INTERNATIONAL UNIVERSITY OF SARAJEVO

FALL 2017

CS 303 DIGITAL DESIGN
MID-TERM EXAM

STUDENT NAME: _____

DATE: _____

Instructions:

• Examination time: **75 min**.

• Print your **name** and **student ID number** in the space provided above.

• This examination is **closed book** and **closed notes**.

• There are 4 questions. The points for each question are given next to the question title. The overall maximum score is 100. **This mid-term weighs 20% of your final grade.**

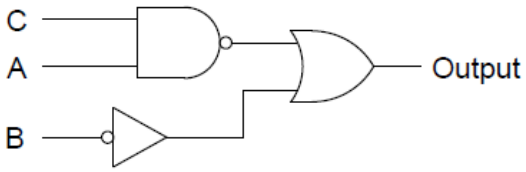
• Answer each question in the space provided. If you need to continue an answer onto the back of the sheet, clearly indicate that and label the continuation with the question number. Clearly indicate your final answers!

QUESTION	1	2	3	4
POINTS	/15	/25	/15	/45
			TOTAL	/100

1. Mark the correct statement(s) (15 %)

1.1 The output of the circuit to the right is logically equal to: **[5]**

a) B' (A + C)',
b) B' + A' C',
c) (B + (C A)')',
d) B' + (A C)',
e) A' + B' + C'.



1.2 The Boolean representation of this truth table is: **[5]**

a) A C'
b) B
c) A' B' C
d) B C'
e) A B + C'

A	B	C	Output
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

1.3 A multiplexer with 4 select inputs has: **[5]**

a) 1 output, b) 4 outputs, c) 16 outputs, d) 32 outputs

2. Number Systems and Boolean algebra (25 %)

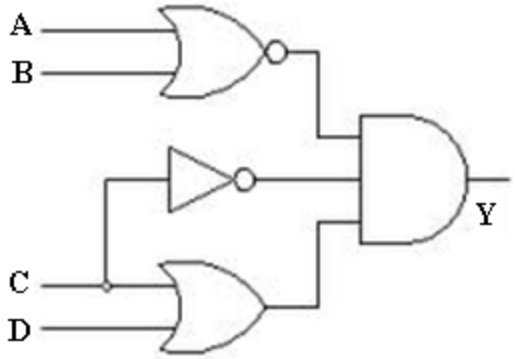
2.1 Complete the following table (except the shaded cells) of equivalent values **[20]**

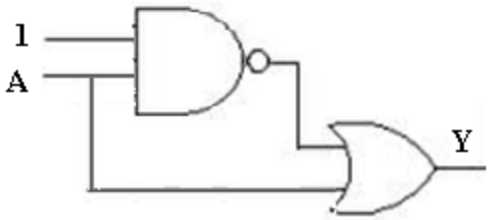
	Binary	**Octal**	**Decimal**	**Hexadecimal**
a)	011110.11			
b)		44		
c)			126	

2.2 Convert the number $143_{(6)}$ into a decimal number! **[5]**

3. Answer the following (15%)

3.1 What are the outputs Y in the figures below?! **[5]**





3.2 **Use Karnaugh Map** to simplify the following Boolean function **[5]**

$F(A,B,C,D)=\prod M(0,1,2,4,5,8,9,11)$ into

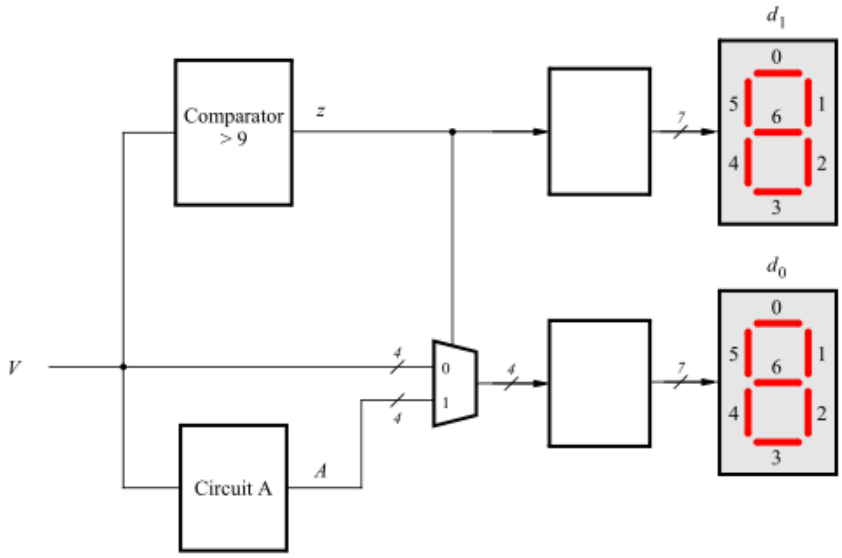
a) sum-of-products form

b) product-of-sums form

3.3 Using 4-to-1 MULTIPLEXER, implement the function from question 3.2 **[5]**

4. VHDL exercise (45%)

A circuit bellow converts a four-bit binary number $V = v_3v_2v_1v_0$ into its two-digit decimal equivalent $D = d_1d_0$. It includes a comparator that checks when the value of V is greater than 9, and uses the output of this comparator in the control of the 7-segment displays. Table 1 shows the required output values. The output z for the comparator circuit can be specified using a single Boolean expression, with the four inputs V_{3-0}.



a) Design this Boolean expression by making a truth table that shows the valuations of the inputs V_{3-0} for which z has to be 1. Use Karnaugh map to optimize the expression for z. Write the VHDL comparator entity with z as output signal. **[15]**

ENTITY comparator IS
PORT(

);
END comparator;
ARCHITECTURE Behaviour OF comparator IS
BEGIN

z<=

END Behaviour;

V_3	V_2	V_1	V_0	z
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

b) Design **circuit_A** entity considering the following:
For the input values V ≤ 9, the circuit A does not matter, because the multiplexer just selects V in these cases. But for the input values V > 9, the multiplexer will select A. You need to design circuit A so that the input V = 1010 gives an output A = 0000, the input V = 1011 gives the output A = 0101, ..., and the input V = 1111 gives the output A = 0101. Design circuit A by making a truth table with the inputs V_{3-0} and the outputs A_{3-0}. **[15]**

```
ENTITY Circuit_A IS
PORT(

                                        );
END Circuit_A;

ARCHITECTURE Behaviour OF Circuit_A IS
BEGIN

A(3)<=

A(2)<=

A(1)<=

A(0)<=

END Behaviour;
```

V_3	V_2	V_1	V_0	A_3	A_2	A_1	A_0
1	0	1	0				
1	0	1	1				
1	1	0	0				
1	1	0	1				
1	1	1	0				
1	1	1	1				

c) Write VHDL code for 4 bit wide 2-to-1 multiplexer, applying PROCESS statement. **[15]**

```
ENTITY mux2to1 IS
PORT (

                                        ) ;
END mux2to1 ;
ARCHITECTURE Behavior OF mux2to1 IS
BEGIN
PROCESS (           )

BEGIN




END PROCESS ;
END Behavior ;
```

```
ENTITY MUX4bit_2to1 IS
PORT(

                                                                    );
END MUX4bit_2to1;
ARCHITECTURE behavior OF MUX4bit_2to1  IS



BEGIN
INSTANCE0:
INSTANCE1:
INSTANCE2:
INSTANCE3:

END behavior;
```