Lecture 1: CSE 370 Introduction

- Welcome!!!
- Instructor: Yoky Matsuoka

  http://www.cs.washington.edu/homes/yoky/
- TA: Brian Dellon, Vince Zanella, undergrad TA TBA

- Class Webpage

http://www.cs.washington.edu/370

Where to find help

- Yoky Matsuoka
  - Office: CSE 650
  - Office hours: Mondays and Fridays 8:45-9:30am
  - Email: yoky@cs.washington.edu
- TA: Brian Dellon
  - Office hours: Tuesdays 4-5:30pm in CSE650
  - Email: btd@cs.washington.edu
- TA: Vince Zanella
  - Office hours: Mondays 4-5:50pm in
  - Email: vincez@cs.washington.edu
- TA: TBA
- http://www.cs.washington.edu/370
- Will set up a class email list, anonymous, TA, etc.

Lectures and Labs

- Lecture: MWF 10:30-11:20am, Gowen Hall 201
  - Introductory course in digital logic and its specification and simulation. Boolean algebra, combinatorial circuits including arithmetic circuits and regular structures, sequential circuits including finite-state-machines, use of programmable logic devices. Simulation and high-level specification techniques are emphasized.
  - Please ask lots of questions in lecture.
- Labs:
  - section AA T 9:30-12:20, CSE 003
  - Section AB W 3:30-6:20, CSE 003
  - Section AC M 2:30-5:20 CSE 003 --- still open
    - The hands-on laboratory meets 3 hours once a week. It provides students an opportunity to put what they learn in lecture to practice using digital logic prototyping kits and modern computer-aided design tools. Laboratory topics will be aligned to lecture and homework topics.

"Why" take CSE 370

- Required (okay, but let's talk about why it is required and will be useful for your future)
- Most basic building blocks of computer science (0's and 1's)
- It is important to understand how they are used as baseline for more complex operations (adding, storing, other logic like if/while)
- It is like studying anatomy of neurons and neuronal connections if you want to become a neuroscientist (even if you become a computational neuroscientist).
- It is good to understand what can be implemented in hardware, and why it is sometimes good to implement certain things on hardware instead of software
- Understand how some of the technology you interact with on daily basis (memory stick, vending machine, etc) at the hardware logic level.
- Knowledge gained in this course is used directly in industry/research

Class Goals

- Understanding of digital logic at the gate and switch level including both combinational and sequential logic elements.
- Understanding of the clocking methodologies necessary to manage the flow of information and preservation of circuit state.
- An appreciation for the specification methods used in designing digital logic and the basics of the compilation process that transforms these specifications into logic networks.
- Facility with a complete set of tools for digital logic design with programmable logic devices as the implementation technology and the realization of medium-sized state machine controller and data paths using PLDs and discrete logic.
- To begin to appreciate the difference between hardware and software implementations of a function and the advantages and disadvantages of each.
- Understand “how” these concepts are used in real world and “why” it is useful for us to know.

Syllabus 1

- Combinational logic basics
  - Binary/hex/decimal numbers
  - Ones and twos complement arithmetic
  - Truth tables
  - Boolean algebra
  - Basic logic gates
  - Schematic diagrams
  - Timing diagrams
  - de Morgan's theorem
  - AND/OR to NAND/NOR logic conversion
  - K-maps (up to 4 variables), logic minimization, don't cares
  - SOP, POS
  - Minterm and maxterm expansions (canonical, minimized)

I like Pink and Blue but not Yellow...

Syllabus 2

- Combinational logic applications
  - Combinational design
    - Input/output encoding
    - Truth table
    - K-map
    - Boolean equation
    - Schematics
  - Multiplexers/demultiplexers
  - PLAs/PALs
  - ROMs
  - Adders

4529
+ 34532

????

Syllabus 3

- Sequential logic building blocks
  - Latches (R-S and D)
  - Flip-flops (D and T)
  - Latch and flip-flop timing (setup/hold time, prop delay)
  - Timing diagrams
  - Asynchronous inputs and metastability
  - Registers

Remember that
the last number was 1

Syllabus 4

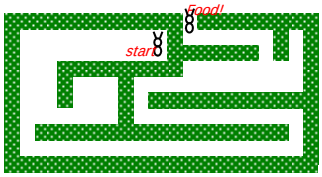
- Counters
  - Timing diagrams
  - Shift registers
  - Ring counters
  - State diagrams and state-transition tables
  - Counter design procedure
    1. Draw a state diagram
    2. Draw a state-transition table
    3. Encode the next-state functions
    4. Implement the design
  - Self-starting counters

1, 2, 3, 4, ...

Syllabus 5

- Finite state machines
  - Timing diagrams (synchronous FSMs)
  - Moore versus Mealy versus synchronized Mealy
  - FSM design procedure
    1. State diagram
    2. state-transition table
    3. State minimization
    4. State encoding
    5. Next-state logic minimization
    6. Implement the design
  - State minimization
  - One-hot / output-oriented encoding
  - FSM design guidelines
    - Separate datapath and control
  - Pipelining, retiming partitioning basics

The last coin was 25cents and already had 50cents deposited so let's pop out a soda



Tentative class schedule

Week	Monday		Wednesday		Friday	
0			9/24	Introduction	9/26	Binary number systems
1 Lab1	9/29	Boolean algebra	10/1	Boolean algebra *Homework1 due*	10/3	Logic gates/truth tables
2 Lab2	10/6	Canonical forms	10/8	Boolean Cubes *Homework2 due*	10/10	Verilog introduction
3 Lab3	10/13	Karnaugh maps	10/15	Multiplexors	10/17	PLDs *Homework3 due*
4 Lab4	10/20	Review	10/22	**MIDTERM1**	10/24	Multi-level logic
5 Lab5	10/27	Adders	10/29	Flip flops *Homework4 due*	10/31	State Diagrams Timing Diagrams
6 Lab6	11/3	Timing Diagrams	11/5	Finite State Machines *Homework5 due*	11/7	More FSM
7 Lab7	11/10	Moore/Mealy	11/12	Review *Homework6 due*	11/14	**MIDTERM2**
8 Lab8	11/17	FSM Robot	11/19	FSM robot Simplification	11/21	Simplification *Homework7 due*
9 Lab9	11/24	Encoding	11/26	NO CLASS Thanksgiving	11/28	NO CLASS Thanksgiving
10	12/1	Partitioning	12/3	Applications *Homework8 due*	12/5	Review
11	12/8	**FINAL EXAM (cumulative)** 12/8 8:30am Gowen 201				

Class Structure

- Lectures: Attendance and participation of all of them is strongly encouraged and expected. Lecture materials are closest to what is covered in the exams (over homework or lab). If you come to the lectures, you will likely do better on the exams.
- 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 TA will be available. We will work hard to ensure that the laboratory assignments take no more than the three hour sessions to complete. 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 8 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 (10/22 and 11/14) and one final exam during finals week (likely 12/8).

Class Policy: Grading

- Your course grade will be computed as follows:
  - 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)
- We will not curve this class --- so the student next to you doing well does not affect your grades (and we want all of you to do well). For example, last quarter, the average was ~3.6. We also provide extra credit assignments/labs.

Class Policy: Collaboration and Cheating

- ***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.

Class Guidelines

- ***Workload***
  - We will try to ensure that the workload is typical for a four-credit course, namely, nine to twelve hours per week outside of the lectures. If we do not succeed, please let us know explain which parts of the course are causing you to spend too much time non-productively.
  - *We have structured the course so that spending an hour or two per day will maximize your efficiency.* You will work this way in the real world—you cannot cram a three-month design assignment into the last night—so you may as well work this way now. Plus, you will understand the material better. If you leave the homework for the day before it is due, then you will not have time to study for the exams, and you will not have time to ask questions when (*not if*) the software misbehaves.
- ***Assignment***
  - The homework assignments are generally due on Wednesdays at the beginning of class (except when there is an exam or a holiday). The homework assignment will be distributed approximately one week before the due date.
  - Your assignments must be neat and legible. We will not spend time trying to decipher messy work. We urge you to use the graphical and word processing tools that are readily available to you in all the labs in the department. Please make good use of the schematic diagram editor in the tools you'll be using to make neat circuit diagrams to include in your assignments.
  - Assignment problems will sometimes be graded on a random basis. To get full credit for an assignment, you must, of course, turn-in solutions for each assigned problem. Only a subset of the problems will actually be graded in detail. You will not know in advance which problems this will be - so make sure to do all of them.
  - Please review the assignment solutions carefully before questioning a grade with either the instructor or the teaching assistants.

Class Guidelines (cont'd)

- ***Exams***
  - We have two midterms and one final exams. I heard smaller quizzes were not taken seriously in the past and one midterm was too stressful. We welcome feedback. There will be no make-up exams so schedule your term accordingly.
- ***Software***
  - Software tools frequently consume more time then they should. We have designed the assignments to get you up to speed gradually (over the period of a few weeks), but undoubtedly there will be some start-up cost (as with any new tool). Essentially, you are learning a new language, a compiler, and getting familiar with a process. Every tool imposes a certain model. Your frustration can be high until you assimilate that model and learn to use it effectively. Be sure to use the tutorials, and do not spend countless hours making no progress. Ask for help. Remember that these tools are written by engineers for engineers and will not necessarily conform to expectations you may have of consumer-oriented tools such as Word.