Name ______________________________

Digital Logic I

EE 2720-2

Midterm Examination 10_2

9 November 2011, 14:40–15:30 CST

Exam Rules

Use only a pencil or pen. No calculators of any kind are allowed. Texting is out of the question.

Problem 1	________	(22 pts)
Problem 2	________	(22 pts)
Problem 3	________	(22 pts)
Problem 4	________	(12 pts)
Problem 5	________	(12 pts)
Problem 6	________	(10 pts)
Exam Total	________	(100 pts)

Alias ______________________________

Good Luck!

Problem 1: (22 pts) The problems below are based on the following Boolean function:

$$(a + bc + b'c')(abc')'$$

(*a*) Draw a logic diagram (using AND, OR, and NOT gates) corresponding to the Boolean function. (Do **not** simplify the expression.)

☐ Logic diagram.

(*b*) Write the Boolean function in minterm canonical form. (Show a Boolean expression, not just a list of minterm numbers.) *Hint: For most people directly constructing a truth table would be easier than algebraic manipulation.*

☐ Expression in minterm canonical form.

(*c*) Write the Boolean function in maxterm canonical form.

☐ Expression in maxterm canonical form.

(*d*) Draw a Karnaugh map for the expression. (Just draw the Karnaugh map, don't use it to simplify the expression.)

☐ Karnaugh map, including variables and row and column numbers.

Problem 2: (22 pts) Consider the Karnaugh map below.

xy \ zw

	1		
1	1		1
1			1
	1	1	

(*a*) Write in the row and column numbers.

☐ Row and column numbers.

(*b*) List all of the prime implicants both on the Karnaugh map above, and as a list below.

☐ Prime implicants circled on Karnaugh map.

☐ List prime implicant expressions below.

(*c*) In the list of prime implicants above, write an "E" next to each *essential* prime implicant.

☐ Write an "E" next to essential prime implicants.

(*d*) Provide an example of an implicant that's neither a prime implicant, nor a minterm. Circle this implicant and show the corresponding Boolean expression.

☐ Circle implicant that's not just a minterm but not prime either.

☐ Show an expression for the implicant.

(*e*) Based on the Karnaugh map show a minimum-cost expression for this logic function.

☐ Minimum-cost expression.

Problem 3: (22 pts) Consider the Boolean function below:

$$ab' + b'c + a'bc'$$

(a) Use a 3×8 decoder plus whatever logic gates are needed to implement this function.

☐ Implement using 3×8 decoder and gates.

(b) Use an 8-input multiplexer to implement this function.

☐ Implement using an 8-input multiplexer.

(c) Use a multiplexer and additional logic, including possibly exclusive-or gates, to implement this function by performing a Shannon expansion with respect to a (use a as the multiplexer control input). *Hint: it might be easier to eyeball a truth table than to do this by algebraic manipulation.*

☐ Implement using a multiplexer based on a.

Problem 4: (12 pts) Show how to implement the 8-input multiplexers described below. In each case the three select input bits should be labeled s_2, s_1, s_0, with s_0 being least significant. Label the data inputs 0 to 7.

(*a*) Implement an 8-input multiplexer using two 4-input multiplexers and a 2-input multiplexer.

☐ Eight-input mux using two 4-input multiplexers.

(*b*) Implement an 8-input multiplexer using four 2-input multiplexers and one 4-input multiplexer.

☐ Eight-input mux using four 2-input multiplexers and a 4-input mux.

Problem 5: (12 pts) Implement the devices as described below.

(a) Show the logic gates needed to implement a 2×4 decoder, include an enable input.

☐ Logic diagram for a 2×4 decoder, just use gates.

☐ Include logic for enable input.

(b) Show how to implement an 8-input multiplexer using a decoder and logic gates.

☐ Logic diagram for an 8-input multiplexer using gates and a decoder.

Problem 6: (10 pts) Answer each question below.

(a) Consider five seats, numbered 0 to 4, arranged in a circle and described by Boolean variables i_0 to i_4. Boolean variable i_0 is true if seat 0 is occupied and i_0 is false if the seat is not occupied (no one is sitting in the seat), likewise for i_1, i_2, i_3, and i_4.

Write a Boolean expression that's true if at least two people are sitting next to each other and at least one seat is not occupied. (Note: Just write one Boolean expression.) *Hint: This can easily be solved without a truth table.*

☐ Boolean expression.

(b) The statement below is not true. Explain why and correct it.

"By implementing a sum-of-products expression using only NAND gates (in place of AND and OR gates) we expose additional opportunities for simplification."

☐ Statement is incorrect because ...

☐ The real reason for using NAND gates is ...