. Book cost $\approx$ \$60.

Commentary

The purpose of this proposed course is to teach students how to write Verilog programs that describe logic systems. This is a programming course—not a VLSI design course. Consequently, the following topics will not be covered:

- construction of HDL compilers/simulators
- VLSI design algorithms
- VLSI timing verification
- anything involving formal verification techniques
- construction of test pattern sets that maximize fault coverage
- scan insertion or anything else concerning modification of existing designs to support test
- logic synthesis techniques that render designs with minimal gate-count, area, or power consumption
- switching level design
- CMOS, BICMOS or any other technology
- any part of the design process following synthesis (i.e., layout, floorplanning, routing, or tapeout.)
- anything relating to IC fabrication

All programming exercises are restricted to describing logic circuitry encountered in ECE271: adders, multiplexers, FSMs, and so on. More advanced circuits—e.g., pipelines—are excluded. Testbench construction is limited to functional testing only. VLSI fabrication, manufacturing error detection, timing analysis, formal verification, and the underlying semiconductor physics will be left to other courses.

It is common practice to use FPGAs for rapid prototyping of HDL synthesized designs. The on-hand Xilinx FPGA boards provide our students a unique opportunity to learn a HDL by writing, simulating, synthesizing and finally verifying their results. This is not a FPGA class—the brief introduction to FPGAs given in ECE271 is sufficient background. The goal is simply to introduce the students to the importance of prototyping prior to committing a design to silicon, and to explain how FPGAs are used in that process.

Despite the title of this course, other HDLs—most notably VHDL—will not be discussed. More specifically, no attempt will be made to compare, contrast, or advocate one HDL over another. The choice of Verilog for the proposed course was made for two primary reasons: Verilog is a HDL supported by many EDA vendors and VHDL is adequately covered by other courses in our curriculum.