Ministry of Science and Education in Ukraine

National Aviation University

ERMIT

Laboratory Work #1

Research logical functions of one and two variables

Prepared by:

Group:

Kiev-2020

Purpose of the laboratory work: Research logical functions of one and two variables using the program Electronics Workbench.

Theoretical information

Widespread in digital technology have received a positional number system with base 2 - binary number system. In this number system uses only two digits 0 and 1. In order to describe the behavior and structure of a digital circuit, its input and output signals, the States of the internal nodes set in accordance variables that take two values (Booleans).

Many functions of *n* variables can be represented by a truth table whose rows are reserved for $N=2^n$ words of length *n*, and the columns for 2^N functions.

The dependence of the output variables, expressed in terms of a set of input variables by means of Boolean operations is called switching functions (SF) or Boolean functions. Set SF - which means define y_i for all possible combinations of variables x_i.

Functionally complete set of logical elements - is a set of elements that can be implemented using any arbitrarily complex logic function. In view of the fact that any one logic function is a combination of simple functions - disjunction, conjunction and inversion, a set of elements OR, AND, NOT are functionally complete.

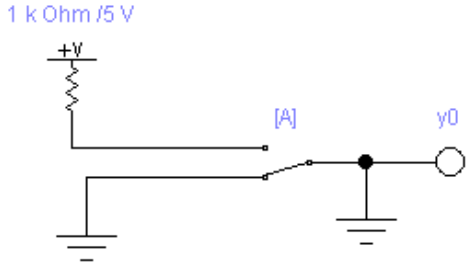
Truth table - a table describing the logic function.

A full set of logical functions of (one and) two variables:

№	Designation	Name	Logical expression
y_0	0	Constant 0	0
y_1	$x_1 x_2$	Conjunction / AND	$x_1 x_2$
y_2	$x_1 \leftarrow x_2$	Negation of implication	$x_1 \overline{x}_2$
y_3	x_1	Repeat x_1	x_1
y_4	$x_2 \leftarrow x_1$	Negation of the reverse implication	$\overline{x}_1 x_2$
y_5	x_2	Repeat x_2	x_2
y_6	$x_1 \oplus x_2$	XOR	$\overline{x}_1 x_2 \vee x_1 \overline{x}_2$
y_7	$x_1 \vee x_2$	Disjunction / OR	$x_1 \vee x_2$
y_8	$x_1 \downarrow x_2$	NOR	$\overline{x_1 \vee x_2}$
y_9	$x_1 \sim x_2$	Equivalence	$\overline{x}_1 \overline{x}_2 \vee x_1 x_2$
y_{10}	$\overline{x}_2$	Inversion / NOT	$\overline{x}_2$
y_{11}	$x_2 \rightarrow x_1$	Reverse implication	$x_1 \vee \overline{x}_2$
y_{12}	$\overline{x}_1$	Inversion / NOT x_1	$\overline{x}_1$
y_{13}	$x_1 \rightarrow x_2$	Implication	$\overline{x}_1 \vee x_2$
y_{14}	$x_1 \mid x_2$	NAND	$\overline{x_1 x_2}$
y_{15}	1	Constant 1	1

1. Constant 0:

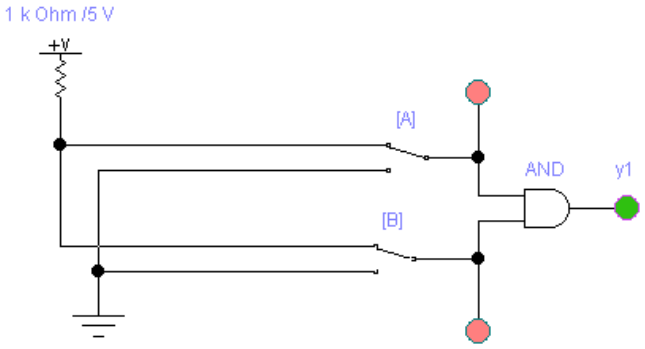
   y = 0



A	Y0
0	

2. Conjunction/AND ($x_1 x_2$):

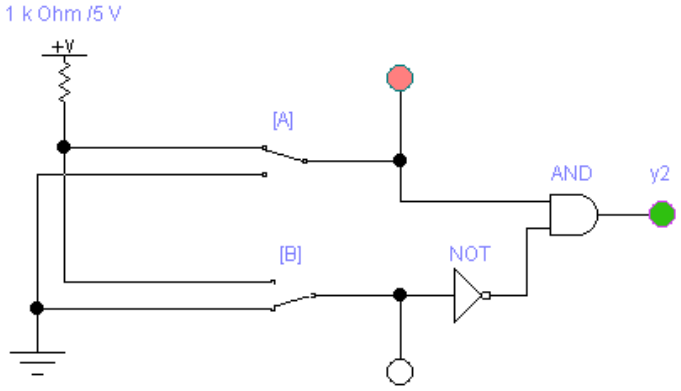
   y = $x_1 x_2$



A	B	Y1
0	0	
0	1	
1	0	
1	1	

3. Negation of implication ($x_1 \overline{x}_2$):

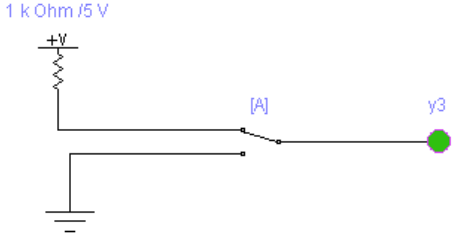
   y = $x_1 \overline{x}_2$



A	B	Y2
0	0	
0	1	
1	0	
1	1	

4. Repeat x_1:

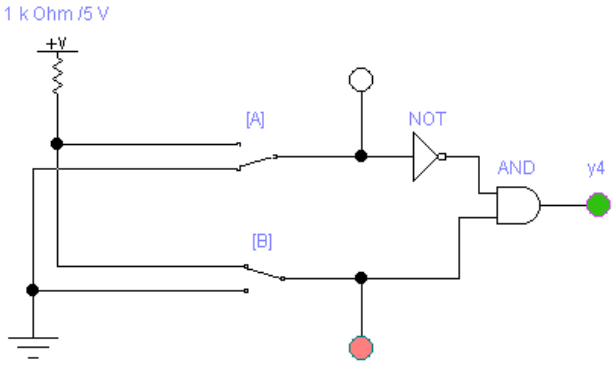
y = x_1



A	Y3
0	
1	

5. Negation of reverse implication($\bar{x}_1 x_2$):

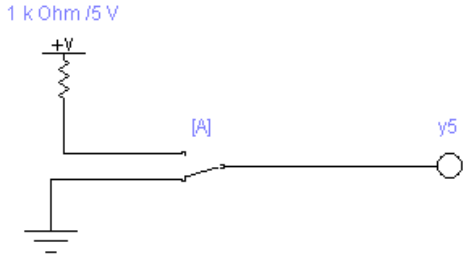
y = $\bar{x}_1 x_2$



A	B	Y4
0	0	
0	1	
1	0	
1	1	

6. Repeat x_2:

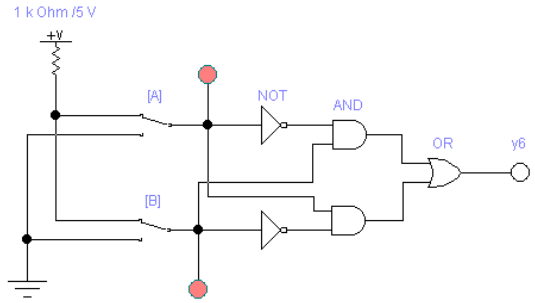
y = x_2



A	Y5
0	
1	

7. XOR ($\bar{x}_1 x_2 \vee x_1 \bar{x}_2$):

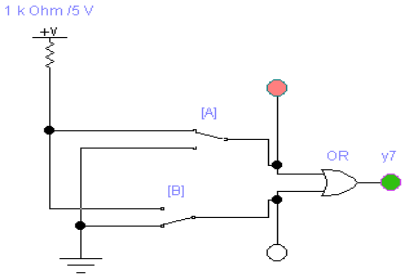
y = $\bar{x}_1 x_2 \vee x_1 \bar{x}_2$



A	B	Y6
0	0	
0	1	
1	0	
1	1	

8. Disjunction / OR ($x_1 \vee x_2$):

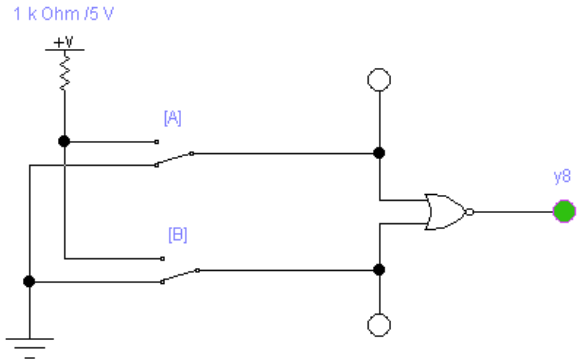
y = $x_1 \vee x_2$



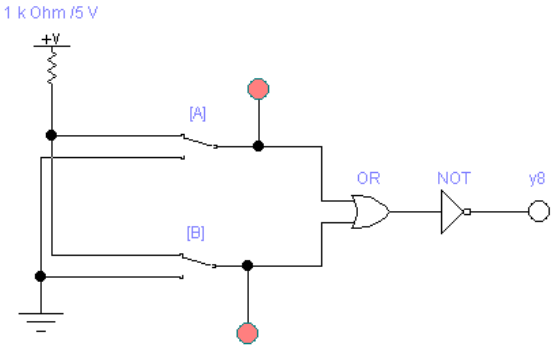
A	B	Y7
0	0	
0	1	
1	0	
1	1	

9. NOR ($\overline{x_1 \vee x_2}$):

y = $\overline{x_1 \vee x_2}$

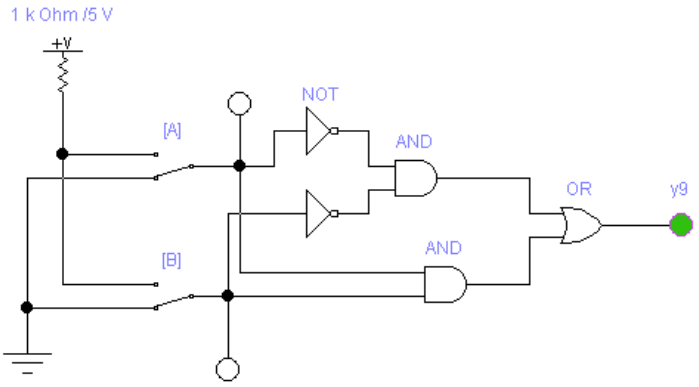


A	B	Y8
0	0	
0	1	
1	0	
1	1	



10. Equivalence ($\overline{x}_1\overline{x}_2 \vee x_1x_2$):

y = $\overline{x}_1\overline{x}_2 \vee x_1x_2$

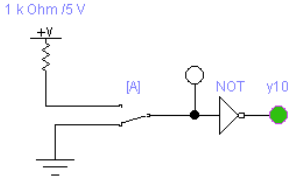


A	B	Y9
0	0	
0	1	
1	0	
1	1	

11. Inversion / NOT ($\overline{x}_2$):

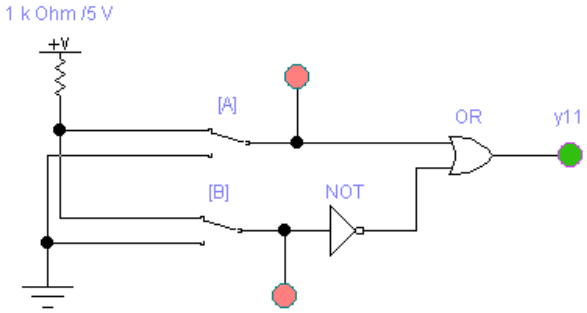
y = $\overline{x}_2$

A	Y10
1	
0	



12. Reverse implication ($x_1 \vee \bar{x}_2$):

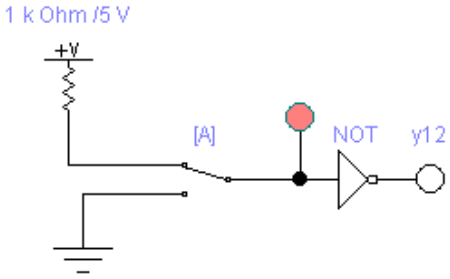
y = $x_1 \vee \bar{x}_2$



A	B	Y11
0	0	
0	1	
1	0	
1	1	

13. Inversion / NOT x_1 ($\bar{x}_1$):

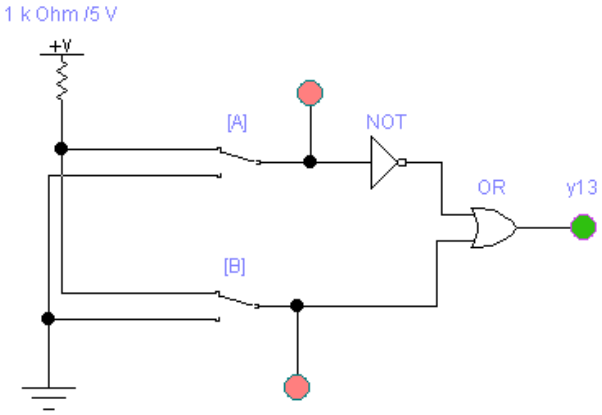
y = $\bar{x}_1$



A	Y12
0	
1	

14. Implication ($\bar{x}_1 \vee x_2$):

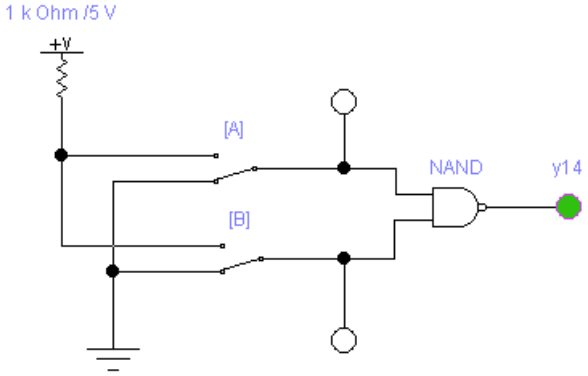
y = $\bar{x}_1 \vee x_2$



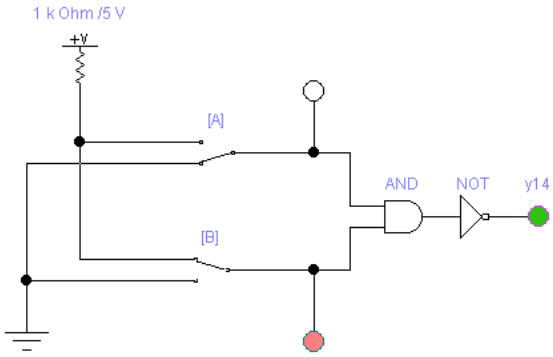
A	B	Y13
0	0	1
0	1	0
1	0	1
1	1	1

15. NAND ($\overline{x_1 x_2}$):

y = $\overline{x_1 x_2}$



A	B	Y14
0	0	0
0	1	1
1	0	1
1	1	1



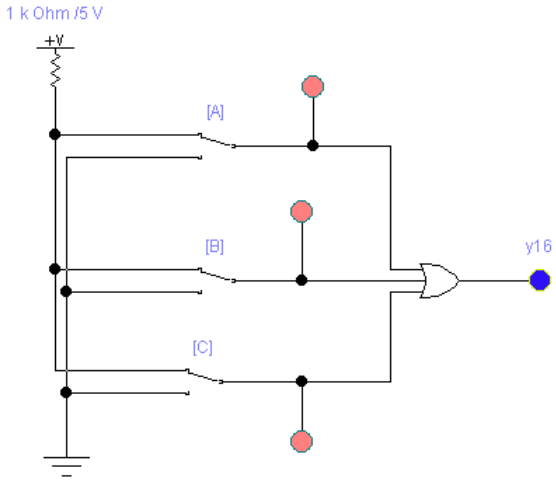
16. Constant 1:

y = 1

A	Y15
1	

17. Three-way OR ($x_1 \vee x_2 \vee x_3$):

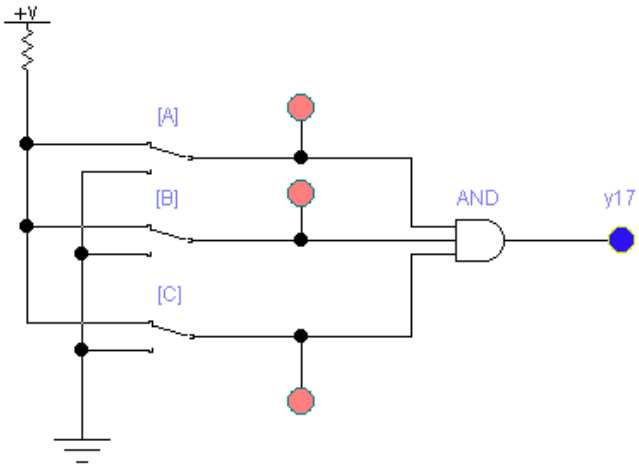
y = $x_1 \vee x_2 \vee x_3$



A	B	C	Y16
0	0	0	
1	0	0	
0	1	0	
0	0	1	
1	1	0	
0	1	1	
1	0	1	
1	1	1	

18. Three-way AND ($x_1 x_2 x_3$):

y = $x_1 x_2 x_3$



A	B	C	Y17
0	0	0	
1	0	0	
0	1	0	
0	0	1	
1	1	0	
0	1	1	
1	0	1	
1	1	1	

CONCLUSION:

Laboratory Work #2

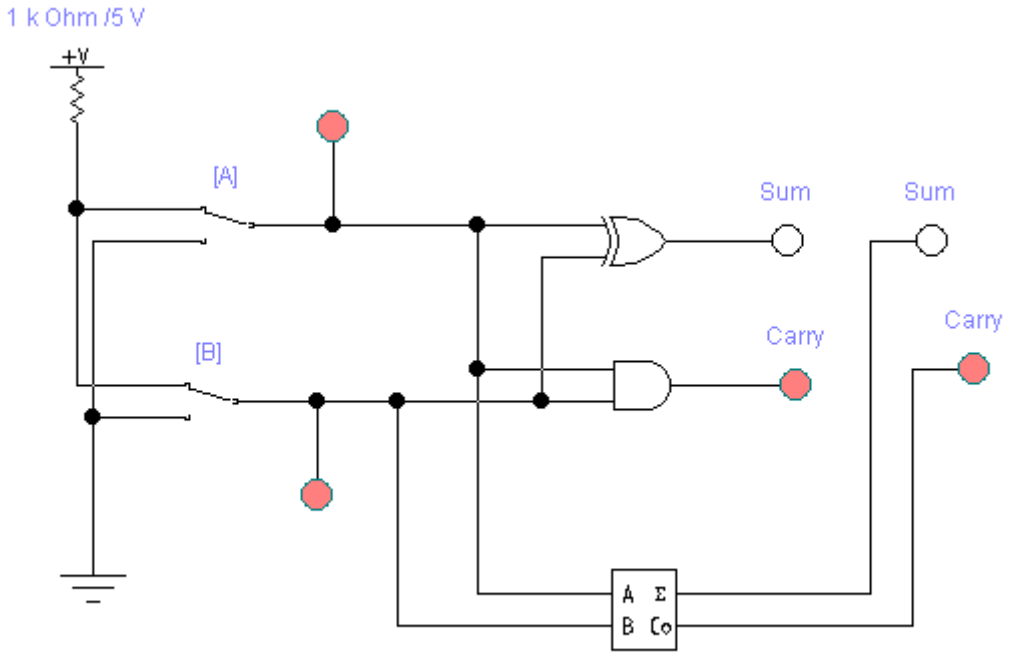
Adders

Aim of the laboratory work: to construct and study adders.

Main theoretical information:

Adder is a complex logical unit that uses for arithmetic addition of numbers that are represented in binary code. Adder makes addition bitwise accounts the transfer from least significant digit, result of addition and transfer to the most significant digit. According to numbers of inputs there are: half-adders and full-adders.

1. Assemble the half-adder and check its work, making its truth table.

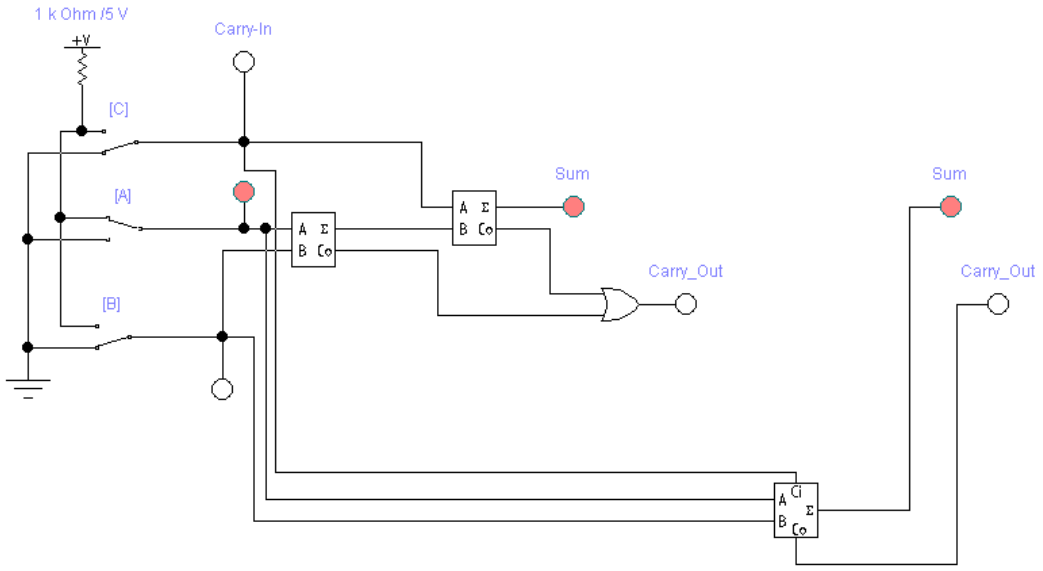


A	B	Sum	Carry
0	0		
0	1		
1	0		
1	1		

Sum = $\overline{A}$B + A $\overline{B}$

Carry = AB

2. Assemble the single-bit adder and check its work, making its truth table.

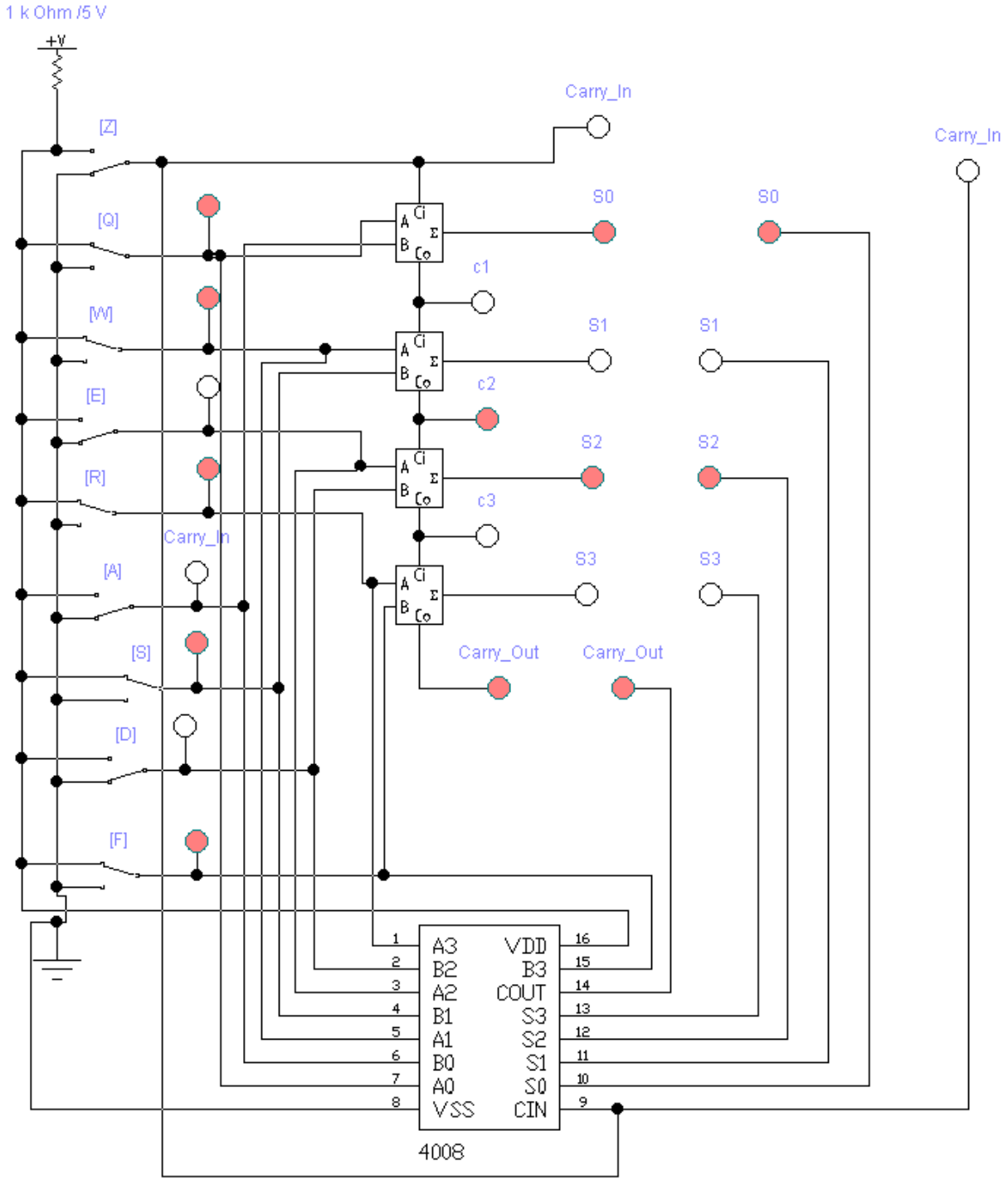


A	B	C	Sum	Carry Out
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

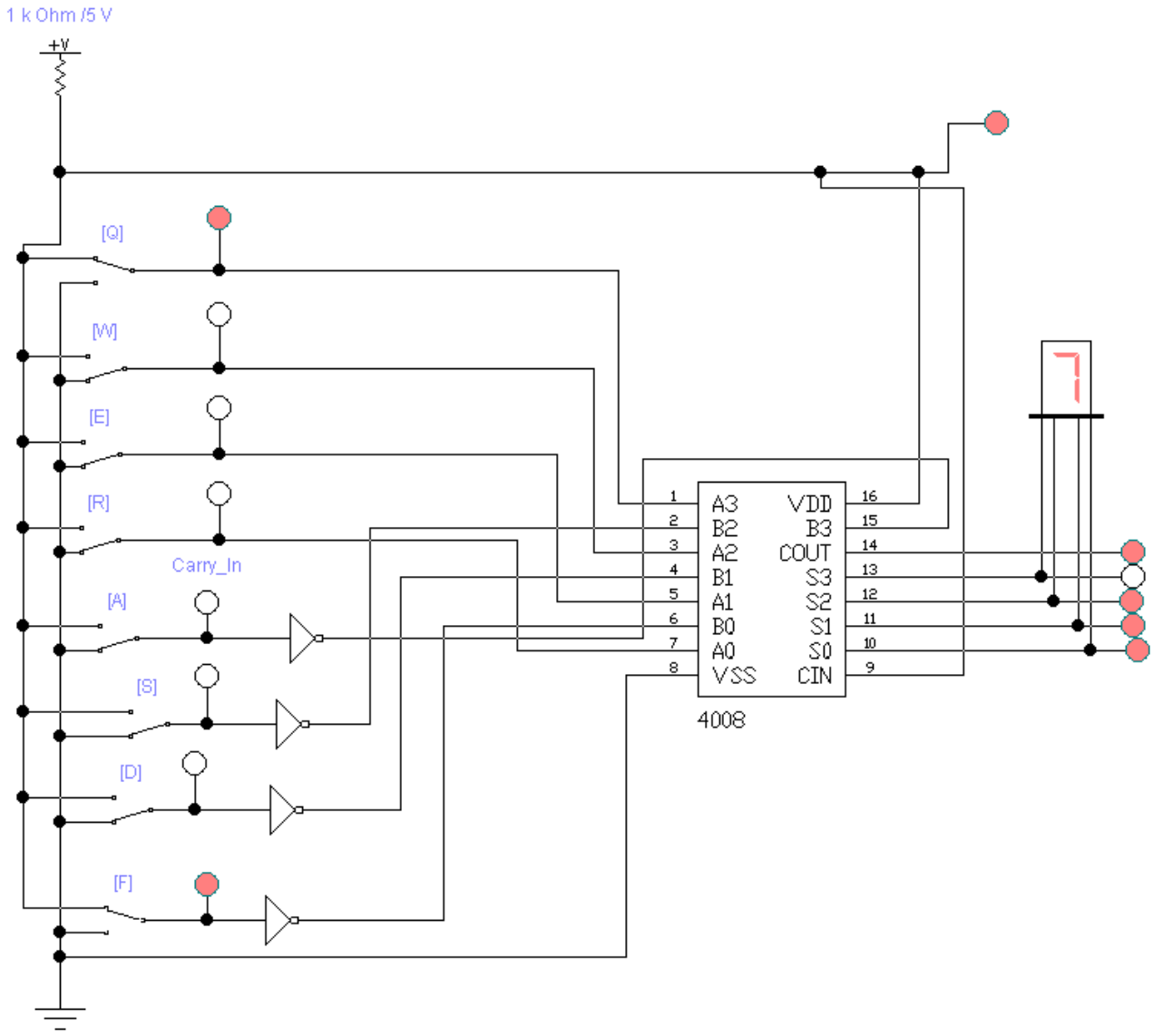
Sum = $\overline{A}\overline{B}C + A\overline{B}\overline{C} + \overline{A}B\overline{C} + ABC$

Carry = $AB\overline{C} + \overline{A}BC + A\overline{B}C + ABC$

3. Assemble four-bit adder and find the sum of the two four-bit numbers A and B.

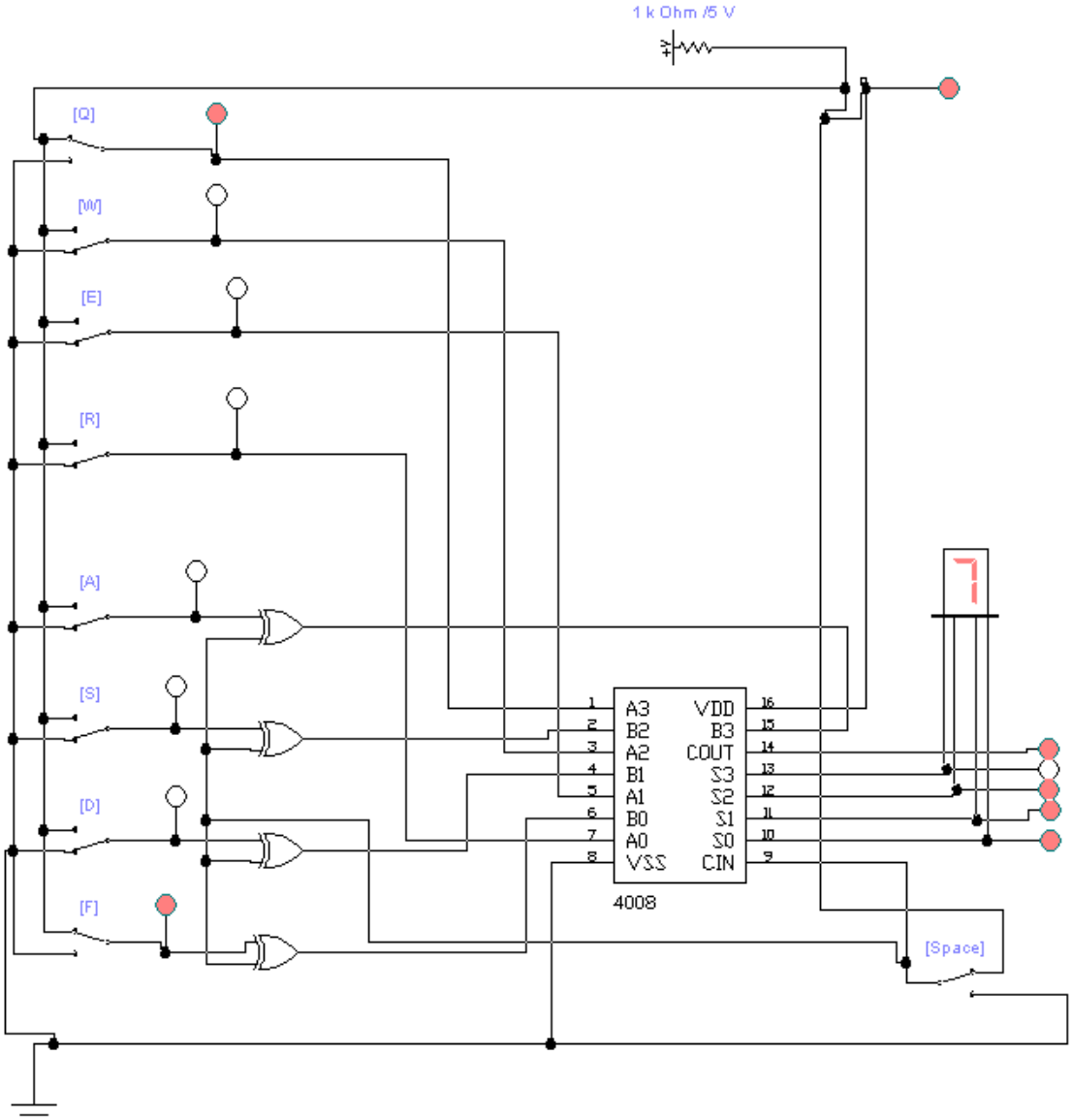


4. Use four-bit adder find the difference between the four-bit numbers A and B

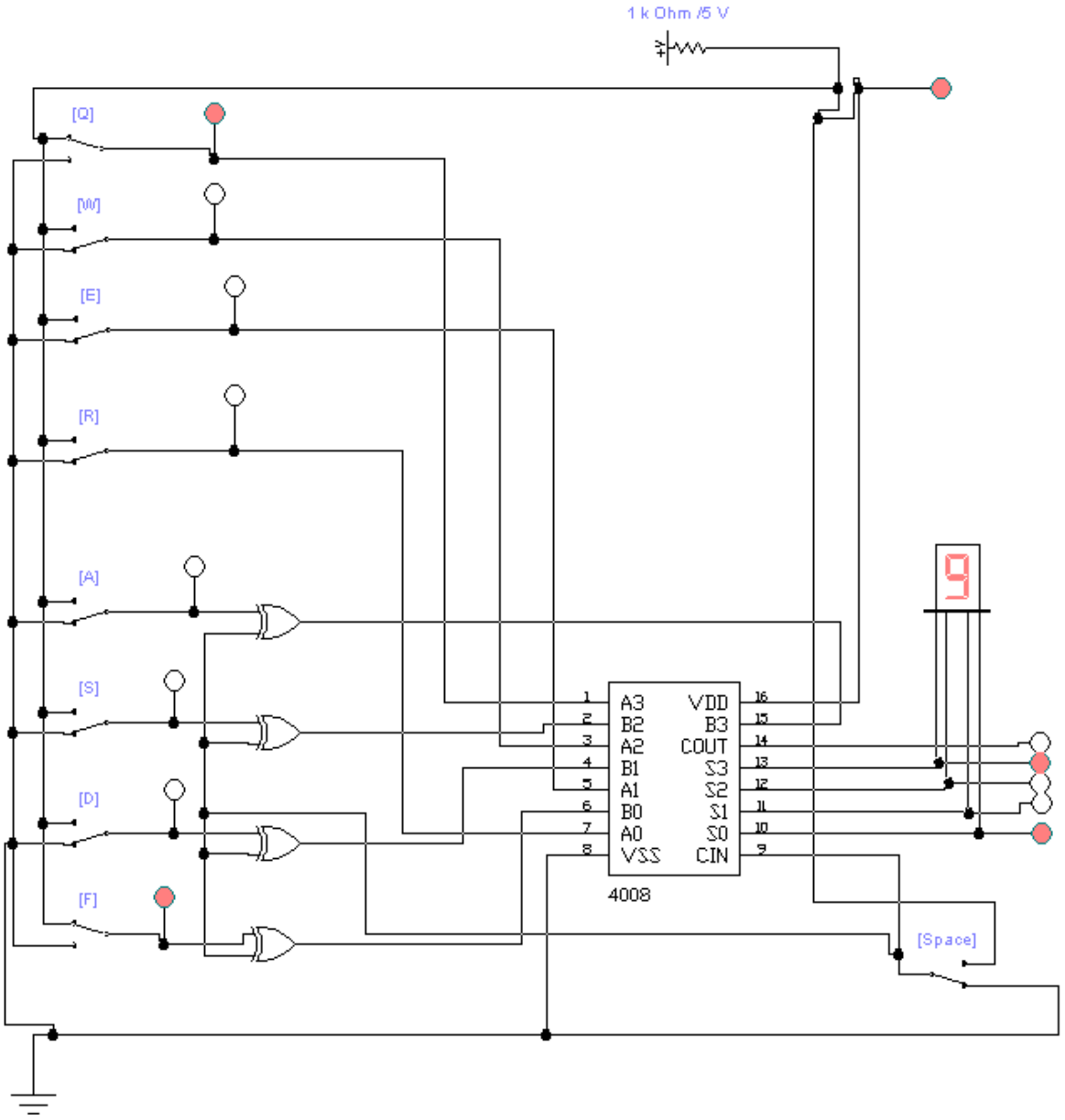


5. Assemble the universal scheme for the addition and subtraction of four-bit numbers A and B.

Subtraction:



Addition:



CONCLUSION

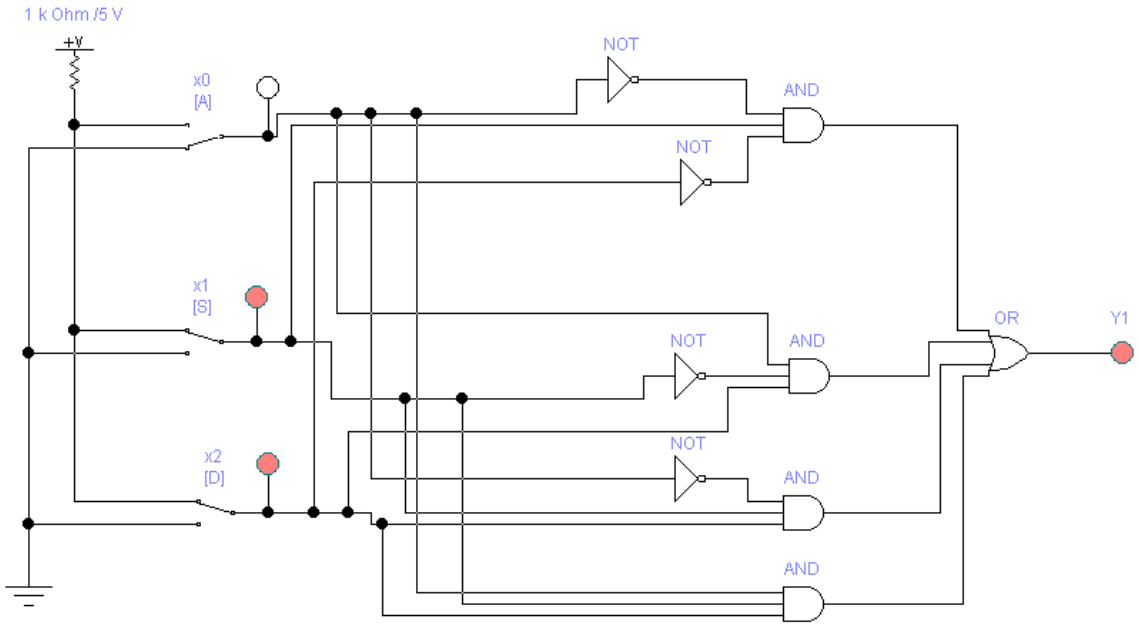
Laboratory Work #3

Design of single-output combinational circuits

Exampleof the table:

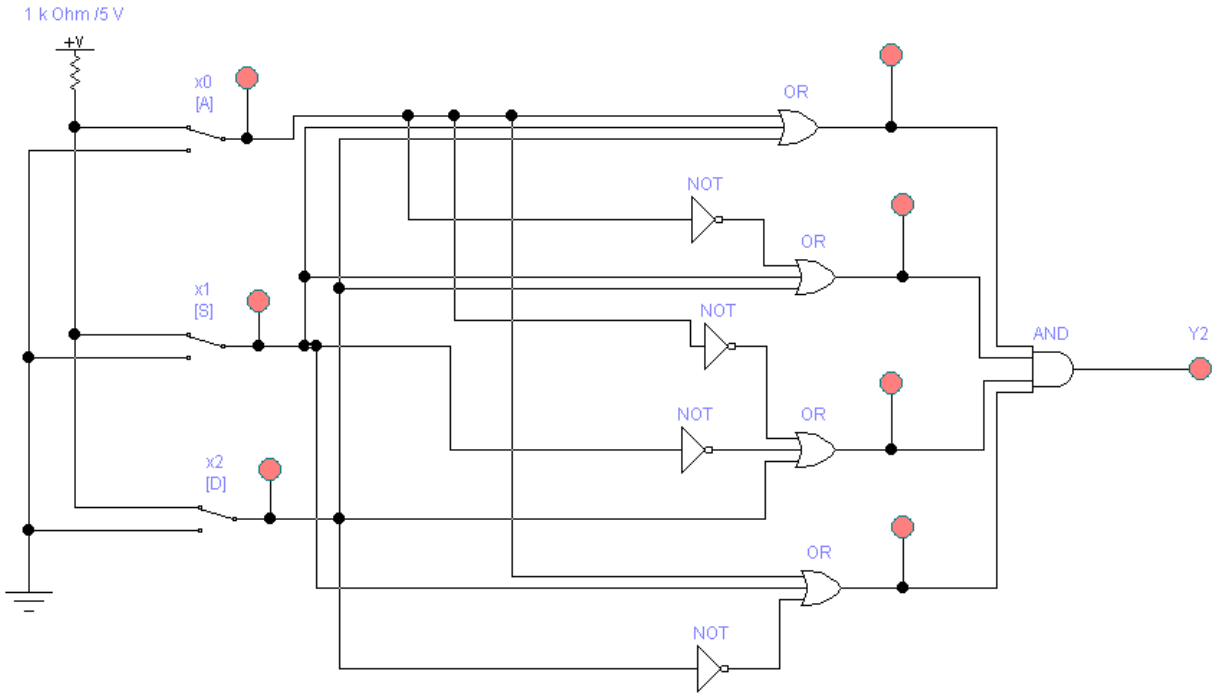
N	X_2	X_1	X_0	y
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

1. **Write down canonical sum of products form (CSPF) and draw a network of the expression.**
   CSPF: y = $(\overline{x_2}\, x_1\, \overline{x_0}) \vee (x_2 \overline{x_1} x_0) \vee (x_2 x_1 \overline{x_0}) \vee (x_2 x_1 x_0)$•



2. **Write down canonical product of sums form (CPSF) and draw a network of the expression.**

CPSF: $y = (x_2 \vee x_1 \vee x_0) \bullet (x_2 \vee x_1 \vee \overline{x_0}) \bullet (x_2 \vee \overline{x_1} \vee \overline{x_0}) \bullet \overline{(x_2} \vee x_1 \vee x_0)$

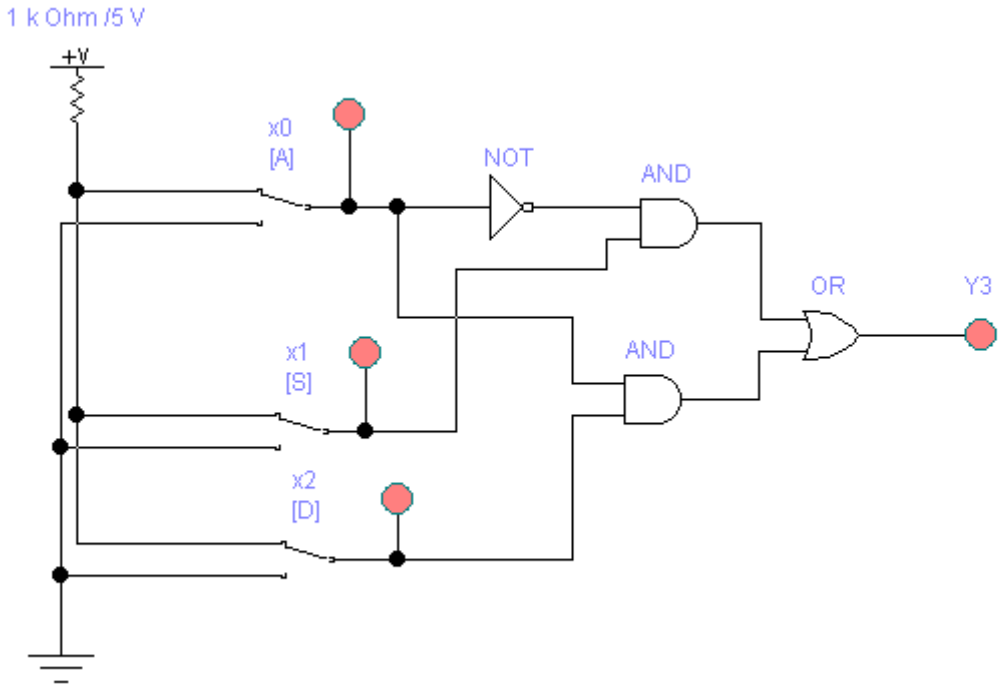


3. **Write down minimal sum of products form (MSPF) and draw a network of the expression.**

x_0 \ x_2x_1	00	01	11	10
0	1	1	0
0	0	1	1

Using groups of 1's, I created a MSPF:

MSPF: y = $(x_1\overline{x_0}) \vee (x_2x_0)$

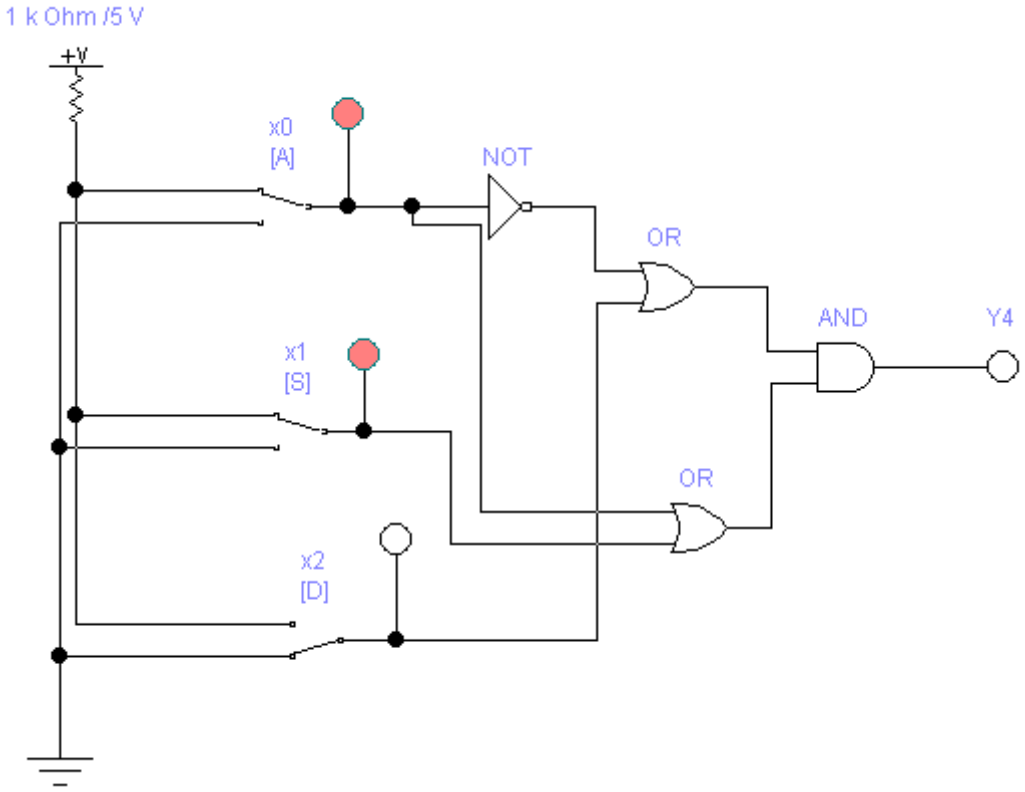


4. Write down minimal product of sums form (MPSF) and draw a network of the expression.

x_0 \ x_2x_1	00	01	11	10
	0	1	1	0
	0	0	1	1

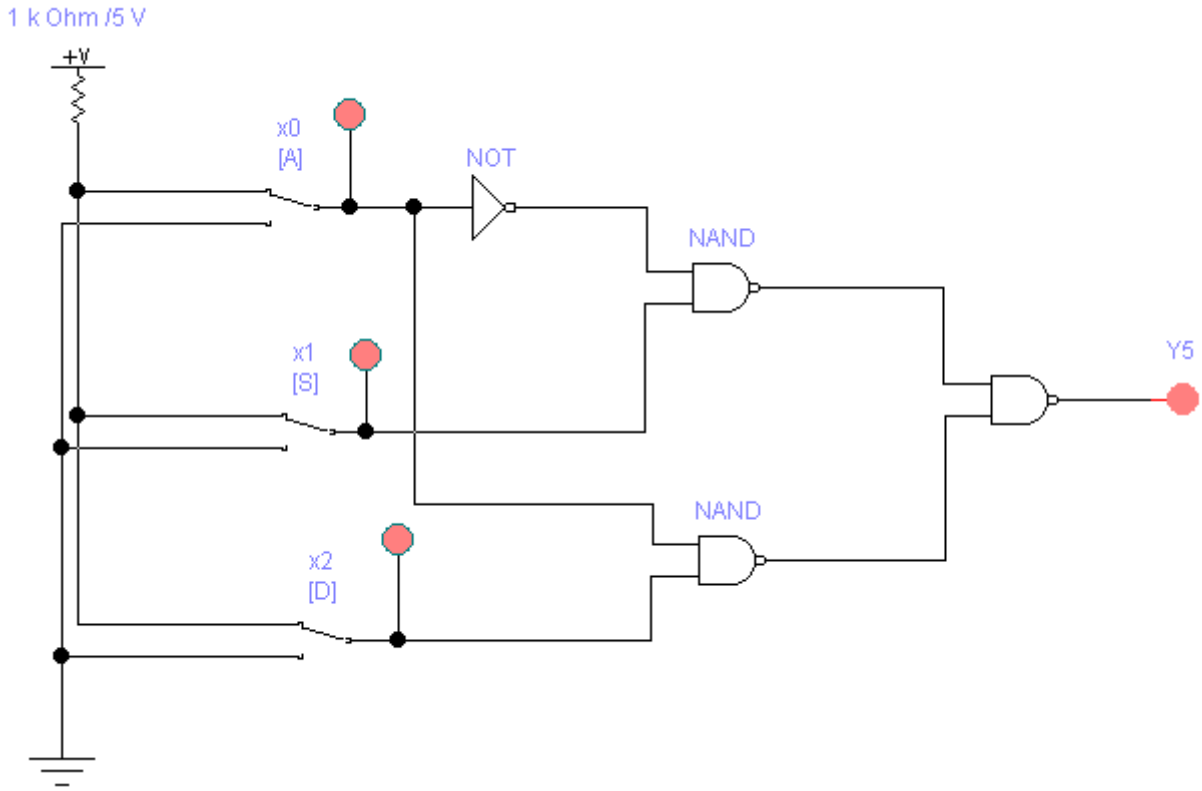
Using 0's groups, I created a MPSF:

MPSF: y = $(x_2 \vee \overline{x_0}) \bullet (x_1 \vee x_0)$



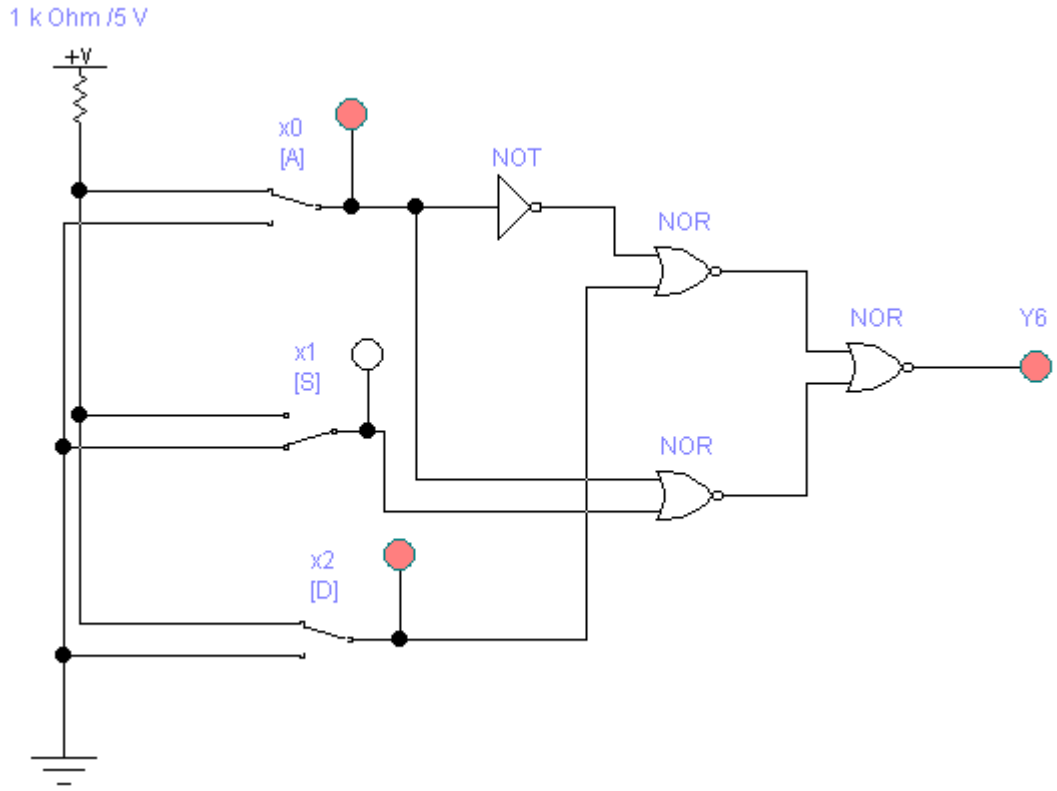
5. Applying de Morgan's law write down NAND form and draw a network of the expression.

$y = \overline{(x_1\overline{x_0}) \vee (x_2x_0)} = \left(\overline{x_1\overline{x_0}}\right) * (\overline{x_2x_0})$



6. Applying de Morgan's law write down NOR form and draw a network of the expression.

$$y = \overline{(x_2 \vee \overline{x_0}) \bullet (x_1 \vee x_0)} = \overline{(x_2 \vee \overline{x_0})} \vee \overline{(x_1 \vee x_0)}$$



CONCLUSION: the most suitable and easy in constructing network and equation is minimal sum of products form or minimal product of sum s form.

Laboratory Work 4

"The study of multiple-output combinational networks"

1. Design the code converter from the binary code to Gray code.

Truth table:

	Binary			Gray		
N	**X_2**	**X_1**	**X_0**	**Y_2**	**Y_1**	**Y_0**
0	0	0	0	0	0	0
1	0	0	1	0	0	1
2	0	1	0	0	1	1
3	0	1	1	0	1	0
4	1	0	0	1	1	0
5	1	0	1	1	1	1
6	1	1	0	1	0	1
7	1	1	1	1	0	0

Carnough maps:

For Y_2:

X_0 / X_2X_1	**00**	**01**	**11**	**10**
0	0	0	1	1
1	0	0	1	1

When we apply the method of uniting highlight units we'll get:

$$Y_2 = X_2$$

For Y_1:

X_0 / X_2X_1	00	01	11	10
0	0	1	0	1
1	0	1	0	1

When we apply the method of uniting highlight units we'll get:

$$Y_1 = (\overline{X_2} X_1) \vee (X_2 \overline{X_1})$$

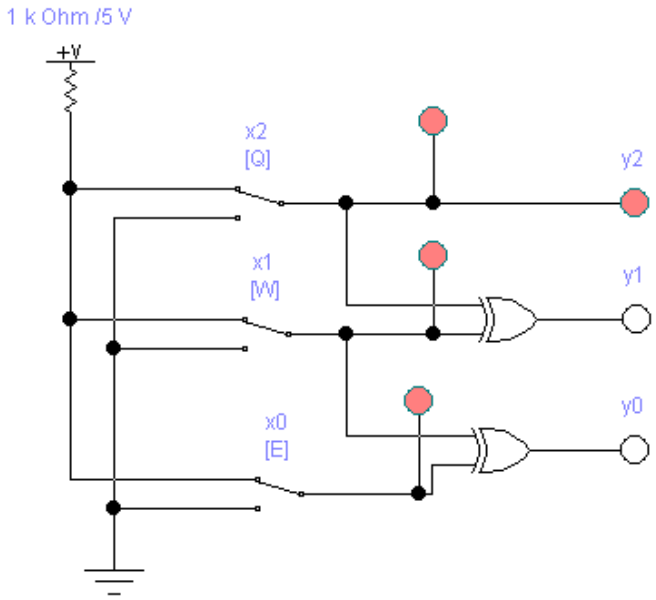
For Y_0:

X_0 / X_2X_1	00	01	11	10
0	0	1	1	0
1	1	0	0	1

When we apply the method of uniting highlight units we'll get:

$$Y_0 = (\overline{X_0} X_1) \vee (X_0 \overline{X_1})$$

The circuit is following:



2. **Design the code converter from Gray code to the binary code.**

Truth table:

	Gray			**Binary**		
N	**X_2**	**X_1**	**X_0**	**Y_2**	**Y_1**	**Y_0**
0	0	0	0	0	0	0
1	0	0	1	0	0	1
2	0	1	1	0	1	0
3	0	1	0	0	1	1
4	1	1	0	1	0	0
5	1	1	1	1	0	1
6	1	0	1	1	1	0
7	1	0	0	1	1	1

Carnough maps:

For Y_2:

X_0 / X_2X_1	**00**	**01**	**11**	**10**
0	0	0	1	1
1	0	0	1	1

When we apply the method of uniting highlight units we'll get:

$$\mathbf{Y_2 = X_2}$$

For Y_1:

X_0 / X_2X_1	00	01	11	10
0	0	1	0	1
1	0	1	0	1

When we apply the method of uniting highlight units we'll get:

$$\mathbf{Y_1} = (\overline{X_2}X_1) \vee (X_2\overline{X_1})$$

For Y_0:

X_0 / X_2X_1	00	01	11	10
0	0	1	0	1
1	1	0	1	0

Here we can't unite units, so simplification will be much harder:

$\mathbf{Y_0}$ = $(X_2X_1X_0) \vee (\overline{X_2}X_1\overline{X_0}) \vee (\overline{X_2}\,\overline{X_1}X_0) \vee (X_2\overline{X_1}\,\overline{X_0}) \vee (\overline{X_2}\,\overline{X_1}X_0) \vee (X_2\overline{X_1}\,\overline{X_0}) \vee (X_2X_1X_0) \vee (\overline{X_2}X_1\overline{X_0})$ Using rule: X + X = X, we'll get:

$$(X_2X_1X_0) \vee (\overline{X_2}X_1\overline{X_0}) \vee (\overline{X_2}\,\overline{X_1}X_0) \vee (X_2\overline{X_1}\,\overline{X_0}) = X_2(X_1X_0 \vee \overline{X_1}\,\overline{X_0}) \vee \overline{X_2}(X_1\overline{X_0} \vee \overline{X_1}X_0) =$$
$$= X_2(X_1 \sim X_0) \vee \overline{X_2}(X_1 \oplus X_0)$$

According to the equality $(X_1 \sim X_0) = (\overline{X_1 \oplus X_0})$, I'll get:

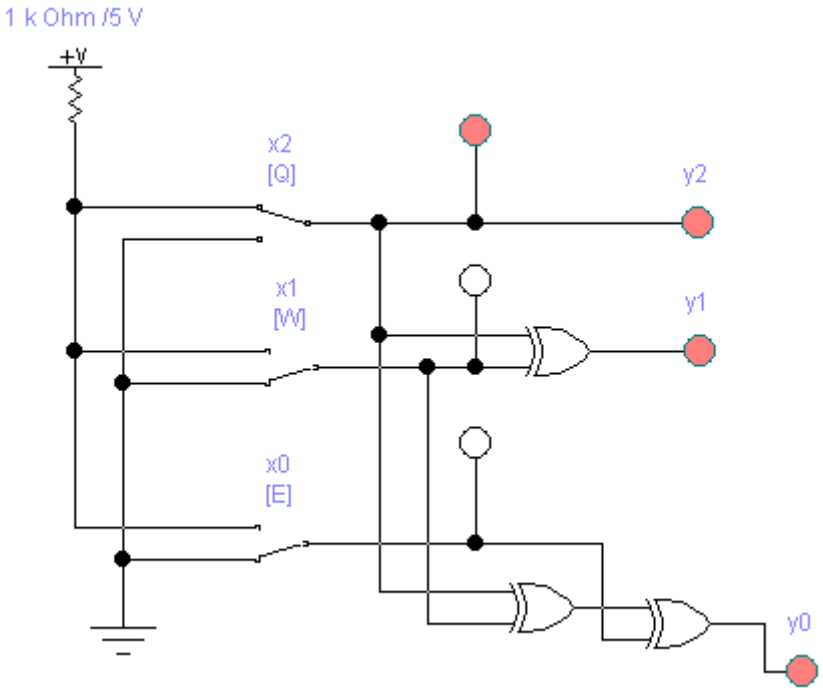
$$X_2(\overline{X_1 \oplus X_0}) \vee \overline{X_2}(X_1 \oplus X_0)$$

Let's replace $(X_1 \oplus X_0)$ with X_3 and $(\overline{X_1 \oplus X_0})$ with $\overline{X_3}$ accordingly:

$$X_2(\overline{X_3}) \vee \overline{X_2}(X_3) \Rightarrow X_2 \oplus X_3 \Rightarrow X_2 \oplus X_1 \oplus X_0$$

So, $\mathbf{Y_0}$ = $X_2 \oplus X_1 \oplus X_0$

The circuit is following:



3. Design the decoders 3-to-8 on AND and NAND logic gates.

Decoder on AND:

Truth table:

N	X_2	X_1	X_0	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
2	0	1	0	0	0	1	0	0	0	0	0
3	0	1	1	0	0	0	1	0	0	0	0
4	1	0	0	0	0	0	0	1	0	0	0
5	1	0	1	0	0	0	0	0	1	0	0
6	1	1	0	0	0	0	0	0	0	1	0
7	1	1	1	0	0	0	0	0	0	0	1

Y_0 = $\overline{X_2}\,\overline{X_1}\,\overline{X_0}$

Y_1 = $\overline{X_2}\,\overline{X_1}\,X_0$

Y_2 = $\overline{X_2}\,X_1\,\overline{X_0}$

Y_3 = $\overline{X_2}X_1X_0$

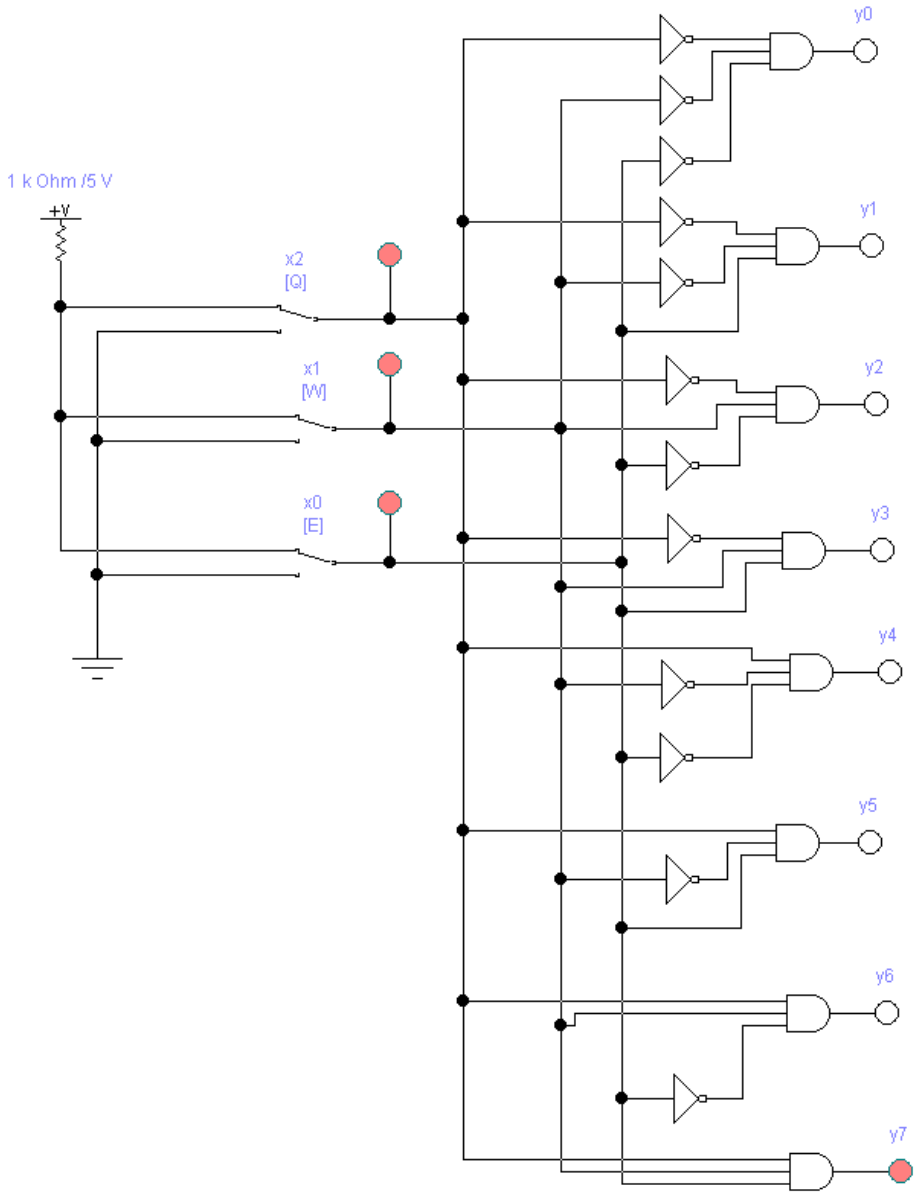
Y_4 = $X_2\overline{X_1}\overline{X_0}$

Y_5 = $X_2\overline{X_1}X_0$

Y_6 = $X_2X_1\overline{X_0}$

Y_7 = $X_2X_1X_0$

The circuit is following:



Decoder on NAND:

Truth table:

N	X_2	X_1	X_0	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	0	1	1	1	1	1	1
2	0	1	0	1	1	0	1	1	1	1	1
3	0	1	1	1	1	1	0	1	1	1	1
4	1	0	0	1	1	1	1	0	1	1	1
5	1	0	1	1	1	1	1	1	0	1	1
6	1	1	0	1	1	1	1	1	1	0	1
7	1	1	1	1	1	1	1	1	1	1	0

Y_0 = $\overline{\overline{X_2}\,\overline{X_1}\,\overline{X_0}}$

Y_1 = $\overline{\overline{X_2}\,\overline{X_1}\,X_0}$

Y_2 = $\overline{\overline{X_2}\,X_1\,\overline{X_0}}$

Y_3 = $\overline{\overline{X_2}\,X_1\,X_0}$

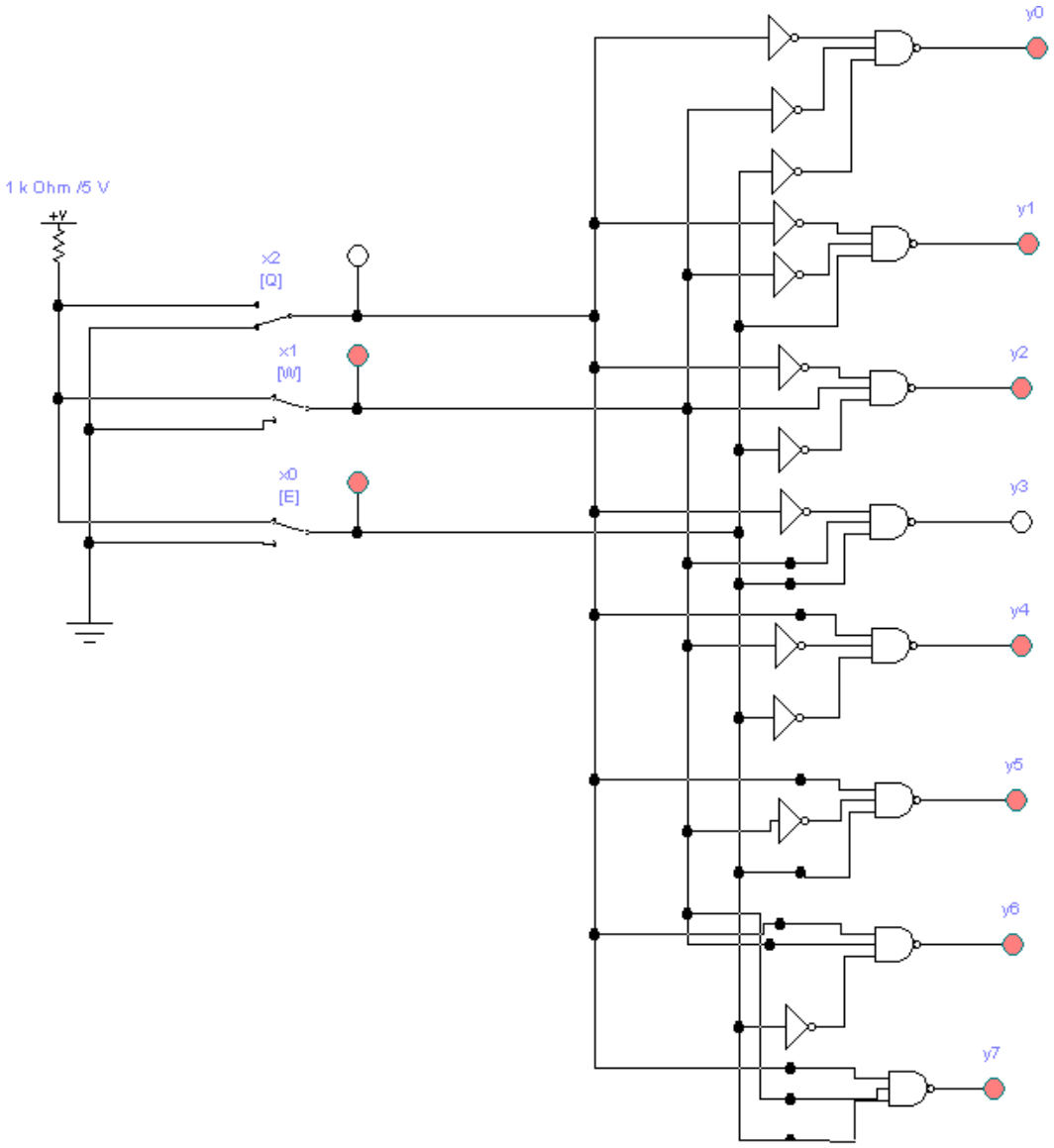
Y_4 = $\overline{X_2\,\overline{X_1}\,\overline{X_0}}$

Y_5 = $\overline{X_2\,\overline{X_1}\,X_0}$

Y_6 = $\overline{X_2\,X_1\,\overline{X_0}}$

Y_7 = $\overline{X_2\,X_1\,X_0}$

The circuit is following:



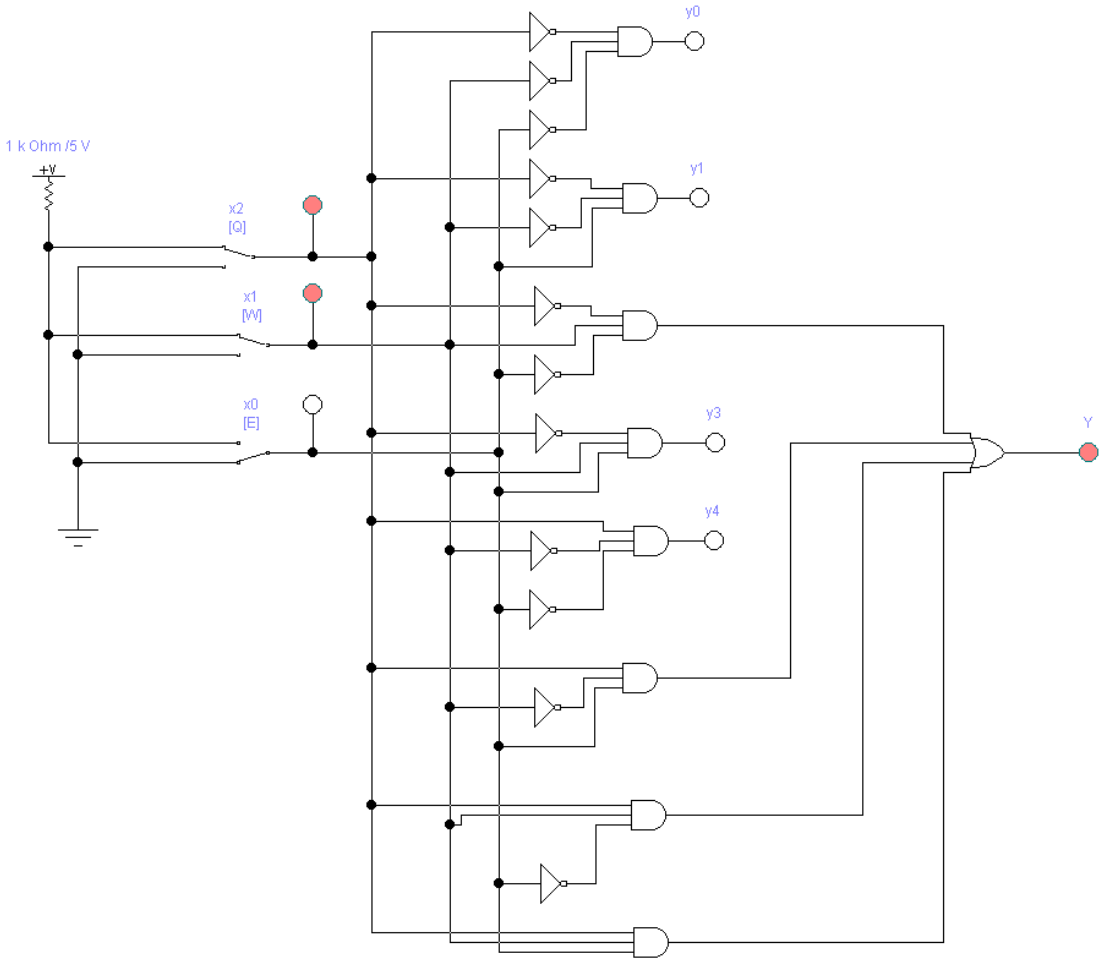
4. **Realize function from laboratory work #3 on decoders 3-to-8 from the previous task.**

N	X_2	X_1	X_0	Y
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Using minterms m_2, m_5, m_6, m_7:

$$\mathbf{Y} = \left(\overline{x_2}\, x_1\, \overline{x_0}\right) \vee \left(x_2\, \overline{x_1}\, x_0\right) \vee \left(x_2\, x_1\, \overline{x_0}\right) \vee \left(x_2\, x_1\, x_0\right)$$

The circuit is following:



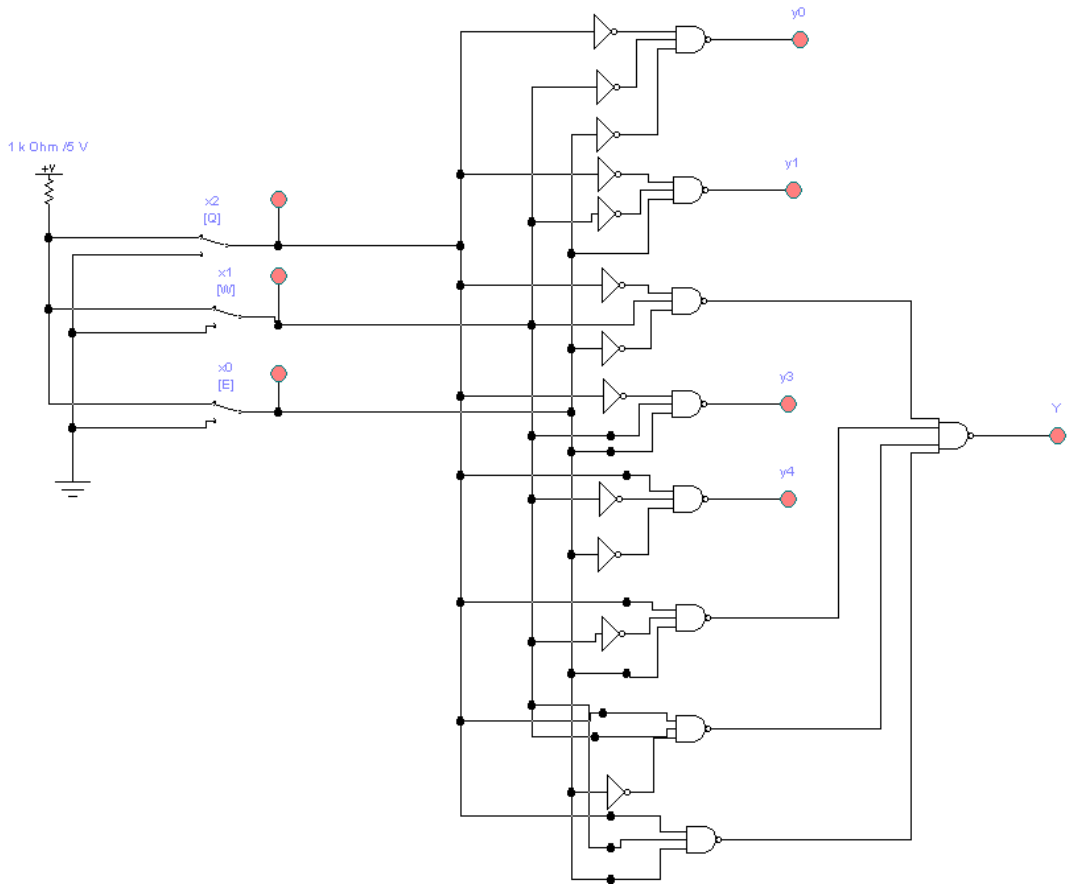
2nd variant:

Truth table:

N	X_2	X_1	X_0	Y
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Y = $\overline{\left(\overline{\overline{X_2}\,X_1\,\overline{X_0}}\right)\bullet\left(\overline{X_2\,\overline{X_1}\,X_0}\right)\bullet\left(\overline{X_2\,X_1\,\overline{X_0}}\right)\bullet\left(\overline{X_2\,X_1\,X_0}\right)}$

The circuit is following:



5. **Design decoder to 7-segment indicator. Connect it to 7-segment indicator.**

Truth table:

N	X_3	X_2	X_1	X_0	A	B	C	D	E	F	G
0	**0**	**0**	**0**	**0**	1	1	1	1	1	1	0
1	**0**	**0**	**0**	**1**	0	1	1	0	0	0	0
2	**0**	**0**	**1**	**0**	1	1	0	1	1	0	1
3	**0**	**0**	**1**	**1**	1	1	1	1	0	0	1
4	**0**	**1**	**0**	**0**	0	1	1	0	0	1	1
5	**0**	**1**	**0**	**1**	1	0	1	1	0	1	1
6	**0**	**1**	**1**	**0**	1	0	1	1	1	1	1
7	**0**	**1**	**1**	**1**	1	1	1	0	0	0	0
8	**1**	**0**	**0**	**0**	1	1	1	1	1	1	1
9	**1**	**0**	**0**	**1**	1	1	1	1	0	1	1

For **A**:

X_1X_0 | X_3X_2	00	01	11	10
00	1	0	-	1
01	0	1	-	1
11	1	1	-	-
10	1	1	-	-

A = $\left(\overline{X_2}\,\overline{X_1}\,\overline{X_0}\right) \vee \left(X_3\right) \vee \left(X_2\,X_0\right) \vee \left(X_1\right)$

For **B**:

X_1X_0 | X_3X_2	00	01	11	10
00	1	1	-	1
01	1	0	-	1
11	1	1	-	-
10	1	0	-	-

B = $(x_3)\vee(\overline{x_1}\,\overline{x_0})\vee(x_1x_0)\vee(\overline{x_3}\,\overline{x_2})=(x_3)\vee(\overline{x_1\oplus x_0})\vee(\overline{x_3}\,\overline{x_2})$

For **C**:

X_1X_0 | X_3X_2	00	01	11	10
00	1	1	-	1
01	1	1	-	1
11	1	1	-	-
10	0	1	-	-

C = $(\overline{x_1})\vee(x_2x_1)\vee(x_1x_0)$

For **D**:

X_1X_0 | X_3X_2	00	01	11	10
00	1	0	-	1
01	0	1	-	1
11	1	0	-	-
10	1	1	-	-

D = $(\overline{x_2}\,\overline{x_1}\,\overline{x_0})\vee(x_3\overline{x_2}\,\overline{x_1})\vee(x_2\overline{x_1}x_0)\vee(\overline{x_3}\,\overline{x_2}x_1)\vee(\overline{x_3}x_1\overline{x_0})$

For **E**:

X_1X_0 | X_3X_2	00	01	11	10
00	1	0	-	1
01	0	0	-	0
11	0	0	-	-
10	1	1	-	-

E = $(\overline{X_2}\,\overline{X_0})\vee(X_2\,X_1\,\overline{X_0})$

For **F**:

X_1X_0 | X_3X_2	00	01	11	10
00	1	1	-	1
01	0	1	-	1
11	0	0	-	-
10	0	1	-	-

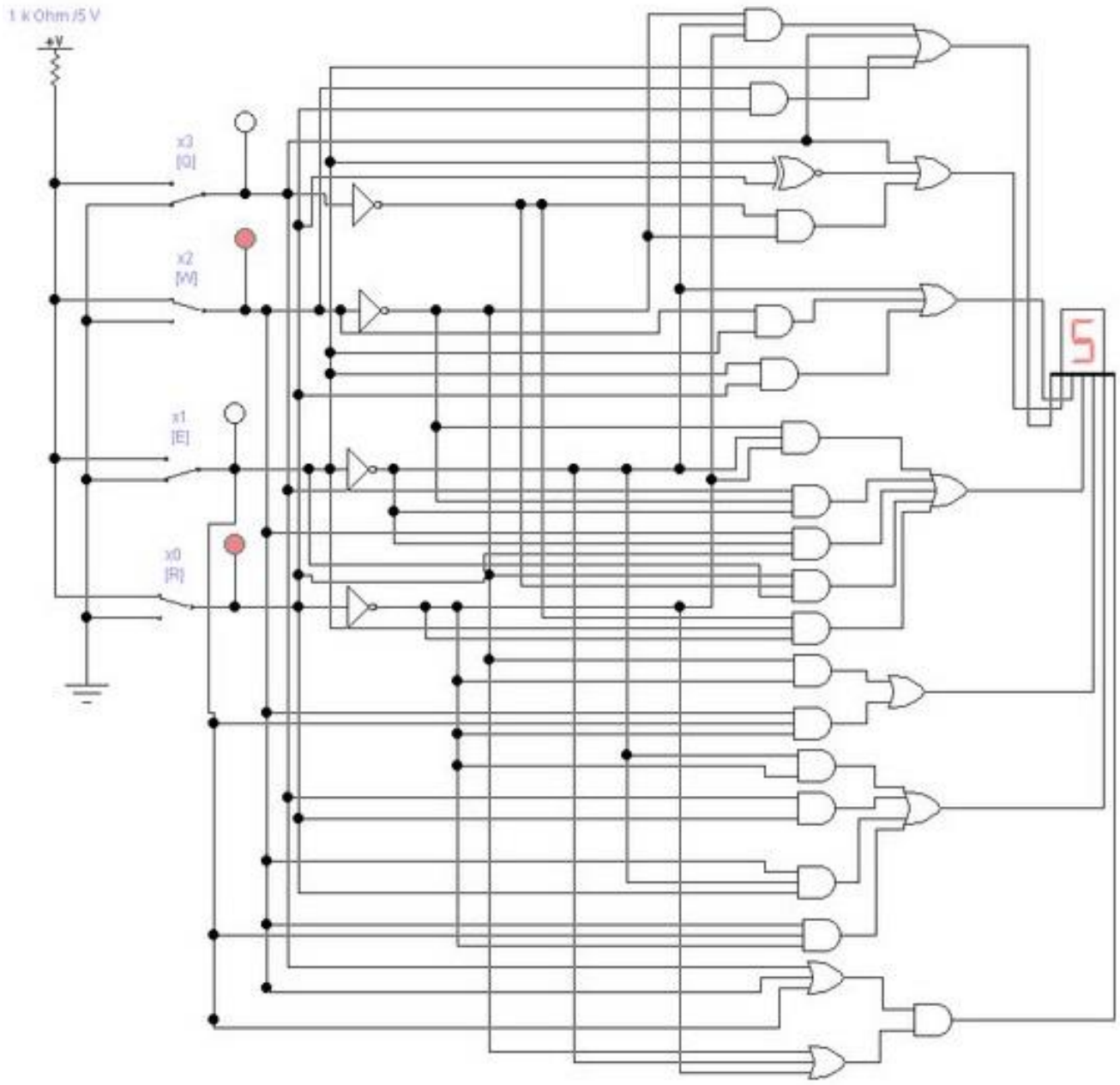
F = $(\overline{X_1}\,\overline{X_0})\vee(X_3\,X_0)\vee(X_2\,\overline{X_1}\,X_0)\vee(X_2\,X_1\,\overline{X_0})$

For **G**:

X_1X_0 | X_3X_2	00	01	11	10
00	0	1	-	1
01	0	1	-	1
11	1	0	-	-
10	1	1	-	-

G = $(X_3\vee X_2\vee X_1)\bullet(\overline{X_2}\vee\overline{X_1}\vee\overline{X_0})$

The circuit is following:



CONCLUSION: As a result of this work, the principle of operation and the device of the decoder were studied.

Laboratory Work 5

"Multiplexers and its usage

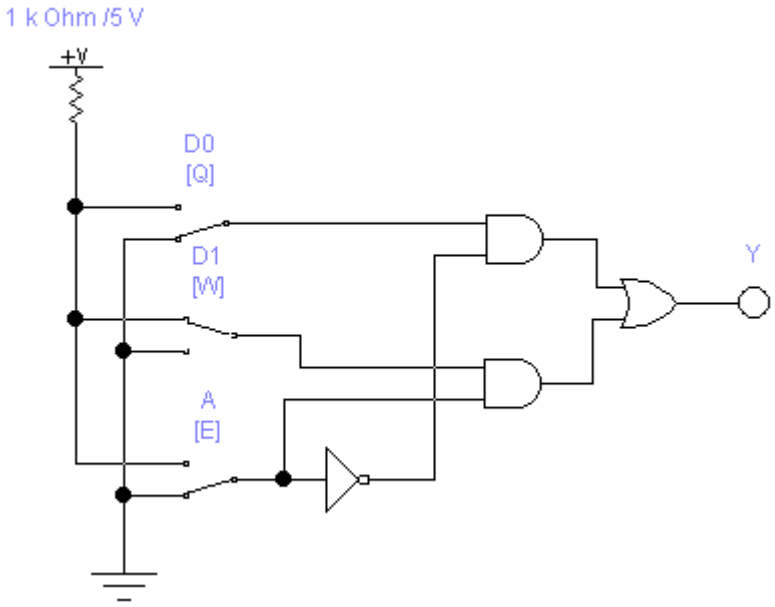
1) Build and test multiplexer circuit 2-1 on logic elements

Truth table:

	A	Y
D_0	0	$\overline{A}$
D_1	1	A

$$Y = \overline{A} D_0 \vee A D_1$$

The circuit is following:



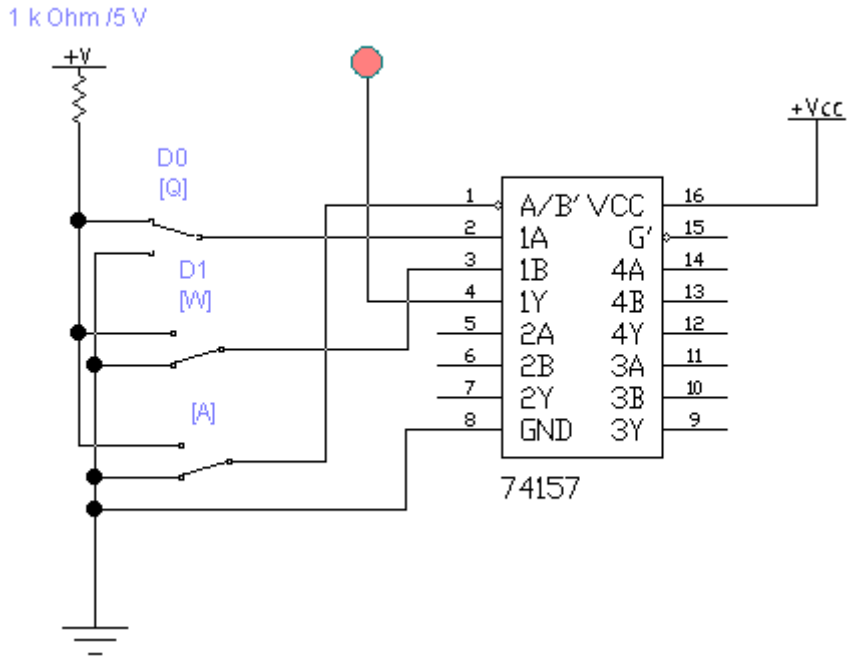
When D_0 is switched to the source of power, D_1 is switched to the ground on the output transfers value of $\overline{A}$.

Value of **A** transfers on the output as D_1 is switched to source of power and D_0 is switched to the ground.

2) Build and test multiplexer circuit at 2-1 on a chip SN74157

The truth table, formula, properties are the same as in task 1 as it is chip version of 2 to 1 multiplexer.

The circuit is following:



3) Build and test multiplexer circuit 4-1 on logic elements

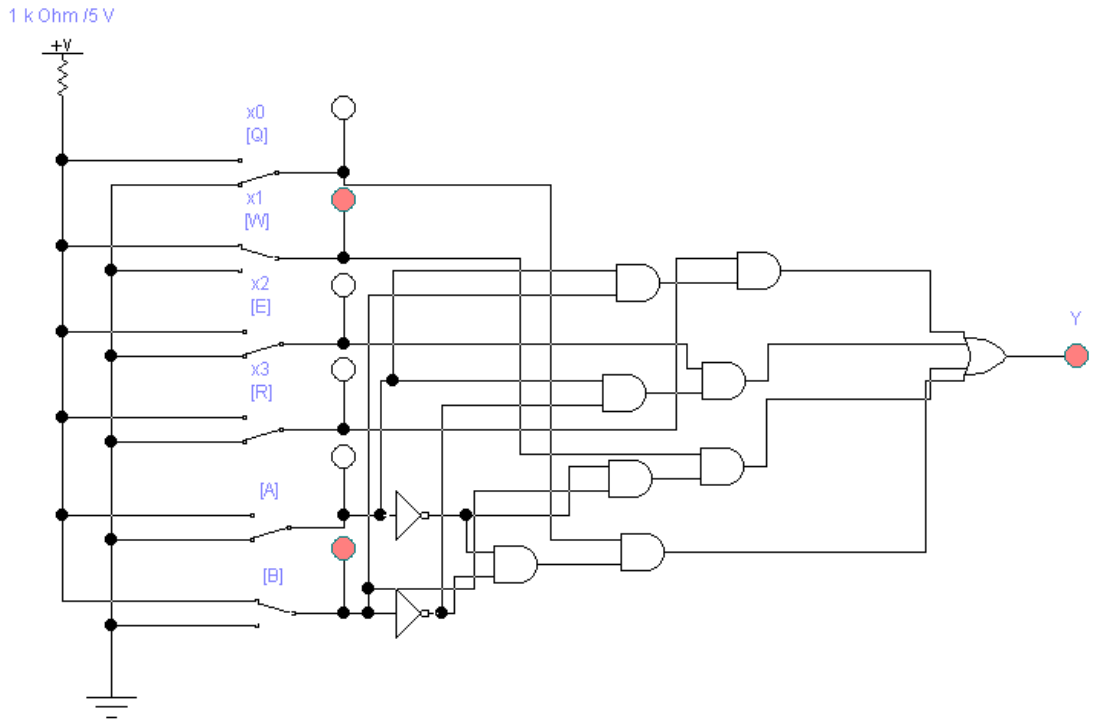
Truth table:

A, B - address inputs; $\mathbf{x_0}$, $\mathbf{x_1}$, $\mathbf{x_2}$, $\mathbf{x_3}$ - data inputs

Address inputs		**Data input**				**Out**
A	**B**	$\mathbf{x_3}$	$\mathbf{x_2}$	$\mathbf{x_1}$	$\mathbf{x_0}$	**Y**
0	0	x	x	x	1	1
0	1	x	x	1	x	1
1	0	x	1	x	x	1
1	1	1	x	x	x	1

Y = $(\overline{A} \bullet \overline{B}) \bullet x_0 \vee (\overline{A} \bullet B) \bullet x_1 \vee (A \bullet \overline{B}) \bullet x_2 \vee (A \bullet B) \bullet x_3$

The circuit is following:



It works as follow:

When address inputs **A** and **B** are: **0; 0** transfers value of $\mathbf{x_0}$;

When address inputs **A** and **B** are: **0; 1** transfers value of $\mathbf{x_1}$;

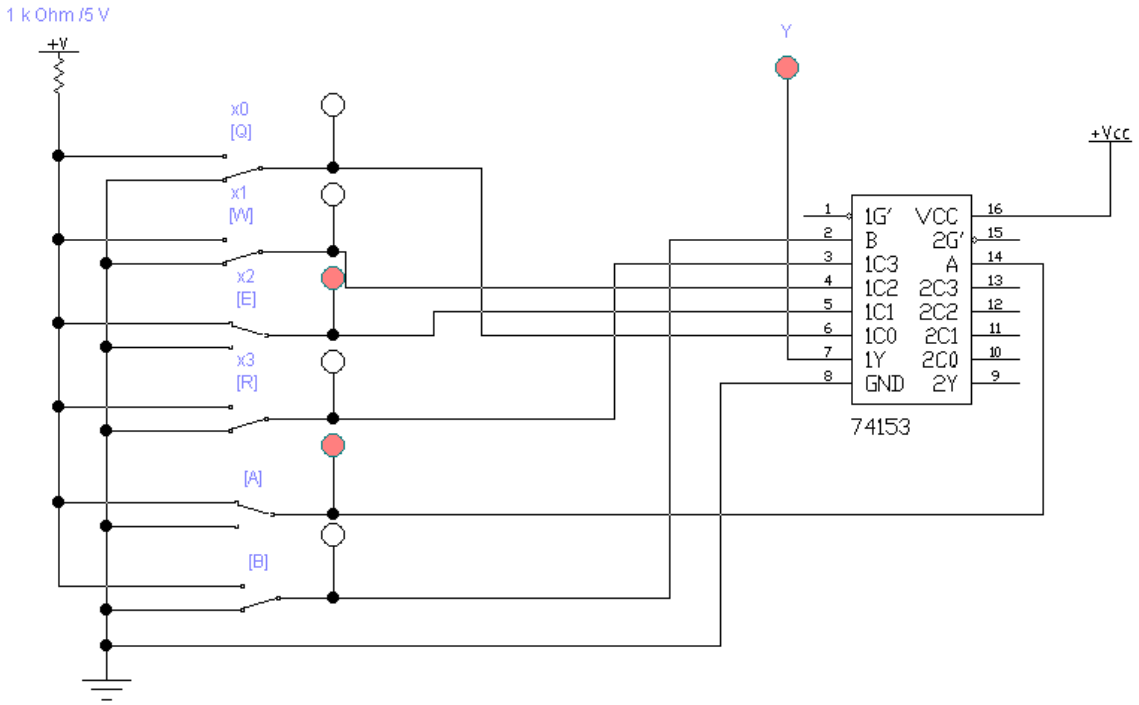
When address inputs **A** and **B** are: **1; 0** transfers value of $\mathbf{x_2}$;

When address inputs **A** and **B** are: **1; 1** transfers value of $\mathbf{x_3}$;

4) Build and verify multiplexer circuit at 4-1 on a chip SN74153

The truth table, formula, properties are the same as in task 3 as it is chip version of 4 to 1 multiplexer.

The circuit is following:

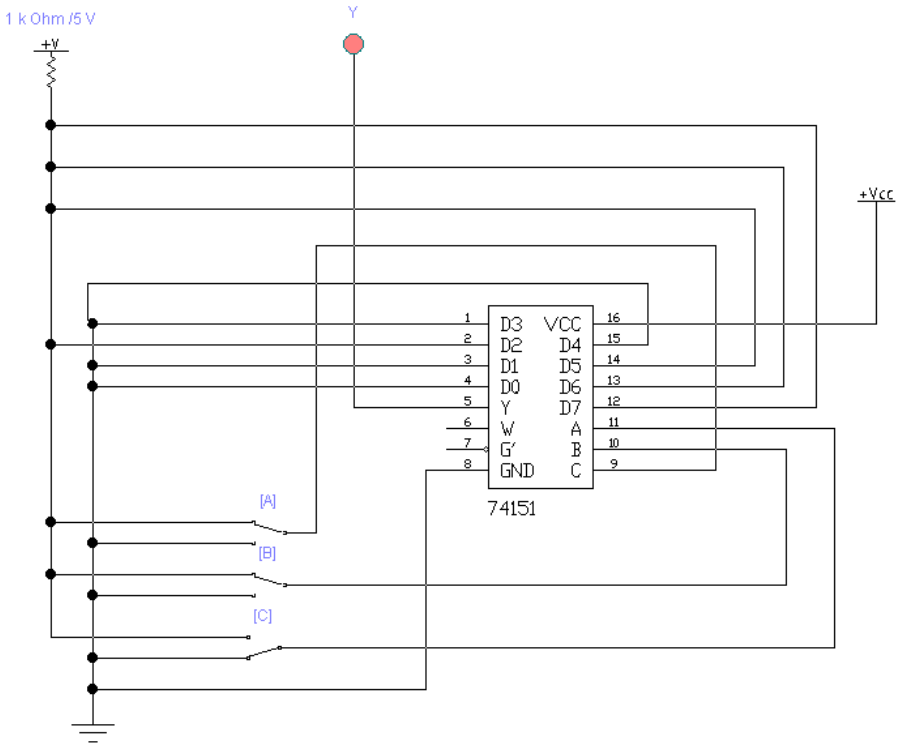


5) Build and test the circuit realization of a given teacher switching function on the chip SN74151 (8-1 multiplexer).

Truth table:

N	A	B	C	Y
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

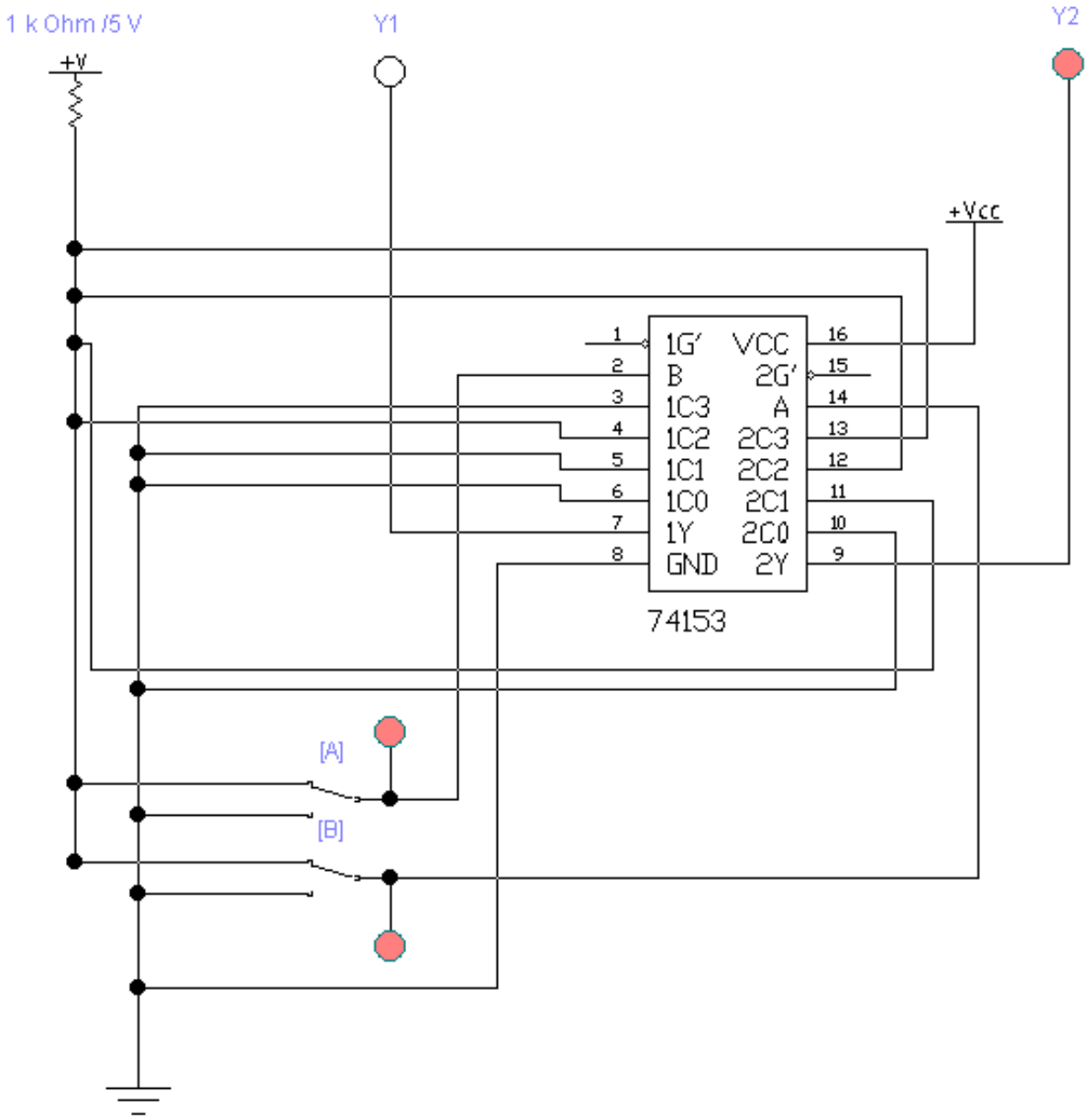
The circuit is following:



7. Build and test the circuit implementation of a given teacher switching function on only one multiplexer 4-1 (chip SN74153).

N		A	B	Y_1	Y_2
0	**4**	0	0	0	0
1	**5**	0	1	0	1
2	**6**	1	0	1	1
3	**7**	1	1	0s	1

The circuit is following:



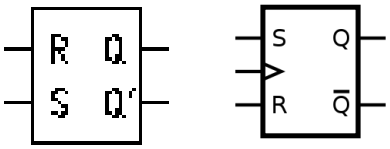
CONCLUSION: As a result of this work, how to design multiplexers and how to use them were studied.

Laboratory Work 6

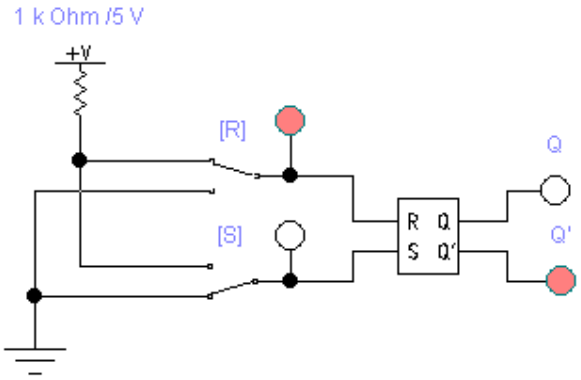
“Flip - flops”

1. Build RS flip-flop circuit and fill in the truth table.

Here schematic representations of RS flip-flop:



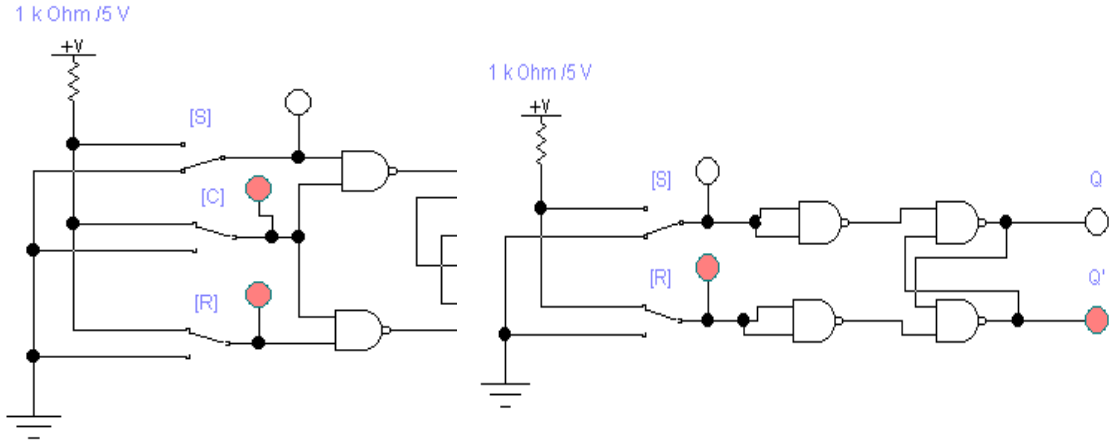
RS flip-flop on single chip:



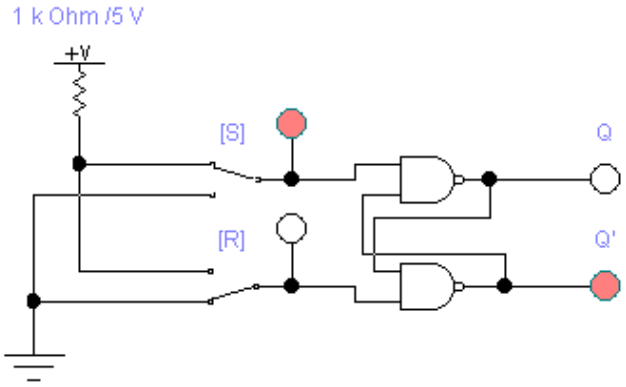
R	**S**	**Q**	$\overline{Q}$
0	0	Q^t	
0	1	1	0
1	0	0	1
1	1	Invalid	

Clocked RS flip-flop:

C	**R**	**S**	**Q**	$\overline{Q}$
0	0	0	Q^t	
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1	1	0
1	1	0	0	1
1	1	1	Invalid	



RS flip-flop on NAND:



S	**R**	**Q**	$\overline{Q}$
0	0	Q^t	
0	1	0	1
1	0	1	0
1	1	Invalid	

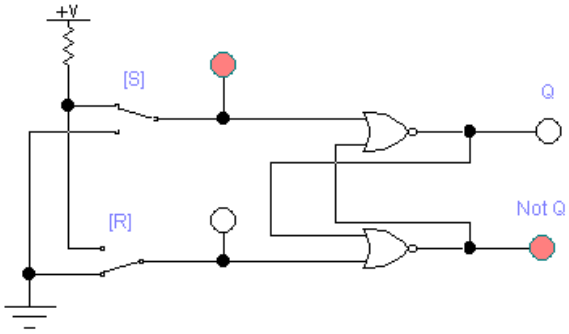
Inverse RS flip-flop on NAND:

S	**R**	**Q**	$\overline{Q}$
0	0	Invalid	
0	1	1	0
1	0	0	1
1	1	Q^t	

RS flip-flop on NOR:

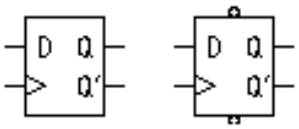
S	R	Q	$\overline{Q}$
0	0	Q^t	
0	1	1	0
1	0	0	1
1	1	Invalid	

Inputs			Output
R^t	S^t	Q^t	Q^{t+1}
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	Invalid
1	1	1	Invalid



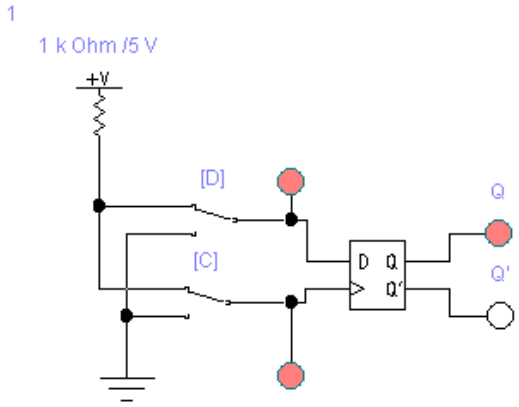
2. Build D flip-flop circuit and fill in the truth table.

Here schematic representations of D flip-flop:



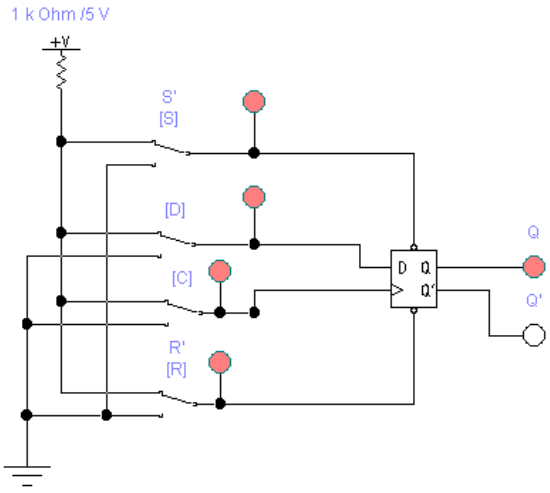
Synchronous D flip-flop on single chip:

C	D	Q	$\overline{Q}$
0	0	Q^t	
0	1		
1	0	0	1
1	1	1	0

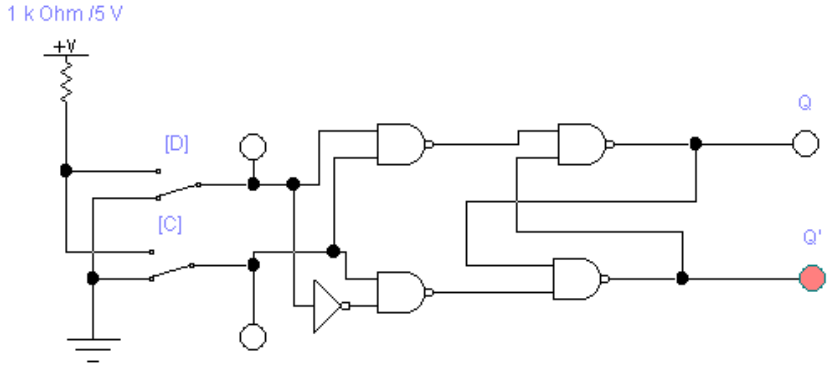


Asynchronous D flip-flop with Low Impulses on single chip:

S'	R'	C	D	Q	$\overline{Q}$
0	0	x	x	1	1
0	1	x	x	1	0
1	0	x	x	0	1
1	1	1	0	0	1
1	1	1	1	1	0



D flip-flop on RS flip-flop



C	D	Q	$\overline{Q}$
0	0	Q^t	
0	1		
1	0	0	1
1	1	1	0

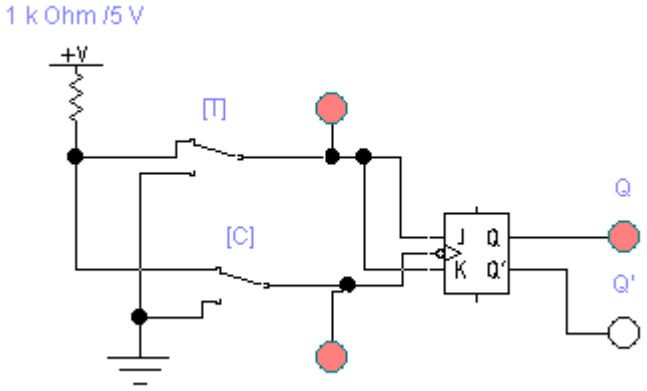
Inputs			Output
C^t	D^t	Q^t	Q^{t+1}
0	0	0	*0*
0	0	1	*0*
0	1	0	*1*
0	1	1	*1*
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$Q^{t+1} = D^t$

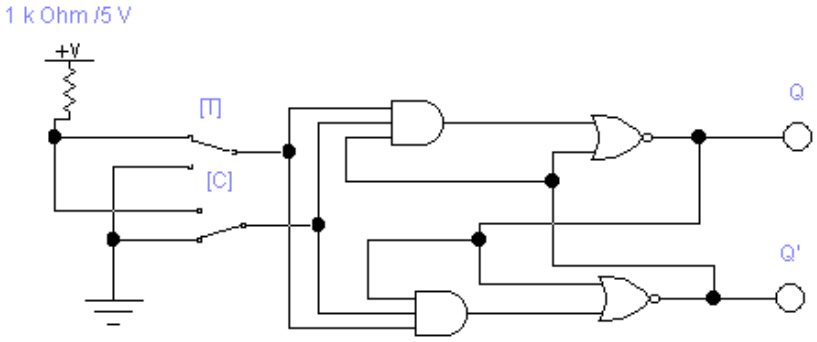
3. Build T flip-flop circuit and fill in the truth table.

T flip-flop on JK flip-flop on single circuit:

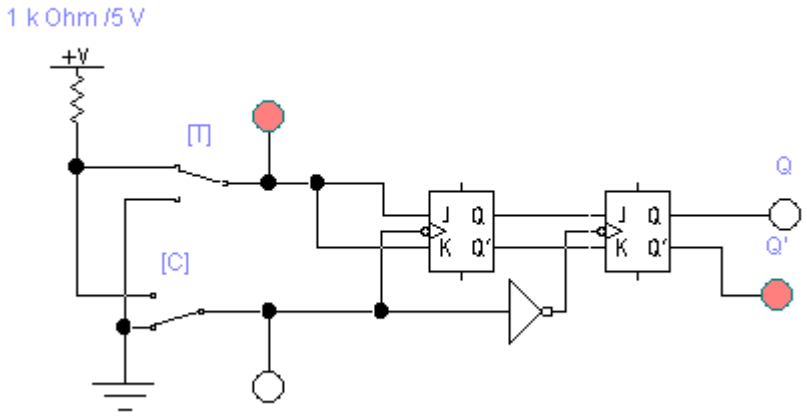
C	**T**	Q^{t+1}
0	0	Q^t
0	1	Q^t
1	0	Q^t
1	1	$\overline{Q^t}$



T flip-flop:



Master - slave T flip-flop on JK flip-flop on single circuit:

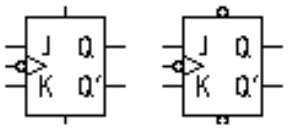


Inputs			Output
C^t	T^t	Q^t	Q^{t+1}
0	0	0	*0*
0	0	1	*1*
0	1	0	*1*
0	1	1	*0*
1	0	0	*0*
1	0	1	*1*
1	1	0	1
1	1	1	0

$Q^{t+1} = Q^t \oplus T^t$

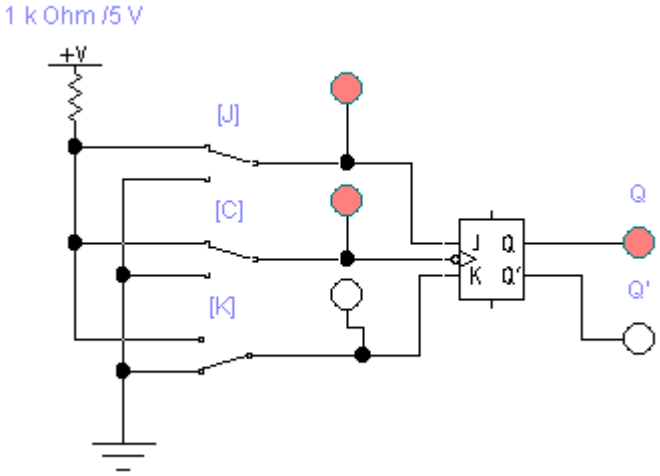
4. Build JK flip-flop circuit and fill in the truth table.

Here schematic representations of JK flip-flop:

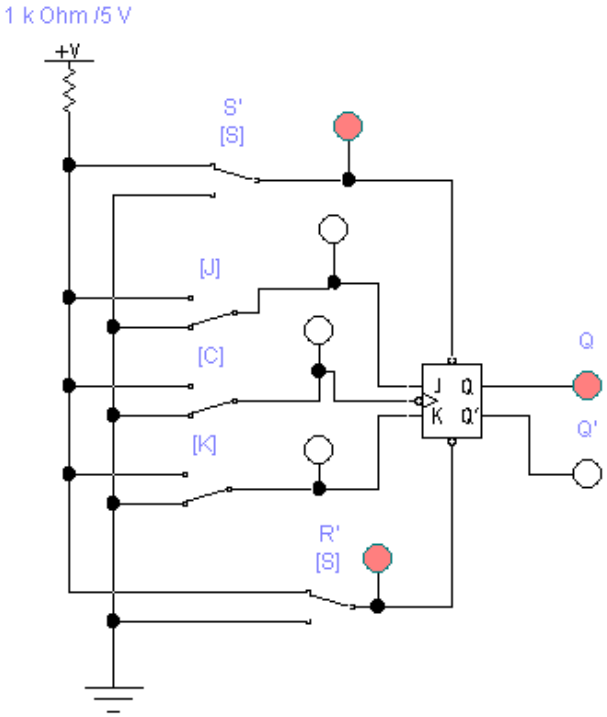


JK flip-flop on single chip:

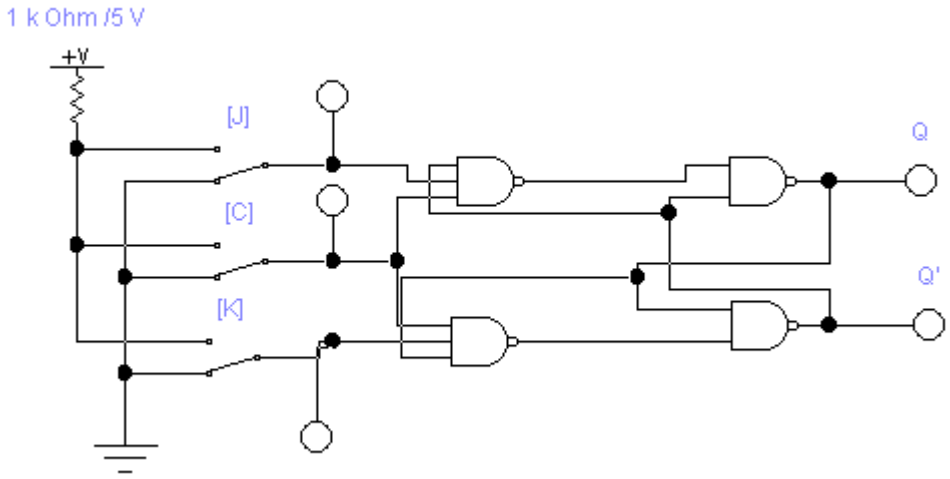
C	J	K	Q^{t+1}
0	X	X	Q^t
1	0	0	Q^t
1	0	1	0
1	1	0	1
1	1	1	$\overline{Q^t}$



Asynchronous JK flip-flop with Low Impulses on single chip:



JK flip-flop on RS flip - flop:



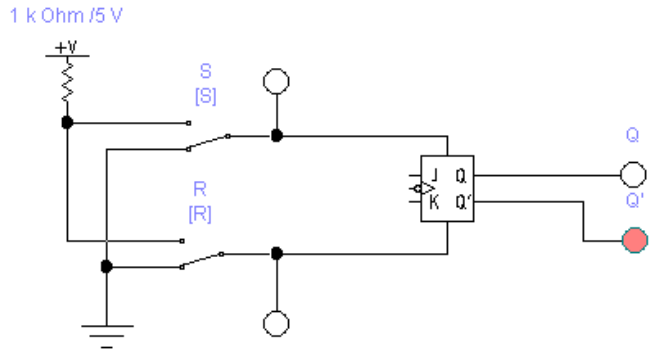
Inputs			Output
J^t	K^t	Q^t	Q^{t+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

For Q^{t+1}:

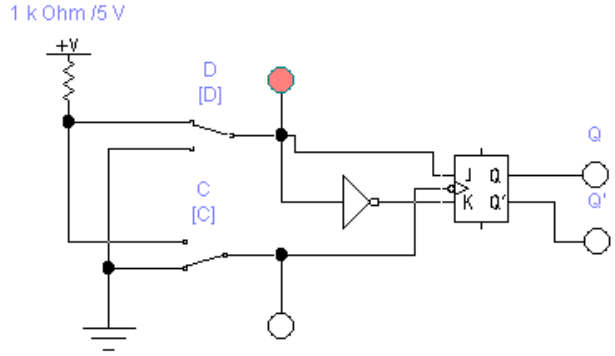
$$Q^{t+1} = \overline{J}K \vee J\overline{Q^t}$$

Q^t / J^tK^t	**00**	**01**	**11**	**10**
0	0	1	1	1
1	0	1	0	0

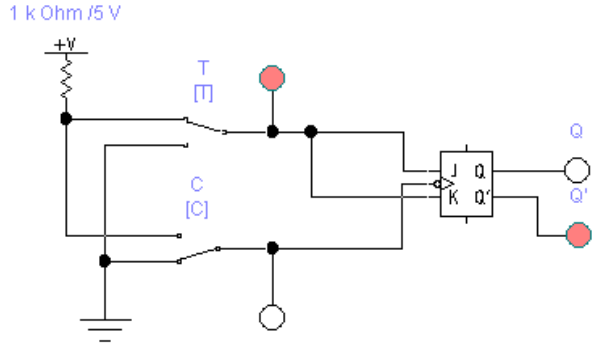
5. **Build RS, D, T flip-flops on JK flip-flop and check its work.**
RS flip-flop on JK flip - flop:



D flip-flop on JK flip - flop:



T flip-flop on JK flip - flop:



CONCLUSION: as a result of this work, the basic principles, structure, properties and functions of various triggers were studied.

Laboratory Work 7

“Registers”

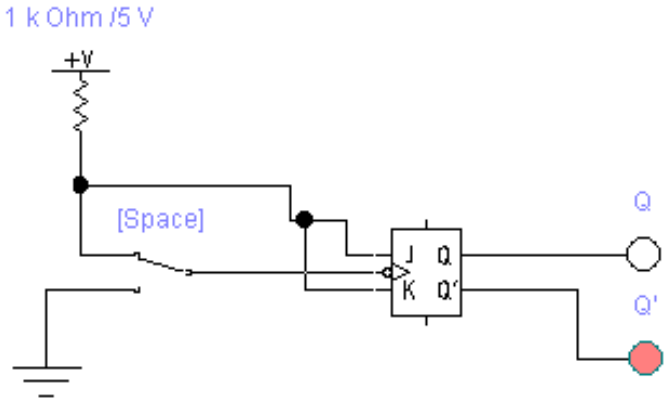
1. Build circuit of a 4-bits parallel register on D flip-flops and check its work.
2. Build circuit of a serial (shift) registers on D flip-flops and check its work.
    a. Build circuit of a 4-bits left shift register on D flip-flops and check its work.
    b. Build circuit of a 4-bits right shift register on D flip-flops and check its work.
    c. Build circuit of a 4-bits ring register on D flip-flops and check its work.
3. Build circuit of 4-bits Johnson register on D flip-flops and check its work.

CONCLUSION:

Laboratory Work 8

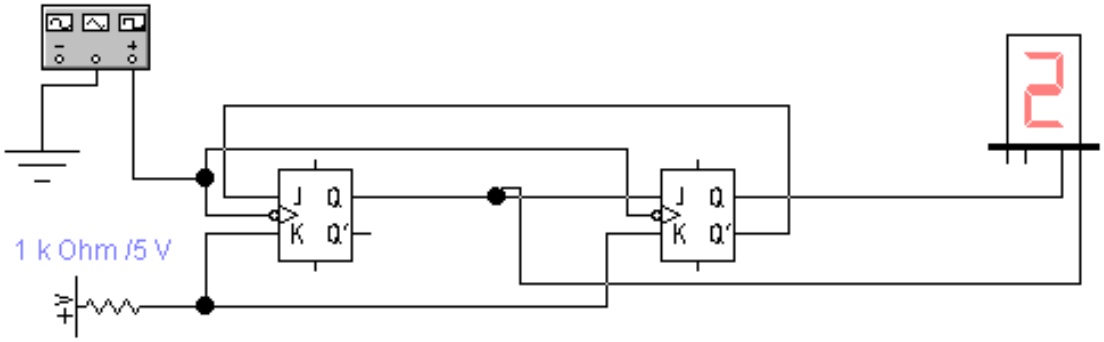
“Counters”

1. **Build and check the work of counter with module (coefficient of counting) 2.**



2. **Build and check the work of counter with module (coefficient of counting) 3.**

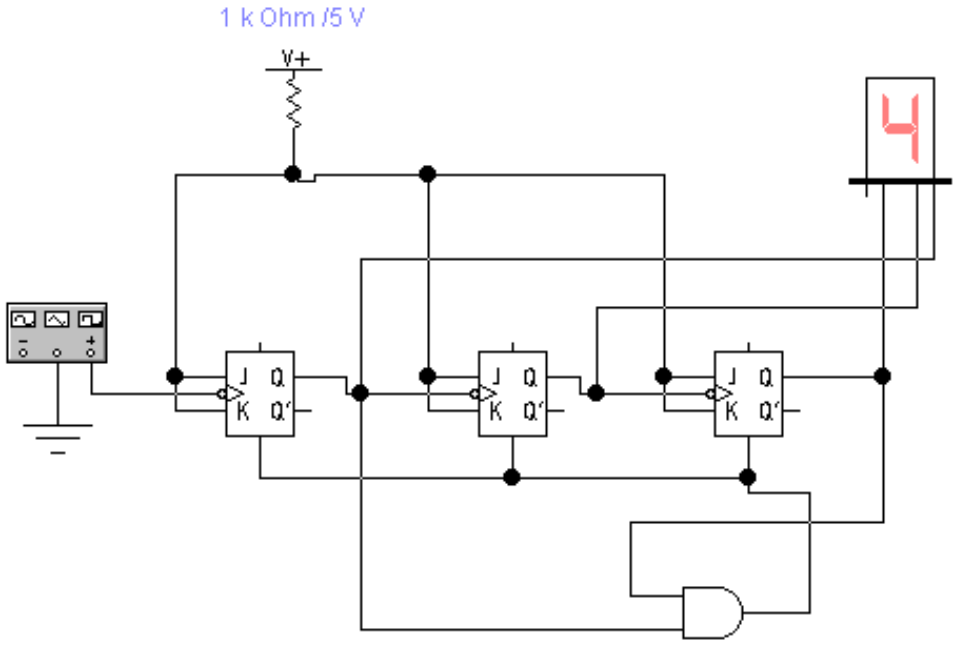
№	t		t+1		t			
	Q_2	Q_1	Q_2	Q_1	J_2	K_2	J_1	K_1



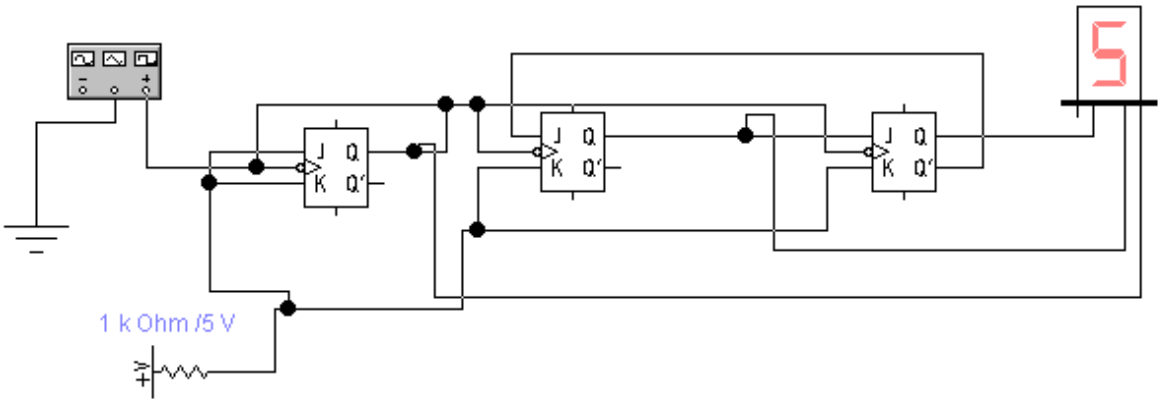
0	0	0	0	1	0	-	1	-
1	0	1	1	0	1	-	-	1
2	1	0	0	0	-	1	0	-
3	1	1	-	-	-	-	-	-

$J_1 = \overline{Q}_2$; K_1= 1; $J_2 = Q_1$; K_2=1;

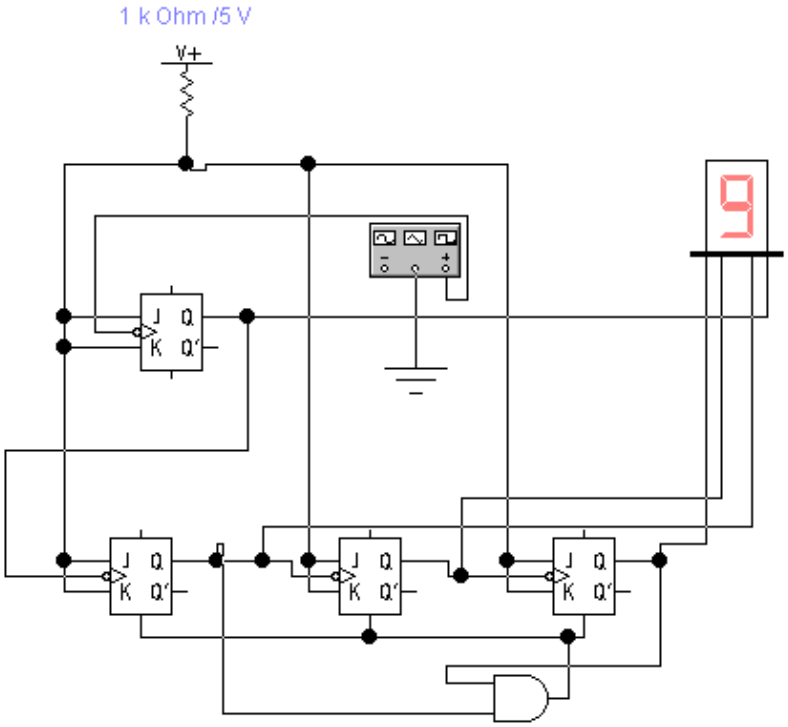
3. **Build and check the work of counter with module (coefficient of counting) 5.**



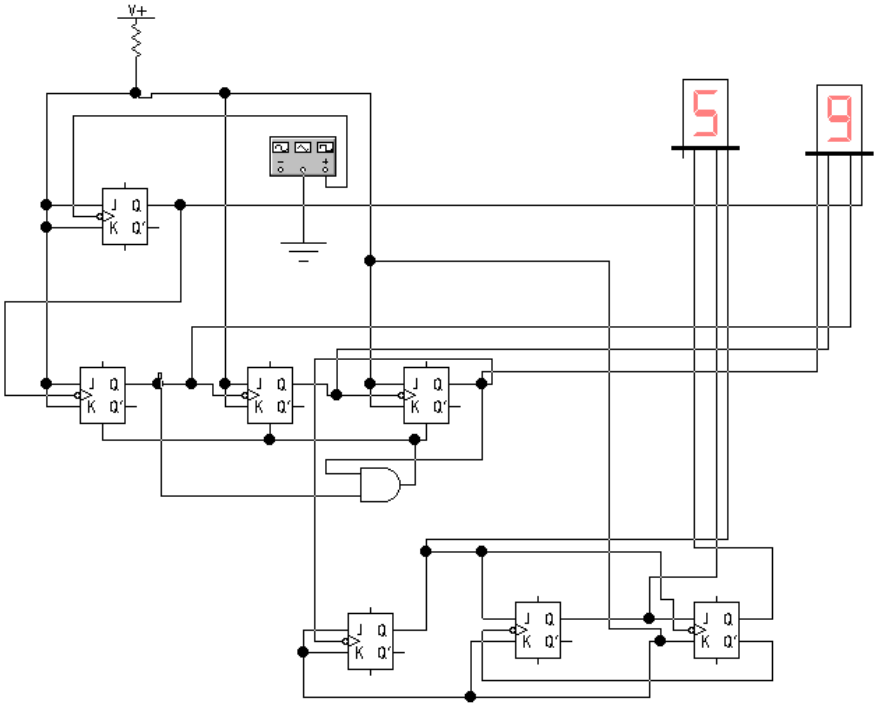
4. **Build and check the work of counter with module (coefficient of counting) 6.**



5. **Build and check the work of counter with module (coefficient of counting) 10.**



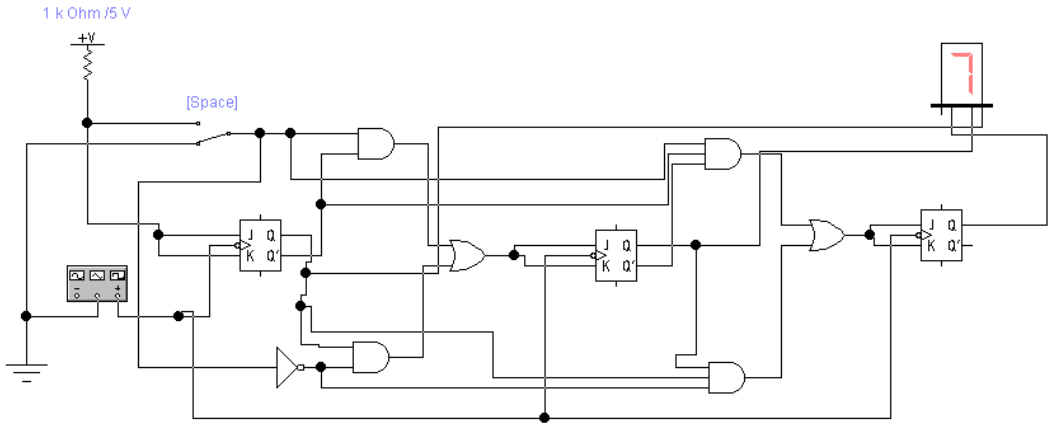
6. **Build and check the work of counter with module (coefficient of counting) 60.**



7. Build and check the work of reverse counter.

№	t				t+1			t					
	x	Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	0	0	0	1	0	-	0	-	1	-
1	0	0	0	1	0	1	0	0	-	1	-	-	1
2	0	0	1	0	0	1	1	0	-	-	0	1	-
3	0	0	1	1	1	0	0	1	-	-	1	-	1
4	0	1	0	0	1	0	1	-	0	0	-	1	-
5	0	1	0	1	1	1	0	-	0	1	-	-	1
6	0	1	1	0	1	1	1	-	0	-	0	1	-
7	0	1	1	1	0	0	0	-	1	-	1	-	1
8	1	0	0	0	1	1	1	1	-	1	-	1	-
9	1	0	0	1	0	0	0	0	-	0	-	-	1
10	1	0	1	0	0	0	1	0	-	-	1	1	-
11	1	0	1	1	0	1	0	0	-	-	0	-	1
12	1	1	0	0	0	1	1	-	1	1	-	1	-
13	1	1	0	1	1	0	0	-	0	0	-	-	1
14	1	1	1	0	1	0	1	-	0	-	1	1	-
15	1	1	1	1	1	1	0	-	0	-	0	-	1

$$J_2 = K_2 = \bar{x}Q_1Q_0 \vee x\overline{Q}_1\overline{Q}_0;\ J_1 = K_1 = \bar{x}Q_0 \vee x\overline{Q}_0;\ J_0 = K_0 = 1.$$



CONCLUSION: