Name:____
Perm #:______________________________________

Lab Section TA:________________________________

ECE 152A-Fall 2007

Prof. Volkan Rodoplu

MIDTERM EXAMINATION

INSTRUCTIONS:

1. READ THIS PAGE THOROUGHLY WHEN YOU RECEIVE IT, AND WRITE YOUR NAME, PERM #, AND LAB SECTION TA NAME AT THE TOP.
BUT DO NOT START TURNING TO THE OTHER PAGES UNTIL YOU ARE INSTRUCTED TO DO SO.

2. WHENEVER INDICATED, YOU MUST WRITE YOUR ANSWERS ON THE ANSWER LINES PROVIDED IN CERTAIN PROBLEMS. NO PARTIAL CREDIT WILL BE GIVEN ON THESE PROBLEMS. (We will check only the answer on that line.)

3. On other problems, PARTIAL CREDIT will be given only to true statements that make progress towards the correct answer. Partial credit may be given to correct reasoning in developing structures such as K-maps and truth tables in which the variables are clearly labeled. NO partial credit will be given for incorrect statements or statements to which no truth value can be assigned (such as a bunch of numbers or algebraic expressions). NO partial credit will be given for statements that use symbols that the problem statement or you have not defined. NO credit will be given for any work that is not clearly labeled with the part and problem number to which this work provides an answer.

4. All the exam rules in the course syllabus apply to this exam.

5. You may remove the staple from the exam pages, if is more convenient. We will provide a stapler at the end of the exam.

6. There are a total of 140 points on this exam.

7. There are a total of 7 problems, and 16 pages on this exam. When you are instructed to start, first check that you have all of the pages.

STUDENTS MUST LEAVE THIS PAGE BLANK

Problem Number	Points	Out of
1		25
2		25
3		18
4		16
5		16
6		20
7		20
TOTAL		140

PROBLEMS:

DIFFICULTY LEVEL 0

Problem # 1 COMBINATIONAL LOGIC DESIGN [25 points]

Design a "decrement-by-1" **combinational logic** block that takes a **2-bit unsigned** odd number x and computes x – 1 (that is, x minus 1).

(a) **(2 points)** Define your input and output variables.

(b) **(3 points)** Draw the interface schematic (also known as the "top-level schematic" or "symbol"). Clearly show all of the input and outputs. For vector variables, indicate their bitwidths.

(c) **(5 points)** Write down the truth table. (**Your design MUST make use of the fact that the input x is guaranteed to be an odd number**.)

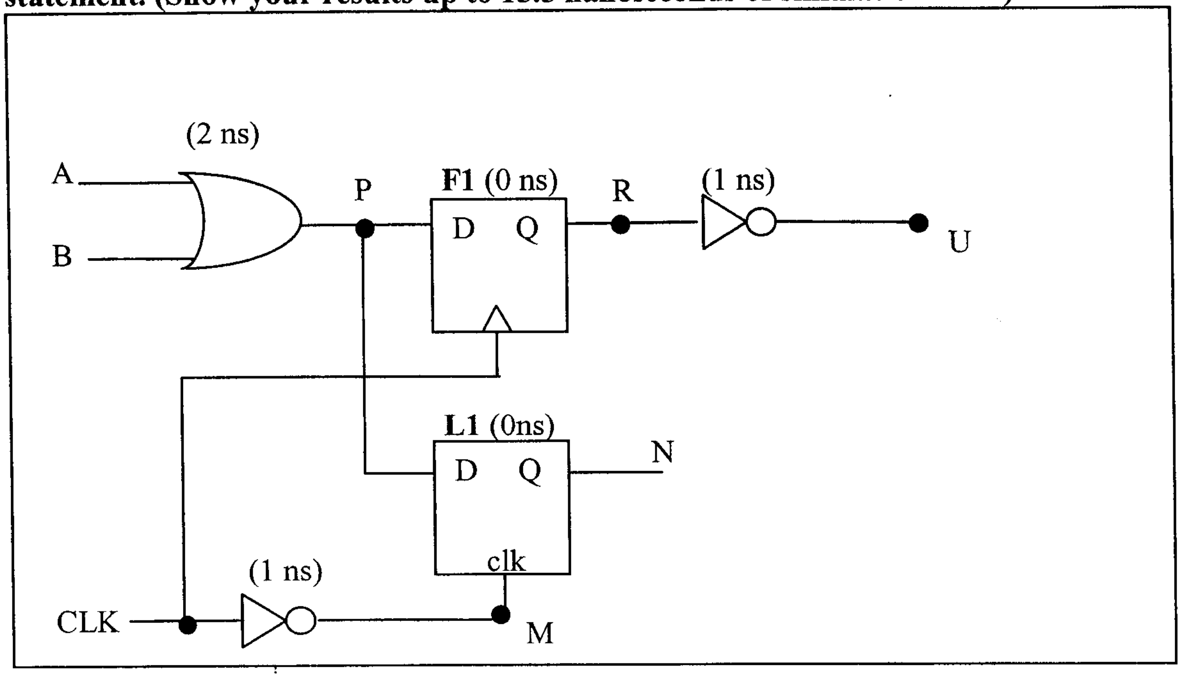
(d) **(10 points)** Perform Boolean minimization using K-maps and obtain a minimum sum of products (SOP) expression for each output. (Your final expressions must be in **algebraic form**.)

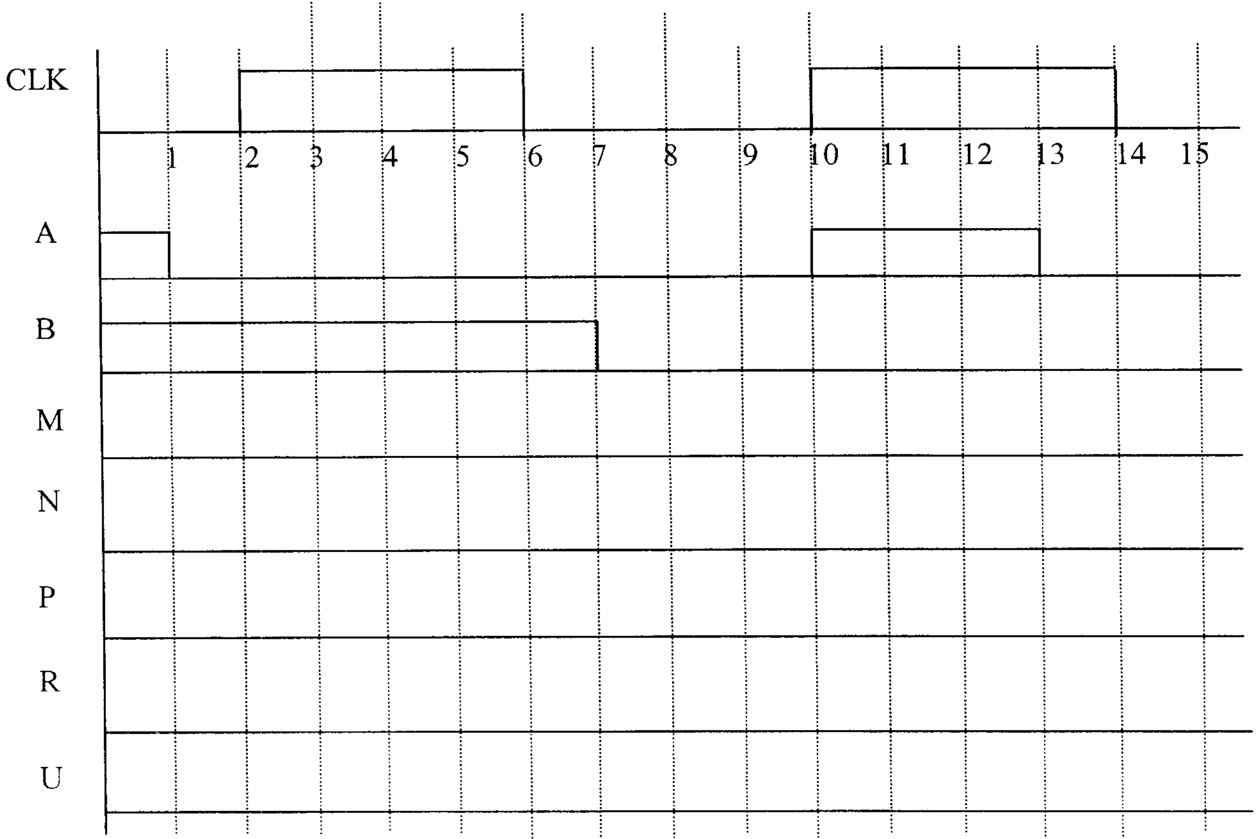
(e) **(5 points)** Draw the circuit schematic that implements the minimum sum-of-products circuit.

DIFFICULTY LEVEL 1

Problem # 2 TIMING DIAGRAMS [25 points]

The circuit for this problem is given below. **See the next page for the problem statement. (Show your results up to 15.5 nanoseconds of simulation time.)**





In this circuit, L1 is a positive level-sensitive D latch.

F1 is **positive** edge-triggered D flip-flop.

CLK is the clock input signal.

A and B are input signals.

M, N, P, R, and U are nodes in the circuit.

The gate delays have been shown in the figure (and are listed below):

Inverter:	1 ns
OR gate:	2 ns

The propagation delays of latch L1 and flip-flop F1 are negligible (0 ns).

External wires have negligible (i.e. zero) delay.

Assume that the inputs A and B have been stable for a very long time before time t = 0, at their initial values shown: A = 1, B = 1.

Assume that the CLK has been running for a long time before t = 0, in the pattern shown in the figure. (The clock period is 8 ns.)

Complete the timing diagram **that appears beneath the circuit on the previous page**. The waveforms for the inputs A, B, and CLK are given.

Fill in the waveforms for M, N, P, R and U.

Each waveform in your answer MUST be shown up to 15.5 nanoseconds of the simulation. (See the time axis on the diagrams on the previous page.)

Grading: Each waveform is worth 5 points. For each waveform, no partial credit will be given.

Problem # 3 COMBINATIONAL LOGIC DESIGN [18 points]

This problem will be concerned with the following Boolean function:

$$F = A B + C'$$

Note that the literals A and B are uncomplemented, whereas the literal C is in complemented form in this expression.

(a) **(8 points)** Implement the function F, as a complete Verilog module, that uses **ONLY continuous assignments (that use the "assign" keyword)**.

(A Verilog module that uses any other types of assignments will get zero credit.)

(b) **(10 points)** Draw the circuit schematic that implements the function F, using only NOR gates.

Instructions:

- Any implementation that uses any other gate will get zero points. **In particular, you may not assume that inverted inputs are available.**
- You do not need to find the implementation that uses the minimum number of NOR gates.
- **You MUST place a BOX around your final answer. (We will check only the final answer, i.e. you need not show how you got to that answer.)**

Problem # 4 VERILOG [16 points]

Write a complete Verilog module that implements a 4:1 MUX using **only procedural assignments.** (Implementations that use other techniques will get zero credit..)

You MUST use the vector notation in Verilog, for the mux inputs. Implementations that do not utilize the vector notation will be penalized.

DIFFICULTY LEVEL 2

Problem # 5 SHIFT REGISTER [16 points]

Using D flip-flops as memory elements, design a 2-bit synchronous shift register (which holds only two bits in memory), that can shift to the left, or to the right, based on a single control input ShiftRight.

If ShiftRight is asserted, then the shift register shifts to the right, and if ShiftRight is not asserted, it shifts the elements to the left, on the active clock edge. The bit that falls out of the memory disappears (that is, it is not fed back in any manner to the shift register).

There is a single data input called SerialInput. The shift register uses the same SerialInput for both the left shift, and the right shift modes.

(a) **(1 point)** Draw the interface schematic for the 2-bit synchronous shift register. Make sure to show all of the inputs, outputs and the bitwidths of each.

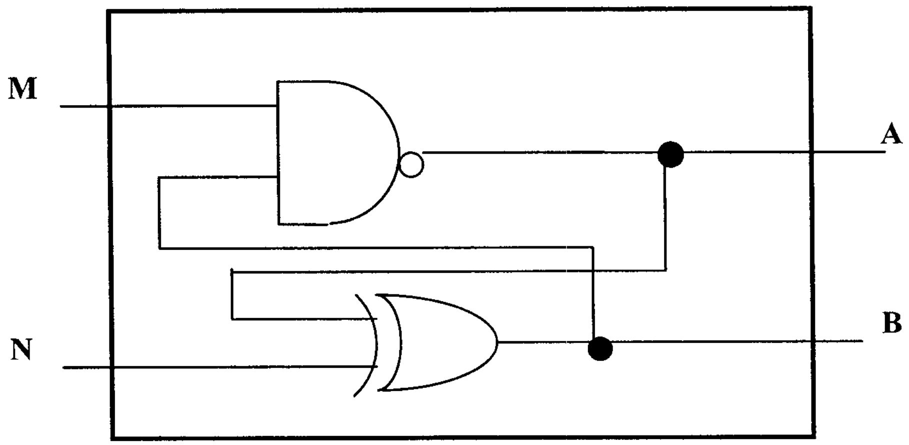
(b) **(15 points)** Draw the implementation schematic for your 2-bit shift register. (Note that you are allowed to use any combinational logic that you wish to implement the logic around the memory elements.)

You must clearly show where the outputs are taken from in the circuit, and the input and output names MUST match those in Part (a).

Problem # 6 A NEW DEVICE [20 points]

Your friend Bittwiddle has proposed a new device that is shown below.

The device consists of two cross-coupled gates: The top one is a NAND gate, and the bottom one is an XOR gate.



The inputs to the device are the terminals M and N, and the outputs are A and B.

(a) **(6 points)** Write down a table that describes the operation of this device, for each of the input combinations.

(b) **(4 points)** Is this a combinational logic, or a sequential logic device? Why?

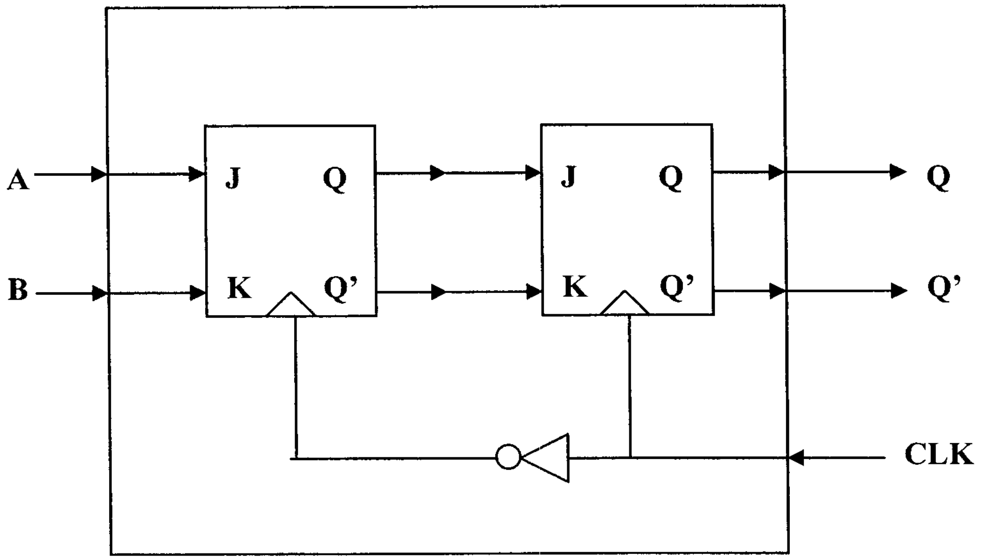
(c) **(5 points)** Can this device be used as a memory element? If so, explain precisely how. Which terminals will be used as input? Which terminals will be used as output? Clearly describe the operation of your memory device in a table, as in Part (a).

(d) **(5 points)** Can you think of any other uses for this device? Be clear and precise when you propose other uses, with a precise description of the settings on the inputs, and the resulting behavior at the outputs.

DIFFICULTY LEVEL 3

Problem # 7 A NEW FLIP-FLOP [20 points]

Your friend Tilda has designed a new flip-flop, the "A-B flip-flop", using two J-K flip-flops and an inverter, as shown below.



The data inputs are A and B, and the outputs are Q and Q ' . The CLK input is as shown above.

Recall that the J-K flip-flop operates as follows:

If J = 1, and K = 0, then Q = 1.
If J = 0, and K = 1, then Q = 0.
If J = 0, and K = 0, then it holds its current state.
If J = 1 and K = 1, then it toggles its current state.

Assume that the inverter delay is 0 ns; that is, the CLK arrives at the two J-K flip-flops at the same time.

(a) **(10 points)** Assume that you have the A-B flip-flop available as a memory element. **Implement a D flip-flop using ONLY A-B flip-flops as memory elements.** (Note: In your implementation, please instantiate the A-B flip-flop using only its interface, not its entire internal implementation. Also, you may use additional combinational logic as needed.)

If it is not possible to implement a D flip-flop using only A-B flip-flops as memory elements, then prove this.

(b) **(10 points)** This part is completely separate from Part (a). We want to do a complete re-design of the A-B flip-flop, using a different implementation, but preserving its interface.
Assume that D flip-flops are available as building blocks. Implement the A-B flip-flop using ONLY D flip-flops are memory elements. (Note: You may use additional combinational logic; however, you are not allowed to use other types of flip-flops or latches, besides D flip-flops.)