370 Final Exam Study Guide

The final exam will be the size of two to three quizzes. One quiz will cover the material that would have been on Quiz #5, i.e., material from the last two weeks of the course. The rest of the exam will be review material from the entire course.

Review topics:

Number systems: binary (incl. 1's, 2's and signed-magnitude negatives), hex, decimal.

Logic gates: truth tables and schematic symbols

Boolean algebra: know all of the basic identities. You will not have to do complicated reductions or proofs.

K-maps: 3 and 4 variable maps. Simplification from K-maps (SOP and POS).

Combinational design: English description -> input/output encoding -> truth table -> K-map -> Boolean equation -> schematic.

Minterm and maxterm expansions (canonical, minimized)

de Morgan's theorem

AND/OR to NAND/NOR logic conversion

K-maps, logic minimization, don't cares

Arithmetic circuits: half-, full-adders and schemes for larger designs

Switching circuits: multiplexers, decoders, PLA, PALs, ROMs. Properties and notation. Use to implement combinational circuits.

Latches and flipflops: definitions, behavior, timing, use.

Clocks and timing concepts.

Verilog basics: modules, ports; continuous vs. procedural statements; language elements; blocking vs. non-blocking

Sequential logic building blocks
- Latches (R-S and D)
- Flip-flops (master/slave D, edge-triggered D & T)
- Latch and flip-flop timing (setup/hold time, prop delay)
- Timing diagrams
- Flip-flop clocking
- Asynchronous inputs and metastability
- Registers

Counters

- Timing diagrams
- Shift registers
- Ripple counters
- State diagrams and state-transition tables

370 Final Exam Study Guide, p. 2

Counter design procedure

1. Draw a state diagram
2. Draw a state-transition table
3. Encode the next-state functions
4. Implement the design

Self-starting counters

Finite state machines

Timing diagrams (synchronous FSMs)

Moore versus Mealy versus registered Mealy; verilog versions

FSM design procedure

1. Understand the problem (state diagram & state-transition table)
2. Determine the machine's states (minimize the state diagram)
3. Encode the machine's states (state assignment)
4. Design the next-state logic (minimize the combinational logic)
5. Implement the FSM

FSM design guidelines

Separate datapath and control

FSM optimization techniques

Computer organization

Instruction sequencing- fetch, decode, execute

Harvard vs. Princeton architecture

Processor datapath building blocks

Suggested problems for reviewing Chapter 8

8.3 (emphasizes difference between Moore and Mealy)
8.5 (state reductions and building state diagrams from FSMs)
8.7 (state assignment)
8.12 (state partitioning)
8.14 (state partitioning)
8.24 (design process)

Suggested problems for reviewing Chapter 9

9.1 (state reduction)
9.4 (counter based FSM)
9.5 (PAL design)
9.19 (Implementing functions using look-up tables)

Suggested problems for reviewing Chapter 10
Though only three problems, they are long design-type problems (kind of like take-home quiz 4).

10.1 (Design Problem)
10.4 (Design Problem)
10.15 (Design Problem)

Appendix A
A7f, A8e,f, A12c, A14c, A14e,f (on boolean arithmetic)