046880: Logic Design Automation of VLSI Circuits

Course abstract:
The course is dedicated to studying the computer-aided VLSI logic design. As system complexity grows, VLSI designers are faced with problems related to validating correctness, reducing design effort, and optimizing speed, area, power and other parameters. The course presents the design methodologies and techniques, as well as a variety of design automation algorithms. The generic design flow is covered top-down. The course provides a broad survey of the electronic design automation (EDA) field, with a focus on front-end logic design. Current and future research areas are outlined throughout the course.

Course Objectives:
Students will be exposed to the discipline of electronic design automation, using an analytical approach based on Boolean algebra, graph theory and optimization techniques. After completing the course, students will have a deep understanding of the VLSI logic design flow, and algorithms used throughout design automation tools. Design representations, data structures, algorithms and heuristics will be practiced in programming assignments. As a result of the course, students will gain insight into EDA, and develop ability to read the professional literature.

Required background:
Digital systems, basic circuit & VLSI concepts.
Programming (in C++ language) and data structures.

Prerequisites:
(044268 Intro. to algorithms and data structures) OR (234246/7 Graph Algorithms 1)
AND
(044262/234262 Logic design) OR (044252/234252 Digital systems and computer organization)

Or instructor's consent.

Recommended:
(046237 Intro. to VLSI) OR (236354 VLSI Circuit Design)
AND
(044101 Intro. to Software Systems) OR (234122 Intro. to Systems Programming)

Grading:

Grades will be based on:

- Homework 50% mandatory
- Final exam 50% mandatory

In case of prohibition exams on campus– the exams will take place online.

Bibliography:

Textbook:

1) G.DeMicheli, "Synthesis and optimization of digital circuits", McGraw-Hill, 1994.

Auxiliary books:

2) M.Sarrafzadeh and C.K. Wong, "An introduction to VLSI physical design", McGraw-Hill, 1996.
3) G.D. Hachtel and F. Somenzi, “Logic Synthesis and verification algorithms”, Kluwer academic publishers, 1996.
4) S. H. Gerez, “Algorithms for VLSI design automation”, Wiley, 1999

Course plan:

Note:

1) The table below lists multiple options for exercises in class– a subset will be selected each time the course is taught.
2) Student will have an average homework load of 4-5 hours per week. For example: turn-in 3 regular homework (H), do 3 programming assignments (P).
   Homework will be done in groups of 2 students to encourage interaction.

Week	Date	Lecture topics	Exercise topics	Homework assignment options

1	**22/10**	**Introduction**
		Evolution and trends in VLSI CAD. Design process, representations, flow and methodology. CAD Tool types and what they do. VLSI flow overview. Decision and optimization problems in CAD. Example methodologies.
			● Quick coding overview
			  ● About C++ STL containers
			  ● Makefile example
			● Basic Graph Algorithms:
			● BFS/DFS comparison
				● W0: setup your environment for Linux C++ programming

2	**29/10**	**Digital hardware modeling** Spec vs. Description; Structure modeling: Logic networks, connectivity & netlists. Hypergraphs. Connectivity data structures.	● Course connectivity model API ● Flattener example	● W1: Connectivity data structure, recursive exploding and flattening **(3ww by 26/11)** ●
3	**5/11**	**Dynamic verification, Simulators** Cycle-based simulation; logic relaxation; event-driven simulator kernels. Value sets. Zero/unit/variable delay simulation & issues. Coverage metrics; instrumentation	● **No Tutorial**	
4	**12/11**	**Static logic verification** (formal) Advantages of static vs. dynamic verification; Equivalence checking: RTL-to-gates, gates-to-gates, LVS Advanced Boolean algebra: Shannon expansion, cofactors and Unate Recursive Paradigm	● Event queues and scheduling in simulators ● Switch level simulation ● Comparisons of zero-delay, unit-delay, variable-delay logic simulation	● W2: Event driven interpreted simulation. **(5ww by 24/12)**
5	**19/11**	**Binary Decision Diagrams** (BDDs) Representation, properties, variable ordering heuristics. Usage for combinational logic verification;	● Recursive complementation ● Properties of cofactors ● Equivalence checking with don't cares	● D1: Working with cube lists: PCN, Cofactors, recursive tautology checking
6	**26/12**	**BDD operations** Apply, restrict, ITE. BDD forests, BDD drawbacks	● Building shared ROBDD ● Complemented edges	● D2: Pencil & paper questions on BDDs
7	**3/12**	**SAT** Principles of SAT solvers, Example problems and application	● ITE algorithm	● D3: BDD application e.g. Constant propagation or matching nodes
8	**10/12**	**Sequential Verification and Formal Model Checking** The cross product machine. Reachability analysis; Symbolic checking using BDDs Reduction of logic	● Minisat	● W3: Gate to Gate Equivalence via minisat **(4ww by 14/1)**

		Inferring well know structures		
9	**24/12**	**Logic synthesis 1** 2- level minimization: exact vs. Heuristic, Espresso operations; on/off/dc sets; multiple-output functions; Covering problems and algorithms	● Optimization Techniques overview	
10	**31/12**	**Logic synthesis 2** Principles of Multilevel minimization Technology mapping; cell libraries; Behavioral synthesis: resource allocation & binding	● Simulated Annealing	
11	**7/1**	**Static timing analysis 1** Clocking schemes , sampling elements; definitions of timing parameters and delays; Delay modeling (block/gate/device level); Path analysis algorithms. Min/Max delay verification	● Dynamic Programming	
12	**14/1**	**Static timing analysis 2** False paths, sensitization, pessimism, DSM effects: clock skew, noise effects	● Timing Analysis Paths ● Enumeration and sorting	D4: Static timing analysis (bonus)
13	**21/1**	Getting physical (or guest lecture)	● Delay Modeling	