Fall 2011 Midterm 2 Exam Review

mr-1

This Summary

This summary may be updated by Monday night.

When / Where

Wednesday, 9 November 2011, 14:40-15:30 CST

213 Tureaud (Here)

Conditions

Closed Book, Closed Notes, No Electronics

No use of communication devices, even for accessing Facebook `:-)`.

Format

Lots of short-answer questions.

Resources

Solved test and homework:. `http://www.ece.lsu.edu/koppel/ee2720/prev.html`

Brown and Vranesic (the textbook).

Dr. Skavantos' handouts.

Study Recommendations

Study this semester's homework assignments. Similar problems may appear on the exam.

Solve Old Problems—memorizing solutions **is not the same** as solving.

Following and understanding solutions **is not the same as** solving.

Use the solutions for brief hints and to check your own solutions.

Topics

mr-3

Summary of Topics

Min- and maxterm canonical forms, etc. (Book Section 2.6)

NAND and NOR networks. (Book Section 2.7).

The exclusive-or gate.

Karnaugh maps and all that that implicates. (Book Sections 4.1-4.5)

Multiplexors, decoders, encoders, priority encoders. (Book Sections 6.1-6.3)

mr-4

Skills

Convert between any of the following representations of a Boolean function:

A Boolean algebraic expression. $a'(b+c)+e$.

A truth table.

A Karnaugh map.

An expression using this minterm shorthand: $\sum_{x,y,z} m(1,3,7)$.

An expression using this maxterm shorthand: $\prod_{a,b,c} M(0,6)$.

A logic diagram.

Determine a Boolean function from a problem description.

mr-5

Skills

Know why NAND and NOR gates are good for SoP and PoS, respectively.

Know how to use just NAND or NOR gates for any function.

mr-6

Skills

Simplify Boolean expressions using a Karnaugh map.

Know how to draw a Karnaugh map.

Understand the different kind of implicants.

Know how to find prime implicants on a Karnaugh map.

Know how to find minimum-cost expressions from a map.

Know how to use don't-cares.

mr-7

Skills

Know what multiplexers, decoders, encoders, and priority encoders do.

Know how they are implemented using basic logic gates.

Know how to implement logic functions using these devices.

Know how to make larger devices of these types from smaller ones.