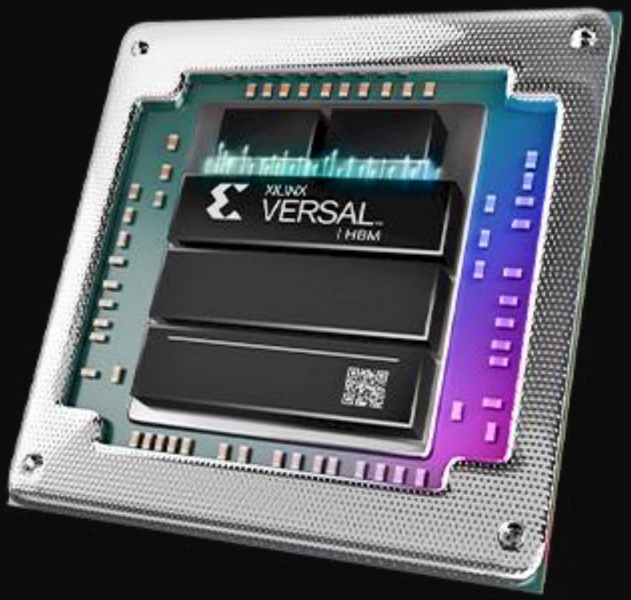
EC204

Digital System Design Lab

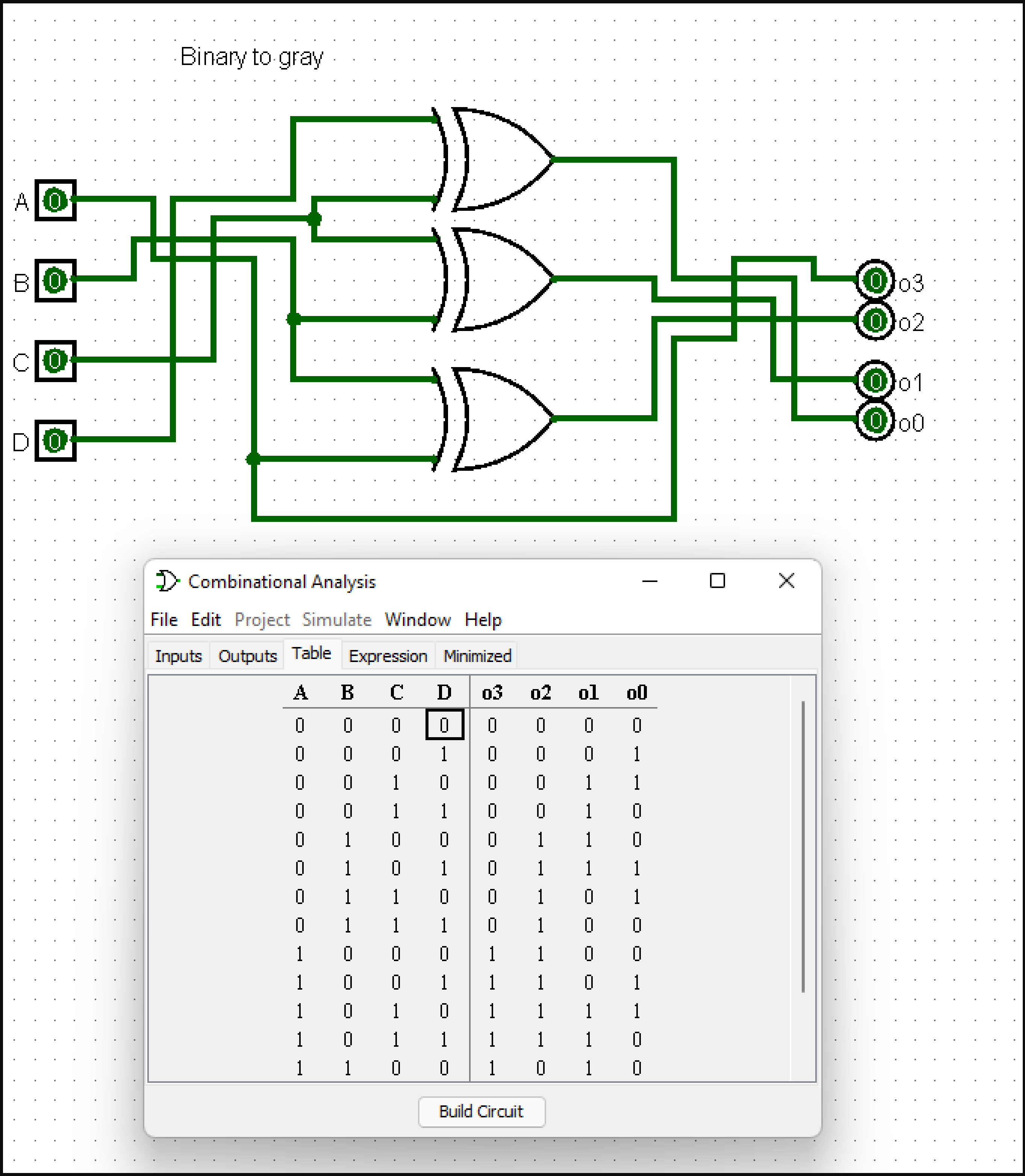
Lab – 3



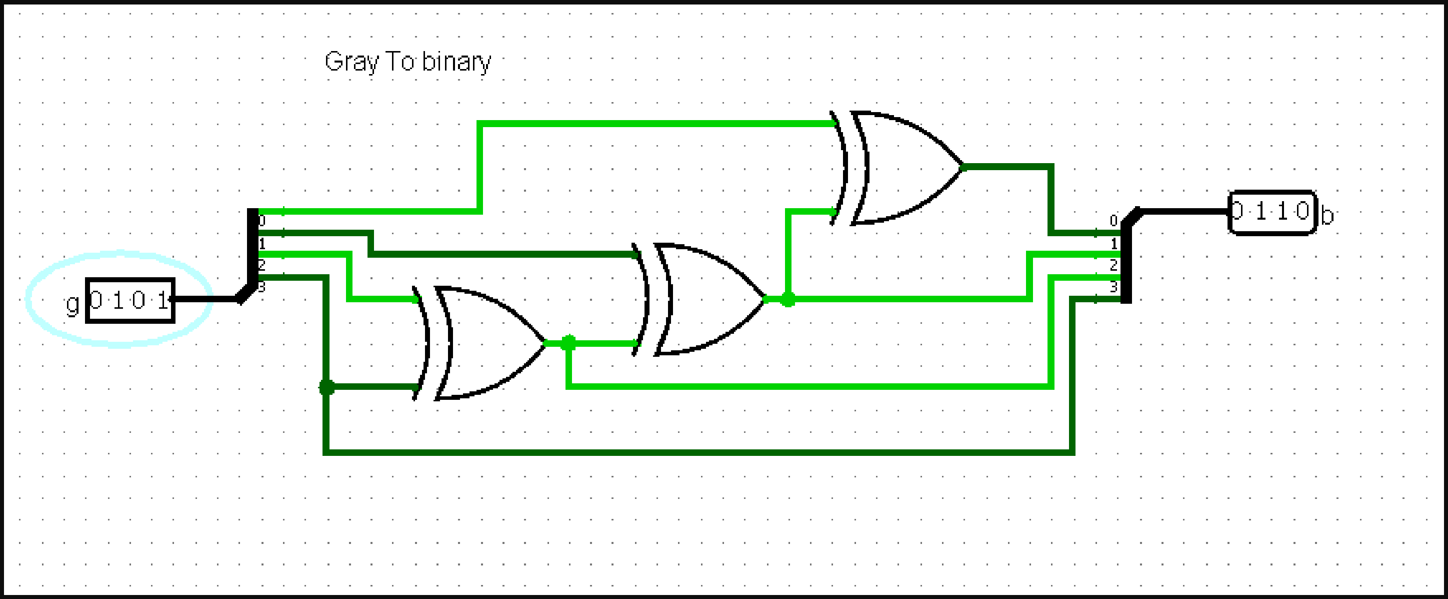
Utkarsh R Mahajan

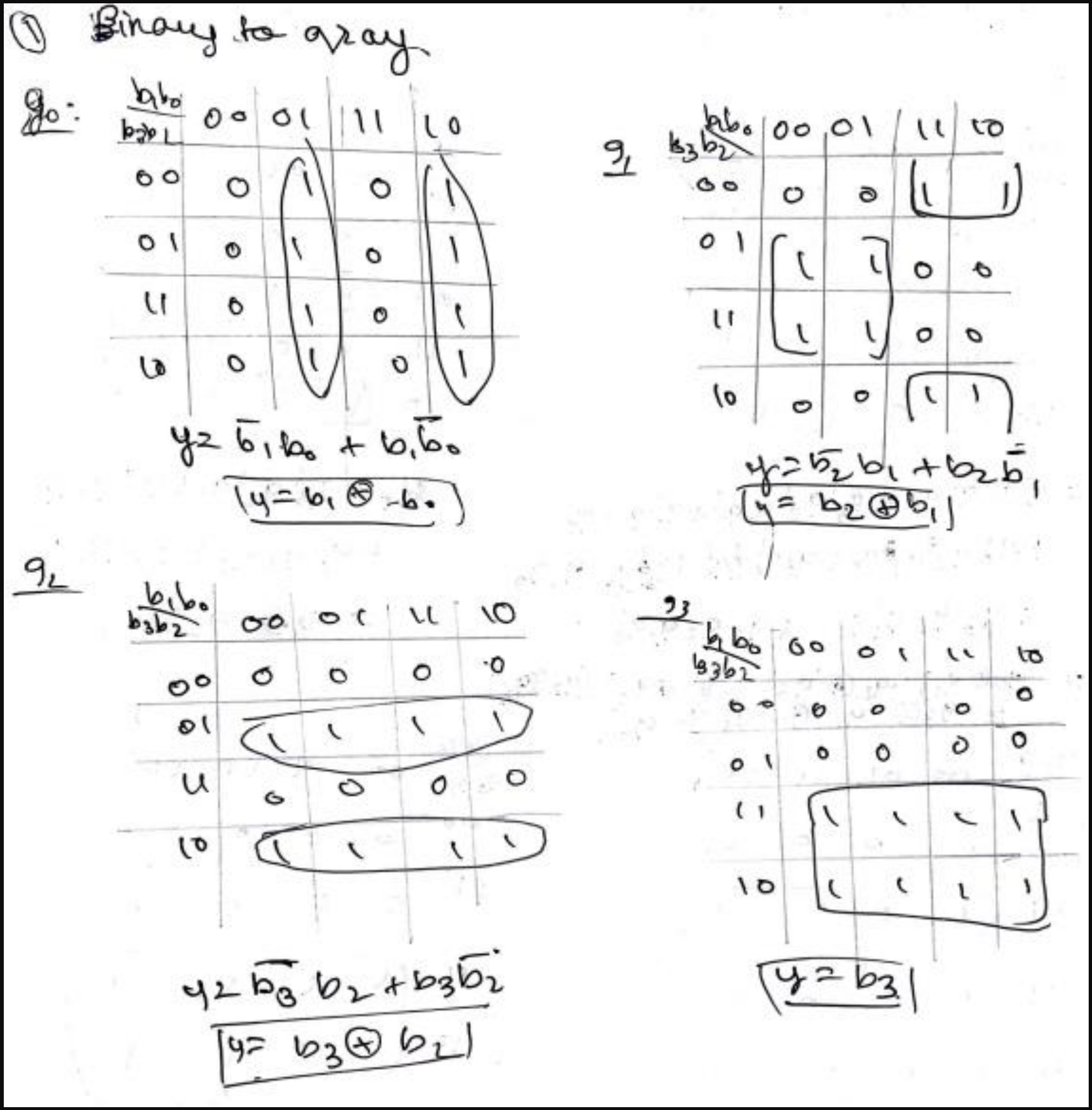
201EC164

1] Design a circuit to convert 4 bit Binary to Gray and 4 bit Gray to Binary Code using XOR gates


Binary to gray
A
B
C
D
o3
o2
o1
o0
Combinational Analysis
File Edit Project Simulate Window Help
Inputs Outputs Table Expression Minimized
Build Circuit

A	B	C	D	o3	o2	o1	o0
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 Binary to gray

g_0:

b_3b_2 \ b_1b_0	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	1	0	1
10	0	1	0	1

$y = \bar{b}_1 b_0 + b_1 \bar{b}_0$

$\boxed{y = b_1 \oplus b_0}$

g_1

b_3b_2 \ b_1b_0	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	1	1	0	0
10	0	0	1	1

$y = \bar{b}_2 b_1 + b_2 \bar{b}_1$

$\boxed{y = b_2 \oplus b_1}$

g_2

b_3b_2 \ b_1b_0	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

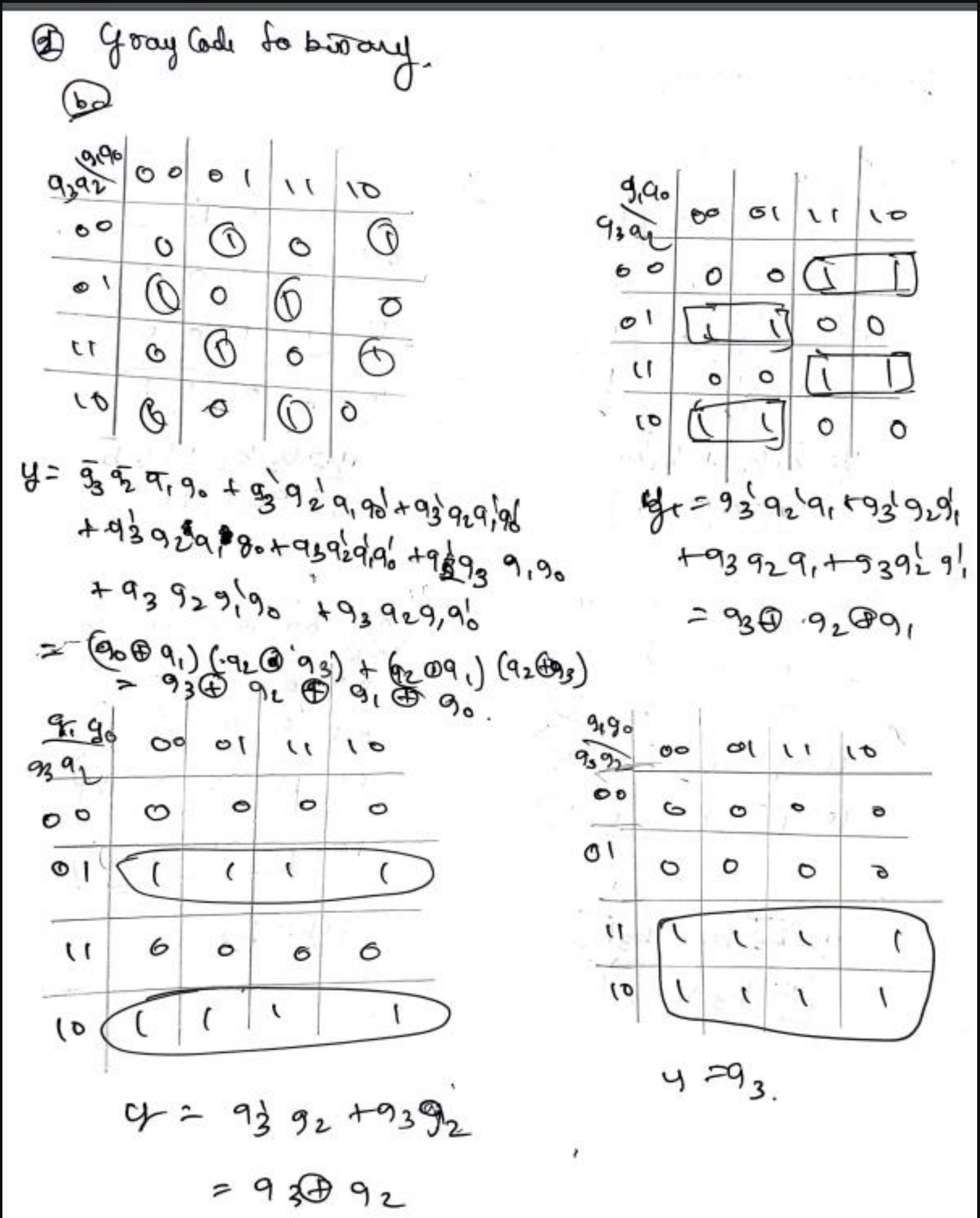
$y = \bar{b}_3 b_2 + b_3 \bar{b}_2$

$\boxed{y = b_3 \oplus b_2}$

g_3

b_3b_2 \ b_1b_0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

$\boxed{y = b_3}$



② Gray Code to binary

(b)

$g_3g_2 \backslash g_1g_0$	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

$$y = \bar{g}_3\bar{g}_2 g_1 g_0 + g_3' g_2' g_1 g_0' + g_3' g_2 g_1' g_0'$$
$$+ g_3' g_2 g_1 g_0 + g_3 g_2' g_1' g_0' + g_2' g_3 g_1 g_0$$
$$+ g_3 g_2 g_1' g_0 + g_3 g_2 g_1 g_0'$$
$$= (g_0 \oplus g_1)(g_2 \odot g_3) + (g_2 \odot g_1)(g_2 \oplus g_3)$$
$$= g_3 \oplus g_2 \oplus g_1 \oplus g_0.$$

$g_3g_2 \backslash g_1g_0$	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	0	0	1	1
10	1	1	0	0

$$y = g_3' g_2' g_1 + g_3' g_2 g_1'$$
$$+ g_3 g_2 g_1 + g_3 g_2' g_1'$$
$$= g_3 \oplus g_2 \oplus g_1$$

$g_3g_2 \backslash g_1g_0$	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

$$y = g_3' g_2 + g_3 g_2'$$
$$= g_3 \oplus g_2$$

$g_3g_2 \backslash g_1g_0$	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

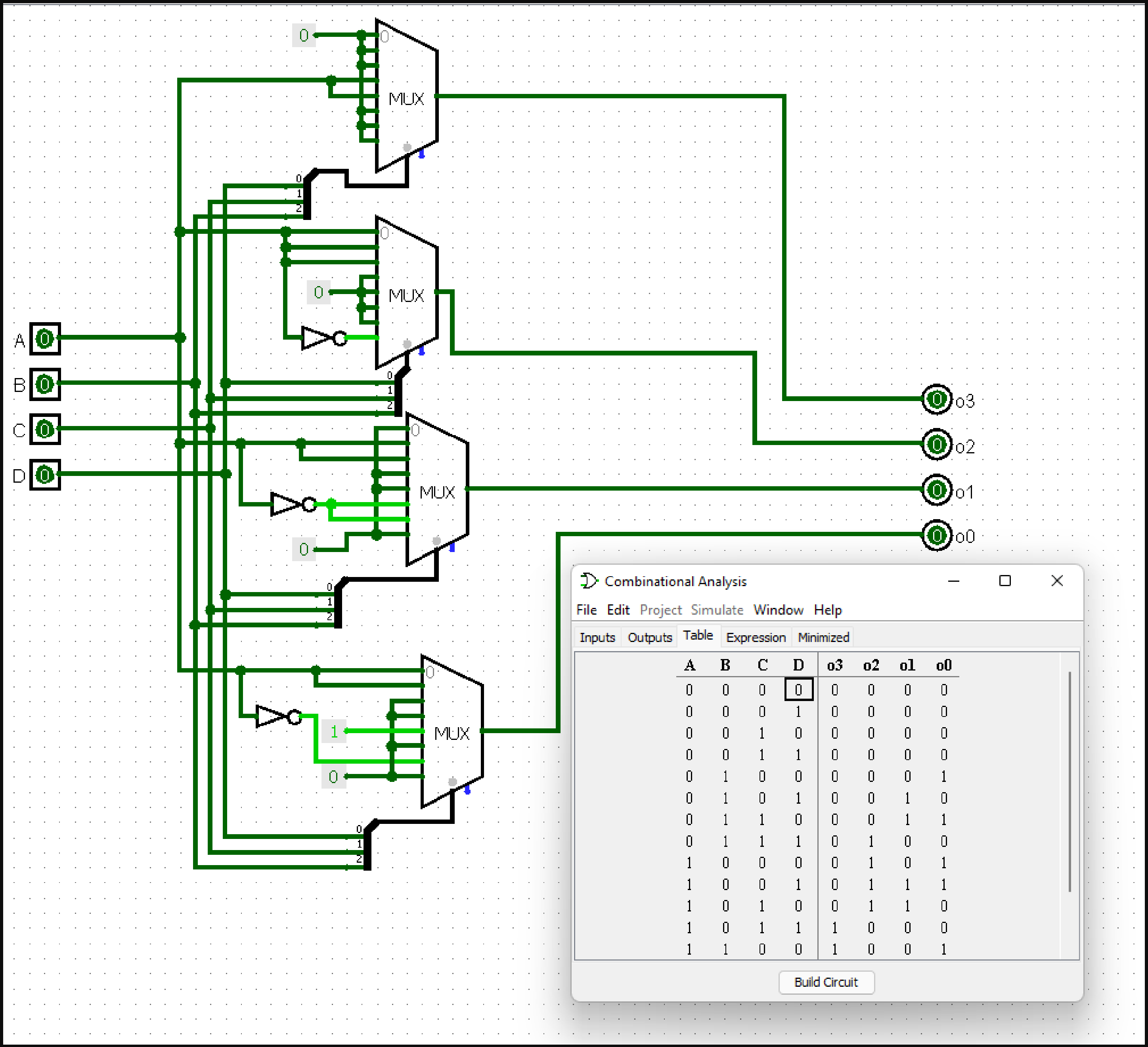
$$y = g_3.$$

Result: Results Obtained can be seen via the analyzed circuit Table, which matches the initial Truth table for the convertors.

2]Design an Excess-3 to BCD code converter and implement using 8:1 multiplexer. Use the multiplexer available in Plexers library in logisim.

We take input excess-3 as ABCD (A is MSB)

We can send the input BCD Via Selector and then A via Input I.


A
B
C
D
MUX
MUX
MUX
MUX
o3
o2
o1
o0
Combinational Analysis
File Edit Project Simulate Window Help
Inputs Outputs Table Expression Minimized
Build Circuit

A	B	C	D	o3	o2	o1	o0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	0	0	1	1
0	1	1	1	0	1	0	0
1	0	0	0	0	1	0	1
1	0	0	1	0	1	1	1
1	0	1	0	0	1	1	0
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	1

Result: the table obtained via analyzing circuit matches the required result.

3]Design a circuit to implement the following functions using (a) 4 to 16 decoder (b) 4 to 1 Multiplexer. Use the Multiplexer and decoder available in Plexers library in logisim.

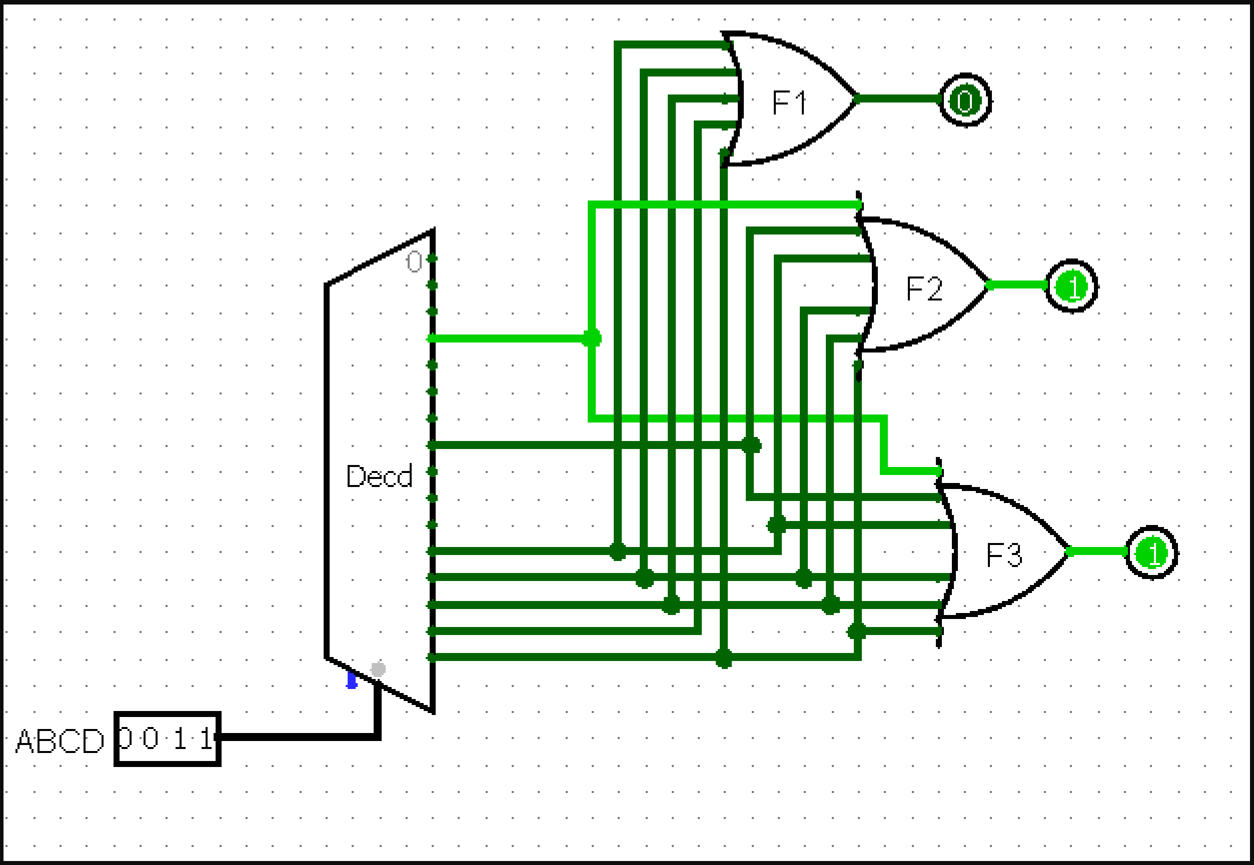
$F_1(A,B,C,D) = \sum m\ (11,12,13,14,15)$

$F_2(A,B,C,D) = \sum m\ (3,7,11,12,13,15)$

$F_3(A,B,C,D) = \sum m\ (3,7,12,13,14,15)$

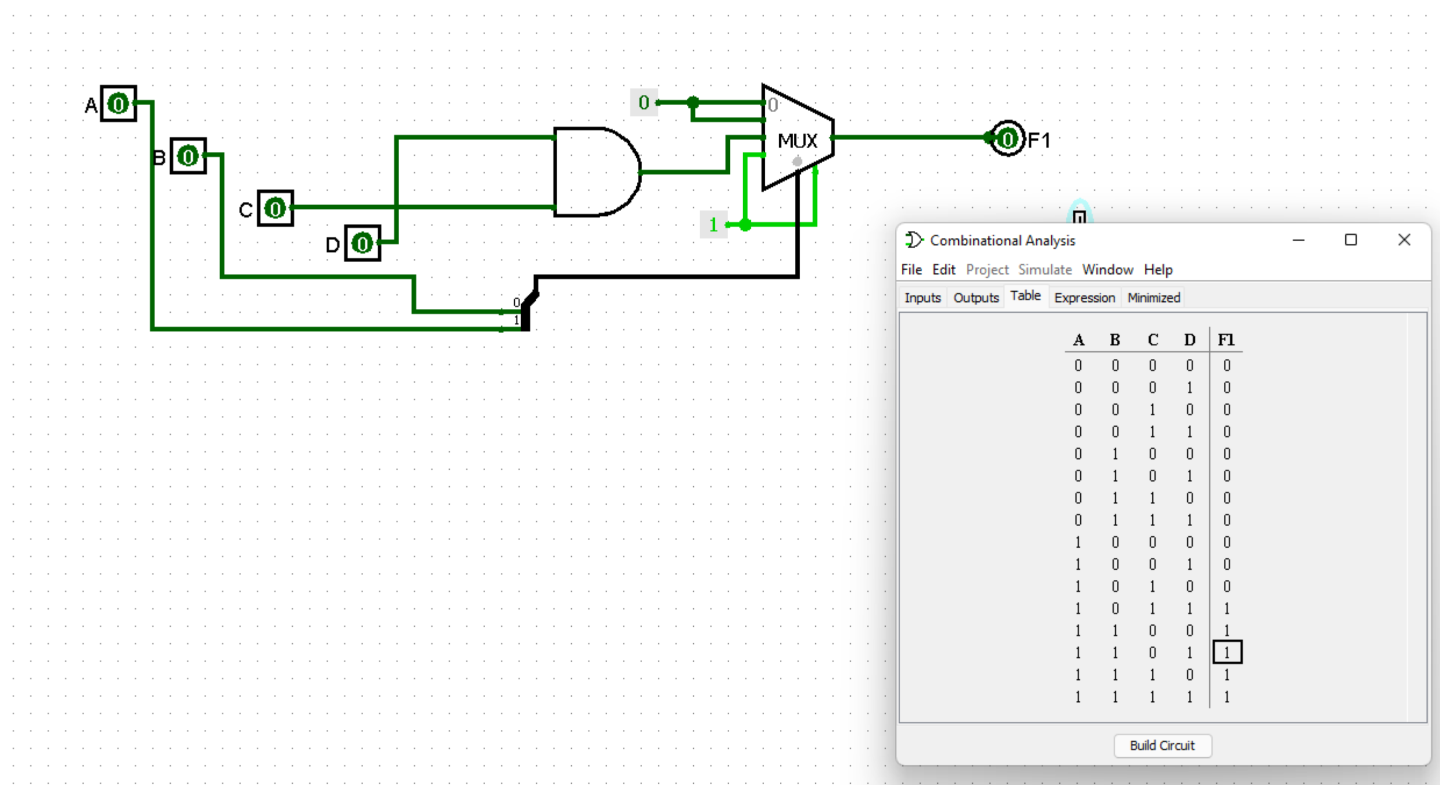
a) Using decoder

We can take input as ABCD while using OR gates for getting the correct output for required functions.

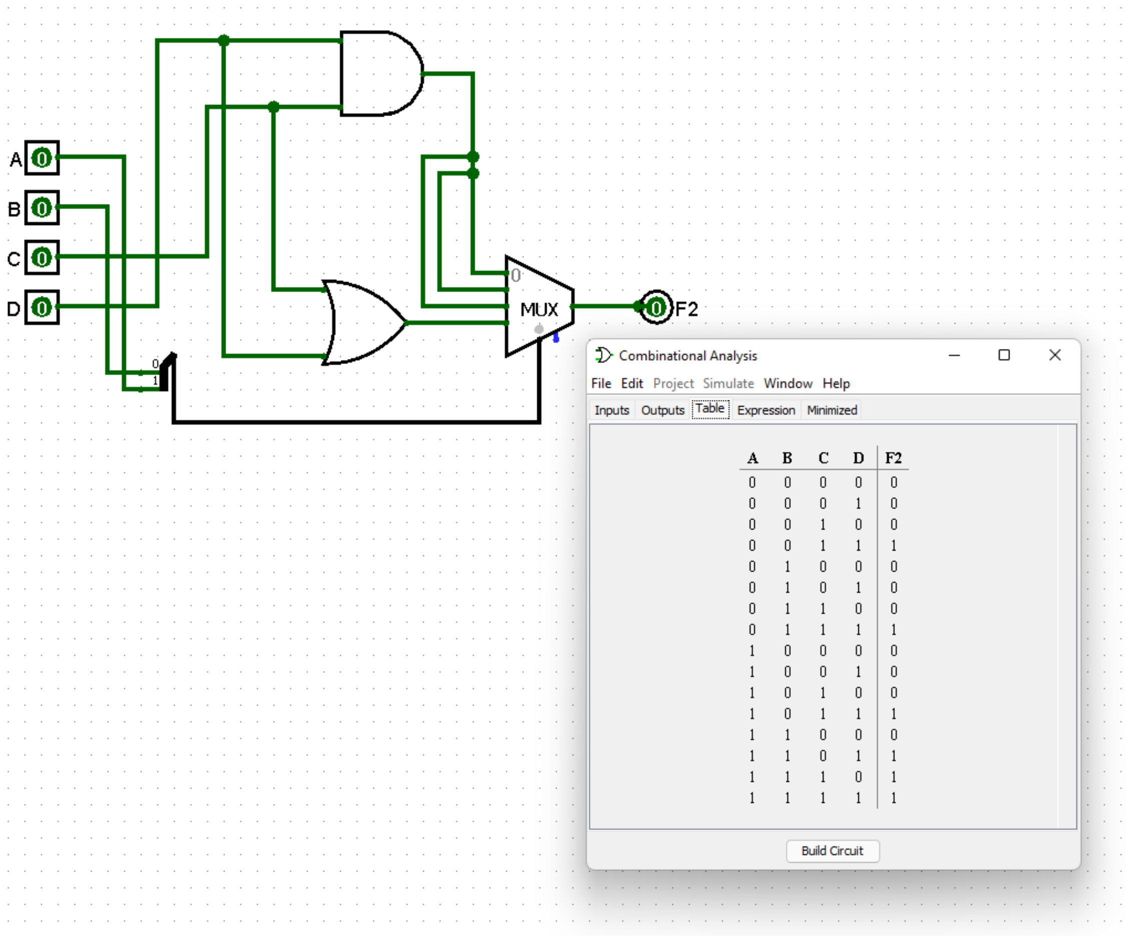


(b) 4 to 1 Multiplexer

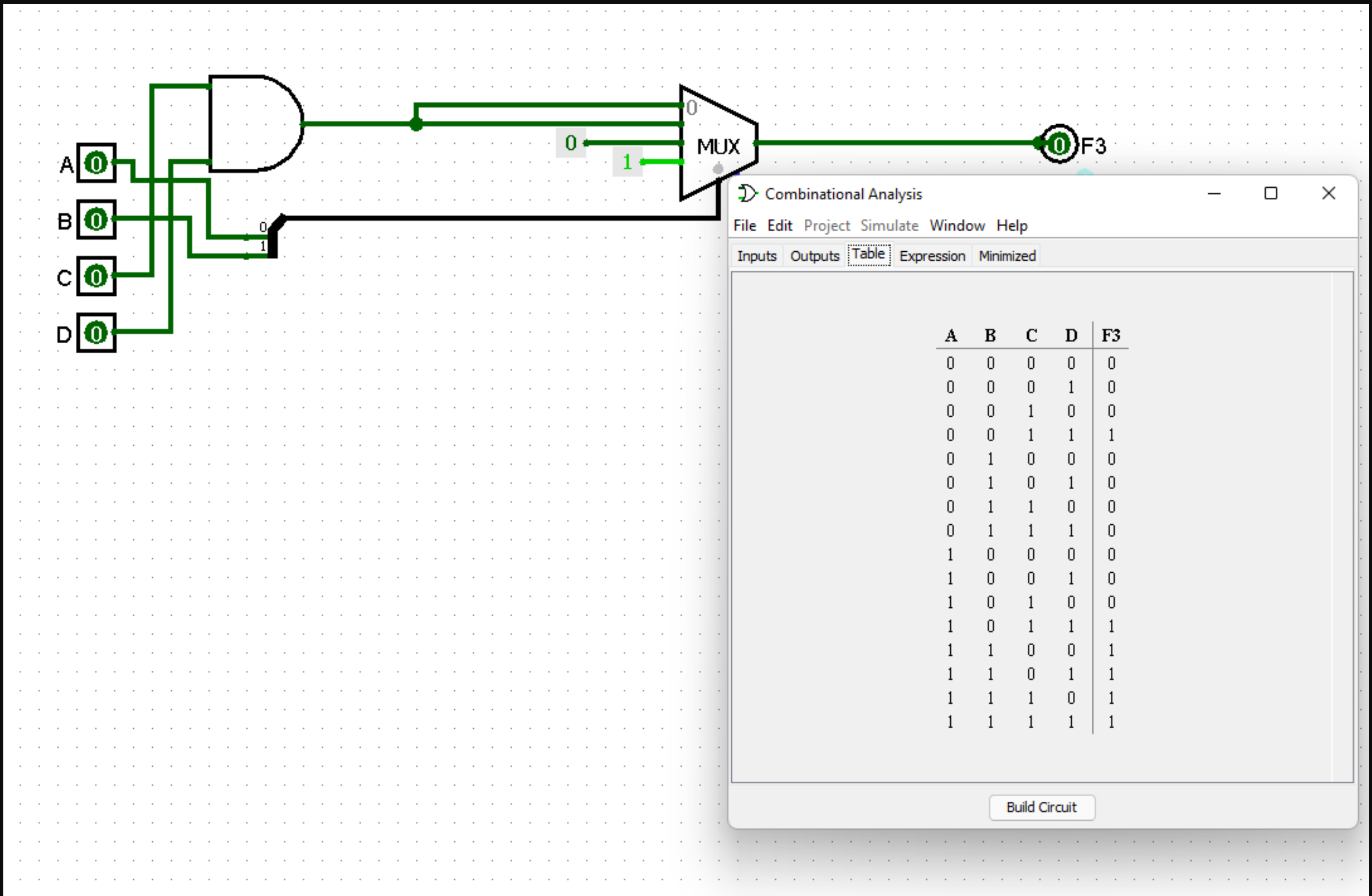
Taking AB as input in Selector for MUX and C and D at inputs I0-I4. We can obtain the following diagram.


A
B
C
D
0
1
MUX
F1
Combinational Analysis
File Edit Project Simulate Window Help
Inputs Outputs Table Expression Minimized
Build Circuit

A	B	C	D	F1
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	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1


A
B
C
D
0
1
MUX
F2
Combinational Analysis
File Edit Project Simulate Window Help
Inputs Outputs Table Expression Minimized
Build Circuit

A	B	C	D	F2
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
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

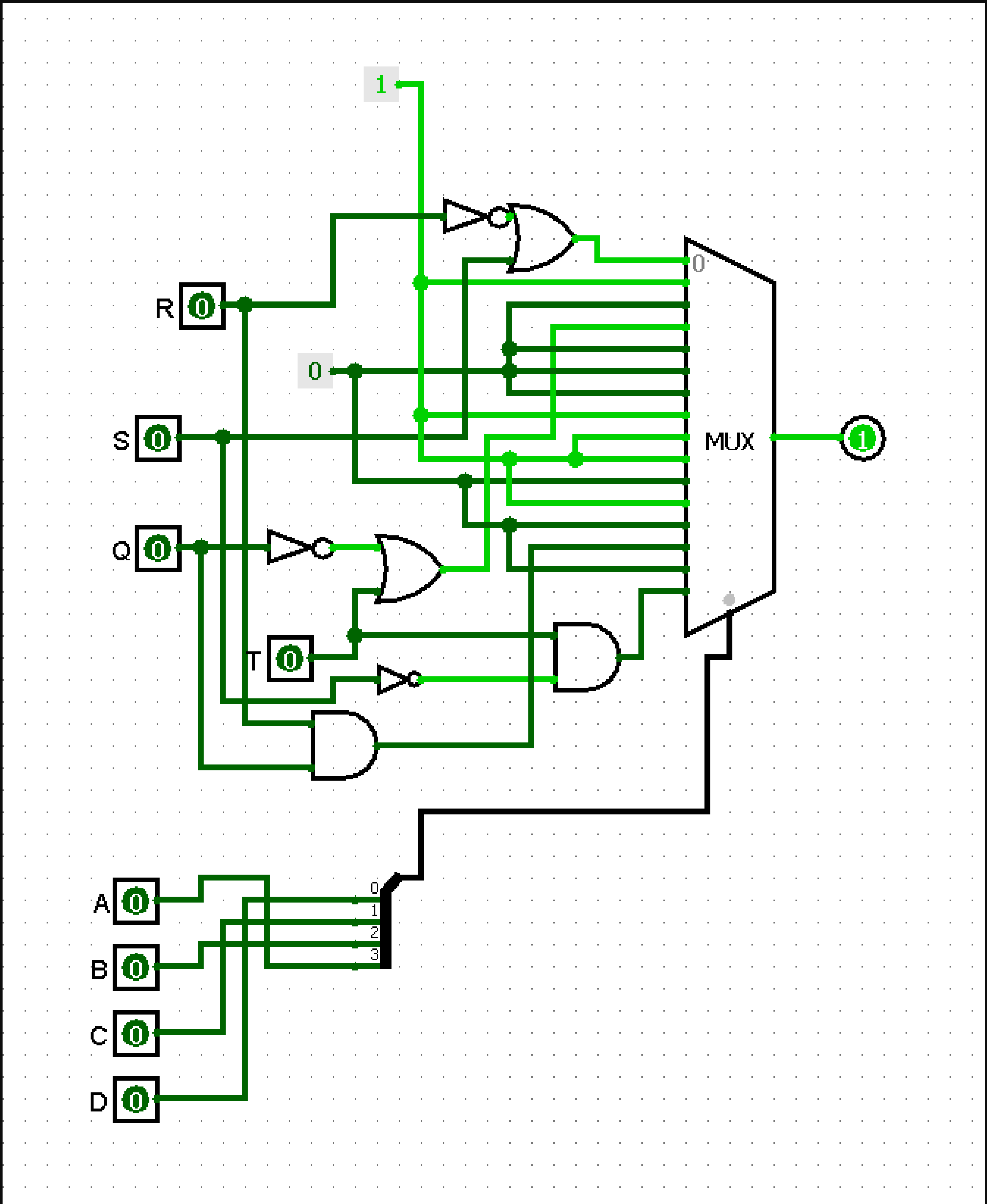

A
B
C
D
0
0
0
0
0
1
0
0
1
MUX
0
F3
Combinational Analysis
File Edit Project Simulate Window Help
Inputs Outputs Table Expression Minimized
Build Circuit

A	B	C	D	F3
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	0
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

Results:from the Logic table obtained from circuit analysis, we can see that it matches with the given functions.

4]Design a circuit to implement the following function using 16 to 1 Multiplexer. Use the Multiplexer available in Plexers library in logisim. F = A'B'C'D'(R'+S) + AB'C'D' + A'B'C'D + ABC'DQR + AB'C'D + A'B'CD (Q'+T) + ABCDS'T + AB'CD + A'BCD

Taking selection input as ABCD for Mux, and using logic gates for setting up specific inputs, we can obtain our circuit as follows.



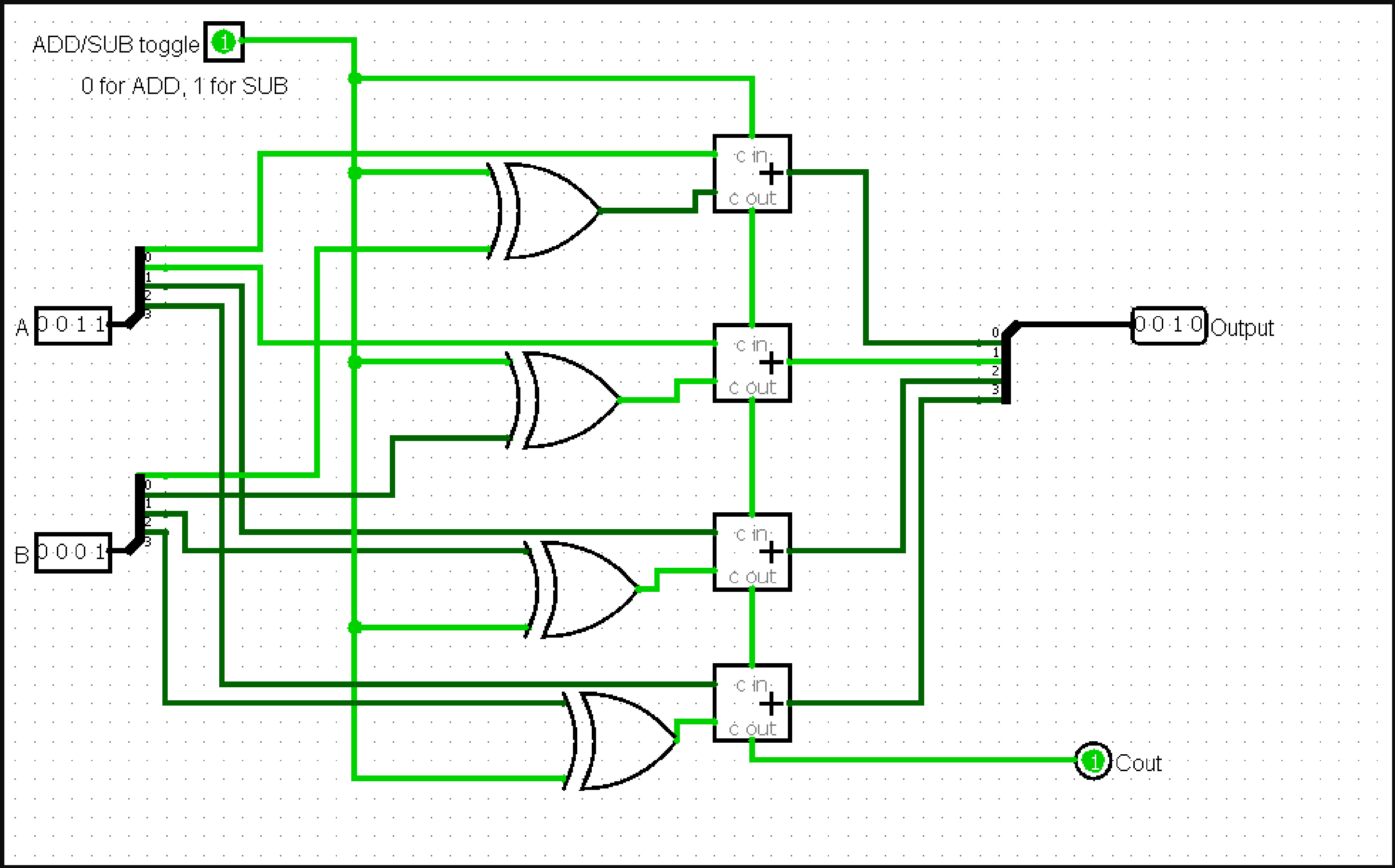
Results:

We obtain the result as required.

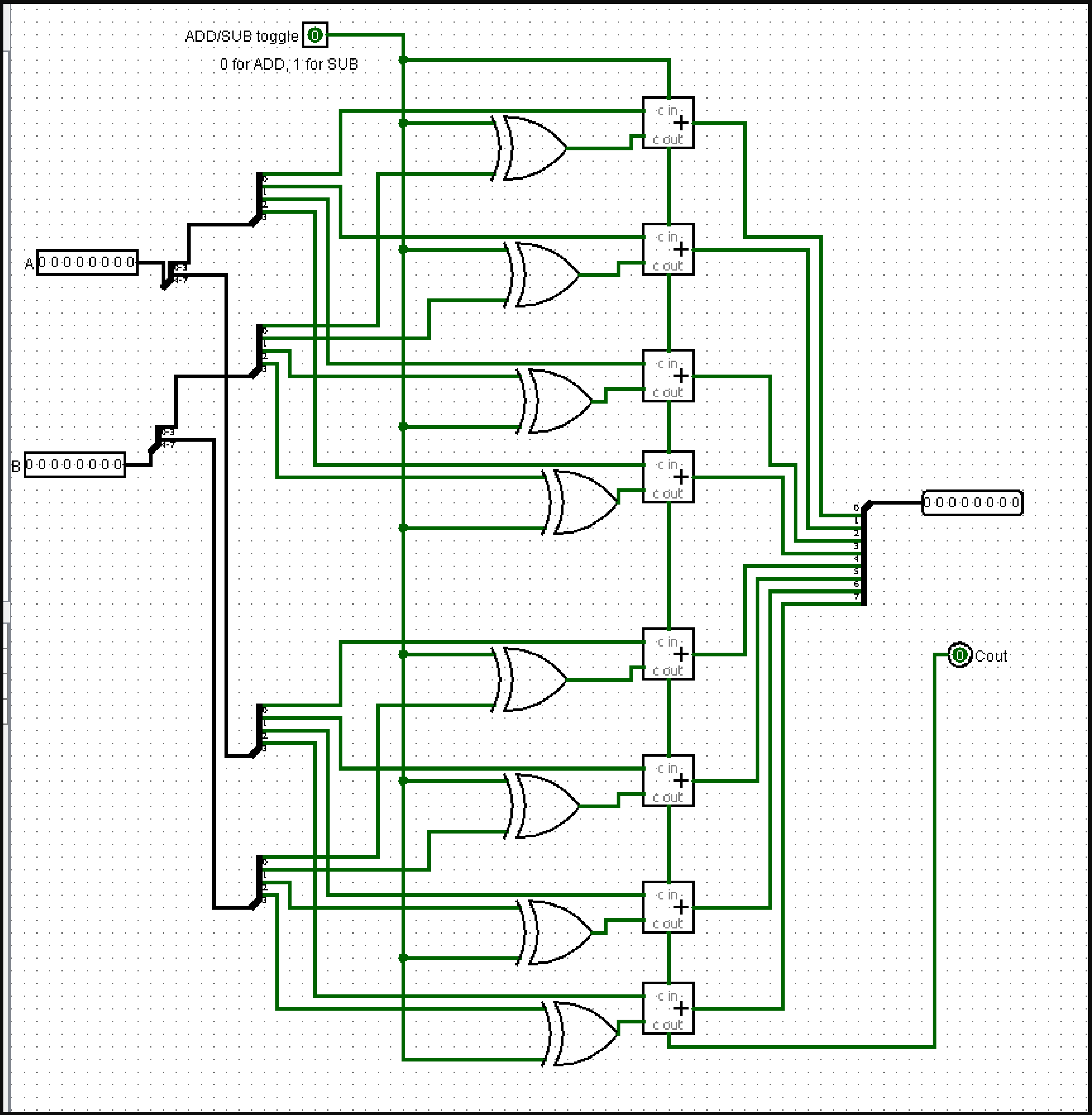
5]Implement (a) 4 bit controllable adder/subtractor (b) 8 bit controllable adder/subtractor. Use the Adder available in Arithmetic library in Logisim

By settings a toggle for Subtraction or Addition, and using xor gates with b input and the toggle input. Connecting them with respective A’s bit to full adders which are in series. Connecting consecutive Couts to each other we can obtain the following circuit.

(a) 4bit adder/subtractor



(b) Similarly for 8bit adder/subtractor


ADD/SUB toggle
0
0 for ADD, 1 for SUB
A
00000000
B
00000000
c in
c out
00000000
0
Cout