
# Course Syllabus

## ENEE 245: Digital Circuits & Systems Laboratory, Spring 2015
## Prof. Bruce Jacob

## Basic Information

*Time & Place*

Lecture: M 10:00–10:50 am, PLS-1130 — All Labs: AVW-1450

Section 0201 Lab: M 11:00am – 1:50pm — *TA: Carson Dunbar, cdunbar@bucknell.edu*
Section 0202 Lab: Tu 2:00pm – 4:50pm — *TA: Ahmed Elshaarany, ahmed.m.elshaarany@ieee.org*
Section 0203 Lab: Th 8:00am – 10:50am — *TA: Carson Dunbar, cdunbar@bucknell.edu*
Section 0204 Lab: Th 2:00pm – 4:50pm — *TA: Honglei Li, lihonglei001@gmail.com*

*Professor*

Bruce L. Jacob: AVW-1333, blj@umd.edu
Office hours: open-door policy

*Class Home Page*

http://www.ece.umd.edu/courses/enee245

*Class Email List*

enee245-02all-spring15@coursemail.umd.edu

## Class Schedule

This is a weekly schedule of my hours, including class time and scheduled office hours, but also including other things that make me unavailable. It is subject to change.

| | MON | TUE | WED | THU | FRI |
|---|---|---|---|---|---|
| 9–9:30 | | | | | |
| 9:30–10 | | | | | |
| 10–10:30 | **ENEE 245** Lecture PLS-1130 | | | | |
| 10:30–11 | | | | | |
| 11–1:30 | | | | | |
| 11:30–12 | | | | | |
| 12–12:30 | | | | | |
| 12:30–1 | | | | | |
| 1–1:30 | | | | | |
| 1:30–2 | | | | | Weekly meetings with graduate students |
| 2–2:30 | | | | | |
| 2:30–3 | | | | | |
| 3–3:30 | | | | | |
| 3:30–4 | | | | | |
| 4–4:30 | | | | | |
| 4:30–5 | | | | | |

## Course Overview

This course covers the design of digital circuits, including some analog support for digital circuits that is found in pretty much all digital chips these days. You will learn digital design by both observing circuits in the lab and building them in the lab. You will build combinational logic and sequential logic on breadboards, design advanced analog and digital components for modern VLSI chips, synthesize your designs and program them onto FPGAs, and observe the behavior of circuits, gates, and transistors in the lab. The term "building" in this course will mean implementing your design using discrete chips on breadboards as well as in software using the Verilog *hardware description language (HDL)* and synthesis onto FPGAs. You will test your designs using modern lab equipment such as oscilloscopes, function generators, and logic analyzers.

The topics will begin with straightforward examples from ENEE 244 and build from there, including combinational adders & multipliers, multiplexers of various types, iterative multipliers and other state machines, memory arrays and peripheral logic support (e.g., decoders and sense amplifiers), pass transistors and their behavior, voltage boosting mechanisms, sensors and actuators, reliability and redundancy, and the control of memory devices. The culmination labs will have you designing an out-of-order memory controller and implementing it in an FPGA.

Building in Verilog (or in any HDL, for that matter) and implementing in FPGAs is interesting for several reasons. First, you must be extremely precise in your design, to get it to work correctly. This forces you to understand all the finer points of your design and the ramifications of your choices—if you are not thorough, it will not work. Second, the performance/power/die-area results that you obtain are infinitely more believable than had you designed something "on paper" or built the model in a high-level language such as C. It is good to learn this because, historically, lots of money has been spent on ideas that looked good on paper but whose implementations fell far short of projected measurements. Lastly, an HDL implementation is just steps away from actual silicon, and FPGA implementations are often excellent replacements for custom ASICs, as they are far less expensive (thousands of dollars instead of millions of dollars) and can perform well enough to substitute.

## Prerequisites

Students must have taken ENEE 244, or have equivalent knowledge of digital logic design. You should understand digital logic concepts such as logic gates, boolean algebra, finite-state machines, and flip-flops. Students also must have taken ENEE 150 or CMSC 132. You should also understand and be reasonably fluent in programming in C, e.g. using arrays, structures, functions, and pointers. Knowing C (or Perl) will be extremely helpful, because Verilog is very C-like (as is Perl).

## Course Material

For learning Verilog, I have posted two handouts on the course website, and I recommend the following textbook very, very highly:

> *Verilog Styles for Synthesis of Digital Systems*, by Smith & Franzon.

Everything else you need will be found in the lab write-ups.

## Laboratory Projects

Twelve labs are assigned, each of which will require a substantial time commitment on your part. You will find the work load in this course to be extremely heavy.

The most common reason for not doing well on labs is not starting them ahead of time. Therefore, each lab has a pre-lab component that requires you to read and think about the lab before starting

the actual lab work. Pre-labs are due at the start of the laboratory; *you will not be allowed into the lab classroom unless you have completed the pre-lab*. These are *individual* assignments (you must do each by yourself), and they must be typed and printed; hand-written work is absolutely not allowed for any reason.

Lab reports describing the results of a lab session are due no later than one week after the lab work has been done—i.e., no later than the start of the next lab. You will work in lab with a partner, and so your lab report is a two-person effort. Lab reports can be turned in to the TA electronically or in person. *Late reports will not be accepted*. As with pre-labs, they must be typed and printed; hand-written work is absolutely not allowed for any reason. ***Note: bring a USB thumb drive to every lab, so you can take your data home with you.***

There are many sources of help on which you can draw. Simple questions can be submitted to the professor and fellow classmates via email (**use the email list given on page 1**). These will typically be answered within the day, often more quickly during working hours. Students are also encouraged to help one another. *One of the best ways for you to make sure that you understand a concept is to explain it to someone else.* Keep in mind, however, that you should not expect anyone else to do any part of your lab for you. The lab report that you turn in must be your own.

## Lecture & Lab Schedule

| Week of | Lecture & Lab Topics |
|---|---|
| **Jan 26** | Lecture/Lab 0: Course Overview, Tutorials on lab equipment & software |
| **Feb 2** | Lecture/Lab 1: Combinational logic (e.g., adders) |
| **Feb 9** | Lecture/Lab 2: Sequential Logic (latches, registers, clocks) |
| **Feb 16** | Lecture/Lab 3: Simple state machine (saturating counter) |
| **Feb 23** | Lecture/Lab 4: Intro to MOSFETs and voltage boosting (charge pumps, etc.) |
| **Mar 2** | Lecture/Lab 5: Intro to Verilog (fundamentals) |
| **Mar 9** | Lecture/Lab 6: Verilog & FPGAs (advantages) |
| **Mar 16** | ***Spring Break*** |
| **Mar 23** | Lecture/Lab 7: Multiplexers (two types & Verilog realization) |
| **Mar 30** | Lecture/Lab 8: Basics on I/O (7-segment display) |
| **Apr 6** | Lecture/Lab 9: More FSMs (vending machine) |
| **Apr 13** | Lecture/Lab 10: More I/O (LCD controller & debugging) |
| **Apr 20** | Lecture/Lab 11: Memory basics (simple controller timing) |
| **Apr 27** | Lecture/Lab 12: Memory and DDR I/O (flash controller timing) |
| **May 4** | Lecture: advanced topics — **no lab (lab make-up week)** |
| **May 11** | **Final Exam** in class — **no lab (lab make-up week)** |

## Exam

You are expected to take the final exam at the scheduled time. Unless a (documented) medical or personal emergency is involved in your missing an exam, you will receive a zero for missing the exam. If you anticipate conflicts with the exam time, you must come talk to the instructor about it at least **1 month** before the exam date. The exam date is given to you at the beginning of the term so

that you can avoid scheduling job interviews or other commitments on exam day. Outside commitments are not considered a valid reason for missing an exam. The exam will be closed book, closed notes.

## Grading Policy

Final grades will be based on the total of points earned on the labs and exam. The point breakdown:

- Labs: 80%
- Final Exam: 20%

Incompletes will generally not be given. According to university policy, doing poorly in a course is not a valid reason for an incomplete. If you are having problems in the course, your best bet is to come talk to the instructor as soon as you are aware of it.

## Special Needs

If you have a documented disability that requires special needs, please see me as soon as possible, and certainly no later than the third week of classes.