CPE 133: Final Exams Study Guide v1.05

I'll provide a cheatsheets for the exam and before the exam. I won't ask you to write Verilog code for either final, but I will ask you to interpret Verilog code and Verilog modeling on the lab final.

Lab Final Overview:

- Closed book and notes; I'll provide a cheatsheet(s), which will be available in advance of exam
- Primarily short answer questions, including a few timing diagrams
- I'll pull many of these questions from the end of the experiments

Lecture Final Overview:

- Closed books and notes; I'll provide cheatsheets (which will be available in advance of exam).
- Primarily solved problems – *Expect:*
  - a problem that asks you to show the four standard circuit forms from a written description (standard POS & SOP, NOR/NOR & NAND/NAND forms)
  - a mixed logic problem analysis problem and/or a mixed logic design problem,
  - a FSM-based sequence detector problem,
  - a FSM-based shift register design problem.
  - non-FSM-based circuit design problem(s)
  - Several FSM-based circuit design problems
  - Low-level counter-based FSM design (3 underlying modules)
  - Maximum FSM clock frequency problem

NOTES:

1) **Make sure you include a high-level BBD with every design-oriented problem**

2) **Make your circuit diagrams are 100% complete. This means they should include:**
   a. All diagrams should be easily readable (should not be small or messy)
   b. All important signals should be clearly labeled
   c. All labeled signals should match the high-level black box diagram
   d. All bundles (buses) should be clearly marked with bus width indicators
   e. All MUXes should include select numbering on inputs
   f. All modules used in circuit should be clearly labeled
   g. All decoders used in BBDs must include tabular descriptions
   h. Any non-foundation modules must be clearly defined
   i. All registers (including shift registers and counters) must clearly define ambiguous control input functionality and precedence

3) **Make your state diagrams 100% complete. This means they should include:**
   a. A legend describing the information state diagram (I/O, states, transitions)
   b. Indicators describing all state transitions (inputs and Mealy outputs)
   c. Indicators describing Moore outputs

4) **Known common errors:**
   a. Including "magic boxes" without proper definitions, particularly generic decoders
   b. Forgetting to include black box diagrams (both high and low levels)
   c. Forgetting to include labels for boxes in black box diagrams
   d. Forgetting signal labels for circuit diagrams (which should match black box diagrams)
   e. Forgetting to account for asynchronous inputs in FSM design problems
   f. Forgetting to completely describe control & ambiguous inputs on registers
   g. Forgetting to include proper units on problems with numerical answers
   h. Forgetting conditions on state diagram transitions
   i. Having state diagram conditions that are not mutually exclusive
   j. Mixing inputs & outputs in state diagrams
   k. Forgetting radix indicators on non-decimal numbers

- Know how to work with all the foundation modules on the provided cheatsheet (not including RAM-type modules)
- Know how to analyze and design basic circuits and supporting FSM (state diagrams) from written problem descriptions using your knowledge of foundation module digital devices such as MUXes, decoders, comparators, and ripple carry adders, etc.
- Know all general aspects of and how to work with unsigned and signed binary numbers (RC format)
- Know how to detect a valid RCA result using RC numbers (for addition and subtraction)
- Know how to generate four circuit forms (standard POS & SOP, NAND/NAND & NOR/NOR forms) from a written problem description
- Know how to size signal widths based on problem descriptions
- Know how to derive the various forms of gates
- Know how to configure basic gates as inverters
- Know how to design and/or analyze mixed logic functions
- Know the meaning of parity generation and parity checking and be able to work with these type of black boxes in a given circuit diagram.
- Know how to analyze basic digital models and their relation to the digital circuits they describe, particularly in the context of timing diagrams
- Understand how Verilog models standard digital circuits, including FSM models
- Know how to find the maximum operating frequency of a FSM-type circuit
- Know the difference between FSM types and types of digital circuits
- Know the relation between state diagrams and timing diagrams particularly as they apply to FSMs, including both Mealy and Moore outputs
- Know how to design state diagrams to solve sequence detector problems (both Mealy/Moore and reset/non-reset) and counters
- Know how to use state diagrams to design shift registers
- Know the underlying hardware implementations of digital foundation modules
- Know what term "duty cycle" means and how it applies to timing waveforms
- Know the general operation of shift registers (including mathematical abilities) and be able to work and/or design with simple shift registers and their associated timing diagrams
- Know the basic operations of counters, particularly their typical control inputs and status output
- Know what setup & hold times are and how they can affect your synchronous circuit
- Know the basic submodules of an accumulator and how to use an accumulator in a circuit
- Know how to use a counter as an event counter
- Know how to hold a pencil or pen using only your toes; be able to do this with your eyes closed