ECE 6095: Special Topics in Computer Engineering: VLSI Design Verification and Testing

Course:

ECE 6095: VLSI Design Verification and Testing, Fall 2009, 3 Credits
Time: Tue (3:00-5:30pm)
Room: ITE 330

Instructor:

Dr. Mohammad Tehranipoor
Office: ITE 441
Phone: 860-486-3471
Office hours: Tuesday 2-3pm or by appointment
Email: tehrani@engr.uconn.edu
URL: http://www.engr.uconn.edu/~tehrani/teaching/test/index.html

Textbook:

M. Bushnell and V. Agrawal, *Essentials of Electronic Testing for Digital, Memory, and Mixed-Signal VLSI Circuits*, Kluwer Academic Publishers, 2000.

Reference:

M. Abramovichi, M. Breuer and A. Friedman, *Digital Systems Testing and Testable Design*, IEEE Press, 1999.
L. T. Wang, C. W. Wu and X. Wen, *VLSI Test Principles and Architectures*, Elsevier, 2006.

Goals:

Introduction to the concepts and techniques of VLSI (Very Large Scale Integration) design verification and testing, details of test economy, fault modeling and simulation, defects, Automatic Test Pattern Generation (ATPG), design for testability, Scan and Boundary scan architectures, built-in self-test (BIST) and current-based testing. Tools are used (in homeworks and projects) for ATPG, DFT, test synthesis and more. Students will use commercial DFT tools such as TetraMax, DFT Compiler and Analyzer, power analysis and management tools such as PrimePower and PowerMill from Synopsys.

Topics to be covered:

- Introduction: Verification vs. Testing, Need for testing, Level of testing, Cost of testing, Roles of testing

- Test Process and Equipment: Types of testing, Manufacturing test, Burn-in and stress test, Functional test, Automatic test equipment (ATE), Electrical parameter testing, DC parameter testing, AC parameter test,
- Test Economics: Basics of cost analysis, Benefit-cost analysis, Economics of design-for-testability (DFT), VLSI chip yield, Defect level,
- Logic and Fault Modeling: Logic modeling, Model types, Models at different levels of abstractions, Fault modeling, Common fault models, Stuck-at-faults, Transistor (switch) faults
- Fault Simulation: Usage of fault simulators, Fault simulator in a VLSI design process, Fault simulation algorithms: Serial, Parallel, Deductive, Concurrent, Fault sampling
- Combinational ATPG (I): Structural vs. functional test, Definition of ATPG, Exhaustive algorithm, Random pattern generation, Boolean difference symbolic method, Path sensitization method, Computation complexity
- Combinational ATPG (II): Major ATPG algorithms, D-Algorithm, PODEM
- ATPG Systems and Testability Measures: ATPG systems, Static and dynamic compaction, Fault coverage and efficiency, Testability analysis, SCOAP measures, Controllability measure, Observability measure,
- Sequential Circuit ATPG: Time frame expansion, Nine-valued logic, Drivability, Complexity of ATPG, Test generation system,
- Functional Testing: Structure independent approach, Structure dependent approach, Microprocessor testing,
- Delay Test: Path delay fault testing, transition fault testing, pattern generation, Scan-based delay fault test
- Design-for-Testability (DFT) (I): Definition, Ad-hoc DFT methods, Scan design, Scan flip-flop, Muxed-DFF, LSSD, Scan test vectors, Multiple scan registers, Hierarchical scan,
- Design-for-Testability (DFT) (II): Partial scan architecture, Scan flip-flop selection methods, Cyclic and acyclic structures, Scan-hold flip-flops
- I_{DDQ} Current Testing: History and motivation, Basic principle of I_{DDQ} testing, Fault detected by I_{DDQ} tests, Limitations of I_{DDQ} testing
- Memory Testing: Motivation, Functional model of a memory, Fault models, March tests
- Built-In Self-Test (BIST) (I): Motivation, BIST definitions, BIST process, BIST pattern generation, BIST response compaction, Aliasing definition
- Built-In Self-Test (BIST) (II): Motivation, Built-in logic block observer, Test/clock systems, Test/scan systems, Test point insertion
- Boundary Scan: Motivation, Bed-of-nails tester, Boundary scan hardware, JTAG standard (IEEE 1149.1), Elementary scan cell. Test access port (TAP) controller, Boundary scan instructions
- Nanometer Technology design and test problems, Future research challenges

Grading

Final grades will be derived using following scale:

- Class Participation/Exercise/Discussion 5%
- Homework Assignments 25%

- Project 20%
- Quizzes 10%
- Midterm Examination* 40%

* The midterm exam will be comprehensive and closed book.

Homework Assignments

There will be assignments almost **one** every **two** weeks (Total of **5-6** assignments). The assignments will mostly be selected from the text book but there will also be questions or problems defined by instructor. Some assignments may need results obtained from **Synopsys** tools and it is your responsibility to learn these tools. The related tutorials will be distributed before posting the homeworks. I assume all the students know how to work with **Unix/Linux** operating systems and editors. No **TA** will be assigned to this course. However, a PhD student from my lab will be assigned to answer your questions when working with Synopsys tools.

Normally there will be **firm deadlines** associated with the assignments. In any case, students will be encouraged to complete the task. After all, the goal here is to learn as much as possible and a deadline should not imply that assignment is no longer due and the grade is lost.

Project

There will be a **final project**. The project will involve Synopsys tools to be used for HDL Compilation and Simulation, Synthesis, Scan insertion, Logic BIST and ATPG. More than one project will be defined and students can choose only one (there may be group projects). Some projects may require C/C++ or Perl programming. The information on project will be provided during the course at an appropriate time. The projects are to be due last lecture Right after the final exams. The **projects report** must be clear and well organized.

Class Participation

Classroom participation is an important aspect of learning. It is a learning process for both the students and the instructor. **Questions**, even if they seem simple, should be asked. Some questions will generate new ideas or start discussion.

General Policies

Classes will be conducted as stated in university schedule. Follow the University calendar for add/drop dates. Assignments will be individual responsibility. Sharing thoughts and discussions on assignments are encouraged. However sharing of answers, programs or part of program is considered cheating. Cheating will lead to an automatic **FAIL** (no exceptions, no withdrawals).