LAB no. 2.

LOGIC GATES

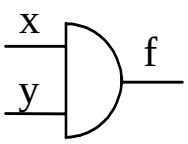
1. Introduction

Combinational logic systems, no matter how complicated they are, they are realized with logic gates. An elementary logic gate implements a two variables function. Thus, the elementary functions are „AND", „OR", „NOT", „NAND", „NOR" and „|XOR". In practice the logic gates can be found as integrated circuits. On such circuit, 1, 2, 3, 4 or 6 gates are availbale according to the number of inputs.

1.1. "AND" gate

The function "AND" has the following intendment:
- if at least one input is logic 0, then the output is logic 0.
- if both inputs are logic 1, then the output is logic 1.

"AND"gate symbol:



Truth table:

y	x	f
0	0	0
0	1	0
1	0	0
1	1	1

Logic ecuation:

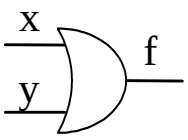
$f=x \cdot y$

$f=x \cap y$

1.2. "OR" gate

The function „OR" has the following intendment:
- the output is true (logic 1) if at least one input is true (logic 1).
- the output is false (logic 0) if both inputs are false (0 logic).

"OR"gate symbol:



Truth table:

y	x	f
0	0	0
0	1	1
1	0	1
1	1	1

Logic ecuation:

$f=x+y$

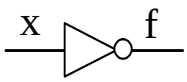
$f=x \cup y$

1.3. "NOT" gate

The function „NOT" has the following intendment:

- the output is true (logic 1) if the input is false (logic 0)
- the output is false (logic 0) if the input is true (logic 0).

"NOT"gate symbol: Truth table: Logic ecuation:



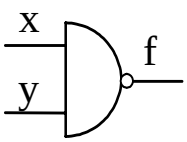
x	f
0	1
1	0

$f=\overline{x}$

1.4. "NAND" gate

The function „NAND" has the following interpretation:
- the output is false (logic 0) if both inputs are true (logic 1)
- the output is true (logic 1) if at least one input is false (logic 0).

"NAND"gate symbol: Truth table: Logic ecuation:



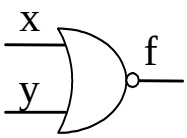
y	x	f
0	0	1
0	1	1
1	0	1
1	1	0

$f=\overline{x \cdot y}$
$f=\overline{x \cap y}$

1.5. "NOR" gate

The function „OR" has the following intendment:
- the output is false (logic 0) if at least one input is true (logic 1).
- the output is true (logic 1) if both inputs are false (0 logic).

"NOR"gate symbol: Truth table: Logic ecuation:



y	x	f
0	0	1
0	1	0
1	0	0
1	1	0

$f=\overline{x+y}$
$f=\overline{x \cup y}$

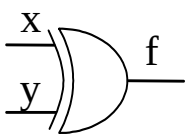
1.6. "XOR" gate (exclussive OR)

The function „XOR" has the following intendment:
- the output is false (logic 0) if both inputs are in the same logic state
- the output is true if the inputs are in opposite logic states.

LAB no. 2.

The XOR gate signals the coincidence of the inputs when its output is false. It realizes the modulo 2 sum ⊕.

"XOR"gate symbol:



Truth table:

y	x	f
0	0	0
0	1	1
1	0	1
1	1	0

Logic ecuation:

$f=x \oplus y$

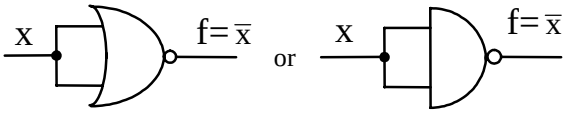
2. Implementation of elementary logic functions using other logic functions

Any of the above elementary logic functions can be realized using other elementary functions. This is useful when implementing a schematic with integrated circuits (IC) in order to minimize the number of ICs used capsules. Thus, you can use free gates available on existing ICs capsules. For example, if you need an inverter (NOT) it is not economic to introduce an IC containing 6 inverters. It can simply be replaced by a gate "NAND" or "NOR" with the inputs tied together. Finally, you will get a price reduction for your circuit.

2.1. Implementing the function "NOT"

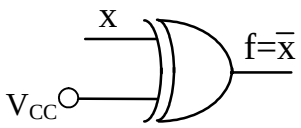
a) using NAND or NOR gates

$f = \overline{x} = \overline{(x \cdot x)} = \overline{(x + x)} \Rightarrow$



b) using XOR gate

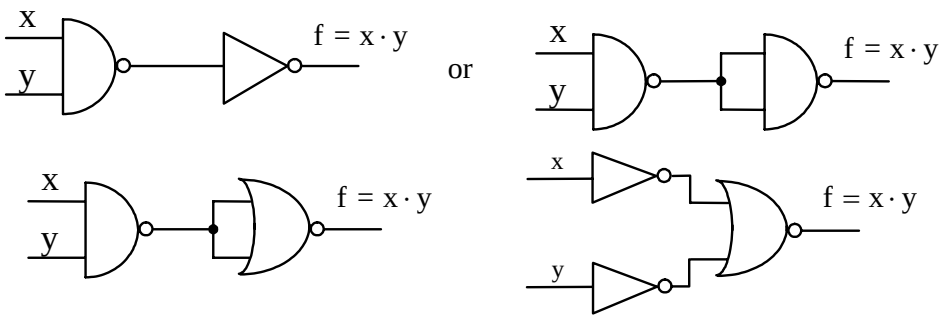
$f = x \oplus 1 = x \cdot \overline{1} + \overline{x} \cdot 1 = \overline{x} \Rightarrow$



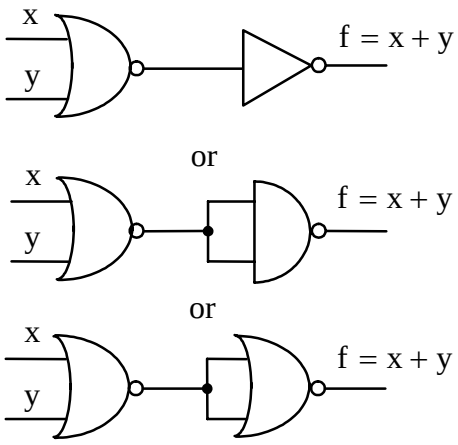
2.2. Implementing the function "AND"

a) $f = \overline{\overline{x \cdot y}}$

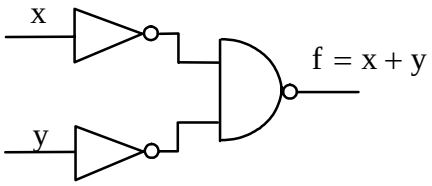
b) $f = x \cdot y = \overline{(\overline{x} + \overline{y})} \Rightarrow$



2.3. Implementing the function "OR"



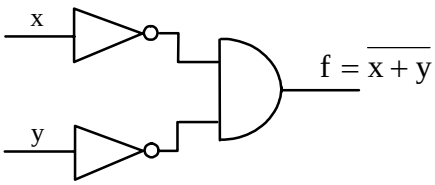
$f = x + y = \overline{(\overline{x + y})} \Rightarrow$



$f = x + y = \overline{(\overline{x} \cdot \overline{y})} \Rightarrow$

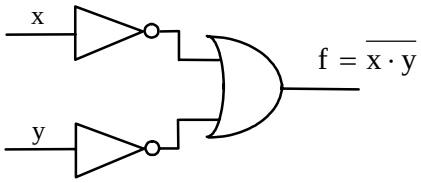
2.4. Implementing the function "NOR"

$f = \overline{x + y} = \overline{x} \cdot \overline{y} \Rightarrow$



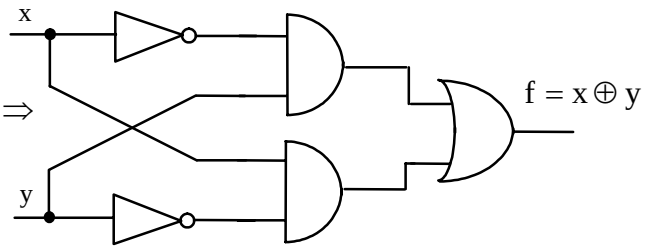
2.5. Implementing the function "NAND"

$$f = \overline{x \cdot y} = \overline{x} + \overline{y} \quad \Rightarrow$$



2.6. Implementing the function "XOR"

$$f = x \oplus y = x \cdot \overline{y} + \overline{x} \cdot y \quad \Rightarrow$$



3. Lab works

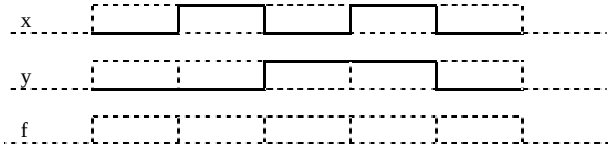
Complete the following sheet according to the indications.

LAB no. 2.

LAB SHEET

Verify the operation of the elementary logic gates (AND, OR, NOT, NAND, NOR, XOR) using MaxPlusII. Setup in the scf file the input signals as shown below and draw the output *f* as obtained from simulation. Extract for each of the gates the truth table from the simulation waveforms and measure the delay time between the inputs and the output. Compare the extracted truth tables with the tables given above.

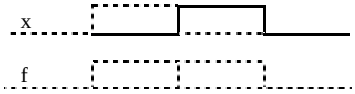
AND:



t_{delay}=

x	y	f

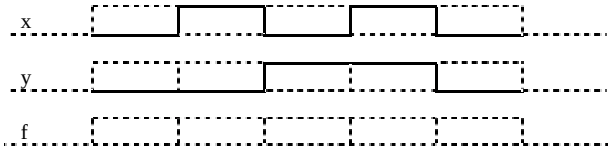
NOT:



t_{delay}=

x	y	f

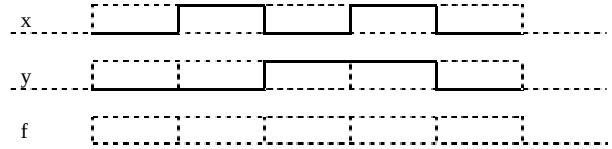
OR:



t_{delay}=

x	y	f

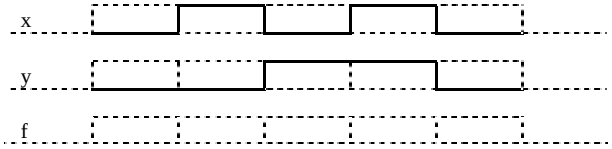
NAND:



t_{delay}=

x	y	f

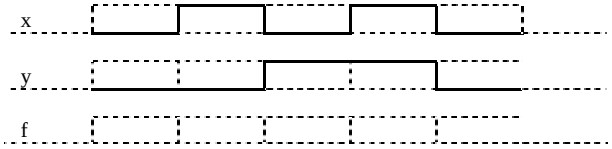
NOR:



t_{delay}=

x	y	f

XOR:



t_{delay}=

x	y	f