EEE 120 Summer C 2014
Final Exam Study Guide

Your preparation for the final exam should include a review of all the lecture notes throughout the semester. You should also have a look the quizzes 1-7. In addition, some questions will be related to the labs (simulation and hardware), so be aware of that as well. The exam will consist of a multiple choice section and a small finite state machine design problem. You may use one 8.5"x11" cheat sheet, a pencil, and a calculator. In addition, you should have scratch paper available for the design problem. If the answer to one of the questions happens to be on your cheat sheet, you must still show your work on the exam. Although there will be the possibility to earn extra credit on the final exam, there are no specific extra credit problems on the exam. The total number of points you can get is 220, while 200 points will count as a 100% score.

Here is a comprehensive list of things you should understand. I highlighted the sections that have not been part of any previous exam.

- Basic logic functions (AND, OR, NOT, NAND, NOR, XOR)
    - Know the truth tables
    - Know how to draw the gates
    - Understand how to use DeMorgan's laws to convert the logic gates to their complementary forms (e.g., a NAND gate can be drawn as an AND gate with an inverted output, or as an OR gate with inverted inputs)

- Boolean functions
    - Represent a Boolean function in a truth table
    - Represent a Boolean function in short hand form, including sum-of minterms and product-of-maxterms forms
    - Represent a Boolean function in algebraic form, including SOP and POS forms
    - Be able to minimize a Boolean function via Boolean algebra and also using K-maps
    - Implement an arbitrary Boolean function using AND/OR/NOT gates, using only NAND gates, and using only NOR gates

- Number systems
    - Represent a number in different bases (base-2, base-8, base-10, base-16)
    - Be able to convert between different number systems
    - Addition/subtraction
        - Understand the 2's complement representation, and the 2's complement operation, and the difference between them
        - Be able to identify the overflow conditions

- Higher order logic blocks
    - Understand the operation of adders/subtractors
    - Understand the operation of multiplexers and decoders

        - Be able to implement a Boolean function using a MUX and using a decoder
    - Understand the operation and purpose of tri-state buffers
    - Understand the operation and purpose of open collector output gates
    - Know how to implement a Boolean function in a ROM, PLA, or PAL
    - Understand how these higher order logic blocks can be used to build more complicated circuits, such as an ALU or the brainless microprocessor

- Memory blocks
    - Understand the operation and timing of SR latches and D latches
    - Understand the operation and timing of D, T, and JK flip flops
        - Truth tables
        - Excitation tables
        - Asynchronous set/reset
    - Know how to interpret timing diagrams

- Memory devices
    - Counters: understand the function of a synchronous counter and how to design one
    - Registers: understand the parallel-in/parallel-out (the register we used in the microprocessor), and serial-in/parallel-out (shift register) designs.

- Synchronous machines
    - Understand the difference between Mealy and Moore finite-state machines
        - advantages and disadvantages of each design
    - Know the design procedure for finite-state machines

- The complete microprocessor
    - Understand the operation of the microprocessor from sim lab 4: the fetch-execute cycle, how instructions and data are stored in memory, how different pieces of the microprocessor share the data bus, how memory addressing works, etc.