Ministry of Education, Culture, and Research of the Republic of Moldova
Technical University of Moldova
Department of Software Engineering and Automatics

Report

Computer Architecture

Laboratory Work No.1

Elaborated by: Lupașcu Felicia, FAF-212

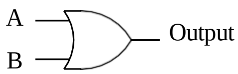
Checked by: Voitcovschi Vladislav

Chisinau 2023

Question 1

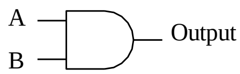
Identify each of these logic gates by name, and complete their respective truth tables:

OR



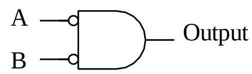
A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

AND



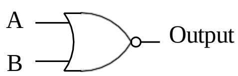
A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

NEG-AND



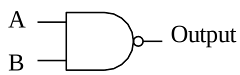
A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0

NOR



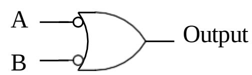
A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0

NAND



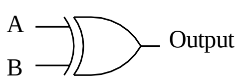
A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

NEG-OR



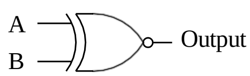
A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

XOR



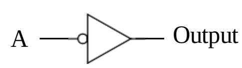
A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

XNOR



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

NOT

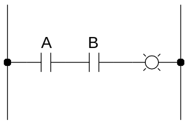


A	Output
0	1
1	0

Question 2

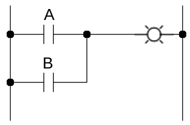
Identify each of these relay logic functions by name (AND, OR, NOR, etc.) and complete their respective truth tables:

AND



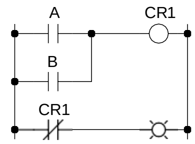
A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

OR



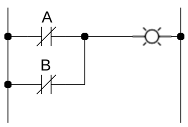
A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

NOR



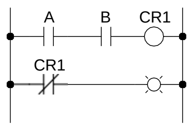
A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0

NEG-OR



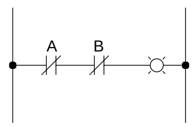
A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

NAND



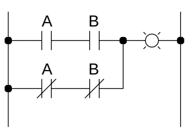
A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

NEG-AND



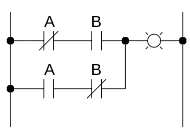
A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0

XNOR



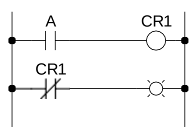
A	B	Output
0	0	1
0	1	0
1	0	0
1	1	1

XOR



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

NOT



A	Output
0	1
1	0

Question 3

A Karnaugh map is nothing more than a special form of truth table, useful for reducing logic functions into minimal Boolean expressions. Here is a truth table for a specific three-input logic circuit:

A	B	C	Out
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Complete the following Karnaugh map, according to the values found in the above truth table:

Answer:

AB \ C	0	1
00	1	1
01	0	1
11	0	0
10	0	1

Question 4

A Karnaugh map is nothing more than a special form of truth table, useful for reducing logic functions into minimal Boolean expressions. Here is a truth table for a specific four-input logic circuit:

A	B	C	D	Out
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

Complete the following Karnaugh map, according to the values found in the above truth table:

Answer:

AB \ CD	00	01	11	10
00	0	1	0	0
01	0	1	0	0
11	0	1	1	0
10	0	1	1	1

Question 5

Here is a truth table for a four-input logic circuit:

A	B	C	D	Out
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

If we translate this truth table into a Karnaugh map, we obtain the following result:

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	0	1	1	0
10	0	0	0	0

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	0	1	1	0
10	0	0	0	0

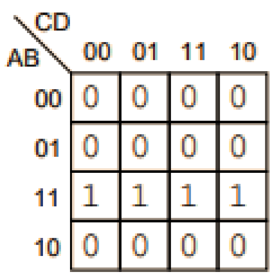
If you look at the input variables (A, B, C, and D), you should notice that only two of them change within this cluster of four 1's. The other two variables hold the same value for each of these conditions where the output is a "1". Identify which variables change, and which stay the same, for this cluster.

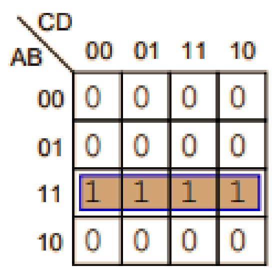
Answer: In this case (for this cluster of four 1's), the only two inputs that change are variables A and C, and variables B and D stay the same (B = 1 and D = 1) for each of the four "high" outputs.

Question 6

Here is a truth table for a four-input logic circuit:

A	B	C	D	Out
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1





Answer: In this case variables A and B are the only inputs that stay constant for each ``1`` conditions on the Karnaugh map a. The simplified Boolean expression for it is AB.

0 0 1
0 1 0
1 0 0
1 1 1

AB

$$AB\bar{C}\bar{D} + AB\bar{C}D + ABC\bar{D} + ABCD = AB\bar{C}\overset{1}{(\bar{D}+D)} +$$

$$+ ABC\overset{1}{(\bar{D}+D)} = ABC\overset{1}{(\bar{D}+D)} = AB\bar{C} + ABC =$$

$$= AB\overset{1}{(\bar{C}+C)} = \quad AB$$

Question 7

One of the essential characteristics of Karnaugh maps is that the input variable sequences are always arranged in gray code sequence. That is, you never see a Karnaugh map with the input combinations arranged in binary order:

Proper form

AB \ CD	00	01	11	10
00				
01				
11				
10				

Improper form

AB \ CD	00	01	10	11
00				
01				
10				
11				

The reason for this is apparent when we consider the use of Karnaugh maps to detect common variables in output sets. For instance, here we have a Karnaugh map with a cluster of four 1's at the center:

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	0	1	1	0
10	0	0	0	0

Arranged in this order, it is apparent that two of the input variables have the same values for each of the four" high" output conditions. Re-draw this Karnaugh map with the input variables sequenced in binary order, and comment on what happens. Can you still tell which input variables remain the same for all four output conditions?

Answer:

AB \ CD	00	01	10	11
00	0	0	0	0
01	0	1	0	1
10	0	0	0	0
11	0	1	0	1

B=1 and D=1 for all four high output conditions.
In this case, the statement is saying that in a particular Karnaugh map, the value of B and D are always 1 when the output is "high." However, this is not immediately clear just by looking at the proximity of the cells in the map.

So, I'm pointing out that the relationship between the inputs and the output is not as straightforward as it may have appeared in a previous map. The value of B and D can still be determined, but it may require a closer examination of the map or a different method of analysis.

Question 8

Examine this truth table and corresponding Karnaugh map:

A	B	C	D	Out
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

AB \ CD	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	0	0	0	0
10	1	0	0	1

Though it may not be obvious from first appearances, the four” high” conditions in the Karnaugh map belong to the same group. To make this more apparent, I will draw a new (oversized) Karnaugh map template, with the Gray code sequences repeated twice along each axis:

Answer:

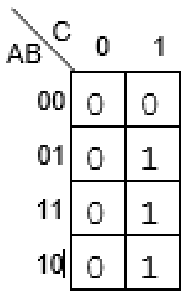
AB \ CD	00	01	11	10	00	01	11	10
00	1	0	0	1	1	0	0	1
01	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0
10	1	0	0	1	1	0	0	1
00	1	0	0	1	1	0	0	1
01	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0
10	1	0	0	1	1	0	0	1

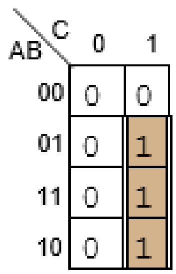
Question 9

A student is asked to use Karnaugh mapping to generate a minimal SOP expression for the following truth table:

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

Following the truth table shown, the student plots this Karnaugh map:





Answer: A cluster of three, as seen in this scenario, leads to an incorrect conclusion. The purpose of this question is to illustrate how it is incorrect to identify clusters of arbitrary size in a Karnaugh map.

Question 10

State the rules for properly identifying common groups in a Karnaugh map.

Answer:

To properly identify common groups in a Karnaugh map, the following rules should be followed:

- Group size: The size of each group should be a power of two, starting with 1 and doubling (1, 2, 4, 8, etc.).

- Adjacent cells: The cells in a group should be adjacent to each other, either vertically, horizontally or diagonally.

- Continuous loop: The cells in a group should form a continuous loop with no breaks.

- Don't overlap: The groups should not overlap or share any cells.

- Cover all 1s: The groups should cover all the "1" cells in the map, without leaving any out.

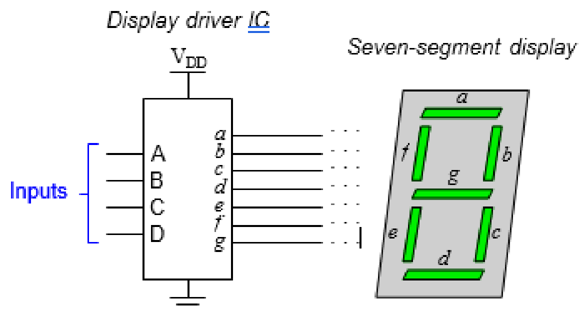
By following these rules, you can ensure that the common groups in a Karnaugh map are properly identified and that the resulting simplified Boolean expression is accurate.

Summmary:

1. No zeros allowed.
2. No diagonals.
3. Only power of 2 number of cells in each group.
4. Groups should be as large as possible.
5. Every one must be in at least one group.
6. Overlapping allowed.
7. Wrap around allowed.
8. Fewest number of groups possible.

Question 11

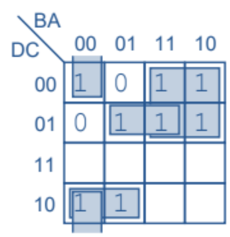
A *seven segment decoder* is a digital circuit designed to drive a very common type of digital display device: a set of LED (or LCD) segments that render numerals 0 through 9 at the command of a four-bit code:



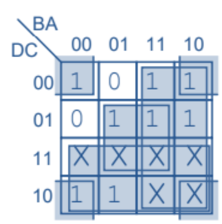
DC \ BA	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11				
10	1	1		

Answer:

Karnaugh map groupings with strict "1" groups: $\overline{D}B + \overline{D}CA + D\overline{C}\,\overline{B} + \overline{C}\,\overline{B}\,\overline{A}$



Karnaugh map groupings with "don't care" wildcards: $D + B + CA + \overline{C}\,\overline{A}$



the ability to use "don't care" states as "wildcard" placeholders in the Karnaugh map cells increases the chances of creating bigger groups.

Question 12

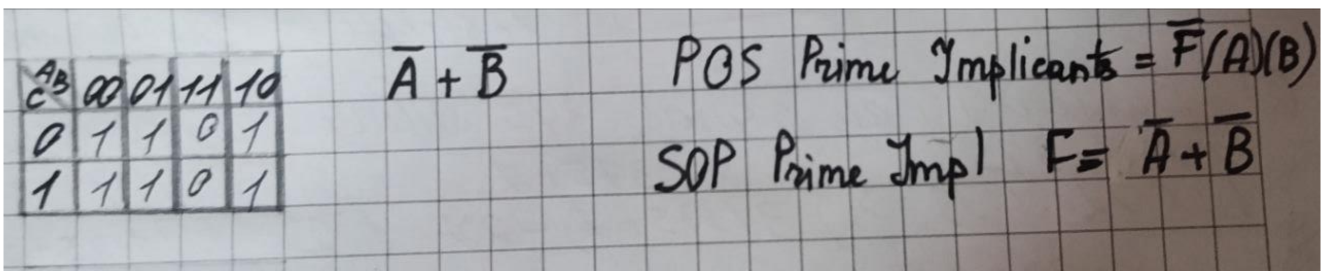
When designing a circuit to emulate a truth table such as this where nearly all the input conditions result in ”1” output states, it is easier to use Product-of-Sums (POS) expressions rather than Sum-of-Products (SOP) expressions:

A	B	C	Output
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	0
1	1	1	0

Is it possible to use a Karnaugh map to generate the appropriate POS expression for this truth table, or are Karnaugh maps limited to SOP expressions only? Explain your answer, and how you were able to obtain it.

Answer:

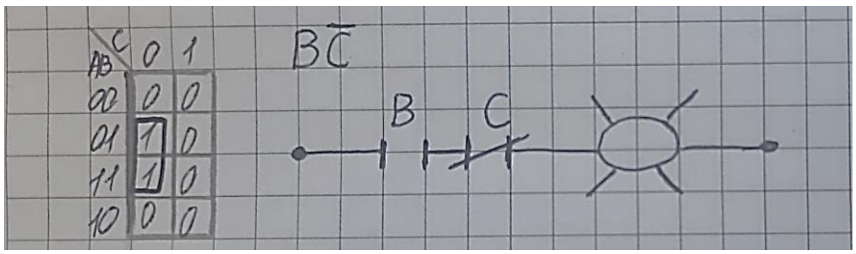
Yes, it is possible to use a Karnaugh map to generate a POS expression for this truth table. Karnaugh maps can be used to generate both SOP and POS expressions. The process for generating a POS expression from a Karnaugh map is the same as for SOP expressions, but instead of marking the cells corresponding to "1" outputs, the cells corresponding to "0" outputs are marked. The resulting expression is then simplified using the same rules for both SOP and POS expressions.



Question 13

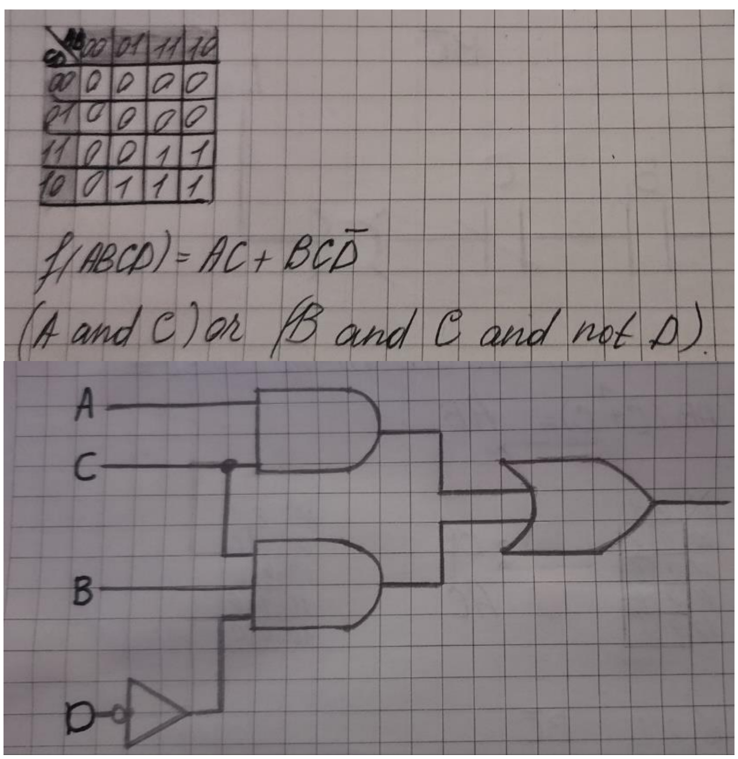
Use a Karnaugh map to generate a simple Boolean expression for this truth table, and draw a relay logic circuit equivalent to that expression:

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



Question 14

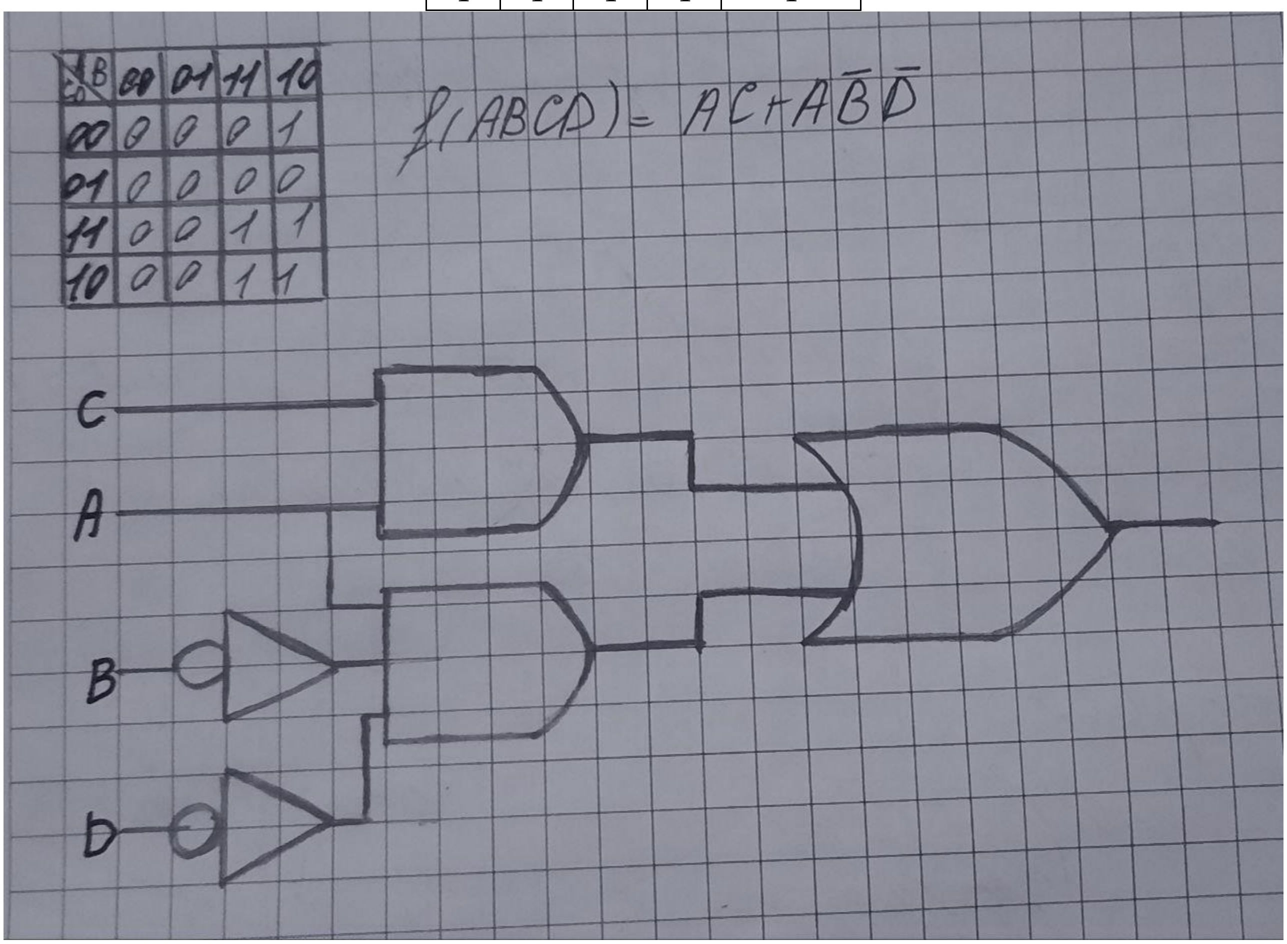
Use a Karnaugh map to generate a simple Boolean expression for this truth table, and draw a gate circuit equivalent to that expression:



Question 15

Use a Karnaugh map to generate a simple Boolean expression for this truth table, and draw a gate circuit equivalent to that expression:

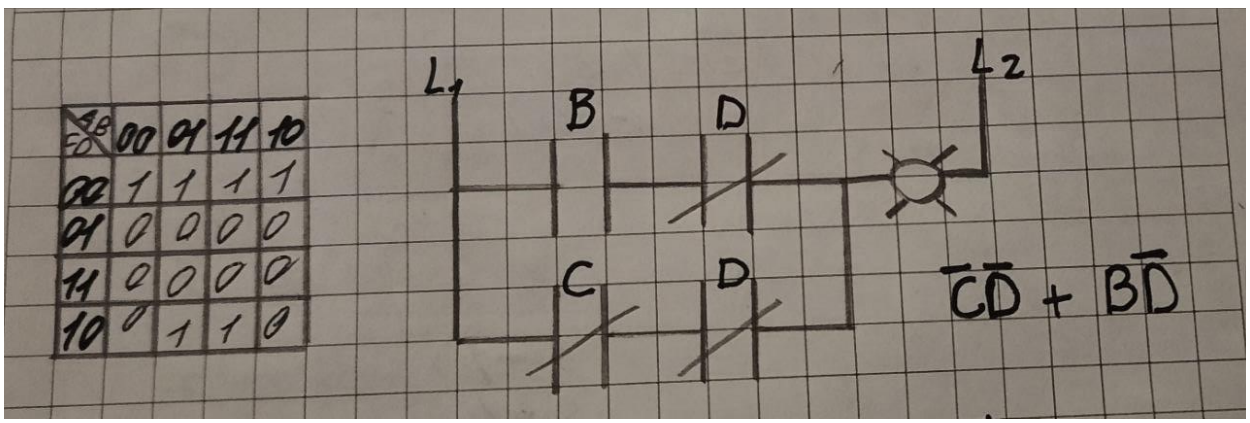
A	B	C	D	Output
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1



Question 16

Use a Karnaugh map to generate a simple Boolean expression for this truth table, and draw a relay circuit equivalent to that expression:

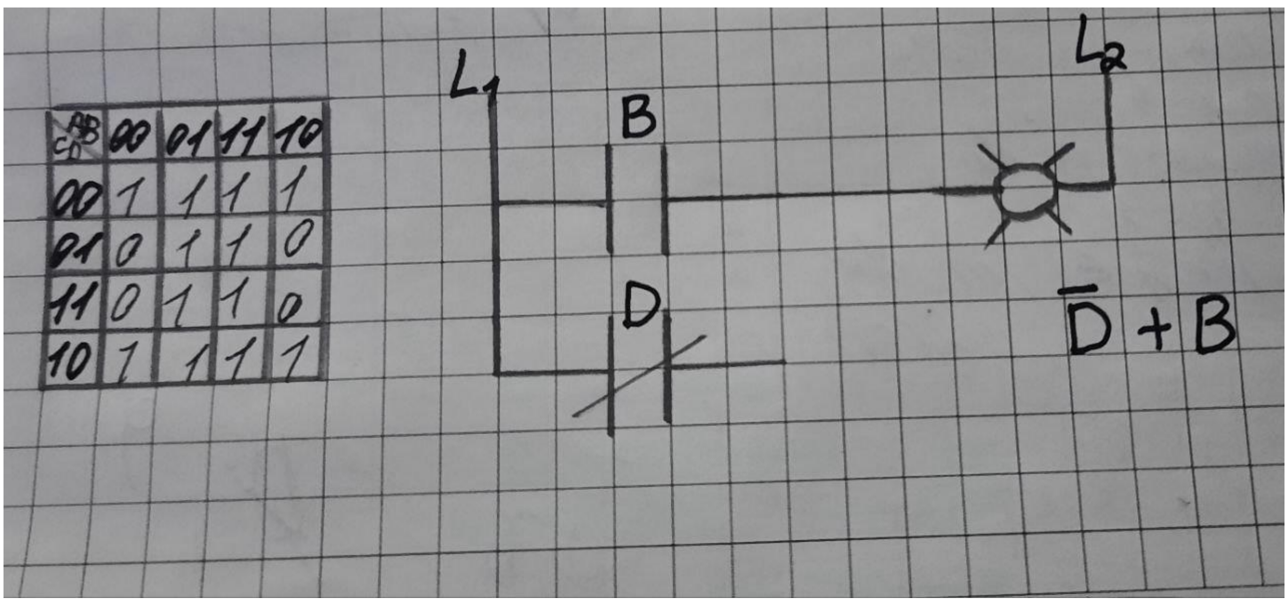
A	B	C	D	Output
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0



Question 17

Use a Karnaugh map to generate a simple Boolean expression for this truth table, and draw a relay circuit equivalent to that expression:

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



Conclusions: After this laboratory work, I managed to exercise and become familiar with Boolean algebra and Karnaugh mapping. Karnaugh Diagrams or Karnaugh Maps are a concise way to represent a Boolean function with n variables, a way to minimize the digital logic. Throughout the exercises, I saw why this method is useful, when it is used and all rules related to the Karnaugh mapping. Even though this theme was studied before, it needed a more in-depth revise.