LOGICAL CIRCUITS

Theory

Controlling automatics and computers involve elements which may have two different states and the operation of the whole system is based on their connection. The description of these systems happens by means of the **Boolean algebra** (logical algebra).

According to rigorous logic, a sentence can be true or false. Assigning the value 1 to „true" and 0 to „false", a complete logical algebra can be built with sentences – a kind of Boolean algebra.
There are three operations defined in Boolean algebra: **inversion** of an element, **sum** and **product** of two or more elements. In logical algebra **inversion corresponds to negation**, **sum corresponds to "OR"** and the **product corresponds to "AND"**.
We have a sentence $\mathbf{A}$: "The teacher talks".
The negation of this sentence is $\overline{\mathbf{A}}$: "The teacher does not talk".
We have another sentence: $\mathbf{B}$ "The teacher sits in the office".
The sentences $\mathbf{A}$ and $\mathbf{B}$ can be combined into a compound sentence.
One combination is $\mathbf{A}$ AND $\mathbf{B}$**:** "The teacher sits in the office and talks". This sentence is true only if both $\mathbf{A}$ and $\mathbf{B}$ are true.
The other combination is $\mathbf{A}$ OR $\mathbf{B}$. "The teacher either sits in the office or talks". This statement is true whenever the teacher sits in the office, either talking or being silent, or talks anywhere.
In Boolean algebra $\mathbf{A}$ AND **B** is written as $\mathbf{A \cdot B}$, and $\mathbf{A}$ OR $\mathbf{B}$ is written as $\mathbf{A+B}$.

Logical variables may assume two values: 1 or 0 (logical "true" or "false").
The **elementary operations** with logical variables are:

- negation: $\mathbf{Y} = \overline{\mathbf{A}}$
  $\mathbf{Y}$ is true whenever $\mathbf{A}$ is false (and vice versa).
- addition, OR connection: $\mathbf{Y = A + B}$
  $\mathbf{Y}$ is true whenever $\mathbf{A}$ is true or $\mathbf{B}$ is true or both are true.
- multiplication, AND connection: $\mathbf{Y = A \cdot B}$
  $\mathbf{Y}$ is true whenever $\mathbf{A}$ and $\mathbf{B}$ are true

The addition and multiplication of more than two variables is defined similarly.

Truth tables give the value of a compound sentence in terms of the truth-level of the constituent sentences. The truth tables of the basic operations are:

negation, inverse
NOT

$\mathbf{A}$	$\overline{\mathbf{A}}$
1	0
0	1

multiplication, product
AND

$\mathbf{A}$	$\mathbf{B}$	$\mathbf{A \cdot B}$
0	0	0
0	1	0
1	0	0
1	1	1

addition, sum
OR

$\mathbf{A}$	$\mathbf{B}$	$\mathbf{A+B}$
0	0	0
0	1	1
1	0	1
1	1	1

These operations – like ordinary algebraic operations – are
commutative:

$$\mathbf{A}+\mathbf{B}=\mathbf{B}+\mathbf{A} \qquad \mathbf{A}\cdot\mathbf{B}=\mathbf{B}\cdot\mathbf{A}$$

associative:

$$\mathbf{A}+(\mathbf{B}+\mathbf{C})=(\mathbf{A}+\mathbf{B})+\mathbf{C} \qquad \mathbf{A}\cdot(\mathbf{B}\cdot\mathbf{C})=(\mathbf{A}\cdot\mathbf{B})\cdot\mathbf{C}$$

and distributive:

$$\mathbf{A}\cdot(\mathbf{B}+\mathbf{C})=\mathbf{A}\cdot\mathbf{B}+\mathbf{A}\cdot\mathbf{C}$$

In Boolean algebra – unlike ordinary algebra – a second distributive rule is also valid:

$$\mathbf{A}+\mathbf{B}\cdot\mathbf{C}=(\mathbf{A}+\mathbf{B})\cdot(\mathbf{A}+\mathbf{C})$$

There is a null element 0 and an identity E or 1. The null element is a sentence that never is true. The unit element 1 is a sentence which is always true.

There is a group of useful theorems of Boolean algebra which help in developing the logic for a given operation.
Single variable theorems:

$$\mathbf{A}+\mathbf{A}=\mathbf{A} \qquad \mathbf{A}\cdot\mathbf{A}=\mathbf{A}$$
$$\mathbf{A}+\overline{\mathbf{A}}=\mathbf{1} \qquad \mathbf{A}\cdot\overline{\mathbf{A}}=\mathbf{0}$$
$$\mathbf{A}=\overline{\overline{\mathbf{A}}}$$

Identity and null operations:

$$\mathbf{A}+\mathbf{1}=\mathbf{1} \qquad \mathbf{A}\cdot\mathbf{1}=\mathbf{A}$$
$$\mathbf{A}+\mathbf{0}=\mathbf{A} \qquad \mathbf{A}\cdot\mathbf{0}=\mathbf{0}$$
$$\overline{\mathbf{1}}=\mathbf{0} \qquad \overline{\mathbf{0}}=\mathbf{1}$$
$$\mathbf{0}+\mathbf{0}=\mathbf{0} \qquad \mathbf{0}\cdot\mathbf{0}=\mathbf{0}$$
$$\mathbf{0}+\mathbf{1}=\mathbf{1} \qquad \mathbf{0}\cdot\mathbf{1}=\mathbf{0}$$
$$\mathbf{1}+\mathbf{1}=\mathbf{1} \qquad \mathbf{1}\cdot\mathbf{1}=\mathbf{1}$$

Two-variable theorems:

$$\mathbf{A}+\mathbf{A}\cdot\mathbf{B}=\mathbf{A}$$
$$\mathbf{A}+\overline{\mathbf{A}}\cdot\mathbf{B}=\mathbf{A}+\mathbf{B}$$

Furthermore the De Morgan's Laws:

$$\overline{\mathbf{A}+\mathbf{B}}=\overline{\mathbf{A}}\cdot\overline{\mathbf{B}}$$
$$\overline{\mathbf{A}\cdot\mathbf{B}}=\overline{\mathbf{A}}+\overline{\mathbf{B}}$$

or with more variables

$$\overline{\mathbf{A}+\mathbf{B}+\mathbf{C}+\cdots}=\overline{\mathbf{A}}\cdot\overline{\mathbf{B}+\mathbf{C}+\cdots}=\overline{\mathbf{A}}\cdot\overline{\mathbf{B}}\cdot\overline{\mathbf{C}+\cdots}=\overline{\mathbf{A}}\cdot\overline{\mathbf{B}}\cdot\overline{\mathbf{C}}\cdot\ldots\,,$$
$$\overline{\mathbf{A}\cdot\mathbf{B}\cdot\mathbf{C}\cdot\ldots}=\overline{\mathbf{A}}+\overline{\mathbf{B}\cdot\mathbf{C}\cdot\ldots}=\overline{\mathbf{A}}+\overline{\mathbf{B}}+\overline{\mathbf{C}\cdot\ldots}=\overline{\mathbf{A}}+\overline{\mathbf{B}}+\overline{\mathbf{C}}+\cdots\,.$$

These relations are very useful in simplification of logical functions.

By means of the basic operations we can define **logical functions**, e.g. $\mathbf{Y}_1=\mathbf{A}\cdot\mathbf{B}+\mathbf{B}\cdot\overline{\mathbf{C}}$, or $\mathbf{Y}_2=\mathbf{A}\cdot\mathbf{B}+\mathbf{B}\cdot\overline{\mathbf{C}}+\overline{\mathbf{B}+\mathbf{C}}$, where $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are the independent variables, and $\mathbf{Y}$ is the value of the function (all of them are logical variables).
Generally a logical function can be defined either with basic operations or with its truth-table or in other ways.

The truth-table of any logical function can be constructed on the basis of the truth tables of the basic operations (by substituting the possible values of the independent variables in the function).

E.g. for the function $\mathbf{Y_1 = A \cdot B + B \cdot \bar{C}}$:

A	B	C	$\mathbf{Y_1 = A \cdot B + B \cdot \bar{C}}$
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	1

and for the function $\mathbf{Y_2 = A \cdot B + B \cdot \bar{C} + \overline{B + C}}$:

A	B	C	$\mathbf{Y_2 = A \cdot B + B \cdot \bar{C} + \overline{B + C}}$
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

The **standard form** of the logical function is derived from the truth-table as the sum of the rows where Y = 1 .

E.g. for the function $\mathbf{Y_1 = A \cdot B + B \cdot \bar{C}}$ the standard form is

$$\mathbf{Y_1 = \bar{A} \cdot B \cdot \bar{C} + A \cdot B \cdot \bar{C} + A \cdot B \cdot C}$$

Simplification leads to the form we started from:

$$\mathbf{Y_1 = [\bar{A} \cdot B \cdot \bar{C} + A \cdot B \cdot \bar{C}] + [A \cdot B \cdot \bar{C} + A \cdot B \cdot C] = B \cdot \bar{C} \cdot (\bar{A} + A) + A \cdot B \cdot (\bar{C} + C) = B \cdot \bar{C} + A \cdot B}$$

Note that the second term was duplicated because it could be used in two simplifications. This is based on the theorem $\mathbf{A + A = A}$, in the opposite direction: $\mathbf{A = A + A}$.

For the function $\mathbf{Y_2 = A \cdot B + B \cdot \bar{C} + \overline{B + C}}$ the standard form is

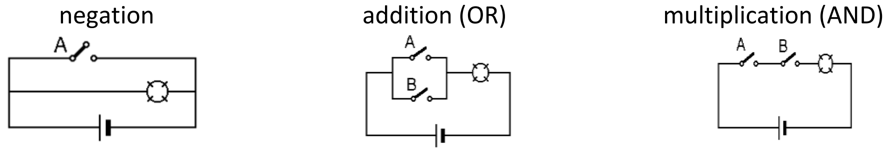
$$\mathbf{Y_2 = \bar{A} \cdot \bar{B} \cdot \bar{C} + \bar{A} \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot \bar{C} + A \cdot B \cdot \bar{C} + A \cdot B \cdot C}$$

In this case simplification leads to a form that is simpler than the starting form:

$$\mathbf{Y_2 = [\bar{A} \cdot \bar{B} \cdot \bar{C} + \bar{A} \cdot B \cdot \bar{C}] + [A \cdot \bar{B} \cdot \bar{C} + A \cdot B \cdot \bar{C}] + [A \cdot B \cdot \bar{C} + A \cdot B \cdot C] =}$$
$$\mathbf{= \bar{A} \cdot \bar{C} \cdot (\bar{B} + B) + A \cdot \bar{C} \cdot (\bar{B} + B) + A \cdot B \cdot (\bar{C} + C) = \bar{A} \cdot \bar{C} + A \cdot \bar{C} + A \cdot B =}$$
$$\mathbf{= \bar{C} \cdot (\bar{A} + A) + A \cdot B = A \cdot B + \bar{C}}$$

The logical operations among sentences can be modeled by **logical circuits**.

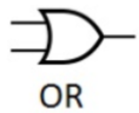
The basic operations can be demonstrated with simple circuits containing switches and a light bulb. (The switches correspond to independent variables, and the light bulb represents the value of the function.)



Mostly such circuits consist of **logic gates**. A gate can have one or more inputs but only one output. Usually the letters A, B, C, etc. are used to label inputs, and Y is used to label the output. Below the inputs are shown on the left and the output on the right.
The electric potential at the inputs and at the output can be high (a few volts) or low (a few tenths of volts). The high level corresponds to the logical value "true" = 1 and the low level corresponds to "false"=0.
Logic gates are available on special IC-s (chips) which usually contain several gates of the same type. Gates are identified by their function.

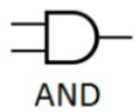
OR gate
This is an OR relation: $\mathbf{Y} = \mathbf{A} + \mathbf{B}$.

$\mathbf{A}$	$\mathbf{B}$	$\mathbf{A+B}$
0	0	0
0	1	1
1	0	1
1	1	1



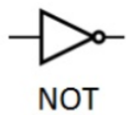
AND gate
This is an AND relation: $\mathbf{Y} = \mathbf{A} \cdot \mathbf{B}$

$\mathbf{A}$	$\mathbf{B}$	$\mathbf{A \cdot B}$
0	0	0
0	1	0
1	0	0
1	1	1



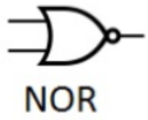
For some reasons OR and AND gates are not used.
The available gates are: **NOT (inverter)**, **NOR** (negated OR), and **NAND** (negated AND). The gate symbols have a circle on their output which means that their function includes inverting.

NOT gate (inverter)
The output is the inverse of the input: $\mathbf{Y} = \overline{\mathbf{A}}$.
The output $\mathbf{Y}$ is true when the input $\mathbf{A}$ is **NOT** true: $\mathbf{Y}$ = **NOT** $\mathbf{A}$
A NOT gate can only have one input. It is also called an inverter.



NOR gate (NOR = Not OR)

This is an OR relation with the output inverted: $\mathbf{Y} = \overline{\mathbf{A} + \mathbf{B}}$.

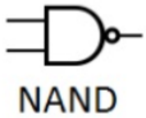


$\mathbf{A}$	$\mathbf{B}$	$\overline{\mathbf{A} + \mathbf{B}}$
0	0	1
0	1	0
1	0	0
1	1	0

A NOR gate can have two or more inputs; its output is true if no inputs are true.

NAND gate (NAND = Not AND)

This is an AND relation with the output inverted: $\mathbf{Y} = \overline{\mathbf{A} \cdot \mathbf{B}}$



$\mathbf{A}$	$\mathbf{B}$	$\overline{\mathbf{A} \cdot \mathbf{B}}$
0	0	1
0	1	1
1	0	1
1	1	0

A NAND gate can have two or more inputs; its output is true if not all inputs are true.

Measurement

You will be given a problem and your task is to realize the given logical function.

1. Fill in the truth-table.

2. Write the standard form of the logical function.

3. Simplify it.

4. Draw the connection and construct the circuit using NOT, OR and AND gates, and test it.

5. Transform the simplified function using the De Morgan's laws to a form involving only NOT, NOR and NAND operations.

6. Draw the connection and construct the circuit using NOT, NOR and NAND gates, and test it.

Example 1:
We have an alternative switch in the corridor: the lamp can be switched on/off at both ends of it. Create a logical circuit using inverters, NOR and NAND gates, where $\mathbf{A}$ represents the switch at one end of the corridor and $\mathbf{B}$ the switch at the other end, and the output $\mathbf{Y}$ represents the lamp.

Solution:
1. Fill in the truth-table:
In the beginning both switches $\mathbf{A}$ and $\mathbf{B}$ are turned off and the lamp is dark. Now, if any of the switches is turned on the lamp will start to light, and then we can turn off the light by changing the position of any of the switches so the truth table of this logical function is:

A	B	Y
0	0	0
1	0	1
0	1	1
1	1	0

2. Write the standard form of the logical function:

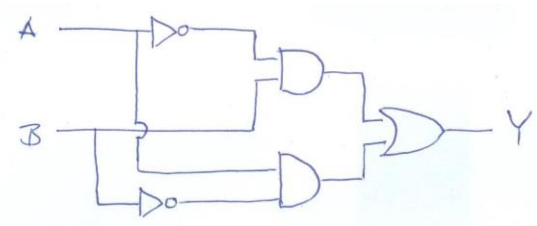
$$\mathbf{Y} = \mathbf{A}\cdot\overline{\mathbf{B}} + \overline{\mathbf{A}}\cdot\mathbf{B}$$

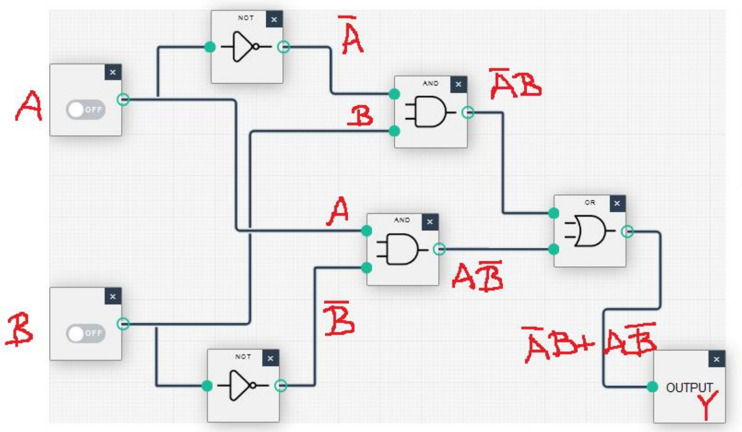
3. Simplify it: This function cannot be simplified.

4. Draw the connection and construct the circuit using NOT, OR and AND gates, and test it.

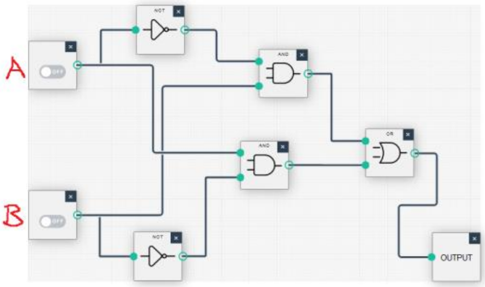
$$\mathbf{Y} = \mathbf{A}\cdot\overline{\mathbf{B}} + \overline{\mathbf{A}}\cdot\mathbf{B}$$

Two NOT gates, two AND gates and one OR gate are needed:

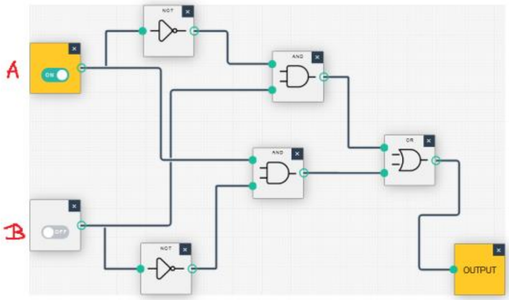




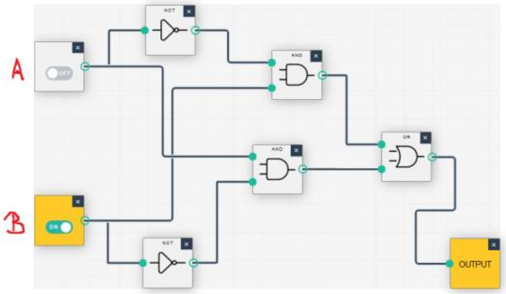
A	B	Y
0	0	0



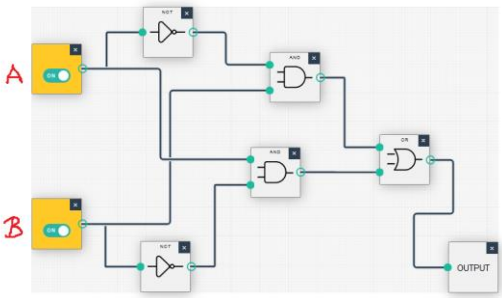
A	B	Y
1	0	1



A	B	Y
0	1	1



A	B	Y
1	1	0



5. Transform it to a form involving only NOT, NOR and NAND operations:

$$\mathbf{Y} = \mathbf{A}\cdot\overline{\mathbf{B}} + \overline{\mathbf{A}}\cdot\mathbf{B} = \overline{\overline{\mathbf{A}\cdot\overline{\mathbf{B}} + \overline{\mathbf{A}}\cdot\mathbf{B}}} = \overline{\overline{\mathbf{A}\cdot\overline{\mathbf{B}}}\cdot\overline{\overline{\mathbf{A}}\cdot\mathbf{B}}}$$

For this 2 inverters and 3 NAND gates are needed,
but we can make further transformations to get less gates:

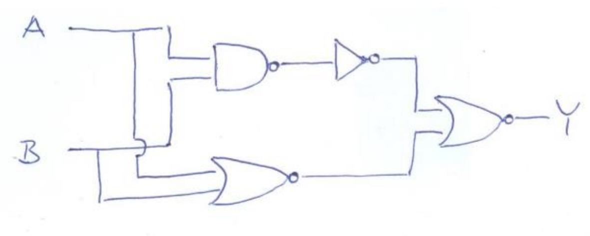
$$\mathbf{Y} = \overline{\overline{\mathbf{A}\cdot\overline{\mathbf{B}}}\cdot\overline{\overline{\mathbf{A}}\cdot\mathbf{B}}} = \overline{(\overline{\mathbf{A}}+\mathbf{B})\cdot(\mathbf{A}+\overline{\mathbf{B}})} = \overline{\mathbf{A}\cdot\mathbf{B} + \overline{\mathbf{A}}\cdot\overline{\mathbf{B}}} = \overline{\overline{\overline{\mathbf{A}\cdot\mathbf{B}}} + \overline{\mathbf{A}+\mathbf{B}}}$$

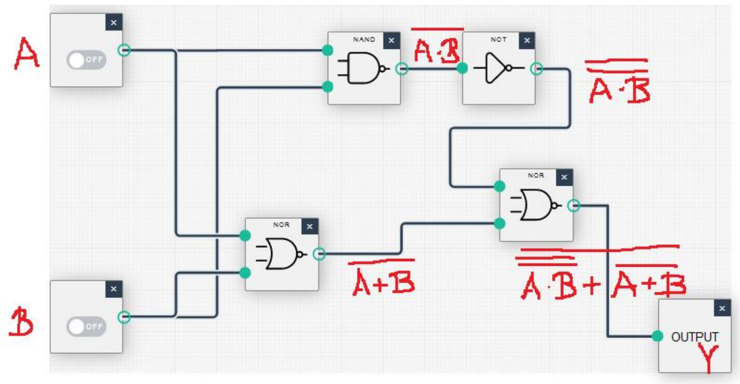
so it can be realized using one inverter, one NAND gate and two NOR gates.

6. Draw the connection and construct the circuit using NOT, NOR and NAND gates, and test it.

$$\mathbf{Y} = \overline{\overline{\overline{\mathbf{A}\cdot\mathbf{B}}} + \overline{\mathbf{A}+\mathbf{B}}}$$

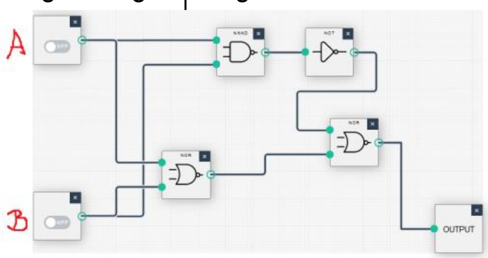
One inverter, one NAND gate and two NOR gates are needed:



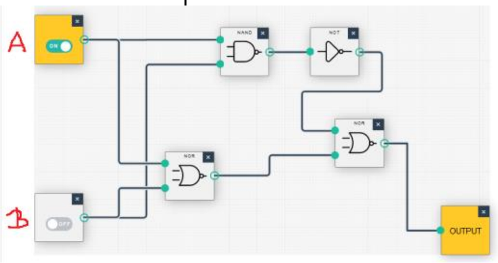


Testing:

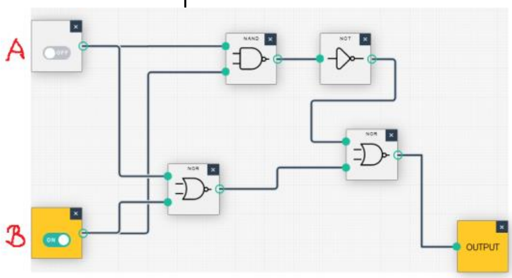
A	B	Y
0	0	0



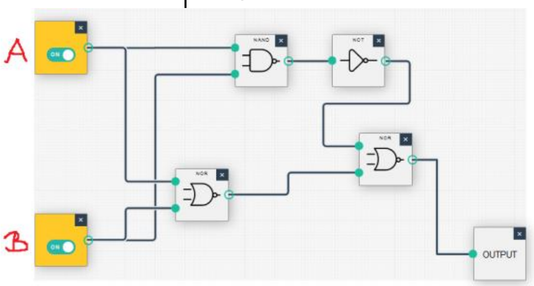
A	B	Y
1	0	1



A	B	Y
0	1	1



A	B	Y
1	1	0



Example 2:

We want to construct a device that gives a signal when a student passes the exam. There is an examining committee of a professor, of a lecturer and of an assistant. If the professor says that the student can pass this means 3 points. The lecturer's vote means 2 points. The vote of the assistants is 1 point. No vote means 0 points. The student passes if he gets at least 4 points. The events P, L, A mean voting of the professor, of the lecturer, of the assistant with "yes", respectively. The output Y is the event that the student passes the exam.

Solution:

1. Fill in the truth-table:

E.g.

in line 6: P = 1: 3 points, L = 1: 2 points, A = 0: 0 points, SUM: 5 points > 4 points → Y = 1
in line 7: P = 0: 0 points, L = 1: 2 points, A = 1: 1 points, SUM: 3 points < 4 points → Y = 0

P	L	A	Y
0	0	0	0
0	0	1	0
0	1	0	0
1	0	0	0
1	0	1	1
1	1	0	1
0	1	1	0
1	1	1	1

2. Write the standard form of the logical function:

$$\mathbf{Y = P{\cdot}\bar{L}{\cdot}A + P{\cdot}L{\cdot}\bar{A} + P{\cdot}L{\cdot}A}$$

3. Simplify it:

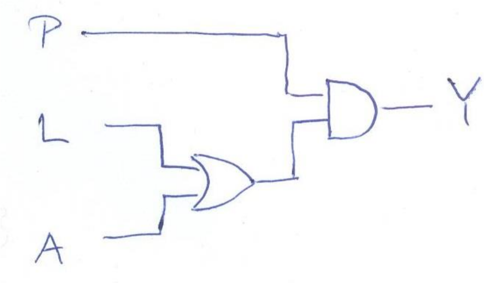
$$\mathbf{Y = [P{\cdot}\bar{L}{\cdot}A + P{\cdot}L{\cdot}A] + [P{\cdot}L{\cdot}\bar{A} + P{\cdot}L{\cdot}A] = P{\cdot}A{\cdot}(\bar{L} + L) + P{\cdot}L{\cdot}(\bar{A} + A) = P{\cdot}A + P{\cdot}L}$$

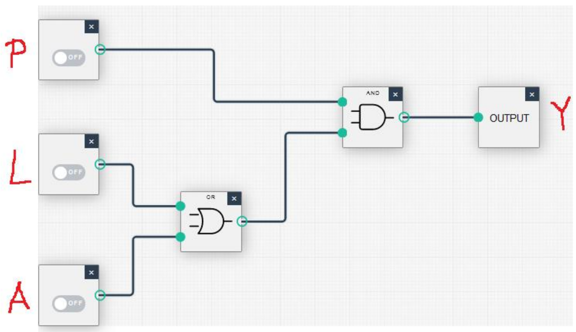
$$\mathbf{Y = P{\cdot}(L + A)}$$

4. Draw the connection and construct the circuit using NOT, OR and AND gates, and test it.

$$\mathbf{Y = P{\cdot}(L + A)}$$

One OR gate and one AND gate is needed.





TEST: 8 variations!!!

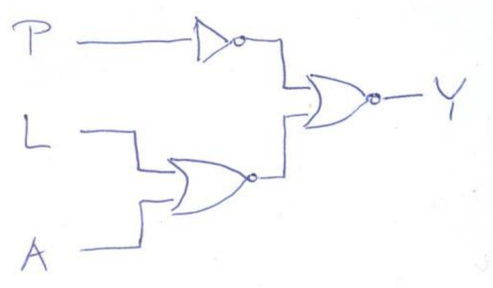
5. Transform the simplified function using the De Morgan's laws to a form involving only NOT, NOR and NAND operations.

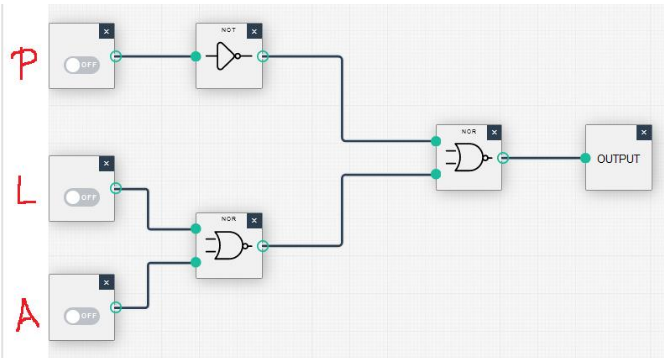
$$\mathbf{Y} = \overline{\overline{\mathbf{P} \cdot (\mathbf{L} + \mathbf{A})}} = \overline{\overline{\mathbf{P}} + \overline{\mathbf{L} + \mathbf{A}}}$$

6. Draw the connection and construct the circuit using NOT, NOR and NAND gates, and test it.

$$\mathbf{Y} = \overline{\overline{\mathbf{P}} + \overline{\mathbf{L} + \mathbf{A}}}$$

This function can be realized by one NOT gate and two NOR gates.

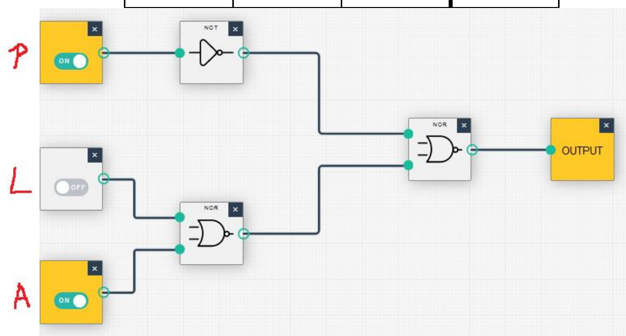




TEST: 8 variations!!!

e.g.

P	L	A	Y
1	0	1	1



P	L	A	Y
0	1	1	0