Name: _____

Logic Design (Fall 2009)

Prof. Chang-Gun Lee

Time: 75 minutes

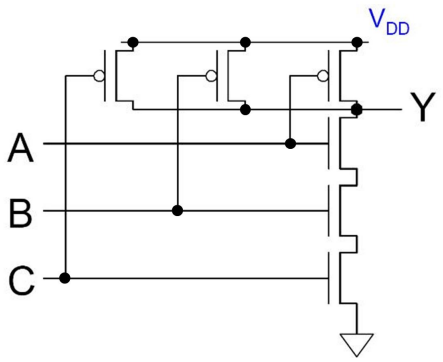
Instructions:

1. This is a closed book examination. Any form of cheating on the examination will result in a zero grade.
2. Do all your work inside this booklet, using the backs of pages if needed. The problems are of varying degrees of difficulty so please pace yourself and answer the questions in the order which best suits you. Answers to questions should be as brief and specific as possible.
3. Partial credit will be given even if your final solution is incorrect, if you show the intermediate steps you take in getting to the final solution. Clearly state any assumptions you make in your answers, and justify your assumptions.
4. Please write your answers neatly and, in case of numerical problems, *put the final answer in a box.* Good luck!

Problem No.	Max Points	Received Points
1	5	
2	5	
3	5	
4	5	
5	10	
6	10	
7	10	
8	10	
9	10	
10	15	
11	15	
12	15	
Total	115	

1. [**5 points**] Why **NOR**, **NAND** are used instead of **OR**, **AND** in practice? Explain your answer briefly.

2. [**5 points**] Briefly explain pros and cons of **ROM, PLA, PAL** in terms of Price, Speed, Design time, Flexibility, and so on.



3. [**5 points**] Draw the truth tables for the above figure. What is this function (***gate***)?

4. [**5 points**] Prove the following formula using laws of Boolean algebra.

$$(\boldsymbol{X} + \boldsymbol{Y}) \cdot (\overline{\boldsymbol{X}} + \boldsymbol{Z}) = \boldsymbol{X} \cdot \boldsymbol{Z} + \overline{\boldsymbol{X}} \cdot \boldsymbol{Y}$$

5. [**10 points**] Show the arrangement of transistors needed to construct:

   (a) A gate whose output is the value of **A OR (B AND C)**.

   (b) A gate whose output is the inverted value of **(a)**.

6. **[10 points]** To digitally manipulate information we must encode it using binary numbers. consider an encoding for playing cards. There are 52 cards in a deck and they are divided into four suit with values from 1 to 13 (an ace is 1, jack is 11, queen is 12, king is 13).

   (a) Show a possible encodings for the cards.

   (b) Given your encoding for playing cards, show the logic expression that describes:

      i. A jack of diamonds.

      ii. A seven of any suit.

      iii. Any card of the heart suit.

7. [**10 points**] Consider a four-input function that outputs a 1 whenever an odd number of its inputs are 1.

   (a) Draw the truth tables for the function.

   (b) Fill in the K-map to find the minimum sum-of-products expression for the function. What is the minimum S-o-P expression?

8. [**10 points**] You got a special deal on some cheap logic packages. Unfortunately, they are all identical and contain two complex logic gates each. The gate's function is $\boldsymbol{Z = \overline{(AB + CD)}}$. Implement the following function using only these parts and state how many packages of this gate you will need for this function. Feel free to set any inputs to 0 or 1 as needed.

$$\boldsymbol{f(P, Q, R, S, T) = PQR + ST}$$

9. **[10 points]** Draw schematics for the following expression, mapped into (a) $\boldsymbol{2-input\ NAND-only}$ and (b) $\boldsymbol{2-input\ NOR-only}$ networks. You may assume that literals and their complements are available.

$$\boldsymbol{f(A,B,C) = \overline{A}B + A + \overline{C} + \overline{D}}$$

10. [**15 points**] Consider the function:

$$\boldsymbol{f(A, B, C, D) = \Sigma m(1, 3, 5, 7, 8, 9, 13, 15) + \Sigma d(4, 12, 14)}$$

(a) Write this as a Boolean expression in canonical minterm form (Ignore **don't care symbols**).

(b) Write the complement of $\boldsymbol{f}$ in **big M** notation.

(c) Fill in the K-map for the above function and find prime implicants, essentials.

(d) Find minimum sum-of-products form for the above function.

11. [**15 points**] Implement the function:

$$\boldsymbol{f(A, B, C) = \Sigma m(0, 2, 6, 7)}$$

(a) using **one** multiplexer and no other logic. The constants logic 1, logic 0, the variables and their complements are available. Try to use the smallest possible multiplexer.

(b) If the variables complements are not available, how can you implement the function using **one** multiplexer? Try to use the smallest possible multiplexer.

12. [**15 points**] Implement to the gate level an ALU bit slice with three selection inputs, S2, S1, S0, that implements the following 8 functions of the two data inputs A and B (and carry-in C0). You can use 2-input gates and MUX/DEMUXs only. If needed, you can explain your own assumption.

S_2	S_1	S_0	*ALU Operation*
0	0	0	$F_i = 0$
0	0	1	$F_i = B$ minus A
0	1	0	$F_i = A$ minus B
0	1	1	$F_i = A$ plus B
1	0	0	$F_i = A$ XOR B
1	0	1	$F_i = A$ OR B
1	1	0	$F_i = A$ AND B
1	1	1	$F_i = 1$