

\*NOTE: “CSEE” refers to a course cross-listed between CS and EE Departments.

**Course:** CSEE E6861y, Computer-Aided Design of Digital Systems  
**Time:** Thursday, 4:10–6:00 p.m.  
**Location:** 227 Mudd Building  
**Credits:** 3 units

**Instructor:** Steven M. Nowick  
**Office:** 508 Computer Science Building  
**Email:** nowick@cs.columbia.edu  
**Office Hours:** Tuesday 2:00–3:00 p.m., Wednesday 2:45–3:45 p.m.  
**Phone:** (212) 939-7056

Extra individual appointments can be made if necessary; send me email.

**TA:** David Jew  
**Office:** TA Room, 1st floor Mudd (see <http://ta.cs.columbia.edu/tamap.shtml>)  
**Email:** drj2109@columbia.edu  
**Office Hours:** Tuesday 11:00a.m.–12:00 p.m., Wednesday 4:00–5:00 p.m.  
**Phone:** 854-4916 (*during office hours only*)

The course is suitable as a technical elective for CS, Computer Engineering, and EE MS and PhD students. For Computer Engineering MS students, it is a “core” course (see SEAS bulletin). For CS PhD students, it counts towards the Systems elective area.

**Prerequisites:** (i) basics of digital logic, such as: CSEE 3827 (half course in digital logic, half course in computer organization) or equivalent; and (ii) a basic course in data structures and algorithms, and familiarity with programming (CS 3133, 3134, 3137, 3139 or equivalent). Students lacking any prerequisite should see the instructor.

(*Note:* No VLSI or EE circuits background is required!)

### **Course Description:**

This course is an introduction to modern computer-aided design (“CAD”) of digital systems. This is an important and highly-active research area, which is also a key driver of the growth of the microelectronics industry.

The course is a nice blend of three areas: (i) optimization algorithms, (ii) digital design, and (iii) software tools and applications. It is suitable for students with a wide range of interests: from those more interested in applied theory and algorithms, to those more interested in digital design, or software tool development.

The course systematically covers some of the major automated synthesis steps used in modern CAD tools: starting from a user’s high-level architectural specification for an entire system, down to optimized low-level digital circuits. A major focus is on how to synthesize and optimize large, complex digital designs (*with tens of thousands to millions of gates*) using efficient and automated techniques. An additional focus is how optimization techniques are targeted to different cost functions, e.g. delay, area, or power.

Many of the techniques we will cover are directly used in commercial CAD tools, or else have influenced their development.

*Topics include:* modern algorithms (both exact and heuristic) for 2-level logic minimization (espresso, espresso-exact); multi-level logic optimization (heuristic transformation steps, creating design scripts); technology mapping (i.e. optimal mapping of logic gates to VLSI library cells) to target area, delay and power;

introduction to physical design (circuit partitioning, place-and-route); sequential retiming (i.e. moving of registers in a pipelined system for optimal clock rate and area); introduction to modern system-level specification (register-transfer Level (RTL) modelling, control-dataflow graphs (CDFG's), algorithmic state machines (ASMs)); architectural synthesis and optimization (scheduling, resource allocation and sharing); modern compact Boolean data structures (binary decision diagrams [BDD's]) and their application to circuit synthesis and verification (i.e. equivalence checking); satisfiability (SAT) solvers and their applications; and advanced Boolean multi-level optimization (using CDC's and ODC's). Two small to medium projects are included, involving software programming and design tool creation and optimization.

For some of the assignments, you will use some widely-influential academic CAD tools, such as the *UC Berkeley SIS package*, the recent *UC Berkeley ABC package*, and others, and perform detailed experiments and evaluation on benchmarks. For the small projects, you will have the opportunity to develop your own tools.

Some of the algorithmic techniques you will learn have wide application to a variety of domains: iterative improvement, hill climbing,unate and binate covering problems, dynamic programming, partitioning techniques, and simulated annealing.

When you have completed the course, you will have a good handle on modern research aspects of digital CAD (i.e., the underlying optimization algorithms used to automatically design digital systems), as well as gain some practical hands-on experience in using existing CAD packages.

**NOTE:** This is **not** primarily a project/lab course; while you will use CAD tools and do two small projects, the focus will be on the techniques behind the tools. Also, you do not need to be an experienced digital designer to take this course: you should simply have a basic background in digital logic, data structures and algorithms.

#### **Required Texts:**

Giovanni De Micheli, *Synthesis and Optimization of Digital Circuits*, McGraw-Hill (1994).

*This book is currently out of print. However, it is available online in used, new and international editions.*

Additional reading material, handouts and slides will be available on the class web page. These include recent research and survey/tutorial papers.

#### **Other Background Texts (optional):**

Srinivas Devadas, Abhijit Ghosh and Kurt Keutzer, *Logic Synthesis*, McGraw-Hill (1994).

Daniel D. Gajski, *Principles of Digital Design*, Prentice Hall (1997).

Gary D. Hachtel and Fabio Somenzi, *Logic Synthesis and Verification Algorithms*, Kluwer Academic (1996).

A.V. Aho, J.E. Hopcroft, J.D. Ullman, *Data Structures and Algorithms*, Addison-Wesley (1983)

Robert K. Brayton, Gary D. Hachtel, Curtis T. McMullen and Alberto L. Sangiovanni-Vincentelli, *Logic Minimization Algorithms for VLSI Synthesis*, Kluwer Academic Publishers (1984).

Randy H. Katz, *Contemporary Logic Design*, Benjamin/Cummings Publishing Company, Inc., Redwood City, CA (1994).

Edward J. McCluskey, *Logic Design Principles: with emphasis on testable semicustom circuits*, Prentice-Hall, Englewood Cliffs, NJ (1986).

Charles H. Roth, Jr., *Fundamentals of Logic Design*, 5th edition, West Publishing Co., St. Paul, MI (1992).

John F. Wakerly, *Digital Design: Principles and Practices*, 4th edition, Prentice-Hall, Upper Saddle River, NJ.

*All books are on reserve in the Monell Engineering Library (422 Mudd).*

**Homework.** There will be several homework assignments throughout the course. These assignments will include both written problems and small exercises with public domain CAD packages, including the UC Berkeley *SIS* package, the UC Berkeley *ABC* package, and others.

**CAD Mini-Projects:** There will be two CAD mini-projects, in addition to the above homework. Each will involve about three weeks of work. Details will be provided later in the course. You will be given an option of several choices of topic for the second project (and I may consider topics which you suggest).

**Final Exam.** Any material covered in assigned book readings, handouts, homework, lectures or discussion sections may appear in the final exam questions.

**Grading:** Course grades will be based on homework (about 30%), the midterm mini-project (about 15%), the final mini-project (about 20%), and the final exam (about 35%).

**Recitation Sections:** Occasional recitation sections may be scheduled to introduce CAD tools. They will be led by the instructor and/or teaching assistant. Stay tuned for further announcements.

**Class Attendance:** You are responsible for the material regardless of your attendance in class. Regular class attendance is the best way to insure that you learn the material. Lectures may often diverge from the book.

**Class Participation:** It is expected that all students will participate to some degree in the class, whether raising question or answering them.

**Late Policy:** If you hand in something after the due date without the explicit approval of the instructor or the TA, you might receive zero credit. Homeworks are due at the beginning of class on the assigned due date. Under real emergencies, extensions might be given by the instructor, if you contact me in advance.

**Cooperation on Homework and Exams:** Collaboration on homework solutions, or sharing or copying of solutions, is *not* allowed. Of course, no cooperation is allowed on exams. *The department academic honesty policy is listed in:* <http://www.cs.columbia.edu/academics/honesty>. It is your responsibility to be aware of this policy, and to conform to it.

**Handouts:** Additional copies of handouts will be available on the class web page. Hardcopy handouts are available from the TA.

**Class Web Page:** The URL of the class web page is: <http://www.cs.columbia.edu/~cs6861>. This page will contain copies of handouts, homework assignments and solutions, and other important information. You should read it regularly.