Name:_____
Perm #:_____

Lab Section TA:_____

ECE 152A-Winter 2007

Prof. Volkan Rodoplu

MIDTERM EXAMINATION

INSTRUCTIONS:

1. READ THIS PAGE THOROUGHLY WHEN YOU RECEIVE IT, 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.

STUDENTS MUST LEAVE THIS PAGE BLANK

Problem Number	Points	Out of
1		25
2		10
3		25
4		15
5		15
6		25
7		15
8		20
TOTAL		180

PROBLEMS:

Problem # 1 COMBINATIONAL LOGIC DESIGN [25 points]

1. (a) Draw the circuit schematic that implements

$$F = A' B + C '$$

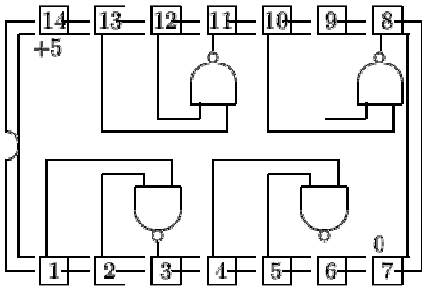
using only NAND gates. Your implementation must use at most 4 NAND gates.

Instructions: a. Any implementation that uses any other gate will get zero points. In particular, you may not assume that inverted inputs are available.

b. You do not need to find the implementation that uses the minimum number of NAND gates. Any correct implementation that uses less than or equal to 4 NAND gates is acceptable.

c. **You MUST place a BOX around your final answer.**

(b) Given the TTL chip below, show all connections (including the ones to VDD and GND) to implement the function F in Part (a) of this problem.



Problem # 2 COMBINATIONAL LOGIC DESIGN [25 points]

Implement a circuit that adds two 2-bit unsigned numbers A and B if and only if a switch K=1. If K=0, the circuit outputs a 2-bit number whose least significant bit (LSB) is the LSB of A, and whose MSB is the LSB of B.

EACH PART BELOW GETS EITHER ZERO OR FULL CREDIT (i.e. no partial credit will be given for each part.)

(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) **(10 points)** Write down a minimum sum of products (SOP) expression for each output. (Your final expressions must be in **algebraic form**.) You may do this via K-maps, or you may observe the structure of the problem and write down the answer in minimum SOP form, by inspection.

(d) **(10 points)** Write down a minimum product of sums (POS) expression for each output. (Your final expressions must be in **algebraic form**.)

Problem # 2 VERILOG [10 points]

(a) **(5 points)** Write a complete Verilog module that implements a 2:1 MUX using the

? :

ternary conditional operator. (Implementations that use other techniques will get zero credit. No partial credit will be given on this part of the problem.)

(b) **(10 points)** By instantiating your 2:1 MUX from Part (a), write a complete Verilog module that implements a 4:1 MUX. You must use only instantiations of 2:1 MUXes, and no other combinational logic in your implementation. You must also use the minimum number of 2:1 muxes that are necessary to implement a 4:1 mux. (Any other implementations will get zero credit.)

Problem # 3 A NEW GATE [20 points]

Your friends in solid-state and device electronics have built a new gate that performs a new binary operation, denoted by *, that is defined as follows:

$$A * B = A' B$$

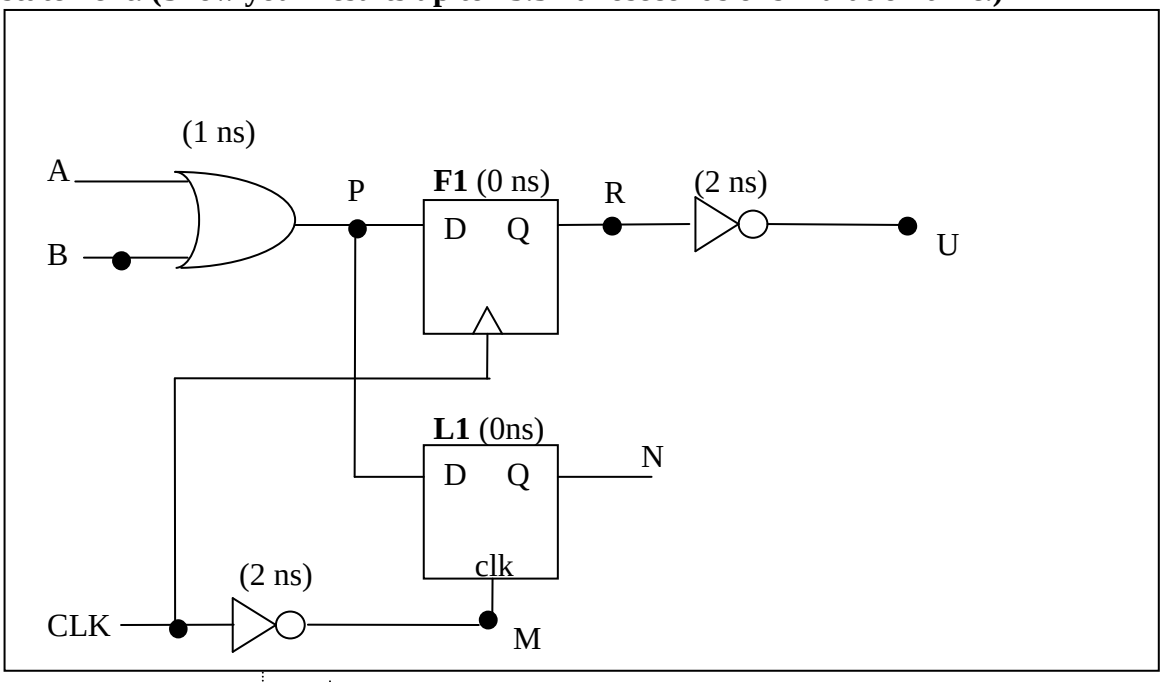
(a) **(8 points)** Is * commutative? Prove your answer.

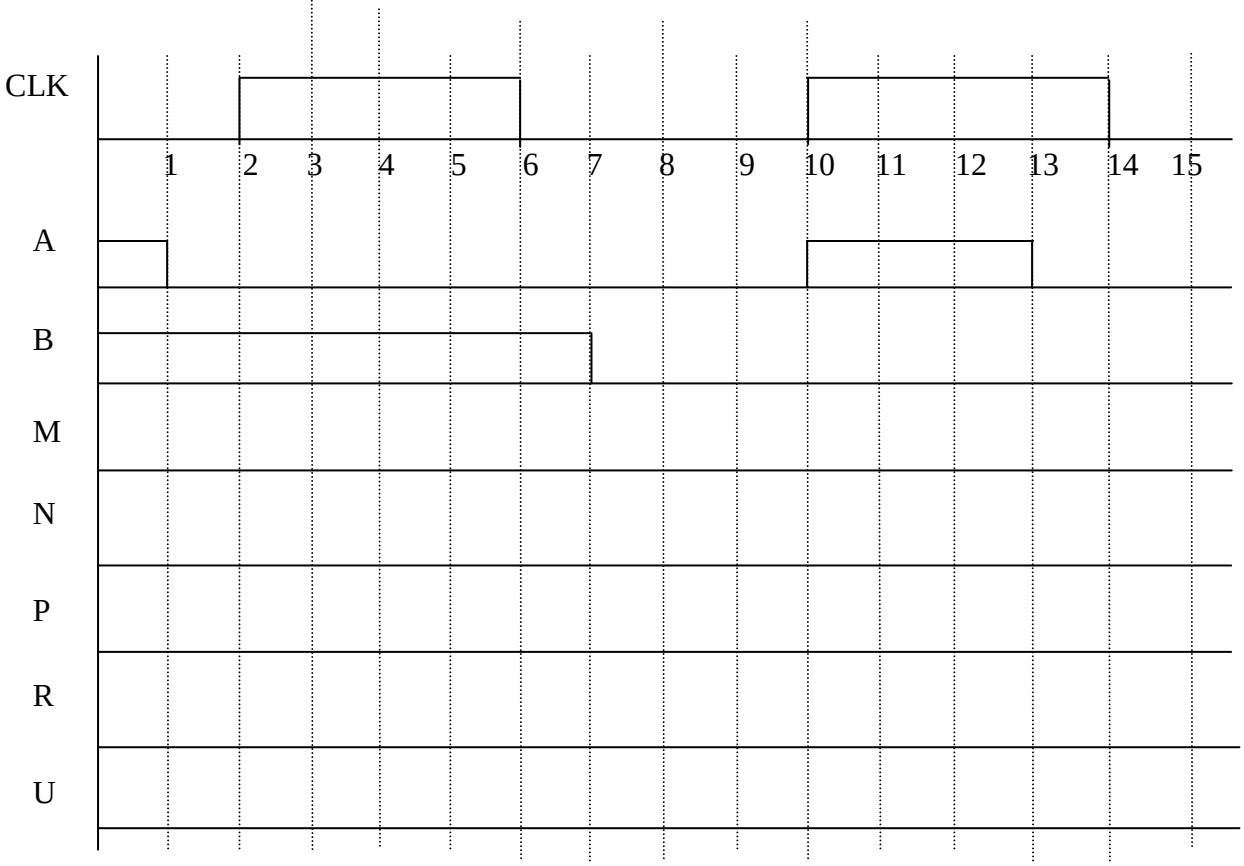
(b) **(8 points)** Is * associative? Prove your answer.

(c) **(4 points)** Find the dual of *.

Problem # 4 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:	2 ns
OR gate:	1 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.

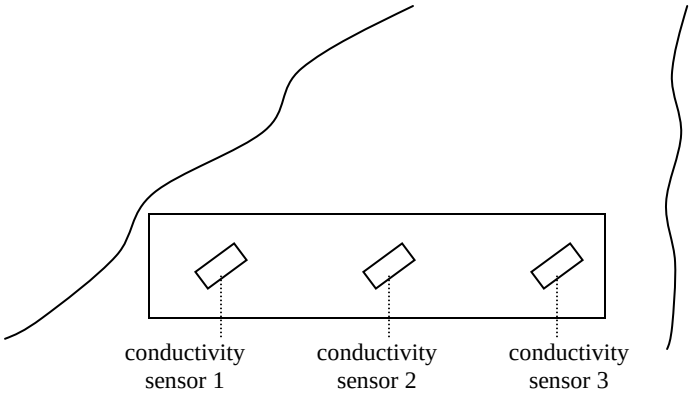
Fill in the waveforms for M, N, P, R and U.

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 # 5 A SENSOR ARRAY [15 points]

A sensor array is used to measure the conductivity of water at the mouth of a river. Conductivity is represented as a real positive number



Let M_i represent the conductivity measured at array element i, $i \in \{1,2,3\}$.

Your job is to build a detector device that signals "Alert!" if and only if the conductivity simultaneously measured by any two adjacent sensors in this array differs by more than 10%. (Note that sensors 1 and 2 are adjacent, and 2 and 3 are adjacent.)

Define all of your variables. (You must make use of M_i's in your definitions).

Draw the schematic for the controller. (Place your final schematic in a box.)

Problem # 6 DELAY ANALYSIS [25 points]

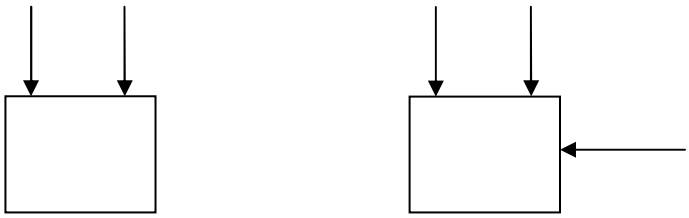
(a) **(5 points)** By drawing the schematic, implement a 2-bit half adder using only one 2-input XOR gate, and one AND gate.

(b) **(10 points)** By drawing the schematic, implement a 2-bit full adder using only two 2-input XOR gates, two 2-input AND gates, and one 2-input OR gate.

(c) **(5 points)** In this part, we implement a 4-bit adder (that adds two 4-bit unsigned numbers) by connecting the half adder from Part (a) and the full adder from Part (b) in cascade; that is, **we are connecting a full adder and a half adder in a ripple carry adder structure**.

The inputs are A[3:0], B[3:0] and C0, and the outputs are sum[3:0] and a signal that indicates overflow. Your 4-bit adder must be able to detect OVERFLOW.

In the implementation schematic below, show how to connect the two adders. YOU MUST LABEL ANY REMAINING VARIABLES CLEARLY ON THE FIGURE.
.



"2nd stage" "1st stage"

(b) **(15 points)** Assume that all of the input variables are available at time t = 0 (indicated by @0).

THE GATE DELAYS ARE AS FOLLOWS:

AND	3 ns
OR	4 ns
Inverter	1 ns
XOR	2 ns

Perform the delay analysis to calculate the worst-case delay (in nanoseconds) of the 4-bit adder in Part (c). On the figure in Part (c), you must show the delays (using the @ notation) on all of the wires that appear in the figure. (Note that no delays should be shown for the internal wires of the 2-bit adders.)

You may use the rest of this page for your scratch work (e.g. you may expand and draw the full and half adders below, and trace through the delays.)

YOU MAY NOT ASSUME A UNIT DELAY FOR EACH GATE. YOU MUST USE THE TABLE ABOVE FOR THE GATE DELAYS.

Write your final answer here: Worst-case delay of 4-bit adder is :_____ ns.

Problem # 7 COUNTER DESIGN [15 points]

Design a synchronous, edge-triggered counter that operates as follows: the counter does not have a Reset button and **may potentially start at any count**. However, **after the minimum number of cycles possible**, it should start counting in the sequence 7, 6, 5, 4, 7, 6, 5, 4, . . ., and it repeats this sequence forever.

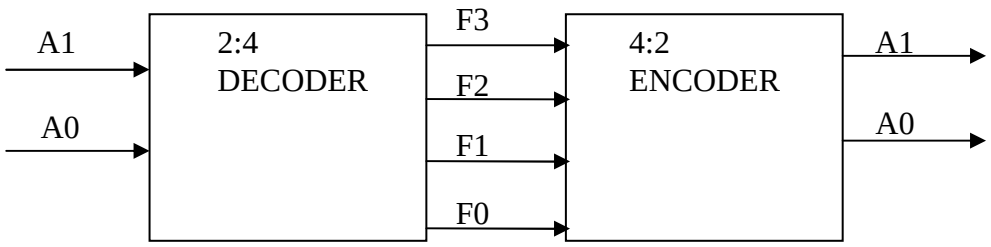
(a) **(1 point)** Define the input, output and state variables for this counter. (You must show all the bitwidths for any vector variables.)

(b) **(1 point)** Draw the interface schematic for this counter. (Show all the bitwidths on the schematic for any vector variables.)

(c) **(13 points)** Write down both the state diagram and the **state table** for this counter. You must implement it as a MOORE machine. (Write down the state diagram for your own convenience, but we will grade only the state TABLE.)

Problem # 8 ENCODER DESIGN IN VERILOG [20 points]

Write a complete Verilog module that uses **only continuous assignments** (which use the "assign" keyword) to implement an 4:2 "encoder", which is defined as the right inverse of a 2:4 decoder.



That is, the encoder takes the output of the 2:4 decoder and must produce the original input into the decoder.

- Implement ONLY the Encoder, not the decoder!
- YOUR VERILOG MODULE MUST USE THE SAME VARIABLES AS SHOWN IN THE DIAGRAM ABOVE, HOWEVER WRITTEN IN VECTOR FORM. YOU WILL BE PENALIZED IF YOU DO NOT USE THE VECTOR NOTATION IN VERILOG.

(You may continue your solutions to Problem # 8 on this page.)