Field of Computer Science
School of Computing, Faculty of Engineering
UNIVERSITI TEKNOLOGI MALAYSIA

COURSE : SECR1013 DIGITAL LOGIC

SESSION/SEM : 2021/2022 - 1

LAB 2 : COMBINATIONAL LOGIC CIRCUIT DESIGN SIMULATION USING DEEDS SIMULATOR

NAME 1: HARCHANA A/P ARULAPPAN (A21EC0028)
NAME 2: NASRUL AMIN BIN AB HADI (A21EC0099)

SECTION & LECTURER: 01/ DR MOHD FOAD

DATE: 31 DECEMBER 2021

A. Objective

i) To expose student with producing digital logic circuit, generating truth table with Deeds Simulator.

ii) To expose student with conversion between basic gates circuits and universal gates circuits.

iii) To expose student with a complete cycle process of a combinatorial circuit design and simulate with Deeds Simulator.

B. Material

Deeds Software for Windows.

Lab Activities

Part A

Simulating and converting logic circuit and construct truth table with Deeds.

1. Draw circuit in Figure 1 in Deeds.

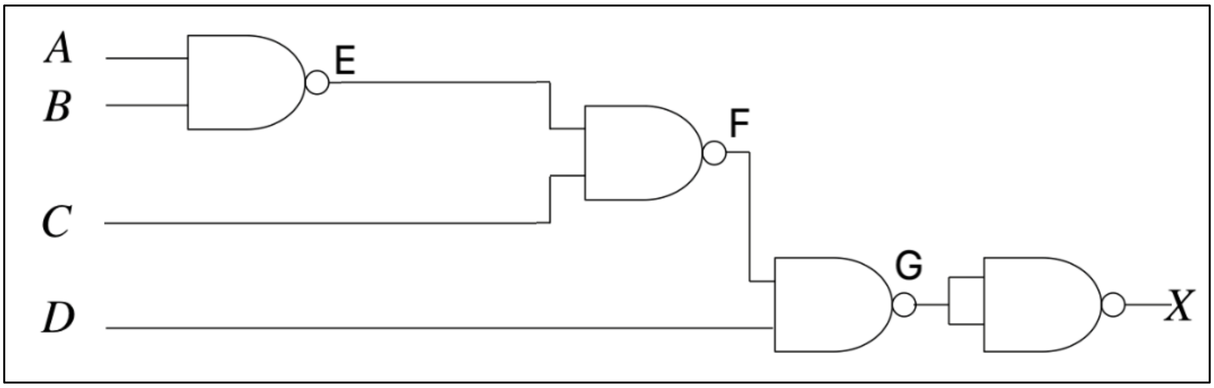
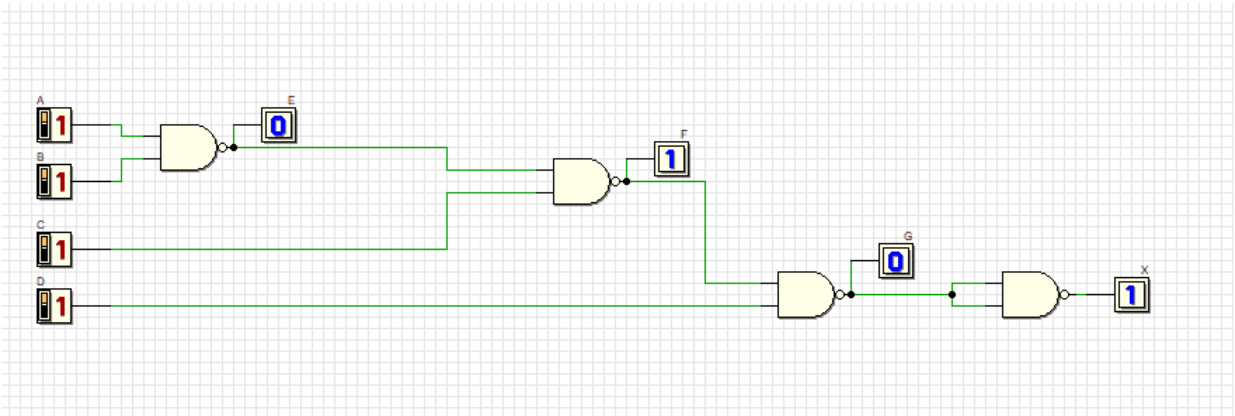


Figure 1: NAND Universal gates circuit



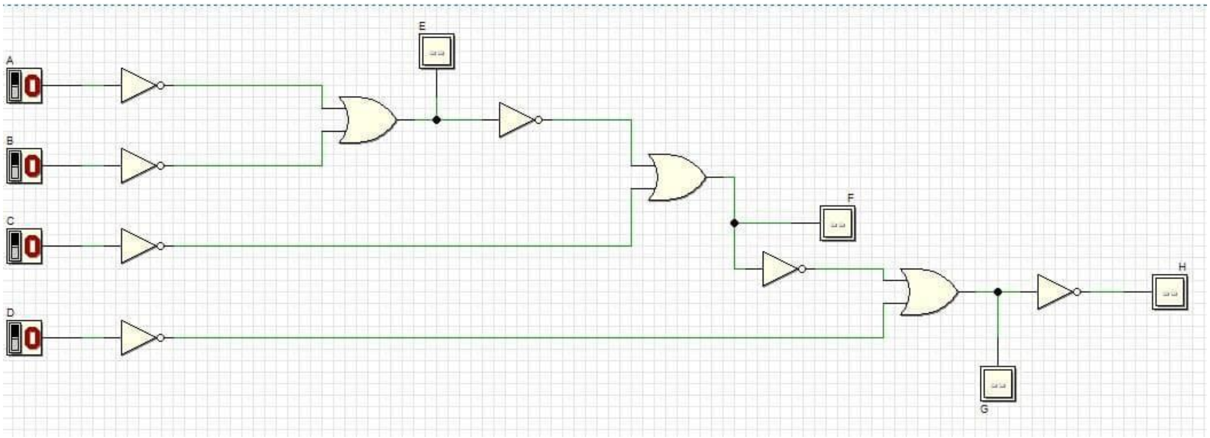
2. Simulate the circuit and built the truth table using the following headers.

Truth Table 1

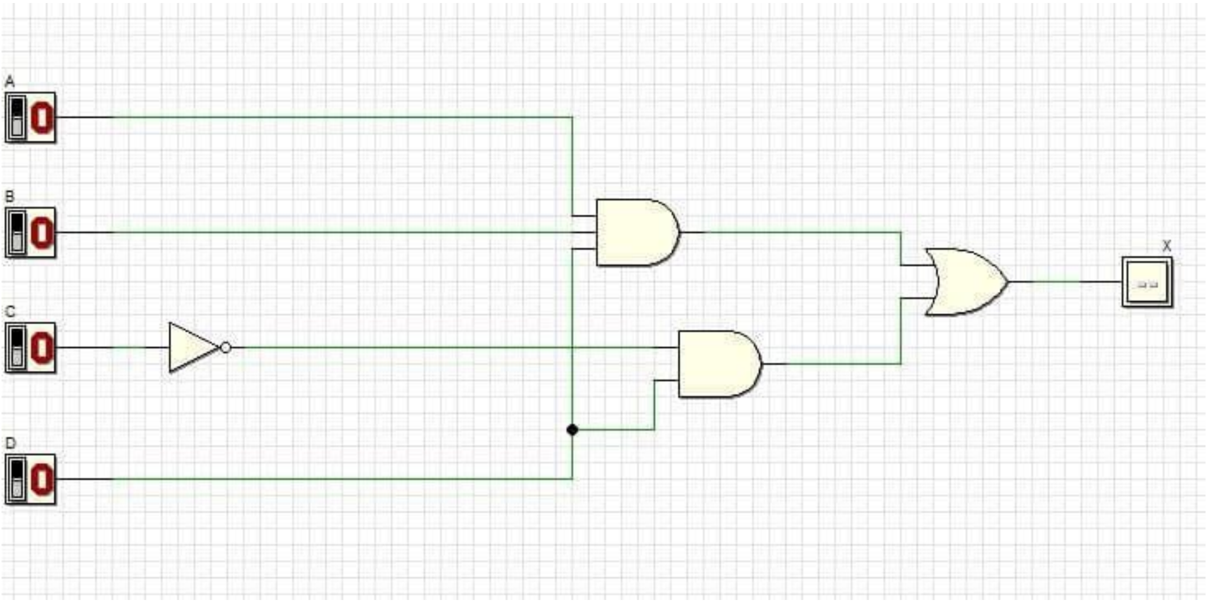
INPUT				OUTPUT			
A	**B**	**C**	**D**	**E**	**F**	**G**	**X**
0	0	0	0	1	1	1	0
0	0	0	1	1	1	0	1
0	0	1	0	1	0	1	0
0	0	1	1	1	0	1	0
0	1	0	0	1	1	1	0
0	1	0	1	1	1	0	1
0	1	1	0	1	0	1	0
0	1	1	1	1	0	1	0
1	0	0	0	1	1	1	0

1	0	0	1	1	1	0	1
1	0	1	0	1	0	1	0
1	0	1	1	1	0	1	0
1	1	0	0	0	1	1	0
1	1	0	1	0	1	0	1
1	1	1	0	0	1	1	0
1	1	1	1	0	1	0	1

3. Choose the level/label you want change and change the gates to dual symbol. Hint: Draw NAND dual symbol using OR gate and 2 NOT gates and change NOT drawn with NAND to basic NOT. Paste the circuit (Figure 2) here.



4. Simplified the circuit in Question 3. Paste the circuit (Figure 3) here.



5. Confirm circuit in Figure 3 is equivalent to circuit in Figure 1 by simulating the circuit and building Truth Table 2.

Truth Table 2

INPUT				OUTPUT
A	**B**	**C**	**D**	**X**
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
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

6. Derive the output expression from Figure 1 and show that it is equivalent to the expressions from Figure 3.

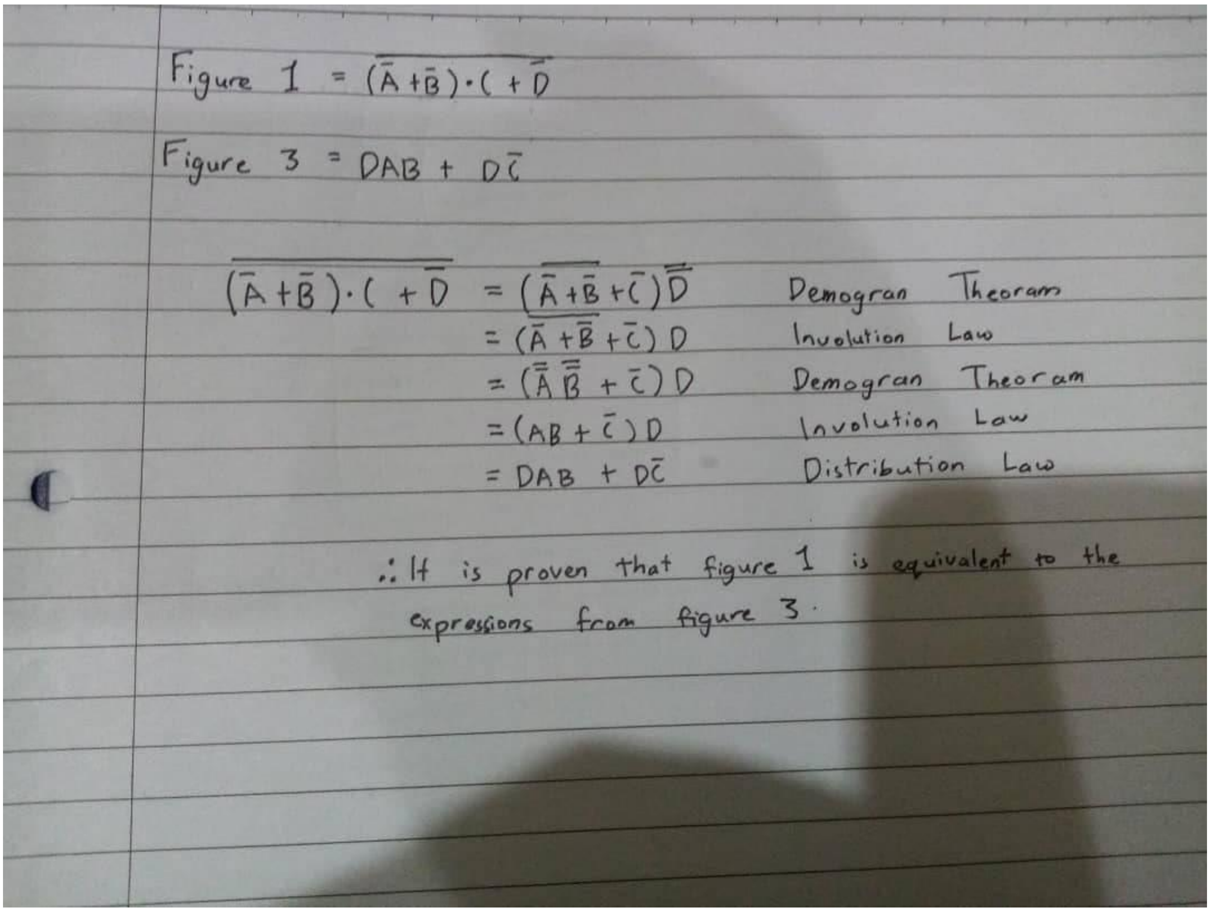


Figure 1 = $\overline{(\bar{A}+\bar{B})\cdot(+ \bar{D}}$

Figure 3 = $DAB + D\bar{C}$

$\overline{(\bar{A}+\bar{B})\cdot(+ \bar{D}}$	$= (\overline{\bar{A}+\bar{B}}+\bar{C})\bar{\bar{D}}$	Demogran Theoram
	$= (\overline{\bar{A}+\bar{B}}+\bar{C})D$	Involution Law
	$= (\bar{\bar{A}}\bar{\bar{B}}+\bar{C})D$	Demogran Theoram
	$= (AB+\bar{C})D$	Involution Law
	$= DAB + D\bar{C}$	Distribution Law

∴ It is proven that figure 1 is equivalent to the expressions from figure 3.

Part B

Combinational circuit design process and simulate with Deeds Simulator.

Design Process

i) Determine Parameter Input/Output and their relations.
ii) Variable definition (not needed here)
iii) Construct Truth Table.
iv) Using K-Map, get the SOP optimized form of all Boolean equation outputs.
iv) Draw the circuit and use duality symbol.
v) Simulate the design using Deeds Simulator. Check the results according to Truth Table.

Problem Situation

A new digital fault diagnoses circuit is requested to be designed for analysing four bit 2's complement input binary number from sensors A, B, C, and D. Sensor A represents input MSB and sensor D represents input LSB. As shown in the following Figure 4, bit pattern analysis from input sensors A, B, C, and D will trigger four different output errors (active HIGH) of type E1, E2, E3, and E4.

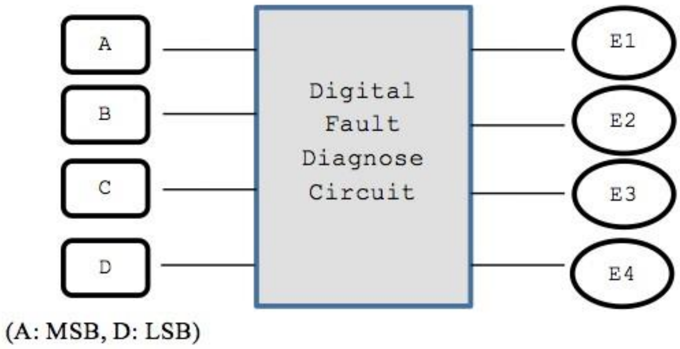


Figure 4

The following rules are used to activate the error's signal type:

RULE 1: E1 is activated if the input number is positive even and the majority of the bits is '0'.
RULE 2: E2 is activated if the input number is positive odd, and the majority of the bits is '0'.
RULE 3: E3 is activated if the input number is negative even and the majority of the bits is '1'.
RULE 4: E4 is activated if the input number is negative odd, and the majority of the bits is '1'.
RULE 5: The output of error signal is invalid if the input has equal bit '0' and bit '1'

NOTE: Positive odd is positive numbers that are odd
Negative even is negative numbers that are even.
Zero is considered positive even number.

Experimental Steps

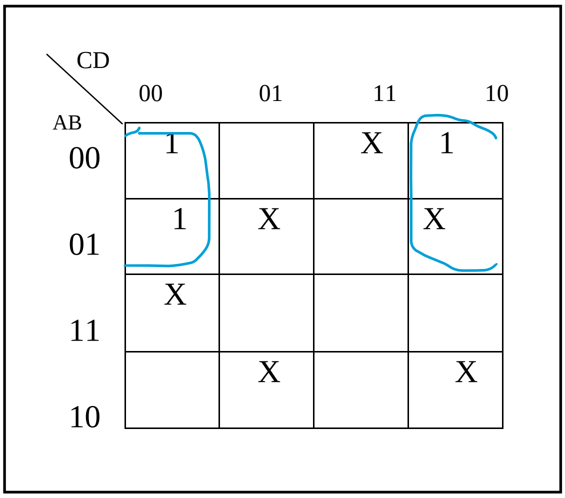
1. Create Truth Table 3 for Digital Fault Diagnose Circuit. Use variables A, B, C and D as inputs; E1, E2, E3 and E4 as outputs.

Truth Table 3

INPUT				OUTPUT			
A	**B**	**C**	**D**	**E1**	**E2**	**E3**	**E4**
0	**0**	**0**	**0**	1	0	0	0
0	**0**	**0**	**1**	0	1	0	0
0	**0**	**1**	**0**	1	0	0	0
0	**0**	**1**	**1**	x	x	x	x
0	**1**	**0**	**0**	1	0	0	0
0	**1**	**0**	**1**	x	x	x	x
0	**1**	**1**	**0**	x	x	x	x
0	**1**	**1**	**1**	0	0	0	0
1	**0**	**0**	**0**	0	0	0	0
1	**0**	**0**	**1**	x	x	x	x
1	**0**	**1**	**0**	x	x	x	x
1	**0**	**1**	**1**	0	0	0	1
1	**1**	**0**	**0**	x	x	x	x
1	**1**	**0**	**1**	0	0	0	1
1	**1**	**1**	**0**	0	0	1	0
1	**1**	**1**	**1**	0	0	0	1

2. Using K-MAP, get minimized SOP Boolean expressions for E1, E2, E3 and E4 circuits. Post your K-MAP here.

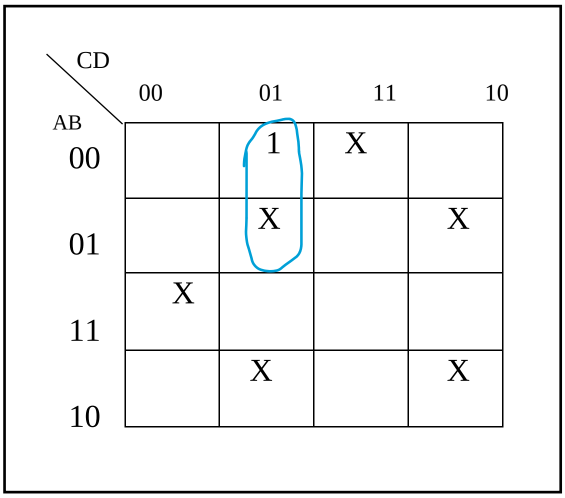
E1



AB \ CD	00	01	11	10
00	1		X	1
01	1	X		X
11	X			
10		X		X

E1 = A'D'

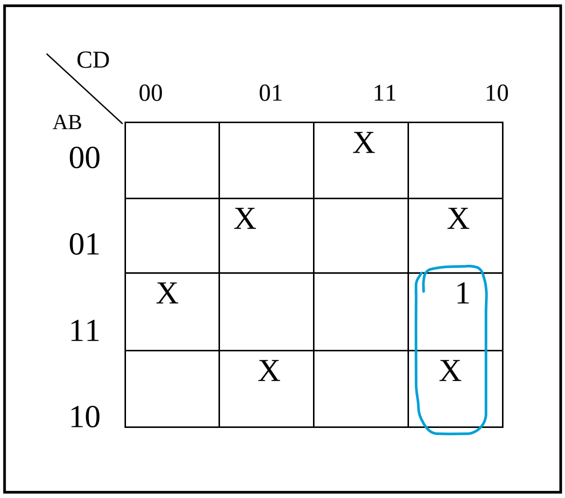
E2



CD AB	00	01	11	10
00		1	X	
01		X		X
11	X			
10		X		X

E2 = A’C’D

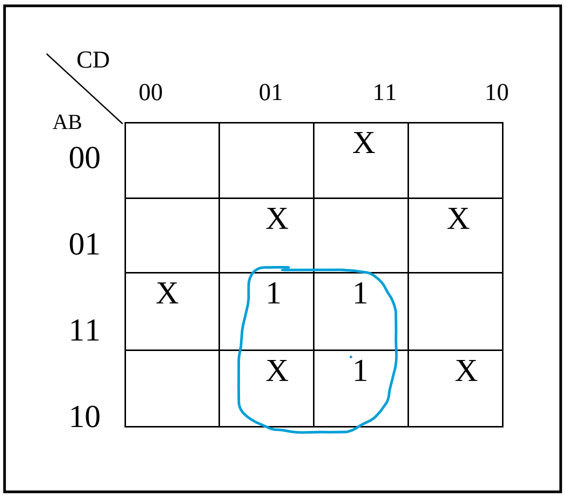
E3



CD AB	00	01	11	10
00			X	
01		X		X
11	X			1
10		X		X

E3 = ACD’

E4

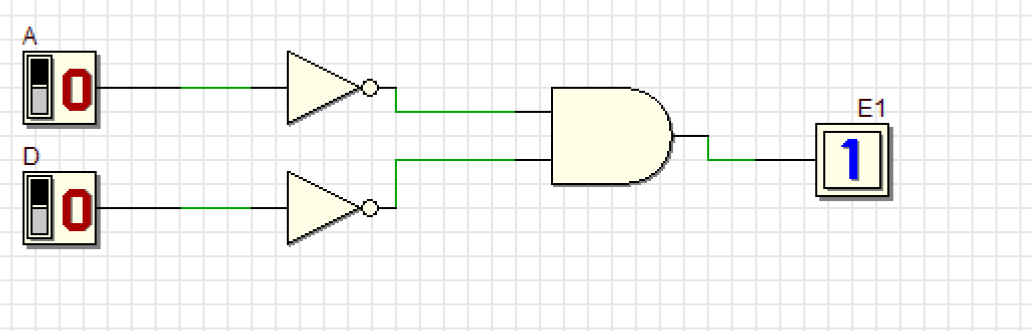


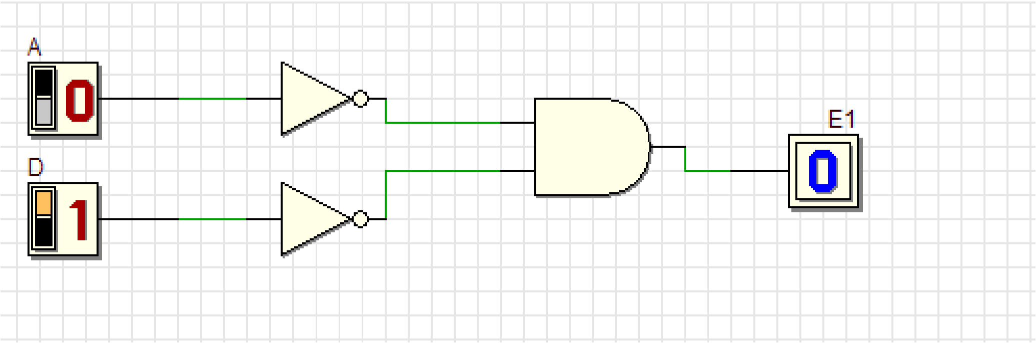
CD AB	00	01	11	10
00			X	
01		X		X
11	X	1	1	
10		X	1	X

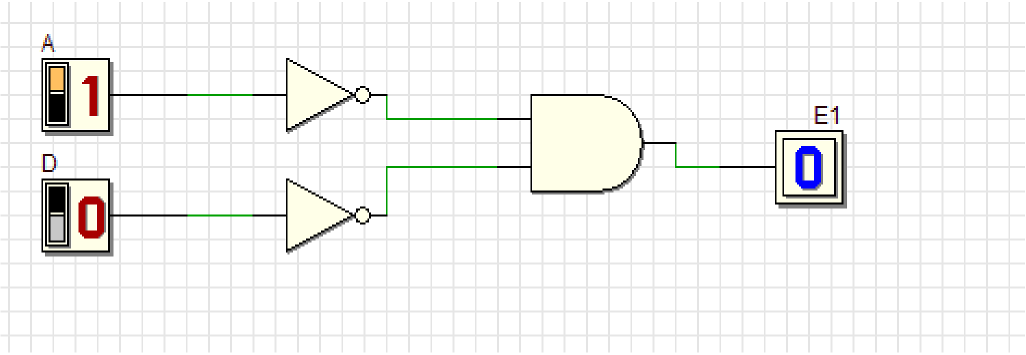
E4 = AD

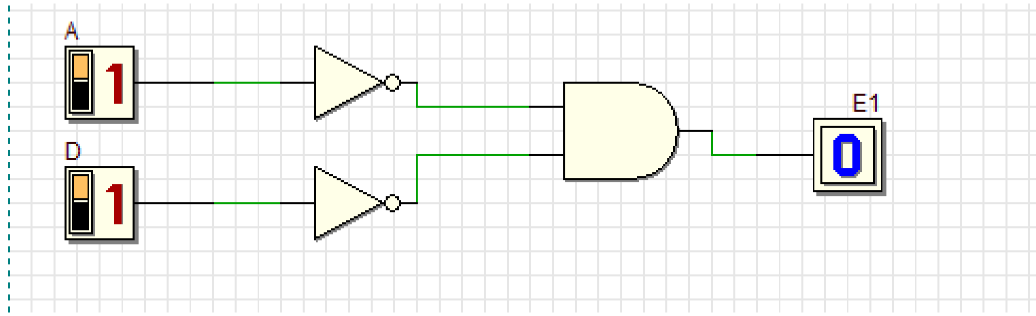
3. From the Boolean expression in the step (2), draw your final E1, E2, E3 and E4 circuits. Use Deeds Simulator and paste your circuit here.

E1

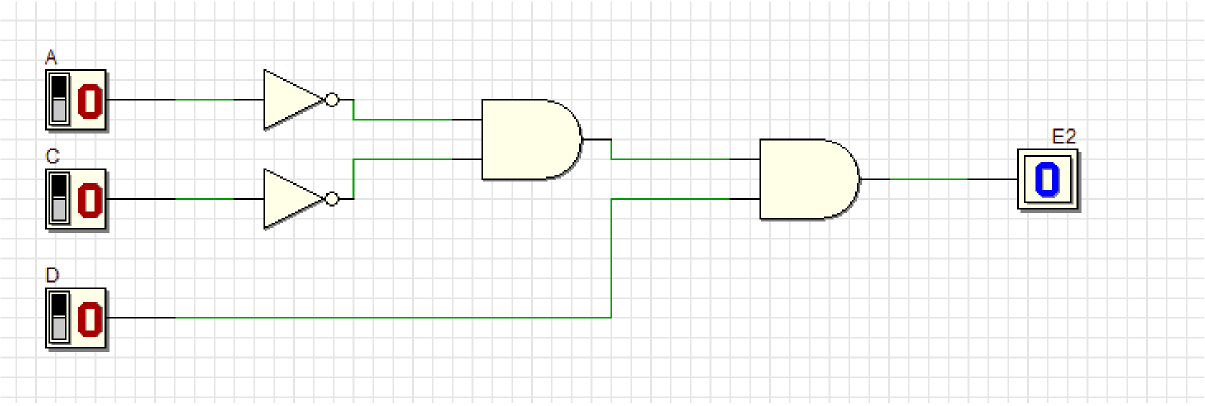


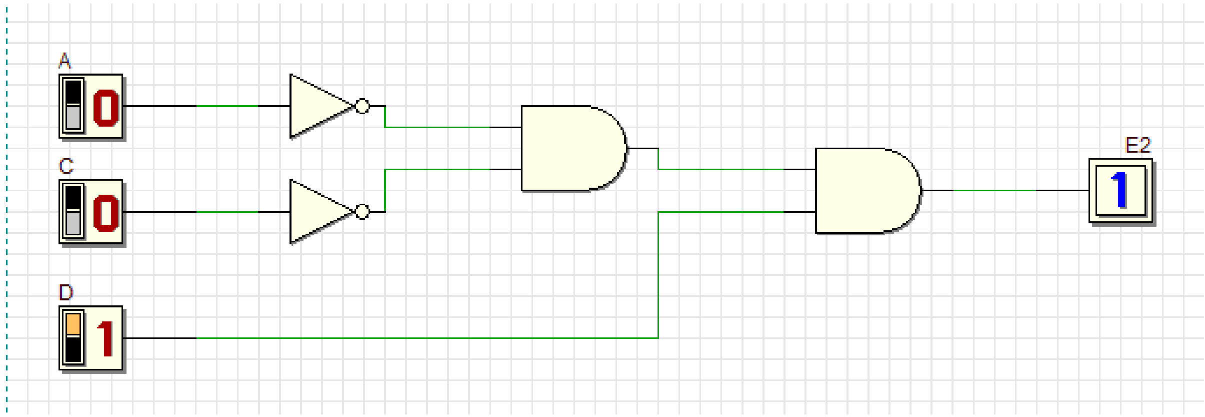


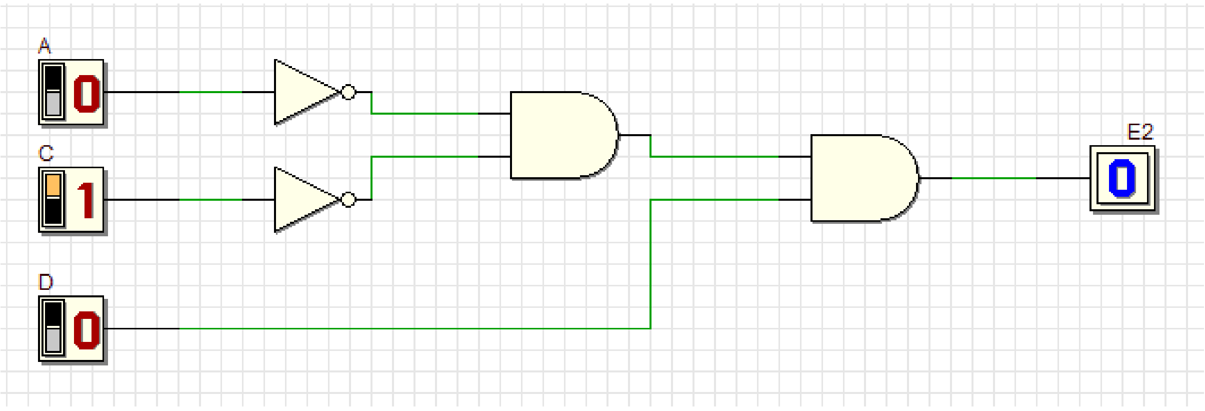


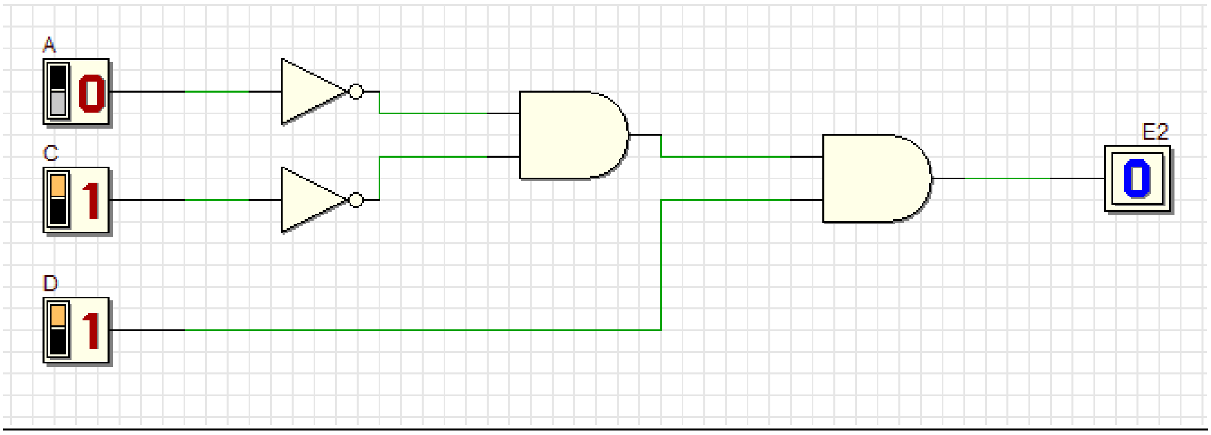


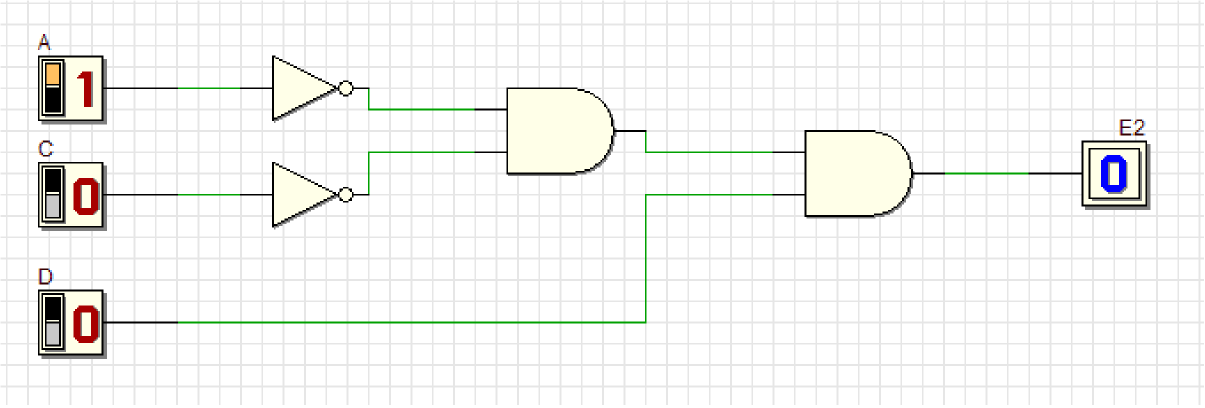
E2

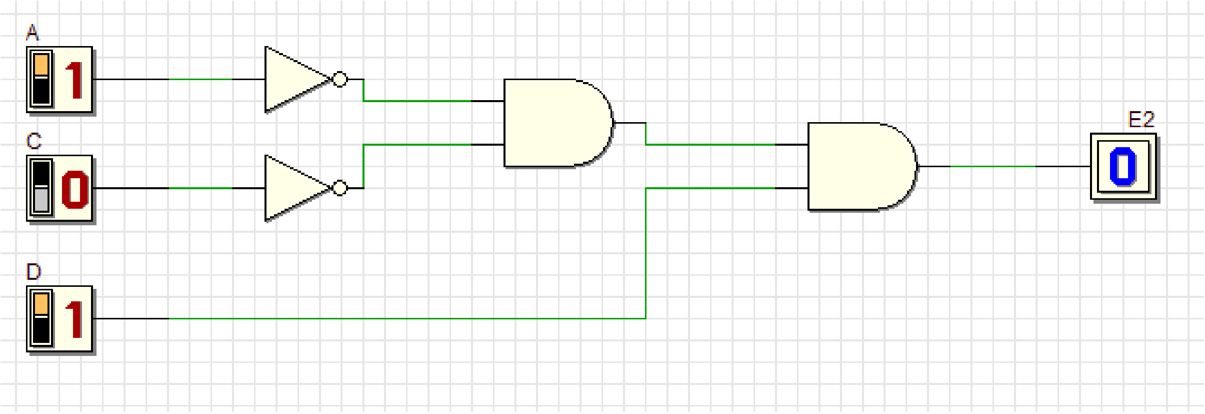

A
0
C
0
D
0
E2
0

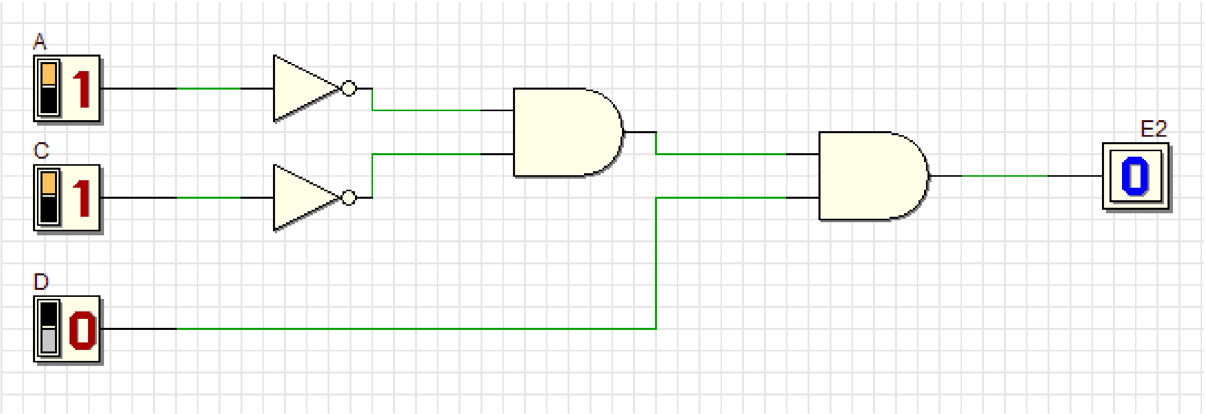

A
0
C
0
D
1
E2
1

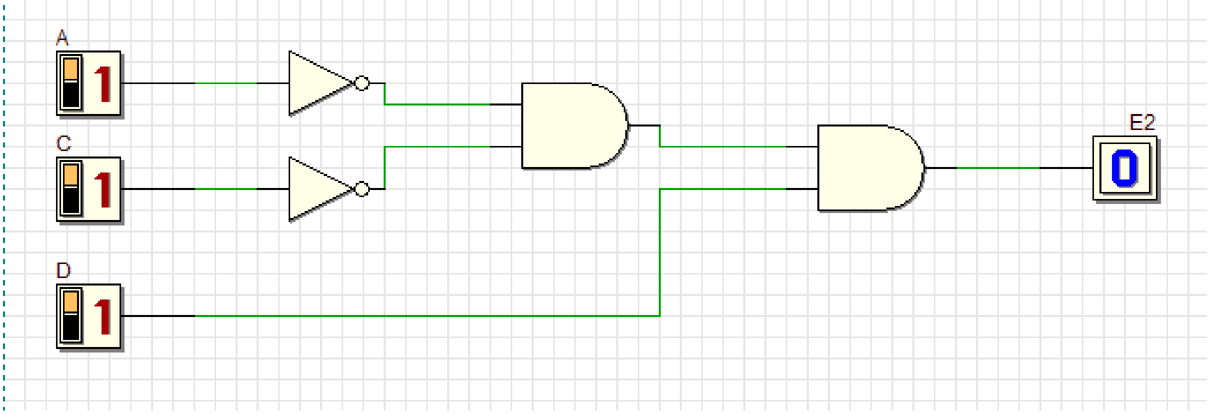

A
0
C
1
D
0
E2
0


A
0
C
1
D
1
E2
0

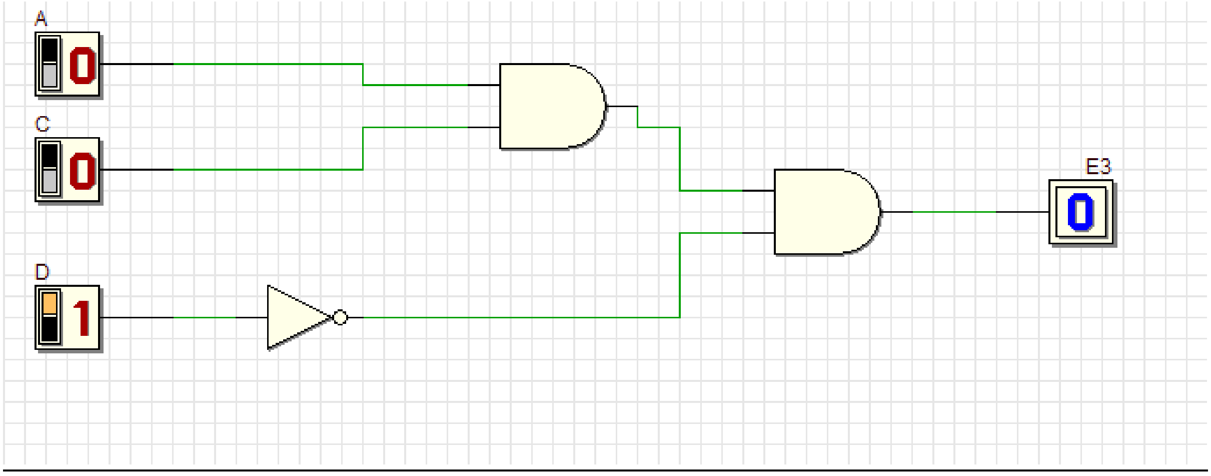


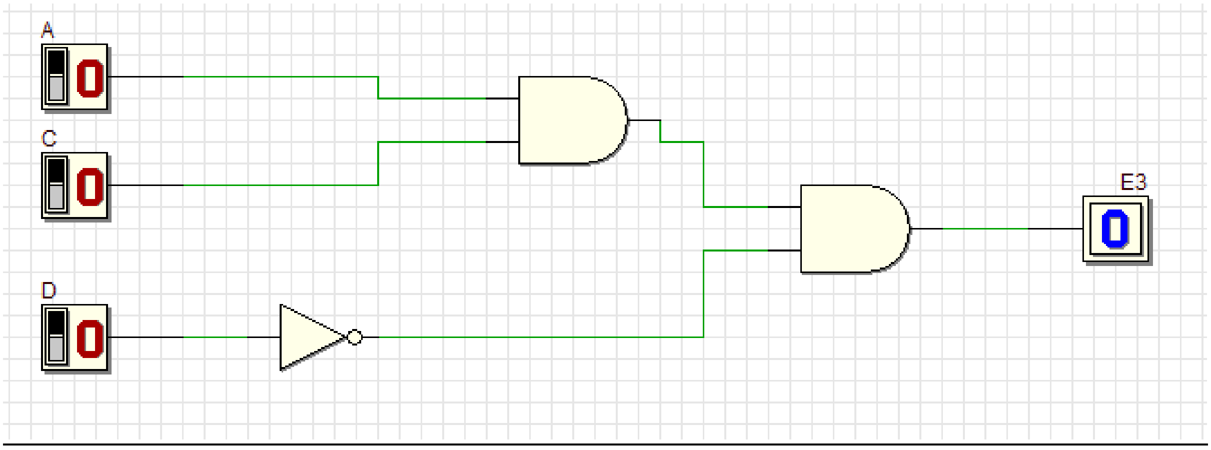


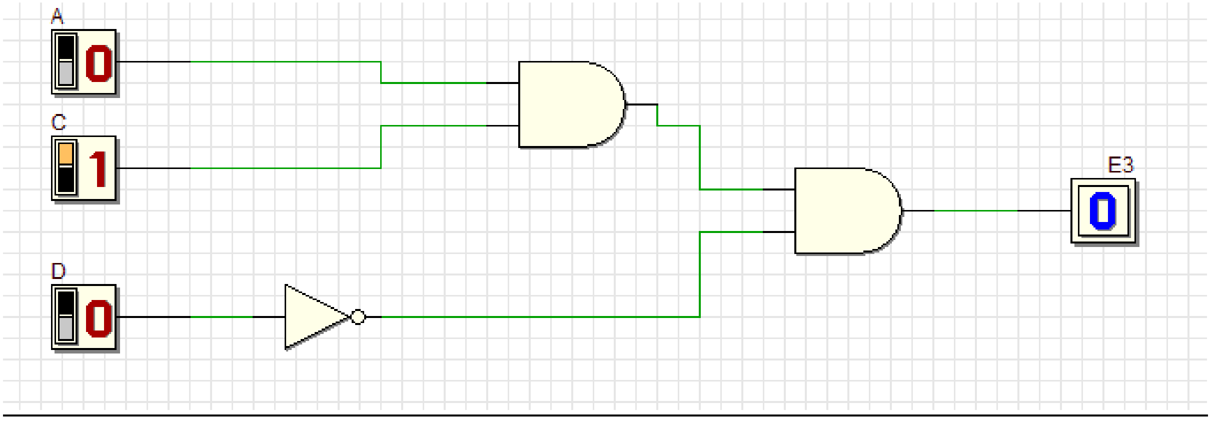


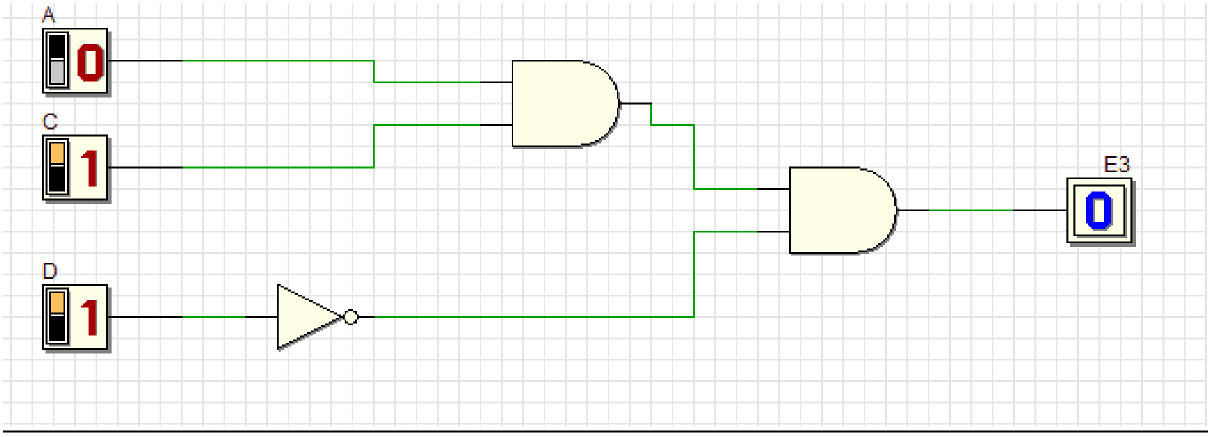


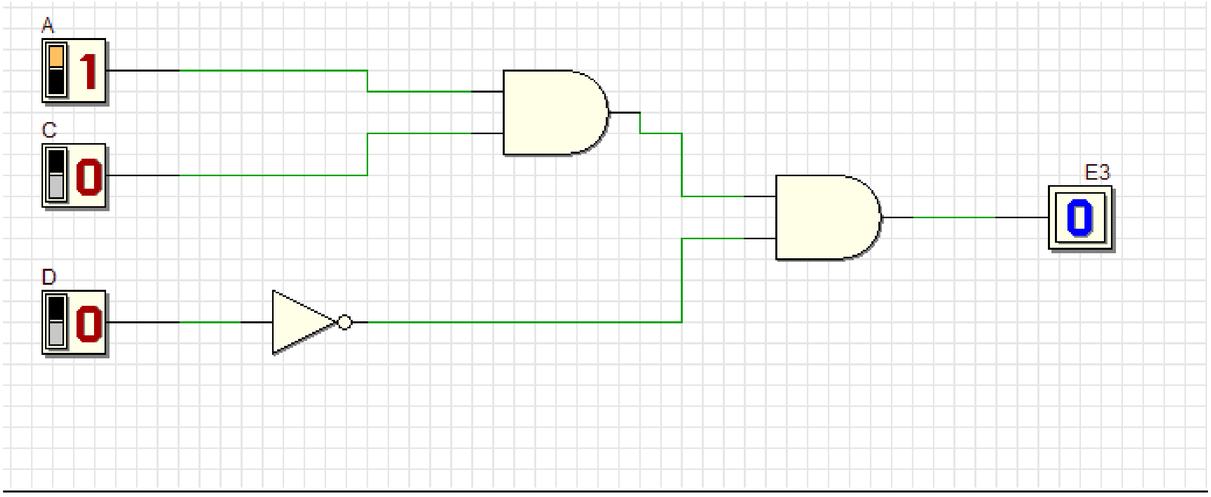
E3

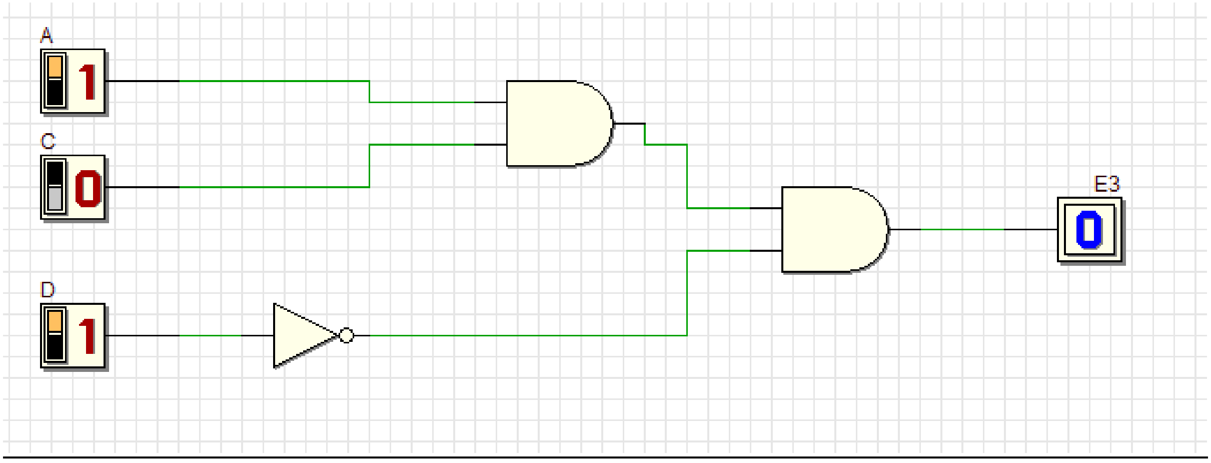

A
0
C
0
D
1
E3
0

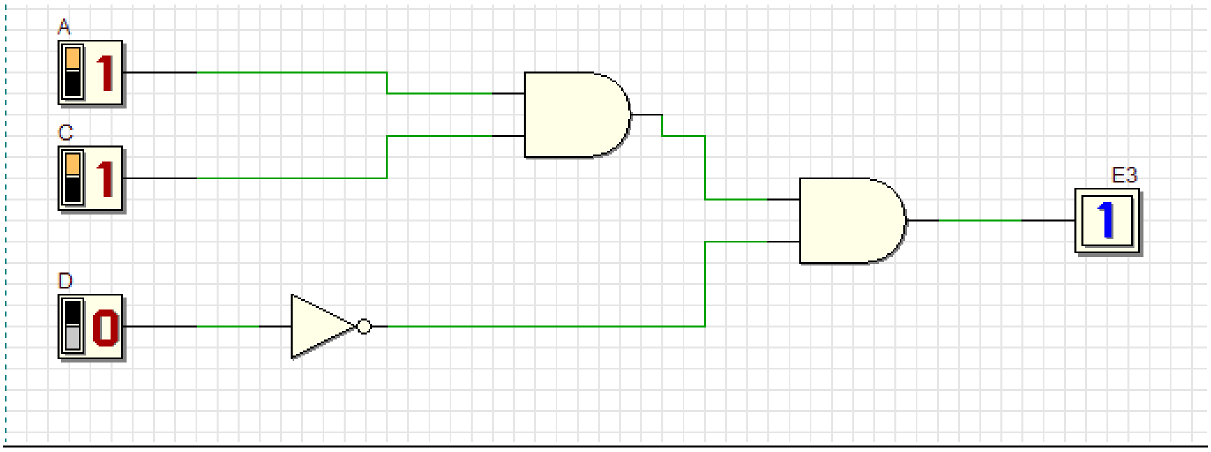

A
0
C
0
D
0
E3
0

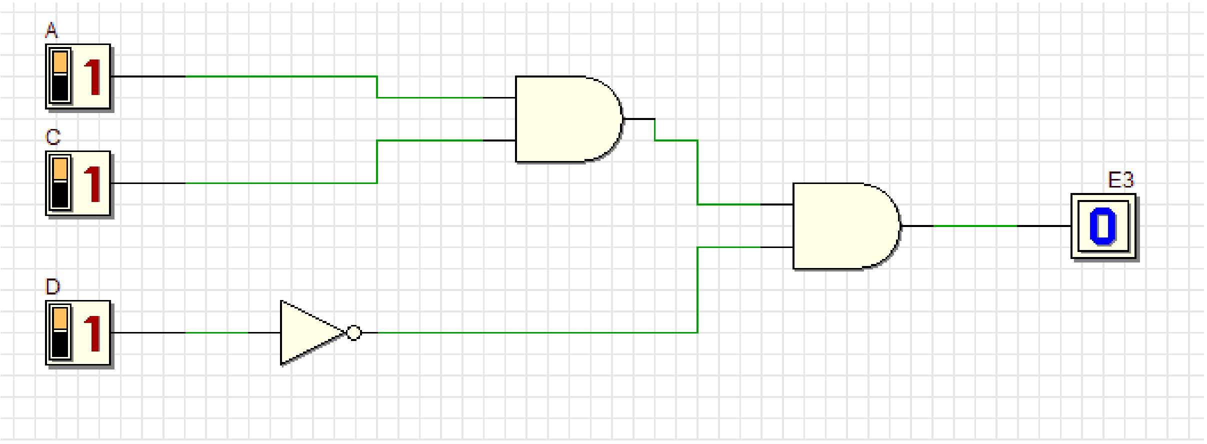

A
0
C
1
D
0
E3
0


A
0
C
1
D
1
E3
0

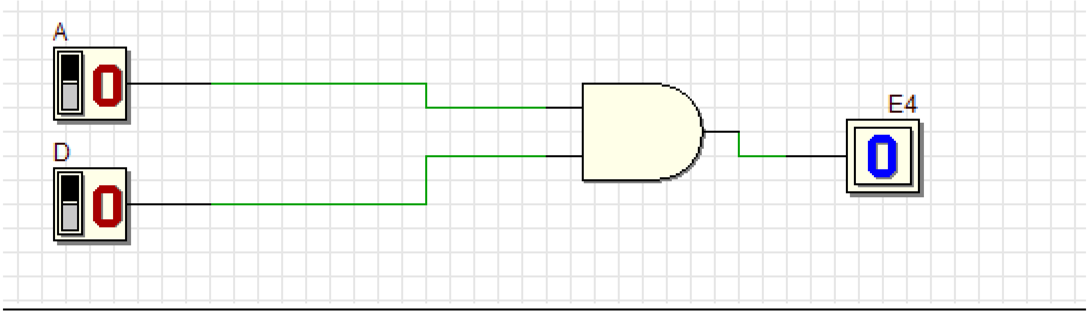

A
1
C
0
D
0
E3
0

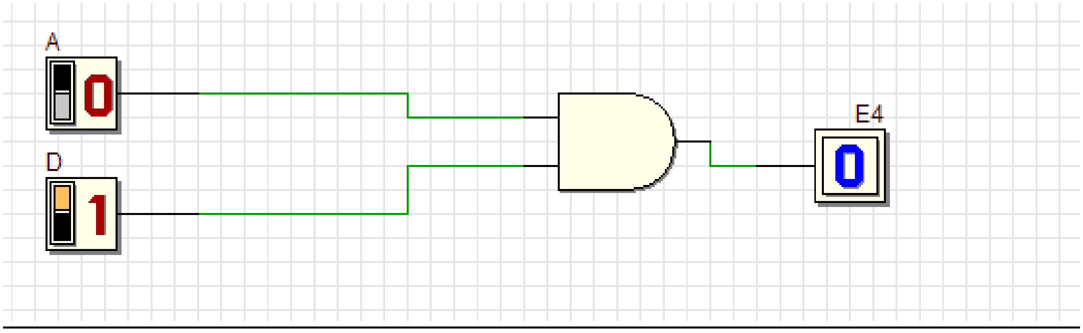

A
1
C
0
D
1
E3
0

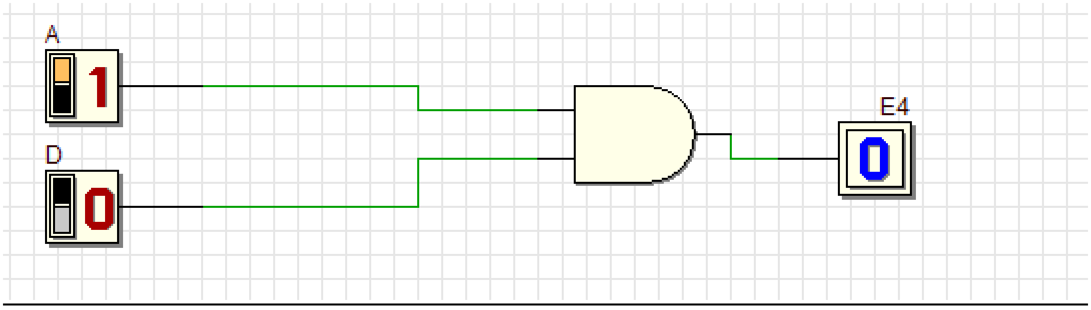

A
1
C
1
D
0
E3
1

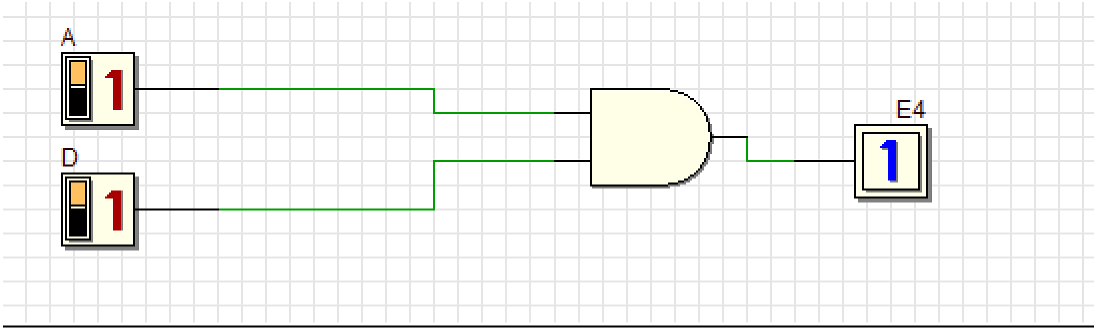

A
1
C
1
D
1
E3
0

E4

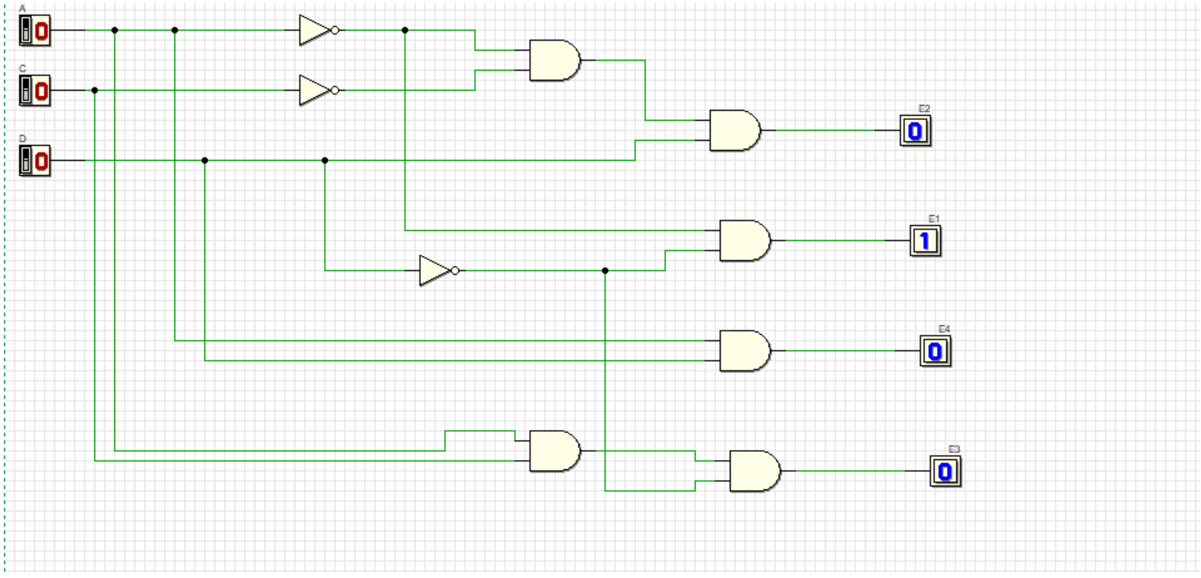


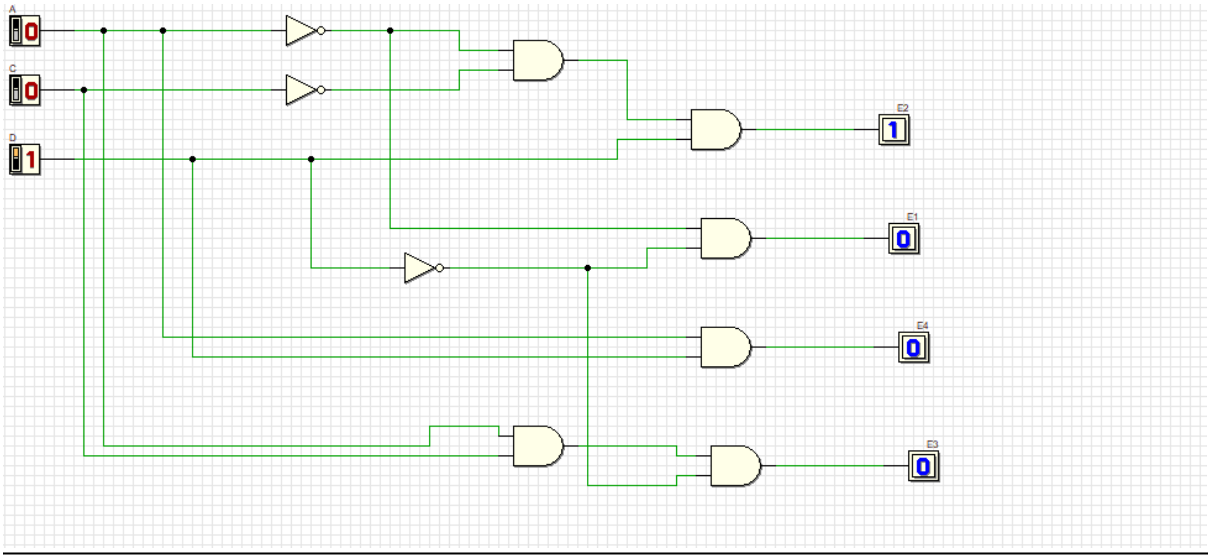


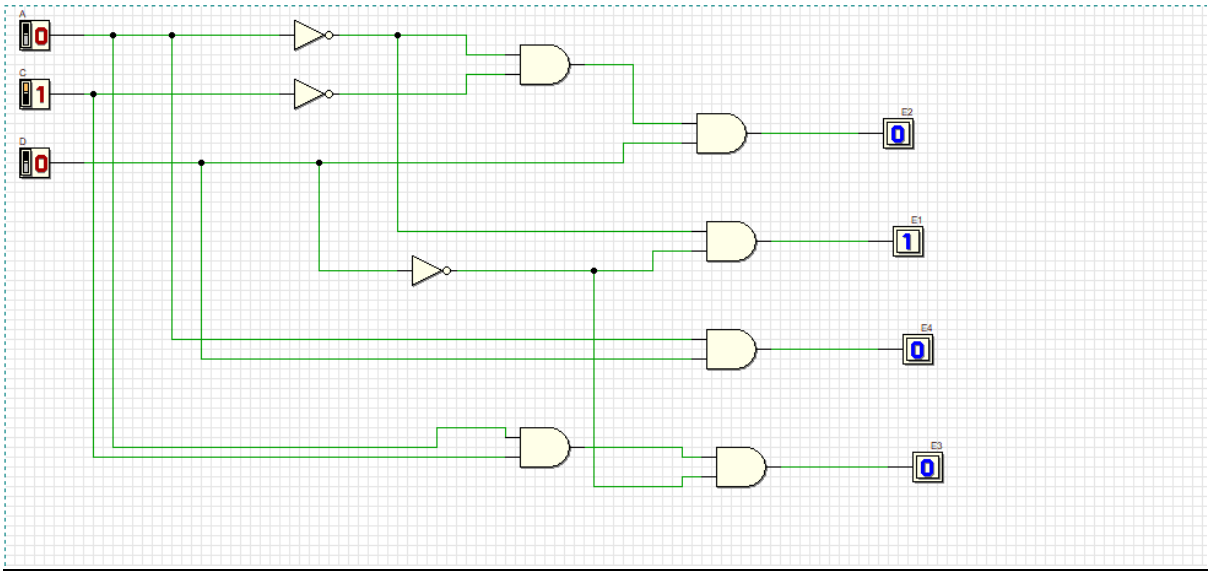


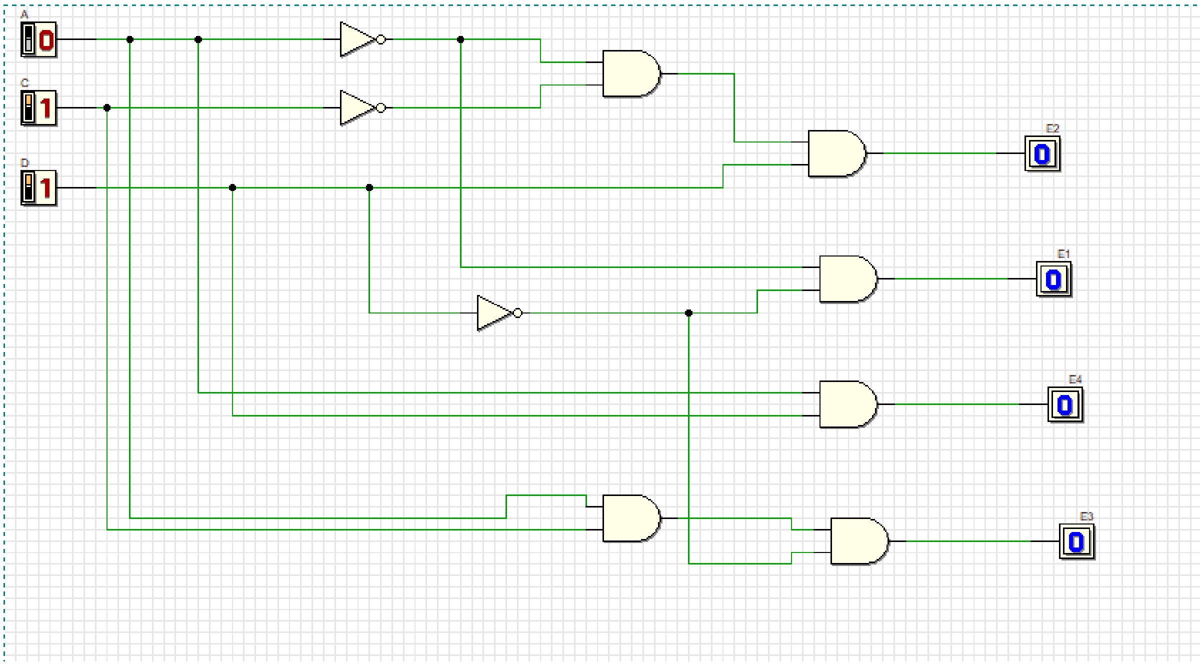


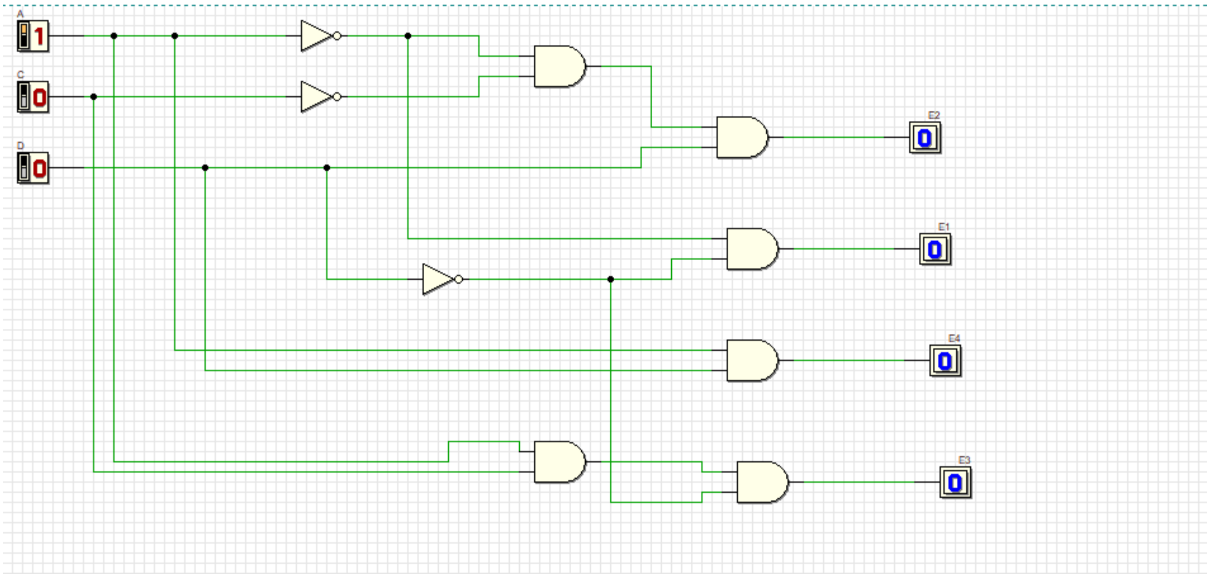
COMBINED CIRCUIT OF E1, E2, E3 AND E4

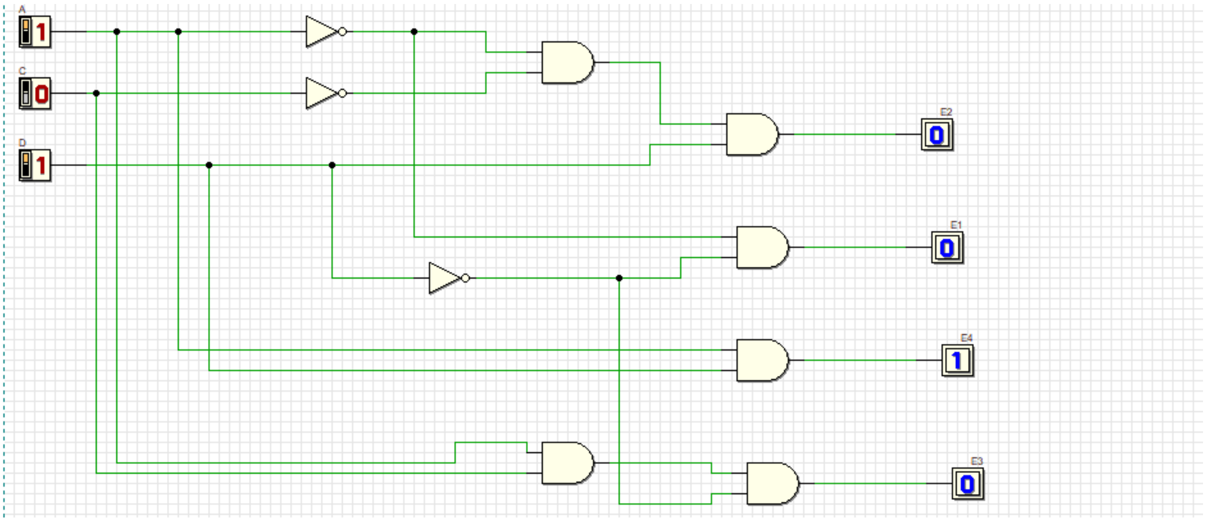

A
0
C
0
D
0
E2
0
E1
1
E4
0
E3
0

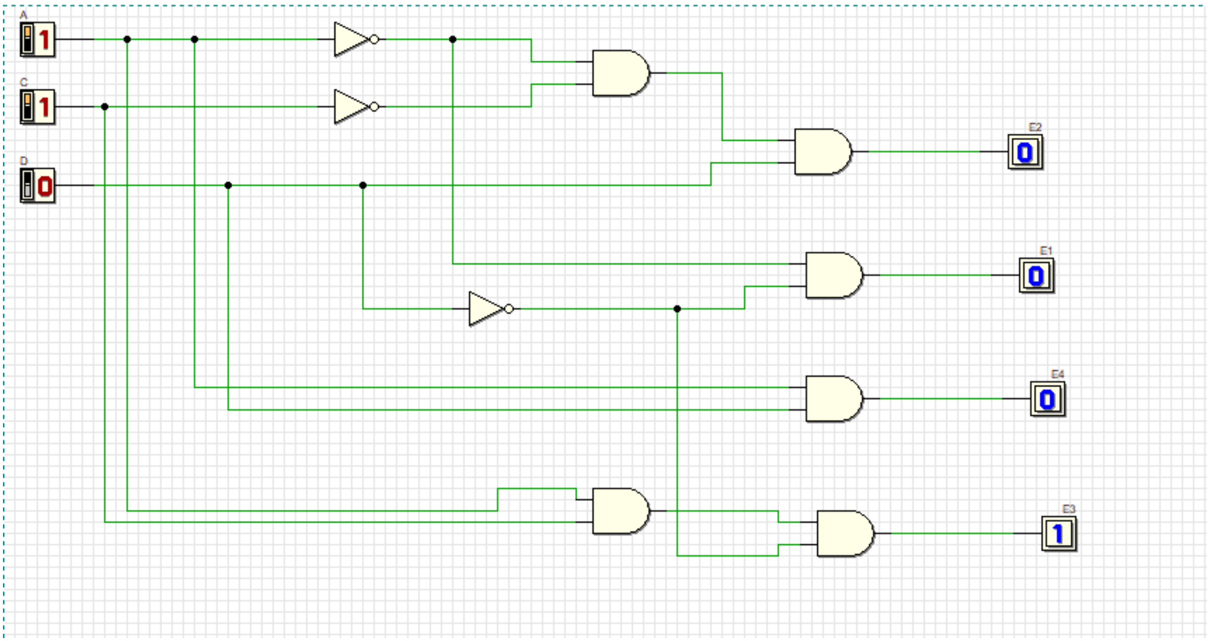

A
0
C
0
D
1
E2
1
E1
0
E4
0
E3
0

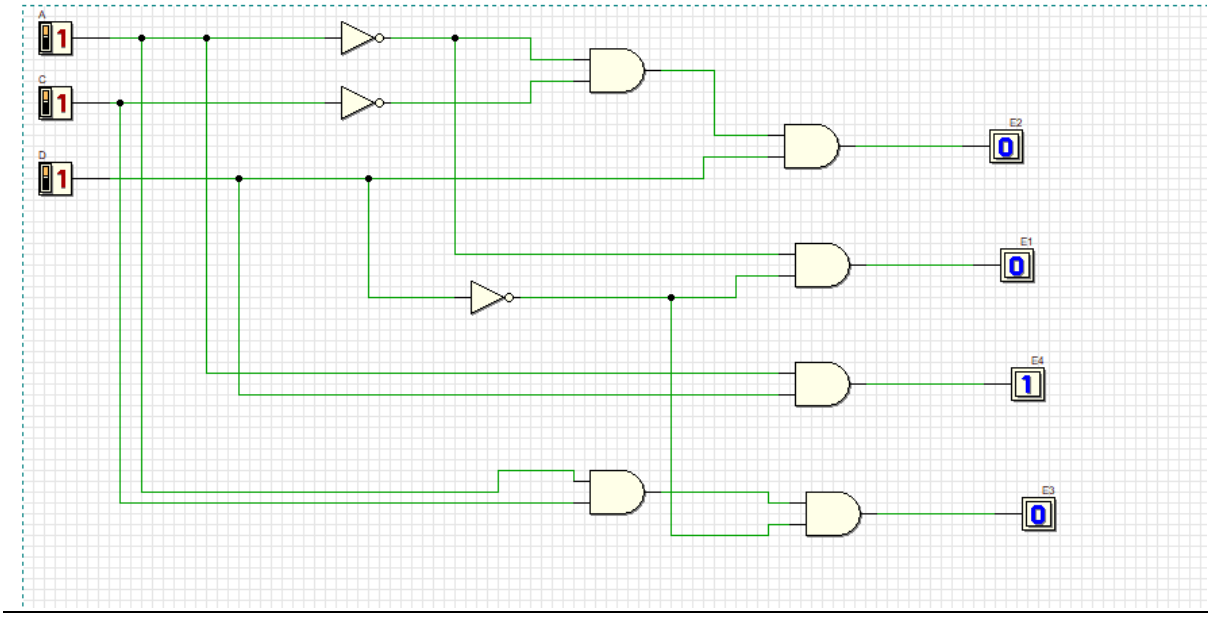

A
0
C
1
D
0
E2
0
E1
1
E4
0
E3
0


A
0
C
1
D
1
E2
0
E1
0
E4
0
E3
0


A
1
C
0
D
0
E2
0
E1
0
E4
0
E3
0


A
1
C
0
D
1
E2
0
E1
0
E4
1
E3
0





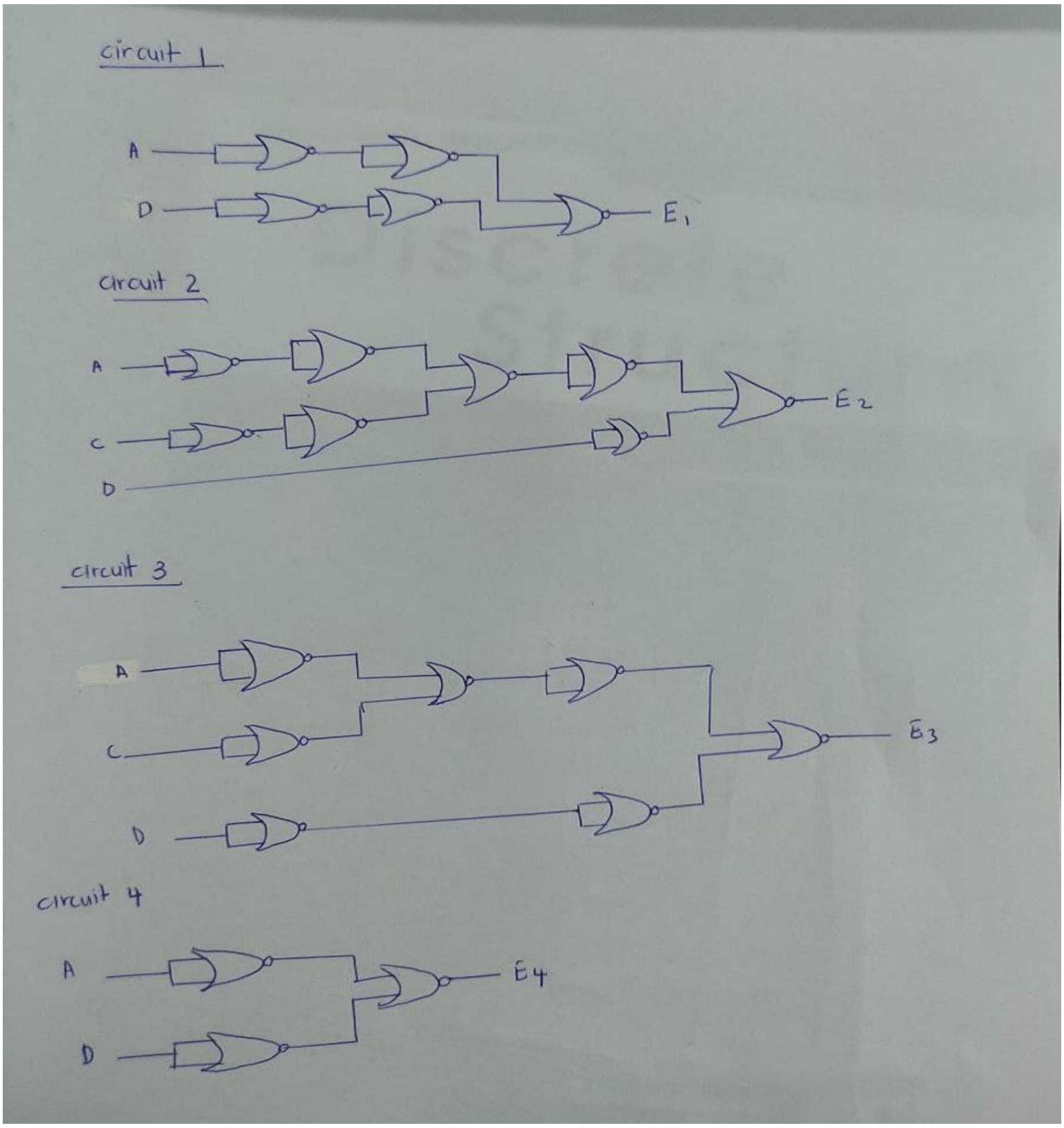
4. Simulate the Deeds circuit in step (3) and update the output (Truth Table 4) based on the simulation result.

Truth Table 4

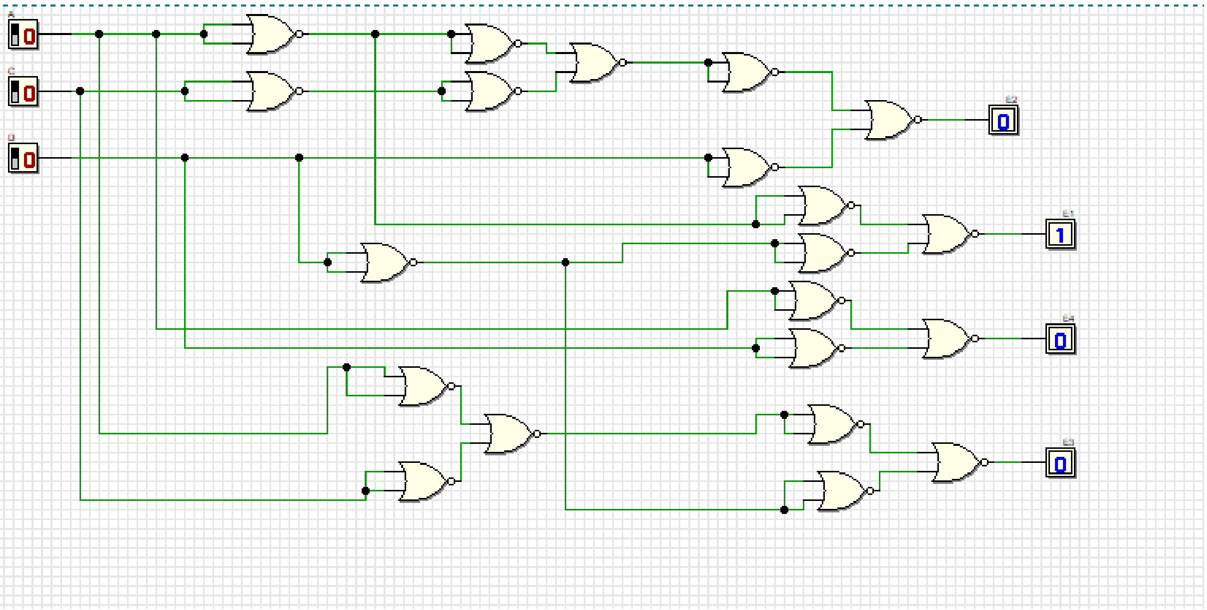
INPUT				OUTPUT			
A	**B**	**C**	**D**	**E1**	**E2**	**E3**	**E4**
0	**0**	**0**	**0**	1	0	0	0
0	**0**	**0**	**1**	0	1	0	0
0	**0**	**1**	**0**	1	0	0	0
0	**0**	**1**	**1**	0	0	0	0
0	**1**	**0**	**0**	1	0	0	0
0	**1**	**0**	**1**	0	1	0	0
0	**1**	**1**	**0**	1	0	0	0

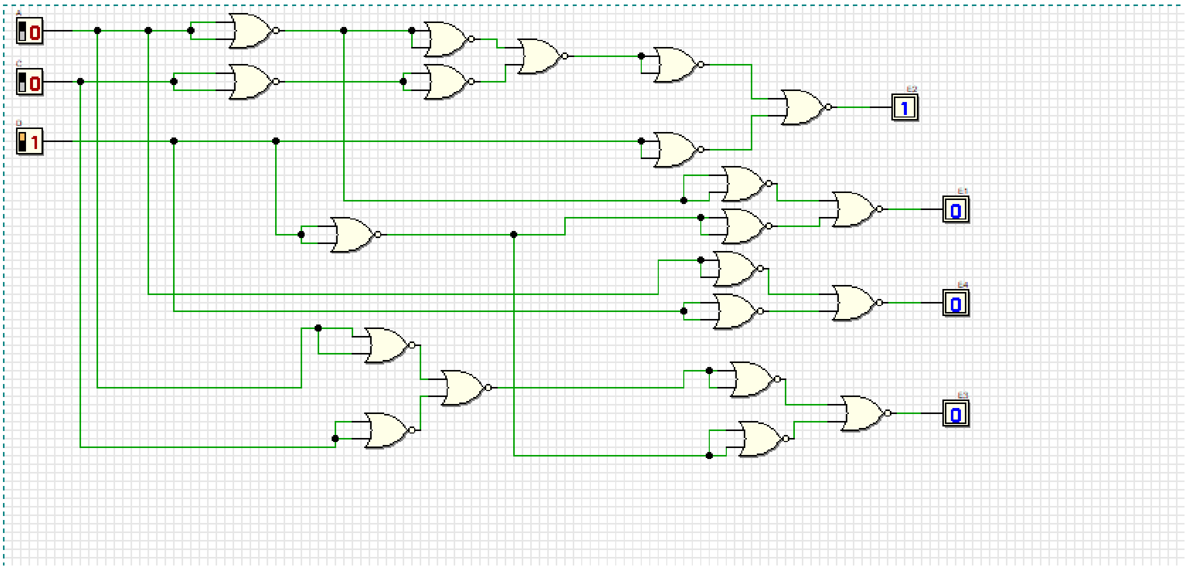
0	**1**	**1**	**1**	**0**	**0**	**0**	**0**
1	**0**	**0**	**0**	**0**	**0**	**0**	**0**
1	**0**	**0**	**1**	**0**	**0**	**0**	**1**
1	**0**	**1**	**0**	**0**	**0**	**1**	**0**
1	**0**	**1**	**1**	**0**	**0**	**0**	**1**
1	**1**	**0**	**0**	**0**	**0**	**0**	**0**
1	**1**	**0**	**1**	**0**	**0**	**0**	**1**
1	**1**	**1**	**0**	**0**	**0**	**1**	**0**
1	**1**	**1**	**1**	**0**	**0**	**0**	**1**

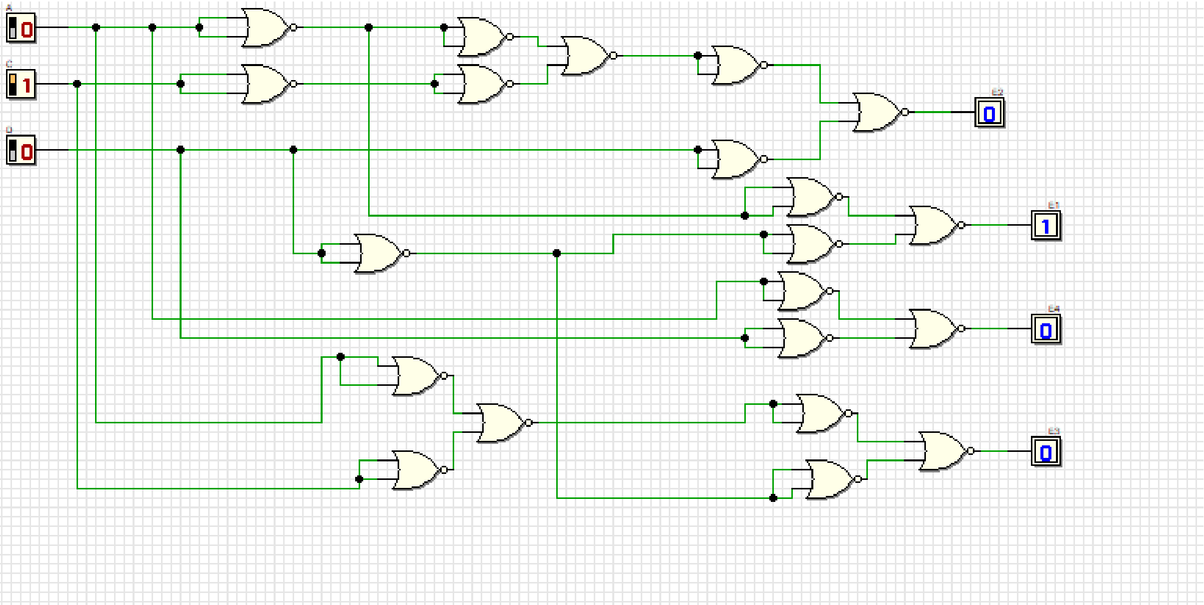
5. From the circuit in step (3), show the conversion of basic gates in E1, E2, E3 and E4 circuits to NOR universal gates. Post your workings here.

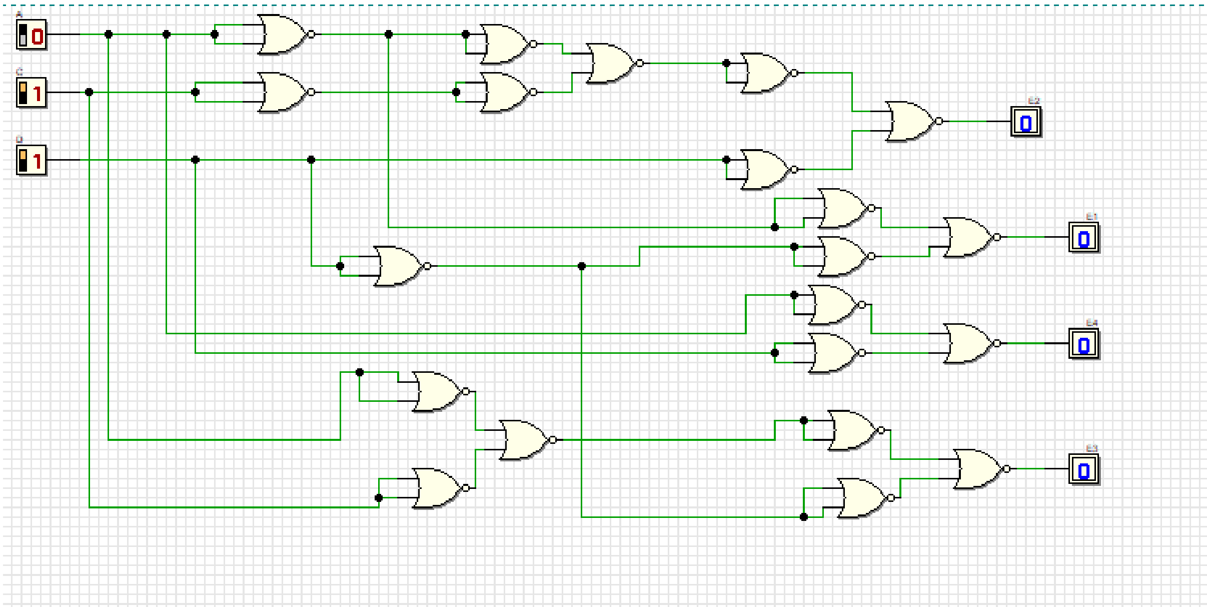


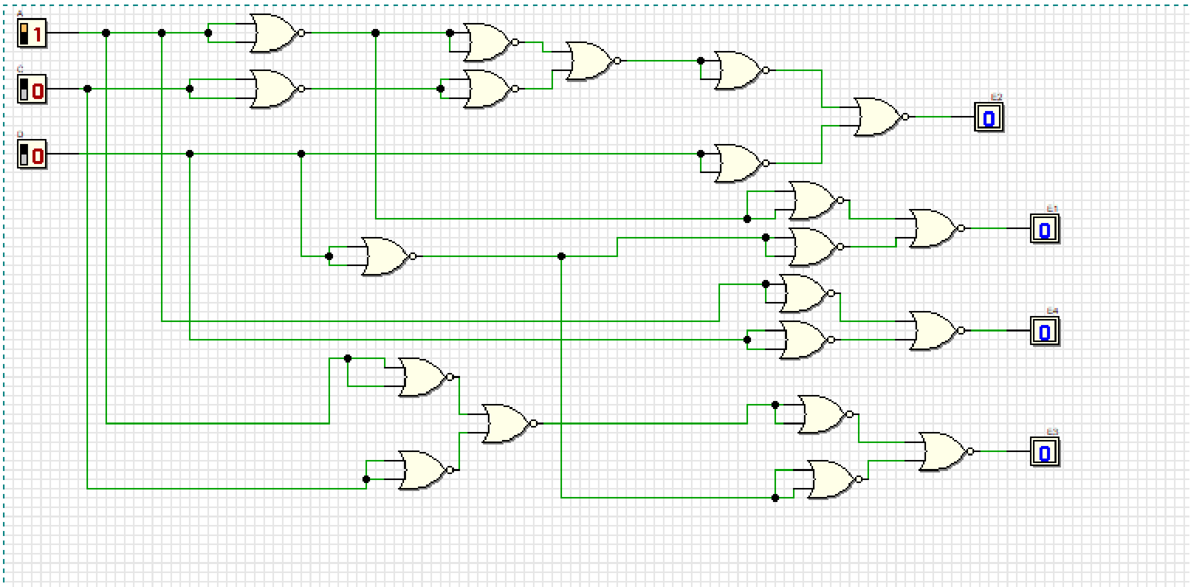
6. Draw the results of Step (5) in DEEDS. Post the circuits here.

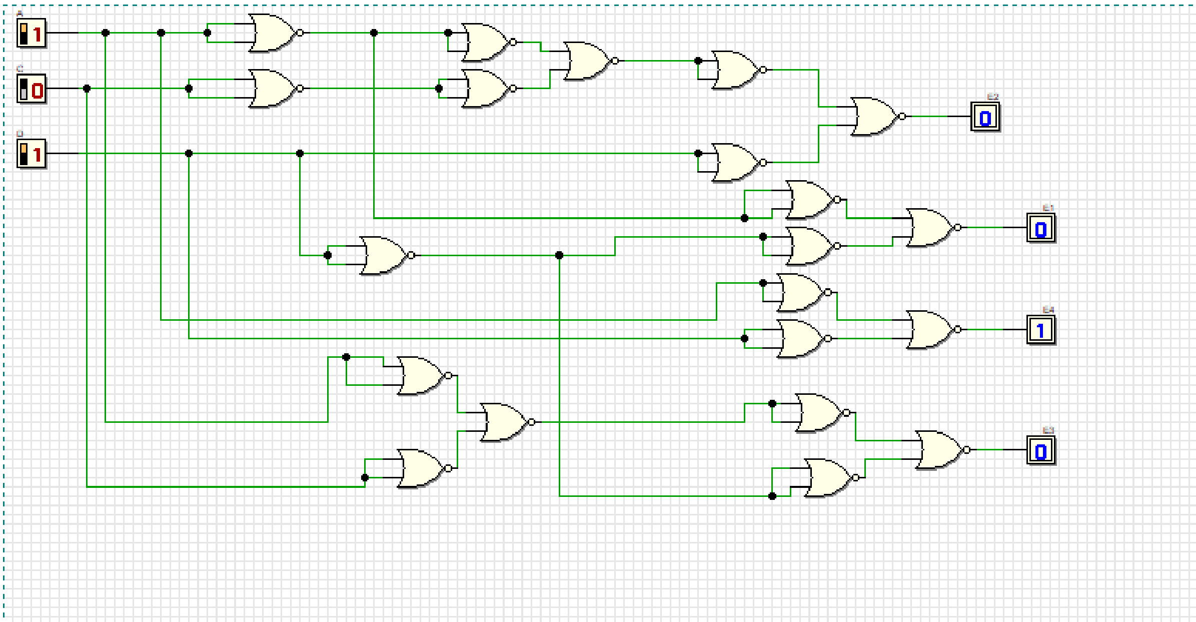

A
0
C
0
D
0
E2
0
E1
1
E4
0
E3
0

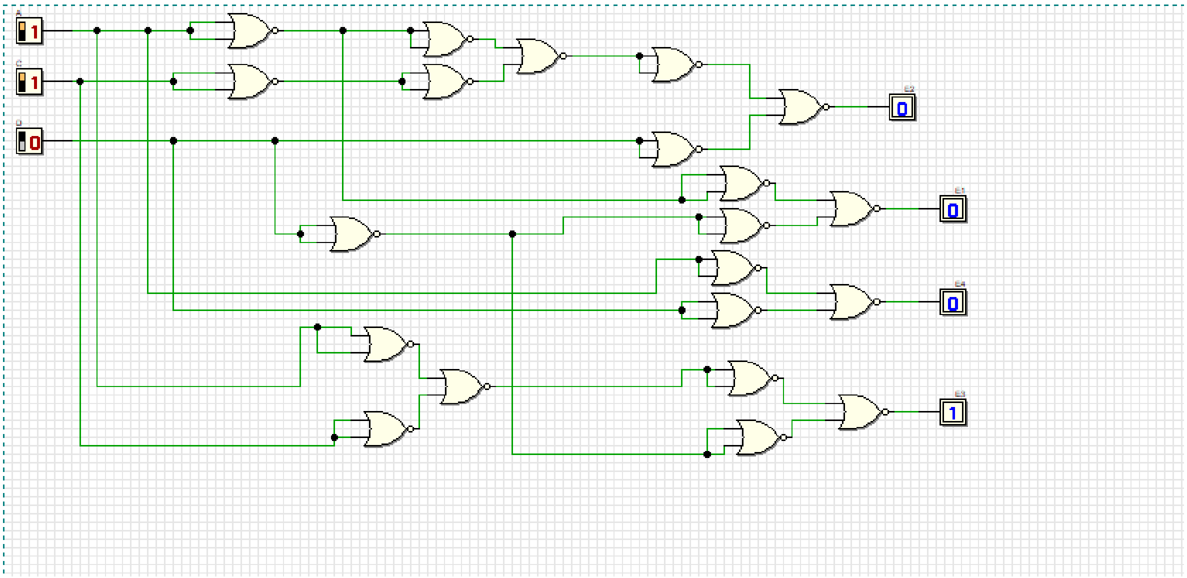

A
0
C
0
D
1
E2
1
E1
0
E4
0
E3
0

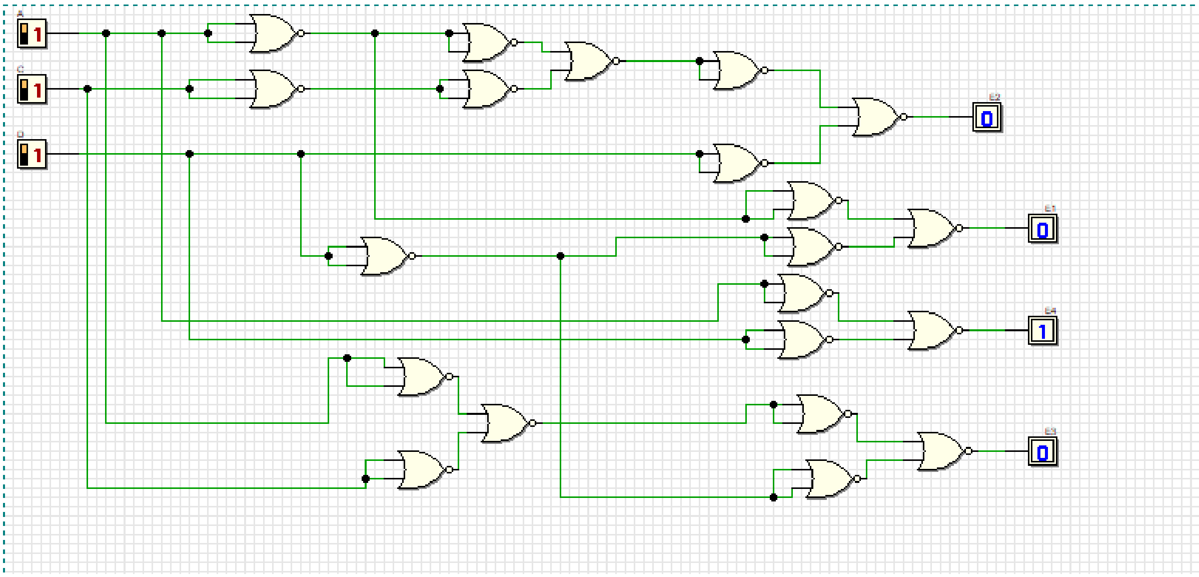

A
0
C
1
D
0
E2
0
E1
1
E4
0
E3
0


A
0
C
1
D
1
E2
0
E1
0
E4
0
E3
0


A
1
C
0
D
0
E2
0
E1
0
E4
0
E3
0


A
1
C
0
D
1
E2
0
E1
0
E4
1
E3
0





7. Confirm your circuit by simulating the circuits. Build Truth Table 5.

Truth Table 5

INPUT				OUTPUT			
A	**B**	**C**	**D**	**E1**	**E2**	**E3**	**E4**
0	**0**	**0**	**0**	1	0	0	0
0	**0**	**0**	**1**	0	1	0	0
0	**0**	**1**	**0**	1	0	0	0
0	**0**	**1**	**1**	0	0	0	0
0	**1**	**0**	**0**	1	0	0	0
0	**1**	**0**	**1**	0	1	0	0
0	**1**	**1**	**0**	1	0	0	0
0	**1**	**1**	**1**	0	0	0	0
1	**0**	**0**	**0**	0	0	0	0
1	**0**	**0**	**1**	0	0	0	1
1	**0**	**1**	**0**	0	0	1	0

1	**0**	**1**	**1**	**0**	**0**	**0**	**1**
1	**1**	**0**	**0**	**0**	**0**	**0**	**0**
1	**1**	**0**	**1**	**0**	**0**	**0**	**1**
1	**1**	**1**	**0**	**0**	**0**	**1**	**0**
1	**1**	**1**	**1**	**0**	**0**	**0**	**1**

8. Are all your Truth Tables 3, 4 and 5 equal or differ? Why are they the same and/or why are they different? Give your reasonings.

The truth table 3 and truth table 4 is not same because the outputs are not the same. It is because, in the truth table 3 'Don't care' condition is applied for all the output which has equal number of '0' and '1' bits in the input. In truth table 4 'Don't care' condition is not applied. So, except for the output which applied 'Don't care' condition in truth table 3, all the output in truth table 3 is same as the output in output in truth table 4. Circuit in step 3 does not fulfil rule 5. Not only that, but input B is also not included to get output E1, E2, E3 & E4. So, it means whether input B is high or low, it does not influence the outputs.

The output in truth table 4 and output in truth table 5 is the same. It is because, in circuit step 3 and circuit in step 5 has the same input. Circuit in step 3 is combinational circuit with basic gates and circuit in step 5 is same as circuit in step 3 but the basic gates in step 3 are replaced with its equivalent NOR-gates. One NOR-gate is equal to inverter gate and 3 NOR-gate equals to AND-gate, even though the type of gates is different the function of the gate is the same in both circuits.

Edited by Marina Md Arshad, 28112021