UCLA Computer Science Department: CS 151C
Design of Digital Systems
Winter 2010, MW 2:00-3:50 pm, BH 5440
Office hours: MW 4-4:50 pm or by appointment

Prof. Miloš D. Ercegovac
4731-H Boelter Hall
825-5414 ; e-mail: milos@cs.ucla.edu

Course material: Lecture viewgraphs and notes; papers; misc. design tools - URLs on the HSSEAS CourseWeb
Optional Textbook: M.D. Ercegovac, T. Lang and J. Moreno, *Introduction to Digital Systems*, John Wiley & Sons, New York, 1999. Digital version available at the CourseSmart site.

Grading: Homeworks 10%, Class Presentations 20%, Midterm 20%, Projects 50%. (No final)

This is an elective course for seniors in CS, CSE and EE. It deals with the hardware design of application-specific and general-purpose digital systems. We cover their high-level specification at algorithmic level, organization at the RTL (Register-Transfer Level), technology-dependent design styles, and main hardware implementation constructs and techniques. We also discuss digital arithmetic units, reconfigurable systems, pipelining, and system timing. Students work on peer-reviewed team design projects to complement the lectures and enhance their understanding of design tools and methods. In addition, there is a reading list of suggested papers and students give presentations and lead discussion on a selected paper.

TOPICS

1. **Overview of digital systems design.** Specification and implementation: hierarchical approach. Analysis and design. Behavioral, structural and physical models. Levels of implementation. Implementation alternatives: programmable logic devices (PLDs and FPGAs), standard cells, custom design. Design process flow and CAD tools. Hardware description languages.
2. **Register-transfer level (RTL) systems.** Computational models and execution graphs: sequential, group-sequential, concurrent. Data and control subsystem organizations. Specification and implementation of RTL systems. Analysis and design of RTL systems. [Chap. 13.]
3. **Data and control subsystems.** Data subsystem: storage modules, functional modules, and datapaths. Control subsystem: hardwired and microprogrammed approaches. Finite state machines as controllers. [Chap.14.]
4. **Programmable modules and reconfigurable systems.** Programmable logic arrays (PLAs), programmable sequential arrays (PSAs), read-only memories (ROMs), and field-programmable gate arrays (FPGAs). Reconfigurable systems: organization, design, and uses. [Chap. 12.]
5. **Arithmetic operations and modules.** Case studies: Algorithms and implementations for fast addition, multiplication, and division.
6. **Enhancing performance.** Pipelining: Partitioning and staging. Latch design. Retiming. Timing and control.
7. **System timing.** Synchronous, asynchronous, and combined approaches. Clock distribution. Synchronization problems: clock skew, arbitration, and metastability. Self-timed systems.

Course Organization :

- My lectures; in-class discussions last 15 minutes - get ready (there will be no separate discussion section).
- Your presentations (30 minutes) related to selected papers or papers of your choice (by 2-student teams): select a partner and a paper to present by Jan. 27; presentations on Feb. 17.
- 4 homeworks
- Midterm (no final)
- Team projects (see Project Guidelines for details). Proposals due: Feb. 8; project proposal presentations in class on Feb. 10. Final project presentations March 10 and 12.
- All work submitted via CS151C CourseWeb site.

CS151C-W2010:Tentative Schedule

Week	Monday	Wednesday
1	1/4: Topic 1	1/6: Topic 2
2	1/11: Topic 2	1/13: Topic 3
3	1/18: No Class	1/20: Topic 4
4	1/25: Topic 4	1/27: Topic 5; select reading
5	2/1: Topic 5	2/3: MIDTERM
6	2/8: Topic 5; project proposals due	2/10: Project presentations
7	2/15: No Class	2/17: Reading presentations
8	2/22: Topic 6	2/24: Topic 6
9	3/1: Topic 7	3/3: Topic 7
10	3/10: Presentations	3/12: Presentations