

Logic Design – Lab 1: Universal Gates. Combinational Logic Circuit Design

For the report, please complete the tasks marked in green boxes only (ver. 2020)

Part 1. NAND/NOR implementation of any logic function/circuit

NAND and NOR gate are called universal gates, using NAND only or NOR only any logic function/circuit can be implemented. Digital circuits are frequently constructed with NAND or NOR gates rather than with AND and OR gates. NAND and NOR gates are easier to fabricate with electronic components and are the basic gates used in all IC digital logic families.

Table 1: Digital Logic Gates using Universal gate (NAND only):




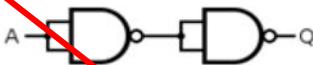

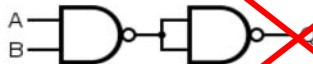








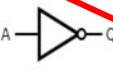


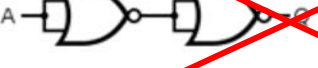

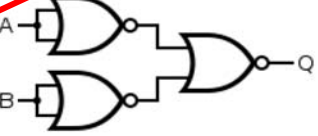
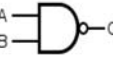
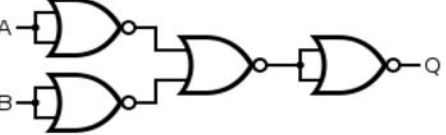
<div></div> <div>NOT</div>	<div></div>	<table><tr><th>A</th><th>Q</th></tr><tr><td>0</td><td></td></tr><tr><td>1</td><td></td></tr><tr><td>0</td><td></td></tr><tr><td>1</td><td></td></tr></table>	A	Q	0		1		0		1						
A	Q																
0																	
1																	
0																	
1																	
<div></div> <div>BUFFER</div>	<div></div>	<table><tr><th>A</th><th>Q</th></tr><tr><td>0</td><td></td></tr><tr><td>1</td><td></td></tr><tr><td>0</td><td></td></tr><tr><td>1</td><td></td></tr></table>	A	Q	0		1		0		1						
A	Q																
0																	
1																	
0																	
1																	
<div></div> <div>AND</div>	<div></div>	<table><tr><th>A</th><th>B</th><th>Q</th></tr><tr><td>0</td><td>0</td><td></td></tr><tr><td>0</td><td>1</td><td></td></tr><tr><td>1</td><td>0</td><td></td></tr><tr><td>1</td><td>1</td><td></td></tr></table>	A	B	Q	0	0		0	1		1	0		1	1	
A	B	Q															
0	0																
0	1																
1	0																
1	1																
<div></div> <div>OR</div>	<div></div>	<table><tr><th>A</th><th>B</th><th>Q</th></tr><tr><td>0</td><td>0</td><td></td></tr><tr><td>0</td><td>1</td><td></td></tr><tr><td>1</td><td>0</td><td></td></tr><tr><td>1</td><td>1</td><td></td></tr></table>	A	B	Q	0	0		0	1		1	0		1	1	
A	B	Q															
0	0																
0	1																
1	0																
1	1																
<div></div> <div>NOR</div>	<div></div>	<table><tr><th>A</th><th>B</th><th>Q</th></tr><tr><td>0</td><td>0</td><td></td></tr><tr><td>0</td><td>1</td><td></td></tr><tr><td>1</td><td>0</td><td></td></tr><tr><td>1</td><td>1</td><td></td></tr></table>	A	B	Q	0	0		0	1		1	0		1	1	
A	B	Q															
0	0																
0	1																
1	0																
1	1																
<div></div> <div>XOR</div>	<div></div>	<table><tr><th>A</th><th>B</th><th>Q</th></tr><tr><td>0</td><td>0</td><td></td></tr><tr><td>0</td><td>1</td><td></td></tr><tr><td>1</td><td>0</td><td></td></tr><tr><td>1</td><td>1</td><td></td></tr></table>	A	B	Q	0	0		0	1		1	0		1	1	
A	B	Q															
0	0																
0	1																
1	0																
1	1																
<div></div> <div>XNOR</div>	<div></div>	<table><tr><th>A</th><th>B</th><th>Q</th></tr><tr><td>0</td><td>0</td><td></td></tr><tr><td>0</td><td>1</td><td></td></tr><tr><td>1</td><td>0</td><td></td></tr><tr><td>1</td><td>1</td><td></td></tr></table>	A	B	Q	0	0		0	1		1	0		1	1	
A	B	Q															
0	0																
0	1																
1	0																
1	1																


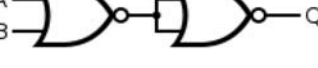
Table 2: Digital Logic Gates using Universal gate (NOR only):


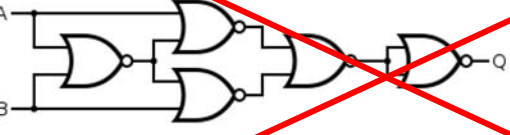
 NOT		A	Q
		0	
		1	
		0	
		1	


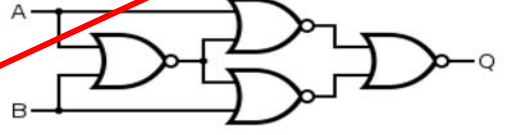
 BUFFER		A	Q
		0	
		1	
		0	
		1	

 AND		A	B	Q
		0	0	
		0	1	
		1	0	
		1	1	

 NAND		A	B	Q
		0	0	
		0	1	
		1	0	
		1	1	

 OR		A	B	Q
		0	0	
		0	1	
		1	0	
		1	1	

 XOR		A	B	Q
		0	0	
		0	1	
		1	0	
		1	1	

 XNOR		A	B	Q
		0	0	
		0	1	
		1	0	
		1	1	

Task:

Construct the logic diagrams of the given circuits (see description on the right side of Tables 1 and Tables 2) with NAND/NOR gates and simulate them using EWB to verify that above designs work correctly (according to tables of truth of specified gates). Result write down to column 'Q' (output) in tables above.

Part 2. Combinational Logic Circuit Design

Design, test and demonstrate circuits, which implement the following functions:

~~1. $f(x_1, x_2, x_3) = x_1 x_2 x_3' + x_1' x_2 x_3 + x_1' x_2' x_3'$~~

~~where $x_1' = \bar{x}_1$ (complement or negation of x_1), etc.~~

2. $f(x_1, x_2, x_3) = x_2 \cdot x_3 + x_1' \cdot x_2 \cdot x_3' + x_1 (x_2' \cdot x_3 + x_2 \cdot x_3')$

~~3. $f(x_1, x_2, x_3, x_4) = \Sigma(1, 2, 4, 5, 9, 11, 12, 13, 14, 15)$~~

~~4. $f(x_1, x_2, x_3, x_4) = \Pi(1, 2, 6, 7, 8, 10)$~~

5. Design a circuit for the following truth table:

x	y	z	m
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

where x, y, z – inputs, m – output.

~~6. Design a circuit with four inputs (a, b, c, d) and one output (y) that produces a "1" at the output if there is an equal number of 0's and 1's at the inputs.
Please make implementation with: 1) NAND gates only and 2) with NOR gates only.~~

For minimization of functions given above should be used Karnaugh Maps method. Perform functional simulation of your designs and have them checked by the lab instructor.

Boolean Algebra

$$X + 0 = X$$

$$X + 1 = 1$$

$$X + X = X$$

$$X + X' = 1$$

$$(X')' = X$$

$$X + Y = Y + X$$

$$X + (Y + Z) = (X + Y) + Z$$

$$X(Y + Z) = XY + XZ$$

$$X + XY = X$$

$$X + X'Y = X + Y$$

$$(X + Y)' = X'Y'$$

$$XY + X'Z + YZ = XY + X'Z$$

$$X \cdot 1 = X$$

$$X \cdot 0 = 0$$

$$X \cdot X = X$$

$$X \cdot X' = 0$$

$$XY = YX$$

$$X(YZ) = (XY)Z$$

$$X + YZ = (X + Y)(X + Z)$$

$$X(X + Y) = X$$

$$X(X' + Y) = XY$$

$$(XY)' = X' + Y'$$

$$(X + Y)(X' + Z)(Y + Z) = (X + Y)(X' + Z)$$

Identity

Idempotent Law

Complement

Involution Law

Commutativity

Associativity

Distributivity

Absorption Law

Simplification

DeMorgan's Law

Consensus Theorem