CS 14 Spring 2002 — Mid-term #1

This is a closed book, closed note exam. Answer all of the questions **clearly, completely, and concisely**. You have 50 minutes, so be sure to budget your time. All work should be written in your blue book.

1. (15 points) Use a Karnaugh map to simplify the boolean function described by the truth table below. Draw your rectangles clearly and express your result as a boolean algebraic equation – **do *not* draw a circuit.**

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

2. (15 points) Show that the NOR operator is sufficient to express any combinational logic formula. That is, show that NOR can be used to compute NOT, OR, and AND operators. Present a circuit for each operator, labeling its inputs and outputs clearly.

3. Provide brief answers to the following questions about MIPS. (You answers should not be more than a paragraph.)
   (a) (5 points) Consider the use of a branching instruction where the *branch target* (that is, the location to which the instruction may jump) is to an earlier instruction. How is this "backwards jump" represented in the machine instruction?
   (b) (5 points) The `li` instruction is a pseudoinstruction. How is this instruction handled given that the instruction isn't "real"? Why does it not exist as a real instruction?
   (c) (5 points) What if the `jal` instruction were removed from the MIPS instruction set. Would procedure calls still be possible? Justify your answer.

4. (15 points) A *demultiplexor* is complementary to a multiplexor. It has one input, and chooses to connect that input to one of n outputs. (Output lines that are not currently being selected emit a 0.) Draw a circuit for a demultiplexor with a 1-bit input and 4 1-bit outputs.

5. (20 points) Assume that you have an EEPROM (that is, a ROM whose contents you can set) that has 8 addressable locations, where each location stores 2 bits. Demonstrate how this EEPROM can be used to implement the following two combinational formulas:

$$Y = \overline{AB}C + A\overline{BC} + \overline{ABC}$$

$$Z = \overline{ABC} + A\overline{BC} + AB$$

6. (20 points) Construct a sequential circuit that cycles repeatedly through the following 3 element pattern with 2 bits per element: *10, 11, 10, ...*