EE 201 Midterm study guide

1. Numbers

- Convert an unsigned integer from decimal to binary and back
- Add numbers in binary
- Convert an integer from decimal to hexadecimal and back
- Convert a signed integer from decimal to 2's complement binary and back
- Explain why we prefer 2's complement to sign-magnitude

2. Boolean equations, truth tables, and circuits

- Write a 2-input truth table for each of the following logic operations: AND, OR, NOT, NAND, NOR, XOR, XNOR
- Write truth tables for AND, OR, NAND, and NOR gates with 3 or more inputs.
- Write a boolean equation from an English description of a logic system, and vice versa, using the rules of operator precedence.
- Draw a logic gate diagram from a boolean equation, and write a boolean equation from a logic gate diagram
- Write a truth table from a logic equation be evaluating it for each value of the inputs
- Write a boolean equation from a truth table using either canonical form (sum of products and product of sums)

3. Manipulating boolean equations

- Use boolean algebra to manipulate boolean equations, using the 12 theorems listed in the book (Commutativity, associativity, distributivity, De Morgan)
- Use "bubble pushing" to manipulate the logic of a circuit
- Use the above logic manipulation techniques to minimize or implement a circuit with constraints (e.g., only NAND gates)
- Draw a Karnaugh map for a 3 or 4-input truth table, and use it to create a minimal boolean equation for the truth table.
- Find a minimal boolean equation for a truth table with "don't care" inputs and/or outputs

4. Multiplexers and FPGAs

- Explain what a multiplexer is, and write out a truth table for a N:1 multiplexer (where N is 2, 4, 8, 16)
- Draw a logic diagram using a 2^M-input multiplexer to implement a boolean equation with M variables
- Describe the basic structure of an FPGA (i.e., explain to someone who just started the class what an FPGA is)

6. Timing combinational logic

- Given a circuit and timing information about the gates, calculate the propagation delay and contamination delay
- Optimize a simple combinational circuit for speed by implementing the same function with fewer or faster gates.
- Explain what glitches are and why they occur

7. SystemVerilog for combinational logic

- Using a reference sheet, write the code for a complete SystemVerilog module
- Use SystemVerilog to implement a boolean equation or truth table

8. Testing combinational logic

- Write a test bench for combinational logic
- Explain the difference between synthesizeable and non-synthesizeable constructs

9. Latches and flip-flops

- Compare and contrast an SR latch, a D latch, and a D flip-flop.
- Given a circuit containing a D latch or a D flip-flop, draw a timing diagram to illustrate how and when the output changes.
- Sketch the general structure of a sequential circuit
- Explain why we often prefer flip-flops over latches.

10. SystemVerilog for sequential logic

- Create flip-flops using SystemVerilog `always` blocks
- Explain the difference between blocking and non-blocking assignments, and when to use each
- Build counters and other simple sequential circuits using SystemVerilog

11. Finite state machines

- Explain what a state machine is in terms someone just beginning this course could understand
- Explain the difference between a Mealy and Moore state machine
- Given an English description of a system, do the following for both Moore and Mealy implementations:
  - Draw the state diagram
  - Determine how many bits of state are necessary, and choose state encodings
  - Write truth tables and logic equations for the state transitions and outputs
  - Draw the complete logic circuit diagram for the FSM
- Implement a finite state machine using behavioral SystemVerilog constructs

12. Timing sequential logic

- Define setup time and hold time, and annotate them on a timing diagram
- Given a sequential circuit and a table of gate/flip-flop delays, draw a timing diagram
- Calculate the maximum speed a sequential circuit coudl run at
- Optimize sequential circuits for speed