ECE 571: Introduction to System Verilog for Design and Verification

Catalog Description

Introduction to SystemVerilog: language features to support both design and verification. Good practices for simulation and synthesis, techniques for constructing reusable testbenches. Additional topics may include hardware acceleration and transaction-based verification techniques. Course includes homework and significant final project with presentation. Familiarity with Verilog and finite state machines required.

Credit hours: 4

Prerequisites

ECE 351 or equivalent, or permission of instructor.

Course Coordinator

Mark Faust

Textbooks

Sutherland, Davidmann, and Flake, *SystemVerilog for Design: A Guide to Using SystemVerilog for Hardware Design and Modeling*, 2nd edition, Springer, 2006.

Spear, *SystemVerilog for Verification: A Guide to Learning the Testbench Language Features*, 3rd edition, Springer, 2012.

The course instructor may choose to use a different textbook. Please check with your instructor before purchasing.

Learning Outcomes

At the end of this course, students will be able to:

1. Use SystemVerilog to create correct, efficient, and re-usable models for digital designs
2. Use SystemVerilog to create testbenches for digital designs
3. Understand and effectively exploit new constructs in SystemVerilog

Topical Outline

- Introduction: Significance/role of SystemVerilog, evolution of SystemVerilog, Tools and vendor support
- Declaration spaces. Packages, unnamed blocks, time values and precision
- Literals and built-in data types, macros, constants, signed and unsigned modifiers, type casting, static and automatic variables
- User-defined and enumerated types, typdefs
- Arrays, structures, unions, foeach loops, dynamic, associative and sparse arrays, strings
- Procedural blocks, tasks, functions
- Procedural statements, new loop constructs, enhanced case statements
- Modeling finite state machines
- Design hierarchy
- Interfaces
- Behavioral and transaction level modeling
- Verification basics
- Object oriented programming
- Randomization
- Assertions

Course Structure

Two 100-minute lectures per week. Weekly reading and homework assignment.

Relevant Program Outcomes

The following program outcomes are supported by this course:

(a) An ability to apply the knowledge of mathematics, science, and engineering.

(c) An ability to design a system, component, or process to meet a range of informal to formal descriptions/specifications.

(e) An ability to identify, formulate, and solve engineering problems.

(i) Recognition of the need for, and an ability to engage in life-long learning.

(k) An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

Updated: 7/05/17