CSE 370 Introduction to Digital Design Spring 2008

Lecture: MEB 103

MWF 10:30-11:20am

Labs: CSE 003
Section AA T 9:30-12:20
Section AB W 3:30-6:20

Lecture 1: Introduction

PEOPLE

Instructor: Yoky Matsuoka

yoky@cs.washington.edu

Office hours: Fridays 11:20-12:15

Office: CSE 650

PEOPLE

TAs: Nikhil Subramanian and Ray Li

niksubr@ee.washington.edu

Office Hours: Monday 1:30 – 2:20pm

Office: EEB 307L

ruiboli@cs.washington.edu

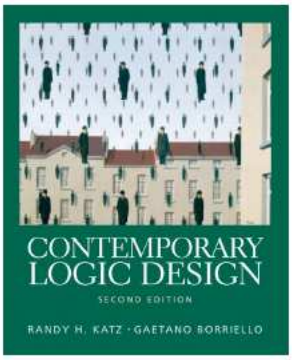
Office Hours: Wednesday 12:30 – 1:30pm

Location: CSE 003

TEXT

Contemporary Logic Design (2nd Edition)

- **Randy H. Katz, U. California, Berkeley and Gaetano Borriello, U. Washington, Seattle**



CLASS STRUCTURE

- **Lectures:** There will be 27 lectures. Attendance and participation of all of them is strongly encouraged and expected.
- **Laboratory:** There will be 9 weekly lab assignments (the last assignment spans 2 weeks). Although you'll be able to use the lab all week, attendance at one of the scheduled times is very important as that is when the TAs will be available. We will use them to reinforce key concepts. You should attend the session for which you are registered. With permission of the TA, you can attend the other section in case of unusual circumstances.
- **Assignments:** There will be 9 weekly homework assignments. They will be based on topics covered in lectures. There will be also reading assignments from the Contemporary Logic Design (2nd edition) text each week which is critical to keep up with the class materials.
- **Exams:** There are two in-class midterms (4/23 and 5/21) and one final exam during finals week.

GRADING

- **30% homework assignments**
  Homework assignment is due at the beginning of the class (10:30am) in class. 10% penalty is applied 24 hours late, and 20% penalty is applied 48 hours late. After 10:30am 2 days after the due date, the solution will be posted and assignment will no longer be accepted.
- **20% lab assignments**
  The lab grades are based on completion checked by the TAs. Don't fall behind because each lab is worth more than 2% of your grades!
- **15% for each midterm (so 30% total)**
- **20% final exam** (cumulative but strong emphasis on materials after both midterms)

CLASS POLICY

- ***Collaboration***
- Unless specifically stated otherwise, we encourage collaboration on assignments, provided (1) you spend at least 15 minutes on each and every problem alone, before discussing its general concepts with others, (2) you only discuss general concepts or related examples - not the specifics of a problem on the assignment, and (3) you write up each and every problem in your own writing, using your own words, and understand the solution fully. Copying someone else's work is cheating (see below), as is copying the homework from another source (e.g., prior year's notes, etc.). Please write down the name of classmates you collaborated with at the top of your assignment.

- ***Cheating***
- Cheating is a very serious offense. If you are caught cheating, you can expect a failing grade and initiation of a cheating case in the University system. Basically, cheating is an insult to the instructor, to the department and major program, and most importantly, to you and your fellow students. If you feel that you are having a problem with the material, or don't have time to finish an assignment, or have any number of other reasons to cheat, then talk with the instructor. Just don't cheat.
- To avoid creating situations where copying can arise, never e-mail or post your solution files in public directories. You can post general questions about interpretation and tool use but limit your comments to these categories. If in doubt about what might constitute cheating, send the instructor e-mail describing the situation.

TENTATIVE CLASS SCHEDULE

Week	Monday		Wednesday		Friday	
1 Lab1	3/31	Introduction	4/1	Binary number systems	4/3	Boolean algebra
2 Lab2	4/7	Logic gates/truth tables	4/9	Canonical forms *Homework1 due*	4/11	Boolean cubes
3 Lab3	4/14	Karnaugh maps	4/16	Logic minimization *Homework2 due*	4/18	Verilog
4 Lab4	4/21	Multi-level logic *Homework3 due*	4/23	**MIDTERM1**	4/25	Multiplexers
5 Lab5	4/28	Structured logic	4/30	Adders *Homework4 due*	5/2	Sequential Logic
6 Lab6	5/5	Flip flops	5/7	Sequential Verilog *Homework5 due*	5/9	State Diagrams
7 Lab7	5/12	Finite state machines (FSM)	5/14	FSM *Homework6 due*	5/16	FSM
8 Lab8	5/19	Computer Organization *Homework7 due*	5/21	**MIDTERM2**	5/23	Computer Organization
9 Lab9	5/26	NO CLASS Memorial Day	5/28	State minimization	5/30	Sequential systems *Homework8 due*
10	6/2	State encoding	6/4	FPGAs *Homework9 due*	6/6	Review

WHY STUDY LOGIC DESIGN?

- Obvious reasons
  - this course is part of the CS/CompE requirements
  - it is the implementation basis for all modern computing devices
    - building large things from small components
    - provide a model of how a computer works
- More important reasons
  - the inherent parallelism in hardware is often our first exposure to parallel computation
  - it offers an interesting counterpoint to software design and is therefore useful in furthering our understanding of computation, in general

WHAT IS LOGIC DESIGN?

- **What is design?**
  - given a specification of a problem, come up with a way of solving it choosing appropriately from a collection of available components
  - while meeting some criteria for size, cost, power, beauty, elegance, etc.
- **What is logic design?**
  - determining the collection of digital logic components to perform a specified control and/or data manipulation and/or communication function and the interconnections between them
  - which logic components to choose? – there are many implementation technologies (e.g., off-the-shelf fixed-function components, programmable devices, transistors on a chip, etc.)
  - the design may need to be optimized and/or transformed to meet design constraints

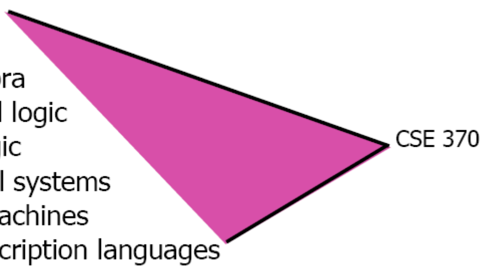
WHAT WILL WE LEARN IN THIS CLASS?

- The language of logic design
  - Boolean algebra, logic minimization, state, timing, CAD tools
- The concept of state in digital systems
  - analogous to variables and program counters in software systems
- How to specify/simulate/compile/realize our designs
  - hardware description languages
  - tools to simulate the workings of our designs
  - logic compilers to synthesize the hardware blocks of our designs
  - mapping onto programmable hardware
- Contrast with software design
  - sequential and parallel implementations
  - specify algorithm as well as computing/storage resources it will use

SCOPE OF CSE 370

- Physical devices (transistors, resistors, wires)

- Switches
- Truth tables
- Boolean algebra
- Combinational logic
- Sequential logic
- State in digital systems
- Finite-state machines
- Hardware description languages



- Register-transfer description
- Concurrent abstract specifications

PREPOSITIONAL LOGIC PUZZLE

- Jim has built up a $800 gambling debt. He has to travel from New York to Las Vegas to pay it back, or face a mafia hit job. A round-trip plane ticket costs $150, and a bus ticket costs $75. Jim has $1,000.
- What is the most flexible and accurate description of what Jim's options are, in order to avoid angering the mafia?

POSSIBLE ANSWERS

- Jim can buy a bus ticket
- Jim can buy a plane ticket
- Jim can buy a bus ticket or a plane ticket
  - If he buys both, he's in trouble
- Jim can buy either a bus ticket or a plane ticket, but not both

PREPOSITIONAL LOGIC

- System for assigning truth values (true or false) to statements that include basic assumptions and logical connectives, like OR, AND, and IMPLIES
- Mathematical basis for the stateless part of digital hardware design

ENCODING

- Alice and Bob are spies. Bob is driving a car with tinted windows and two foreign dignitaries in the back. He wants to let Alice know who is in the car by signaling her with his fingers. Before the operation they knew that in the left seat could only be Mr D, Ms E or Mr F, and in the right seat could only be Ms G, Mr H, Ms I, Mr J or Ms K.
- How few fingers can Bob get away with using?

TWO POSSIBLE ENCODINGS

- Use one finger on the right hand to represent each of the right seat possibilities G, H, I, J, K, and one finger on the left hand to represent each of the left seat possibilities: A, B, C
  - # of fingers: 8
- Recognize that there are 15 possible pairs of dignitaries: AG, AH, ..., BG, ..., CK
- Use a unique pattern of raised fingers to represent each possible pair
  - # of fingers: 4

DISCRETE DATA

- Computers must encode all information (numbers, colors, tones, etc.) as sequences of 1's and 0's
- All discrete data can be so encoded
- Different encodings are more or less efficient, depending on what you want to do with the data

FINITE MEMORY

- A mysterious government agency (MGA) puts Jill on top of a mountain and tells her to report back when she has observed four days in sequence that are rainy, rainy, sunny and cloudy
- Unfortunately, Jill can't remember the weather from day to day
- The only material Jill is given are 5 index cards on which she can write anything before she embarks on her mission
- Can she pull it off?

ONE SOLUTION

- Jill can write down a prefix of the sequence she is looking for on each of the cards (None, R, RR, RRS, RRSC)
- On each day she observes the weather and the card that is currently on top of the pile, and puts a new card on top of the pile

FINITE STATE MACHINES

- Computers use finite state machines, which can remember some fixed number of bits of data
- On receiving each new input, a finite state machine can look at the input, the data currently in its memory and update the memory to some new pattern