

# Digital Logic Practical Lab Report

Comparator Circuit and Parity  
Checker/Generator

Student:

Abhiroop Mukherjee

Teacher:

Prof. Surajeet Ghosh

Prof. Sekhar Mandal

Course:

Digital Logic Laboratory (CS351)

## EXPERIMENT 1 (a)

### Objective

Design 1-bit comparator using XOR and NAND

Gates with three possible condition; ie  $A > B$ ,  $A = B$ ,  
and  $A < B$ , where A and B are two 1 bit words.

### Theory

- A Magnitude comparator is a digital circuit that compares two digital num or binary numbers in order to find out whether one binary number is equal, less than or greater than the other binary numbers.
- A comparator used to compare two bits is called a single bit comparator. It consists of two single bit input and three outputs to generate less than, equal to and greater than between two binary numbers.
- Truth table of 1 bit comparator:

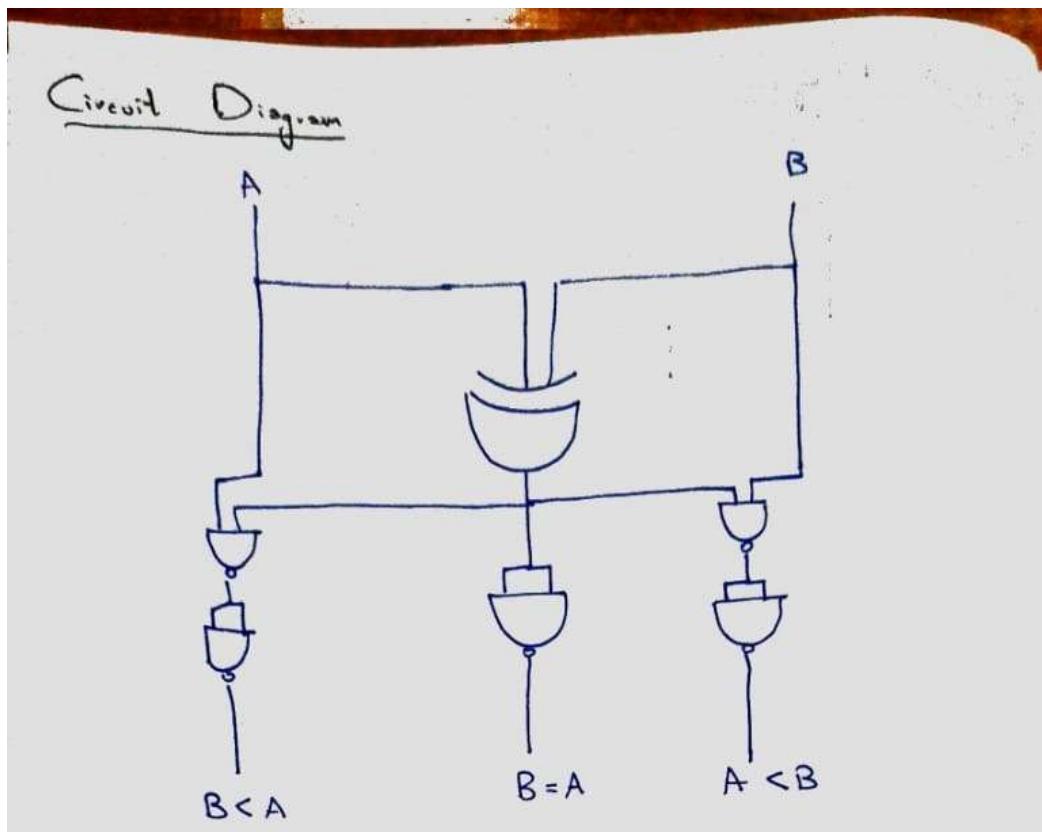
| Inputs |   | Outputs |         |         |
|--------|---|---------|---------|---------|
| A      | B | $A < B$ | $A = B$ | $A > B$ |
| 0      | 0 | 0       | 1       | 0       |
| 0      | 1 | 1       | 0       | 0       |
| 1      | 0 | 0       | 0       | 1       |
| 1      | 1 | 0       | 1       | 0       |

From the above truth table, we can express,

$$A > B : AB' = A\bar{B}$$

$$A < B : A'B = \bar{A}B$$

$$A = B : \bar{A}\bar{B} + AB = A'B' + AB = A \oplus B$$



## Input/Output

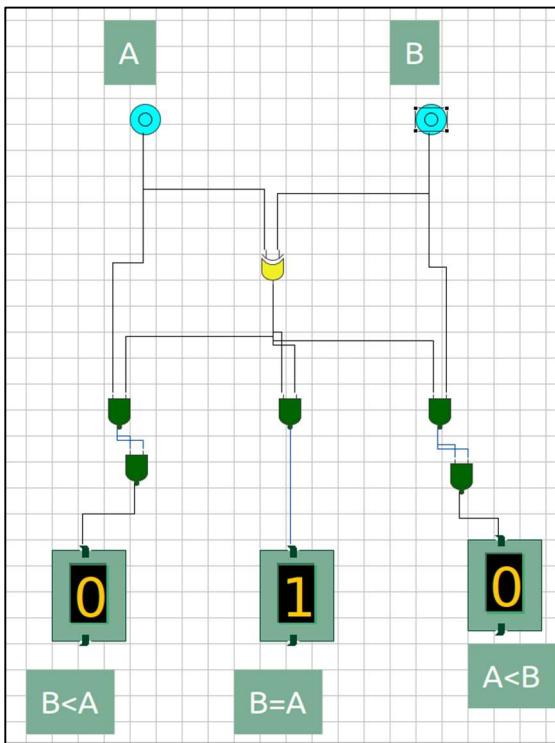


Figure 1:  $A = 0; B = 0$

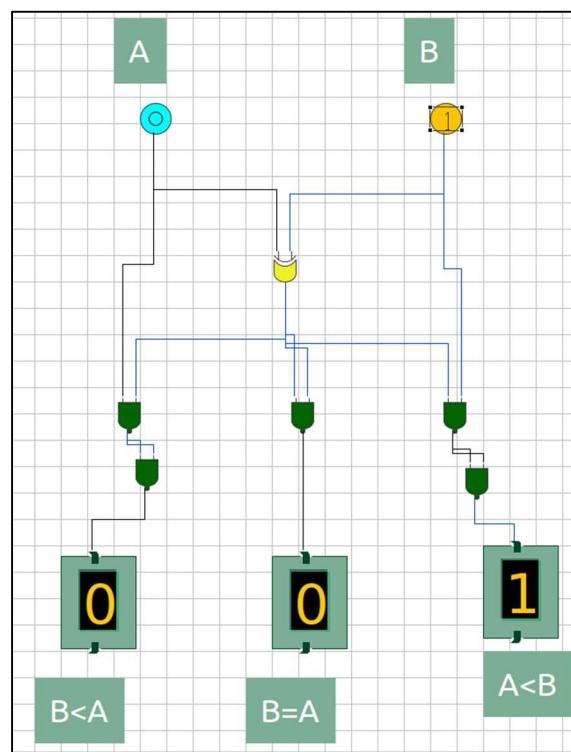


Figure 2:  $A = 0; B = 1$

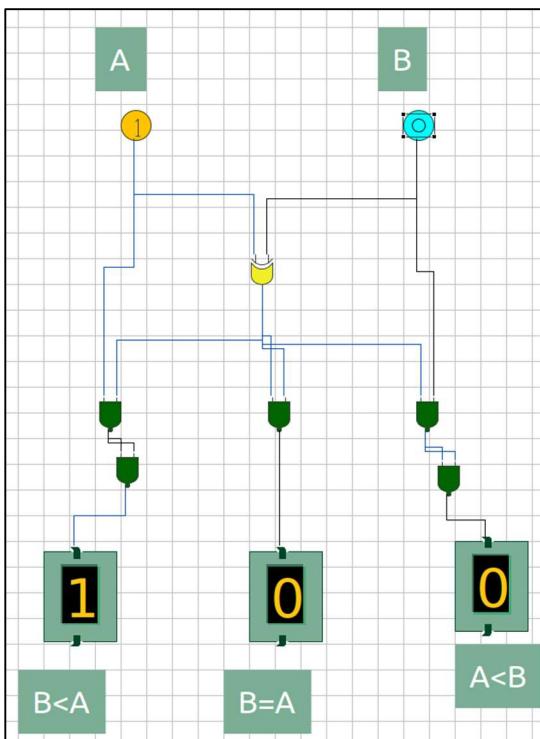


Figure 3:  $A = 1; B = 0$

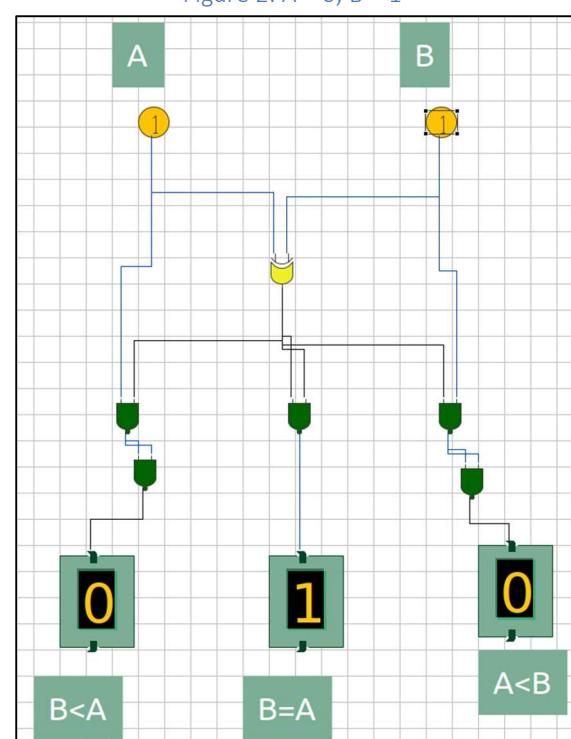


Figure 4:  $A = 1; B = 1$

## EXPERIMENT 1(b)

### Objective

Use 1-bit comparators block to design 2-bit comparator where the outputs are showing three possible condition; i.e.,  $A_1A_0 > B_1B_0$ ,  $A_1A_0 = B_1B_0$ ,  $A_1A_0 < B_1B_0$ .

### Theory

→ The Truth Table for 2 bit comparator:

| Input |    |    |    | Output |       |       |
|-------|----|----|----|--------|-------|-------|
| A1    | A0 | B1 | B0 | A < B  | A = B | A > B |
| 0     | 0  | 0  | 0  | 0      | 1     | 0     |
| 0     | 0  | 0  | 1  | 1      | 0     | 0     |
| 0     | 0  | 1  | 0  | 1      | 0     | 0     |
| 0     | 0  | 1  | 1  | 1      | 0     | 0     |
| 0     | 1  | 0  | 0  | 0      | 0     | 1     |
| 0     | 1  | 0  | 1  | 0      | 1     | 0     |
| 0     | 1  | 1  | 0  | 1      | 0     | 0     |
| 0     | 1  | 1  | 1  | 1      | 0     | 0     |
| 1     | 0  | 0  | 0  | 0      | 0     | 1     |
| 1     | 0  | 0  | 1  | 0      | 0     | 1     |
| 1     | 0  | 1  | 0  | 0      | 1     | 0     |
| 1     | 0  | 1  | 1  | 1      | 0     | 0     |
| 1     | 1  | 0  | 0  | 0      | 0     | 1     |
| 1     | 1  | 0  | 1  | 0      | 0     | 1     |
| 1     | 1  | 1  | 0  | 0      | 0     | 1     |
| 1     | 1  | 1  | 1  | 0      | 1     | 0     |

From the above truth table, we get the following:

$$\text{I) } A > B = \Sigma (4, 8, 9, 12, 13, 14)$$

|  |  | B <sub>1</sub> B <sub>0</sub> |    |    |    |    |
|--|--|-------------------------------|----|----|----|----|
|  |  | A <sub>1</sub> A <sub>0</sub> | 00 | 01 | 11 | 10 |
|  |  | 00                            | 1  |    |    |    |
|  |  | 01                            |    | 1  |    |    |
|  |  | 11                            | 1  | 1  |    | 1  |
|  |  | 10                            | 1  | 1  |    |    |

$$A > B = A_1 B_1' + A_0 B_1' B_0 + A_1 A_0 B_0'$$

$$\text{II) } A = B$$

|  |  | B <sub>1</sub> B <sub>0</sub> |    |    |    |    |
|--|--|-------------------------------|----|----|----|----|
|  |  | A <sub>1</sub> A <sub>0</sub> | 00 | 01 | 11 | 10 |
|  |  | 00                            | 1  |    |    |    |
|  |  | 01                            |    | 1  |    |    |
|  |  | 11                            |    |    | 1  |    |
|  |  | 10                            |    |    |    | 1  |

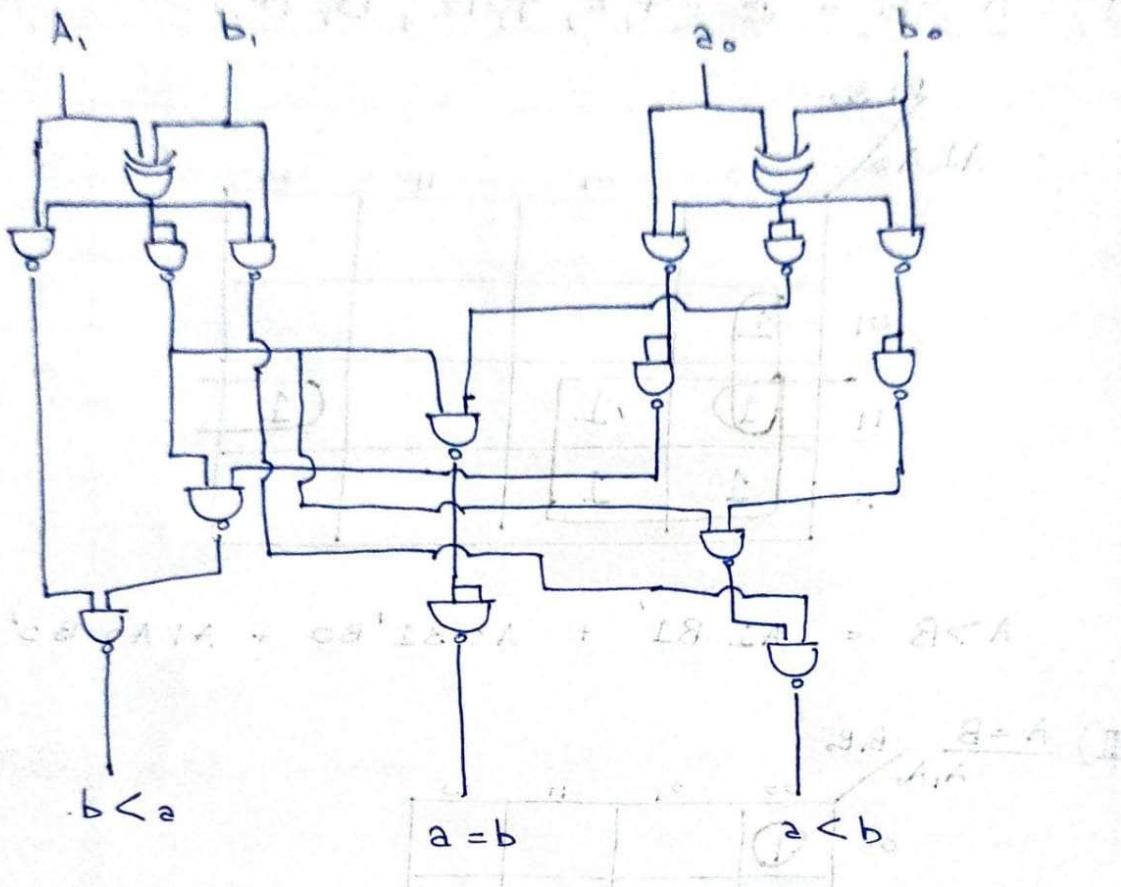
$$\begin{aligned} A = B : & A_1' A_0' B_1' B_0 + A_1' A_0 B_1' B_0 + A_1 A_0 B_1 B_0 + A_1 A_0' B_1' B_0 \\ & \Rightarrow (A_0 B_0 + A_0' B_0') \neq (A_1 B_1 + A_1' B_1') \\ & \therefore (A_0 \text{ XNOR } B_0) \quad (A_1 \text{ XNOR } B_1) \end{aligned}$$

$$\text{III) } A < B$$

|  |  | B <sub>1</sub> B <sub>0</sub> |    |    |    |    |
|--|--|-------------------------------|----|----|----|----|
|  |  | A <sub>1</sub> A <sub>0</sub> | 00 | 01 | 11 | 10 |
|  |  | 00                            | 1  | 1  |    |    |
|  |  | 01                            |    | 1  | 1  |    |
|  |  | 11                            |    |    |    | 1  |
|  |  | 10                            |    |    |    |    |

$$\begin{aligned} A < B &= A_1' B_1 + A_0' B_1 B_0 + A_1' B_0 A_0' \\ &= A_1' B_1 + A_0' B_1 B_0 + A_1' A_0' B_0 \end{aligned}$$

## Circuit Diagram



## Result

As we can see from the simulation that minimum number of XOR gates and NAND gates used is, i.e 2 XOR, and 14-NAND gates used, and simulation results match with the expected output of truth table.

## Input / Output

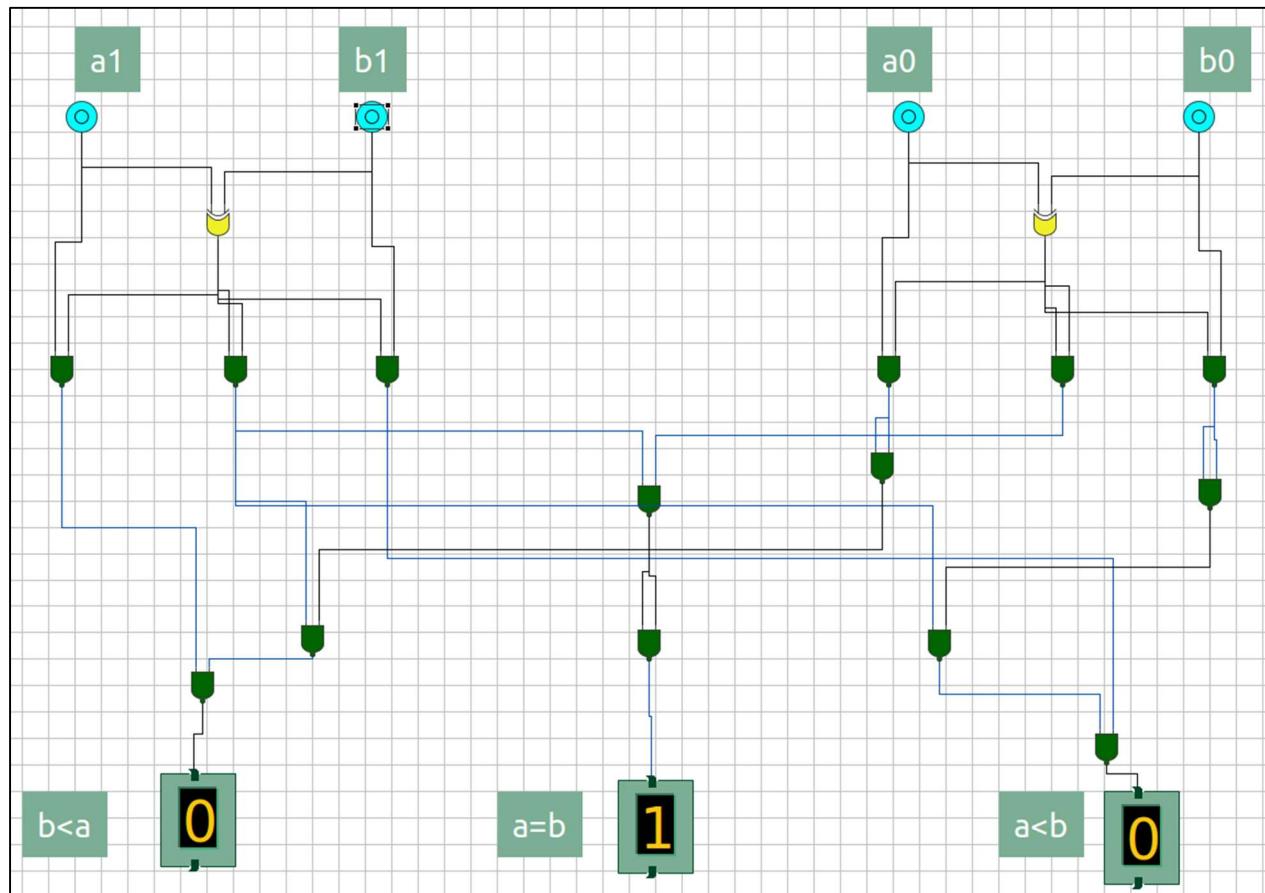


Figure 1: a1 = 0; a0 = 0; b1 = 0; b0 = 0

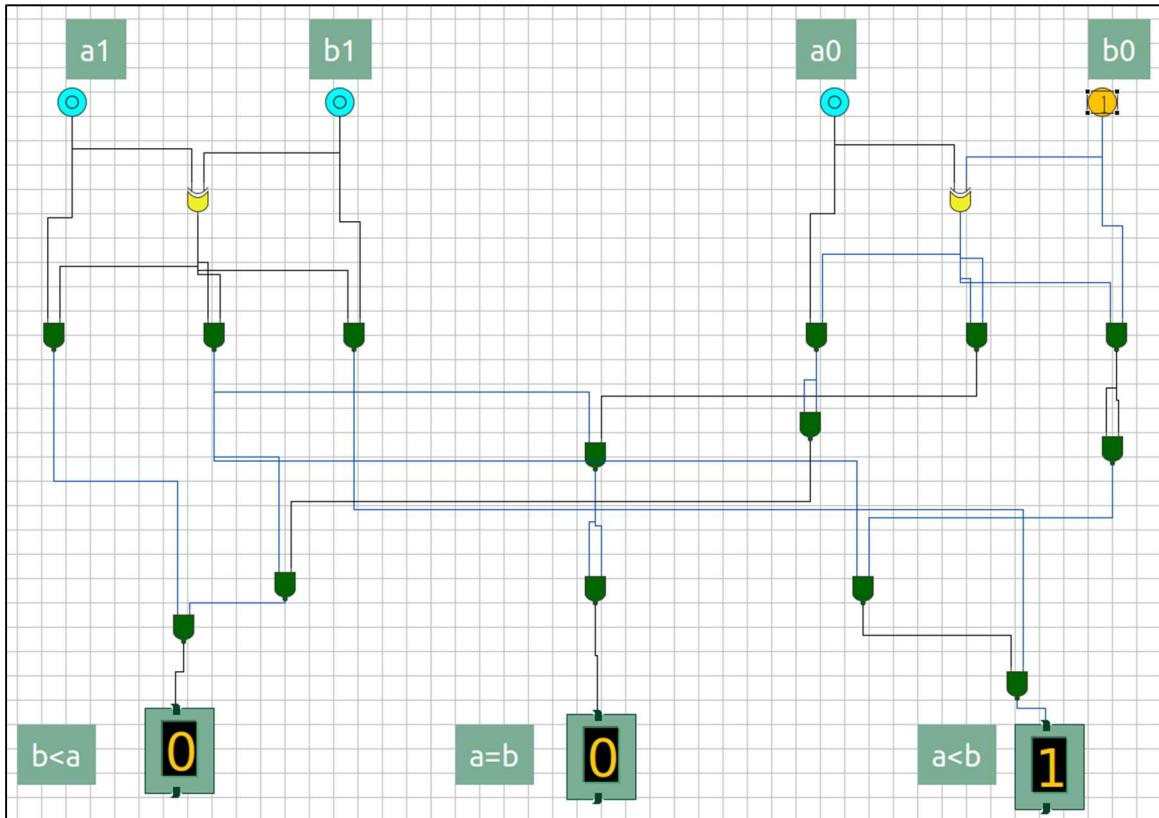


Figure 2:  $a_1 = 0$ ;  $a_0 = 0$ ;  $b_1 = 0$ ;  $b_0 = 1$

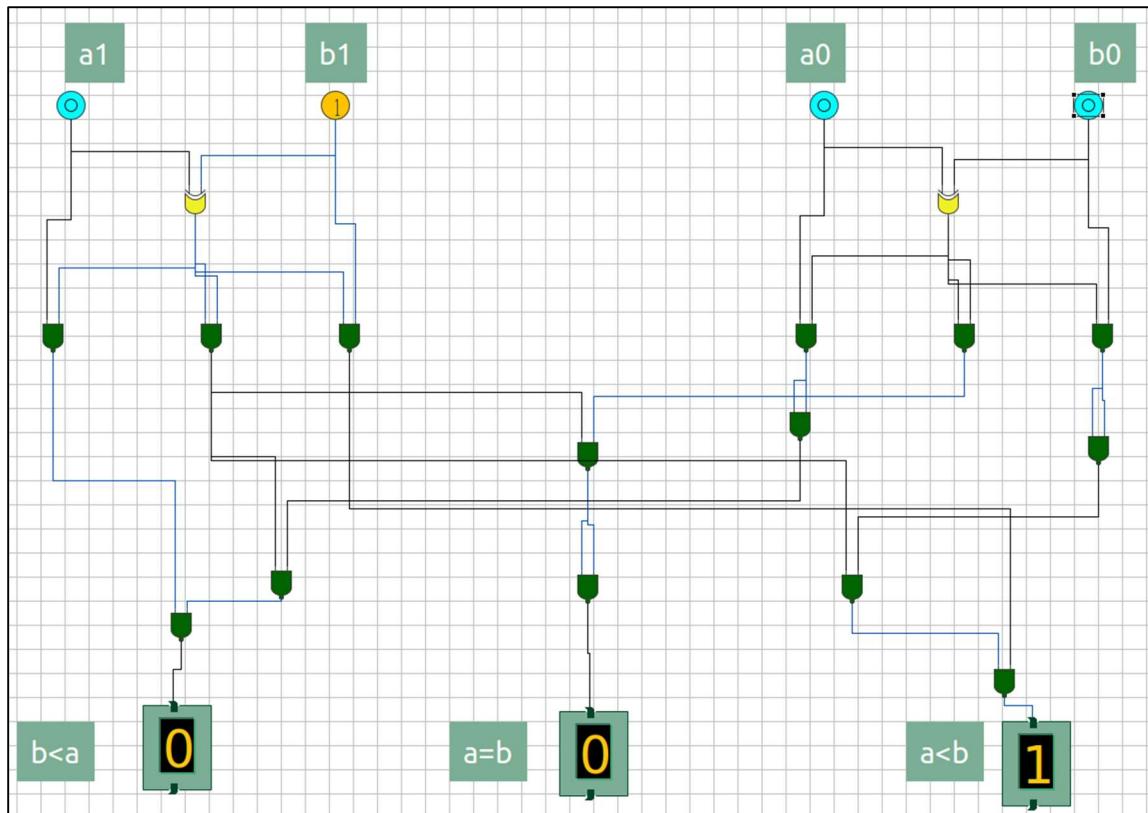


Figure 3:  $a_1 = 0$ ;  $a_0 = 0$ ;  $b_1 = 1$ ;  $b_0 = 0$

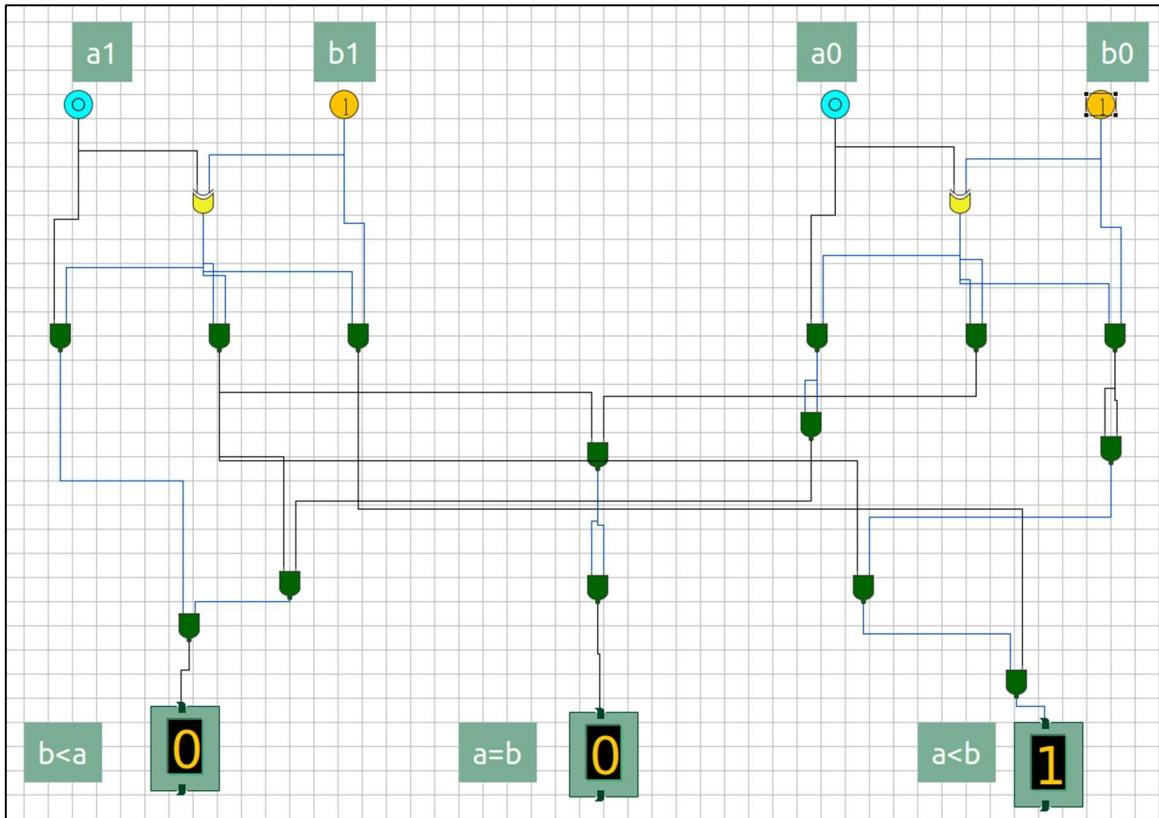


Figure 4:  $a_1 = 0; a_0 = 0; b_1 = 1; b_0 = 1$

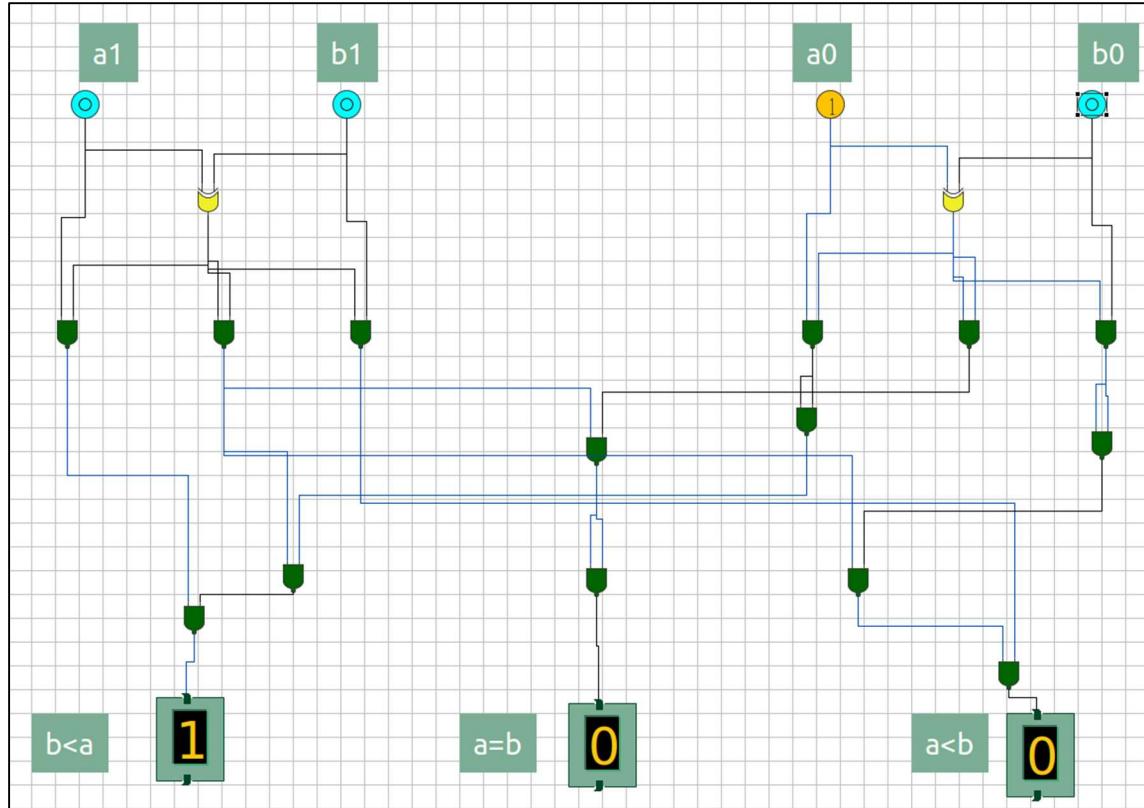


Figure 5:  $a_1 = 0; a_0 = 1; b_1 = 0; b_0 = 0$

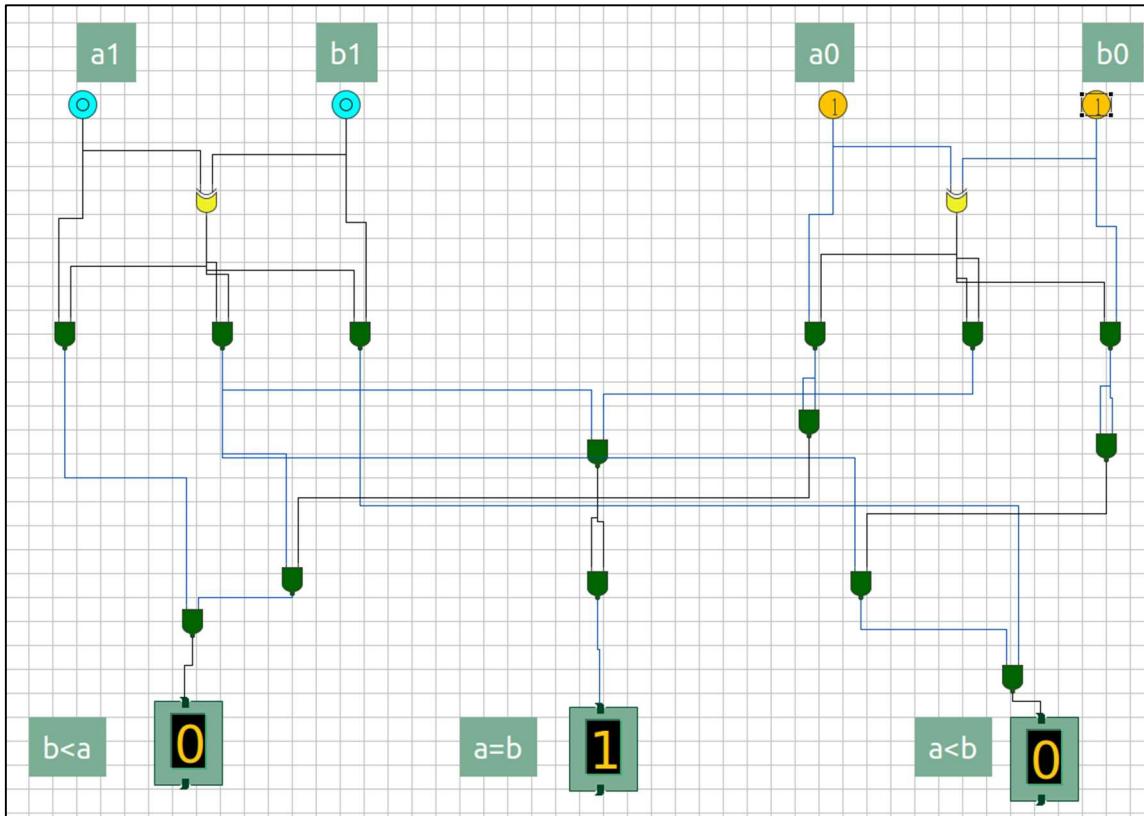


Figure 6:  $a_1 = 0$ ;  $a_0 = 1$ ;  $b_1 = 0$ ;  $b_0 = 1$

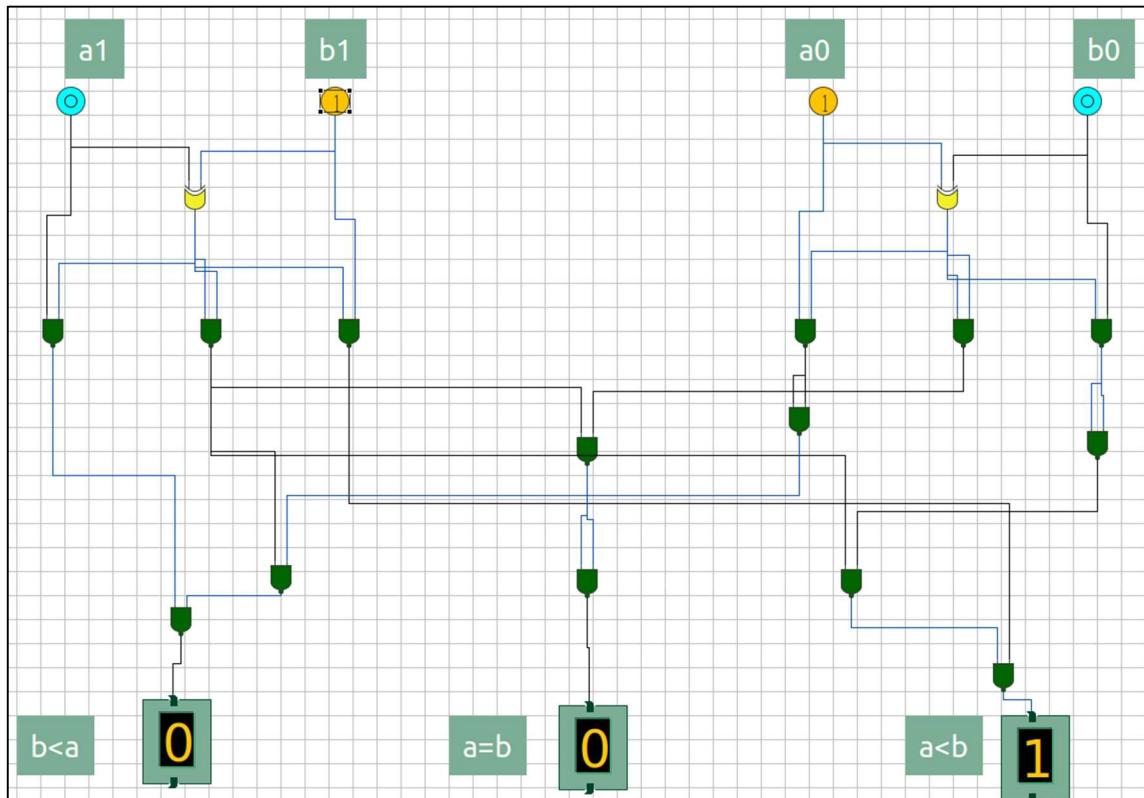


Figure 7:  $a_1 = 0$ ;  $a_0 = 1$ ;  $b_1 = 1$ ;  $b_0 = 0$

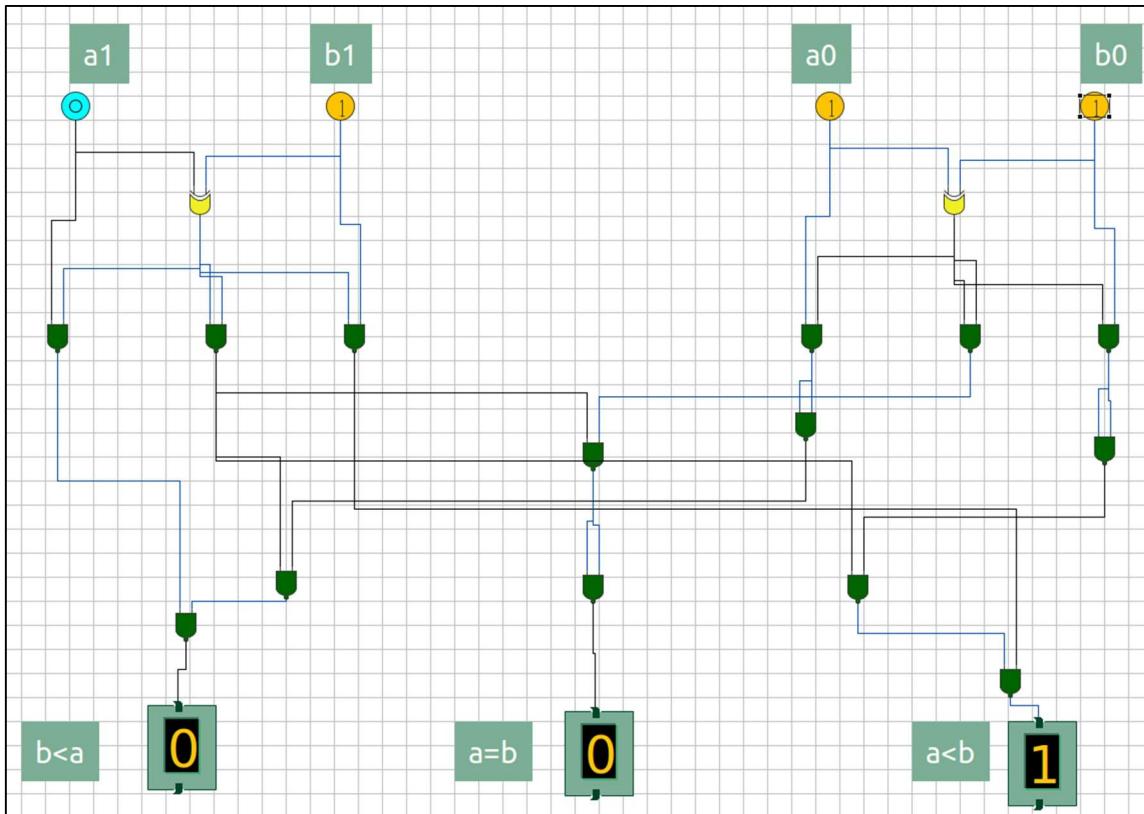


Figure 8:  $a_1 = 0$ ;  $a_0 = 1$ ;  $b_1 = 1$ ;  $b_0 = 1$

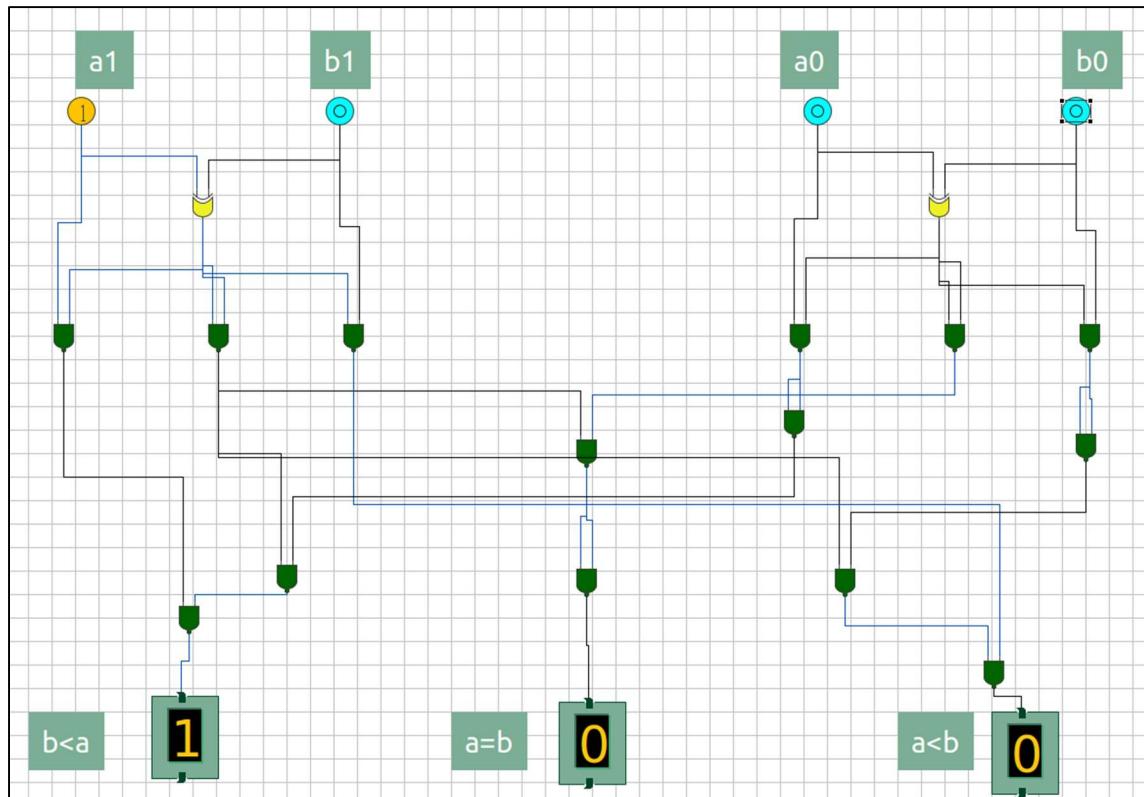


Figure 9:  $a_1 = 1$ ;  $a_0 = 0$ ;  $b_1 = 0$ ;  $b_0 = 0$

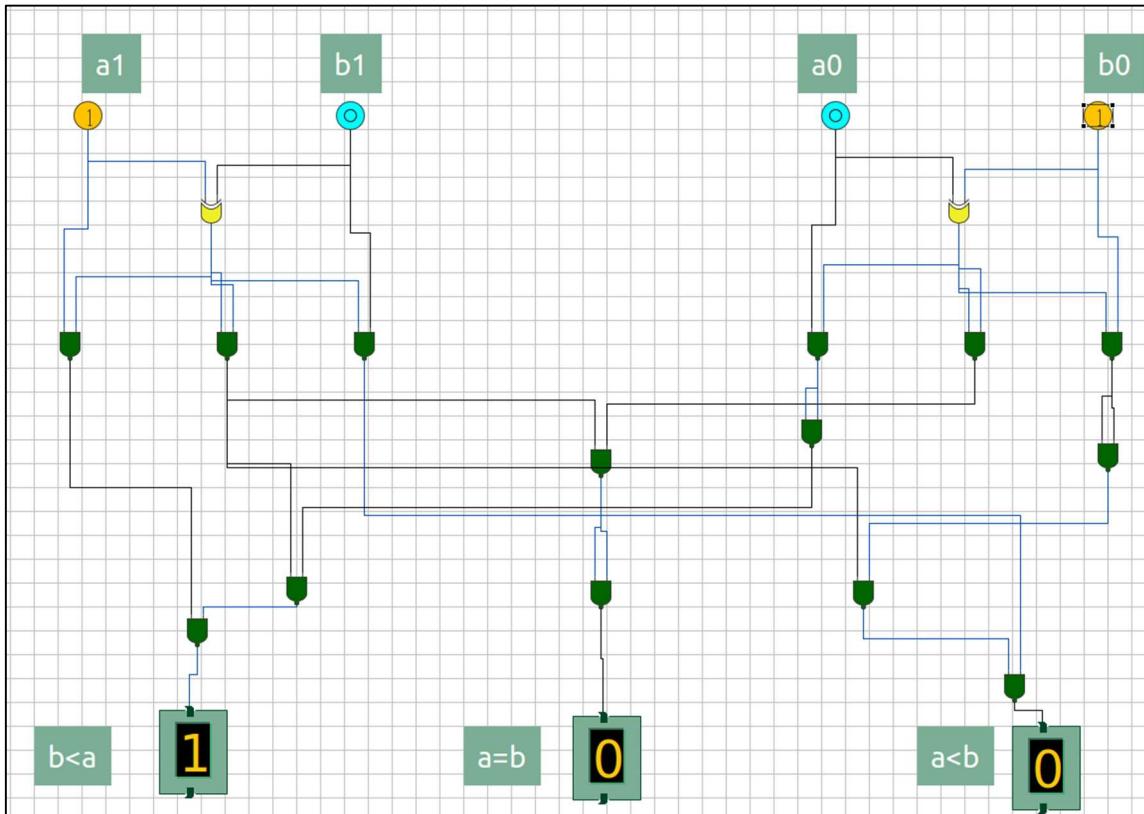


Figure 10:  $a_1 = 1$ ;  $a_0 = 0$ ;  $b_1 = 0$ ;  $b_0 = 1$

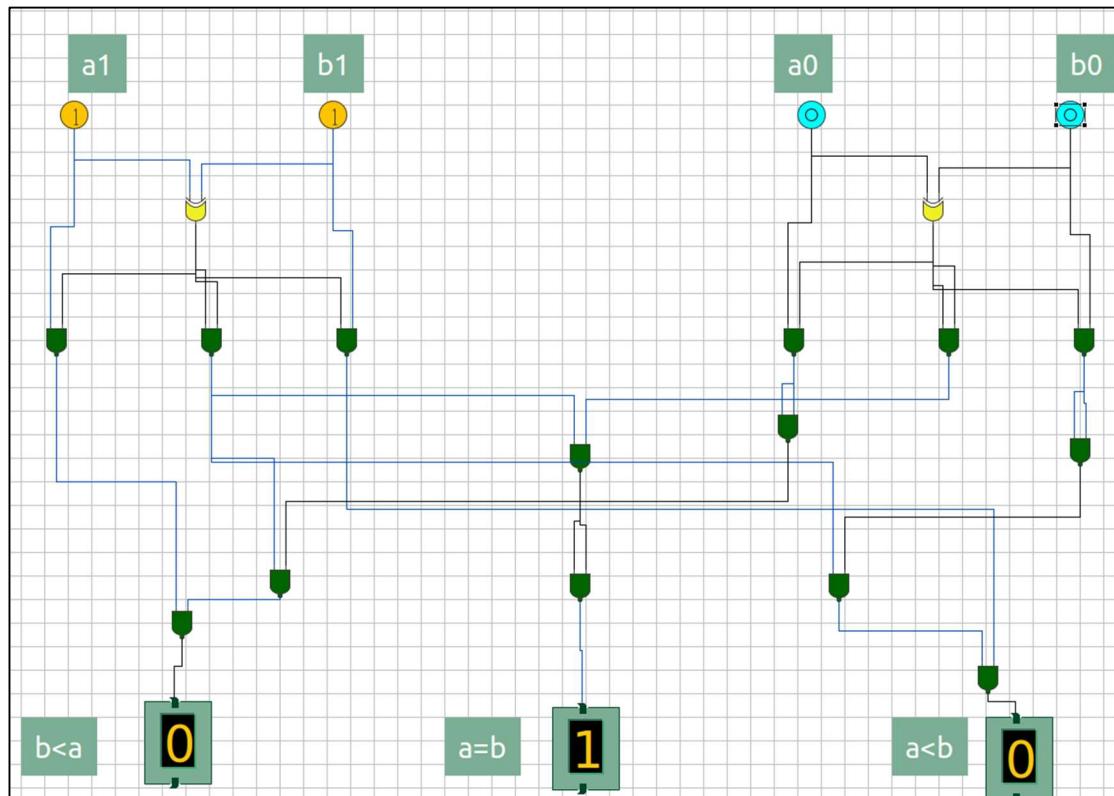


Figure 11:  $a_1 = 1$ ;  $a_0 = 0$ ;  $b_1 = 1$ ;  $b_0 = 0$

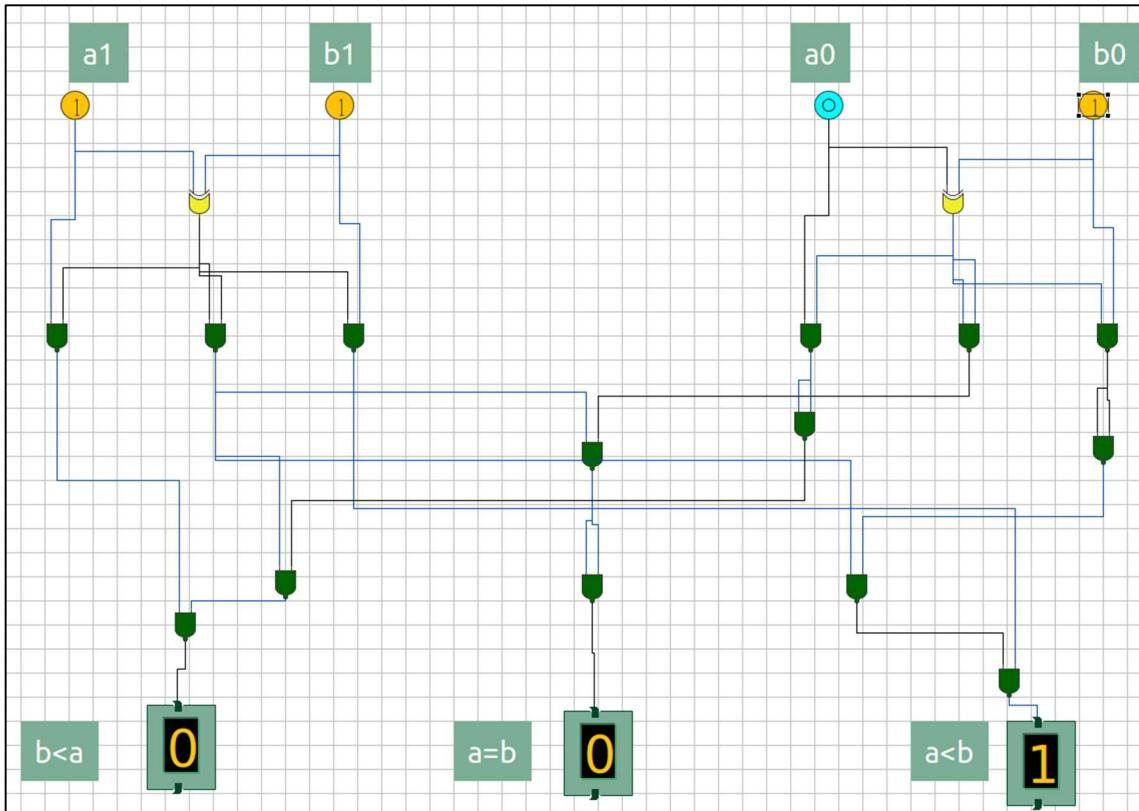


Figure 12:  $a_1 = 1$ ;  $a_0 = 0$ ;  $b_1 = 1$ ;  $b_0 = 1$

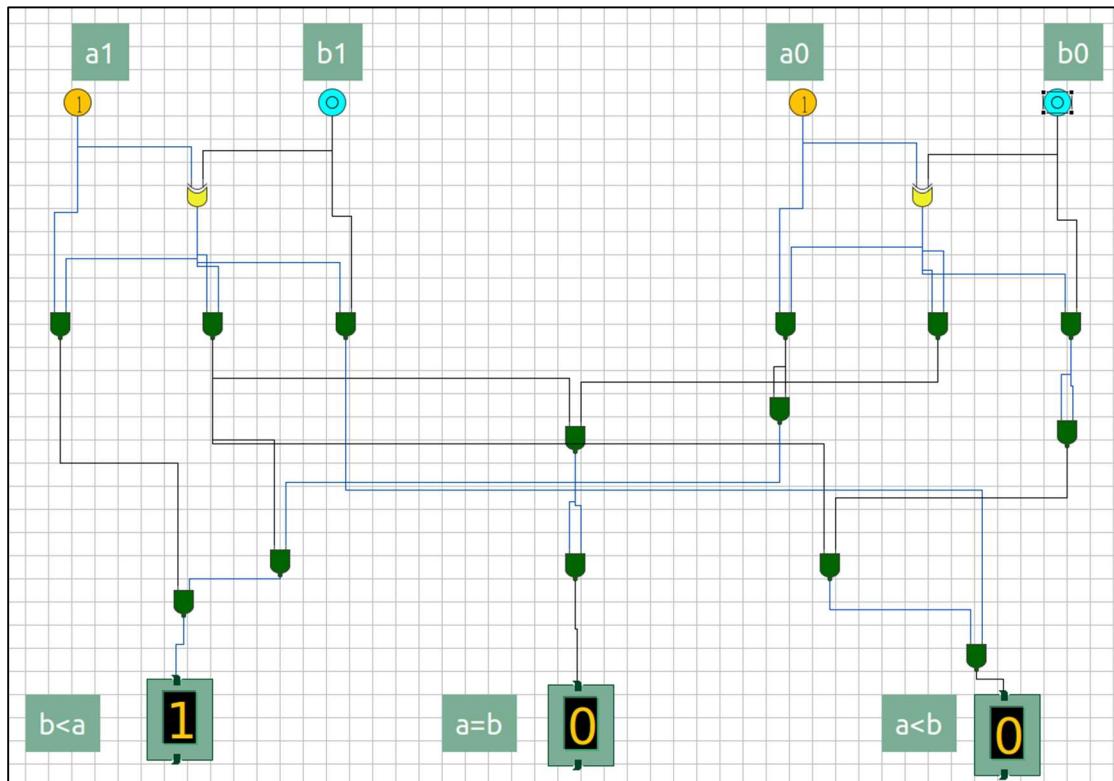


Figure 13:  $a_1 = 1$ ;  $a_0 = 1$ ;  $b_1 = 0$ ;  $b_0 = 0$

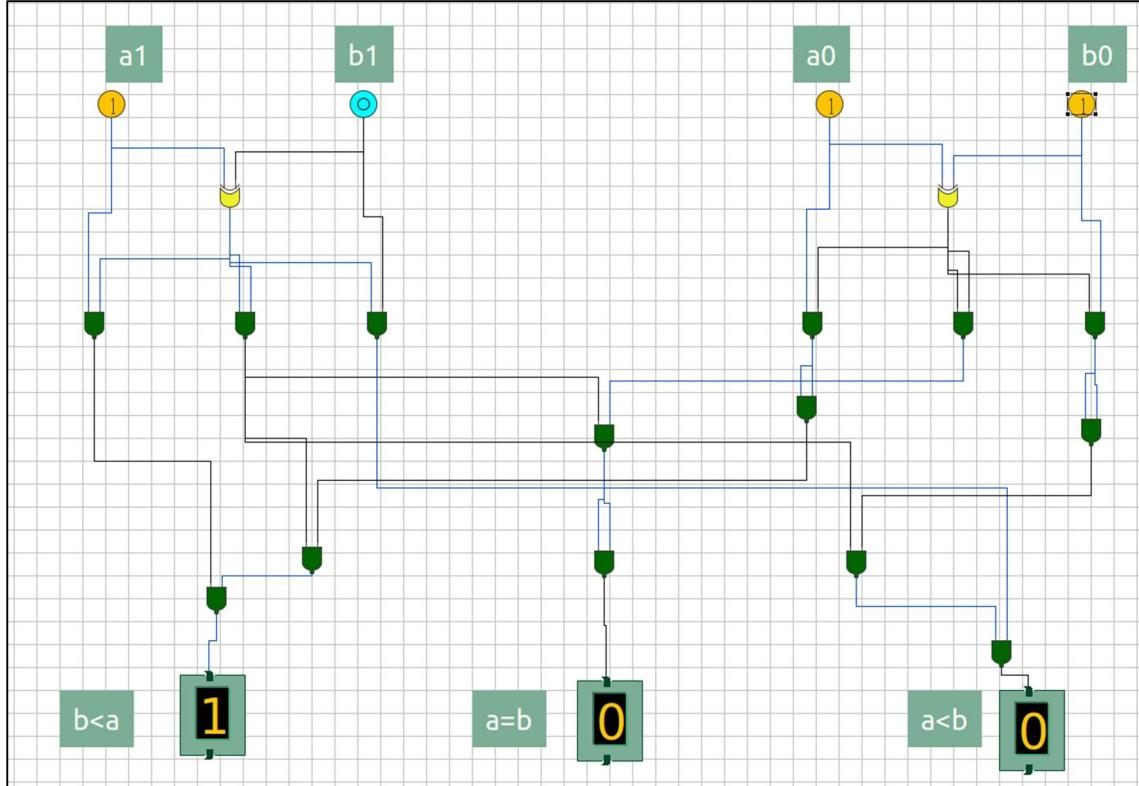


Figure 14:  $a1 = 1$ ;  $a0 = 1$ ;  $b1 = 0$ ;  $b0 = 1$

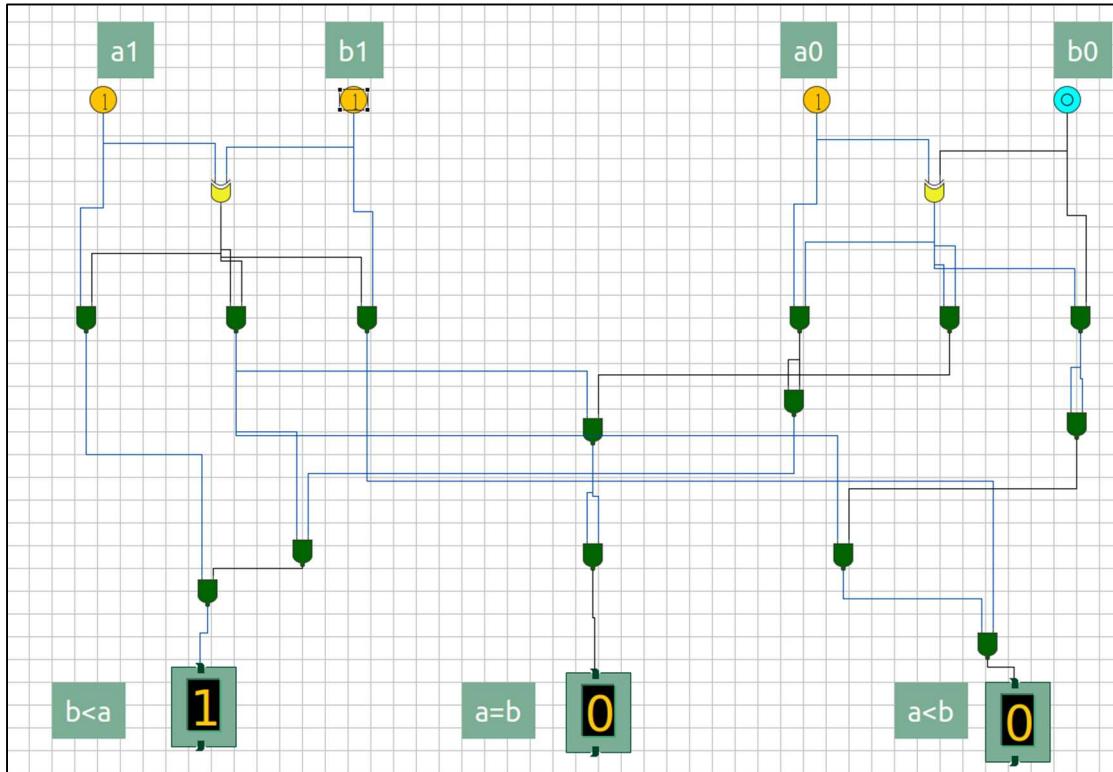


Figure 15:  $a1 = 1$ ;  $a0 = 1$ ;  $b1 = 1$ ;  $b0 = 0$

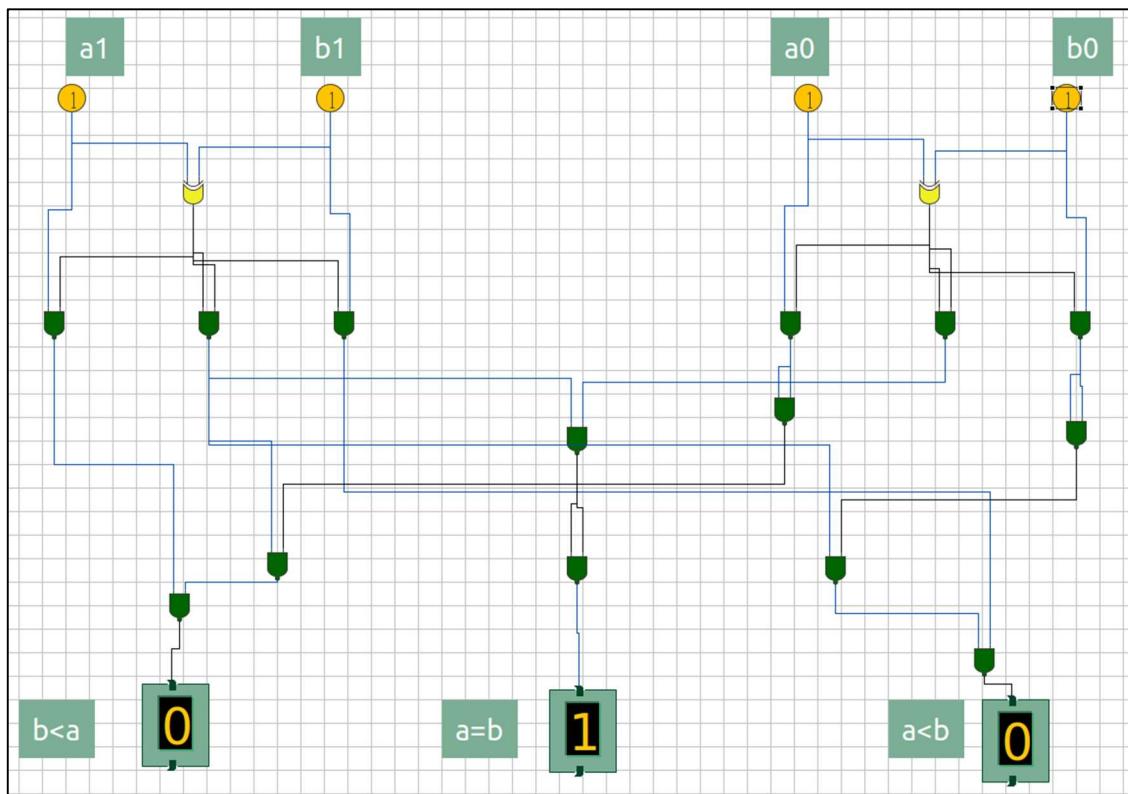


Figure 16:  $a_1 = 1; a_0 = 1; b_1 = 1; b_0 = 1$

## EVEN PARITY GENERATOR Cum CHECKER

### Objective

To implement 4 bit even parity generator cum checker circuit using minimum number of XOR gates.

### Truth Table

Let us assume 3 bit message is to be transmitted with a parity bit. Let the input message with bits A, B, C and parity bit P and output Z. The circuit behaves as an even parity generator when  $P=0$  & as even parity checker when  $P=1$ .

| Input |   |   | Output |   |
|-------|---|---|--------|---|
| A     | B | C | P      | Z |
| 0     | 0 | 0 | 0      | 0 |
| 0     | 0 | 0 | 1      | 1 |
| 0     | 0 | 1 | 0      | 1 |
| 0     | 0 | 1 | 1      | 0 |
| 0     | 1 | 0 | 0      | 1 |
| 0     | 1 | 0 | 1      | 0 |
| 0     | 1 | 1 | 0      | 0 |
| 0     | 1 | 1 | 1      | 1 |
| 1     | 0 | 0 | 0      | 1 |
| 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      | 0 |

From the truth table, expression for  $Z$  is

$$Z = \sum (1, 2, 4, 7, 8, 11, 13, 14)$$

The K-Map simplification of  $Z$  is

| AB \ CP | 00  | 01  | 11  | 10  |
|---------|-----|-----|-----|-----|
| 00      |     | (1) |     | (1) |
| 01      | (1) |     | (1) |     |
| 11      |     | (1) |     | (1) |
| 10      | (1) |     | (1) |     |

$$\therefore Z = \overline{A} \overline{B} \overline{C} P + \overline{A} \overline{B} C \overline{P} + \overline{A} B \overline{C} \overline{P} + A \overline{B} \overline{C} \overline{P}$$

$$+ \overline{A} B C P + A \overline{B} C P + A B \overline{C} P + A B C \overline{P}$$

$$= \overline{A} \overline{B} (CP + \overline{C}\overline{P}) + \overline{A} B (\overline{C}\overline{P} + CP) + AB(C\overline{P} + \overline{C}\overline{P})$$

$$+ A \overline{B} (C\overline{C}\overline{P} + CP)$$

$$\text{now, } CP + \overline{C}\overline{P} = C \oplus P$$

$$\text{and } \overline{C}\overline{P} + CP = ((\overline{C}\overline{P} + CP)')' ; \text{ Involution Law}$$

$$= [(\overline{C}\overline{P})' \cdot (CP)']' ; \text{ De-Morgan's Law}$$

$$= [(C+P)(\overline{C}+\overline{P})]' ; \text{ Demorgan's Law}$$

$$= [C\overline{P} + P\overline{C}]'$$

$$= \overline{C \oplus P}$$

Therefore:

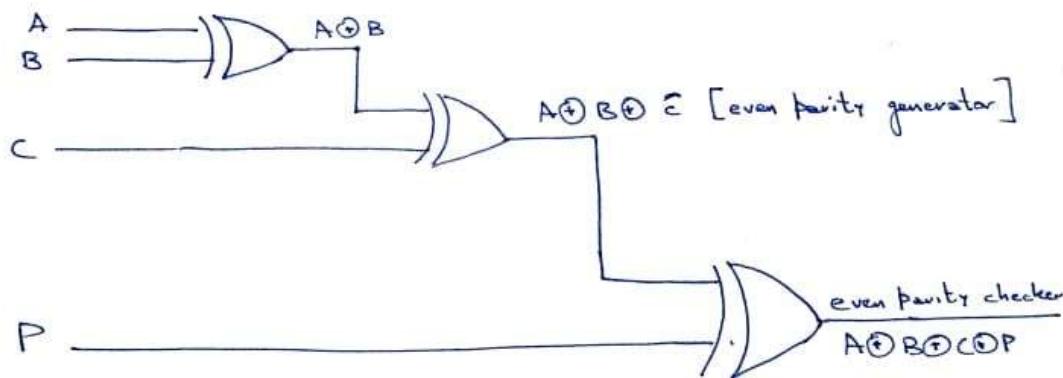
$$\begin{aligned} Z &= (\overline{A}\bar{B} + A\bar{B}) (C \oplus P) + (\bar{A}\overline{\bar{B}} + \bar{A}B) (\overline{C \oplus P}) \\ &= \cancel{(\overline{A}\bar{B})} (\overline{C \oplus P}) + (A \oplus B) (\overline{C \oplus P}) \\ &= (C \oplus P) \oplus (A \oplus B) \\ Z &= A \oplus B \oplus C \oplus P \end{aligned}$$

→ when  $P=0$ :

$$Z = A \oplus B \oplus C \oplus 0 = A \oplus B \oplus C$$

which behaves like an even parity checker, generator.  
→ when  $P=1$ ;  $Z = A \oplus B \oplus C \oplus P$ , behaves as even parity checker.

Circuit Diagram



## Results

As we can see from the simulation that the minimum no. of XOR gate required to implement an even parity generator, even checker, is 3

Moreover the Simulation Results match with the expected output of the truth table.

## ODD PARITY GENERATOR Cum CHECKER

### Objective

To implement 4 bit odd parity generator cum checker circuit using minimum no. of XNOR gates

### Truth Table

$A, B, C \rightarrow$  input messages in 3 bit

$Z \rightarrow$  output

Circuit behaves as an odd parity generator when  $P=0$ ,  
 & checker when  $P=1$

| Input |   |   |   | Output |
|-------|---|---|---|--------|
| A     | B | C | P | Z      |
| 0     | 0 | 0 | 0 | 1      |
| 0     | 0 | 0 | 1 | 0      |
| 0     | 0 | 1 | 0 | 0      |
| 0     | 0 | 1 | 1 | 1      |
| 0     | 1 | 0 | 0 | 0      |
| 0     | 1 | 0 | 1 | 1      |
| 0     | 1 | 1 | 0 | 1      |
| 0     | 1 | 1 | 1 | 0      |
| 1     | 0 | 0 | 1 | 1      |
| 1     | 0 | 1 | 0 | 1      |
| 1     | 0 | 1 | 1 | 0      |
| 1     | 1 | 0 | 0 | 1      |
| 1     | 1 | 0 | 1 | 0      |
| 1     | 1 | 1 | 0 | 0      |
| 1     | 1 | 1 | 1 | 1      |

K-Map simplification for  $Z = \Sigma(0, 3, 5, 6, 9, 10, 12, 15)$

| $AB \backslash CP$ | 00 | 01 | 11 | 10 |
|--------------------|----|----|----|----|
| 00                 | 1  |    | 1  | 1  |
| 01                 |    | 1  |    | 1  |
| 11                 | 1  |    | 1  |    |
| 10                 |    | 1  |    | 1  |

$$Z = \overline{A} \overline{B} \overline{C} \overline{P} + \overline{A} \overline{B} C P + \overline{A} B \overline{C} P + \overline{A} B C \overline{P} +$$

$$+ A \overline{B} \overline{C} P + A \overline{B} C \overline{P} + A B \overline{C} \overline{P} + A B C P$$

$$= (\overline{B} + \overline{C} + BC)(\overline{A} \overline{P} + AP) + (B \overline{C} + \overline{B} C)(A \overline{B} \overline{P} + \overline{A} P)$$

$$\text{now we know: } \overline{B} \overline{C} + BC = B \odot C$$

$$\overline{A} \overline{P} + AP = A \odot P$$

$$\text{and } \overline{B} C + B \overline{C} = [(\overline{B} C + B \overline{C})']' ; \text{ involution law}$$

$$= [(\overline{B} C)' (B \overline{C})']' ; \text{ de-morgan's Law}$$

$$= [(\overline{B} + \overline{C})(\overline{B} + C)]'$$

$$= [B C + \overline{B} \overline{C}]'$$

$$= \overline{B \odot C}$$

$$\text{and similarly } \overline{A} \overline{P} + \overline{A} P = \overline{A \odot P}$$

Therefore

$$Z = (\overline{B} \overline{C} + BC)(\overline{A} \overline{P} + AP) + (\overline{B} C + B \overline{C})(A \overline{P} + \overline{A} P)$$

$$= (B \odot C) \odot (A \odot P)$$

$$Z = A \odot B \odot C \odot P$$

→ when  $P=0$

$$Z = A \oplus B \oplus C \oplus 0$$

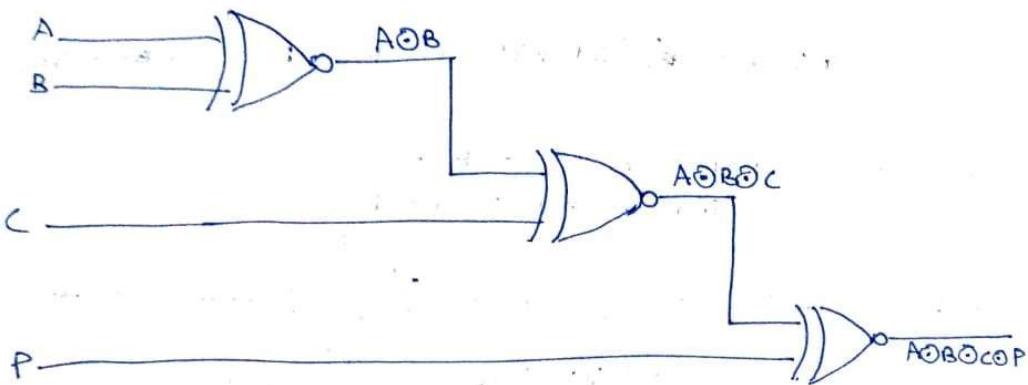
=  $\overline{A \oplus B \oplus C}$ ; i.e. odd parity generator

→ when  $P=1$

$$Z = A \oplus B \oplus C \oplus 1$$

=  $A \oplus B \oplus C$ ; i.e. odd parity checker

Circuit Diagram



Result

As we can see from the simulation, the minimum no. of XNOR gates required to implement an odd parity generator cum checker is 3.

Moreover the simulation results matches with the expected outputs of the truth table.

## Input / Output

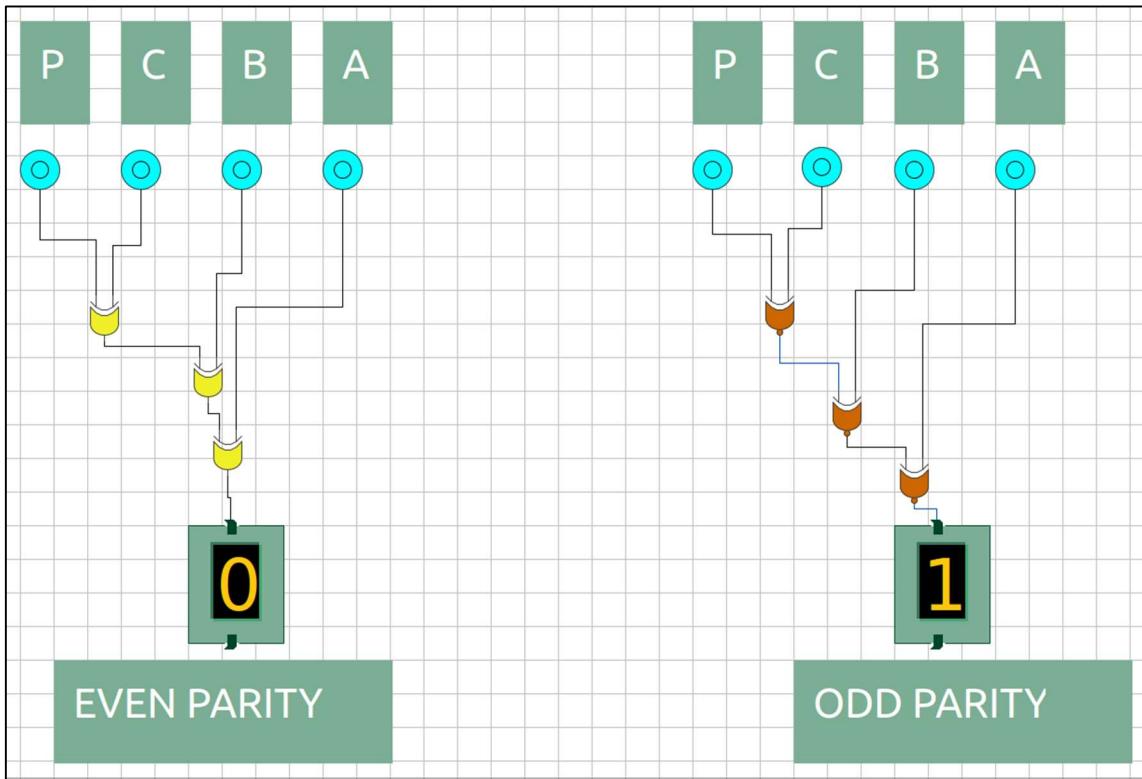


Figure 1:  $P = 0; C = 0; B = 0; A = 0$

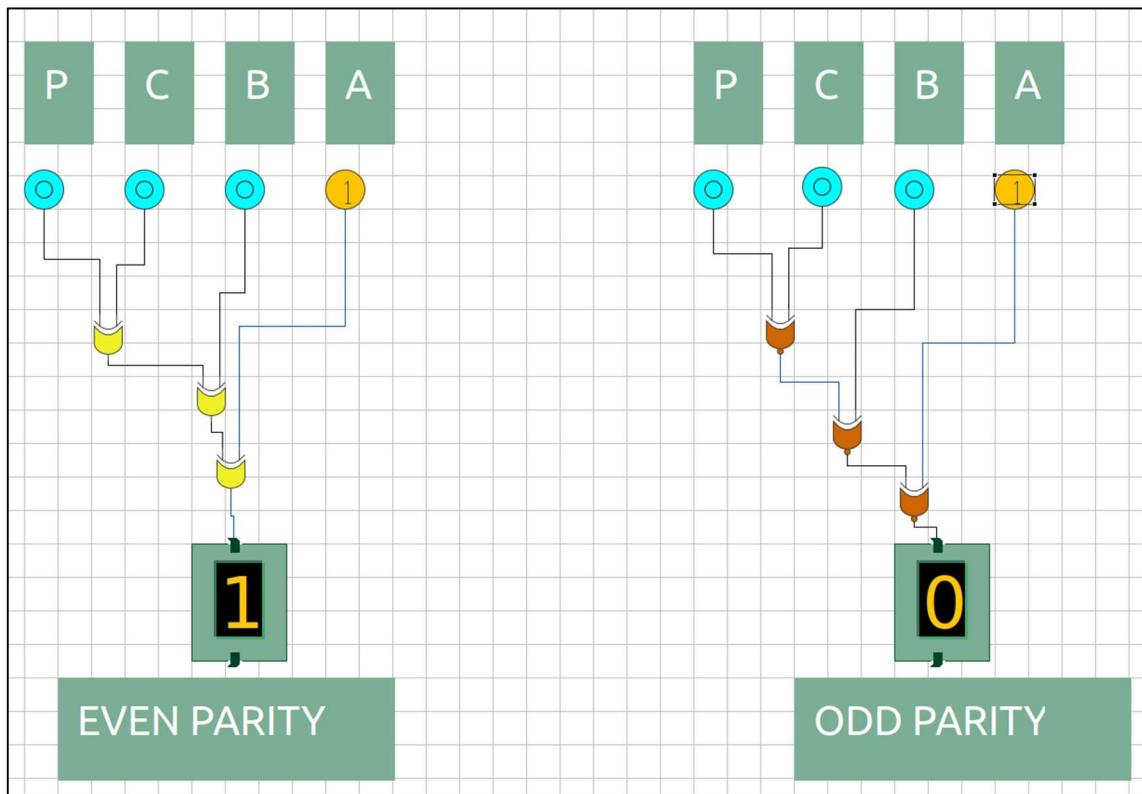


Figure 2:  $P = 0; C = 0; B = 0; A = 1$

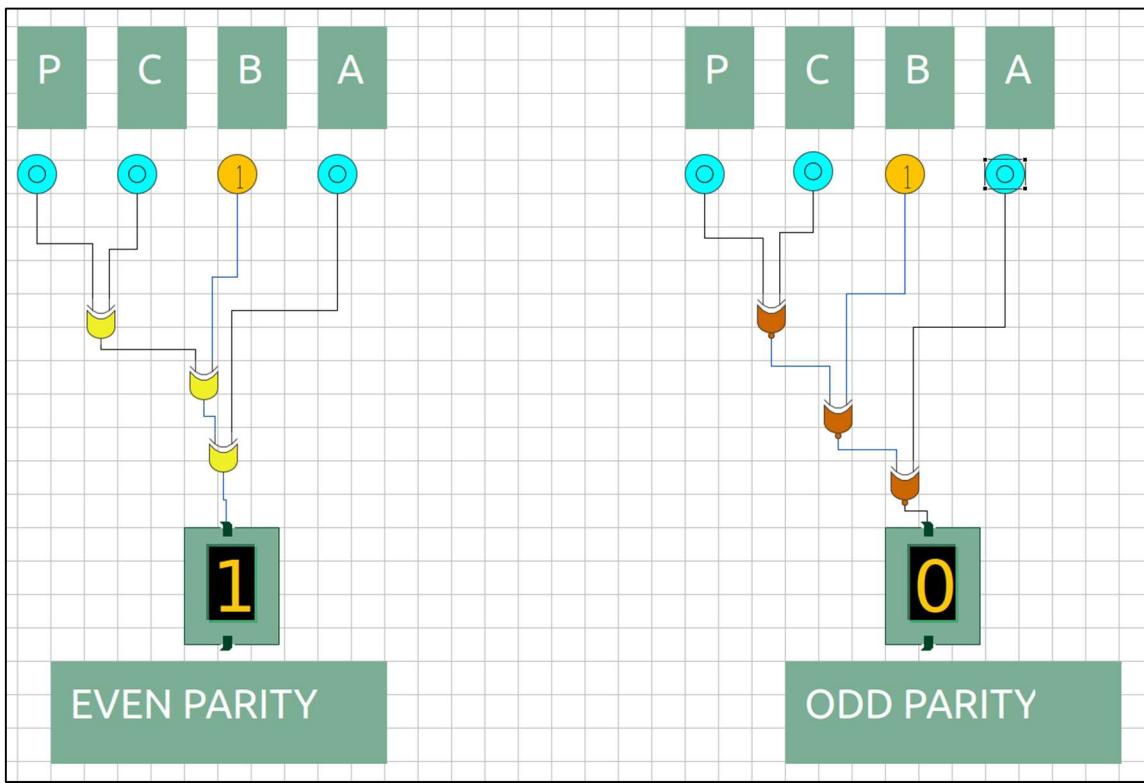


Figure 3:  $P = 0; C = 0; B = 1; A = 0$

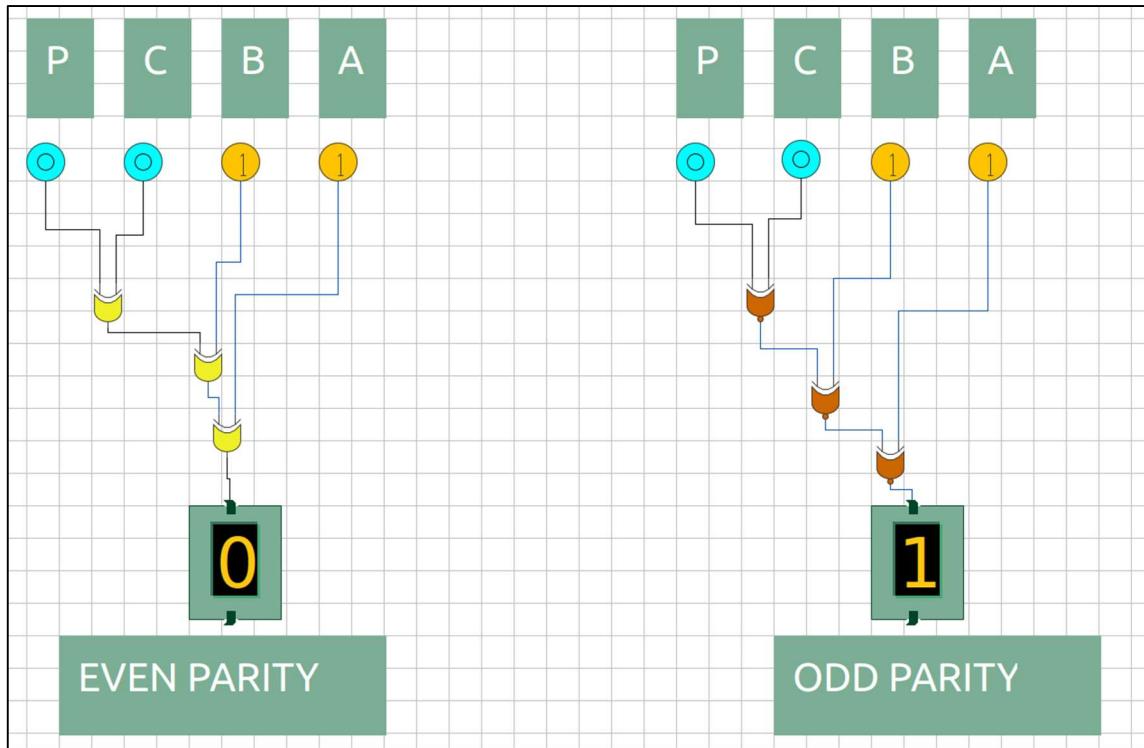


Figure 4:  $P = 0; C = 0; B = 1; A = 1$

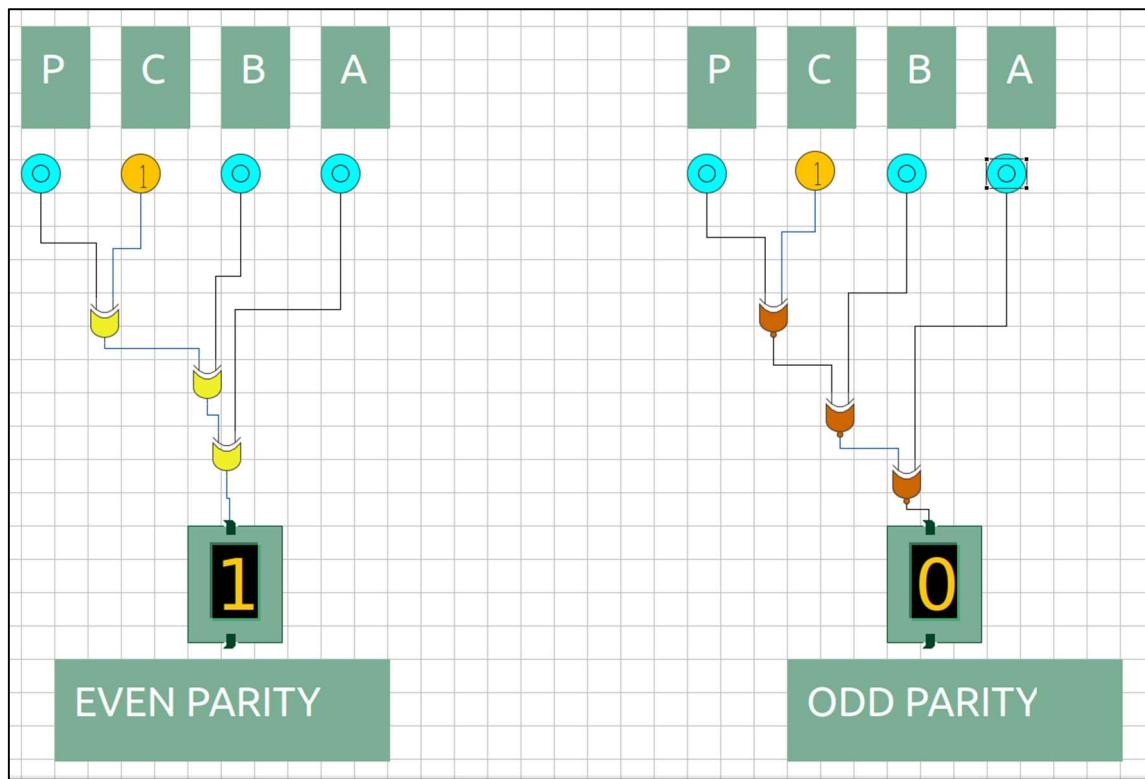


Figure 5:  $P = 0; C = 1; B = 0; A = 0$

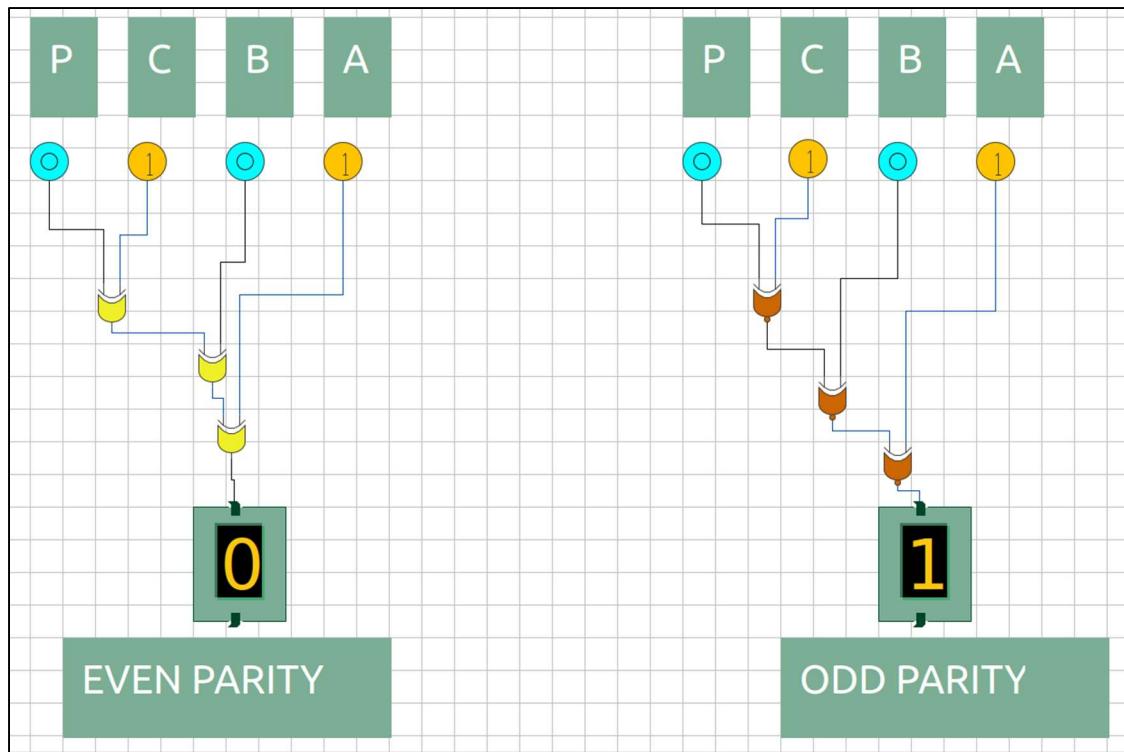


Figure 6:  $P = 0; C = 1; B = 0; A = 1$

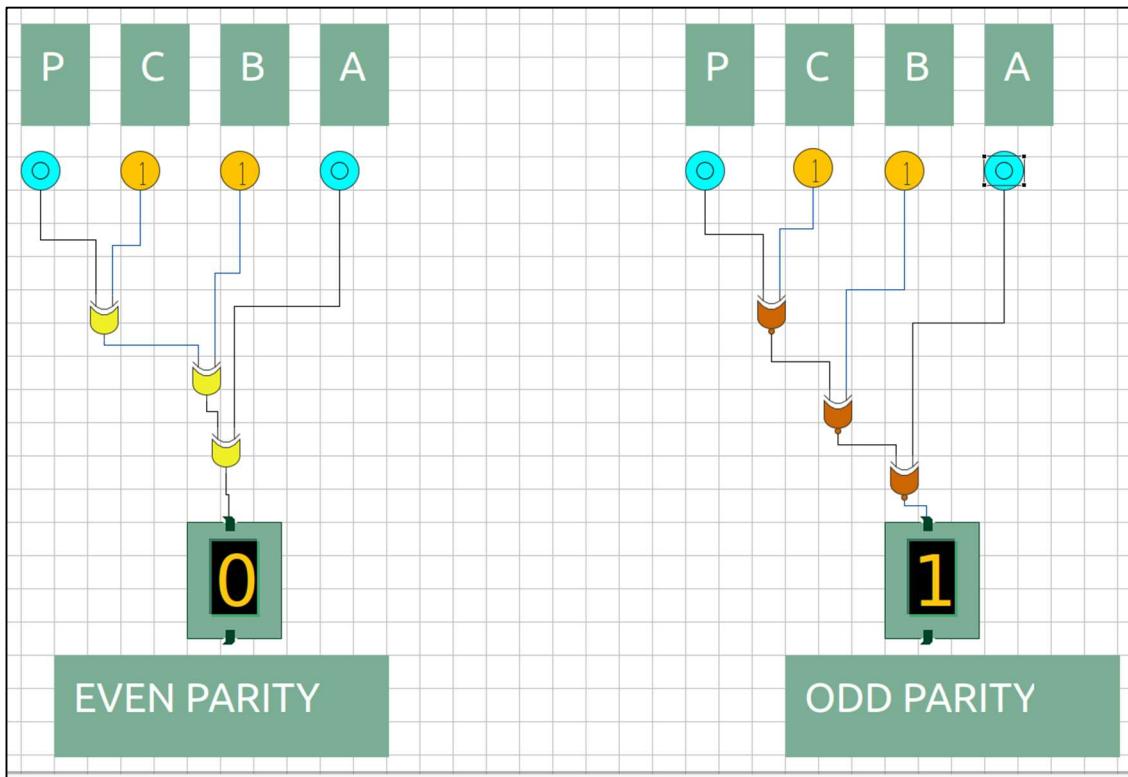


Figure 7:  $P = 0; C = 1; B = 1; A = 0$

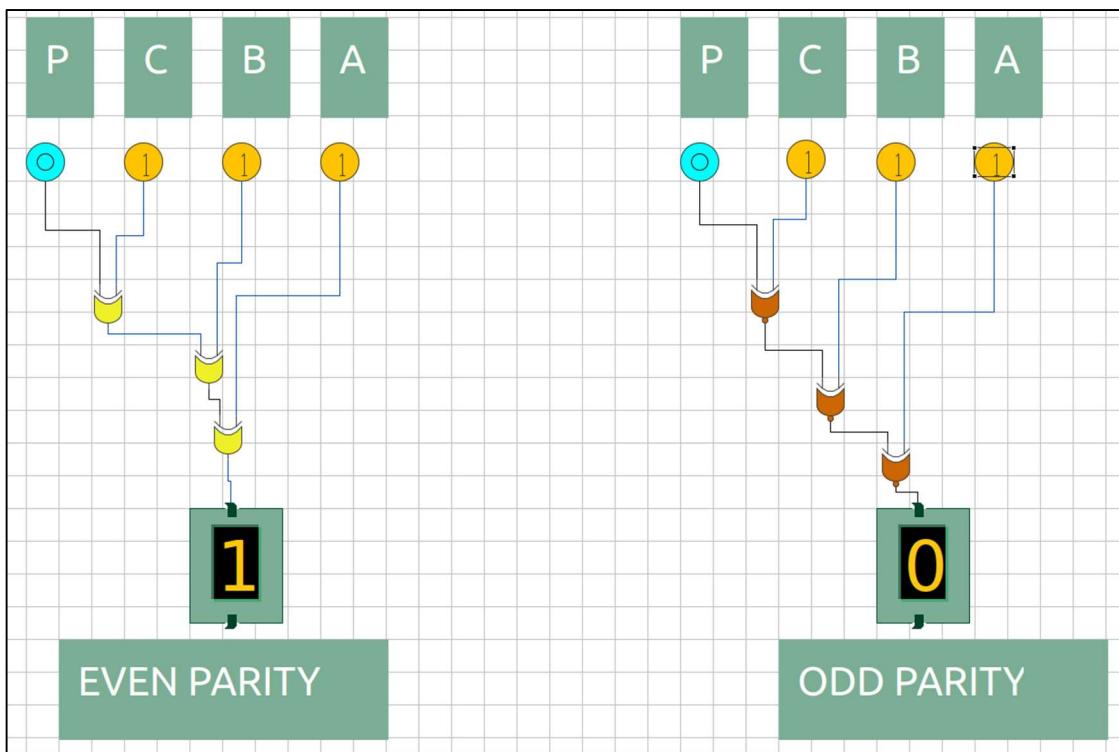


Figure 8:  $P = 0; C = 1; B = 1; A = 1$

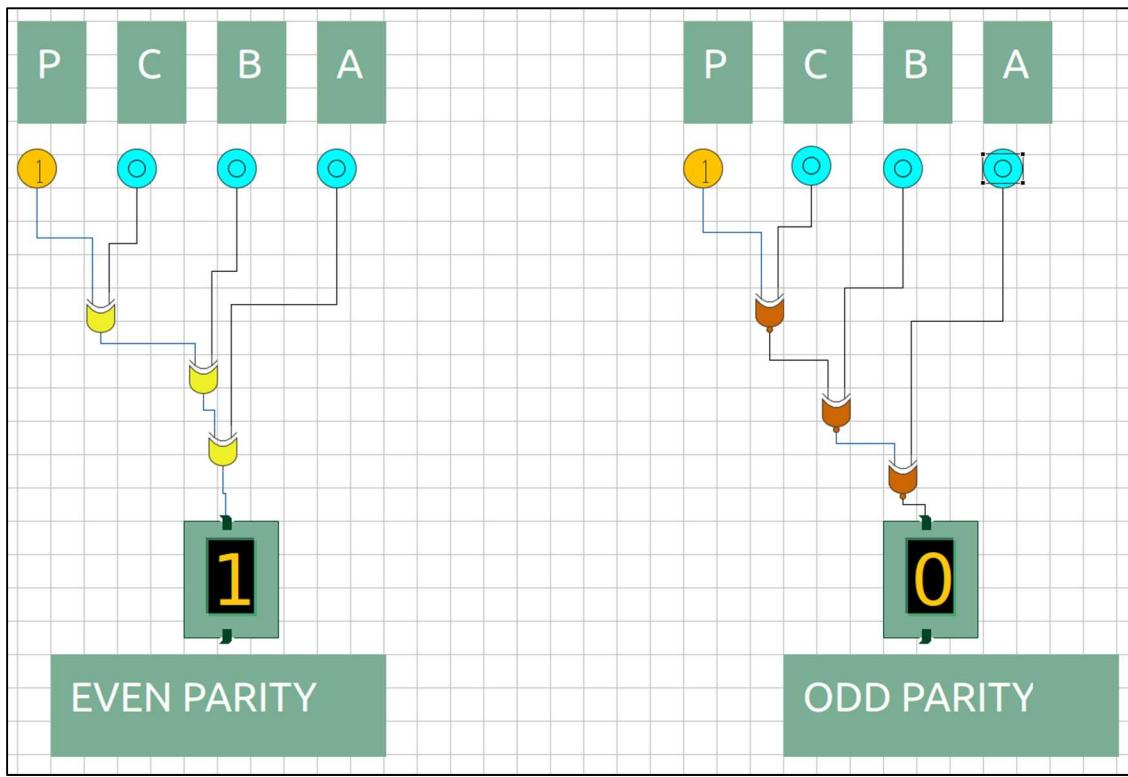


Figure 9:  $P = 1; C = 0; B = 0; A = 0$

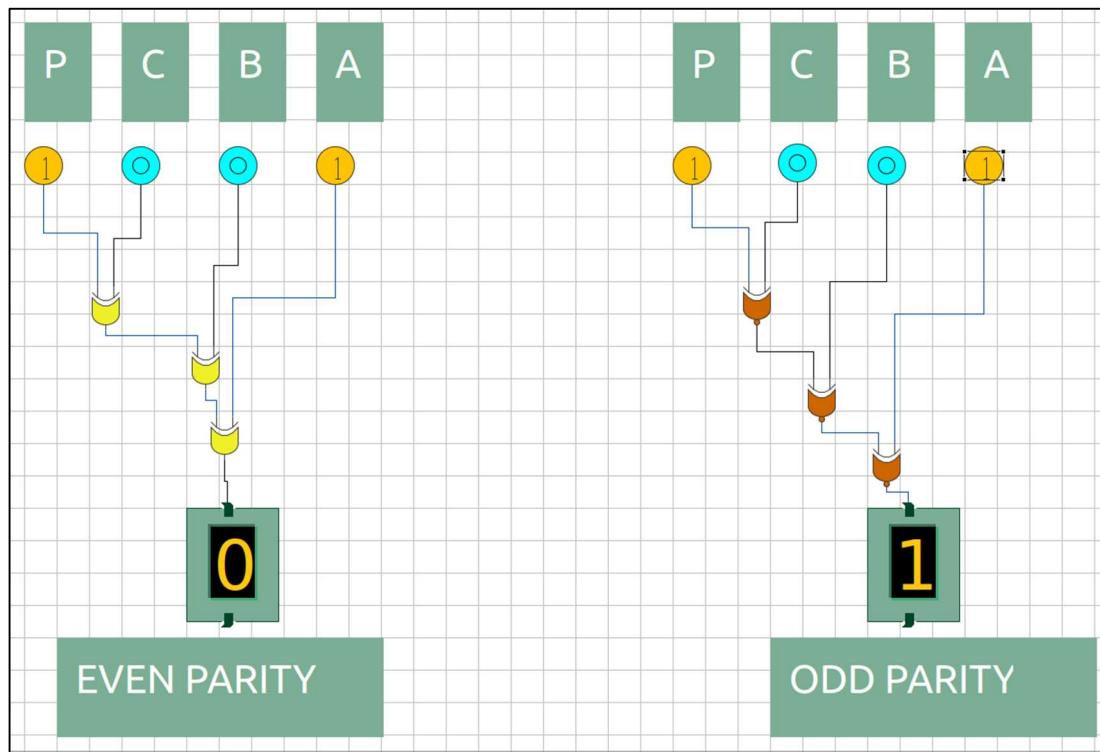


Figure 10:  $P = 1; C = 0; B = 0; A = 1$

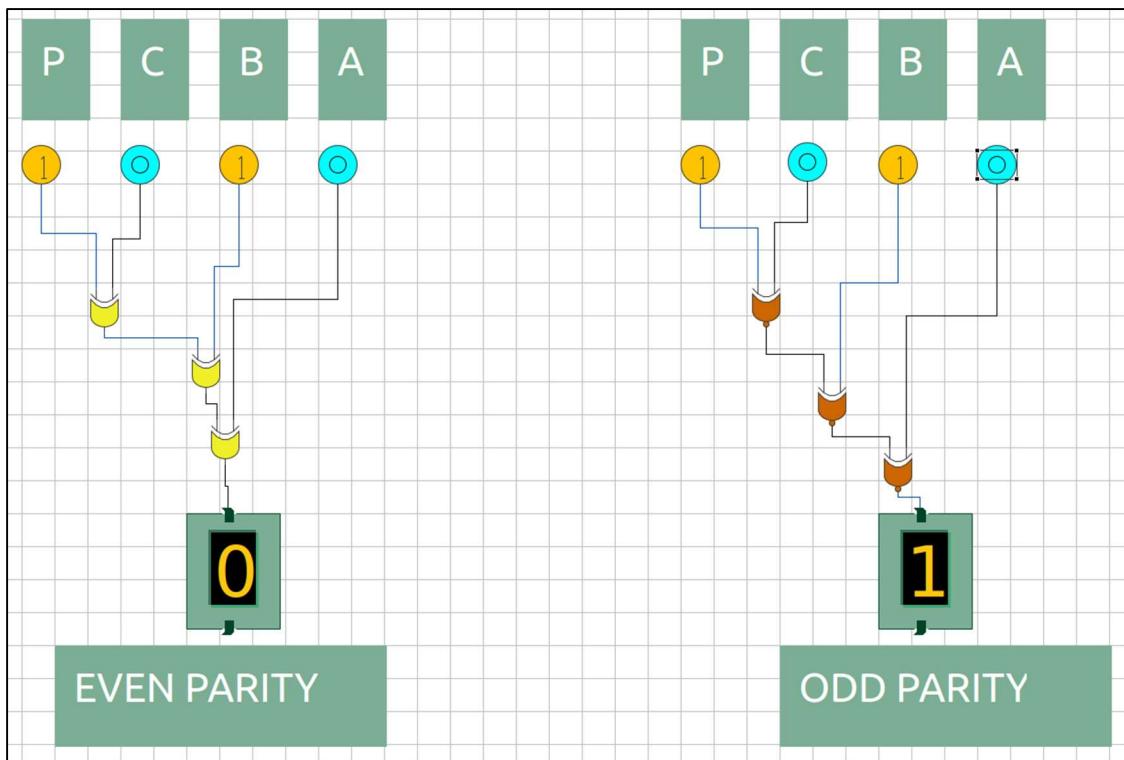


Figure 11: P = 1; C = 0; B = 1; A = 0

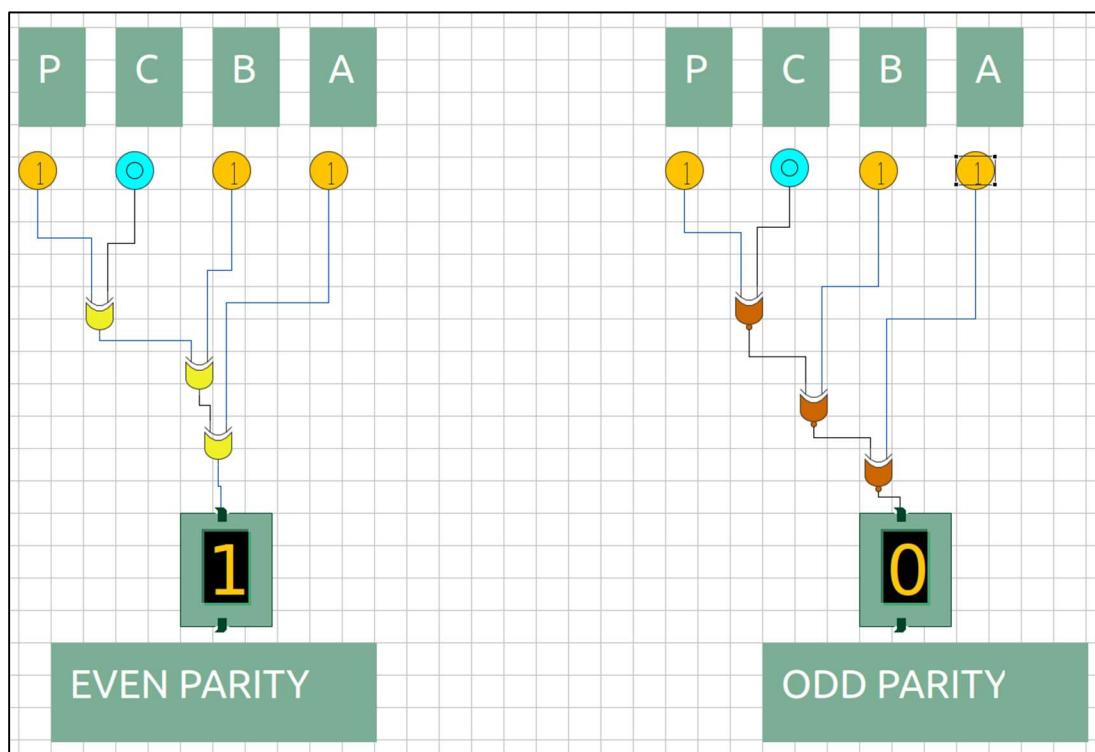


Figure 12: P = 1; C = 0; B = 1; A = 1

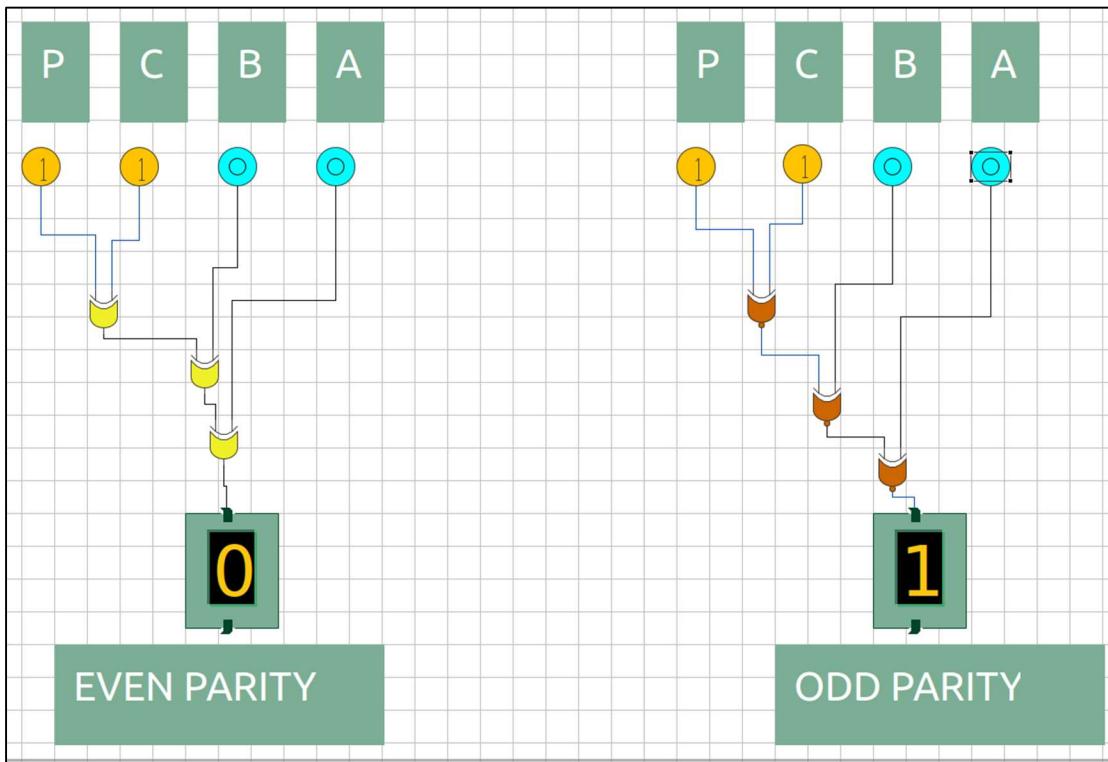


Figure 13:  $P = 1; C = 1; B = 0; A = 0$

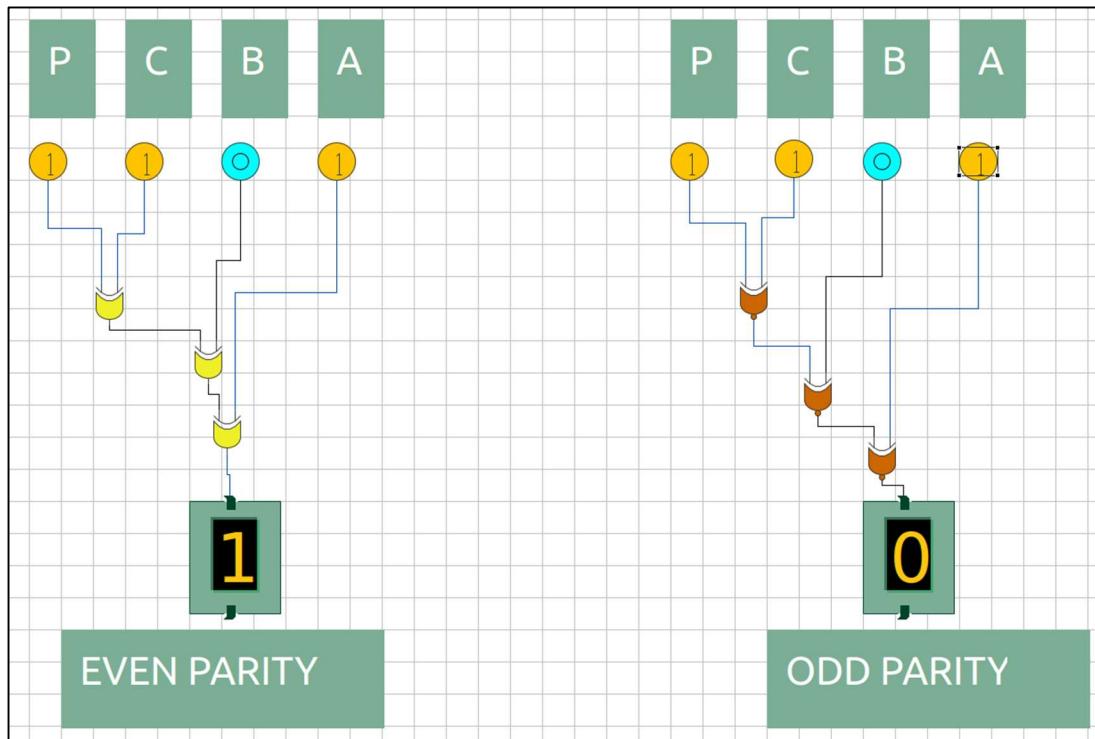


Figure 14:  $P = 1; C = 1; B = 0; A = 1$

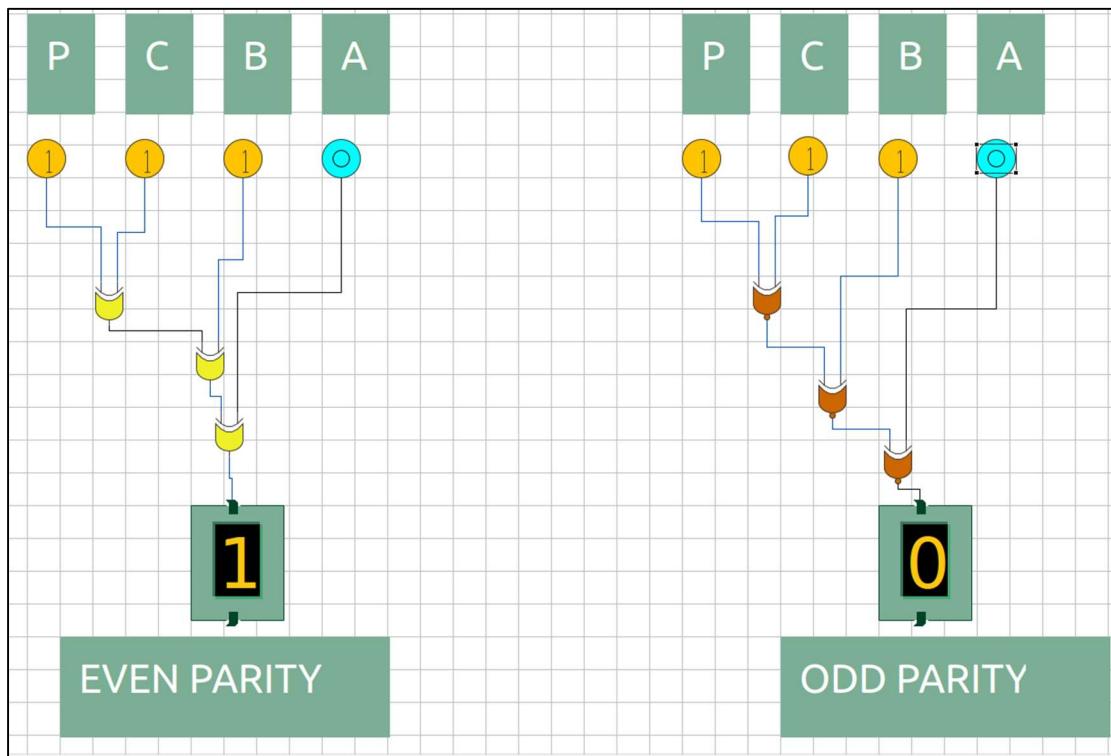


Figure 15:  $P = 1; C = 1; B = 1; A = 0$

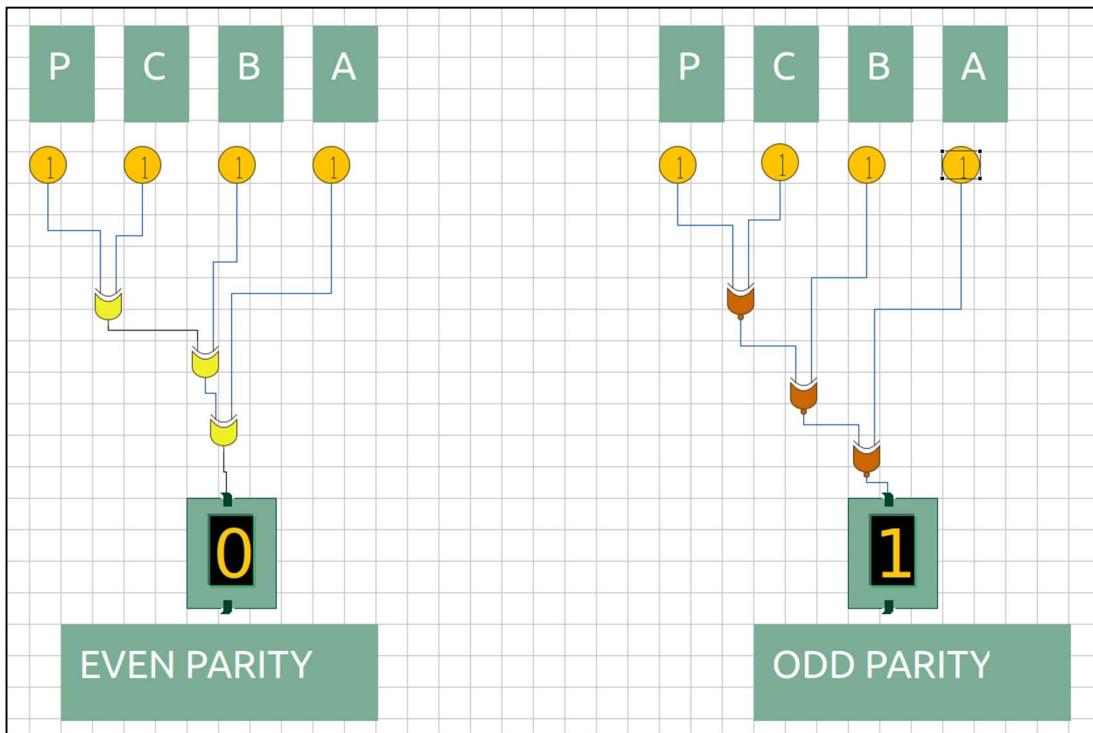


Figure 16:  $P = 1; C = 1; B = 1; A = 1$