ECE 241F – Digital Systems

Fall 2016 – J. Anderson, P. Chow

Basic Information

Instructors and Lecture Information

Section	L101		L102		L103	
Instructor	Jason Anderson		Paul Chow		Jason Anderson	
Office	EA 314		EA 320		EA 314	
Phone	416-946-7285		416-978-2402		416-946-7285	
Email	janders@eecg.toronto.edu		pc@eecg.toronto.edu		janders@eecg.toronto.edu	
Office hours	Arrange with your instructor by email					
Lecture Rooms/Times	T 5-6	MB 128	T 5-6	MP 102	T 12-1	BA 1170
	R 4-5	MB 128	W 12-1	BA 1180	R 12-1	BA 1170
	F 2-3	BA 1190	R 1-2	BA 1190	F 12-1	BA 1170

Informal Tutorial

There will be seven informal tutorials held starting Thursday, Sept. 15. These tutorials will be held the week before each of the labs to give some guidance on the lab preparation for the following week. The tutorial will be held in BA3008 from 2 to 3pm. The tutorial will also be recorded and made available online. More details will be posted online when available.

Grading

Labs 10%	**Project** 10%	**Midterm** 30%	**Exam** 50%

Web Page:

Blackboard Portal – mainly for grades, secure stuff

www.piazza.com – discussion board, handouts – see Blackboard for more details

Midterm

Wednesday, October 12, 2016 in EX100 and EX300 from 6-8pm. This is Thanksgiving week, so there is no lab. Same conditions as exam.

Exam

Type D – examiner specified aids: **One single sheet of letter size paper (8.5x11 inch) with your written notes of your own choosing**. Both sides may be used. **No calculators, no cell phones.**

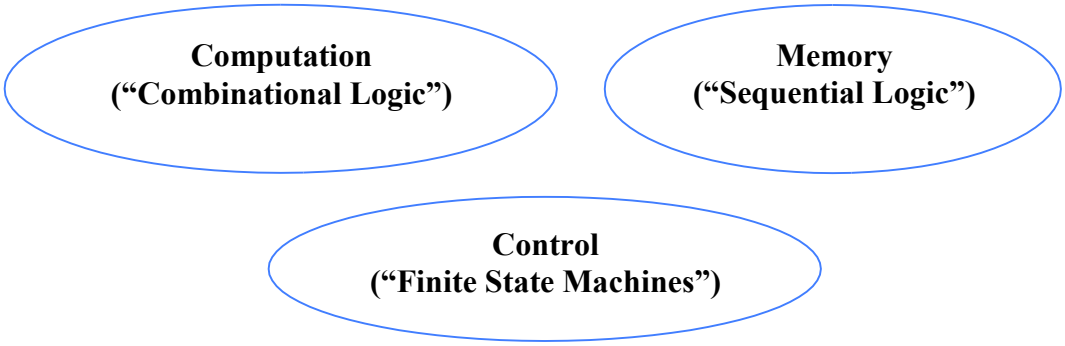
Required Text

Title: Fundamentals of Digital Logic with Verilog Design, **3rd Edition**
Authors: Stephen Brown and Zvonko Vranesic
Publisher: McGraw-Hill
ISBN: 0073380547

Running CAD Software at Home

See the page on CAD software to find out how to obtain the software for your computer and what kind of computer is required.

ECE 241F - Digital Systems - Course Outline



Goals

1. To understand basic digital logic circuit design, optimization and concepts.
2. To become comfortable using Computer-Aided Design (CAD) tools in design.
3. To gain hands-on experience with the design and debug of digital systems, using programmable logic.

1. Introduction to Combinational Logic

- switches and logic gates
- logic functions, truth tables and variables
- Boolean axioms and laws, sum of products, product of sums
- simple algebraic minimization - making things cheaper with a new kind of algebra

2. Technology

- logic voltage levels
- transistors as a switch
- NMOS and CMOS logic gates
- real propagation delay, and timing diagrams, timing analysis of digital circuits
- Field-Programmable Gate Arrays (FPGAs)
- introduction to Verilog (a language for describing hardware) and CAD tools that implement hardware given the Verilog description language

3. Combinational Logic Optimization

- minimization goals - speed and cost
- Karnaugh Map optimization technique
- optimization of logic that have "Don't Care" conditions
- critical path through combinational logic

4. Sequential Logic

- cross-coupled NOR/NAND gates basic latch
- gated latch
- Master-Slave D flip-flop
- shift registers
- counters
- set-up & hold time, clock-to-Q

5. Finite State Machines

- how logic is controlled
- state diagrams
- Moore-type state machines, Mealy-type machines
- state machine synthesis
- state machines in Verilog
- state encoding and optimization

6. Numbers and Arithmetic

- number representation, binary, ones & twos complement representation of negative numbers
- basic adder/subtractor
- carry look-ahead methods for fast addition
- bit serial addition

7. Miscellaneous

- multiplexors & tristate gates
- multiplexors as logic; decoders
- fanout-dependent delay
- power dissipation, I/O devices and FPGAs
- Static Random Access Memory (SRAM);
- controller for digital display
- de-bouncing mechanical switches
- VGA display interface

ECE 241F - Digital Systems - Lab Schedule and Information

Fall 2016 J. Anderson, P. Chow

The real learning in this course goes on in the laboratory where you design, build, test and fix real circuits. There are **seven mandatory** labs and you will have one 3-hour lab every week. You will work in groups of **two**.

There is also a project based on the material of this course that you will also work with your partner on. This will take three lab periods done after the mandatory labs.

There are two parts to the lab experience: preparation, which you must do outside of the lab hours, and the actual implementation of circuits in the lab.

Preparation

Each lab usually requires you to do a significant amount of preparation, and is where you must do much of the work to understand the concepts. Preparation must be complete before the lab begins. Preparation will usually require design using the CAD software supplied. **Each partner in the group of 2 must perform and submit a separate preparation.** While it is acceptable to discuss your preparation with your partner, your work may not be copied from your partner; you will be required to explain your preparation. Please be aware that severe penalties will be imposed for copying of labs, as evidenced by an inability to explain the work given as preparation. It will be graded by the TAs at the beginning of the lab, on the following basis:

Judgement of TA	Grade
Did nothing, did not try or cannot explain anything	0
Below expectations - did not attempt at least one part of the prep or answers to questions show a lack of understanding of what is shown in the preparation	BE
Meets expectations - all parts attempted, looks like it is close to working in simulation, reasonable effort spent, answers to questions show knowledge about the design	ME
Above expectations - everything done with working simulations, preparation is neat, clear and well organized, code looks clear and understandable and has comments	AE

In-Lab Work

In each lab you will typically have to build a working circuit. Once this is done, for each such circuit, show it to your TA for grading:

Judgement of TA	Grade
Did nothing, did not try	0
Nothing works, but tried	NW
Something works - Got at least one part working	SW
All works - Everything works as required	AW

Note: Although the lab portion of the course is worth only 10%, both the midterm and the final exam will contain questions directly related to skills learned in the lab.

Asking Questions in the Lab

The goal here is to help the TA assess your problem quickly and for you not to ask questions unless you have enough information to help the TA quickly assess what you are doing or need. It is important to be able to pose precise questions. "Help me, it does not work" is not precise! "I have tried these things, here's where it fails in simulation and here's the code. I think it should do this, but look at what it is doing. Can you help me understand why this does not work?" is a well-constructed question.

If you are having tool issues, the TA will try to help right away.

If your design is not working, the TA will want to see two things before attempting to help. This is to help the TA quickly understand your design.

1) You must have a schematic of your design. This is the quickest way for a TA to understand how your design is supposed to work. It can be hand drawn. Until you gain more experience with Verilog, you should really start with a schematic, but if you did not do this, you should be able to create one quickly. The schematic should be at least at the detail of how all the modules are connected as well as showing how the logic and registers are connected. An example schematic with corresponding code can be found at http://www.eecg.toronto.edu/~pc/courses/241/schematic.

2) Simulation is the key to solving most bugs when your code is not working. It is not sufficient to say there is a bug and ask for it to be fixed. If you see a bug in simulation, it is actually a good thing! Wait till you have a bug and the simulation is perfect! You must have a simulation that demonstrates the bug. Show the last working point in the simulation and what is causing the bug. Be prepared to relate that point to the logic in your code where the bug is arising.

Lab Workstation Number and Maintenance

Each digital workstation that you'll be sitting at has a number. Please use the same station each week. If a piece of equipment is not working, please tell a TA to tag the board with the problem and notify someone to have it repaired. Otherwise it will be broken the next time you need to use it!

Lab Sections, Day Time and Location

Section	Day	Time	Location(s)
1,2	Monday	3pm – 6pm	BA 3135, 3145, 3155
3,4	Monday	noon – 3pm	BA 3135, 3145, 3155
5,6	Monday	9am – noon	BA 3135, 3145, 3155

Lab and Project Schedule for 2016

See the ECE241 Syllabus document for the schedule.

ECE 241F - Digital Systems - Access to CAD Software

The lab in this course depends heavily on the use of Computer-Aided Design (CAD) software to implement the circuits, usually on a programmable logic device. We will be using Field-Programmable Gate Arrays (FPGAs) from Altera, and Altera's **Quartus** Version 16.0 CAD software. You will be able to access this software in one of three ways:

1. On your own home computer.

If you have a Windows 7 or 8.1, or a Linux-based home computer with 6-8 GB of physical RAM and about 40 GB of disk space, you can download the Quartus *Prime Lite Edition* Version 16.0 directly from the Altera web site. If you choose to do the download via the individual files make sure your installation also includes the ModelSim-Altera Starter Edition software and Cyclone V device support. In the lab we will be using the full version of the software, but the Lite edition has all you need.

https://www.altera.com/downloads/download-center.html

Windows 10 is not officially supported for the Modelsim simulator, but we have unofficial word from Altera that no one has reported issues running 16.0 under Windows 10. While the recommendation for the Cyclone V device we are using is to have 6-8GB of physical memory, our designs will be small, so you should be able to get away with less. Worst case is that it runs slower… You can find specific details about recommended OS and system configurations at the download site.

If you do not have this class of computer, you will be able to use the University's machines instead.

2. On the Windows Machines in the lab – Bahen 3135, 3145, 3155

There are a total of one hundred Windows 7-based machines in the Bahen Centre labs with 16GB of memory. These will have the latest release of the full license (not the Lite edition) of Quartus 16.0 software installed on them. These machines will have access to your home directory on the ECE network. You will only have access to these labs during your scheduled lab periods.

3. Other Windows Machines

The Quartus Prime Lite Edition Version 16.0 is also available in all ECF Windows labs and the ECE Windows lab in BA3128. The ECE Design Centre will have stations with the same setup as the BA labs. These facilities are available outside lab hours.

ECE 241 Project

The Task

The purpose of the project is to:

1. Gain experience dealing with the design of a larger digital system, and to deal with the issues in going from a soft simple specification ("make a digital toaster controller") to an actual complete design. There is no substitute for experience!

2. Express your creativity by applying what you have learned in this course to a project of your own choosing.

You will design and implement a project of your own choosing that uses digital logic in some creative way. You may use any of the parts available in the lab, but are restricted to using just one of the Altera DE1-SoC boards. An important part of this lab is the creativity required to think up an interesting project, and then negotiate with a TA or instructor as to the final form of the project.

Originality/Uniqueness Approval

The first step in your project is to come up with an original idea. You must submit your idea in a 1-3 line description online for "originality" or "uniqueness" approval. More details on the process will be provided later. Please note that this approval is only the first step and only deals with the basic idea, and not the scope/effort required for the project; that comes next:

Before the First Project lab

You will submit a short project proposal of what your project is about. This should be a short description that gives:

- The basic idea of the project, and the basic function of your circuit.
- Describe the inputs and outputs, and give simple block diagrams describing how the various parts of your circuit interact.
- Your plan of action for each of the three lab periods - "milestones"
- Present this to your TA to get their opinion on whether the project is viable. This is just a check to make sure that you do not try something overly ambitious.

Demonstration and Report

You will demonstrate your project to your supervising TA in the final lab period, and will be required to provide a short report describing your project.

Alternate Reversi Competition

We hope to have an optional alternative to the project by running a Reversi competition, but building the game in **hardware.** You should all be familiar with the rules from APS105 where you did this in software. Stay tuned for more details about this possibility.

TENTATIVE Lecture and Lab Schedule for ECE 241F, 2016

Week	Topics covered	Textbook Sections	Lab Exercises
Sept 5 1	Overview of the course: topics covered, mark breakdown (midterm test, labs, final exam), overview of how the lab exercises are organized (students work in groups of two), marks assigned for preparation and lab performance); Quick overview of digital systems and Moore's law, examples of digital systems; Transistors as simple on-off switches	Chapter 1	
Sept 12 2	Project intro, videos Binary numbers, hex numbers Introduction to logic expressions; AND, OR, NOT circuits built using switches; AND, OR, NOT gate symbols; truth tables; simple example of logic circuit with AND, OR, NOT gates Boolean algebra: duality, axioms, rules, identities; proof of identities using perfect induction (i.e., truth tables); algebraic manipulation of Boolean expressions; timing diagrams; Venn Diagrams and their use to prove some identities Simple synthesis of logic circuits; sum-of-products (SOP) form; minterms; canonical SOP; product-of-sums form (POS); maxterms; canonical POS; examples of algebraic manipulation Intro to Lab 1	 3.1 2.1 – 2.4 2.5 2.6 B.5	
Sept 19 5	Example logic functions: 2-to-1 multiplexer, XOR gate, full-adder, ripple-carry adder, 7-seg; NAND and NOR logic networks; convert SOP to NAND-NAND, POS to NOR-NOR. Verilog introduction, including hierarchy	2.7, 2.8, 3.2 2.10, Appendix A	Lab 1: Building Circuits Using 7400-Series Chips Lab tutorial: Read on your own time the tutorial: Quartus Introduction (you must at least do the version that uses Verilog, and can consider also doing the Schematic design version). Download tutorials from https://www.altera.com/support/training/university/materials-tutorials.html; perform tutorial steps outside of the lab using simulation only
Sept 26 8	Introduction to Field Programmable Gate Arrays (FPGAs), lookup tables; Introduction to CAD tools Introduction to cost of a logic circuit; terminology: implicant, prime implicant (PI), essential PI, cover, minimum-cost cover; introduction to K-maps (2, 3, 4 variables) More examples of K-maps use, including as a guide to algebraic manipulation	B.6.5, 2.9 2.11 – 2.14	 Lab 2: Multiplexers, Hierarchy, and HEX Displays
Oct 3 11	5-variable K-maps; don't cares, examples, including 7-seg with don't cares Storage elements: introduction, RS latches, timing diagrams, gated RS latch. Gated D latch, D flip-flops, Flip-flop reset/preset, setup and hold times	2.8.3 5.1 - 5.4, 5.7	 Lab 3: The case statement, Adders, and ALUs
Oct 10 14	Verilog latches in if-else, case statements Registers, shift registers, Verilog for registers, blocking vs non-blocking FFs in FPGAs	A.11.1 – A.11.4 5.8, 5.12, 5.13, A.11.7	**Midterm on Wednesday, Oct. 12, 2016 in EX 100 and EX 300 from 6-8pm.**
Oct 17 17	T FF Counters; ripple and synchronous counters; Verilog for counters, enable inputs Timing, skew	5.5 5.9, 5.10, 5.13 5.15	Lab 4: Latches, Flip-flops, and Registers
Oct 24 20	Finite state machines intro FSM state assignment, binary encoding, one-hot, Verilog code for FSMs	6.1 6.2, 6.4, 6.5	Lab 5: Clocks and Counters
Oct 31 23	FSM timing issues (Moore vs Mealy models) RAM and ROM, including FPGA embedded memory Discussion of course project, incl VGA and videos	6.3 B.9	Lab 6: Finite State Machines
Nov 7 26	Signed numbers; 2's complement; adders/subtractors, arithmetic overflow Carry lookahead adders, multipliers	3.3 3.4, 3.6	Lab 7: Memory and VGA Display
Nov 14 29	Arithmetic coding with Verilog Combinational circuits: implementing logic functions using only multiplexers, Shannon's Expansion Logic Synthesis using LUTs	3.5 4.1 B.6.5	 Project 1
Nov 21 32	Decoders, other combinational circuits; Verilog code FSM State Minimization Design example: Introduction to processors	4.2 – 4.6 6.6 7.1 – 7.2	Project 2
Nov 28 35	More simple processor Transistors and gates, structure Passing 1's and 0's, transmission gates, fan-in	 B.1 – B.3, B.8.1 B.8.7 – B.8.9	Project 3
Dec 5 38	Clock skew, clock synchronization, switch debouncing	7.8	

Informal tutorials will be held every Thursday, 2 to 3pm in BA 3008 starting September 15, 2016 every week before Labs 1 to 7.