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 171 (Fall 2017)
Major Exam 2
Saturday, November 25th, 2017

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

Question	Maximum Points	Your Points
1	**15**	
2	**15**	
3	**15**	
4	**10**	
5	**20**	
6	**15**	
7	**10**	
Total	**100**	

Question 1: **[15 points]**

Given the following two 8-bit binary numbers **A** and **B**:

A = 1010 0011 B = 1000 0010

a) What are the decimal values of A and B if they are unsigned numbers? (**2 pts**)

b) What are the decimal values of A and B if they are signed numbers in signed-magnitude? (**2 pts**)

c) What are the decimal values of A and B if they are signed numbers in 2's complement? (**2 pts**)

d) If A and B are signed numbers in 2's complement, **Compute A + B** and **specify if there is an overflow** **(4 pts)**

e) If A and B are signed numbers in 2's complement, **Compute** A - B and **specify if there is an overflow** **(5 pts)**

Question 2. **Let** f(a, b, c, d) = ∑m (0, 2, 3, 4, 5, 8, 11, 12, 13, 14, 15) **[15 Points]**

a) Draw the Karnaugh map of the function f. **(2 points)**

b) Write the terms of ALL the prime Implicants and indicate which ones are Essential. **(7 points)**

c) Find ALL minimum sum-of-product expressions of f **(3 solutions).** **(6 points)**

Question 3. **[15 Points]**

Given the following function $g(a, b, c, d) = \sum m(0, 1, 6, 15)$
with the *don't care conditions* = $\sum d(3, 5, 7, 11, 14)$,

a) Find the minimal **Sum-of-Product** expressions of g (**1 Solution**). **(5 points)**

b) Find the minimal **Product-of-Sum** expressions of the function g (**2 Solutions**). **(6 points)**

c) Assuming that all inputs are available as true and complement, implement g using minimum number of the same gate type **(i.e. using only one type of gates) (4 points)**

Question 4. **[10 Points]**

It is required to design a circuit that receives a BCD input *N* and compute two outputs; *Q is the quotient = N/3*, and *R* = remainder. e.g. if the input N=7, then Q=2, and R=1.

Determine the number of bits in N, Q, and R, and draw the truth table for this circuit.

Question 5. **Label all your components, inputs, and outputs** **[20 Points]**

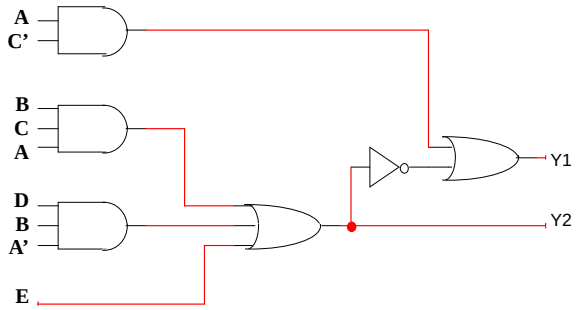
1) Given the function **F (A,B,C) = AB' + AC + A'B'C + A'BC'** ,

Implement F using minimum number of **2-to-1 Multiplexors**. **(10 Points)**

2) Given the functions **G(X,Y,Z) = ∏M (0, 2,3,5,7)** , and **H = ∑m(0,1,4,6,7)**

Implement **G** and **H** using minimum number of **2-to-4 Decoders** and other gates **with minimum sizes (i.e. number of inputs per gate)**. **(10 Points)**

Question 6. For the logic diagram below: **[15 Points]**



1. Complete the Gate level Verilog description of this circuit using the Gate delays in the table below **(6 Points)**

Gate	Delay
NOT	1 ns
2-IP AND	2 ns
2-IP OR	3 ns
3-IP AND	3 ns
3-IP OR	4 ns

2. Describe the same circuit again (**including the delay**) in Verilog, but using the *assign* statement. **(4 Points)**

3. Convert it to NAND-only gates (using minimum number of NAND gates) **(5 Points)**

Question 7. **Label all your components, inputs, and outputs** **[10 Points]**

1. Using any standard functional block (Adders, Multiplexors, decoders …etc.) design a circuit that compute the absolute difference between two unsigned 4-bit numbers A and B (If $A \geq B$, Diff(A,B) = A - B, else Diff(A,B) = B-A). **(4 Points)**

2. Now using your circuit above, and any other components, design an ALU that has two 4-bit data inputs A and B, two control bits C1 and C0, and performs the following functions: **(6 Points)**

(for the bitwise ANDing and ORing of A and B, Y4 should be 0)

C1 C0	Functionality
0 0	**Y ← A & B (Y= bitwise ANDing of A and B)**
0 1	**Y ← A | B (Y= bitwise ORing of A and B)**
1 0	**Y ← A + B (Y= SUM of A and B)**
1 1	**Y ← Diff(A,B) (Y= the absolute difference between A and B, i.e. |A-B|)**