Lab Assignment-6

Name – Md Junaid Mahmood

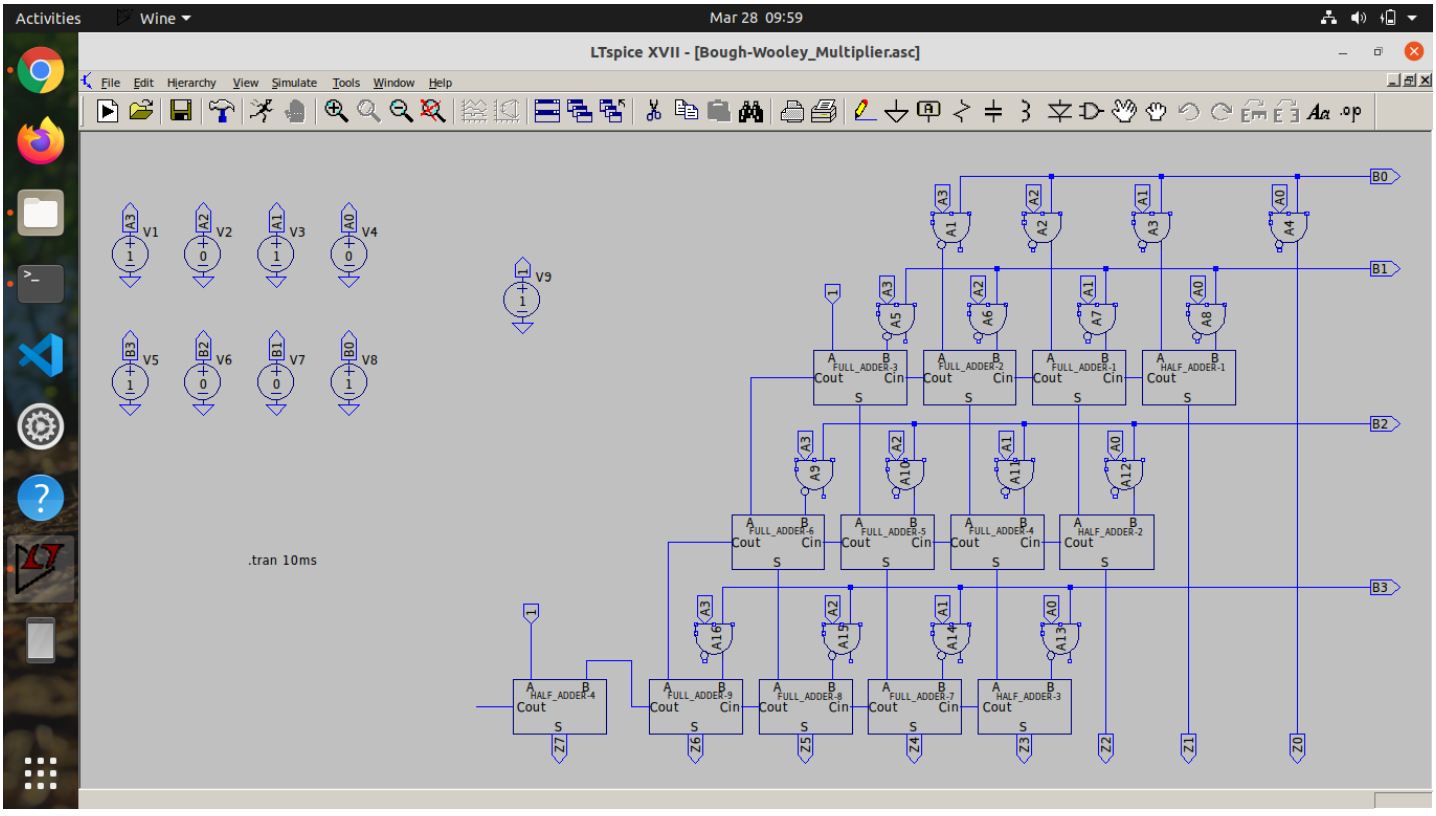
Enrolment Number – 19116040

Department – Computer Science & Engineering

Year – 2nd year

Part 1) To understand the operation of signed multiplier and implement the Bough-Wooley Multiplier Circuit.

The circuit diagram for the Bough-Wooley Multiplier Circuit is attached below. Explanation for the symbol of Half Adder and Full Adder is also attached subsequently. In addition, we have mod (19116040, 4) + 1 = 1. Thus, numbers corresponding to Set 1 of CSE group, that is -6 and -7 were chosen.



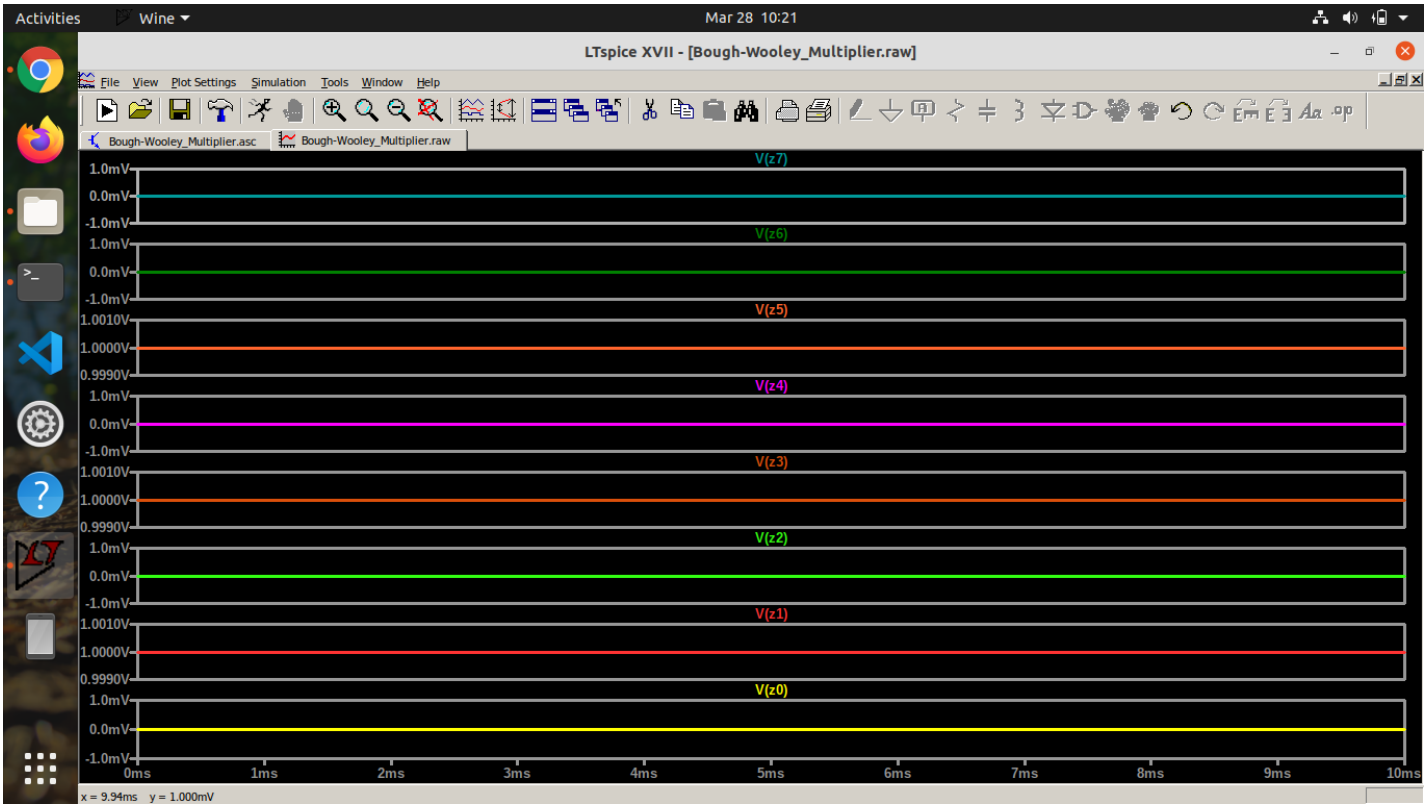
Now, for simulation we have A = -6 and B = -7.

Thus, A = A3 A2 A1 A0 = $(1010)_2$ and B = B3 B2 B1 B0 = $(1001)_2$.

In addition, a high voltage was applied through input labelled as “1”.

Result of simulation is given by Z, where Z = Z7 Z6 Z5 Z4 Z3 Z2 Z1 Z0

Corresponding Simulation result obtained is attached below:

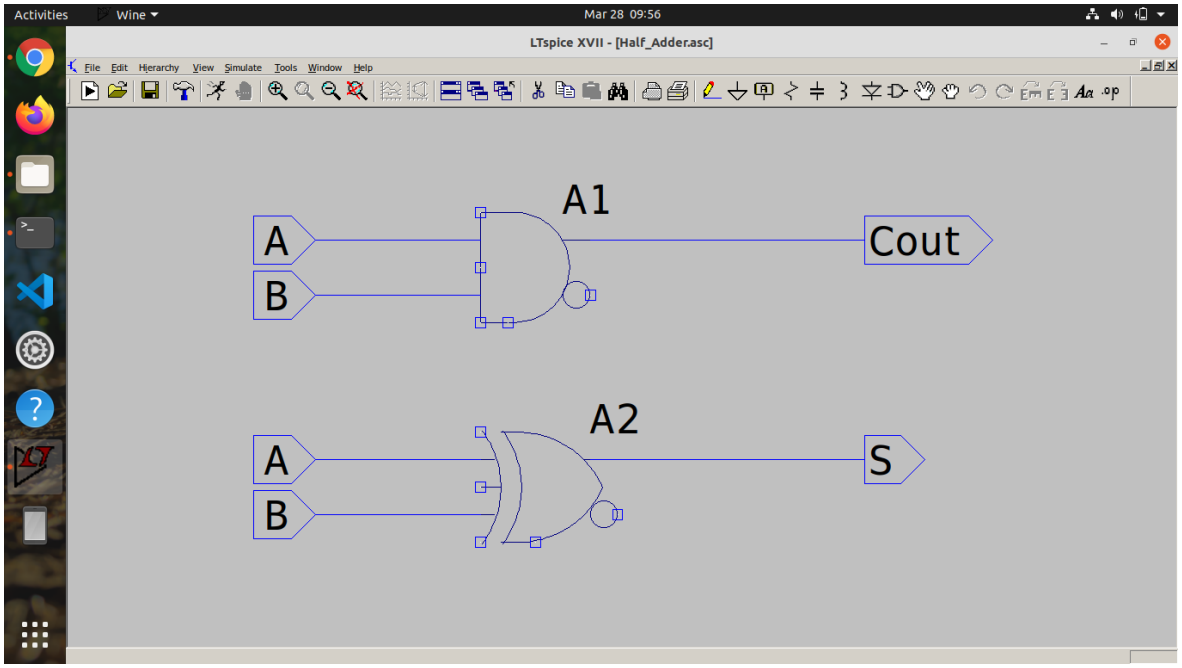


Thus, from above simulation Z = $(00101010)_2$ = $(42)_{10}$

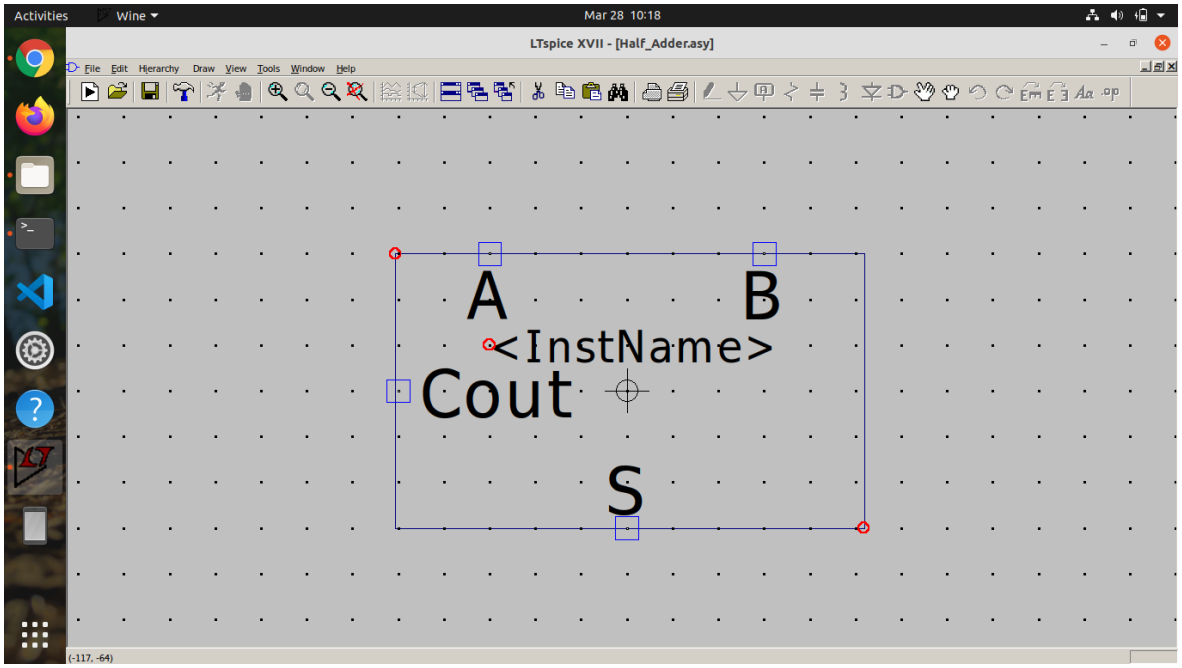
Thus, $(1010)_2$ * $(1001)_2$ = $(00101010)_2$, which is correct as (-6) * (-7) = (42)

Now, details of the **Half Adder** used in the implementation of the Bough-Wooley Circuit is attached below.

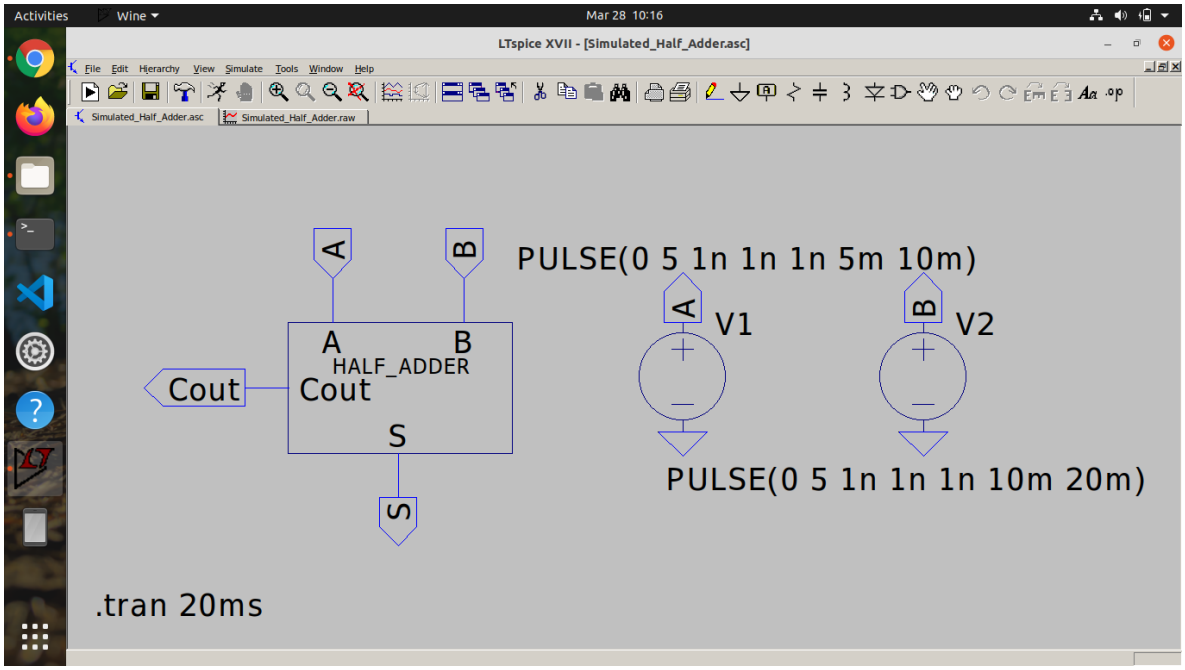
Circuit Diagram for Half Adder:



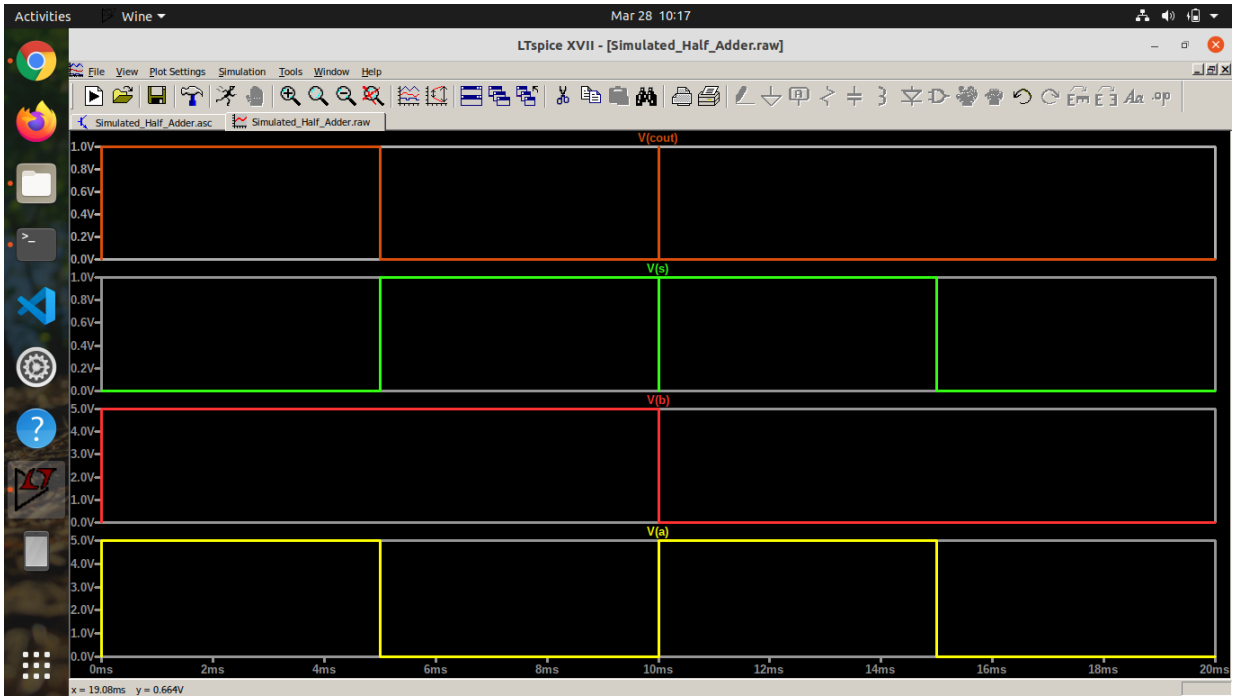
Corresponding Symbol for the Half Adder:



Circuit Used for Simulation of above Half Adder:

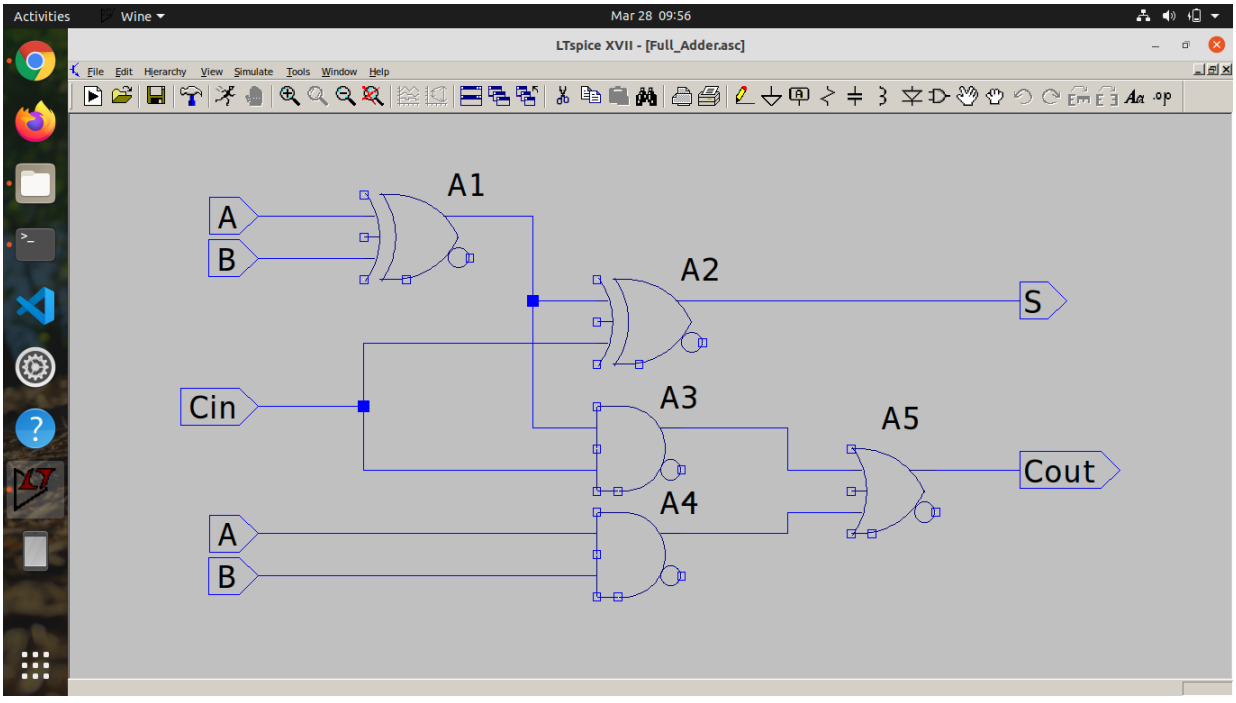


Corresponding Simulation for Half Adder:

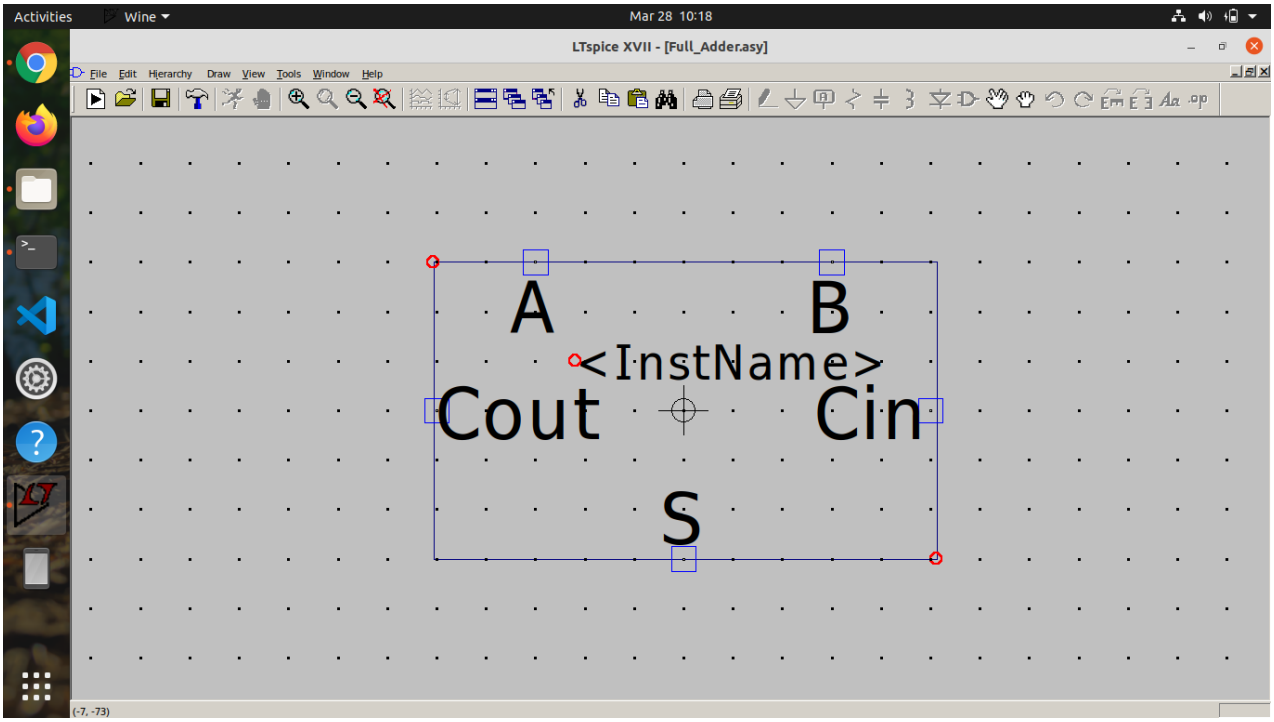


Now, details of the **Full Adder** used in the implementation of the Bough-Wooley Circuit is attached below.

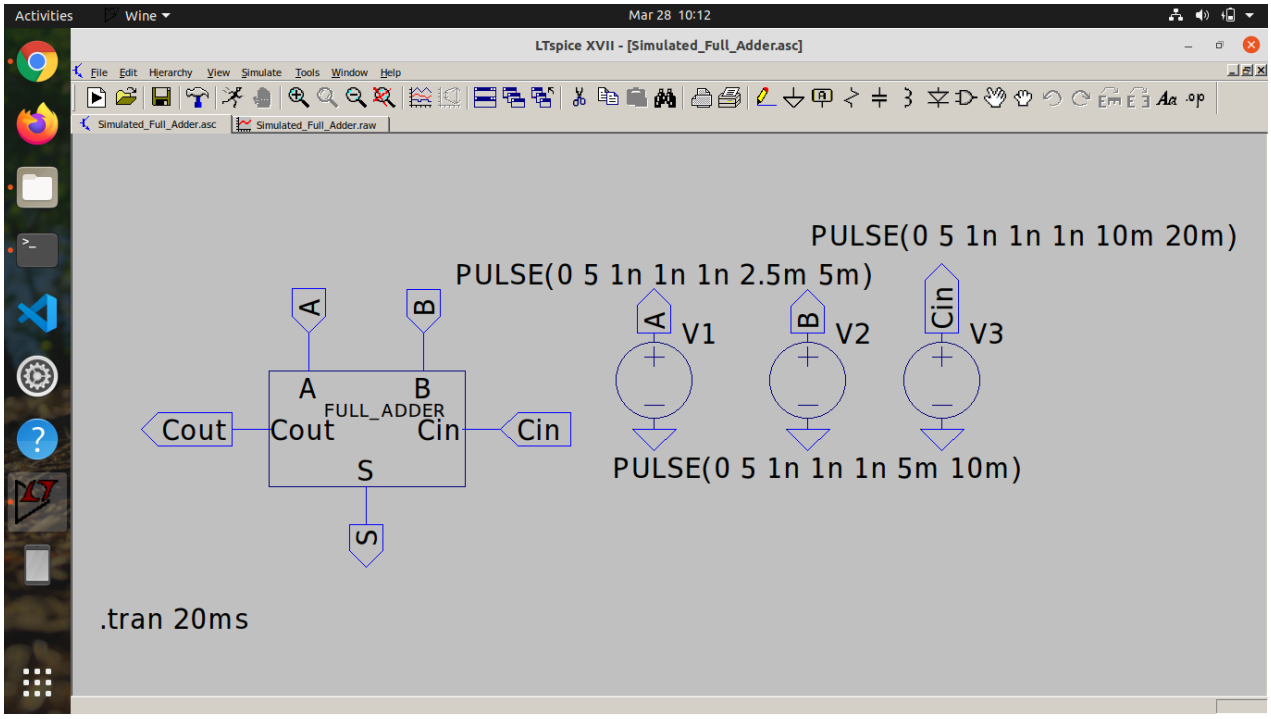
Circuit Diagram for Full Adder:



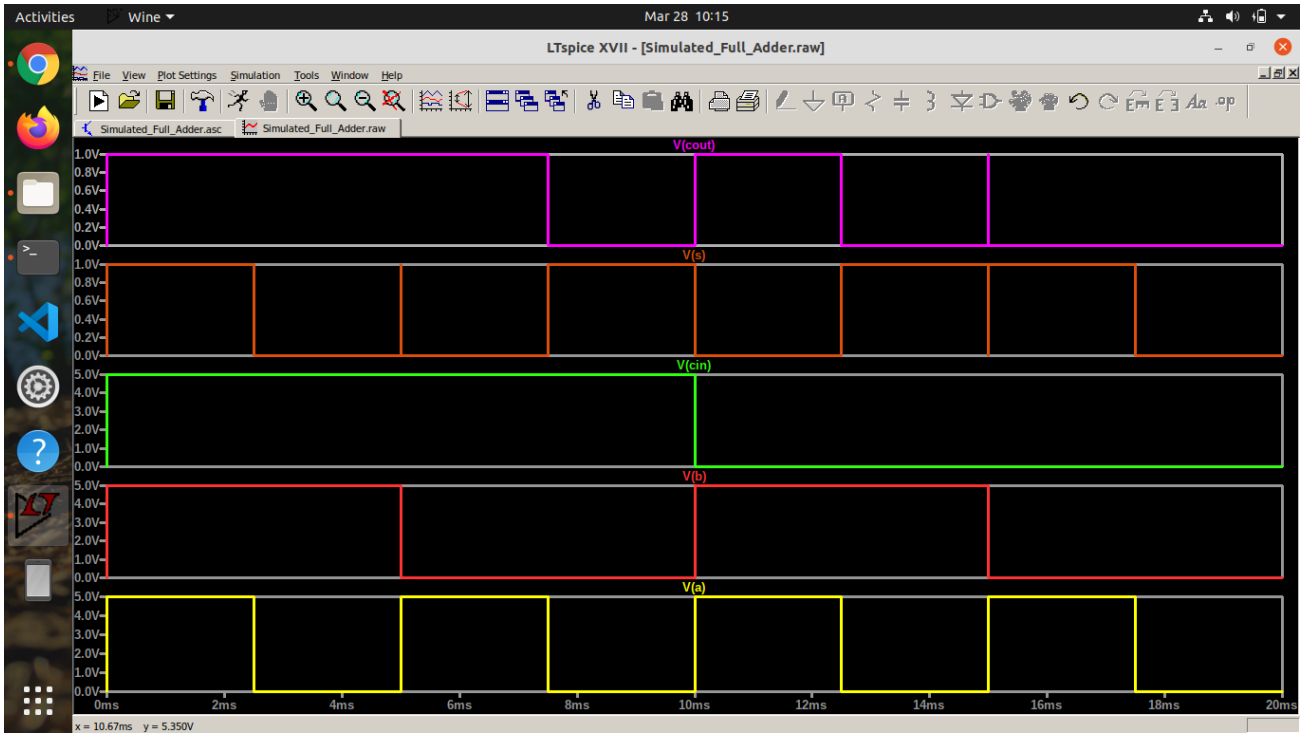
Corresponding Symbol for the Full Adder:



Circuit Used for Simulation of above Full Adder:

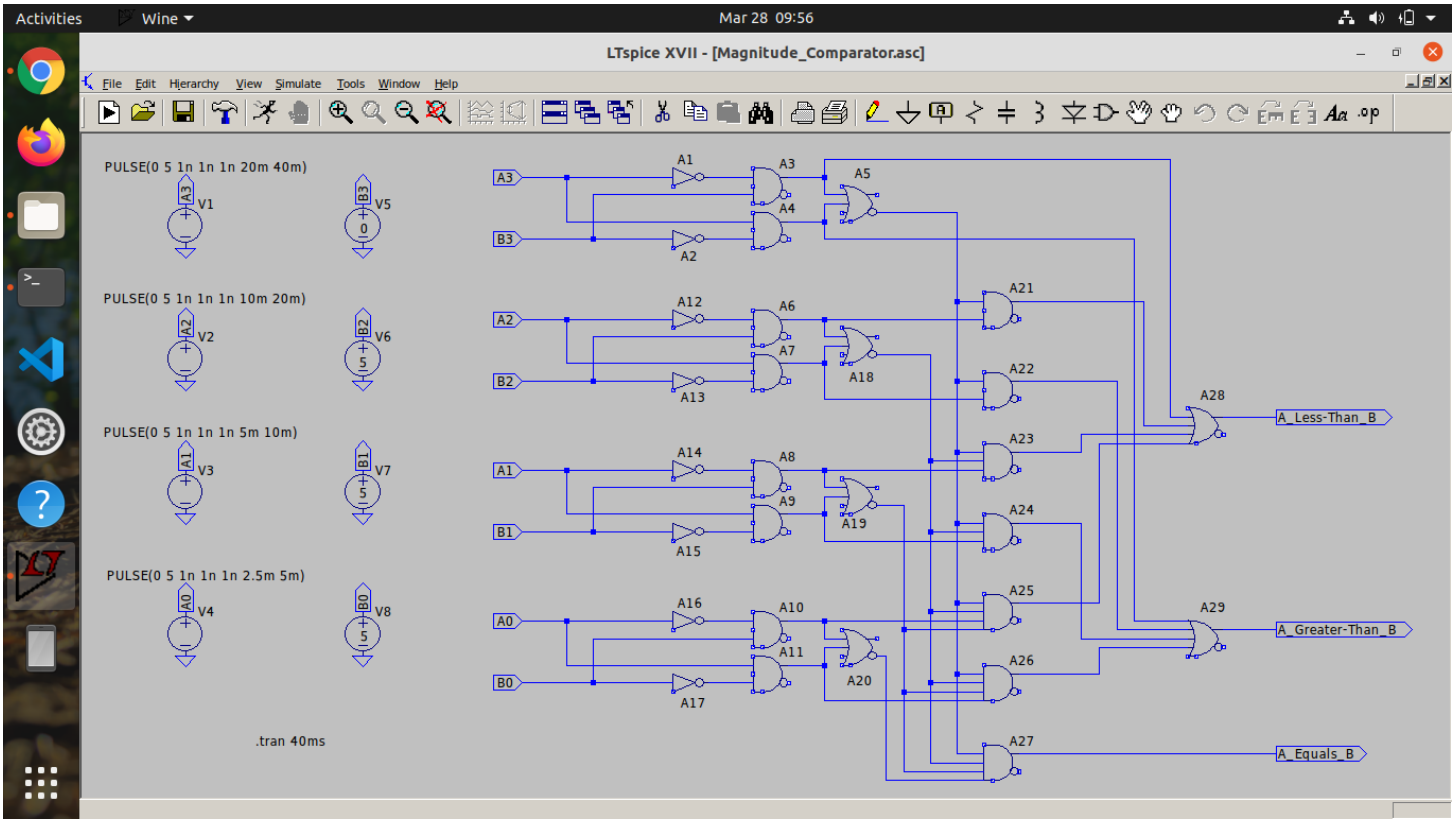


Corresponding Simulation result for the Full Adder:



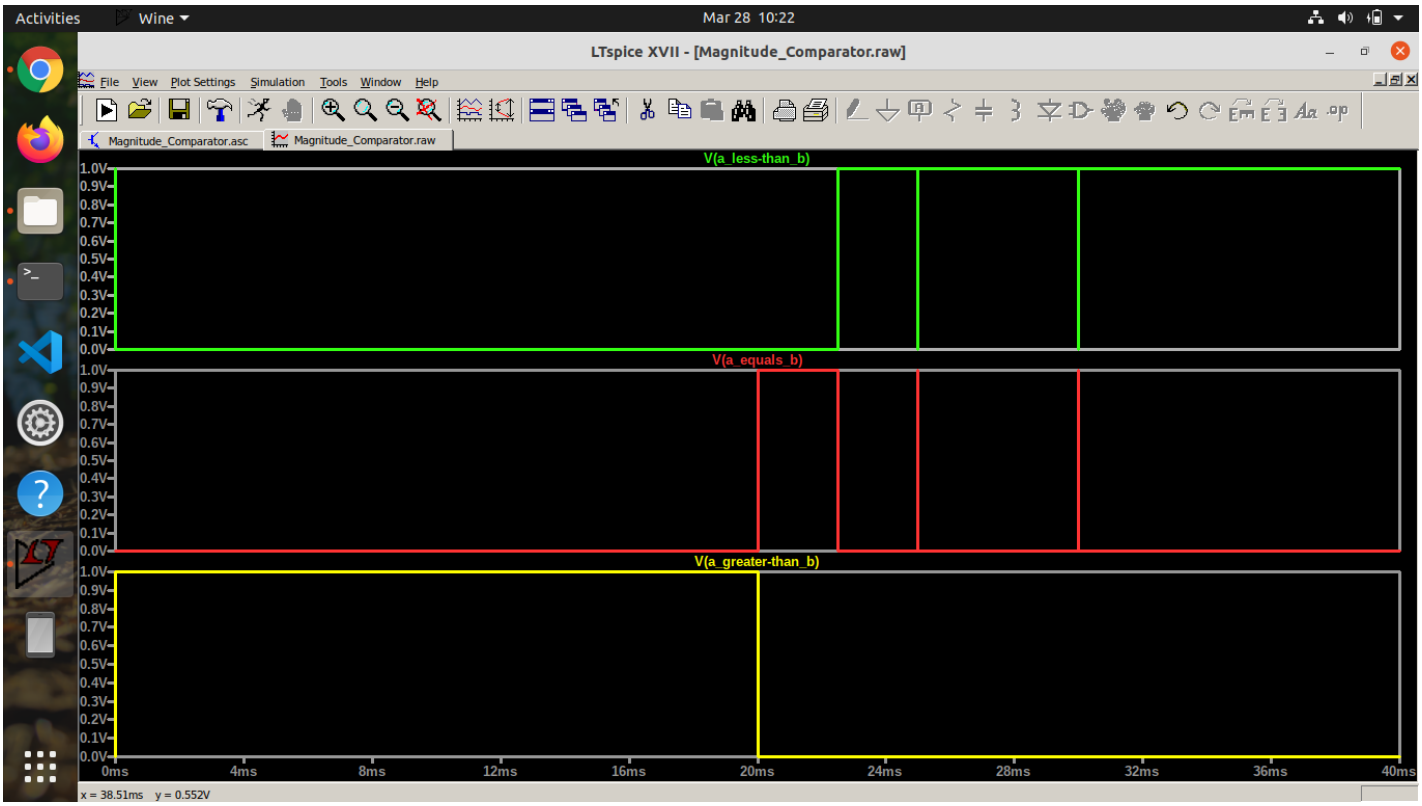
Part 2) To design a 4-bit magnitude comparator.

Circuit Diagram used for the 4-bit magnitude comparator is attached below.



For Simulation, one value was made constant. Here, B was made constant and was given the value of 7. The value of A was kept initially at 15. The value of A was then decreased by 1 unit after every 2.5ms. Simulation was run for 40ms and during this interval all the values of A between 15 to 0 was covered.

Corresponding Simulation result are attached below:



Now, we have following observations form the above simulation:

- During entire simulation, value of B was kept 7 (a constant value).
- During 0ms to 20ms, value of A decreased from 15 to 8. In this interval, A is greater than B. This is shown by yellow line in simulation.
- During 20ms to 22.5ms, value of A was 7. In this interval, A is equal to B. This is shown by red line in simulation.
- During 22.5ms to 40ms, value of A decreased from 6 to 0. In this interval, A is less than B. This is shown by green line in submission.

Thus, based on above observations, we can conclude that the above 4-bit magnitude comparator is working correctly.

However, this circuit will not work for signed integers in 2's complement form. Thus, we need to modify it to make it work in that case as well.

To make it work for signed integers in 2's complement form, **we need to interchange the location of A3 input and B3 input and keep rest of the circuit as it is**. This will have following impact:

- When we have two positive numbers, then A3 = 0 and B3 = 0. In that case, comparison will depend on rest of the bits of A and B. Since, this part of the circuit is not modified, we conclude that we will get correct result in case of two positive numbers.
- When we have one positive and one negative number. In that case, we have, $x_3 = A_3.B_3 + A_3'.B_3' = 0$. We have, following two cases:
    - If A3 = 1 and B3 = 0. In this case A is any negative number and B is any positive number. Now, $A_3.B_3' = 1$ and as a result our modified comparator will predict correctly that A is less than B.
    - If A3 = 0 and B3 = 1. In this case A is any positive number and B is any negative number. Now, $A_3'.B_3 = 1$ and as a result our modified comparator will predict correctly that A is greater than B.

  Thus, we conclude that we will get correct result in case there is one positive number and one negative number.

- When we have two negative numbers, then A3 = 1 and B3 = 1. In that case, comparison will depend on rest of the bits of A and B. Now, we have following list:

  $(-1)_{10} = (1111)_2$ $(-5)_{10} = (1011)_2$

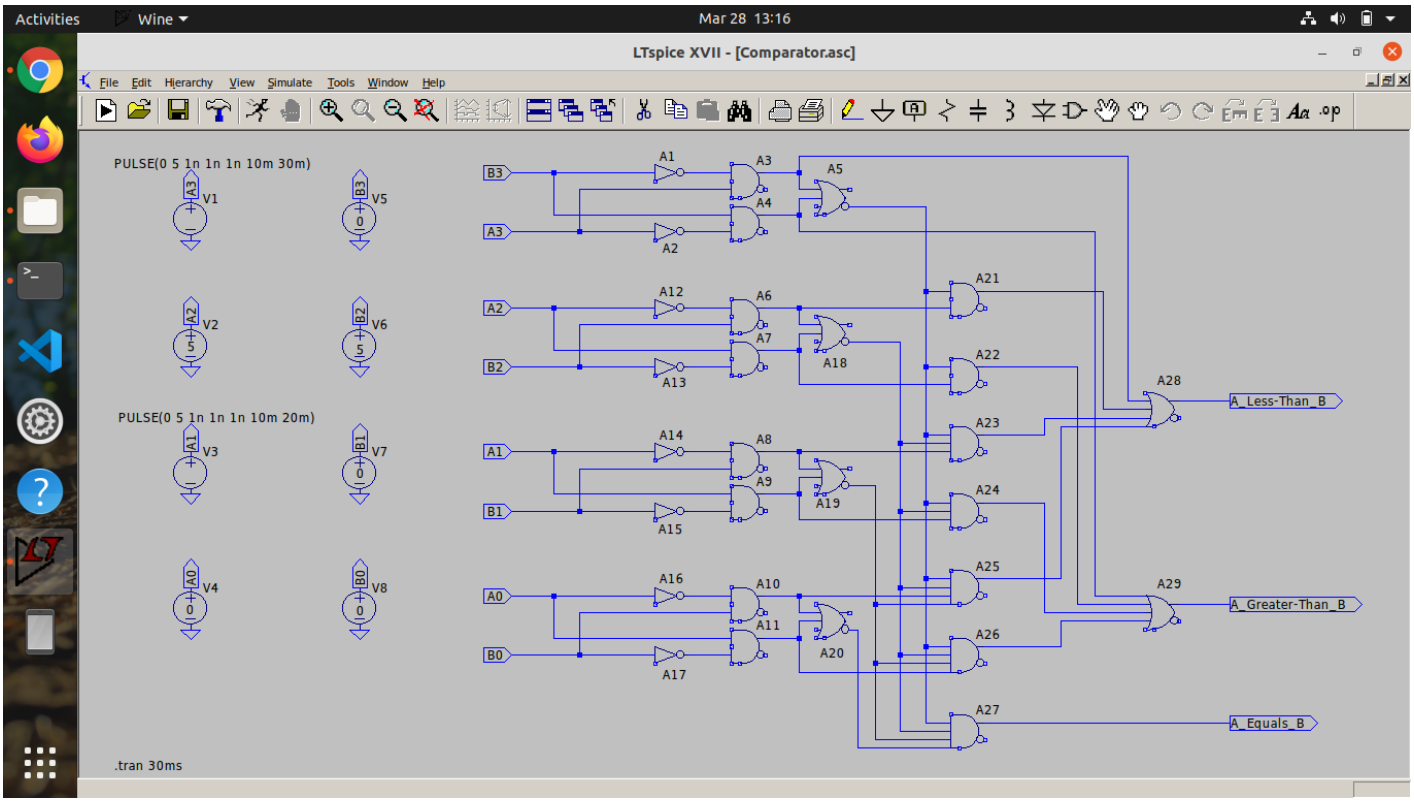
  $(-2)_{10} = (1110)_2$ $(-6)_{10} = (1010)_2$

  $(-3)_{10} = (1101)_2$ $(-7)_{10} = (1001)_2$

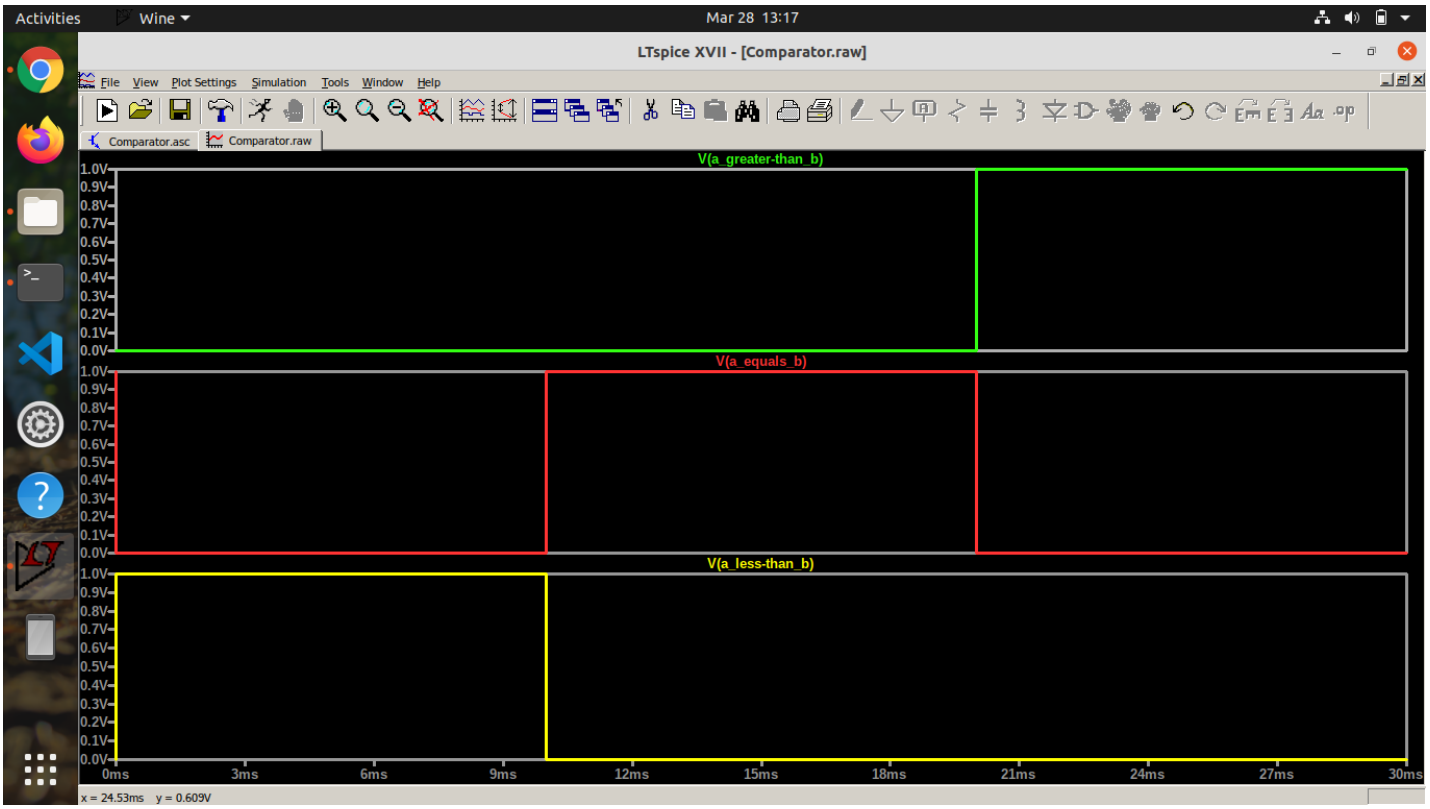
  $(-4)_{10} = (1100)_2$ $(-8)_{10} = (1000)_2$

  Now, since A3 = B3, our comparator will look at the remaining three bits of A and B and based on that it will conclude $(-1)_{10}$ as the largest negative number and $(-8)_{10}$ as the smallest negative number in the above list. This is because, remaining part of the circuit is not modified. Thus, we conclude that we will get correct result in case of two negative number.

Circuit Diagram for signed integer comparator case:



Corresponding Simulation Result:



Following observations can be obtained from the above simulation result:

- B was kept constant throughout the simulation result. The value of B was $(4)_{10}$ or $(0100)_2$.
- During 0ms to 10ms, A = $(1110)_2$ or $(-2)_{10}$. Thus, A is less than B which is depicted by the yellow line of simulation.
- During 10ms to 20ms, A = $(0100)_2$ or $(4)_{10}$. Thus, A is equal to B which is depicted by the red line of simulation.
- During 20ms to 30ms, A = $(0110)_2$ or $(6)_{10}$. Thus, A is greater than B which is depicted by the green line of simulation.

Thus, from above observations, we conclude that the modified comparator is working fine in case of signed integer as well.