Digital Design and Computer Architecture (252-0028-00L), Spring 2023
Optional HW 1:Combinational Logic

Instructor: Prof. Onur Mutlu
TAs: Juan Gomez Luna, Mohammad Sadrosadati, Mohammed Alser, Ataberk Olgun, Giray Yaglikci, Can Firtina, Geraldo De Oliveira Junior, Rahul Bera, Konstantinos Kanellopoulos, Nika Mansouri Ghiasi, Nisa Bostancı, Rakesh Nadig, Joel Lindegger, İsmail Emir Yüksel, Haocong Luo, Yahya Can Tuğrul, Julien Eudine

Released: Wednesday, Mar. 1, 2023

1 Logical Completeness

The set of {AND, OR, and NOT} gates is logically complete. We can build a circuit to carry out the specification of any truth table we wish, without using any other kind of gate. From Lecture 4, you know that the NOR gate by itself is also logically complete. Prove that you can build a circuit to carry out the specification of any truth table, by using only NOR gates.

2 Boolean Algebra

(a) Find the simplest sum-of-products representation of the following Boolean equation. Show your work step-by-step.

$F = (\overline{A} + B + C).(A + B + \overline{C}).C + A$

(b) Using Boolean algebra, simplify the following min-terms:
$\sum(1111, 1110, 1000, 1001, 1011, 1010, 0000)$. Show your work step-by-step.

(c) Convert the following Boolean equation so that it only contains NAND operations. Show your work step-by-step.

$F = \overline{A} + \overline{(B.C + \overline{A.C})}$

(d) Convert the same Boolean equation given in part (c) so that it only contains NOR operations. Show your work step-by-step.

$F = \overline{A} + \overline{(B.C + \overline{A.C})}$

3 Boolean Algebra and Combinational Logic Design

In this question we ask you to derive the boolean equations for two 4-input logic functions, X and Y. Please use the truth table below to answer the following three questions.

Inputs				Outputs	
A_3	A_2	A_1	A_0	X	Y
0	0	0	0		
0	0	0	1		
0	0	1	0		
0	0	1	1		
0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

(a) The output X is *one* when the input does **not** contain two consecutive 1's in the word A_3, A_2, A_1, A_0. **Fill in the truth table above** and write the corresponding Boolean equation for X, using *only* **2-input *NAND* gates**. Derive the expression using Boolean algebra laws.

(b) The output Y is *one* when no two adjacent bits in the word A_3, A_2, A_1, A_0 are the same (e.g., if A_2 is 0 then A_3 and A_1 cannot be 0). The output Y is *zero*, otherwise (e.g., 0000). **Fill in the truth table above** and use the *sum of products* form to **write the corresponding boolean equation** for Y. (*No simplification needed.*)

(c) Please represent the circuit of Y using *only* 2-input XOR and AND gates.

4 Transistor-Level Circuit Design

In Lecture 4, we learned how to implement digital circuits using the CMOS technology (i.e., p-type and n-type MOS transistors). In this assignment, we ask you to schematically design circuits using CMOS transistors for the following logic gates:

- Exclusive OR Gate (XOR)

- Exclusive NOT OR Gate (XNOR)

5 Multiplexer (MUX)

Draw the following schematics for an 8-input (8:1) MUX.

- Gate level: as a combination of basic AND, OR, NOT gates. Use as few gates as possible.

- Module level: as a combination of 2-input (2:1) MUXes. Use as few 2-input MUXes as possible.