EXP.NO: 10
Design and implement 3-bit full adder using logisim simulator.

AIM: To design and implement 32-bit full adder by using logisim simulator.

TRUTH TABLE:-

Row	Inputs			Outputs		Comment
	x	y	c_{in}	c_{out}	s	
0	0	0	0	0	0	$0+0+0=00_2$
1	0	0	1	0	1	$0+0+1=01_2$
2	0	1	0	0	1	$0+1+0=01_2$
3	0	1	1	1	0	$0+1+1=10_2$
4	1	0	0	0	1	$1+0+0=01_2$
5	1	0	1	1	0	$1+0+1=10_2$
6	1	1	0	1	0	$1+1+0=10_2$
7	1	1	1	1	1	$1+1+1=11_2$

STEP-0:-
Make a truth table with input coloumns X,Y and cin,in one column give the result of X XOR Y.In another column,give the result for(x XOR y)XOR cin,you should now see that :
S = x XOR y XOR Cin.

STEP-1:-
The sum-of-products equation for the carry output (Cout) is:Cout = x'·y·Cin + x·y'·Cin + x·y·Cin + x·y·Cin', is not a minimal expression. Show step by step how you can reduce the expression for Cout to end up with:
Cout = Cin·(x XOR y) + x·y.

STEP-2:-
It's now time to implement your 1-bit full adder in Logisim.
Start Logisim. On the department Unix System, type logisim in a shell and press enter. If you work on a laptop or form home, downlad and install Logisim from here. Open up add.circ in Logisim. Start by double-click on add1 to select the add1 circuit.

STEP-3:-
Complete the add8 circuit by combining eight 1-bit adders.

⟶Add three splitters to the circuit. Each splitter should have an input bit width of 8 and a fan out of 8. Attach two east-facing splitters to the 8-bit inputs A and B. Attach a west-facing splitter to the 8-bit output S.

⟶Create eight instances of the add1 circuit. connect the S outputs of the eight add1 instances to the splitter for the 8-bit Soutput.

⟶Connect the carry inputs and outputs of the eight add1 instances so that carries will propagate appropriately from the Ci input, through the 1-bit adders, to the Co output.

⟶Connect the A inputs of the eight add1 instances to the splitter for the Ainput.

⟶Connect the B inputs of the eight add1 instances to the splitter for the Binput.
Change the values of the Ci , A, and B inputs and observe the Co and S outputs to verify the correct operation of the circuit.

STEP-4:-

Complete the add32 circuit by combining four 8-bit adders.

⟶You will find three splitters in the circuit. Each splitter has an input bit width of 32 and a fan out of 4. Thus, each connection to a splitter represents 8 bits.

⟶Create four instances of the add8 circuit.

⟶Connect the 8-bit S outputs of the four add8 instances to the splitter for the 32-bit Soutput.

⟶Connect the carry inputs and outputs of the four add8 instances so that carries will propagate appropriately from the Ci input, through the 8-bit adders, to the Co output.

⟶Connect the 8-bit A inputs of the four add8 instances to the splitter for the Ainput.

⟶Connect the 8-bit B inputs of the four add8 instances to the splitter for the Binput.

STEP-5:-

⟶Within the main circuit, you will find a 32-bit adder connected side-by-side with your add 32 circuit.

⟶Change the values of the Ci , A, and B inputs and observe the Co and S outputs to verify the correct operation of your add32 circuit.

STEP-6:-

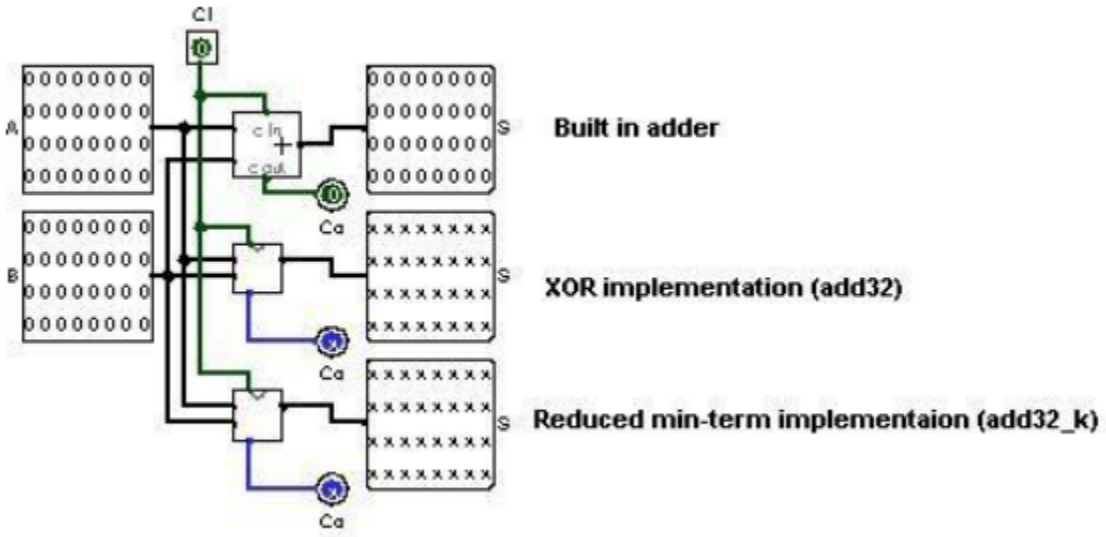
⟶There is more than one way to implemt a logical function. An alternative expression for Cout can be found by reducing the expression using only min-terms

⟶Use a Karnaugh map to reduce the expression for Cout.Note: using Karnaugh map to reduce the expression for the Sum will no be possible, it will result in the original sum-of-products for the Sum.

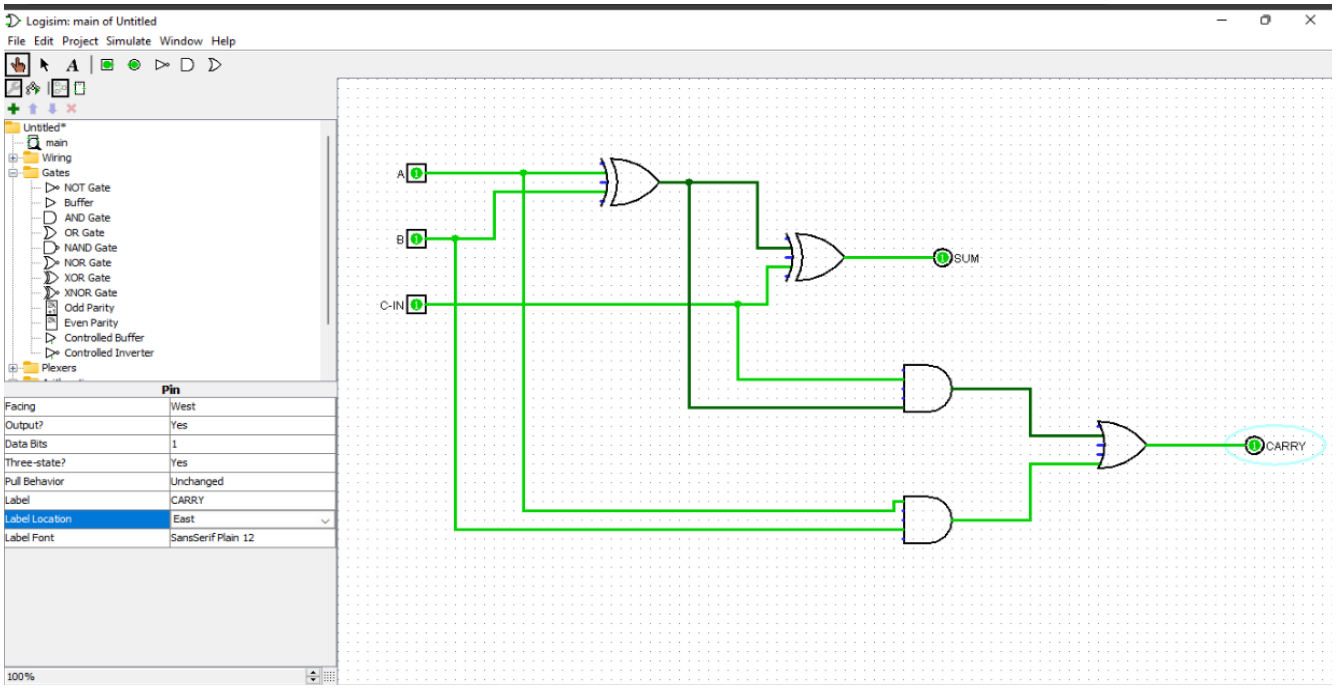
⟶Add a new circuit to the project named add1_k and implement a new version of a 1 bit full adder using the new expression for Cout and the original sum-of-producs exreppsions for the SumSimlary, add new circuits named add8_k and add32_k to construct an alternate version of the 32 bit full adder.

⟶Add the add32_k component to the main circuit along with the other adders and verify that they give the same results.

CIRCUIT DIAGRAM:



OUTPUT:



RESULT:-

Thus the designing of the 32-bit full adder using logisim simulator has been implemented successfully.