Advanced Computer Architecture & Advanced Microprocessor System

2ND Semester 2023

Homework

Release Date

21/07/2023

Due Date

Morning Session: 27/07/2023 @ 9:00 AM

Evening Session: 27/07/2023 @ 5:30 PM

(**Note: Those outside Accra should send their answers to my email address**)

Overview

This homework is to help you revise digital logic design which is essential for this course.

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 the digital logic course you studied in second year, 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:

∑ (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 I 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 1 when the input does not contain two consecutive 1's in the word **A3**, **A2**, **A1**, **A0**. 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 1 when no two adjacent bits in the word **A3**, **A2**, **A1**, **A0** are the same (e.g., if **A2** is 0 then A3 and A1 cannot be 0). The output **Y** is 0, 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).

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.