Name:_____
Perm #:_____

Lab Section TA:_____

ECE 152A-Summer 2008

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		10
2		10
3		23
4		30
5		20
6		21
7		18
8		18
TOTAL		150

PROBLEMS:

DIFFICULTY LEVEL 0

Problem # 1 KARNAUGH MAPS [10 points]

(a) **(5 points)** The Karnaugh map ("K-map") of the Boolean function F is below.

B / A	0	1
0	x	x
1	1	1

Mark ALL of the essential prime implicants, and write down each essential prime implicant below in algebraic form.

(b) **(5 points)** By drawing the schematic, design a circuit to compare two 1-bit unsigned numbers, A and B.

If A > B then Z = 1, otherwise Z = 0.

DIFFICULTY LEVEL 1

Problem # 2 NAND, NOR GATES [10 points]

(a) **(5 points)** Implement a NAND gate using only NOR gates. If it is not possible, prove that it is not possible.

(b) **(5 points)** Implement a NOR gate using only NAND gates. If it is not possible, prove that it is not possible.

Problem # 3 VERILOG [23 points]

Write a complete Verilog module to implement a 2-input permutation network (shown below).

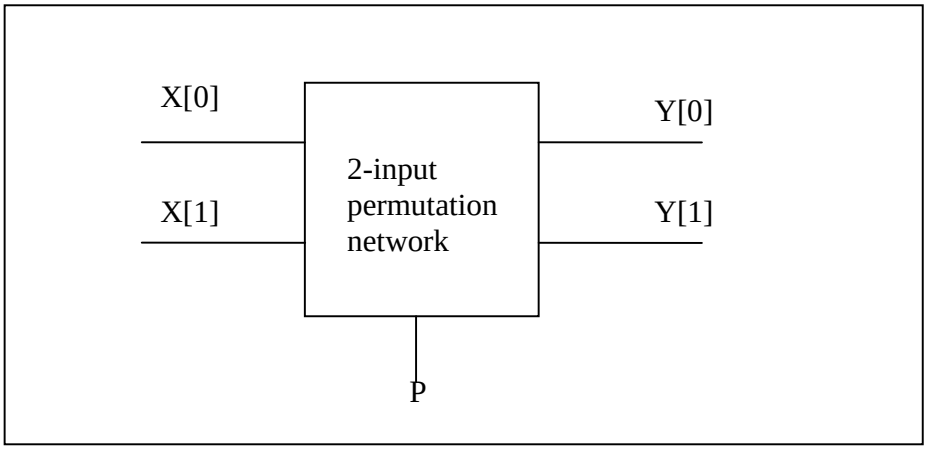


Figure 1

The operation is as follows:
If P == 1, then Y[1] = X[0] and Y[0] = X[1]
If P == 0, then Y[1] = X[1] and Y[0] = X[0].

YOU MUST USE THE VECTOR NOTATION IN VERILOG IN ALL OF YOUR IMPLEMENTATIONS IN THIS PROBLEM.

(a) **(9 points)** Write this as a complete Verilog module using **only continuous assignments (which use the "assign" keyword).**

(You may continue your work here.)

(b) **(14 points)** Write this as a Verilog module using **only procedural assignments**.

Problem # 4 SCISSORS-PAPER-STONE GAME [30 points]

In this problem, you will implement a **combinational logic circuit** that will be used in the implementation of one round of the Scissors-Paper-Stone game.

This game is played with 2 players: At each round of the game, each player enters "Scissors", "Paper" or "Stone" as an input. The player cannot see what the other player enters. Then, the circuit compares what the two players have entered, and announces the result, which is one of the following: "Player A wins", "Player B wins", or "Tie".

The game rules are as follows: Let A and B be any pair of labels of the two players. (That is, all of the arguments that will follow will apply also if A and B labels are interchanged.) If Player A enters "Scissors" and Player B enters "Paper", then Player A wins (because the Scissors can cut the paper). If Player A enters "Paper" and Player B enters "Stone", then Player A wins (because Paper can wrap the Stone). If Player A enters "Stone" and Player B enters "Scissors", then Player A wins (because Stone can smash the Scissors).

If the players both enter the same input (both Scissors, both Paper, or both Stone), then it is a "Tie" between the two players.

We would like you to build a combinational logic circuit that takes the inputs from the two players, and reports the output of the game (for only one round as described above) by writing answers to the following parts:

(a) **(8 points)** Define your Boolean inputs and outputs. Clearly show what each input and output combination corresponds to in the actual game.

(You may continue your work here.)

(b) **(8 points)** Write down the **truth table**, based on your choice of the variables in Part (a) of this problem. (Your variables MUST match your declarations in Part (a).)

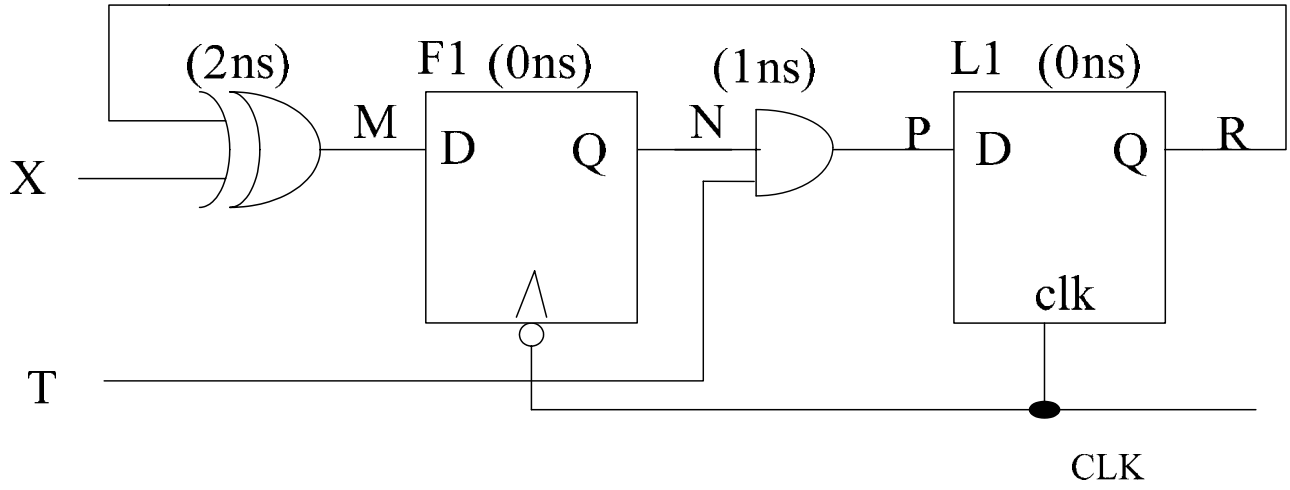
(c) **(14 points)** Draw the **K-map** for each output variable, and write down the **algebraic expression** for the **minimum SOP** for each output variable.

(You may continue your work here.)

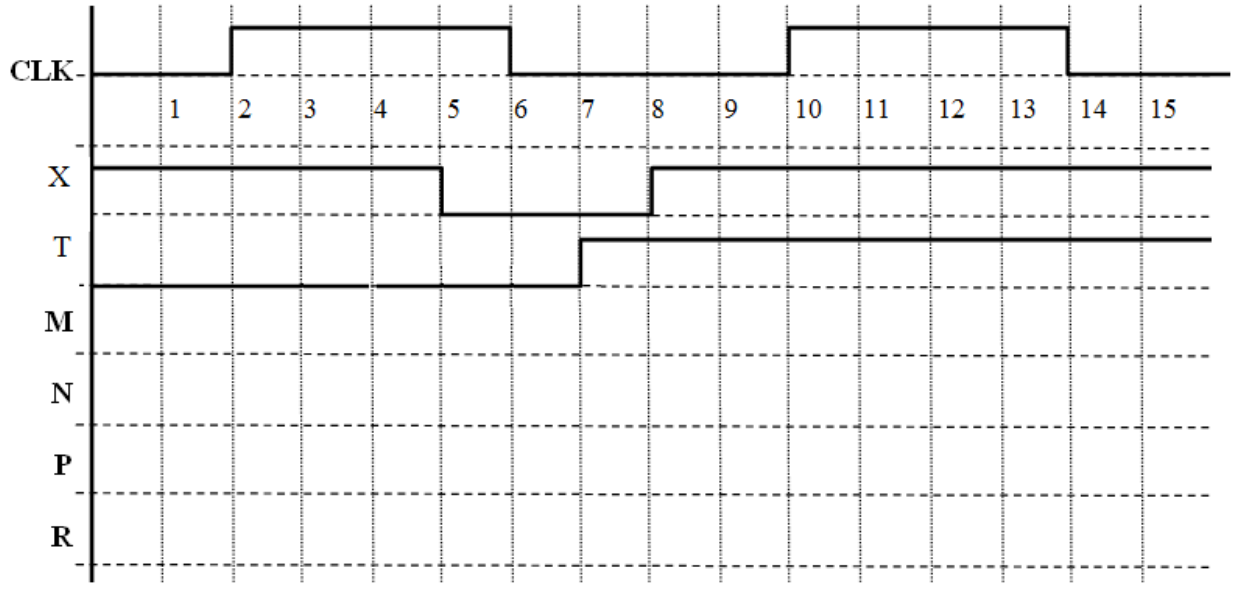
(You may continue your work here.)

Problem # 5 LATCHES AND FLIP-FLOPS [20 points]

This problem concerns the circuit below:



The problem statement is on the next page. Fill out the timing diagram below. Your answers must go until the end of 15.5 nanoseconds of simulation time.



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

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

CLK is the clock input signal.

X and T are the input signals.

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

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

XOR gate: 2 ns
AND 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 X and T have been stable for a very long time before time t = 0, at their initial values shown: X = 1 and T = 0.

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 X, T and CLK are given.

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

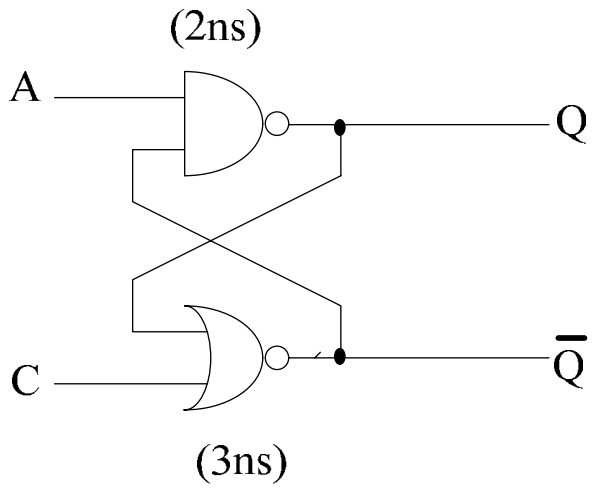
Each waveform in your answer MUST be shown up to 15.5 nanoseconds of the simulation. (See the time axis on the diagrams on the previous page.)

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

DIFFICULTY LEVEL 2

Problem # 6 A NEW DEVICE [21 points]

Your friend Tilda has proposed a new device, as a new type of latch. The proposed device is shown in the figure below. The upper gate is a NAND gate, and the lower one is a NOR gate. We will now investigate the operation of this device.



(a) **(8 points)** Draw a table that shows the operation of this device for each of the possible input combinations.

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

(c) **(9 points)** Does this circuit behave as a latch? If it does, show exactly why. If it does not, modify the circuit such that it operates as a latch. What is the definition of a latch? Clearly show how your modified circuit now behaves as a latch. If it cannot be turned into a latch, then explain why.

(You may continue your work here.)

Problem # 7 2:4 DECODER [18 points]

By drawing the schematic, implement a 2:4 decoder using **ONLY** 2:1 muxes. (You are NOT allowed to use any other combinational logic elements besides 2:1 muxes.)

Aim for an implementation that uses the minimum number of 2:1 muxes possible. We will first check for functional correctness. If your answer is functionally correct, then we will check for minimality. (If your answer is not minimal, you will lose some points.)

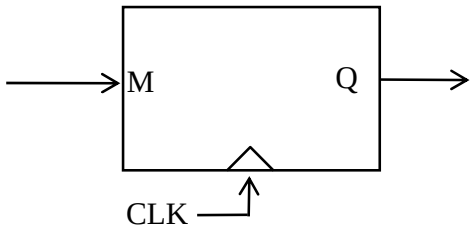
(YOU MUST FIRST WRITE DOWN A TRUTH TABLE THAT CLEARLY DEFINES YOUR VARIABLES. USE VECTOR NOTATION. WE WILL CHECK YOUR IMPLEMENTATION AGAINST THE VARIABLE NAMES IN YOUR TRUTH TABLE.)

(You may continue your work here.)

DIFFICULTY LEVEL 3

Problem # 8 A NEW FLIP-FLOP [18 points]

Your friend Bittwiddle has proposed a new flip-flop that has the following symbol.



Above, M is the data input, Q is the output, and this is a positive edge-triggered device.

The operation is as follows: The device samples M and makes it available at Q, however, with the following difference: The device samples the input M on the rising edge of the CLK and the output Q becomes available only at the falling edge of the CLK in the same clock cycle.

Your job: By drawing the schematic, implement this flip-flop using **only NAND gates**.

We will first check for functional correctness. If it is functionally correct, then we will check to see whether your implementation uses a reasonable number of NAND gates.

(You may continue your work here.)

(You may continue your work here.)