

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 191 (Fall 2019)
Major Exam 2
Saturday Nov. 16, 2019

Time: 120 minutes, Total Pages: 12

Name: KEY ID: _____ Section: _____

<input type="checkbox"/> Dr. Aiman El-Maleh	<input type="checkbox"/> Dr. Muhamed Mudawar
<input type="checkbox"/> Dr. Ali Al-Suwaiyan	<input type="checkbox"/> Dr. AbdulAziz Tabakh

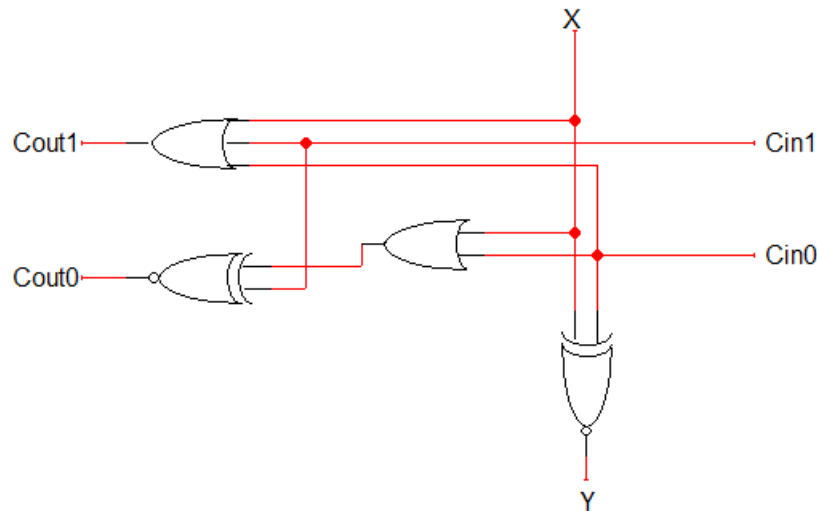
Notes:

- 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
Q1	12	
Q2	10	
Q3	19	
Q4	14	
Q5	15	
Q6	5	
Total	75	

Question 1:**(12 marks)**

Consider the combinational circuit given below modeling a 1-bit cell for computing the equation $Y=X-3$, which has three inputs X , $Cin1$ and $Cin0$ and three outputs Y , $Cout1$, and $Cout0$:



- a) (4 points) Write a Verilog module to model the combinational circuit by modeling $Cout0$ using assign statement while modeling Y and $Cout1$ using primitive gates. Add gate delay to your model assuming that the delay of a 2-input XNOR gates is 3 units, the delay of a 2-input OR is 2 units and the delay of a 3-input OR gate is 3 units.
- b) (4 points) Write a Verilog module for modeling a 3-bit circuit for computing the equation $Y=X-3$ by instantiating three copies of this cell and making necessary connections. Note that the carry in signals for the first cell should be all connected to 0. Use the following module interface:

```
module DXM3 (output Cout1, Cout0, output [2:0] Y, input [2:0] X);
```

- c) (4 points) Write a test bench that tests the 3-bit circuit modeled in (b) that applies the following input patterns to your circuit $X=\{3'b011, 3'b100, 3'b111\}$ with a delay of 30 time units between each input combination.

a)

```
module OneCellXM3 (output Cout1, Cout0, Y, input Cin1, Cin0, X);
```

```
XNOR #3 (Y, X, Cin0);
```

```
OR #3 (Cout1, Cin1, Cin0, X);
```

```
assign #5 Cout0 = Cin1 ~^ (Cin0 | X);
```

```
endmodule
```

b)

```
module DXM3 (output Cout1, Cout0, output [2:0] Y, input [2:0] X);  
  
    wire [1:0] C1, C0;  
  
    OneCellXM3 M1 (C1[0], C0[0], Y[0], 0, 0, X[0]);  
    OneCellXM3 M2 (C1[1], C0[1], Y[1], C1[0], C0[0], X[1]);  
    OneCellXM3 M3 (Cout1, Cout0, Y[2], C1[1], C0[1], X[2]);  
  
endmodule
```

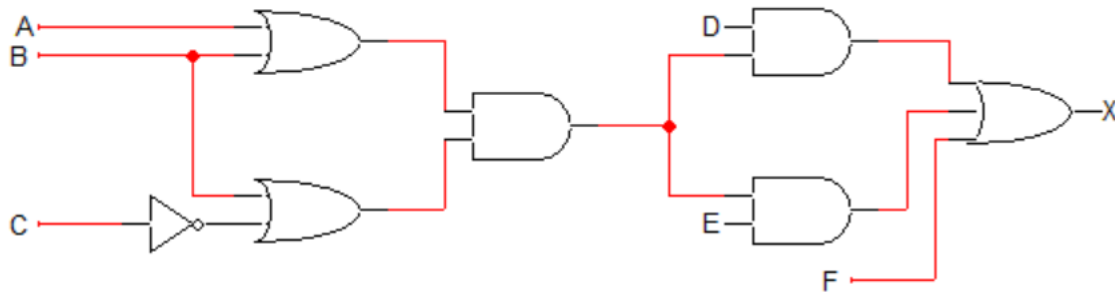
c)

```
module DXM3_Test();  
  
    reg [2:0] X;  
    wire Cout1, Cout0;  
    wire [2:0] Y;  
  
    DXM3 M1 (Cout1, Cout0, Y, X);  
  
    initial begin  
        X=3'b011;  
        #30 X=3'b100;  
        #30 X=3'b111;  
    end  
  
endmodule
```

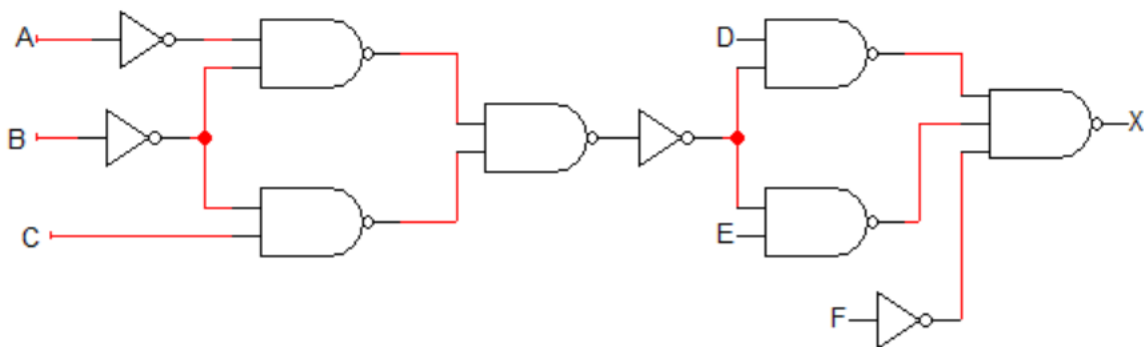
Question 2.

(10 points)

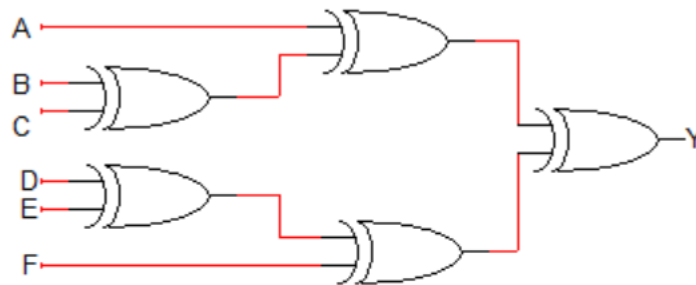
a) (4 points) Given the following circuit diagram with AND/OR/NOT gates:



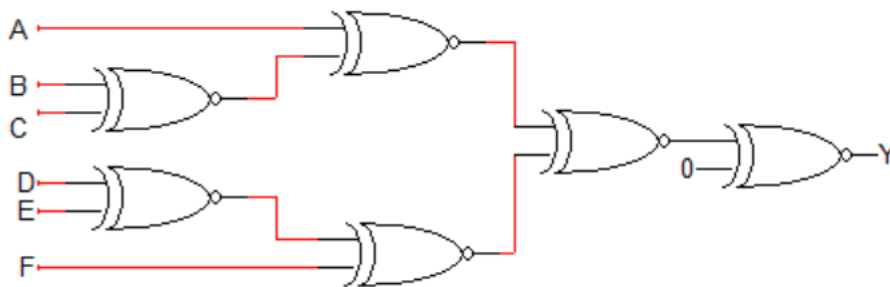
Redraw the above circuit diagram **using only NAND gates** and a minimum number of inverters (NOT gates). You may insert inverters only when necessary.



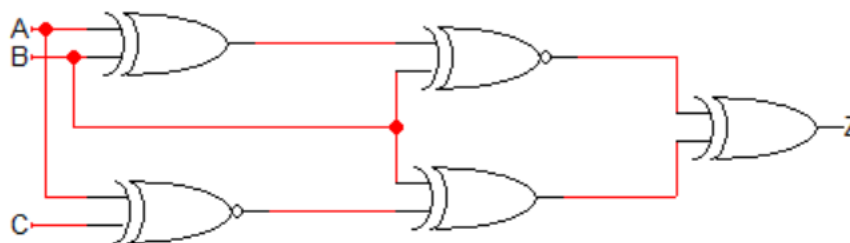
b) (3 points) Given the following circuit diagram with 2-input XOR gates:



Redraw the above circuit using **only 2-input XNOR gates**. Minimize the number of 2-input XNOR gates used.



c) (3 points) Reimplement the circuit given below using minimum number of 2-input XOR gates:



$$\begin{aligned}
 Z &= [(A \oplus B) \oplus B]' \oplus [(A \oplus C)' \oplus B] = [(A \oplus B) \oplus B]' \oplus [(A \oplus C) \oplus B]' \\
 &= [(A \oplus B) \oplus B] \oplus [(A \oplus C) \oplus B] \\
 &= [A] \oplus [(A \oplus C) \oplus B] = B \oplus C
 \end{aligned}$$



Question 3

(19 points)

- I) (5 points) Show the **8-bit binary representation** of the following decimal numbers using the specified representations in the table below:

Decimal Number	8-bit Sign-Magnitude representation	8-bit 1's complement representation	8-bit 2's complement representation
+21	0001 0101	0001 0101	0001 0101
-74	1100 1010	1011 0101	1011 0110

- II) (5 points) Given that **A** and **B** are **8-bit signed integers** that use the **2's complement representation**:

$$A = 11100110 \quad B = 01010101$$

- a) (2 points) What are the decimal values of **A** and **B**?

$$A = 2 + 4 + 32 + 64 - 128 = -26$$

$$B = 1 + 4 + 16 + 64 = +85$$

- b) (3 points) Compute **A – B** in binary by converting subtraction into addition to the 2's complement of **B**. Indicate whether there is overflow or not.

$$\begin{array}{r}
 1\ 1\ 1\ \ 1\ 1\ 1 \\
 1\ 1\ 1\ 0\ 0\ 1\ 1\ 0 \\
 +\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 1 \\
 \hline
 1\ 0\ 0\ 1\ 0\ 0\ 0\ 1
 \end{array}
 \quad
 \begin{array}{l}
 \text{Carry bits} \\
 \text{2's complement}
 \end{array}$$

= -26
= -85
= -111

There is **NO** overflow because the carries **C7** and **C8** are identical. Final carry bit **C8** is ignored.

III) (9 points) It is required to design a circuit that computes: $S = 2 * X - Y + 1$. The inputs $X[3:0]$ and $Y[3:0]$ are 4-bit signed integers that use the 2's complement representation.

a) (2 points) What is the minimum negative value of X? What is the maximum positive value?

Minimum negative value of X = -8

Maximum positive value of X = +7

b) (3 points) What is the minimum negative value of S? What is the maximum positive value? How many bits are needed to represent S without overflow?

Minimum negative value of S = $2 * (-8) - (+7) + 1 = -22$

Maximum (positive) value of S = $2 * (+7) - (-8) + 1 = +23$

S must have 6 bits to represent the signed range -22 to +23

S cannot be 5 bits because the 5-bit signed range is -16 to +15 only.

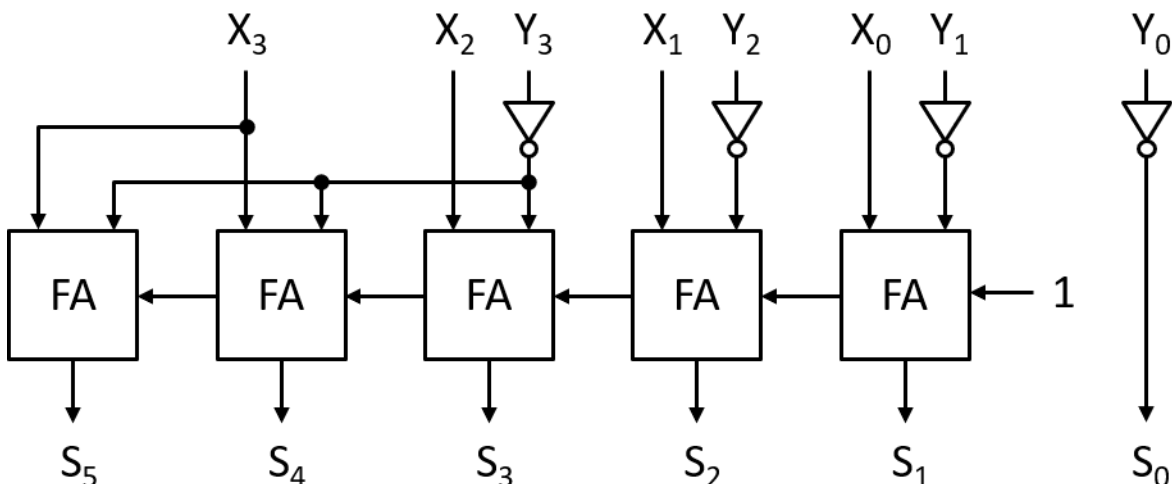
c) (4 points) Using Full-Adders and inverters, draw a circuit that computes: $S = 2 * X - Y + 1$. Hint: $2 * X$ is computed by left-shifting X.

$S = 2 * X - Y + 1 = X \ll 1 + (2\text{'s complement of } Y) + 1 = X \ll 1 + (1\text{' complement of } Y) + 2$

$S = X \ll 1 + Y' + 2$ (2 is 'b10 in binary)

Signed integers are sign-extended, not zero-extended.

Most significant sign bits (X3 and Y3) are extended to produce a 6-bit Sum = S[5:0]



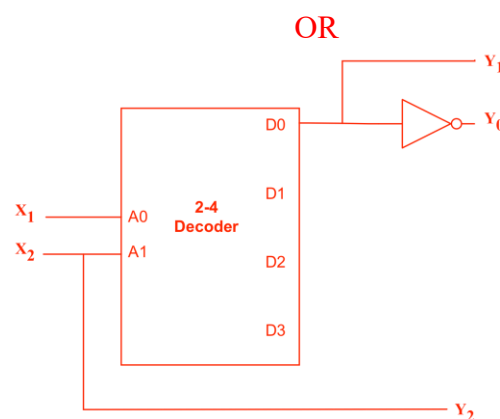
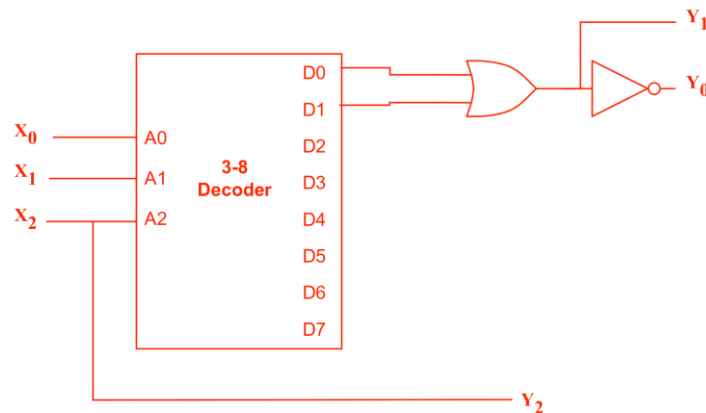
Question 4.**(14 points)**

In this question, clearly label all components, their inputs, and outputs.

- a) (4 points) Consider the following truth table, in which X_2 , X_1 , and X_0 are the inputs and Y_2 , Y_1 , and Y_0 are the outputs.

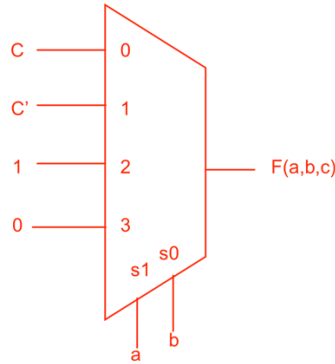
X_2	X_1	X_0	Y_2	Y_1	Y_0
0	0	0	0	1	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	0	0	1
1	0	0	1	0	1
1	0	1	1	0	1
1	1	0	1	0	1
1	1	1	1	0	1

Using a minimum-size decoder and a minimum number of additional gates, show how to implement Y_2 , Y_1 , and Y_0 . Your additional logic gates must use the smallest possible number of inputs.

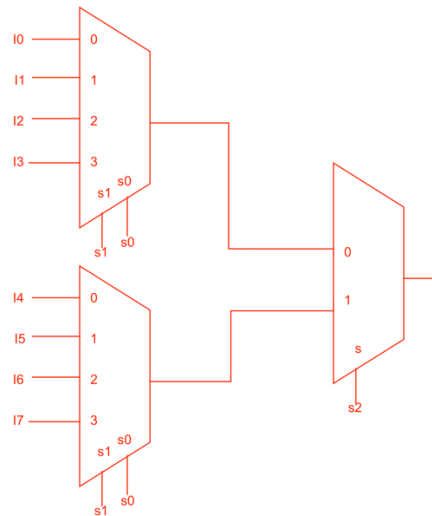


Both solutions are accepted as correct answers

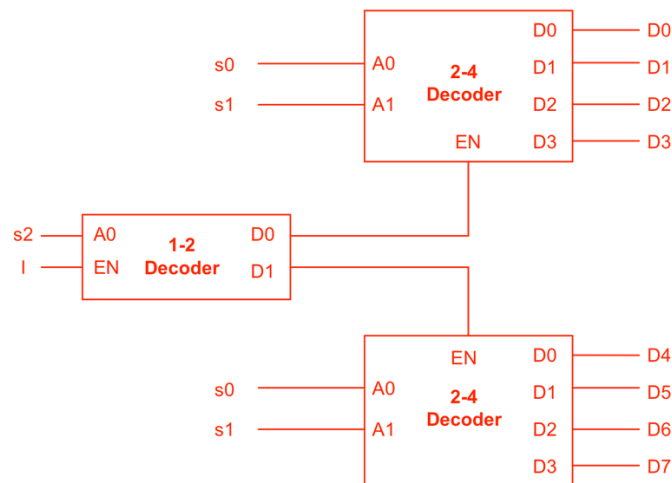
b) (3 points) Consider the function $F(a, b, c) = \sum m(1,2,4,5)$. Assuming the availability of true and complemented values of a, b, and c, show an implementation of output F using a single 4-1 MUX.



c) (3 points) Show how to implement an 8-1 MUX using two 4-1 MUXes and a 2-1 MUX.



d) (4 points) Show how to implement a 1-8 demultiplexer using the minimum number of 2-4 and 1-2 decoders with enable inputs and nothing else.



Question 5.**(15 points)**

l) (7 points) You are required to design a combinational circuit that computes the equation $Z = 3 - |X|$ Where $X[3:0]$ and $Z[3:0]$ are 4-bit signed binary number in 2's complement representation.

a) (4 points) Fill in the following table with the values of Z for each input value of X.

$X_3X_2X_1X_0$	$Z_3Z_2Z_1Z_0$
0000	0011
0001	0010
0010	0001
0011	0000
0100	1111
0101	1110
0110	1101
0111	1100
1000	1011
1001	1100
1010	1101
1011	1110
1100	1111
1101	0000
1110	0001
1111	0010

b) (3 points) Use K-map to find the minimal SOP of the most significant bit of Z (z_3).

		X_1X_0			
		00	01	11	10
X_3X_2	00	0	0	0	0
	01	1	1	1	1
	11	1	0	0	0
	10	1	1	1	1

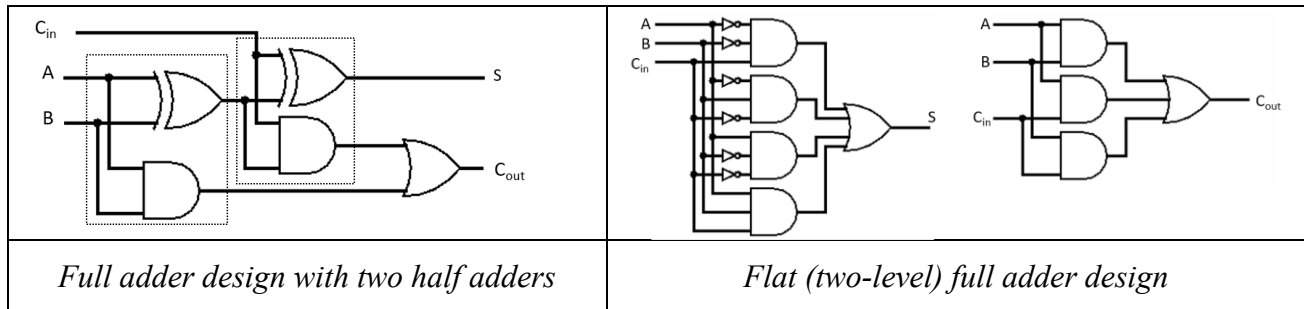
$$Z_3 = X_3'X_2 + X_3X_2' + X_3X_1'X_0'$$

$$Z_3 = (X_3 \oplus X_2) + X_3X_1'X_0'$$

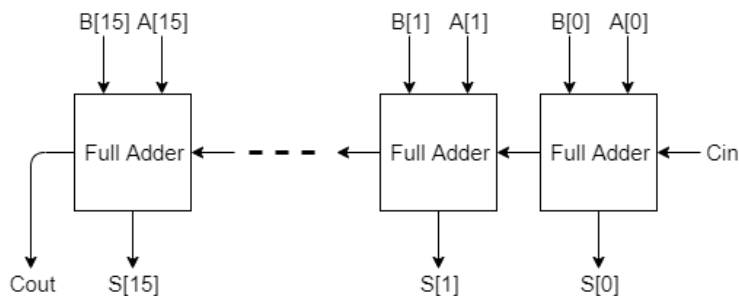
$$\text{OR } Z_3 = X_3'X_2 + X_3X_2' + X_2X_1'X_0'$$

$$\text{OR } Z_3 = (X_3 \oplus X_2) + X_3X_1'X_0'$$

II) (8 points) Assume two different full adder implementations: one with two half adders and the other is flat (Two-Level) as shown below.



A 16-bit ripple carry adder is to be built using full adders as shown below. Based on gate delays given below, calculate the propagation delay of the output carry (Cout) and the most significant bit of the sum (S₁₅) for both designs of full adders.



16-bit Ripple Carry Adder

Gate	Delay
2-input XOR	3ns
Inverter	1ns
2-input AND/OR	2 ns
3-input AND/OR	3ns
4-input OR	4ns

Implementation	Delay of C _{out}	Delay of S ₁₅
16-bit RCA with full adder implemented with two half adders	$d(\text{XOR}) + 16 \times (d(2\text{-AND} + 2\text{-OR}) = 3 + 16 \times (2+2) = 67 \text{ ns}$	$2 \times d(\text{XOR}) + 15 \times (d(2\text{-AND} + 2\text{-OR}) = 6 + 60 = 66 \text{ ns}$
16-bit RCA with full adder implemented using flat (two-level) full adder	$16 \times d(\text{Z} \rightarrow \text{C}) = 16 \times 5 = 80 \text{ ns}$	$15 \times d(\text{Z} \rightarrow \text{C}) + d(\text{Z} \rightarrow \text{S}) = 15 \times 5 + 8 = 83 \text{ ns}$

For design 2:
 Delay from input Cin to Cout = $d(\text{Cin} \rightarrow \text{Cout}) = 2+3 = 5\text{ns}$
 Delay from input Cin to S = $d(\text{Cin} \rightarrow \text{S}) = 1 + 3 + 4 = 8\text{ns}$

Question 6:**(5 points)**

Design a combinational circuit that receives three 4-bit **unsigned** numbers (A, B, C) and calculates the absolute value of the difference between the largest and the smallest (i.e., $D = \text{MAX}(A,B,C) - \text{MIN}(A,B,C)$). You may use only inverters and MSI components (adders, comparators, multiplexers, decoders, encoders and demultiplexers) of minimum size. Please note that you are not allowed to use subtractors as components. If you need a subtractor, show its design using an adder.

Mark your inputs and outputs of each component clearly.

(For example: $A = 7, B = 3, C = 5 \rightarrow D = A - B = 4$).

