King Fahd University of Petroleum and Minerals
College of Computer Science and Engineering
Computer Engineering Department

COE 202: Digital Logic Design (3-0-3)
Term 181 (Fall 2018)
Major Exam 2
Saturday, November 11th, 2018

Time: 120 minutes, Total Pages: 9

Name:_____ **ID:**_____ **Section:** _____

Notes:
Do not open the exam book until instructed
Calculators are not allowed (*basic, advanced, cell phones, etc*.)
Answer all questions
All steps must be shown
Any assumptions made must be clearly stated
Clearly label all inputs and outputs of any circuit component

Question	Maximum Marks	Your Marks
1	**18**	
2	**19**	
3	**20**	
4	**13**	
Total	**70**	

Question 1. **(18 Marks)**

a) Consider the following Boolean function and its don't care conditions:

$$F(A,B,C,D) = \prod M(3,11,12,13) + \sum d(2,6,10,14)$$

Represent F on a K-map along with its don't care conditions. **(4 Marks)**

b) The K-map below represents a function G(X,Y,Z,W):

XY \ ZW	00	01	11	10
00	1	1	0	1
01	1	X	X	1
11	0	X	X	0
10	1	1	0	0

1. List all the **prime implicants** and **essential prime implicants** of G. **(4 Marks)**

2. Minimize G to a minimal SOP expression **(2 Marks)**

3. Minimize G to a minimal POS expression **(4 Marks)**

c) Let F = A' (B' + C') (A+C) Implement F using minimum number of NOR gates **only**. **(4 Marks)**

Question 2: **(19 Marks)**

It is required to design a circuit that receives **a 3-bit signed** number in **signed-magnitude** representation, **X**, and computes the equation **Y = 3*X – 2** where **Y** is also **represented in signed-magnitude** representation.

a. Determine the number of bits needed for the output Y. Justify your answer. **(2 Marks)**

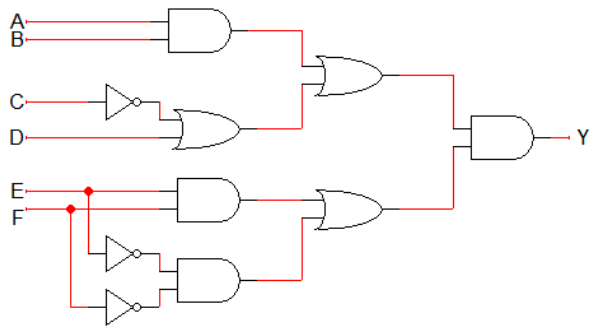
b. Fill the table below with the truth table for the output Y. **(5 Marks)**

X2 X1 X0	Decimal Value of Y	Output Y

c. If the output of the circuit Y is fed as input to another circuit that will compute the equation: **$Z = Y^2$,** what will be the don't care conditions for **Z.** **(2 Marks)**

d. Implement the following two functions using **a decoder** and **additional gates, each** with the **smallest possible Fan-in (i.e. number of inputs). Clearly mark all inputs and outputs of the decoder.** F1 (A,B,C) = ∑m(0,4,5,6,7) F2 (A,B,C) = ∑m(4,5,7) **(5 Marks)**

e. Convert the circuit below to **2-input NAND gates only.** Redraw the circuit to obtain a multi-level NAND circuit implementation. Assume that **both the true and complement** forms of each input variable are available. **Use the circuit as is and do not attempt to simplify it. (5 Marks)**



Question 3. **(20 Marks)**

a) Represent the given signed numbers using **8-bit** signed binary numbers in the specified representations in the table below: **(6 Marks)**

Decimal Value	Sign-Magnitude representation	1's complement representation	2's complement representation
-34			
-67			

b) **A** and **B** below are 8-bit signed integers represented in the 2's complement representation:

A = 11011000 B = 01110101

1. What are the decimal values of A and B? **(2 Marks)**

2. Compute A + B in binary and indicate whether there is overflow or not. **(3 Marks)**

3. Compute A – B by converting subtraction into binary addition. Indicate whether there is overflow or not. Show all the steps. **(3 Marks)**

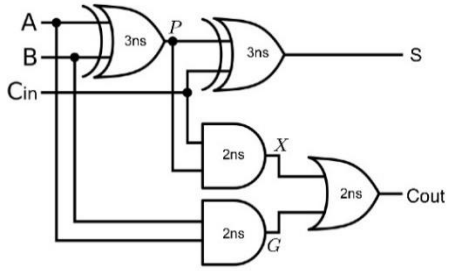
c) Suppose that **X**, **Y**, and **Z** are 4-bit unsigned integers. Let **S = X+Y+Z** be the unsigned sum.

1. What is the number of bits required to represent the output **S**? **(1 Mark)**

2. Draw a circuit that computes **S = X+Y+Z** using only full adders and half adders. **(5 Marks)**

Question 4 **(13 Marks)**

Consider the following implementation of a full-adder circuit:



a) Write **a gate-level** Verilog description of the above full adder using the Inputs, Outputs, and wire names and gate delays shown on the diagram above. **(5 Marks)**

b) Write another Verilog description of the full adder, with delay modeling, using **a single assign statement per output (i.e. two assign statements).** **(3 Marks)**

c) Using the full adder in (a) above, write a testbench to test **a 2-bit adder** (i.e. uses two of the full adder in (a)) to add two 2-bit numbers **A** and **B** with a carry in **Ci**, and produces a 2-bit Sum **S** and carry out **Co**. The testbench should test the 2-bit adder for the following input values (with 10 time units delay in-between the two input patterns): **(5 Marks)**

A = 00, B = 00, Cin = 0, and the second input is **A = 10, B = 01, Cin = 1**.