Exercise 1.

Using the truth table, prove the following equalities.

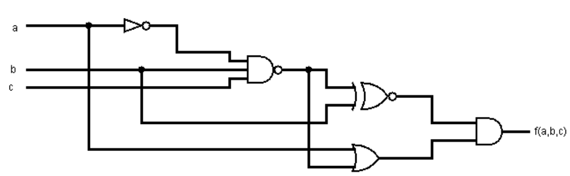
$$a + \overline{a} \cdot b = a + b, \; a \oplus b = \overline{a} \cdot b + a \cdot \overline{b}, \; \overline{a \oplus b} = a \cdot b + \overline{a} \cdot \overline{b}.$$

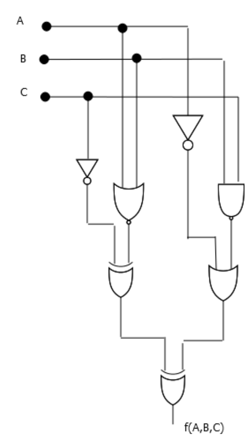
Exercise 2.

1. *How would you hardwave-implement a four-input OR gate using only two-input OR gates ?*
2. *How would you hardwave-implement a four-input AND gate using only two-input AND gates ?*
3. *How dou you implement three-input EX-OR logic functions with the help of two-input EX-OR gates ?*
4. *How can you implement a NOT circuit using a two-input EX-OR gates ?*
5. *How do you implement a three-input EX-NOR function using only two-input EX-NOR gates ?*

Exercise 3.

1. *Give the output-logical function expressions of the figures bellow*





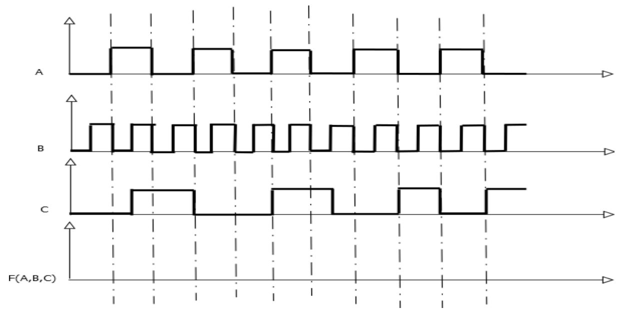
2. *Give the truth table of each boolean function.*

Exercise 4.

Let us define a circuit by the following logical function

$$F(A, B, C) = \overline{A \oplus B} + C.$$

1. *Give the graphic representation of the circuit.*
2. *Describe the output waveform for the input signals given in the figure bellow.*



3. *Give the corresponding truth table*

Exercise 5. 1. *Find the dual of* $a \cdot b \cdot c \cdot \overline{d} + a \cdot \overline{b} \cdot \overline{c} \cdot d + \overline{a} \cdot \overline{b} \cdot \overline{c} \cdot \overline{d}$.

2. *Find the complement of* $a + [(b + \overline{c}) \cdot d + \overline{e}] \cdot f$.

Exercise 6. *Simplify the following Boolean expressions.*

1. $S = \overline{a} \cdot b \cdot (a + \overline{b} + c)$.
2. $S = \overline{a} \cdot \overline{b} \cdot \overline{c} \cdot \overline{d} + a \cdot \overline{b} \cdot \overline{c} \cdot \overline{d} + a \cdot b \cdot \overline{c} \cdot \overline{d} + a \cdot b \cdot \overline{c} \cdot \overline{d}$.
3. $S = a + b \cdot \overline{c} + \overline{a} \cdot (\overline{b \cdot \overline{c}}) \cdot (a \cdot d + b)$.
4. $S = (a \oplus b) \cdot b + a \cdot b$.

Exercise 7.

Let us define a circuit by the truth table bellow.

1. *Give the logical expression of the output* Y.
   — *In the form of sum of products.*
   — *In the form of product of sums*
2. *Simplify the two expressions using the theorems of Boolean algebra.*

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

Exercise 8.

Let us consider the following logic functions

$$f(a,b,c,d) = a.b + \overline{a}.\overline{c}.\overline{d} \text{ and } g(a,b,c,d) = (\overline{a} + \overline{b} + \overline{c}).(a+d).$$

— *Give a logic circuit based on 2-input NAND gates and a logic circuit based on 2-input NOR gates for each of these functions.*

Exercise 9.

We define a logical function f by the following truth table.

1. *Write the disjunctif canonical form of the output.*
2. *Using theorems and laws of Boolean algebra to simplify the logical expression.*
3. *Give a logic circuit, using only 2-input NAND gates to implement the function f.*

a	b	c	$f(a,b,c)$
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Exercise 10. *Let $f$be a boolean function defined by the truth table bellow.*

1. *Represent f by his disjunctive canonical form. (First canonical form)*
2. *Use the Karnaugh maping method to obtain the simplified expression of f.*
3. *Implement the logical function f, using only 2-input NAND gates.*

x	y	z	$f(x,y,z)$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Exercise 11.

Le us give the boolean function defined by

$$f(a,b,c) = \prod(1,4,6)$$

.

1. *Write function's expression as product of sums and sum of products.*
2. *Let us define the Boolean function g by $g(a,b,c) = a.b + \bar{c}$.*
   — *Write minterm and maxterm expressions of f.*
3. *Simplify the Boolean functions f and g using the Karnaugh mapping method*
4. *Using the Karnaugh mapping method, simplify the Boolean function h, defined by $h(a,b,c,d) = \sum(0,3,4,6,7,8,9,10,12,14,15) + \sum_{\varphi}(1,2,5,11,13,15)$.*

Exercise 12.

c \ ab	00	01	11	10
0	1	1	1	1
1	1	0	0	1

c \ ab	00	01	11	10
0	1	x	1	x
1	1	0	1	1

cd \ ab	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

cd \ ab	00	01	11	10
00	0	1	1	0
01	0	0	0	0
11	1	0	0	1
10	1	1	1	1

cd \ ab	00	01	11	10
00	1	1	0	1
01	0	0	x	0
11	1	x	0	1
10	1	1	x	1

cd \ ab	00	01	11	10
00	x	0	0	1
01	1	0	0	1
11	x	0	0	0
10	1	1	x	1