CSE 591: Digital Logic Synthesis and Verification Algorithms
Fall 2006
Class Meeting: MW 1:40 PM - 2:55 PM, BYAC 220

Instructor:	Sarma Vrudhula
email:	vrudhula@asu.edu
phone:	(480) 965-4748
Office:	BYENG 486
Office Hours':	3:00 - 5:00 MW

Important Dates:

Test I:	Sept. 27, 2006
Test II:	Nov. 6, 2006
Final:	Dec. 13, 2006

Grading

Test I:	15%
Test II:	20%
Final:	25%
Assignments:	40%

It is the advances in design automation and CAD algorihtms for VLSI that have made it possible for designers to fully exploit the advances in IC scaling. The high performance general purpose processors, ASCIs and FPGAs that we have today would not have been possible without the development of methods for automating nearly all phases of the design flow.

This graduate level course provides the theoretical underpinnings of digital VLSI CAD tools for logic synthesis and verification. The theory and algorithms covered in this course have been the result of four decades of research and development, and have been incorporated into many commercial tools over the past two decades.

The emphasis of this course will be on translating the mathematical results into efficient algorithms. Students will gain experience and deeper understanding of the methods by implementing some of the algorithms using a high-level programming language with hooks into the pre-defined datastructures and routines to manipulate them.

Main topics include:

1. Discrete mathematical stuctures: relations, partial orders, lattices, Boolean algebras.
2. Boolean and pseudo Boolean functions as sets. Representation of sets using decision diagrams.
3. Exact two level minimization: minimization over product spaces, set covering over lattices, implicit enumeration using branch and bound. Heuristic methods.
4. Multi-level logic optimization: algebraic methods based on Kernels and co-kernels.
5. Decomposition theory and algorithms. BDD based decompostion.

6. Technology Mapping: Mapping logic networks to network of cells from a cell library. Delay and area optimality.
7. Equivalence checking of combinational circuits
8. Models of sequential systems, state minimization, state assignment
9. Equivalence checking of sequential machines by state space exploration using symbolic image computation
10. Combinational and sequential timing analysis and optimization

Sources: The topics covered in this course can be found spread over several textbooks. None of them are cover all the topics in this course. A basic textbook is "Logic Synthesis and Verification Algorithms" by Hactel and Somenzi, is available onlilne. We will use this book as a guide. Other sources include lecture notes, and papers published in conferences and journals. These will be made available online.

Workload: The course will stress reading lots of papers. Challenging homework assignments will be given to deepen your understanding of the theory. Some of these will include small programming exercises that will involve interfacing to existing packages (in C or C++).

Prerequisites: A undergraduate course covering the essentials of logic design (Boolean algebra, SOP and POS forms, design of basic combinational and sequential logic components, combinational function minimization, finite state machine design, etc.) is essential. This course also requires some mathematical maturity, especially in discrete mathematics. However, all that is needed for the course will be covered in the lectures. Basic knowledge of data structures, and experience in programming will be helpful.

Broader Impact: While the basic mathematical concepts and algorithms will be taught in terms of logic design, optimization and verfication, nearly everything one learns in this class can be applied to a wide range of discrete function representations and optimization problems that arise in electrical engineering, computer engineering and computer science. Certainly, this will benefit any graduate student in CSE or EE.