RAMAIAH UNIVERSITY OF APPLIED SCIENCES

ASSIGNMENT

Course Code	19CSC204A
Course Name	Logic Design
Programme	B. Tech
Department	Computer Science & Engineering
Faculty	Faculty of Engineering Technology

Name of the Student	SUBHENDU MAJI
Reg. No	18ETCS002121
Semester/Year	3RD / 2019
Course Leader/s	Mr. Narasimha Murthy K. R. and Mr. V. Deepak

Declaration Sheet				
Student Name	SUBHENDU MAJI			
Reg. No	18ETCS002121			
Programme	B. Tech		Semester/Year	3rd / 2019
Course Code	19CSC204A			
Course Title	Logic Design			
Course Date		To		
Course Leader	Mr. Narasimha Murthy K. R. and Mr. V. Deepak			

Declaration

The assignment submitted herewith is a result of my own investigations and that I have conformed to the guidelines against plagiarism as laid out in the Student Han dbook. All sections of the text and results, which have been obtained from other sources, are fully referenced. I understand that cheating and plagiarism constitute a breach of University regulations and will be dealt with accordingly.

Signature of the Student		Date	
Submission date stamp (by Examination & Assessment Section)			

Signature of the Course Leader and date	Signature of the Reviewer and date

Contents

Faculty of Engineering and Technology			
Ramaiah University of Applied Sciences			
Department	Computer Science and Engineering	Programme	B. Tech. in CSE
Semester/Batch	3rd Semester/2018		
Course Code	19CSC204A	Course Title	Logic Design
Course Leaders	Mr. Narasimha Murthy K. R. and Mr. V. Deepak		

Sections	Marking Scheme		Marks		
			Max Marks	First Examiner Marks	Moderator
Question 1					
	A	**Introduction**	**01**		
	B	**Circuit Design**	**04**		
	C	**Simulation of Circuit**	**03**		
	D	**Circuit using NAND Gates**	**02**		
		Question 1 Max Marks	**10**		
Question 2					
	A	**Introduction**	**01**		
	B	**Circuit Design**	**03**		
	C	**Simulation of Circuit**	**02**		
	D	**Circuit using Ex-OR Gates**	**04**		
		Question 2 Max Marks	**10**		

Course Marks Tabulation				
Component-1 (C) Assignment	**First Examiner**	**Remarks**	**Moderator**	**Remarks**
Question 1				
Question 2				
Marks (out of 20)				

Signature of First Examiner **Signature of Moderator**

Question No. 1

Solution to Question No. 1:

1.1 Introduction to the functionality of the circuit

A calculator is a device that performs arithmetic operations on numbers. The simplest calculators can do only addition, subtraction, multiplication, and division.
We are taking two 2-bit binary number as input. And getting 4-bit binary numbers output.

From fig 1:

- $\boldsymbol{A = A1.A0}$
- $\boldsymbol{B = B1.B0}$

And 2-bit selector lines $\boldsymbol{S1.S0}$ choose which operations to perform.

Selector lines		**operation**
S1	**S0**	
0	**0**	addition
0	**1**	subtraction
1	**0**	multiplication
1	**1**	division

Fig 1 shows block diagram of adder, subtractor, multiplier and divider circuit.

Figure Number	**Circuit Name**	**Inputs**	**Outputs**
2	Adder	**A1, A0 --- B1, B0**	**S3, S2, S1, S0**
3	Subtractor	**A1, A0 --- B1, B0**	**D3, D2, D1, D0**
4	Multiplier	**A1, A0 --- B1, B0**	**P3, P2, P1, P0**
5	Divider	**A1, A0 --- B1, B0**	**Q1, Q0 --- R1, R0**

1.2 Circuit design using Logisim

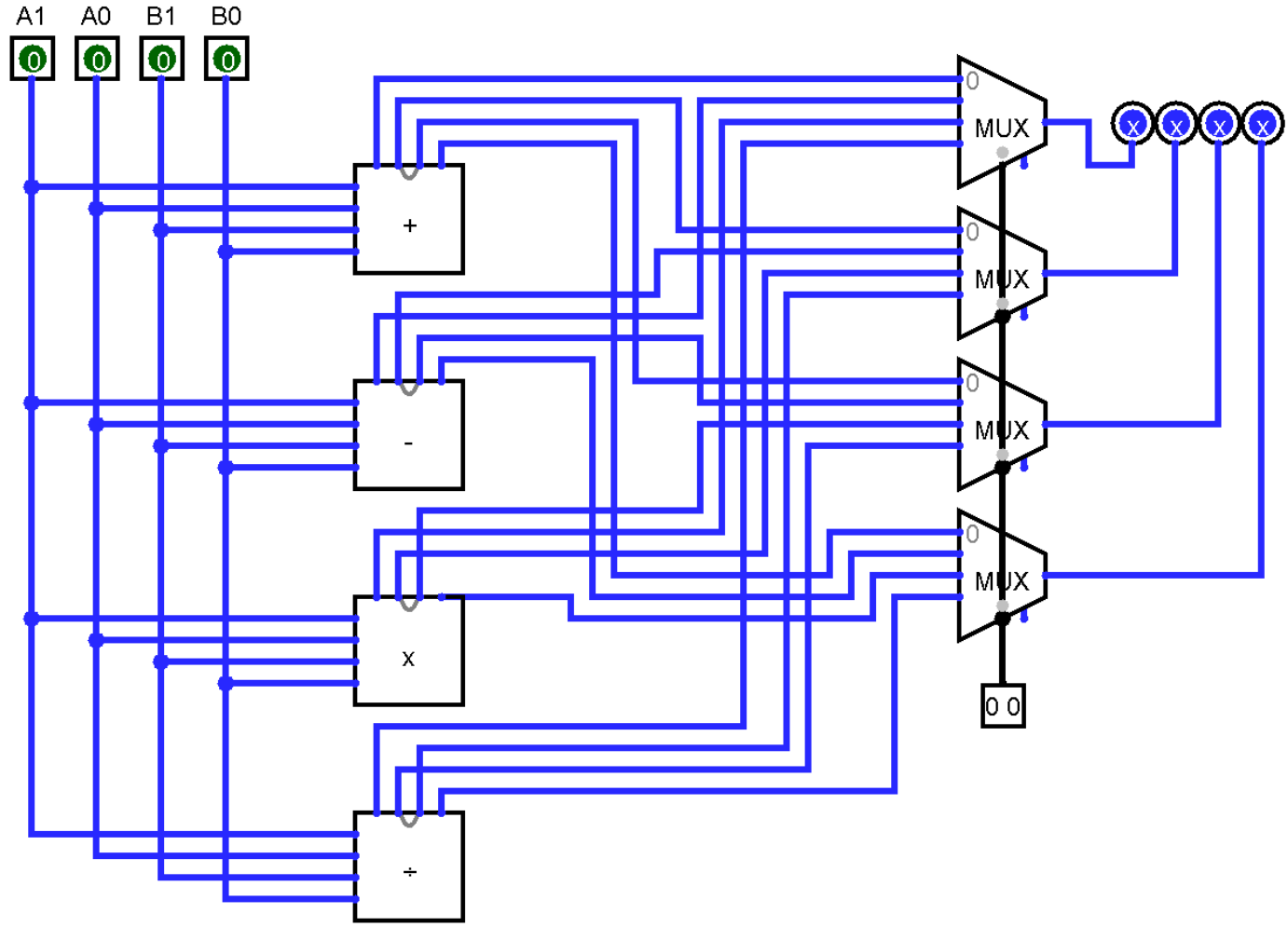

Figure 1 combined circuit for all the operations

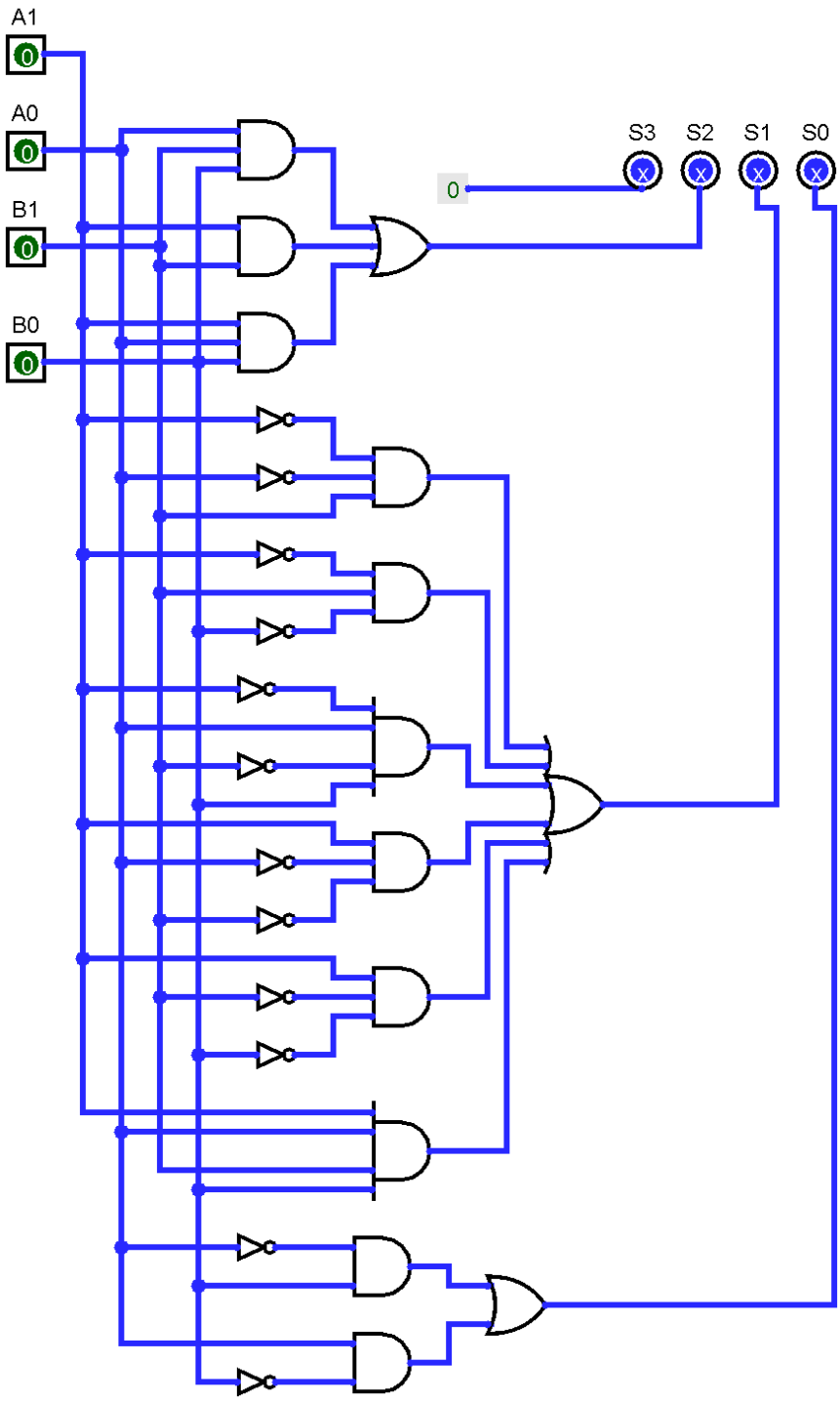

Figure 2 adder circuit using basic gates

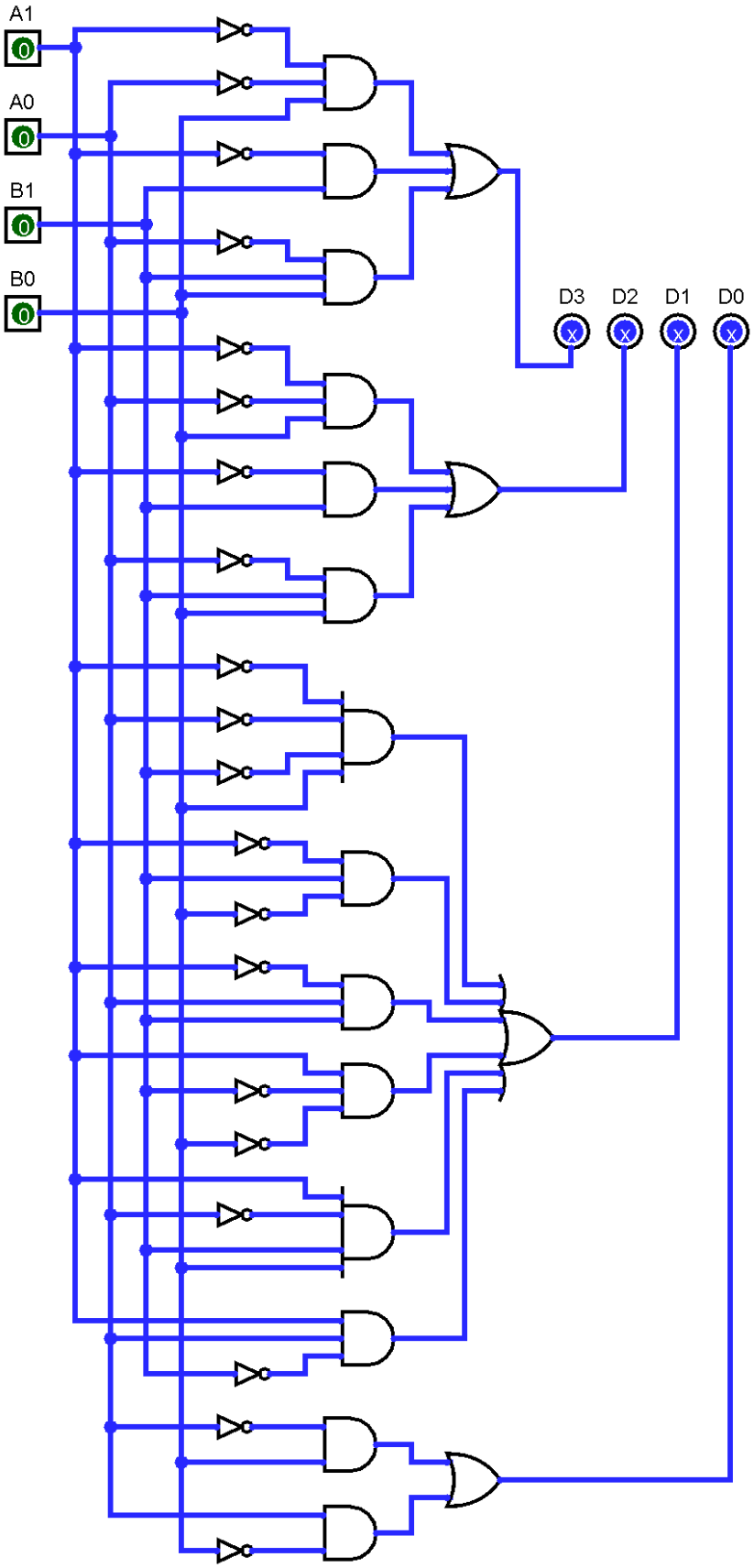

Figure 3 subtraactor using basic gates

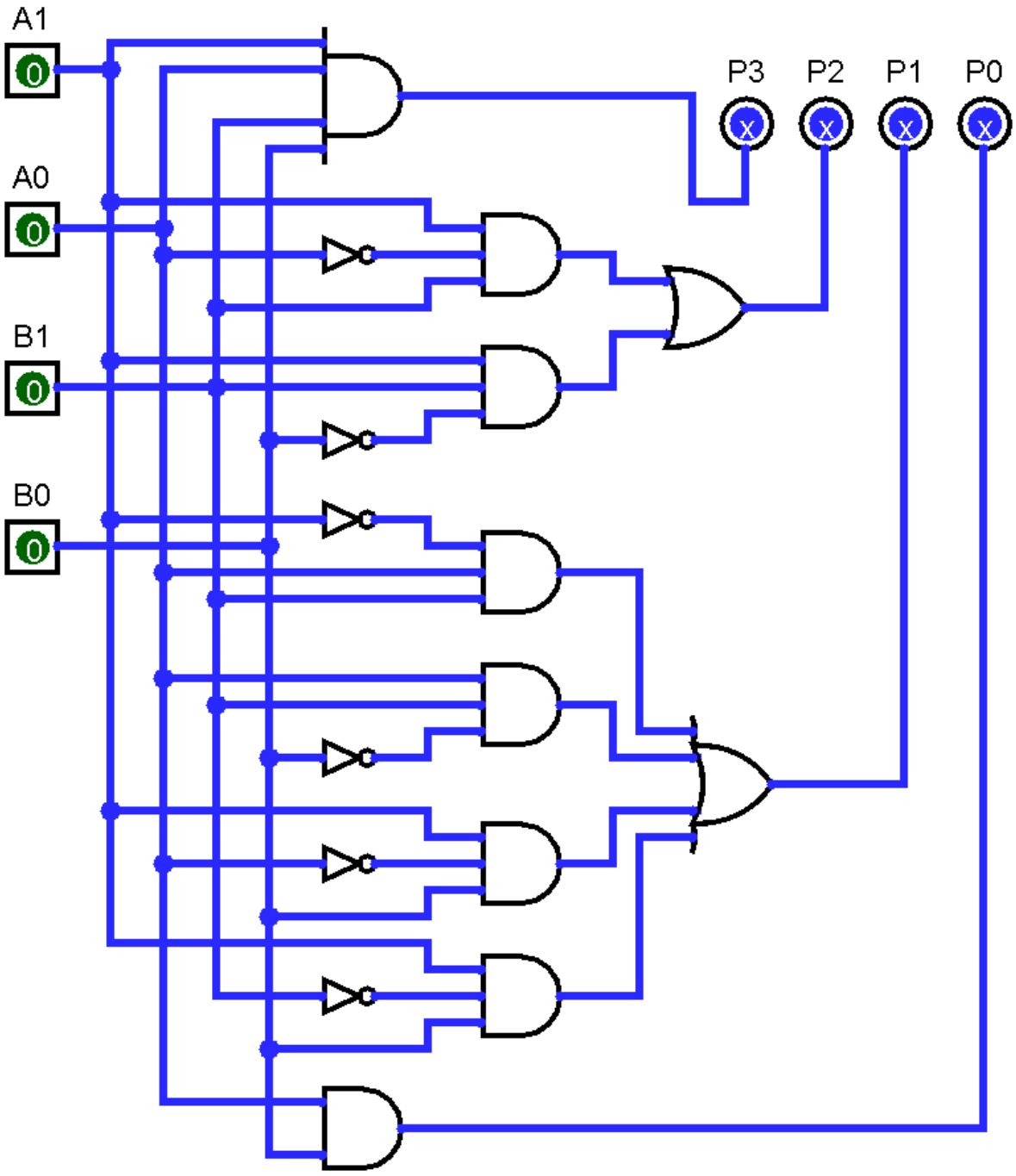

Figure 4 multiplier using basic gates

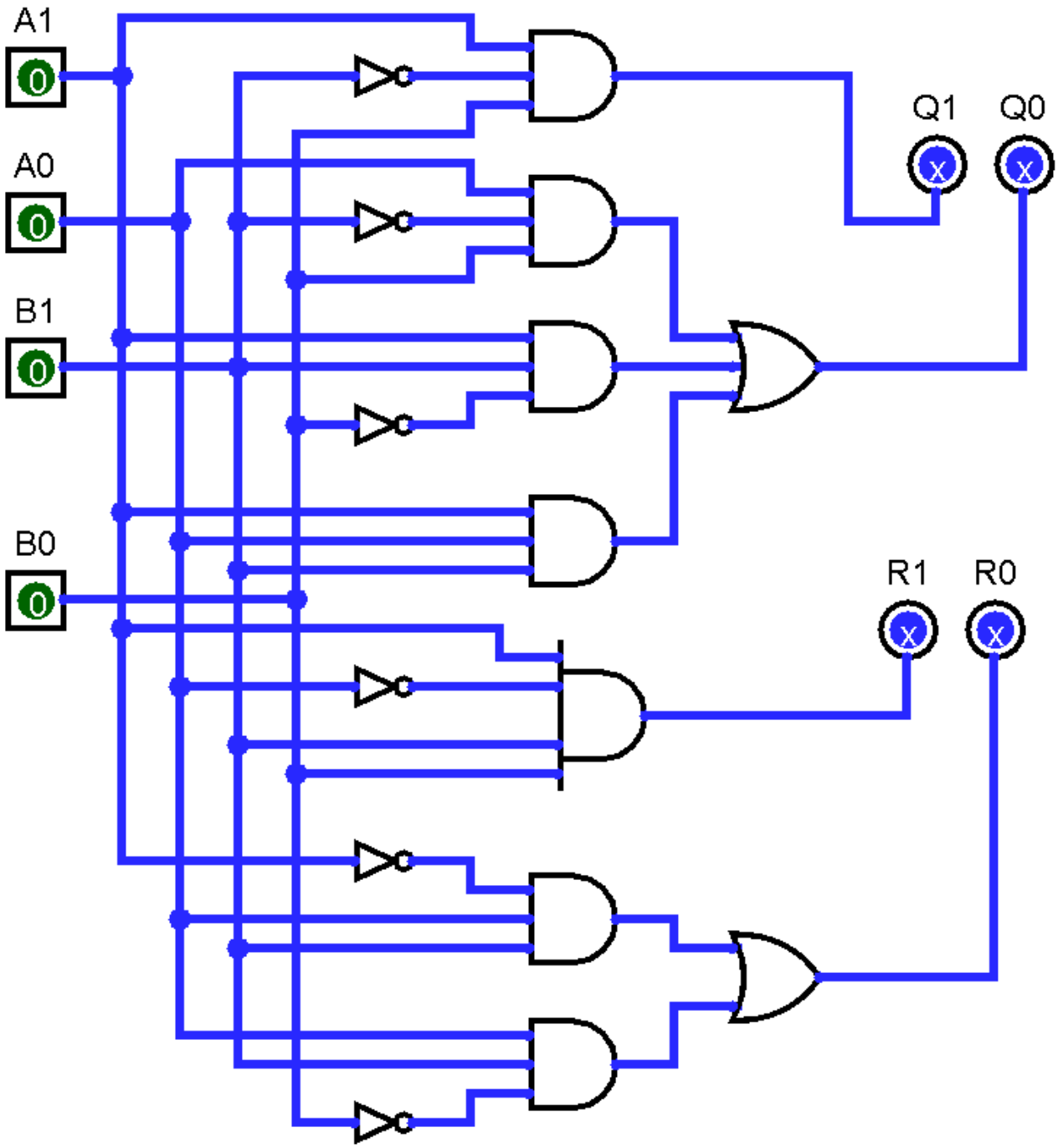

Figure 5 divider using basic gates

1.3 Simulation of the designed circuit

Adding $\boldsymbol{A = 01}$ and $\boldsymbol{B = 11}$, $\boldsymbol{S = 00}$ which is giving output $\boldsymbol{S = 0100.}$

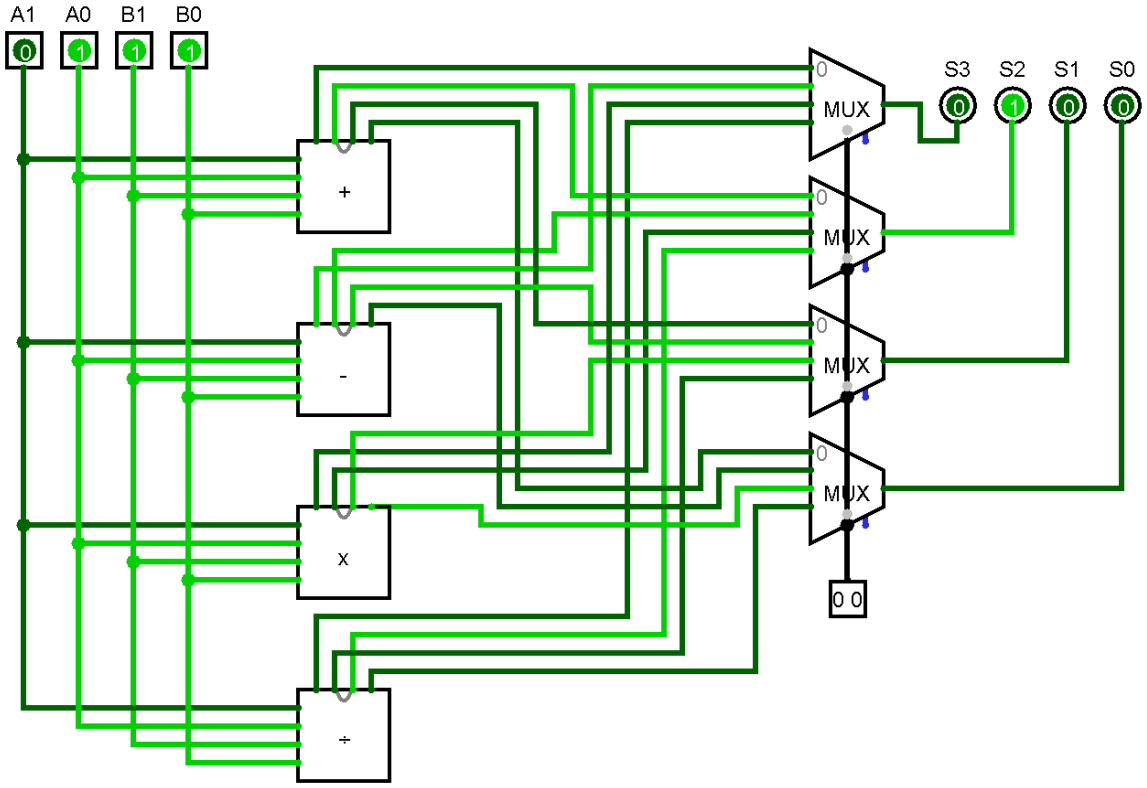

Figure 6 simulation of adder (adding 1 and 3)

Subtracting $\boldsymbol{A = 01}$ and $\boldsymbol{B = 11}$, $\boldsymbol{S = 01}$ which is giving output $\boldsymbol{S = 1110.}$

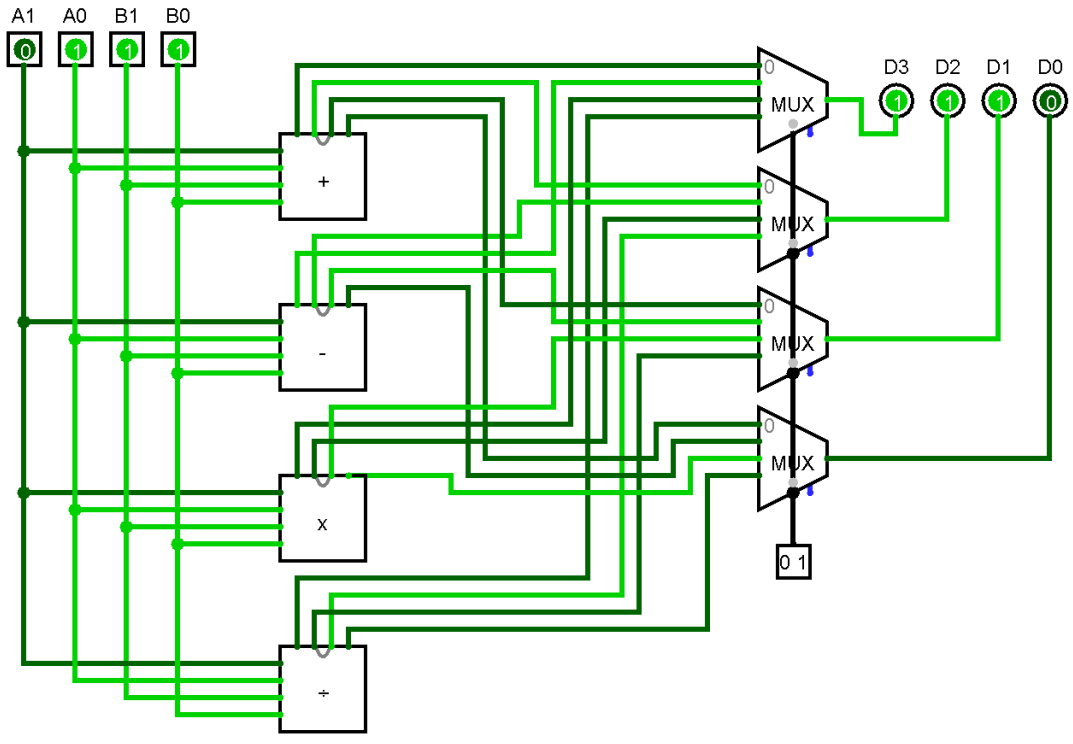

Figure 7 simulation of subtractor (subtracting 1 and 3)

Multiplying $\boldsymbol{A = 01}$ and $\boldsymbol{B = 11}$, $\boldsymbol{S = 10}$ which is giving output $\boldsymbol{S = 0011.}$

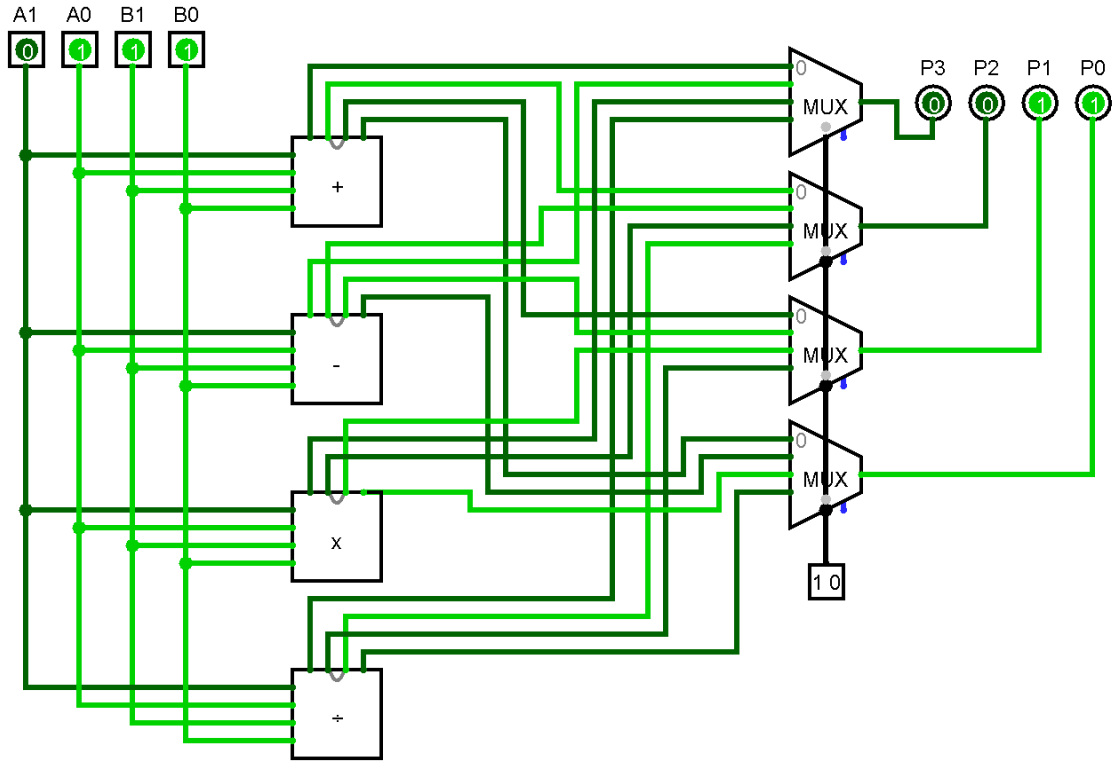

Figure 8 simulation of multiplying (multiplying 1 and 3)

Dividing $\boldsymbol{A = 01}$ and $\boldsymbol{B = 11}$, $\boldsymbol{S = 11}$ which is giving output $\boldsymbol{Q = 00, R = 01.}$

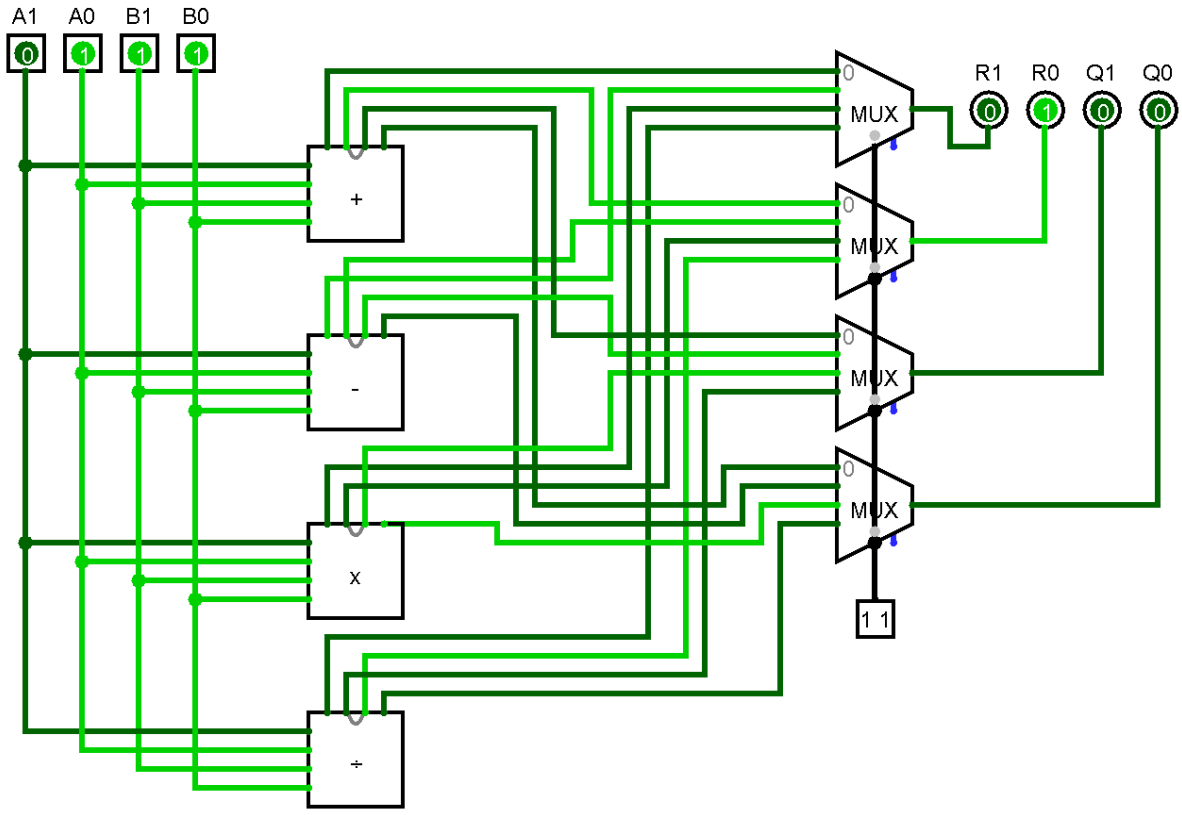

Figure 9 simulation of divider (dividing 1 and 3)

1.4 Changes in the circuit if you had to implement it using only NAND gates

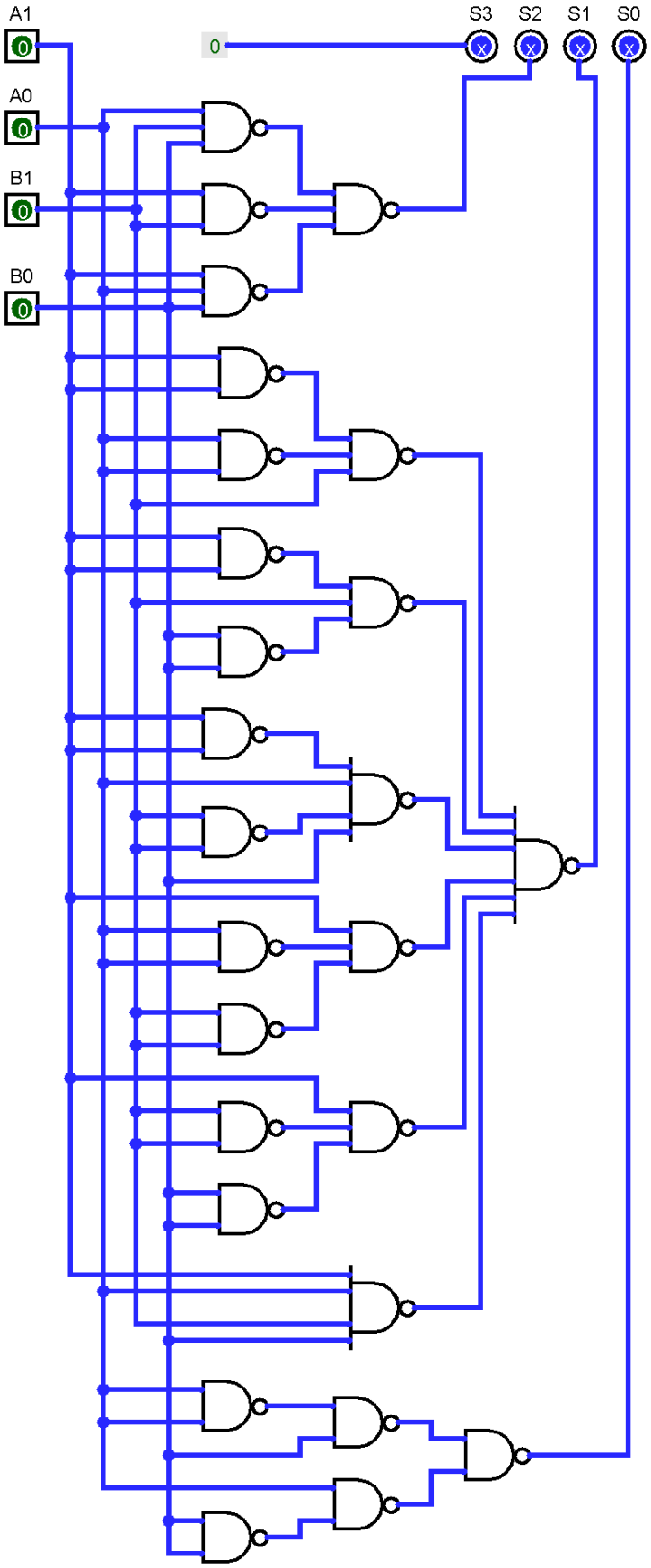

Figure 10 adder circuit using NAND gates

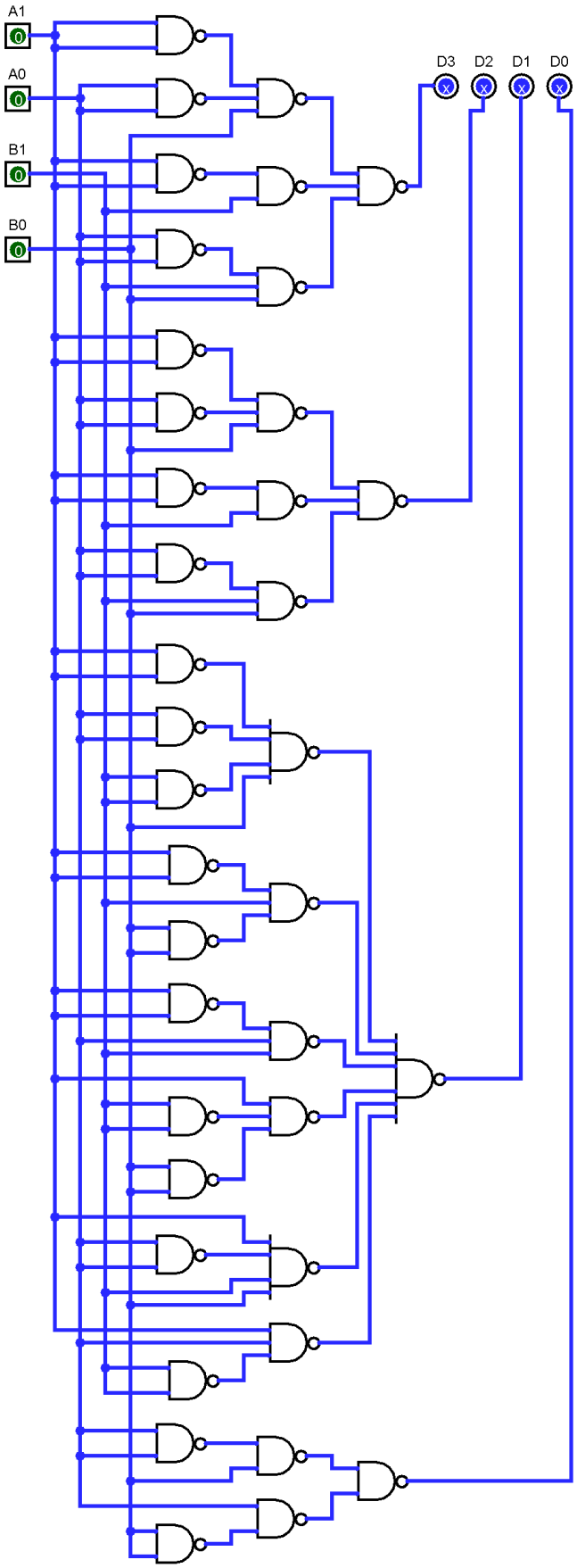

Figure 11 subtractor circuit using NAND gates

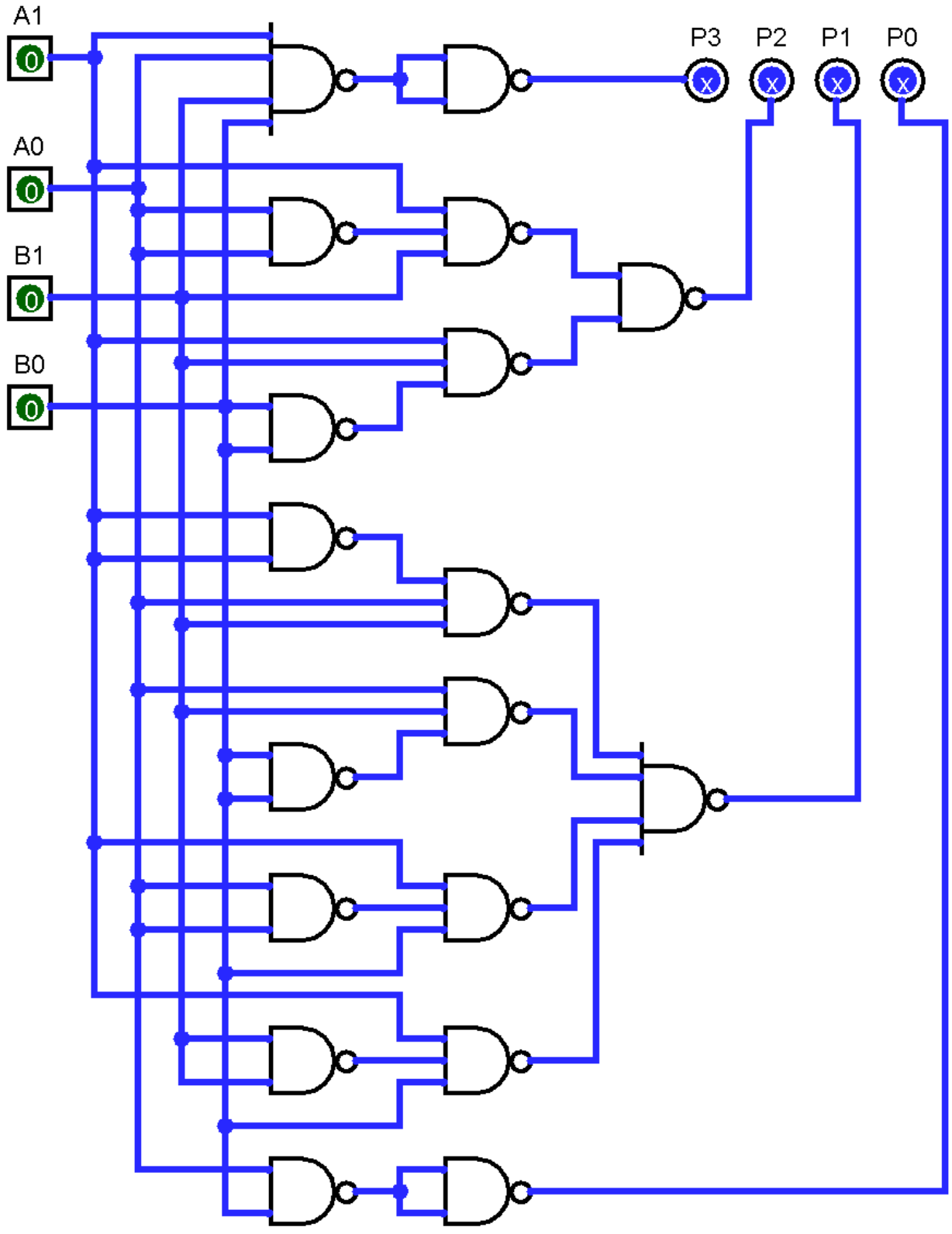

Figure 12 multiplier circuit using NAND gates

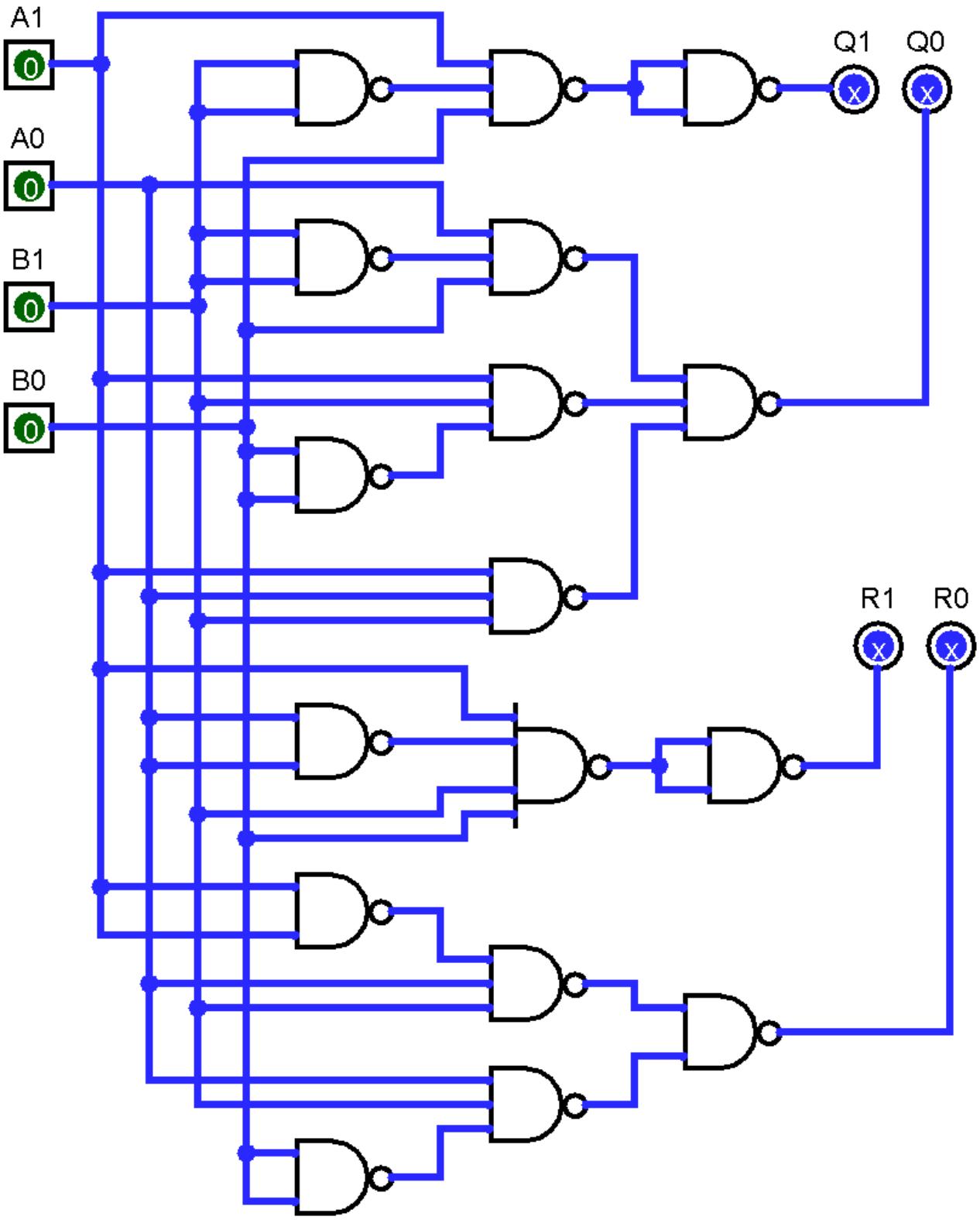

Figure 13 divider circuit using NAND gates

Question No. 2

Solution to Question No. 2:

2.1 Introduction to the functionality of the circuit

This is a combinational circuit that generates 2's complement of a 4-bit input binary number. It takes $A\ (A3\ A2\ A1\ A0)$ as input and gives $Y\ (\ Y3\ Y2\ Y1\ Y0\ )$ which is 2's complement of A as output.

Equations:

- $\boldsymbol{Y3 = \sim A0\ \sim A1\ \sim A2\ A3\ +\ A2\ \sim A3\ +\ A1\ \sim A3\ +\ A0\ \sim A3}$
- $\boldsymbol{Y2 = \sim A0\ \sim A1\ A2\ +\ A1\ \sim A2\ +\ A0\ \sim A2}$
- $\boldsymbol{Y1 = \sim A0\ A1\ +\ A0\ \sim A1}$
- $\boldsymbol{Y0 = A0}$

2.2 Circuit design using Logisim

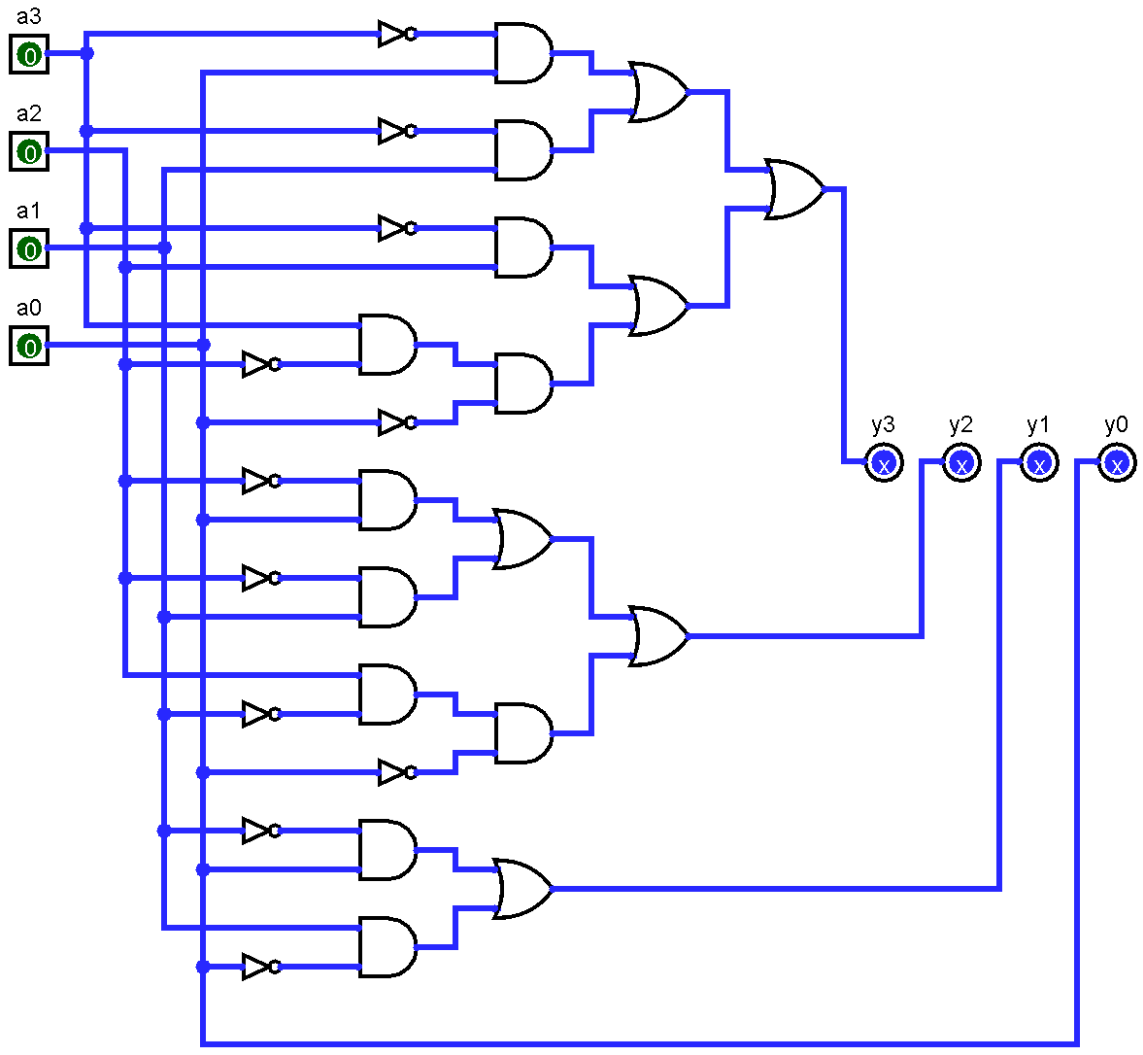

Figure 14 2's complement calculator using basic gates

2.3 Simulation of the designed circuit

In Fig 15,
We can see for $\boldsymbol{A} = \mathbf{0110}$ we are getting $\boldsymbol{Y} = \mathbf{1010}$, which is the 2's complement of A.

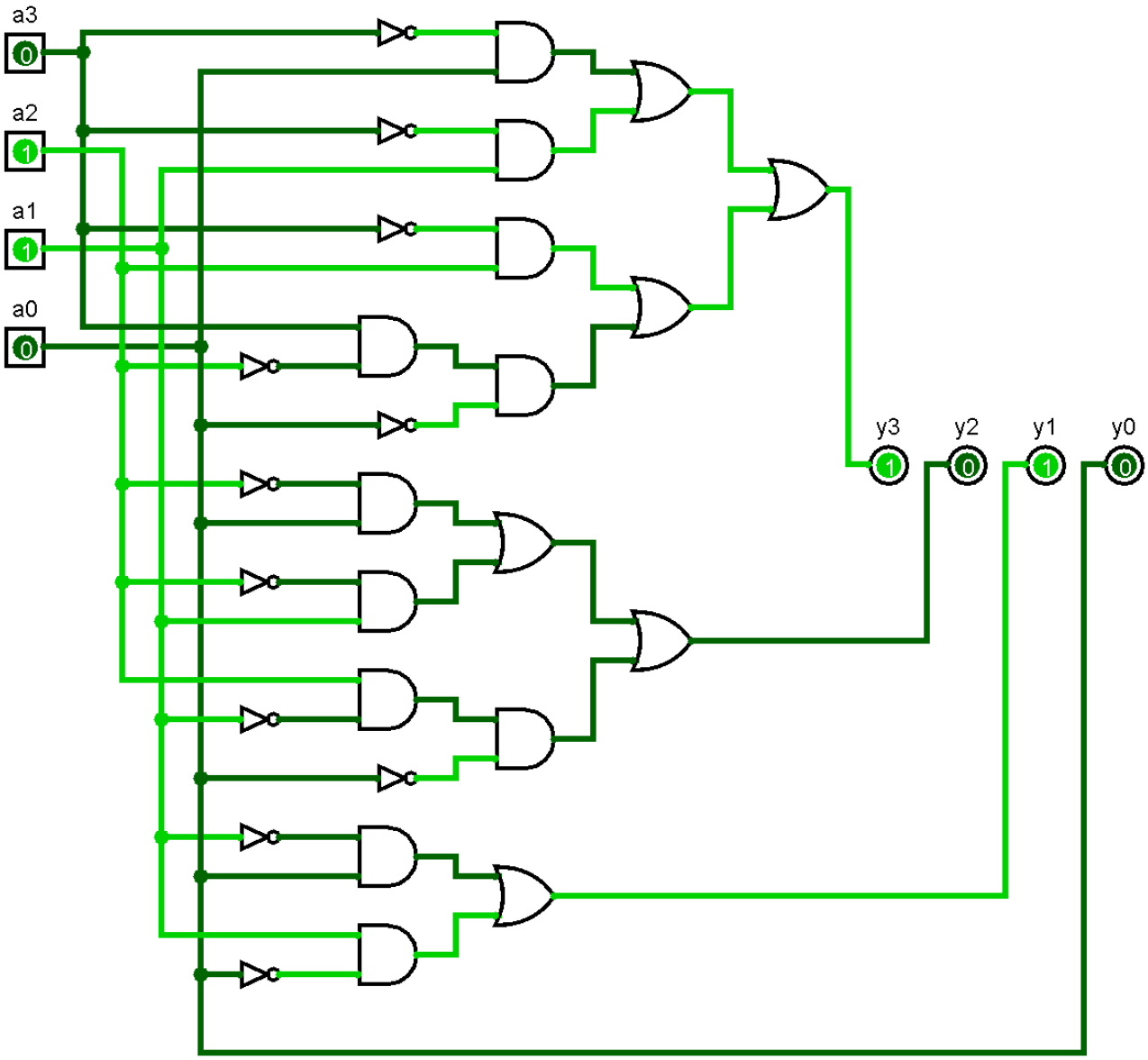

Figure 15 simulating 2's complement calculator

2.4 Changes in the circuit if you had to implement it using only XOR gates

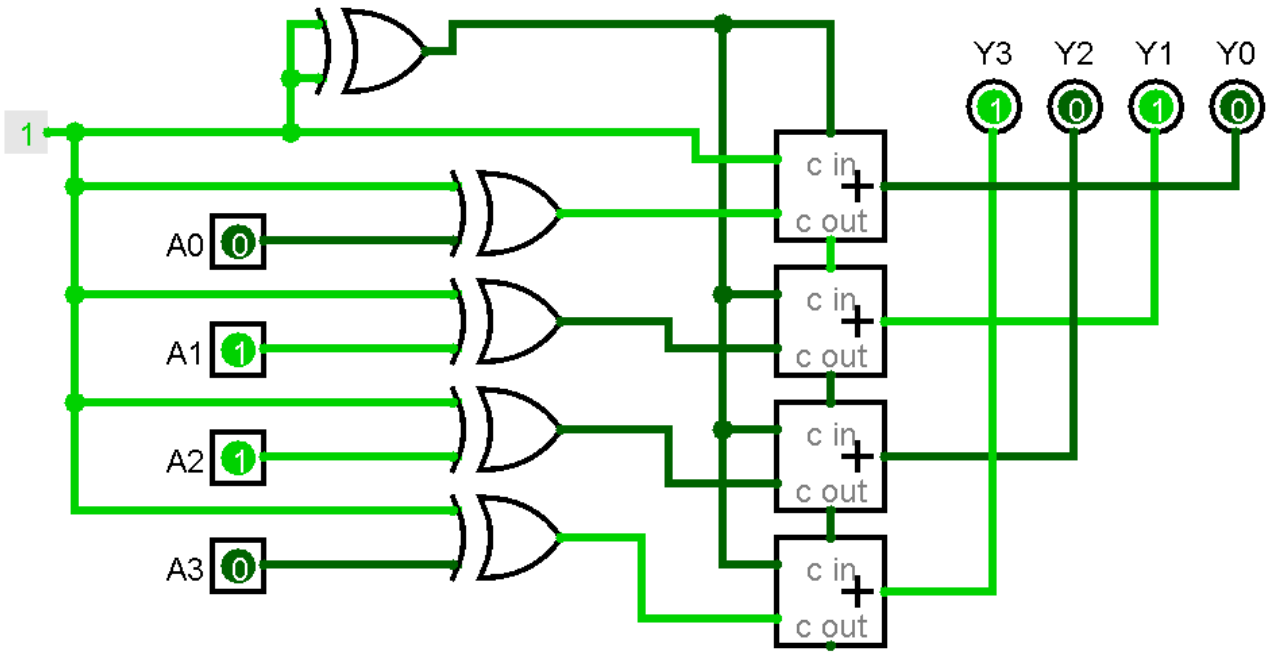

Figure 16 2's complement calculator using XOR gate only.

A0	A1	A2	A3	Y3	Y2	Y1	Y0
0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0
0	0	1	0	1	1	0	0
0	0	1	1	0	1	0	0
0	1	0	0	1	1	1	0
0	1	0	1	0	1	1	0
0	1	1	0	1	0	1	0
0	1	1	1	0	0	1	0
1	0	0	0	1	1	1	1
1	0	0	1	0	1	1	1
1	0	1	0	1	0	1	1
1	0	1	1	0	0	1	1
1	1	0	0	1	1	0	1
1	1	0	1	0	1	0	1
1	1	1	0	1	0	0	1
1	1	1	1	0	0	0	1

Figure 17 truth table of 2's complement

Bibliography

- Class notes / ppts
- https://circuit-diagramz.com/circuit-diagram-calculator-using-logic-gates/
- https://rosettacode.org/wiki/Four_bit_adder
- For Logisim files,
- Refer to: https://github.com/subhendu17620/RUAS-sem-03/tree/master/LD%20lab/assignment/logisim%20files