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	

Key

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**)

```
A = (1010 0011)unsigned = 128+32+2+1 = 163

B = (1000 0010)unsigned = 128+2 = 130
```

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

```
A = 1010 0011 = -(32+2+1) = -35
B = 1000 0010 = -2
```

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

```
A = 1010 0011 = -128+32+2+1 = -93
B = 1000 0010 = -128+2 = -126
```

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

```
      1      1

   A = 1010 0011

 + B = 1000 0010
 ---------------
       0010 0101
```

Overflow? **Yes**

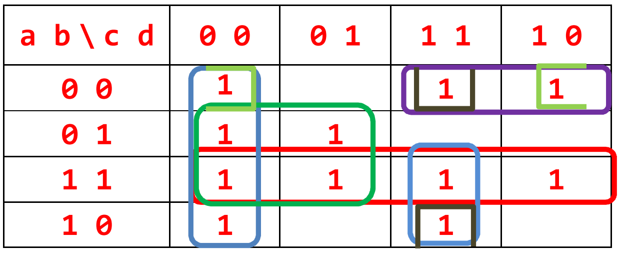
e) If A and B are signed numbers in 2's complement, **Compute** A - B and **specify if there is an overflow** **(5 pts)**

```
d) A - B = A + (2's complement of B)

                 Carry    1 1111 11

   A                     = 1010 0011

 + (2's complement of B) = 0111 1110
                          -----------
                           0010 0001
```

Overflow? **No**

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)**



a b\c d	0 0	0 1	1 1	1 0
0 0	1		1	1
0 1	1	1		
1 1	1	1	1	1
1 0	1		1	

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

There are Seven Prime Implicants: ***a b, c' d', b c' , a' b' c, a' b'd' , a c d, b' c d***

Only three prime implicants are Essential: ***a b, c' d', b c'***

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

$f = a\,b + c'\,d' + b\,c' + \underline{a'\,b'\,c + a\,c\,d}$ **(Solution 1)**
$f = a\,b + c'\,d' + b\,c' + \underline{a'\,b'\,c + b'\,c\,d}$ **(Solution 2)**
$f = a\,b + c'\,d' + b\,c' + \underline{a'\,b'\,d' + b'\,c\,d}$ **(Solution 3)**

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)**

a b\c d	0 0	0 1	1 1	1 0
0 0	1	1	x	
0 1		x	x	1
1 1			1	x
1 0			x	

Minimal SOP expression: ***g = a' b' c' + b c (minimum terms → minimum # of NAND gates)***

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

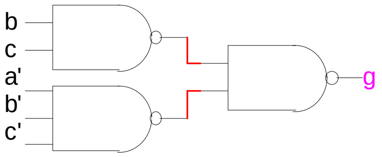
a b\c d	0 0	0 1	1 1	1 0
0 0			x	1
0 1	1	x	x	
1 1	1	1		x
1 0	1	1	x	1

Minimal POS expressions of *g*:

g' = b c' + b'c + a b' **→** ***g = (b' + c) (b + c') (a' + b)*** **(Solution 1)**

g' = b c' + b'c + a c' **→** ***g = (b' + c) (b + c') (a' + c)*** **(Solution 2)**

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.

Solution: N is 4-bits, Q max value is 3 → 2 bits, R max value is also 2 → 2 bits

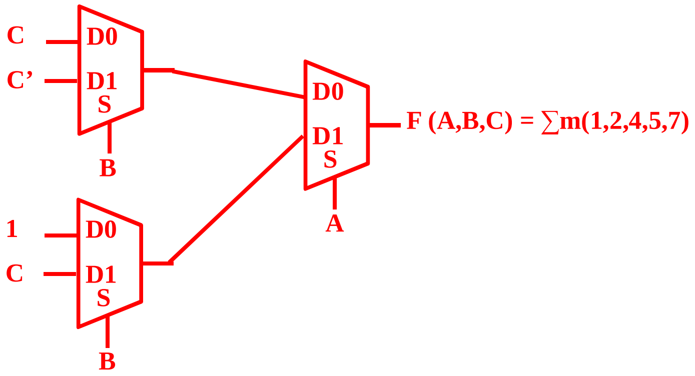
Truth Table

Input N N_3 N_2 N_1 N_0	Output Q Q_1 Q_0	Output R R_1 R_0
0 0 0 0	0 0	0 0
0 0 0 1	0 0	0 1
0 0 1 0	0 0	1 0
0 0 1 1	0 1	0 0
0 1 0 0	0 1	0 1
0 1 0 1	0 1	1 0
0 1 1 0	1 0	0 0
0 1 1 1	1 0	0 1
1 0 0 0	1 0	1 0
1 0 0 1	1 1	0 0
1 0 1 0	x x	x x
1 0 1 1	x x	x x
1 1 0 0	x x	x x
1 1 0 1	x x	x x
1 1 1 0	x x	x x
1 1 1 1	x x	x x

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’** ,

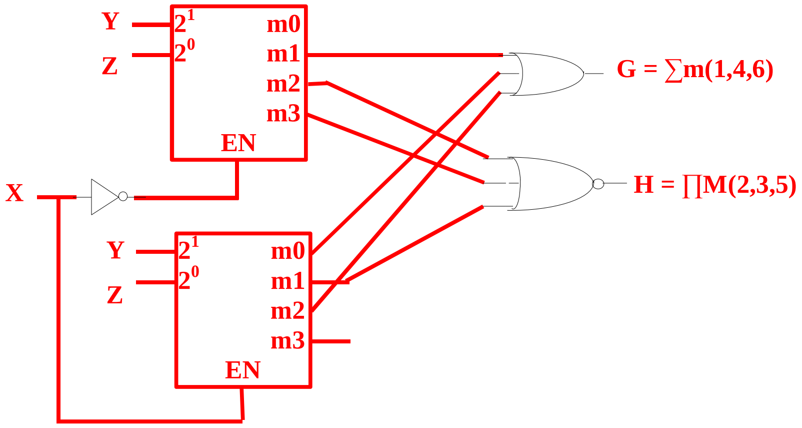
Implement F using minimum number of **2-to-1 Multiplexors**. **(10 Points)**



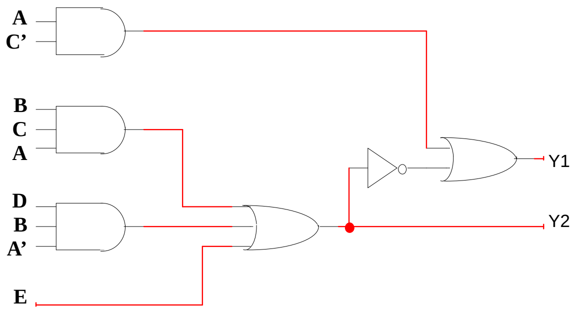
2) Given the functions **G(X,Y,Z) = ∏M (0, 2,3,5,7)** , and **H = ∑m(0,1,4,6,7)**

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)**

We need to construct a 3-to-8 decoder from two 2-to-4 decoders and an inverter.
Since G has only 3 minterms (1,4,6) we will use a decoder + OR design,
Since H has only 3 Maxterms (2,3,5) we will use a decoder + NOR design:



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)**

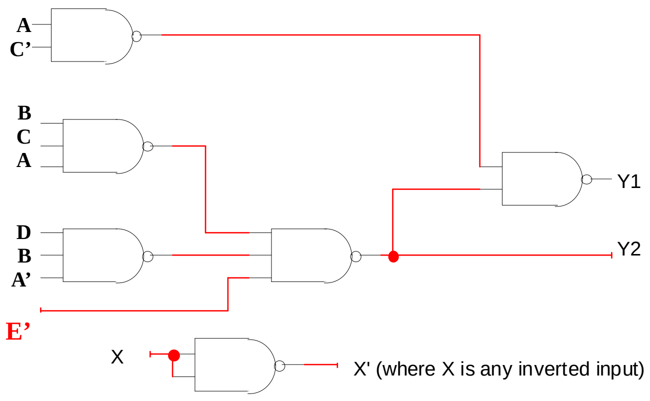
```
module Y1_Y2 (input A,B,C,D,E, output Y1,Y2);
wire n1,n2,n3,A_b,C_b,Y2_b ;
not #1 (A_b,A) ; not #1 (C_b,C); and #2 (n1,A_b,C);
and #3 (n2,B,C,A); and #3 (n3,D,B,A_b) ;
not # 1 (Y2_b,Y2) ; or #3 (Y1,n1,Y2_b); or #4 (Y2,n2,n3,E) ;
endmodule
```

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 (**<u>including the delay</u>**) in Verilog, but using the *assign* statement. **(4 Points)**

```
module Y1_Y2 (input A,B,C,D,E, output Y1,Y2);
assign #8 Y2 = E | ~A & B & D | A & B & C ;
assign #12 Y1 = ~( E | ~A & B & D | A & B & C) |  A & ~C ; // if Y1 is defined as a
endmodule                                  // function of Y2, the delay would be wrong!
```

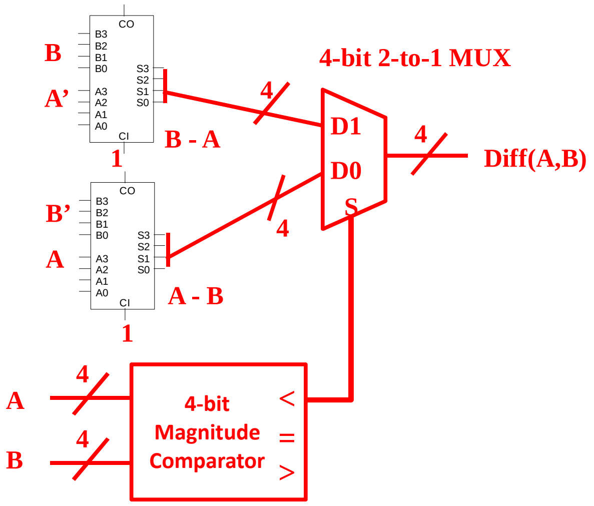
3. Convert it to NAND-only gates (using minimum number of NAND gates) **(5 Points)**



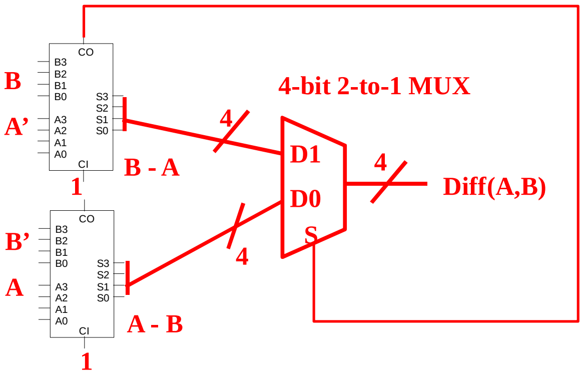
Question 7. **Label all your components, inputs, and outputs** **[10 Points]**

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 ≥ B, Diff(A,B) = A - B, else Diff(A,B) = B-A). **(4 Points)**

SOLUTION 1



SOLUTION 2

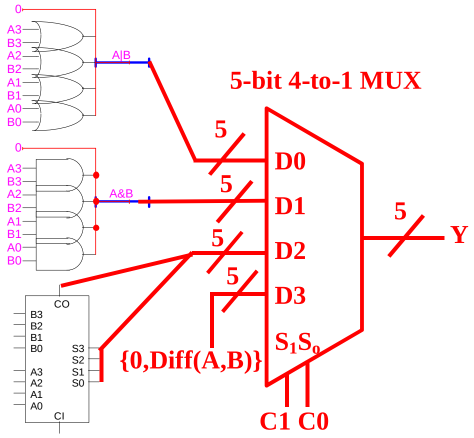


The CO (output carry) of (B – A) is 1 when (B – A ≥ 0) and 0 otherwise. It is used as the select input of the mux. It works only when A and B are unsigned.

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|)


0
A3
B3
A2
B2
A1
B1
A0
B0
A|B
0
A3
B3
A2
B2
A1
B1
A0
B0
A&B
5-bit 4-to-1 MUX
5
D0
5
D1
5
D2
5
D3
5
Y
CO
B3
B2
B1
B0
A3
A2
A1
A0
S3
S2
S1
S0
CI
{0,Diff(A,B)}
S1S0
C1 C0