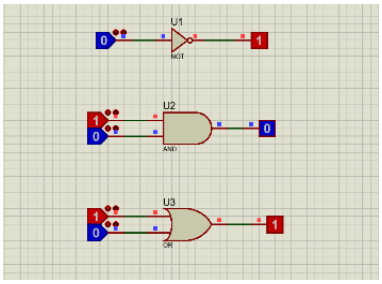
Lab 01:

Objective: To design and verify the operation of basic logic gates- AND,OR and NOT using proteus simulation software.

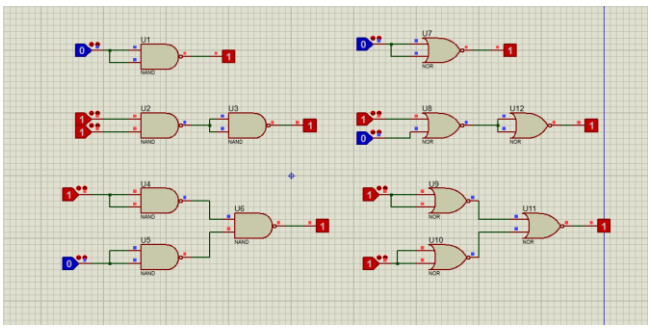


Truth table:

Input A	Output B	AND (Y=A·B)	OR (Y=A+B)	NOT ($Y=\overline{A}$)
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

Lab 02:

Objective: To design and verify the operation of universal logic gates-NAND and NOR using proteus simulation software.

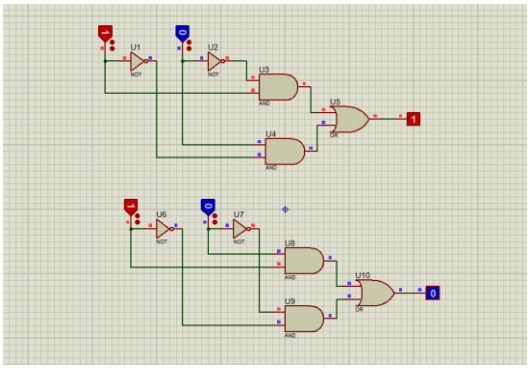


Truth Table:

A	B	A·B	A+B	NAND ($Y=\overline{A\cdot B}$)	NOR ($Y=\overline{A+B}$)
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	1	0
1	1	1	1	0	0

Lab 03:

Objective: To implement and verify XOR and XNOR gates using basic logic gates in proteus simulation software.

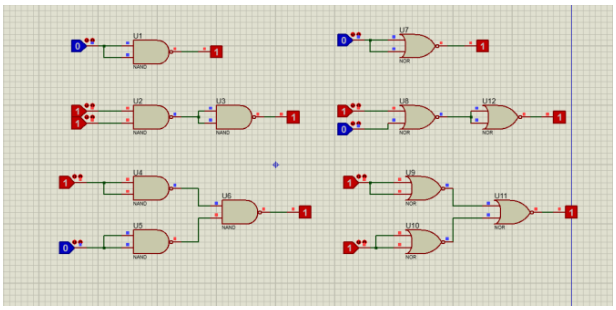


Truth table:

A	B	XOR	XNOR
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

Lab 04:

Objective: To verify De Morgan's two laws using logic gates implementation in proteus and compare both sides of the expression.

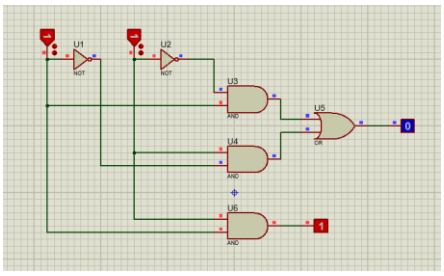


Truth table:

A	B	A·B	A+B	$\overline{A \cdot B}$	$\overline{A}$	$\overline{B}$	$\overline{A}+\overline{B}$	$\overline{A+B}$	$\overline{A}\cdot\overline{B}$
0	0	0	0	1	1	1	1	1	1
0	1	0	1	1	1	0	1	0	0
1	0	0	1	1	0	1	1	0	0
1	1	1	1	0	0	0	0	0	0

Lab 06:

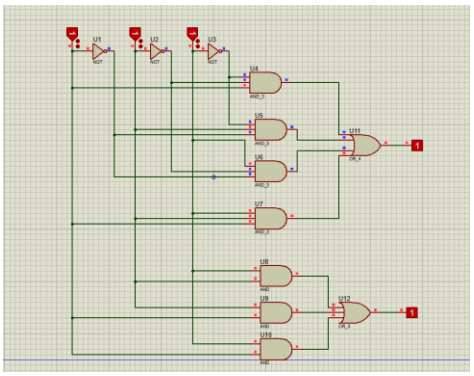
Objective: To design a **Half Adder** circuit using fundamental logic gates (XOR and AND).



A	B	A ⊕ B
0	0	0
0	1	0
1	0	0
1	1	1

Lab 07:

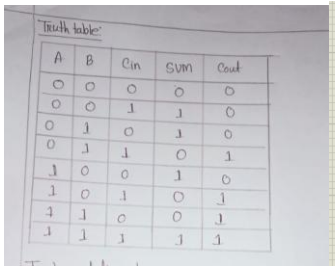
Objective: To simulate the designed Full Adder circuit using proteus simulation software.



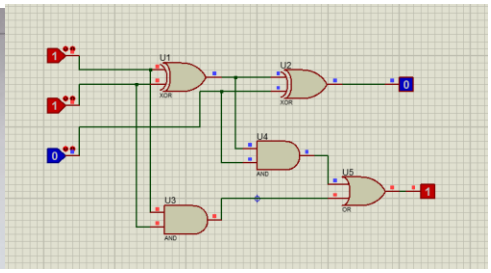
A	B	Cin	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Lab 08:

Objective: To design a full adder circuit by combining two half adders and verift it's output.

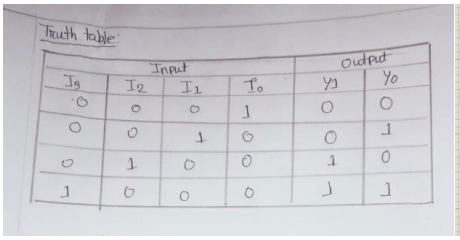

Truth table:

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

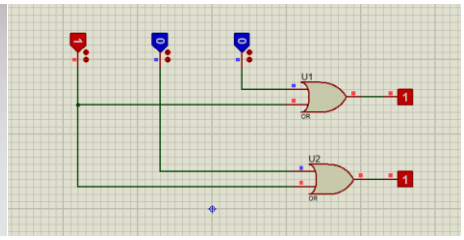


Lab 09:

Objective: To design a 4 bit encoder using logic gates and verify the binary output codes for all valid input.

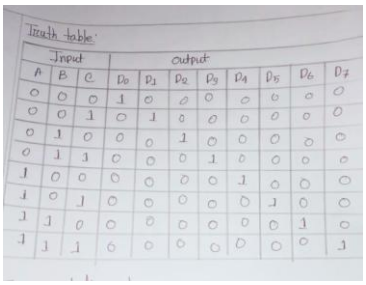

Truth table:

Input				Output	
I_3	I_2	I_1	I_0	y_1	y_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

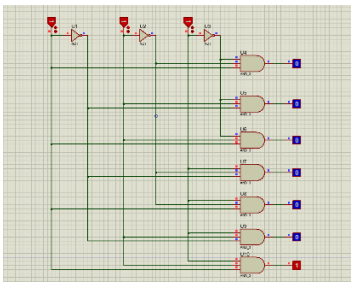


Lab 10:

Objective: To design a 3 to 8 bit decoder using logic gates and verify that only one output activates for each input combination.

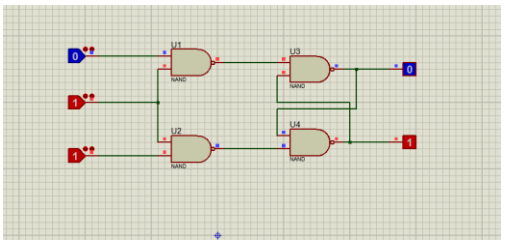

Truth table:

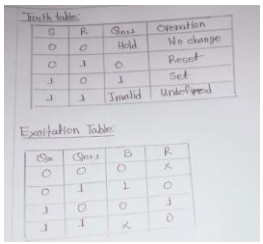
Input			output							
A	B	C	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1



Lab 11:

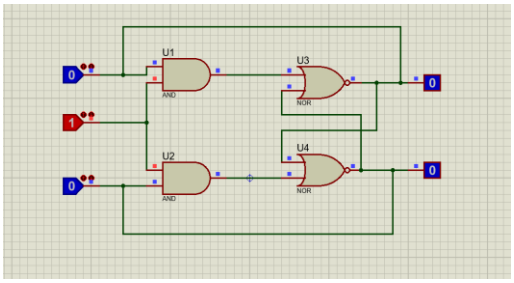
Objective: To design an SR flip=flop verify it's truth and extitation tables.

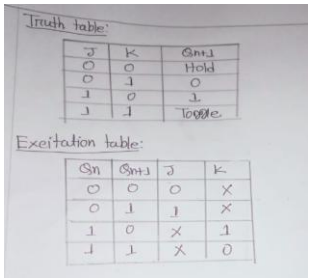




Lab 12:

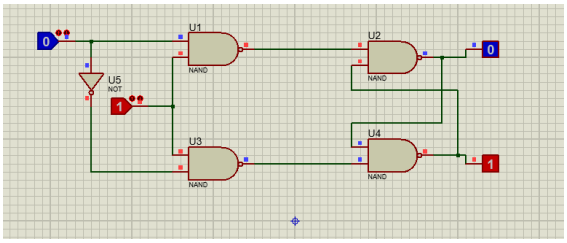
Objective: To design and verify the JK flip-flop operation including toggle behavoir.

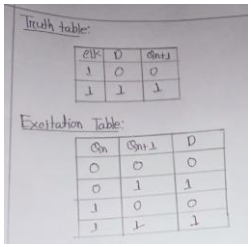




Lab 13:

Objective: To implement and test the D flip-flop using logic gates.





Lab 14:

Objective: To design a T flip-flop and verify it's toggle behavoir.