Laboratory 1

Introduction to Logisim and Circuit Development

1. Introduction and Purpose of Experiment

   Students will learn to use Logisim simulator to simulate logic circuits and implement them using appropriate ICs

2. Aim and Objectives

   Aim: To use Logisim to simulate logic circuits and implement them

   Objectives: At the end of this lab, the student will be able to

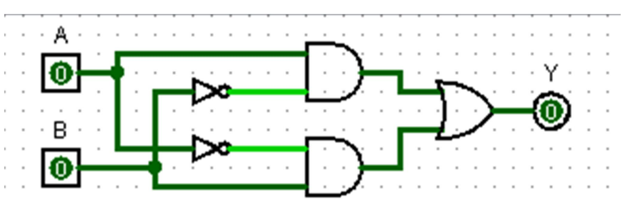
   - Use Logisim to simulate logic circuits
   - Choose appropriate ICs to implement the logic circuits
   - Implement the logic circuits using the ICs and hardware kit

3. Experimental Procedure

   a. Draw the truth tables and circuit diagrams for the following expressions.
      1. $Y = A{\sim}B + {\sim}AB$
      2. $W = BC + {\sim}BC$
      3. $O = {\sim}ABC + A{\sim}BC + ABC$
      4. $X = {\sim}AB + {\sim}AB{\sim}C + {\sim}ABCD + {\sim}AB{\sim}C{\sim}D$
      5. $F = {\sim}WXYZ + {\sim}WXY{\sim}Z + WXYZ + WXY{\sim}Z$

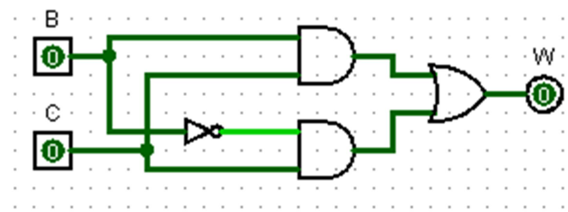
   b. Use Logisim to generate the truth tables and circuit diagrams for the above expressions.

      1) Question 1



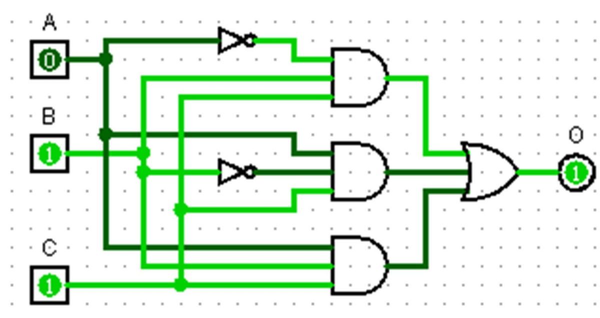
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Question 2



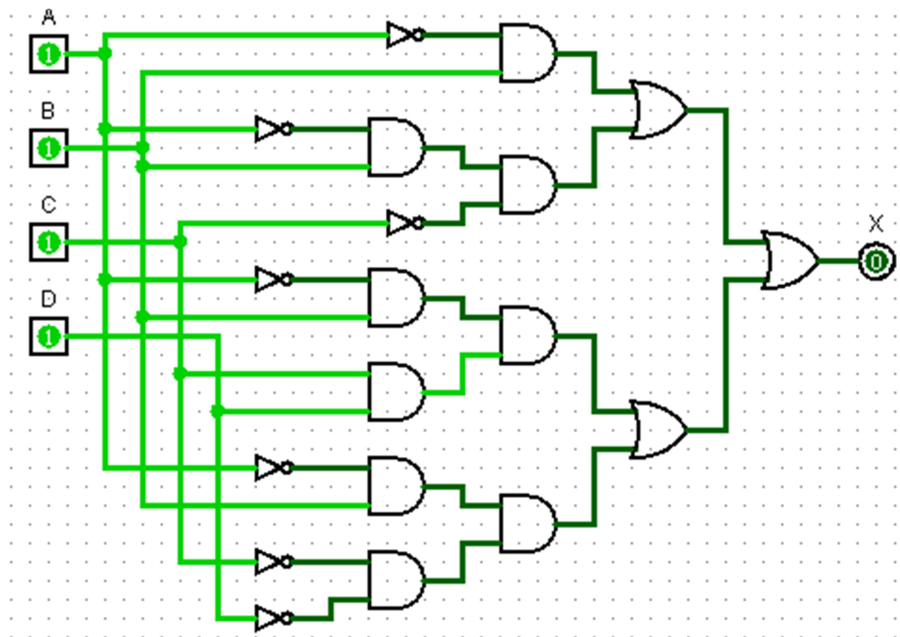
B	C	W
0	0	0
0	1	1
1	0	0
1	1	1

Question 3



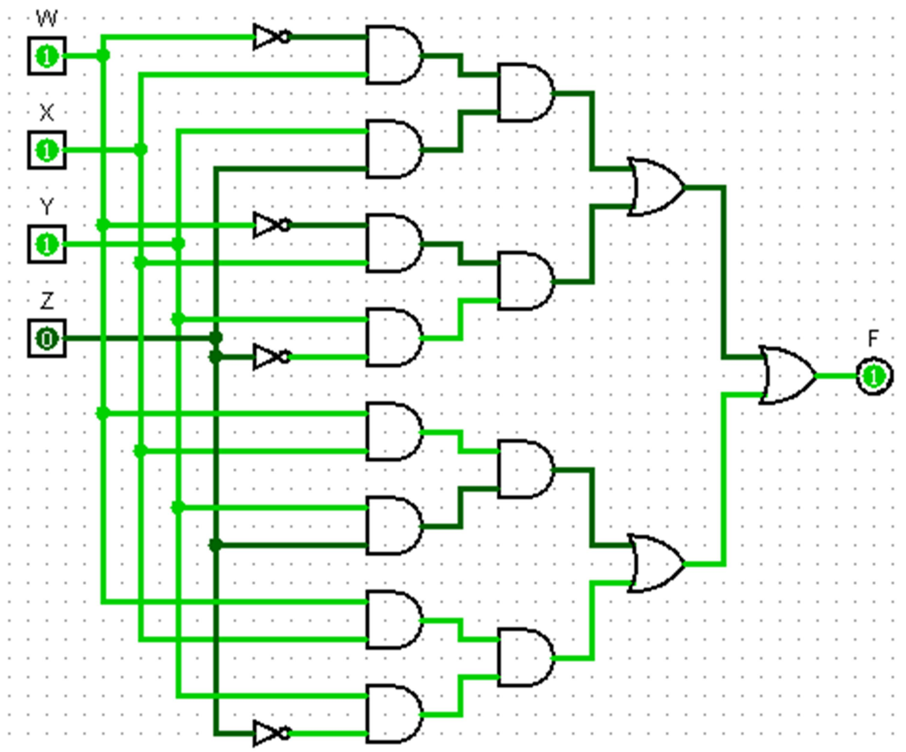
A	B	C	O
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Question 4



A	B	C	D	X
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
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	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	0
1	1	1	0	0
1	1	1	1	0

Question 5



W	X	Y	Z	F
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	1
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	0
1	1	1	0	1
1	1	1	1	1

c. Implement the first three expressions in the non-minimized form and verify the truth tables. Show the output to the course leader.

d. Do you see any limitations in the simulator and/or the hardware kit? Discuss how these can be overcome.

Your document should include:

- Handwritten truth tables and circuit diagrams for the expressions
- Logisim screenshots
- Answer to 3(d)

Laboratory 2

Boolean Expressions using Universal Gates

1. Introduction and Purpose of Experiment

   Students will learn to simulate and implement logic circuits using only universal gates.

2. Aim and Objectives

   Aim: To simulate and implement logic circuits using only NAND and NOR gates

   Objectives: At the end of this lab, the student will be able to

   - Use Logisim to simulate Boolean circuits using only NAND gates
   - Describe the procedure to convert all the gates in a circuit to universal gates
   - Draw circuit diagrams for Boolean expressions using only universal gates

3. Experimental Procedure

   a. Draw truth tables and circuit diagrams for the following expressions using only NAND gates.
      1. $F = AB + CD$
      2. $F = X{\sim}Y + {\sim}XY + Z$
      3. $F = A(CD + B) + BC$
      4. $F = AB + BC + AC$

   b. Draw truth tables and circuit diagrams for the following expressions using only NOR gates.
      1. $F = (A + B)(C + D)E$
      2. $F = (A + B + C)({\sim}A + {\sim}B + {\sim}C)$

   c. Use Logisim to generate truth tables and circuit diagrams for the expressions in 3(a).

   d. Implement the first three expressions in the non-minimized form and verify the truth tables. Show the output to the course leader.

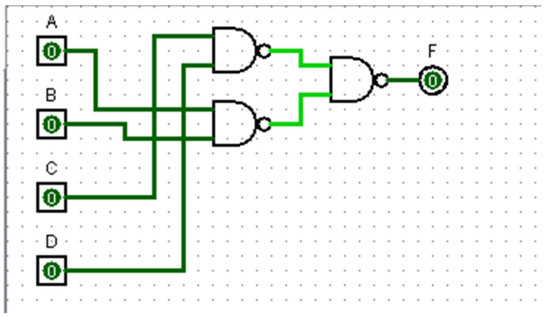
   e. Why is it easier to draw a circuit diagram using universal gates if the Boolean expression is in standard/canonical form?

   Your document should include:

   - Handwritten truth tables and circuit diagrams for the expressions
   - Logisim screenshots
   - Answer to 3(e)

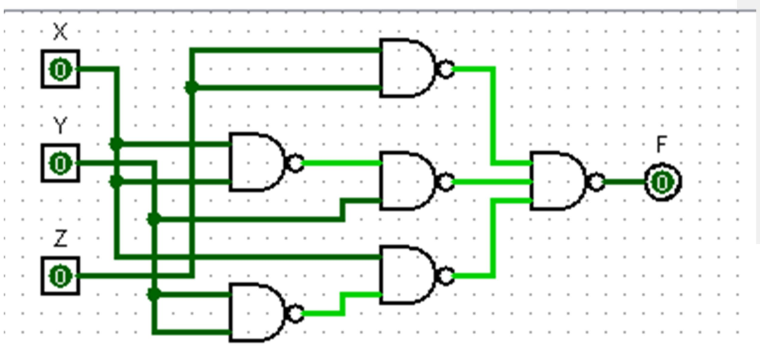
Solutions:

1.



A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0

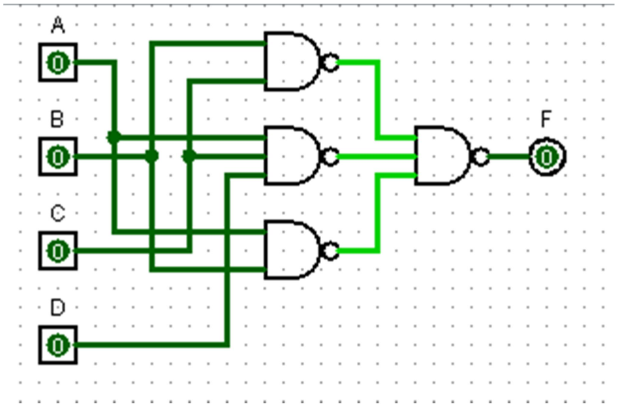
2.



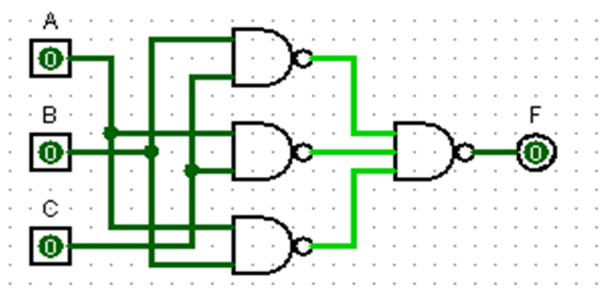
X	Y	Z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

3.

A	B	C	D	F
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	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

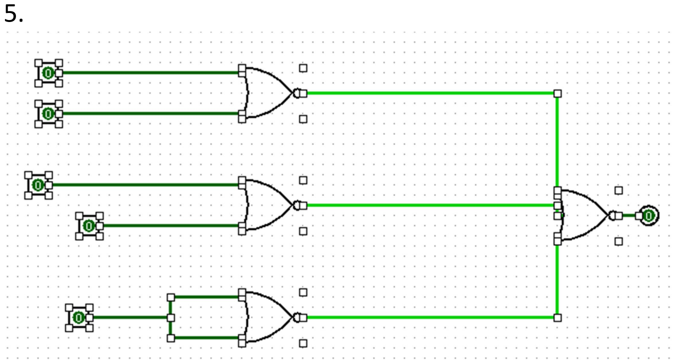


4.



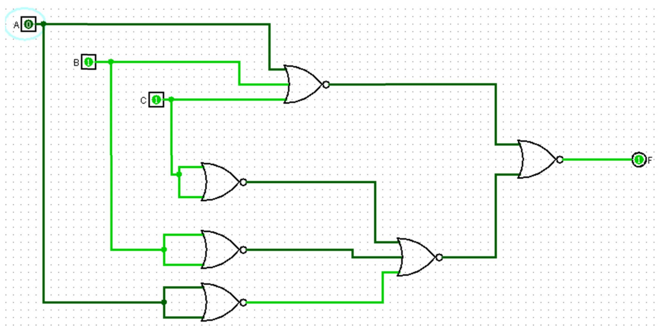
A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

5.



A	B	C	D	E	F
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	0
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	0	1	0
0	0	1	1	0	0
0	0	1	1	1	0
0	1	0	0	0	0
0	1	0	0	1	0
0	1	0	1	0	0
0	1	0	1	1	0
0	1	1	0	0	0
0	1	1	0	1	0
0	1	1	1	0	0
0	1	1	1	1	0
1	0	0	0	0	0
1	0	0	0	1	0
1	0	0	1	0	0
1	0	0	1	1	0
1	0	1	0	0	0
1	0	1	0	1	0
1	0	1	1	0	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	0	1	0
1	1	0	1	0	0
1	1	0	1	1	0
1	1	1	0	0	0
1	1	1	0	1	0
1	1	1	1	0	0
1	1	1	1	1	1

6.



Laboratory 3

Gate Level Minimization using Karnaugh Maps

1. Introduction and Purpose of Experiment

   Students will learn to minimize Boolean Expressions using K-Maps and then simulate and implement them.

2. Aim and Objectives

   Aim: To apply K-Maps to minimize Boolean expressions

   Objectives: At the end of this lab, the student will be able to

   - Apply K-Maps to simplify three- and four-variable Boolean Expressions
   - Implement minimized expressions using basic and universal gates

3. Experimental Procedure

   a. Minimize the following expressions using K-Maps.
      1. $F(A,B,C) = \sum(1,3,5,7)$
      2. $F(A,B,C,D) = \sum(0,2,3,7,11,13,14,15)$
      3. $F(A,B,C) = \sum(2,3,4,5) + \phi(6,7)$
      4. $F = {\sim}A{\sim}B{\sim}C{\sim}D + A{\sim}C{\sim}D + {\sim}BC{\sim}D + {\sim}ABCD + B{\sim}CD$
      5. $F(A,B,C,D) = \prod(1,3,5,7,13,15)$
      6. $F(A,B,C,D) = \prod(1,3,6,9,11,12,14)$
      7. $F = ({\sim}A + B + {\sim}D).({\sim}A + {\sim}B + {\sim}C).({\sim}A + {\sim}B + C).({\sim}B + C + {\sim}D)$

   b. Draw truth tables and circuit diagrams for the minimized expressions in 3(a) considering:
      - 3(a) 2: Use basic gates
      - 3(a) 3 and 3(a) 4: Use NAND gates
      - 3(a) 5: Use basic gates

- 3(a) 6 and 3(a) 7: Use NOR gates

c. Use Logisim to generate truth tables and circuit diagrams for the expressions in 3(a)2 to 3(a)5.

d. Implement the minimized expressions of 3(a)4 to 3(a)6. Show the output to the course leader.

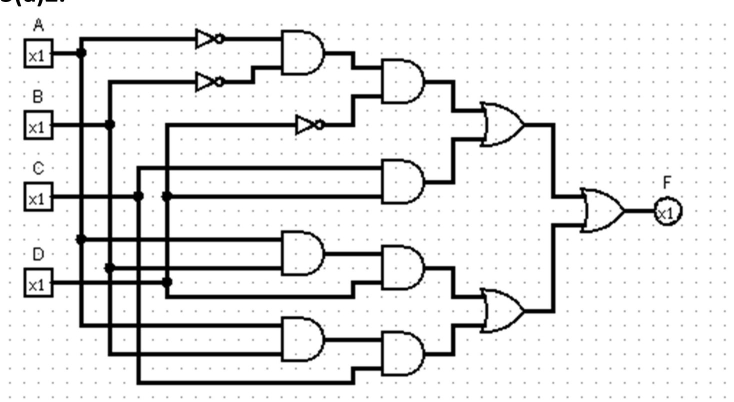
e. With an example, show why incorrect grouping in K-Maps may result in a non-minimized expression

Your document should include:

- Handwritten truth tables and circuit diagrams for the expressions
- Logisim screenshots
- Answer to 3(e)

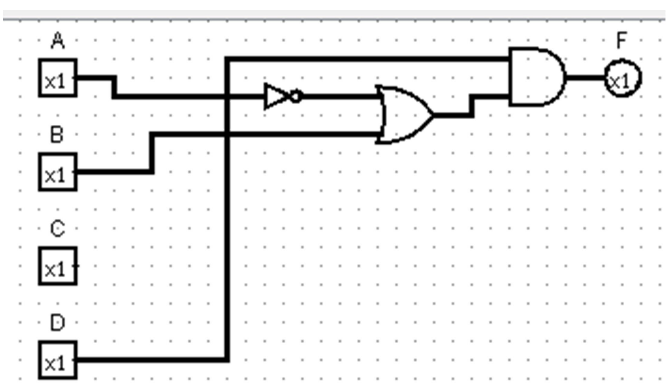
Solutions:

3(a)2.



A	B	C	D	F
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
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	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

3(a)5.



A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
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

Laboratory 4

Code Conversion Circuits

1. Introduction and Purpose of Experiment

   Students will learn to design, simulate and implement circuits for various code converters.

2. Aim and Objectives

   Aim: To generate code words using various codes and convert them from one code to another

   Objectives: At the end of this lab, the student will be able to

   - Develop expressions that convert code words from one code to another
   - Develop code converter circuits

3. Experimental Procedure

   a. Write truth tables for the following code converters.
      I. BCD to Gray
      II. Gray to BCD
      III. 4-bit BCD to Excess-3

   b. Use K-Maps to develop the minimized expressions for the above code converters

   c. Draw the circuit diagrams using the following gates:
      I. Binary to Gray: Using NAND gates
      II. Gray to BCD: Using Ex-OR gates
      III. 4-bit BCD to Excess-3 using Basic Gates

   f. Use Logisim to generate truth tables and circuit diagrams for all the code converters.

   g. Implement the code converters and show the output to the course leader.

   h. Show in detail how adders and subtractors are useful in developing these code converters.

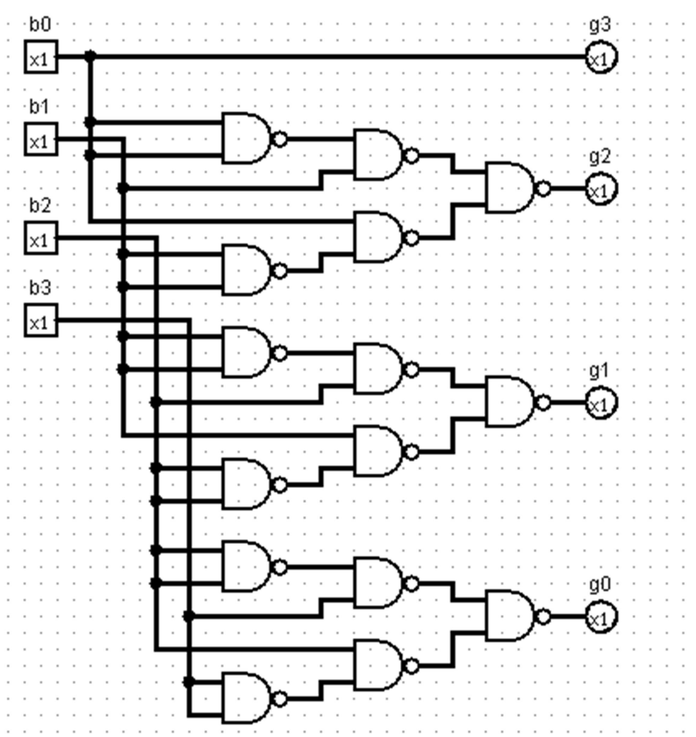
   Your document should include:

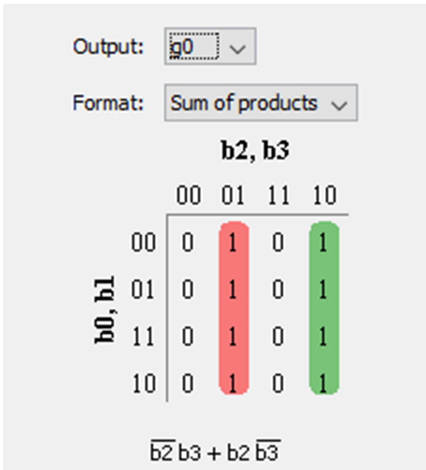
   - Handwritten truth tables, expressions and circuit diagrams for the code converters
   - Logisim screenshots
   - Answer to 3(h)

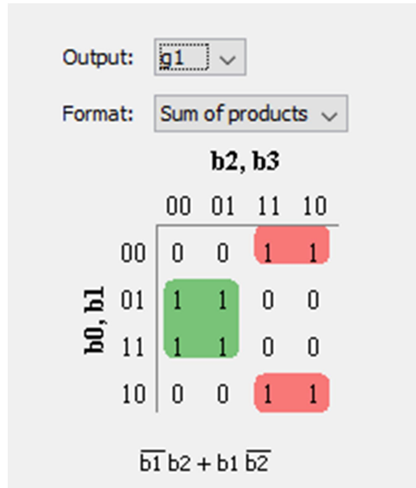
Solutions:

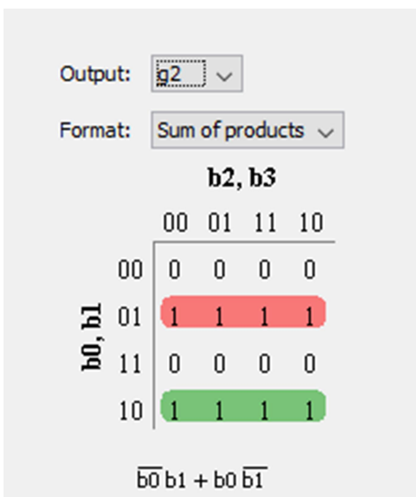
3.A.1

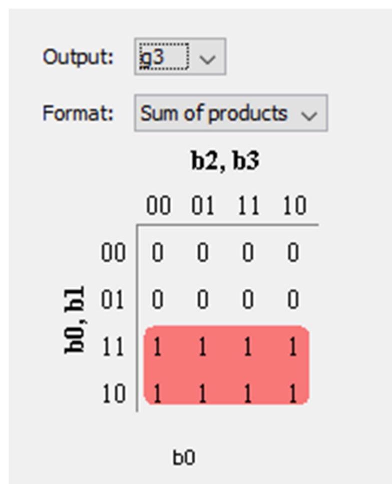
b0	b1	b2	b3	g3	g2	g1	g0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0




Output: g0
Format: Sum of products
b2, b3
00 01 11 10
b0, b1
00 0 1 0 1
01 0 1 0 1
11 0 1 0 1
10 0 1 0 1
$\overline{b2}\,b3 + b2\,\overline{b3}$


Output: g1
Format: Sum of products
b2, b3
00 01 11 10
b0, b1
00 0 0 1 1
01 1 1 0 0
11 1 1 0 0
10 0 0 1 1
$\overline{b1}\,b2 + b1\,\overline{b2}$


Output: g2
Format: Sum of products
b2, b3
00 01 11 10
b0, b1
00 0 0 0 0
01 1 1 1 1
11 0 0 0 0
10 1 1 1 1
$\overline{b0}\,b1 + b0\,\overline{b1}$


Output: g3
Format: Sum of products
b2, b3
00 01 11 10
b0, b1
00 0 0 0 0
01 0 0 0 0
11 1 1 1 1
10 1 1 1 1
b0

3.A.2

g3	g2	g1	g0	b3	b2	b1	b0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
1	0	0	0	1	1	1	1
1	0	0	1	1	1	1	0
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	0	1	0	1	1
1	1	1	1	1	0	1	0

Output: b0

Format: Sum of products

g1, g0

g3, g2	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

$\overline{g3}\,\overline{g2}\,\overline{g1}\,g0 + \overline{g3}\,\overline{g2}\,g1\,\overline{g0} + \overline{g3}\,g2\,\overline{g1}\,\overline{g0} + \overline{g3}\,g2\,g1\,g0 + g3\,\overline{g2}\,\overline{g1}\,\overline{g0} + g3\,\overline{g2}\,g1\,g0 + g3\,g2\,\overline{g1}\,g0 + g3\,g2\,g1\,\overline{g0}$

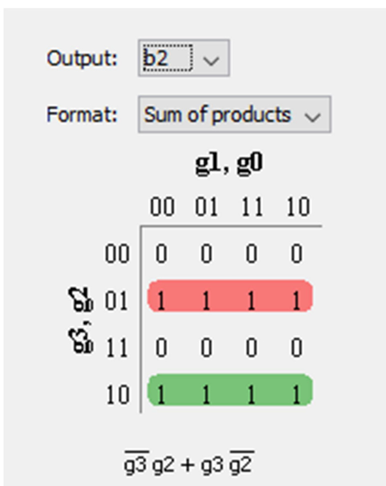
Output: b1

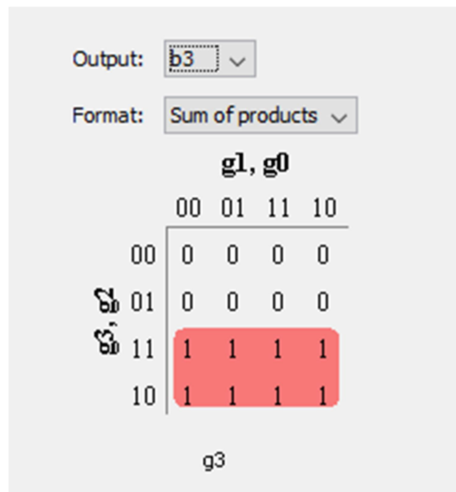
Format: Sum of products

g1, g0

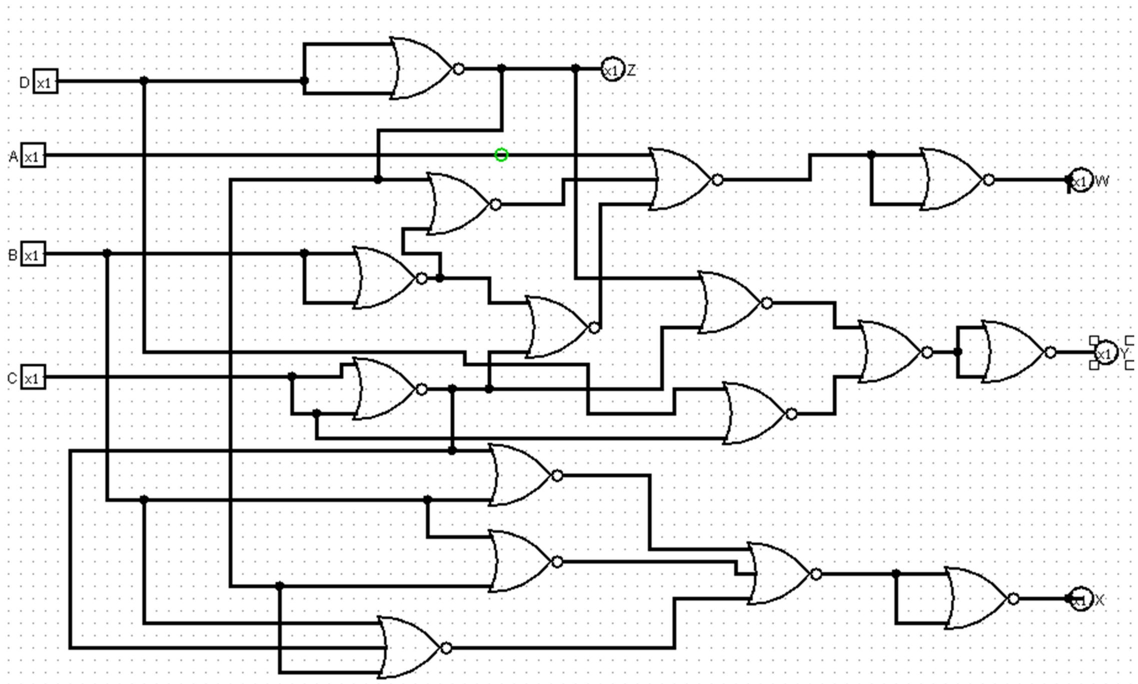
g3, g2	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	0	0	1	1
10	1	1	0	0

$\overline{g3}\,\overline{g2}\,g1 + \overline{g3}\,g2\,\overline{g1} + g3\,\overline{g2}\,\overline{g1} + g3\,g2\,g1$

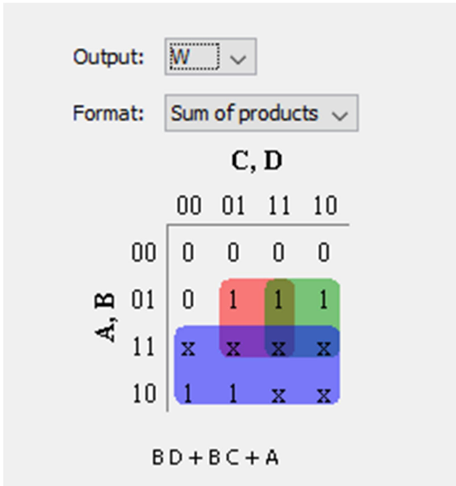

Output: b2
Format: Sum of products
g1, g0
00 01 11 10
g3, g2
00 0 0 0 0
01 1 1 1 1
11 0 0 0 0
10 1 1 1 1
$\overline{g3}\,g2 + g3\,\overline{g2}$

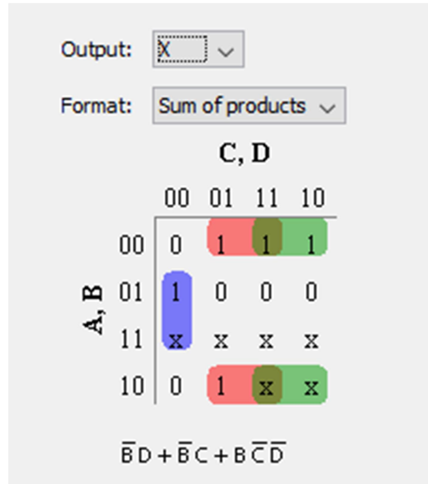

Output: b3
Format: Sum of products
g1, g0
00 01 11 10
g3, g2
00 0 0 0 0
01 0 0 0 0
11 1 1 1 1
10 1 1 1 1
g3

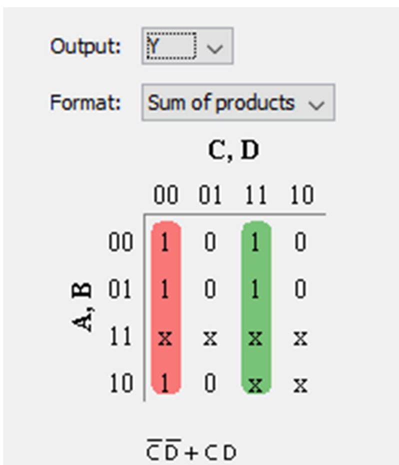
3.A.3

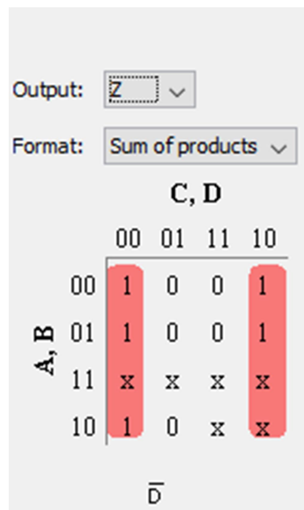


A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	$\boxed{0}$
1	0	1	0	x	x	x	x
1	0	1	1	x	x	x	x
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x


Output: W
Format: Sum of products
C, D
A, B
$BD + BC + A$


Output: X
Format: Sum of products
C, D
A, B
$\overline{B}D + \overline{B}C + B\overline{C}\overline{D}$


Output: Y
Format: Sum of products
C, D
A, B
$\overline{C}\overline{D} + CD$


Output: Z
Format: Sum of products
C, D
A, B
$\overline{D}$

Laboratory 5

Combinational Circuits-I
Adders and Subtractors

1. Introduction and Purpose of Experiment

   Students will learn to design, simulate and implement adders and subtractors.

2. Aim and Objectives

   Aim: To use adders and subtractors and perform mathematical operations on binary numbers

   Objectives: At the end of this lab, the student will be able to

   - Add and subtract 4-bit binary numbers
   - Use adder circuits to perform subtraction

3. Experimental Procedure

   a. Write truth tables, Boolean expressions and circuit diagrams for the following combinational circuits:
      I. Half and full adder
      II. Half and full subtractor

   b. Using 4 full adders and other additional circuitry, design a circuit capable of adding and subtracting two 4-bit numbers.

   c. Use Logisim to simulate the circuit designed above.

   d. Implement the circuit and show the output to the course leader.

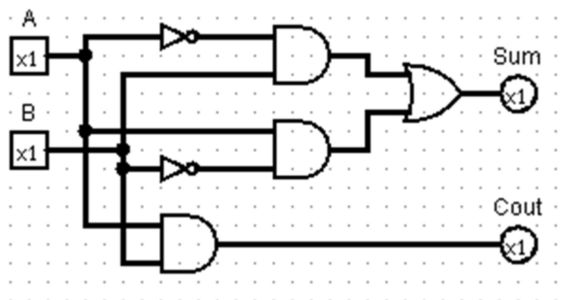
   e. Show in detail how a full adder can be implemented using two half adders.

Your document should include:

- Handwritten truth tables, expressions and circuit diagrams for the combinational circuits in 3(a).
- Design of the circuit
- Logisim screenshots of the designed circuit
- Answer to 3(e)

Solutions:

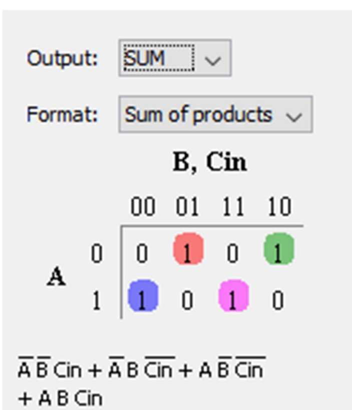
Half - Adder:

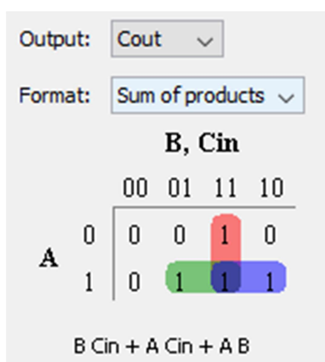


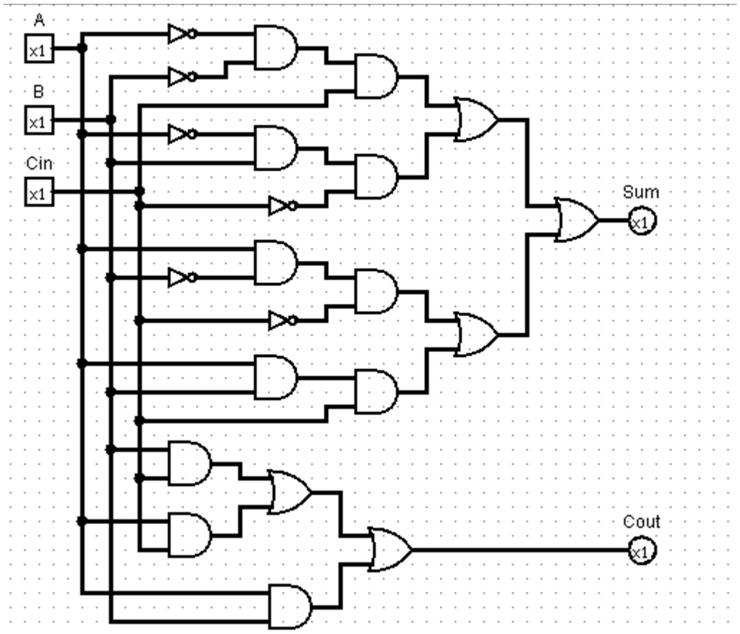
A	B	SUM	Cout
0	0	0	0
0	1	1	0
1	0	1	$\boxed{0}$
1	1	0	1

Full - Adder:

A	B	Cin	SUM	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

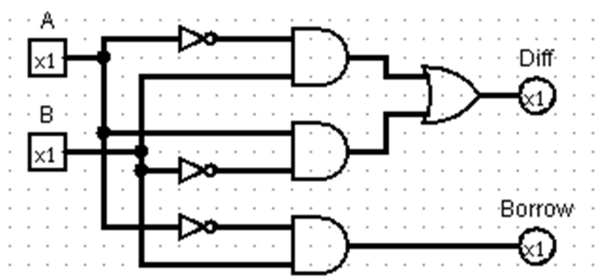


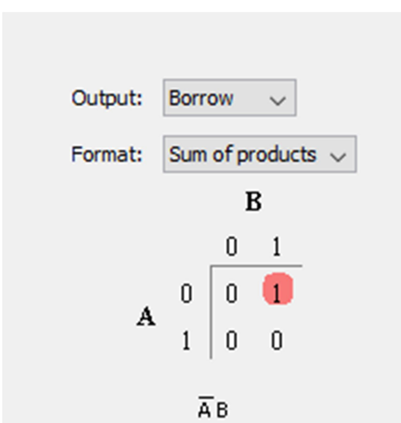


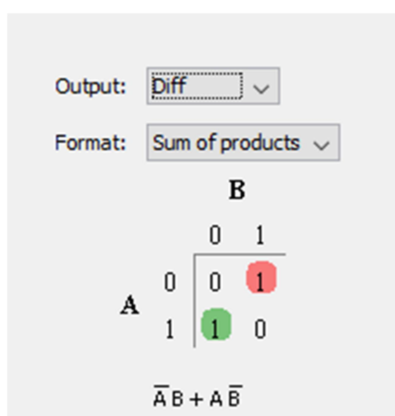


Half- Subtractor :

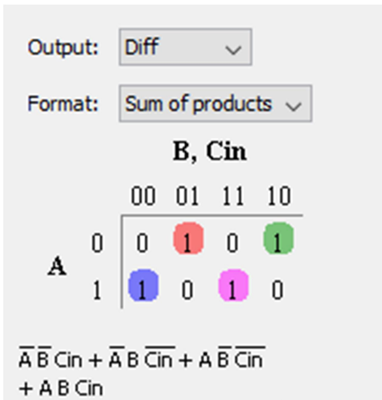
A	B	Diff	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

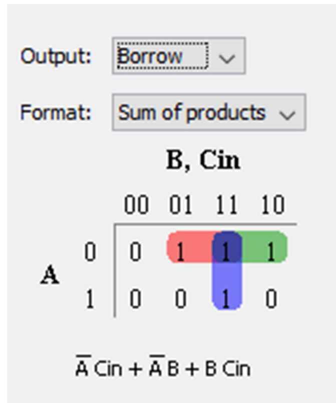




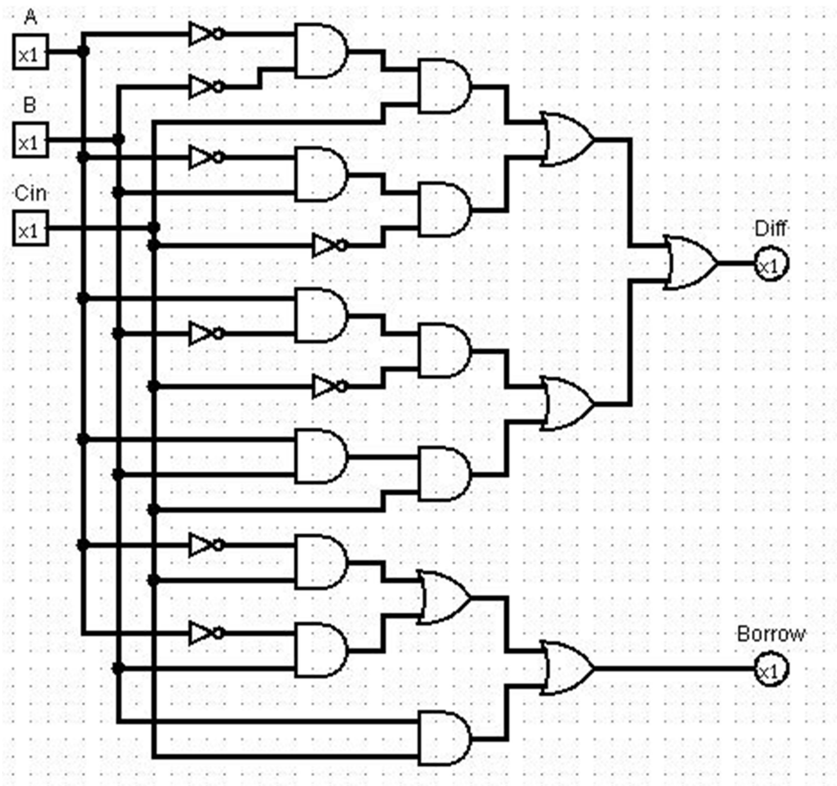


Full - Subtractor :





A	B	Cin	Diff	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	$\boxed{1}$



A
x1
B
x1
Cin
x1
Diff
x1
Borrow
x1

RAMAIAH UNIVERSITY OF APPLIED SCIENCES

Laboratory 6

Combinational Circuits-II
Multiplexers and Demultiplexers

1. Introduction and Purpose of Experiment

   Students will learn to design, simulate and implement circuits using Multiplexers and Demultiplexers.

2. Aim and Objectives

   Aim: To verify functionality of Mux and Demux and use Multiplexer to implement Boolean functions

   Objectives: At the end of this lab, the student will be able to

   - Verify the functionality of Mux and Demux
   - Use Multiplexers to implement given Boolean Functions

3. Experimental Procedure

   a. Write truth tables and block diagrams for
      I. 4 to 1 Multiplexer
      II. 8 to 1 Multiplexer
      III. 1 to 4 Demultiplexer
      IV. 1 to 8 Demultiplexer

   b. Construct the circuits for 3 (a) (I) to 3 (a) (IV) above using appropriate ICs. Verify the functionality and show the output to the course leader

   c. Use Logisim to simulate the circuits designed above.

   d. Using an example, show how any Boolean Expression in SoP form can be implemented using a Multiplexer. Simulate the same using Logisim.

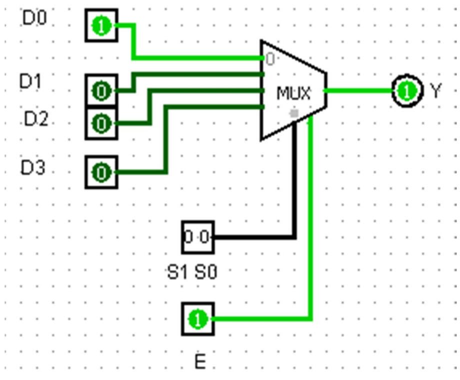
Your document should include:

- Handwritten truth tables and block diagrams for the circuits in 3(a).
- Logisim screenshots of all the Multiplexers and Demultiplexers.
- Answer to 3(d)
- Logisim screenshots for 3(d)

3.a

1)4x1 Multiplexer

Block diagram

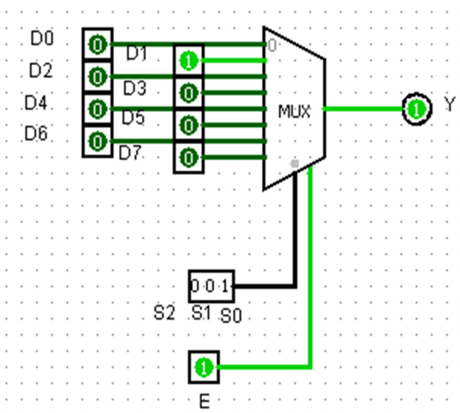


truth table

S1	S0	I3	I2	I1	I0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

2)8X1 Multiplexer

Block diagram



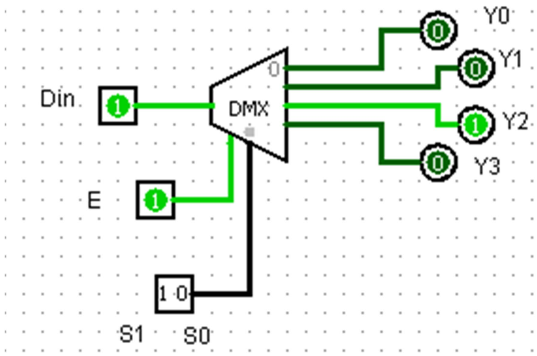
truth table

S2	S1	S0	I7	I6	I5	I4	I3	I2	I1	I0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

3)1X4 Demultiplexer

Block diagram

truth table

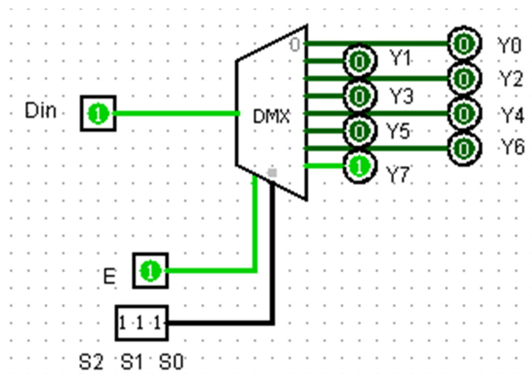


S1	S0	D	Y3	Y2	Y1	Y0
0	0	1	0	0	0	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	1	0	0	0

4)1X8 Demultiplexer

Block diagram

truth table

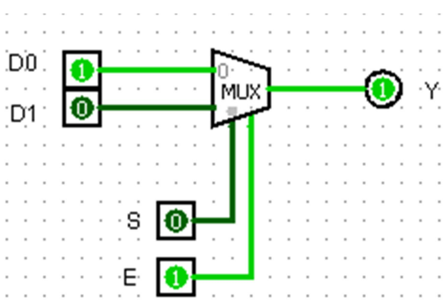


S2	S1	S0	D	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
0	0	0	1	0	0	0	0	0	0	0	1
0	0	1	1	0	0	0	0	0	0	1	0
0	1	0	1	0	0	0	0	0	1	0	0
0	1	1	1	0	0	0	0	1	0	0	0
1	0	0	1	0	0	0	1	0	0	0	0
1	0	1	1	0	0	1	0	0	0	0	0
1	1	0	1	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0

4)2X1 Multiplexer

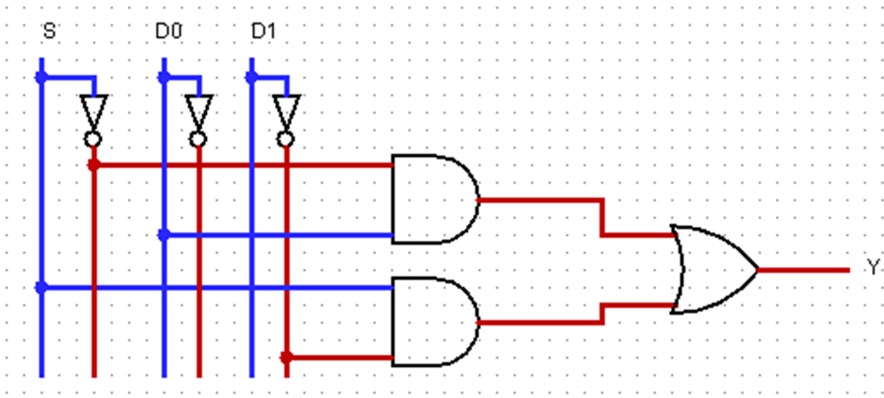
Block diagram

truth table



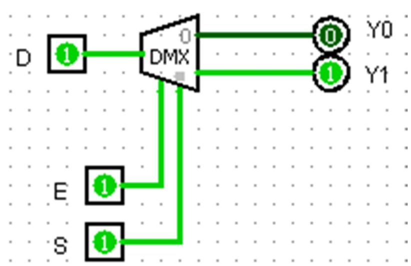
S	D1	D0
0	0	1
1	1	0

Circuit diagram



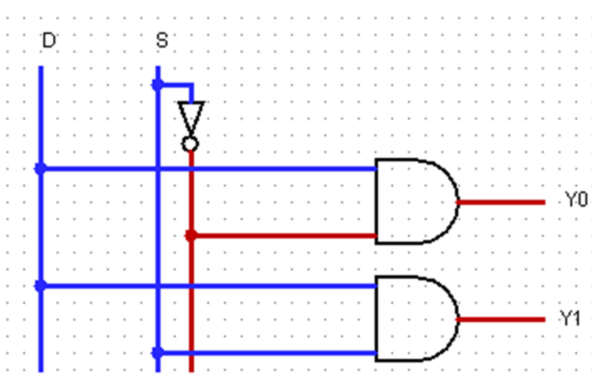
5)1X2 Demultiplexer

Block diagram truth table



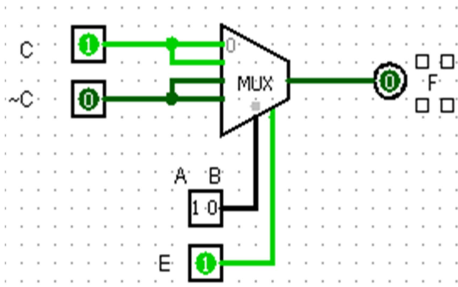
I	S	Y1	Y0
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	0

Circuit diagram



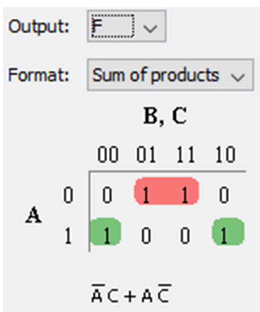
3.d

Block diagram of 4X1 Multiplexer truth table



A	B	F
0	0	1
0	1	1
1	0	0
1	1	0

Sum of product expression using multiplexer

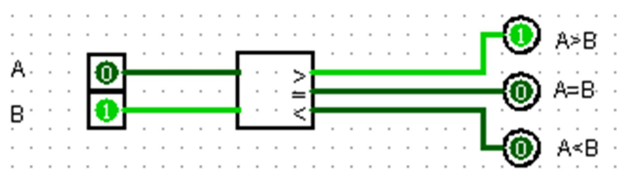


Comparators

1) 1-bit comparator

Block diagram

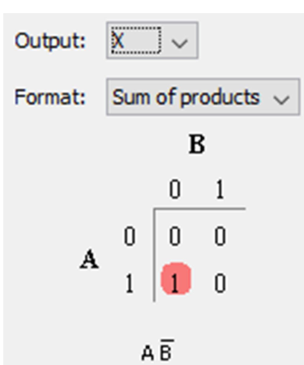
truth table



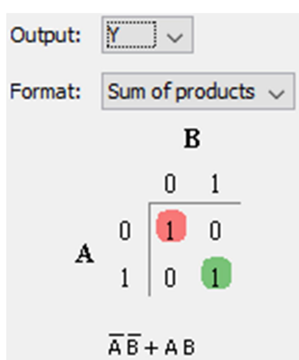
A	B	X	Y	Z
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

Minimized expression:

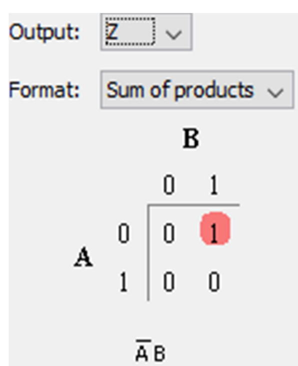
X=A>B



Y=(A=B)

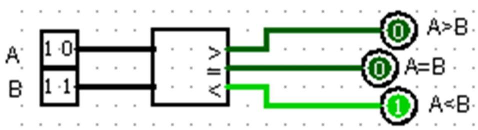


Z=A<B



2) 2-bit comparator

Block diagram

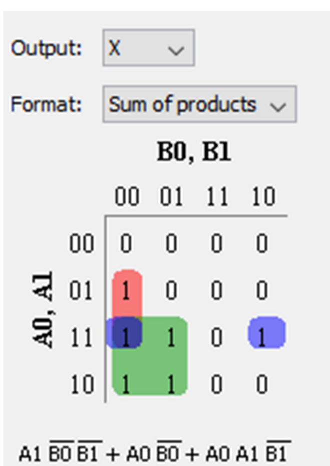


Truth table

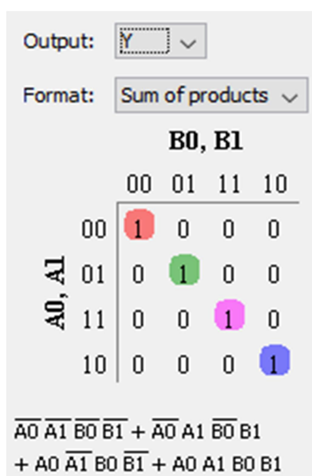
A0	A1	B0	B1	X	Y	Z
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

Minimized expression:

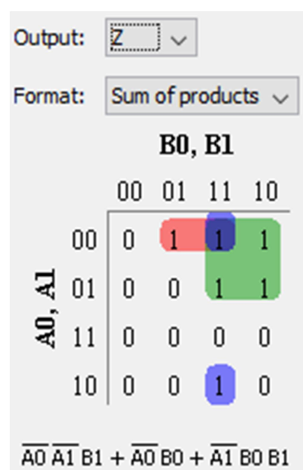
X=A>B



Y=(A=B)



Z=A<B



Laboratory 7

Combinational Circuits-III
Decimal to BCD Encoder and Decoders

1. Introduction and Purpose of Experiment

   Students will learn to design and implement a circuit for Decimal to BCD Encoder.

2. Aim and Objectives

   Aim: Design and implement a circuit for Decimal to BCD Encoder

   Objectives: At the end of this lab, the student will be able to

   - Develop a circuit for Decimal to BCD Encoder
   - Understand the basics of Decoders

3. Experimental Procedure

   a. Write truth table and block diagram for Decimal to BCD Encoder

   b. Construct the circuits for Decimal to BCD Encoder using appropriate ICs. Verify the functionality and show the output to the course leader

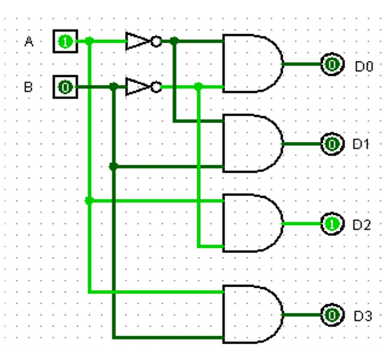
   c. Using an example, describe how a decoder can be implemented using a Demultiplexer.

Your document should include:

- Handwritten truth table and block diagrams for the circuit in 3(a).
- Answer to 3(c)

2:4 decoders

Circuit diagram

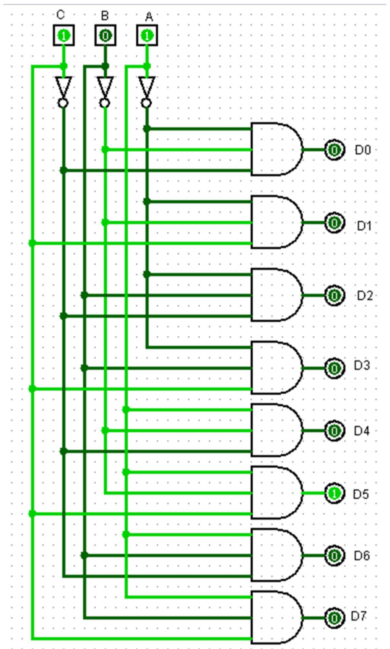


truth table

A	B	D3	D2	D1	D0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

3:8 decoders

Circuit diagram

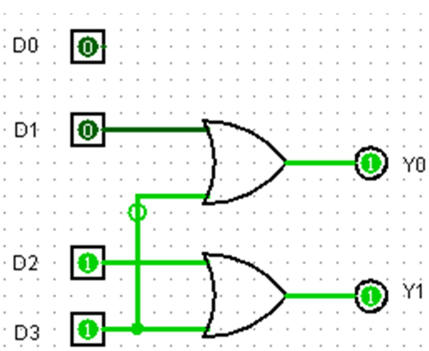


truth table

A	B	C	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

4:2 encoders

Circuit diagram

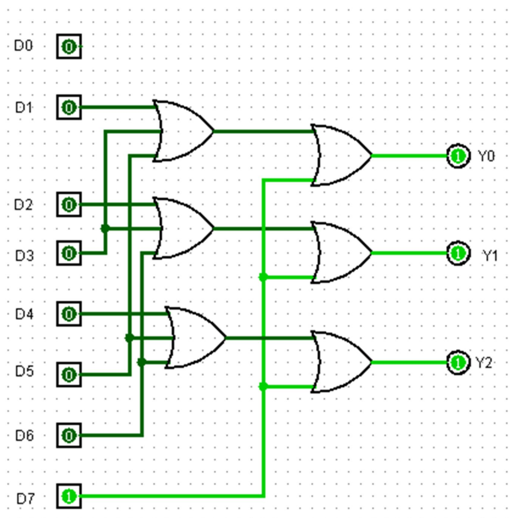


truth table

Y1	Y0	D3	D2	D1	D0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

8:3 decoders

Circuit diagram

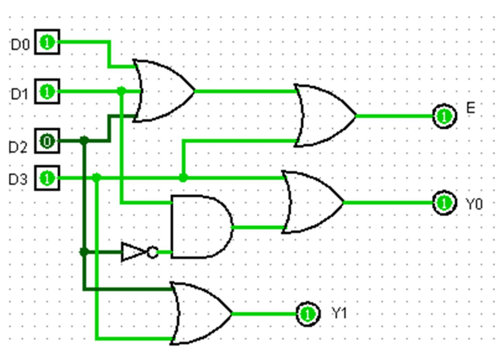


truth table

Y2	Y1	Y0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

4:2 priority encoders

Circuit diagram

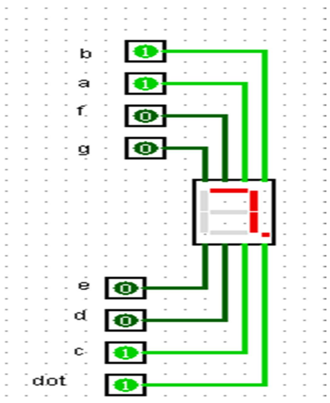


truth table

Y1	Y0	E	D3	D2	D1	D0
0	0	0	0	0	0	0
0	0	1	0	0	0	1
0	1	0	0	0	0	0
0	1	1	0	0	1	x
1	0	0	0	0	0	0
1	0	1	0	1	x	x
1	1	0	0	0	0	0
1	1	1	1	x	x	x

Bcd to 7 segment display

Block diagram



truth table

A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1

RAMAIAH
UNIVERSITY
OF APPLIED SCIENCES