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 172 (Spring 2018)
Major Exam 2
Saturday April 7, 2018

Time: 120 minutes, Total Pages: 12

Name:__KEY________________________________ ID:___________________ Section: _______

Notes:

- Do not open the exam book until instructed
- **No Calculators are 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	10	
2	6	
3	7	
4	7	
5	14	
6	7	
7	8	
8	11	
Total	70	

Question 1. **(10 points)**

Given the following K-map of the function f, where **X** is a don't-care:

		c d			
		0 0	0 1	1 1	1 0
a b	0 0	**1**			**1**
	0 1	**1**	**1**		
	1 1	**1**		**1**	**X**
	1 0	**X**			**1**

a) **(4 points)** Write the terms of **all prime Implicants** and **all essential prime Implicants** of f.

b) **(4 points)** Find **ALL** minimum **sum-of-products** expressions of f.

c) **(2 points)** Find **ALL** minimum **product-of-sums** expressions of f.

a) FIVE Prime Implicants: $b'd'$, $c'd'$, ad', $a'bc'$, abc

Only THREE prime Implicants are essential: $b'd'$, $a'bc'$, abc

b) Two Solutions:

Solution 1: $f = b'd' + a'bc' + abc + c'd'$

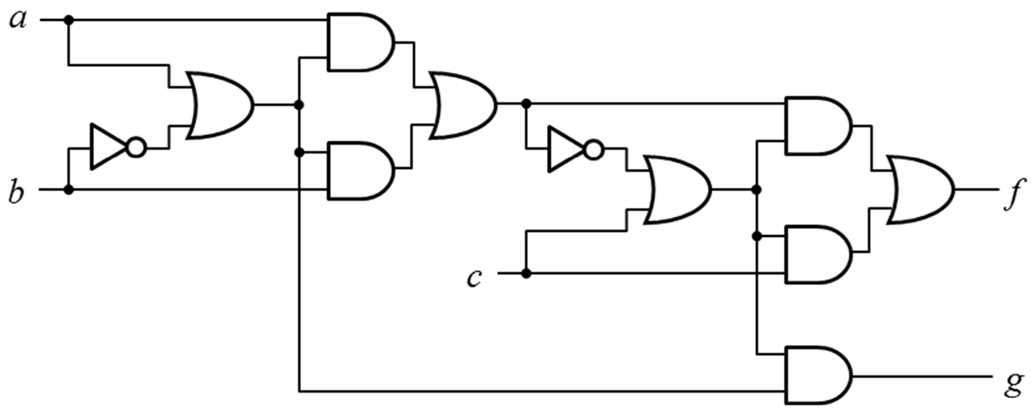
Solution 2: $f = b'd' + a'bc' + abc + ad'$

c) One Solution:

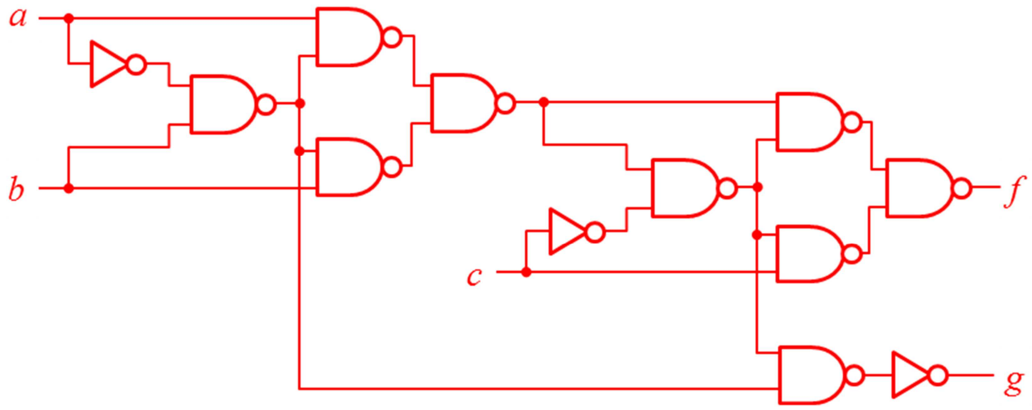
$f = (b + d')\,(a + b' + c')\,(a' + c + d')$

Question 2. **(6 points)**

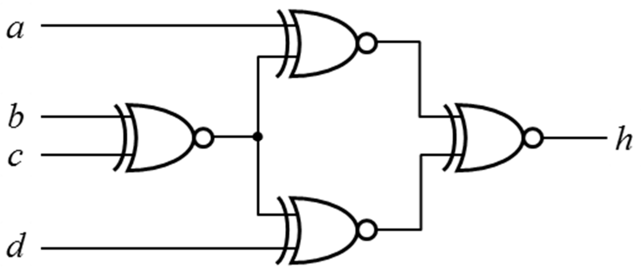
a) **(4 points)** Given the following circuit diagram with AND/OR/NOT gates:



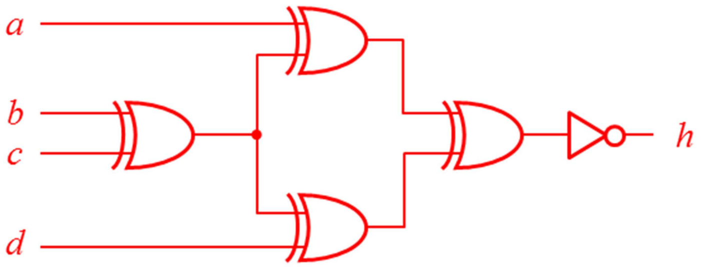
Redraw the above circuit diagram **using only NAND gates** and a minimum number of inverters (NOT gates). You may insert inverters only when necessary.



b) **(2 points)** Given the following circuit diagram with XNOR gates:



Redraw the above circuit **using only XOR gates** and a minimal number of inverters. You may insert inverters only when necessary.

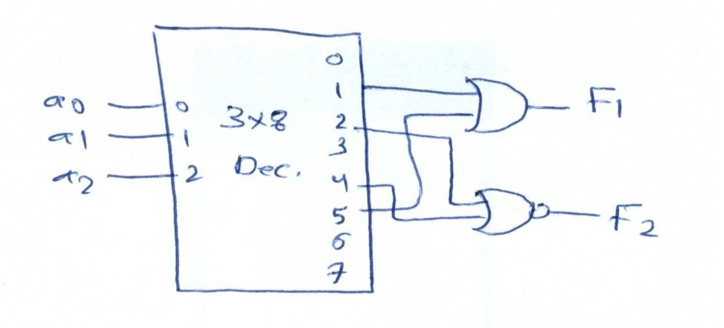


Question 3. **(7 points)**

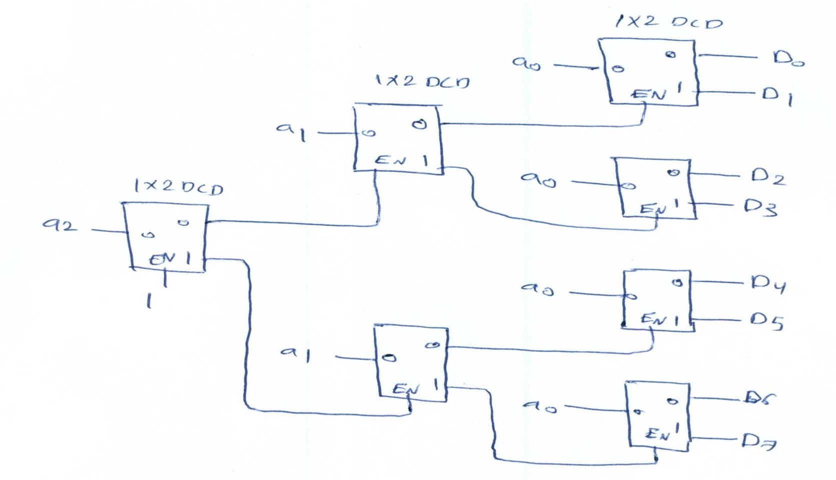
a) **(4 points)** Consider the functions $F1(a_2,a_1,a_0)$ and $F2(a_2,a_1,a_0)$, whose truth table is shown below:

a_2	a_1	a_0	*F1*	*F2*
0	0	0	0	1
0	0	1	1	1
0	1	0	0	0
0	1	1	0	1
1	0	0	0	0
1	0	1	1	1
1	1	0	0	1
1	1	1	0	1

Sketch an implementation of *F1 and F2* using a single 3-8 decoder and minimum additional gates with minimum number of inputs.



b) **(3 points)** Show how you can construct a 3-8 decoder using minimum number of 1-2 decoders with enable inputs. In your implementation, mark which input corresponds to a_0,a_1, and a_2. Also, mark which output corresponds to $D_{0...}D_7$.

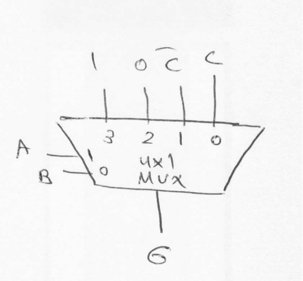


Question 4. **(7 points)**

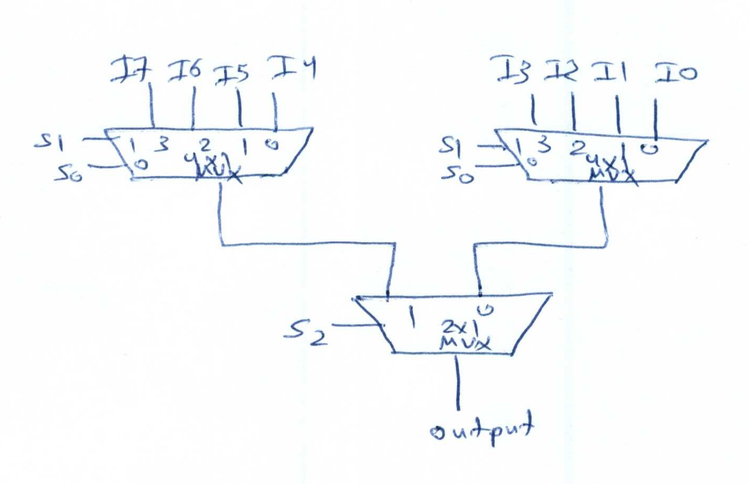
a) **(3 points)** Consider a function *G* with the following truth table:

A	*B*	*C*	*G*
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Sketch an implementation of *G* using a single 4-1 multiplexer.



b) **(4 points)** Show how to construct an 8-1 MUX using only a single 2-1 MUX and minimum number of 4-1 MUXs.



Question 5. **(14 points)**

a) **(6 points)** Represent the given decimal numbers using **4-bit** sign-magnitude, 1's complement and 2's complement representations by filling in the missing information in the following table. If the number is outside the range, please indicate so in the corresponding table cell.

Decimal number	Sign-and-Magnitude (1-bit for sign and 3bit for magnitude)	1's Complement (4-bit)	2's Complement (4-bit)
-3	1011	1100	1101
-8	Outside the range	Outside the range	1000

b) **(3 points)** Given the following **6-bit** binary number, determine its **decimal value** based on the given representations by filling in the missing information in the following table.

6-bit Binary Number	Equivalent decimal value with the binary number interpreted as:		
	Sign-and-Magnitude	1's Complement	2's Complement
101010	-10	-21	-22

c) **(1 point)** Given the 4-bit binary number 1001 in 2's complement representation, the 8-bit binary number in 2's complement representation with equivalent decimal value is 11111001.

d) **(4 points)** Perform the following arithmetic operations using 2's complement representation and indicate if there is an overflow:

```
       01011111                 01001111
+      01100010         -       11011010
      __________               __________
       11000001
                                01001111
                        +       00100110
                               __________
                                01110101
```

Overflow? (Yes/No) Yes

Overflow? (Yes/No) No

Question 6: **(7 marks)**

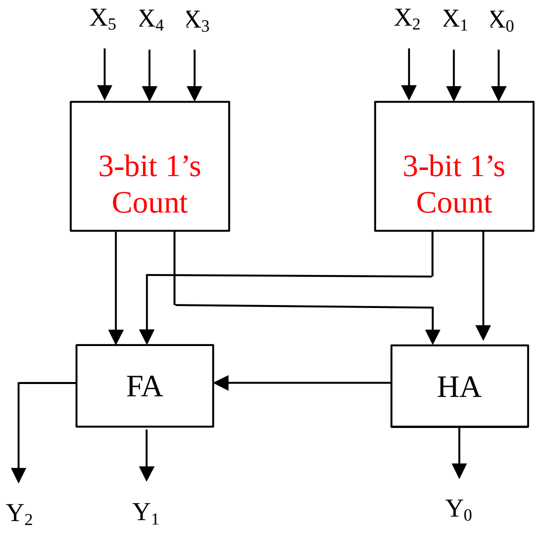
a) **(3 mark)** It is required to design a combinational circuit that receives a 3-bit input X ($X_2X_1X_0$) and produces an output Y that represents the number of 1's in the 3-bit input X. Show the truth table of this circuit.

b) **(4 marks)** It is required to design a combinational circuit that receives a 6-bit input X ($X_5X_4X_3X_2X_1X_0$) and produces an output Y that represents the number of 1's in X. Design the required circuit using as many as needed from the 3-bit 1'count circuit and minimum number of needed half and full adders.

a)

X_2	X_1	X_0	Y_1	Y_0
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

b)

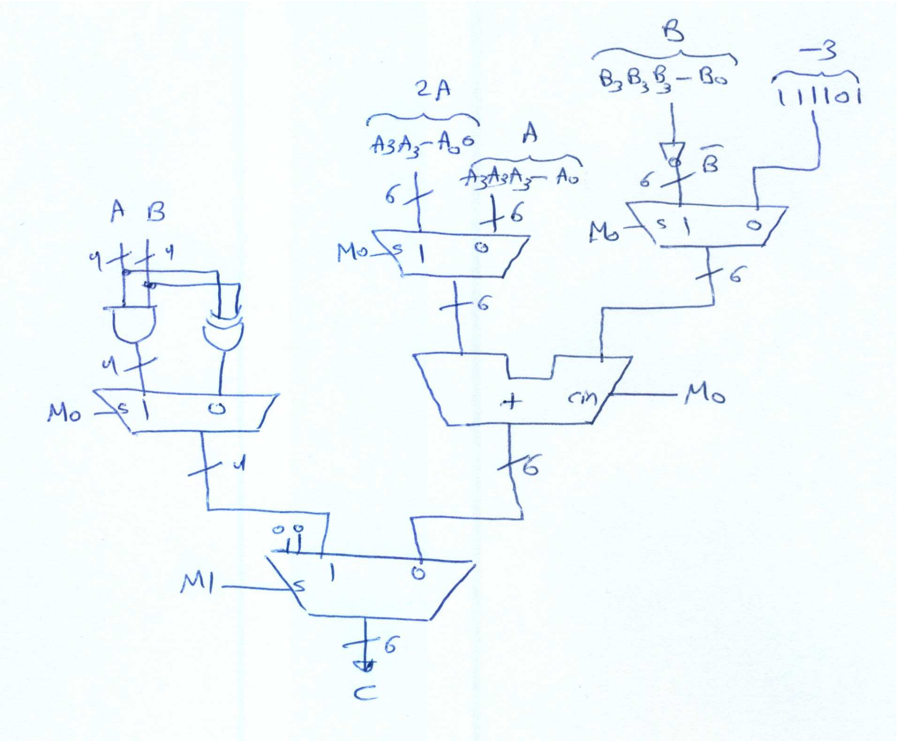


Question 7: **(8 marks)**

It is required to design an arithmetic and logic circuit with two **__4-bit__** **__signed__** 2's complement inputs A ($A_3A_2A_1A_0$) and B ($B_3B_2B_1B_0$), and two control inputs M1 and M0 that produces a **__6-bit__ output C** ($C_5C_4C_3C_2C_1C_0$) according to the following table:

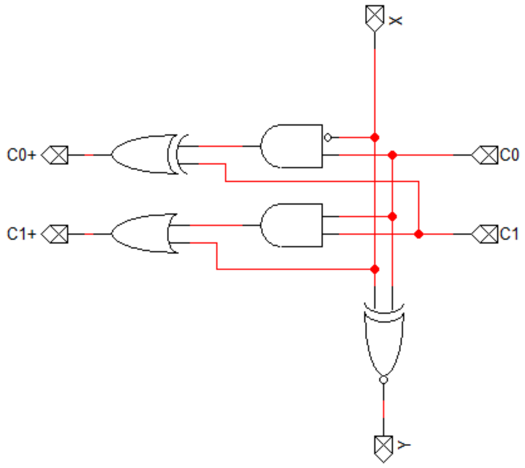
M1	*M0*	*F(A, B)*
0	0	C ← A – 3
0	1	C ← 2A – B
1	0	C ← A XOR B
1	1	C ← A AND B

Design this circuit with minimum needed number of standard components (MUX, Decoders, Adders, Comparators, Logic Gates …etc.).
<u>Properly label all components, their inputs, outputs and sizes.</u>



Question 8: **(11 marks)**

Consider the combinational circuit given below which has three inputs C1, C0, and X, three outputs C1+, C0+ and Y:



a) **(3 marks)** Write a Verilog module to model the combinational circuit using assign statements.

b) **(4 marks)** Write a Verilog module that instantiates three copies of this circuit and connects C1 and C0 of the first instance to 00, and connects the carry outs of each cell to the carry ins of the consecutive cell.

c) **(4 marks)** Write a test bench that tests the 3-bit circuit modeled in (ii) that applies the following input patterns to your circuit ($X_2X_1X_0$)={010, 100, 101} with a delay of 30 time units between each input combination.

a)

```
module Cell3XM1 (output C1P, C0P, Y, input C1, C0, X);

    assign Y = X ~^  C0;

    assign C0P = C1 ^ (~X &  C0);

    assign C1P = X | (C1 &  C0);

endmodule
```

b)

```
module D3XM1 (output Cout1, Cout0, output [2:0] Y, input [2:0] X);

wire  [1:0] C1P, C0P;

Cell3XM1 M1 (C1P[0], C0P[0], Y[0], 0, 0, X[0]);
```

```
Cell3XM1 M2 (C1P[1], C0P[1], Y[1], C1P[0], C0P[0], X[1]);
Cell3XM1 M3 (Cout1, Cout0, Y[2], C1P[1], C0P[1], X[2]);

endmodule
```

c)

```
module D3XM1_Test();

reg [2:0] X;
wire Cout1, Cout0;
wire [2:0] Y;

D3XM1 M1 (Cout1, Cout0, Y, X);

initial begin
X=3'b010;
#30 X=3'b0100;
#30 X=3'b0101;
End

endmodule
```