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 161 (Fall 2016)
Major Exam 2
Saturday, Dec. 3rd, 2016

**Time: 120 minutes, Total Pages: 11**

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	14	
2	16	
3	20	
4	6	
5	14	
6	20	
7	20	
Total	**110**	
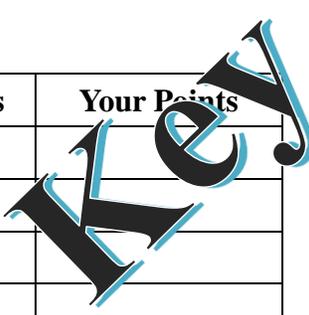
Question 1: <u>Fill in the Spaces</u>: (Show all work needed to obtain your answer)   [14 marks]
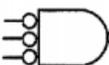
1. The symbol is an equivalent representation of the ___**OR**___ (AND/OR/NOT) gate.
    [1 mark]

2. A two input XOR gate can be used as an inverter of an input, if the other input is set to _____ (**High**/Low).   **[1 mark]**

3. The output for *n* inputs XNOR gate (*n* > 3) will be _____ (**The Same**/Inverted) if any two inputs are inverted.   **[1 mark]**

4. The two-input gate symbol shown on the right performs the function   *f*
    (*A*, *B*) = *A'B*. It _____ (**Can**/Cannot) be used to implement any
    Boolean function similar to Universal gates.   **[1 mark]**

    $A \longrightarrow$ $B \longrightarrow$ $A'B$
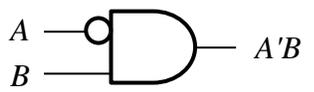
5. Using 6-bit registers, the smallest negative number that can be represented is **-32** and that would be using _____ (Sign-Magnitude / **2's complement**) representation.
    [2 marks]

6. Using 6-bit 2's complement representation, the largest number that can be added to +5 without causing an overflow is **+26** (in decimal).   **[1 mark]**

7. Complete the circuit on the right to implement the function

    F(A, B, C) = **Π**M(0, 1, 4, 6) using a single 3-to-8 decoder
    and a NOR gate. The inputs to the NOR gates will be_ **Q0,
   Q1, Q4, and Q6__**.   **[2 marks]**
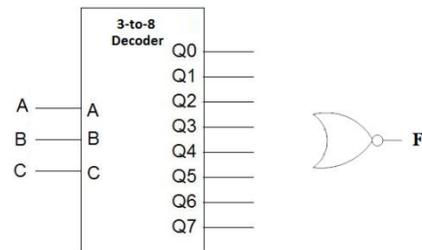
8. For the function F(X, Y, Z) represented in the following K-map, F has ___**3**____ (How many) prime implicants, and of these _____**2**_____ (How many) are essential prime implicants of F.
    [2 marks]
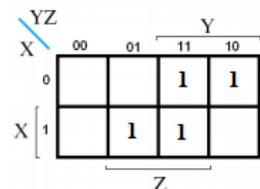
9. The largest decoder we can build using five 2-to-4 decoders with Enable without any additional components is a __**4**___ to __**16**___ decoder.   **[1 mark]**

10. In the priority encoder shown below with **D₀ having highest priority and D₇ the lowest priority**, the output $Y_2Y_1Y_0 = 101$ when the status at inputs D_0-D_7 is as described in _____ (select one of I, II, III, or IV) with all other inputs being in a don't care condition.
   **[2 marks]**

    I. $D_0 = 1$ and $D_2 = 1$.
    II. $D_1 = 1$ and $D_2 = 1$.
    III. $D_3 = 1$ and $D_5 = 1$.
    IV. $D_6 = 1$ and $D_5 = 1$.
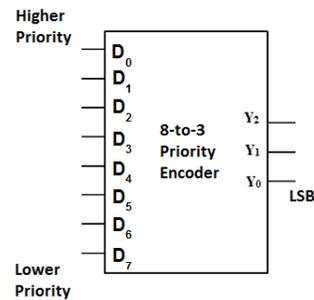
Question 2. Consider the Boolean function $F(W, X, Y, Z) = \Pi M(1, 4, 6, 9)$

1) Identify **all** the *prime implicants* and the *essential prime implicants* of **F**.   **[6 marks]**

Solution:
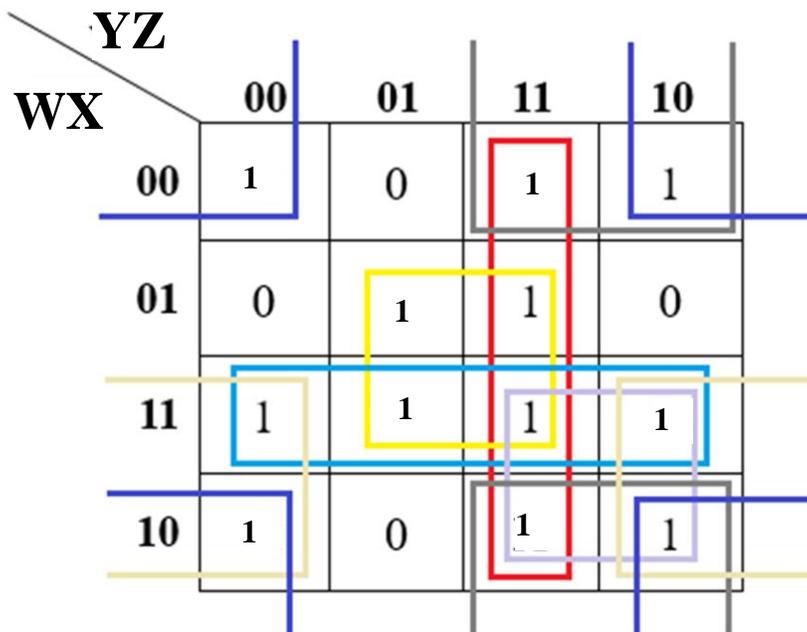
Prime implicants:

$$W\bar{Z}, \bar{X}\bar{Z}, WX, XZ, YZ, \bar{X}Y, WY$$

Essential prime implicants:

2 EPIs: $\bar{X}\,\bar{Z}$ and $X\,Z$.

2) Simplify the Boolean function **F(w,x,y,z)** shown in the K-map below together with its don't care conditions, into *minimal **sum-of-products*** expression.   **[4 marks]**

wx\yz	00	01	11	10
00	1	0	0	1
01	0	0	1	1
11	x	x	x	x
10	1	0	x	x

**Minimal Solution:**

$$F = \bar{X}\bar{Z} + XY$$

3) Implement the following function using minimum number of *NAND* gates. **Assume that all inputs are available in true and complement forms.**   **[2 marks]**

F= A'B + B'C + AC'
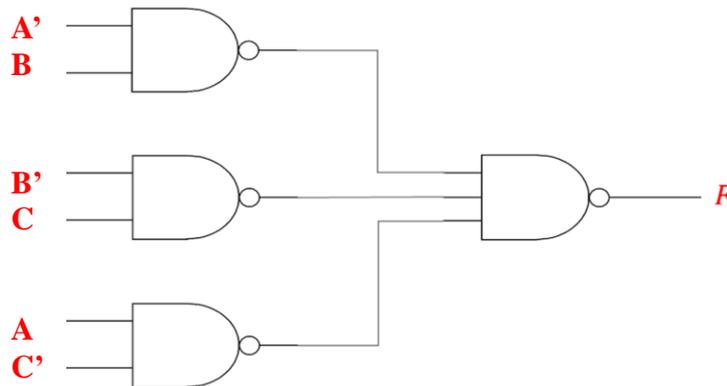
4) Implement the function in part (3) above using minimum number of **NOR** gates only. **Assume that all input are available in true and complement forms.** **[4 marks]**
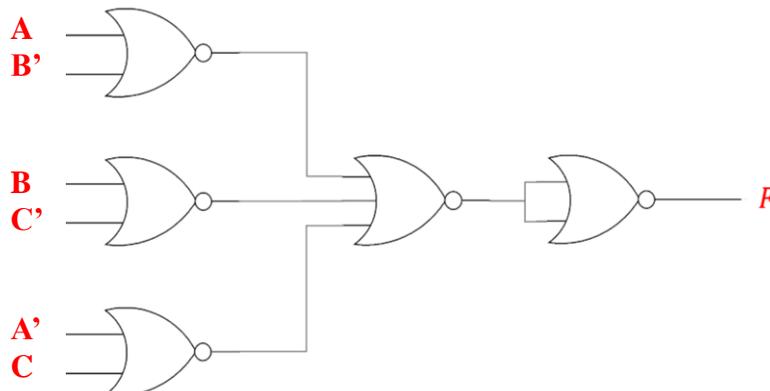
Question 3: [20 marks]

a) Show the **8-bit binary representation** of the following decimal numbers. **[4 marks]**

Decimal Number	**8-bit** Sign-Magnitude representation	**8-bit** 2's complement representation
+35	**0010 0011**	**0010 0011**
-76	**1100 1100**	**1011 0100**

b) Given the following 8-bit binary value, show the equivalent decimal value when the binary number is interpreted as unsigned, as a sign-magnitude number, as a 1's complement, or as a 2's complement signed number. **[4 marks]**

8-bit Binary	Equivalent Decimal Value when the binary number is interpreted as:			
	Unsigned Number	Sign-Magnitude Decimal Number	1's complement Decimal Number	2's complement Decimal Number
1011 1010	**186**	**-58**	**-69**	**-70**

c) Show the addition / subtraction of the following 8-bit signed numbers represented in 2's complement. Indicate whether there is an overflow (Yes / No). **[6 marks]**

```
  1 1      1 1 1 1
    0 1 1 0 1 0 1 1
  + 1 1 0 0 1 1 0 1
  _____
    0 0 1 1 1 0 0 0



Overflow? NO
```

```
    0 1 1 1 0 0 1 0
  - 1 0 0 1 0 1 1 1
  _____
    1 1
    0 1 1 1 0 0 1 0
  + 0 1 1 0 1 0 0 1
  _____
    1 1 0 1 1 0 1 1

Overflow? YES
```

Question 4:   **[6 marks]**

It is required to design a circuit to multiply two 2-bit unsigned binary numbers; A_1A_0 and B_1B_0.

a)   Obtain the truth table for this circuit.   **[4 marks]**

**Solution: To multiply 2 2-bit unsigned numbers, the maximum result is 9, which requires 4 bits →
the O/P is 4-bits -- > The truth Table**

A_1 A_0	B_1 B_0	p_3 p_2 p_1 p_0
0 0	0 0	0 0 0 0
0 0	0 1	0 0 0 0
0 0	1 0	0 0 0 0
0 0	1 1	0 0 0 0
0 1	0 0	0 0 0 0
0 1	0 1	0 0 0 1
0 1	1 0	0 0 1 0
0 1	1 1	0 0 1 1
1 0	0 0	0 0 0 0
1 0	0 1	0 0 1 0
1 0	1 0	0 1 0 0
1 0	1 1	0 1 1 0
1 1	0 0	0 0 0 0
1 1	0 1	0 0 1 1
1 1	1 0	0 1 1 0
1 1	1 1	1 0 0 1

b)   Using the Karnaugh map, find a minimal sum-of-product expression for the output bit with
weight 2 (i.e. 2nd bit from the right).   **[2 marks]**

Karnaugh Map for p1

	B1 B0			
A1 A0	**0 0**	**0 1**	**1 1**	**1 0**
0 0				
0 1			1	1
1 1		1		1
1 0		1	1	

p1 = A1 B1' B0 + A1 A0' B0 + A1' A0 B1 + A0 B1 B0'

Question 5: For this question, **Properly label all components, their inputs and outputs.**

Given the function $F(A, B, C) = A\bar{C} + \bar{B}C$ **[14 marks]**

a) Implement F using the smallest size MUX. Do not use any additional gates. **[6 marks]**
b) Implement F using a single 3-to-8 decoder, and a single OR gate. **[4 marks]**
c) Implement F using two 2-to-4 decoders with enable, one inverter, and one NOR gate. **[4 marks]**

a)
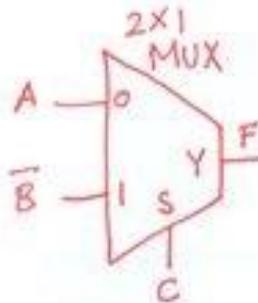
b)
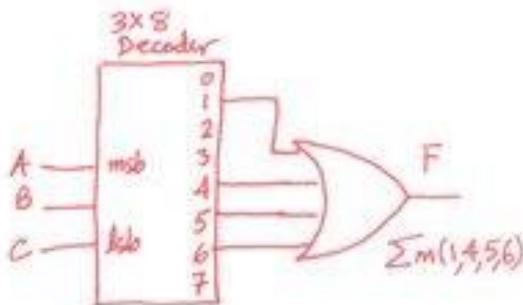
c)
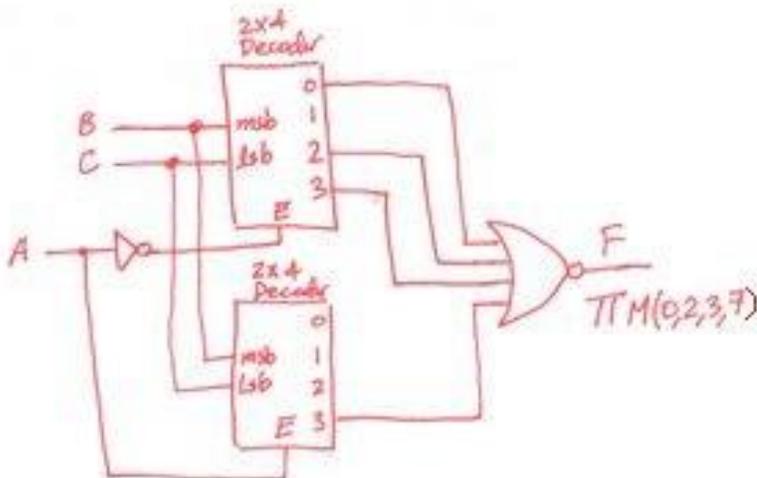
**Question 6:** For this question, **properly label all components, their inputs and outputs.**

Using a **minimal** number of standard components (such as **decoders, encoders, multiplexers, adders, magnitude comparators**) and any other necessary logic gates, design a circuit that:

- Takes three 4-bit unsigned binary numbers $A = A_3A_2A_1A_0$, $B = B_3B_2B_1B_0$, and $C = C_3C_2C_1C_0$.,

- And produces two 4-bit outputs $X = X_3X_2X_1X_0$, and $Y = Y_3Y_2Y_1Y_0$ such that X equals the <u>smallest number</u> among A, B, and C (i.e. X = min(A,B,C)), while Y corresponds to the <u>largest number</u> among A, B, and C (i.e. Y = max(A,B,C)).   **[20 marks]**

One Possible Solution:
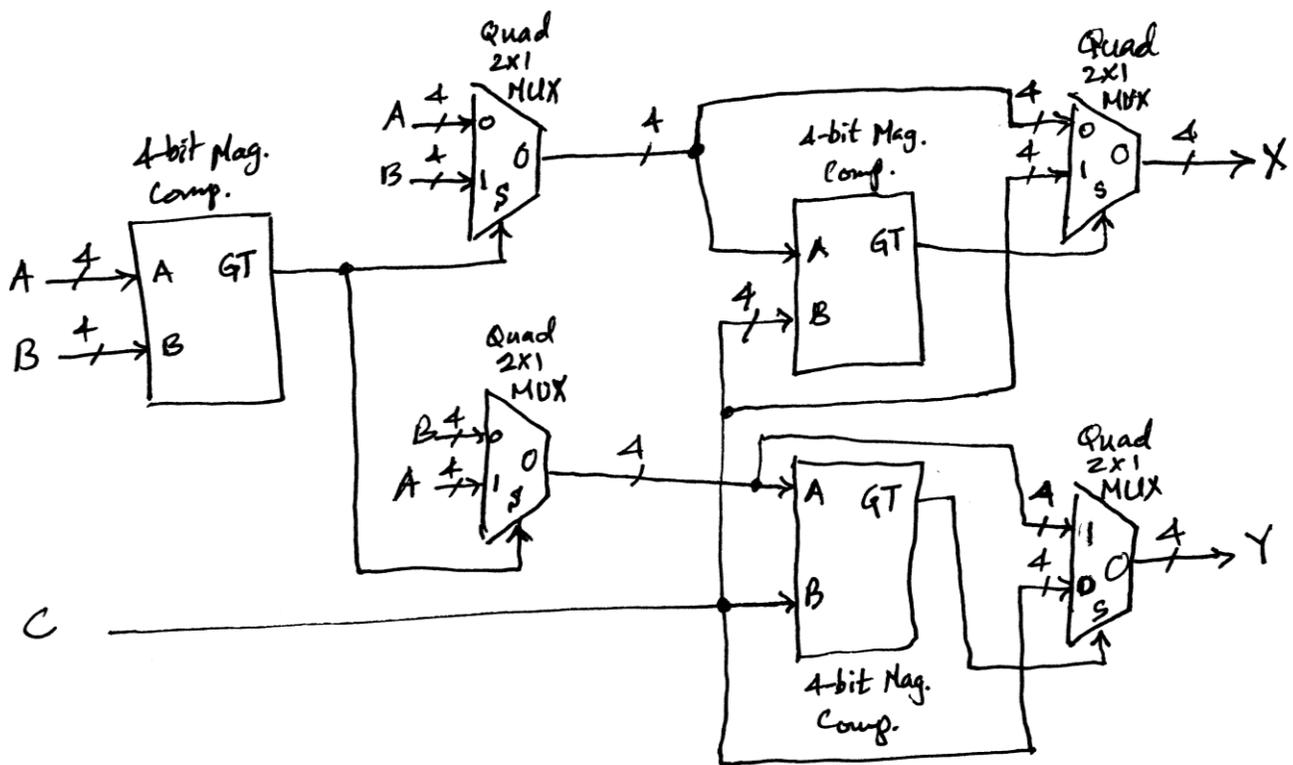
Question 7: Consider the combinational circuit shown below with inputs X, Y, Z, and W, and outputs A and B: **[20 marks]**
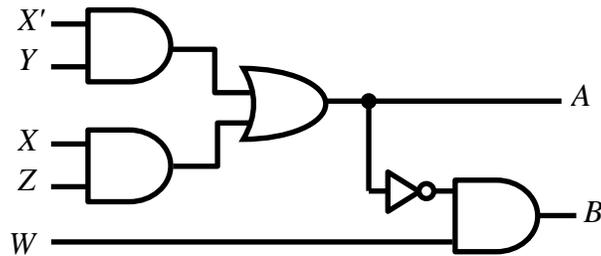
a) Write a Verilog module (call it **CC**) describing this circuit, *as is*, using primitive gates. Assume the following gate delays; inverter's delay= 1 (time unit), AND's delay= 2, OR's delay=3.   **[5 marks]**

```
'timescale 1ns/1ns
module CC (input X,Y,Z,W, output wire A,B);
 wire w1,w2,w3,X_b,A_b ;
 not #1 (X_b,X) ;
 and #2 (w1,X_b,Y);
 and #2 (w2,X,Z);
 or #3 (A,w1,w2);
 not #1 (A_b,A) ;
 and #2 (B,A_b,W);
endmodule
```

b) Write a test bench to test the above circuit for the following input combinations (add 10ns delay between the application of each input set):
   XYZW=**0000**, then **0110**, then **0010**   **[5 marks]**

```
'timescale 1ns/1ns
module CC_tb ();
wire A,B;
reg X,Y,Z,W ;
CC m1 (X,Y,Z,W,A,B) ;
initial begin
X=0;    Y=0; Z=0; W=0;
#10     Z=1; W=1;
#10     Y=1; Z=0; W=0;
end
endmodules
```

c) Write another Verilog module (Call it CC_2) describing the same circuit but using the continuous assignment (i.e. assign statement). **<u>The output delays should also be modeled in this module</u>**.
   [5 marks]

```
'timescale 1ns/1ns
   module CC_2 (input X,Y,Z,W, output wire A,B);
   assign #6 A = (~X&Y)|(X&Z) ; //we set the delays to the worst
   assign #3 B = ~A & W ;       // case delay
   endmodule
```

d) What do we need to change in the test bench of part (b) to simulate the 2nd module (**CC_2**) (b)?
   [2 marks]

```
   Only the instantiation statement → CC_2 m1 (X,Y,Z,W,A,B) ;
```

e) Also, if we modify the test bench to try all possible 16 input combinations, would the two modules produce exactly the same simulation results? Briefly explain why   **[3 marks]**

```
   No. The 1st module models the delay at the gate-level, so different input
   combinations may result in different outputs delay, e.g. when the input
   changes from 0011 to 0010, output B's delay will be 2ns, but when the
   input changes from 0011 to 1011 (or 0111), output B's delay will be 9ns
   (or 8ns). In the 2nd module we model the worst case delay, so outputs A
   and B's delays will always be 6 and 9 ns, respectively.
```