Eshetu Wekjira

Lab 07- Adder/Subtractor

Task 1: 1-bit Full Adder

B) **Truth Table**

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

C) Sum of products

Sum = **A'B'Cin + A'BCin' + AB'Cin' +ABCin**

Cout = A**'BCin +AB'Cin +ABCin' +ABCin**

To find the sum of the product just by the sum of all the values of output 1. In this case, the output is (sum and Cout).

D) K-Map for Sum

BCin / A	00	01	11	10
0	0	1	0	1
1	1	0	1	0

We can't minimize the sum values of the k-map

Sum = **A'B'Cin + A'BCin' + AB'Cin' +ABCin**

K- Map for Cout

A \ BCin	00	01	11	10
0	0	0	1	0
1	0	1	1	1

Cout = **AB + BCin+ ACin**

E, F) Implement a Full adder

After I found the truth tables for the Sum and Cout, I tried to use different gates to see the result that much with sum or Cout, and after trying and error; I got the three circuit gates that gave the correct output that much my truth tables for Sum and Cout.

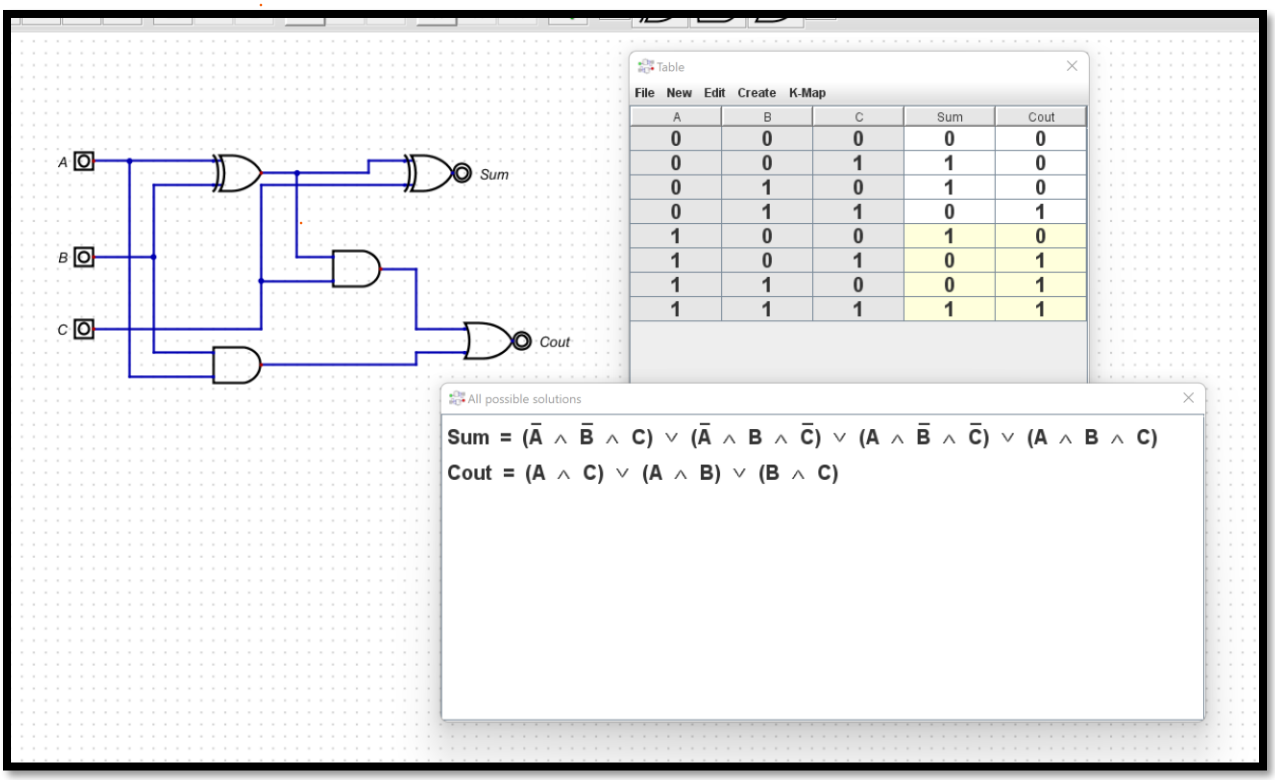

***Figure 1*: The digital schematic for the adder**

The test result for the **Sum** schematic

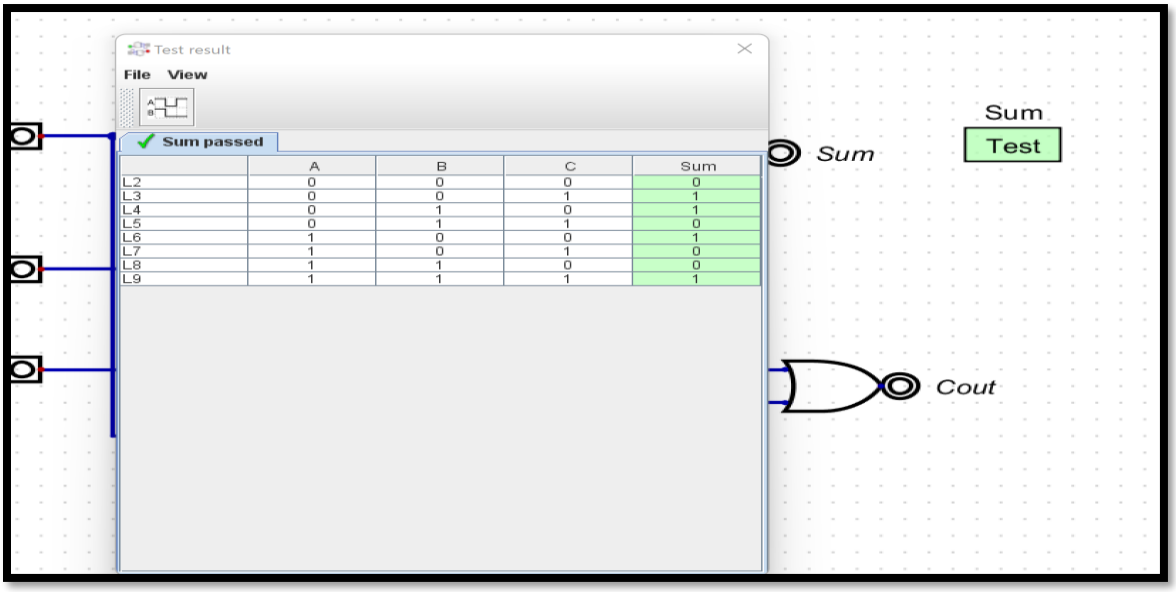

Figure 2: test result of the sum schematic

The test result for the **Cout** schematic

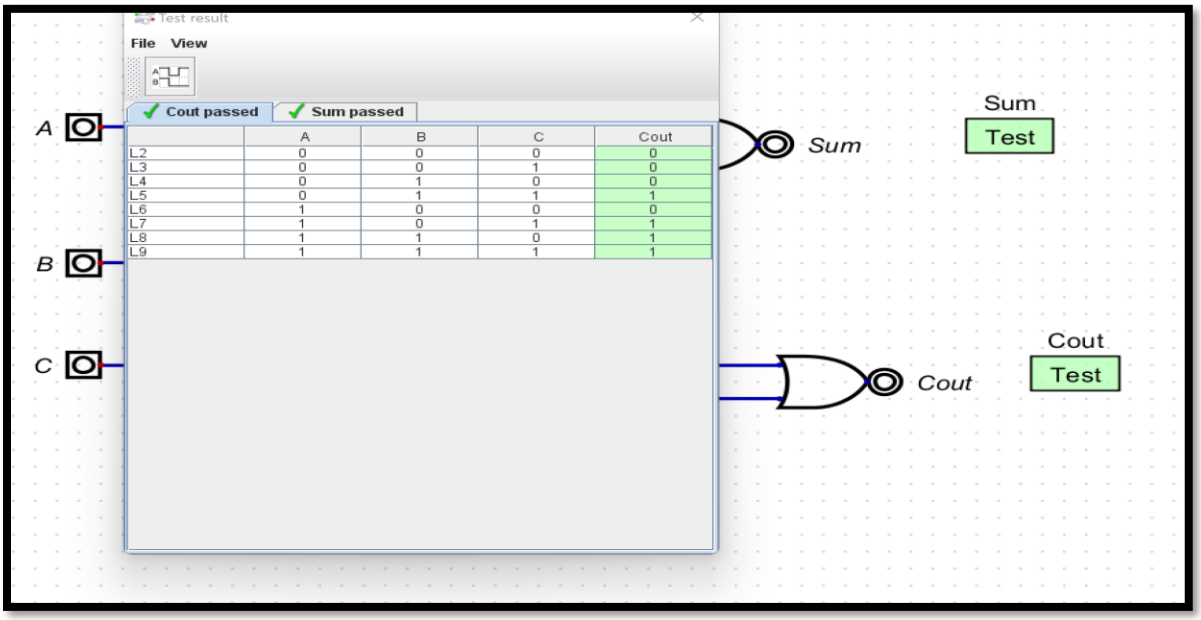

Figure 3: test result of the Cout schematic

Task 2: Adder/Subtractor

In this case, we have three inputs and two outputs, such as the sum and Cout; in this case, the Cout is the previous borrow value and is not a carryout. We must borrow values to subtract one or more from zeros in a subtractor circuit.

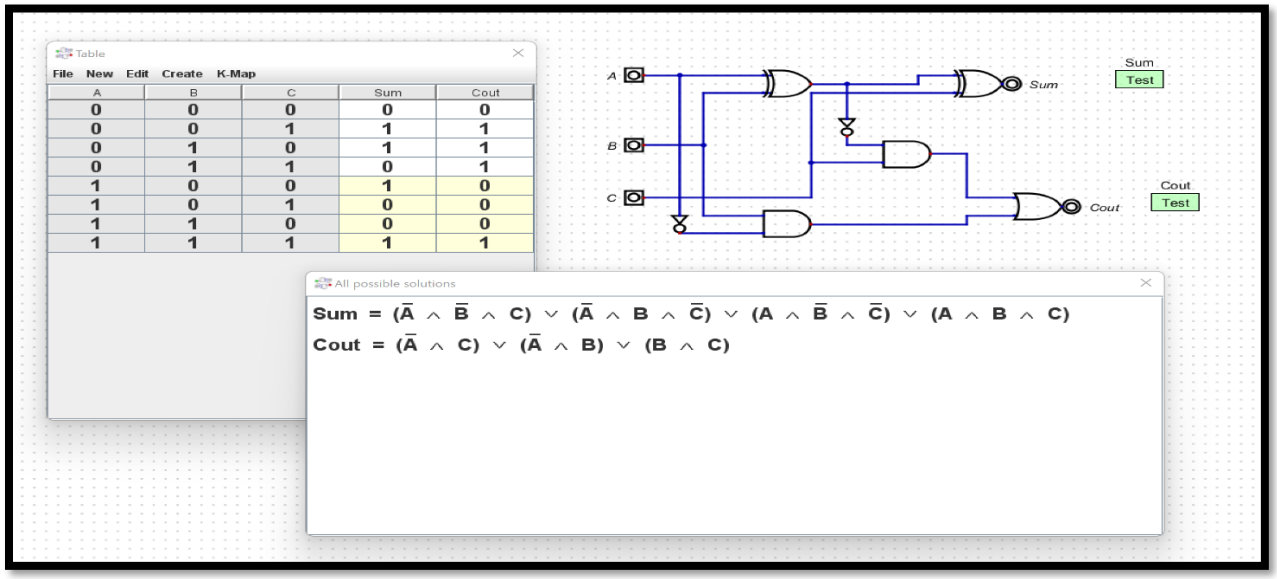

Figure 4: **The digital schematic for the adder/subtractor**

The test results for the **Cout** schematic; in this case, Cout is the previous borrow values.

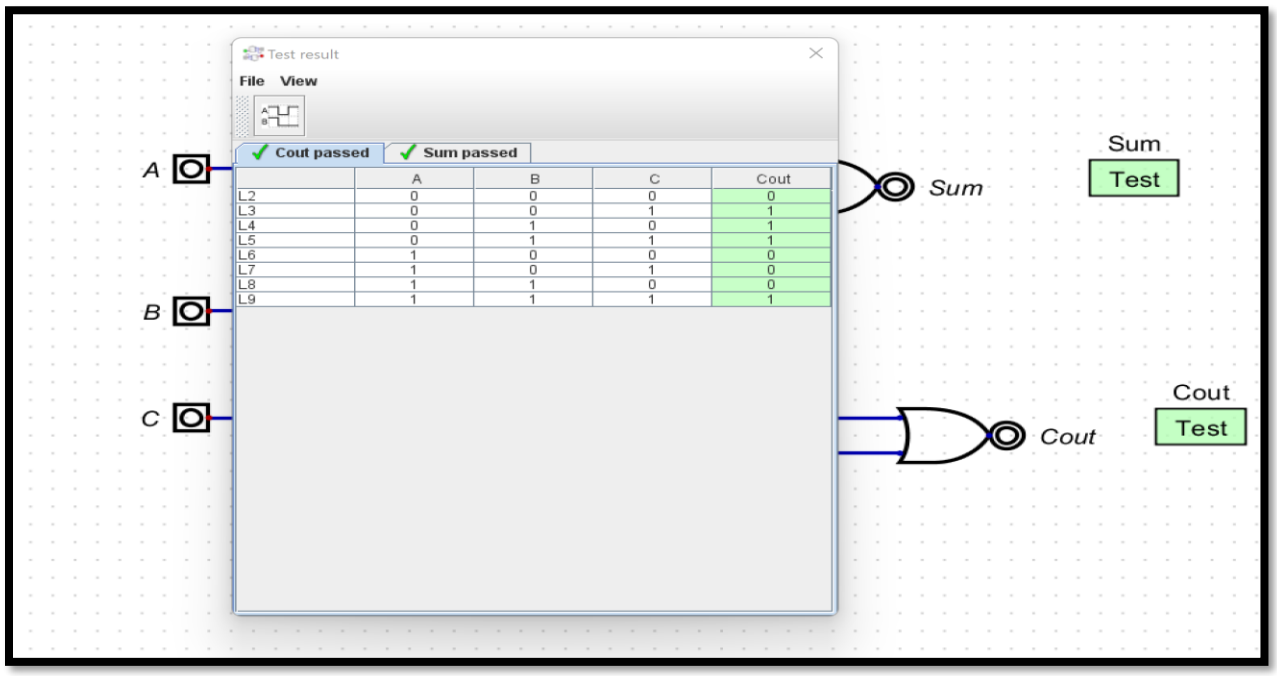

Figure 5: test result of the Subtractor schematic