Digital Electronics

Experiment 2

Name: Ayush Jain

SAP ID: 60004200132

Div: B1

Branch: Computer Engineering

AIM : To implement the logic functions i.e. AND, OR, NOT, Ex-OR, Ex- NOR and a logical expression with the help of NAND and NOR universal gates respectively.

NAND as AND gate

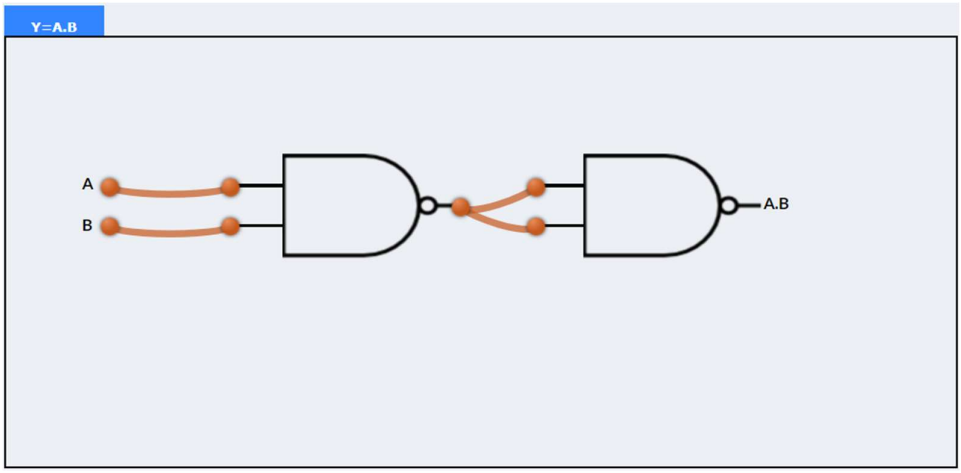
THEORY : A NAND produces complement of AND gate. So, if the output of a NAND gate is inverted, overall output will be that of an AND gate.

$$Y = ((A.B)')'$$

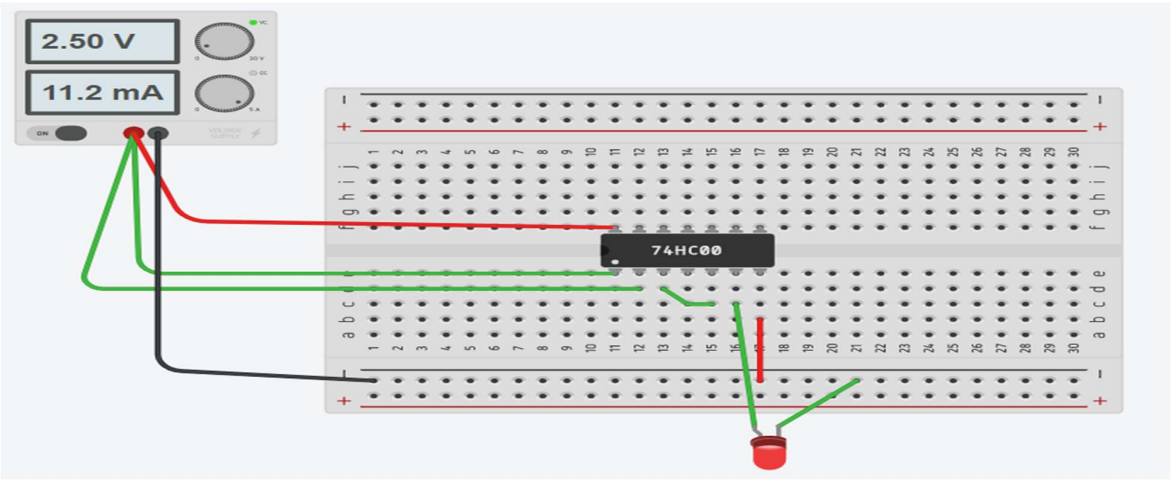
$$Y = (A.B)$$

Input		Output
A	B	F = A.B
0	0	0
0	1	0
1	0	0
1	1	1

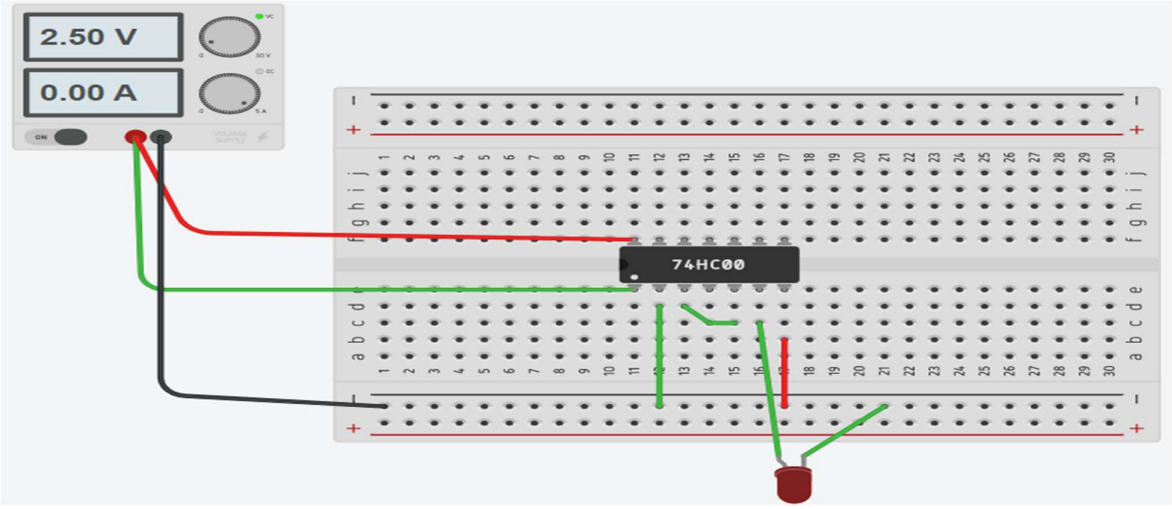
OUTPUT:



Tinkercad:



NAND as AND gate with inputs 1 and 1



NAND as AND gate with inputs 1 and 0

NAND as OR gate

THEORY : From DeMorgan's theorems:

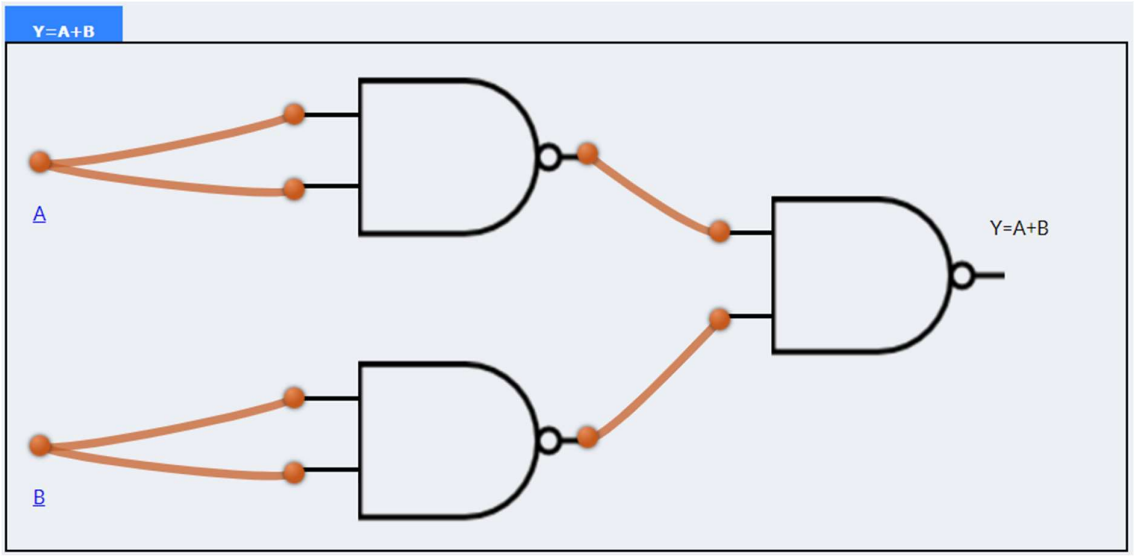
(A.B)' = A' + B'

(A'.B')' = A'' + B'' = A + B

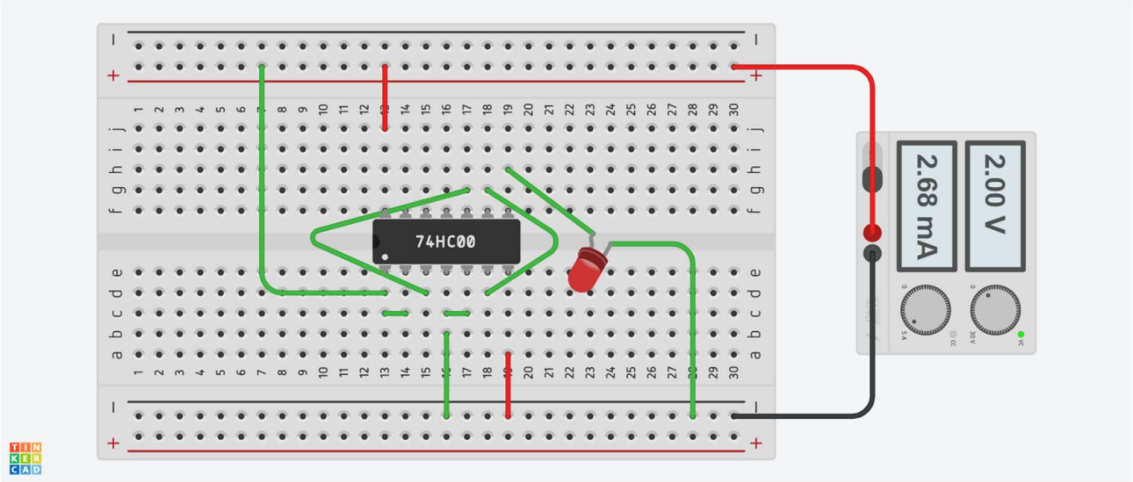
So, give the inverted inputs to a NAND gate, obtain OR operation at output.

A	B	X = A+B
0	0	0
0	1	1
1	0	1
1	1	1

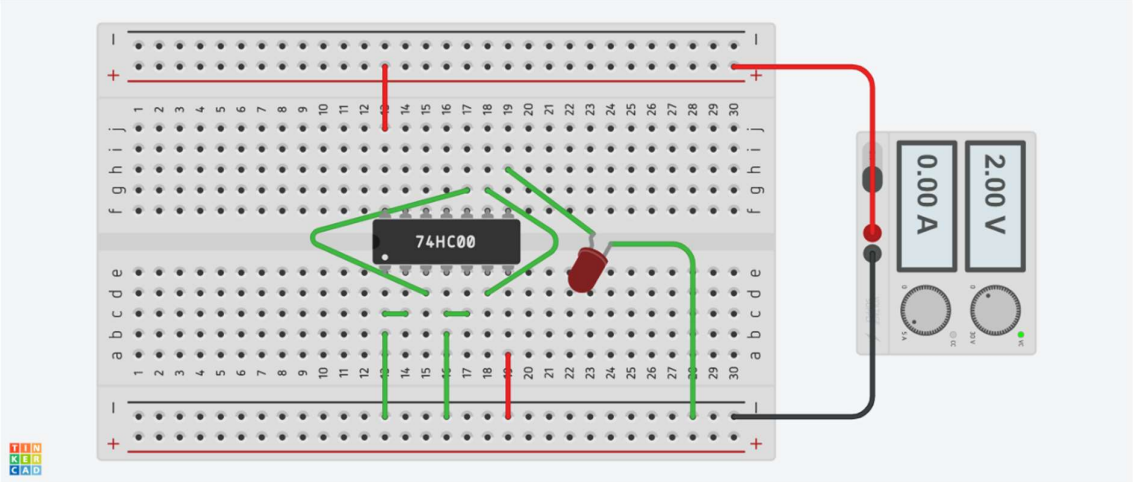
OUTPUT:



Tinkercad:



NAND as OR gate with inputs 1 and 0



NAND as OR gate with inputs 0 and 0

NAND as X-OR gate

THEORY: The output of a two input Ex-OR gate is shown by: Y = A'B + AB'. This can be achieved with the logic diagram shown in the left side.

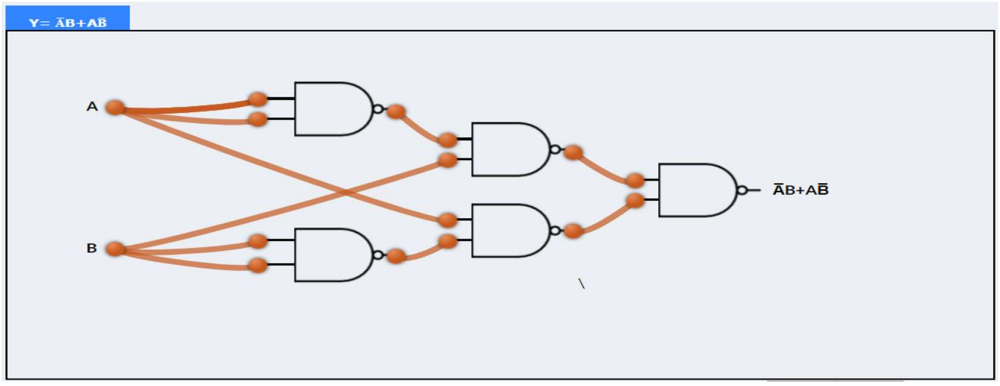
Gate No.	Inputs	Output
1	A, B	(AB)'
2	A, (AB)'	(A (AB)')'
3	(AB)', B	(B (AB)')'
4	(A (AB)')', (B (AB)')'	A'B + AB'

Now the ouput from gate no. 4 is the overall output of the configuration.

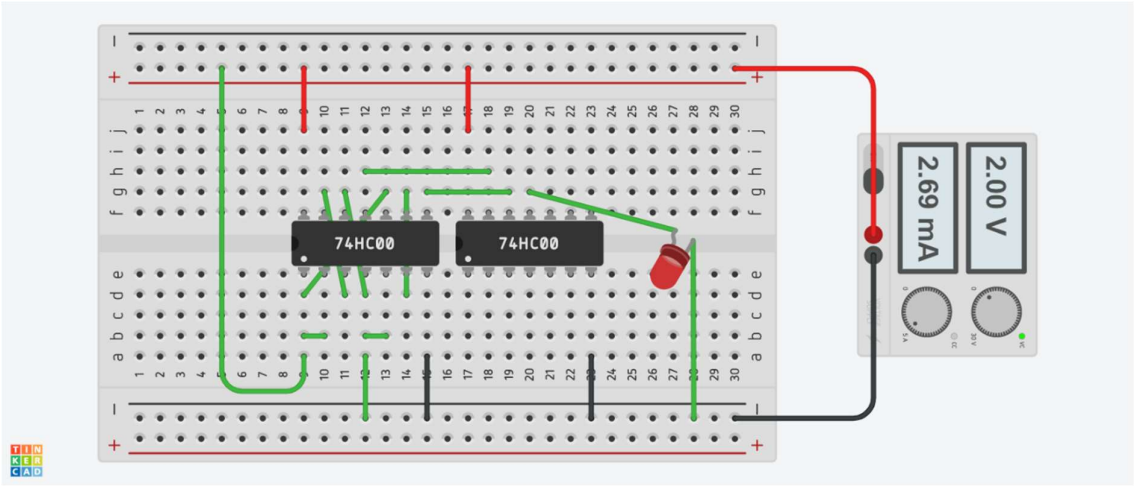
$$
\begin{aligned}
Y &= ((A\,(AB)')'\,(B\,(AB)')')' \\
&= (A(AB)')'' + (B(AB)')'' \\
&= (A(AB)') + (B(AB)') \\
&= (A(A' + B)') + (B(A' + B')) \\
&= (AA' + AB') + (BA' + BB') \\
&= (0 + AB' + BA' + 0) \\
&= AB' + BA' \\
\Rightarrow \quad Y &= AB' + A'B
\end{aligned}
$$

A	B	A **XOR** B
0	0	0
0	1	1
1	0	1
1	1	0

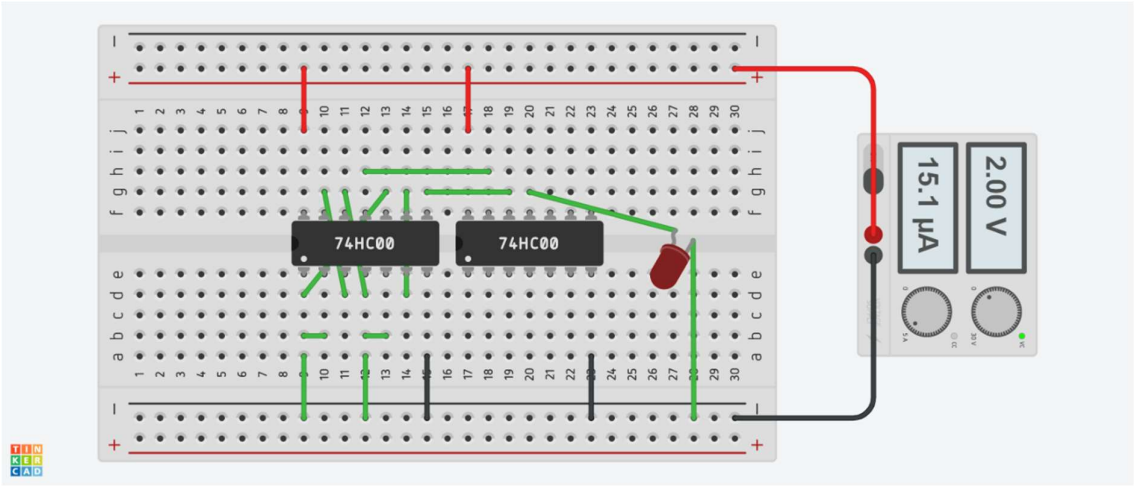
OUTPUT:



Tinkercad:



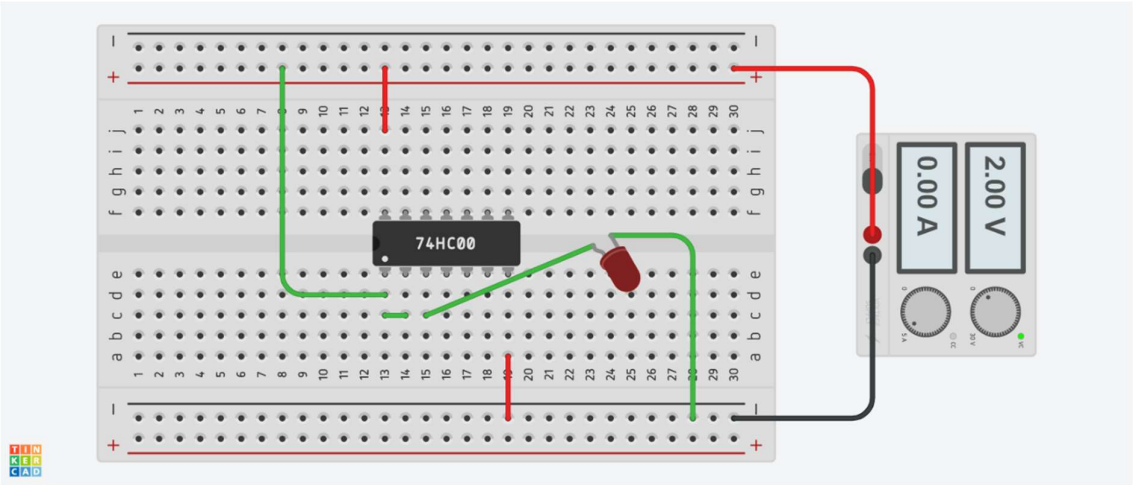
NAND as X-OR gate with inputs 1 and 0



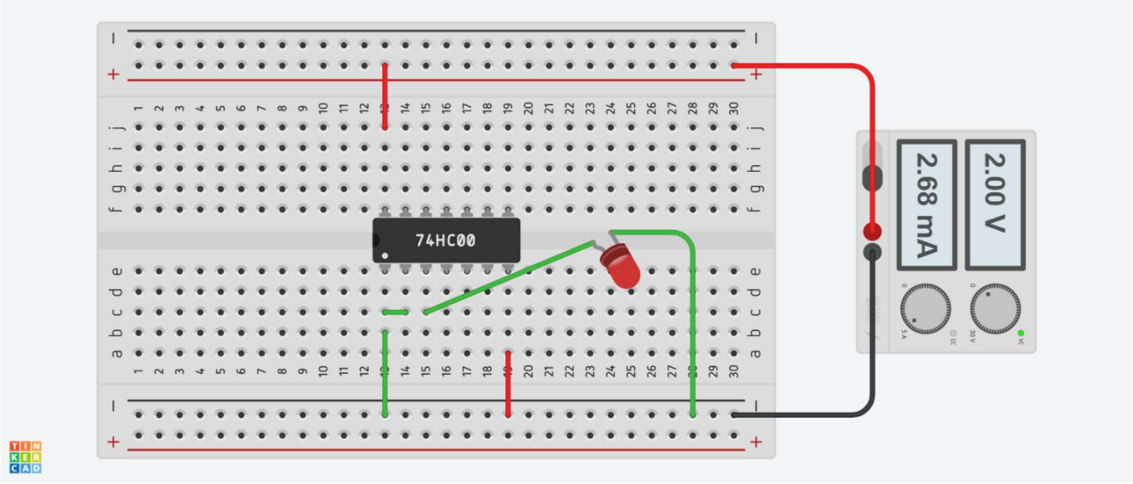
NAND as X-OR gate with inputs 0 and 0

NAND as NOT gate

Tinkercad:



NAND as NOT gate with input 1



NAND as NOT gate with input 0

NOR as AND gate

THEORY : From DeMorgan's theorems:

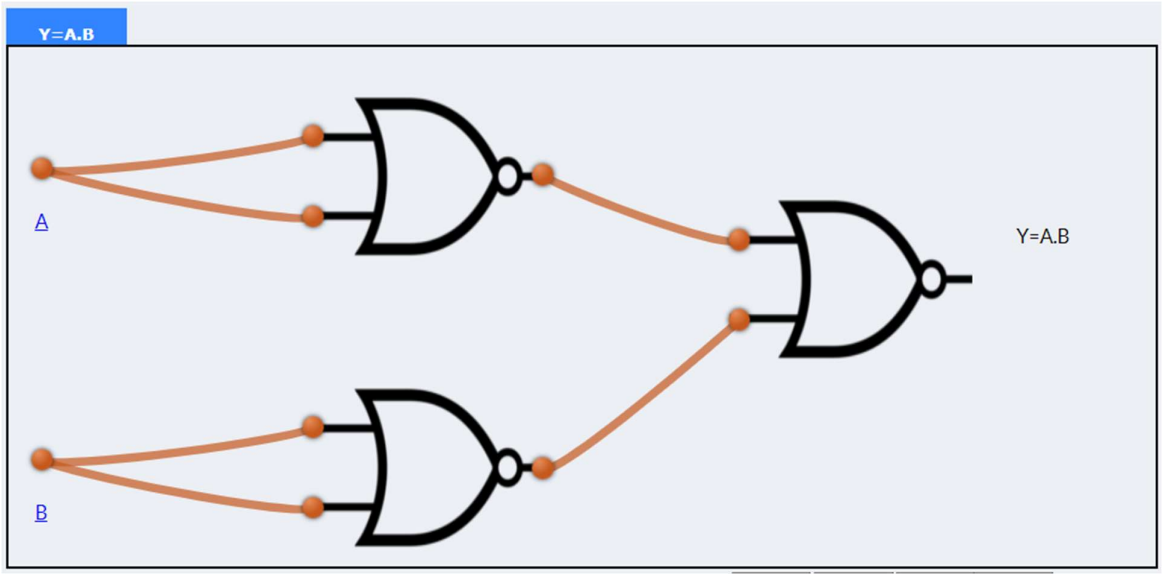
(A+B)' = A'B'

(A'+B')' = A''B'' = AB

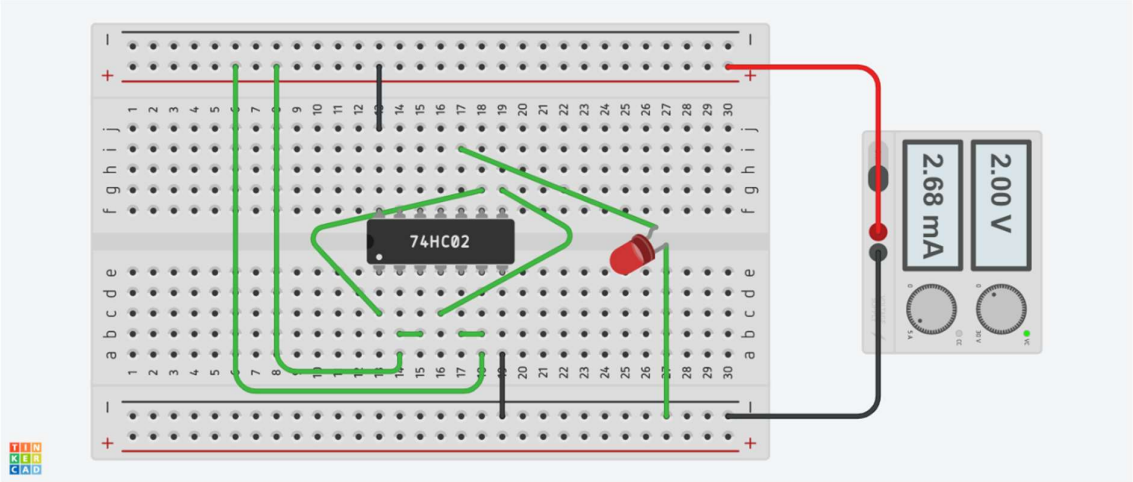
So, give the inverted inputs to a NOR gate, obtain AND operation at output.

Input		Output
A	B	F = A.B
0	0	0
0	1	0
1	0	0
1	1	1

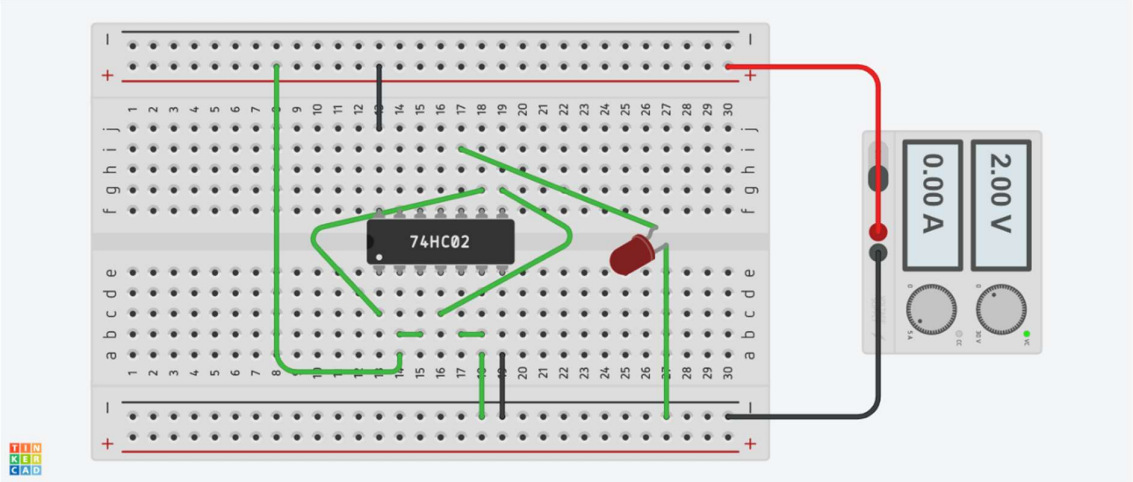
OUTPUT:



Tinkercad:



NOR as AND gate with inputs 1 and 1



NOR as AND gate with inputs 1 and 0

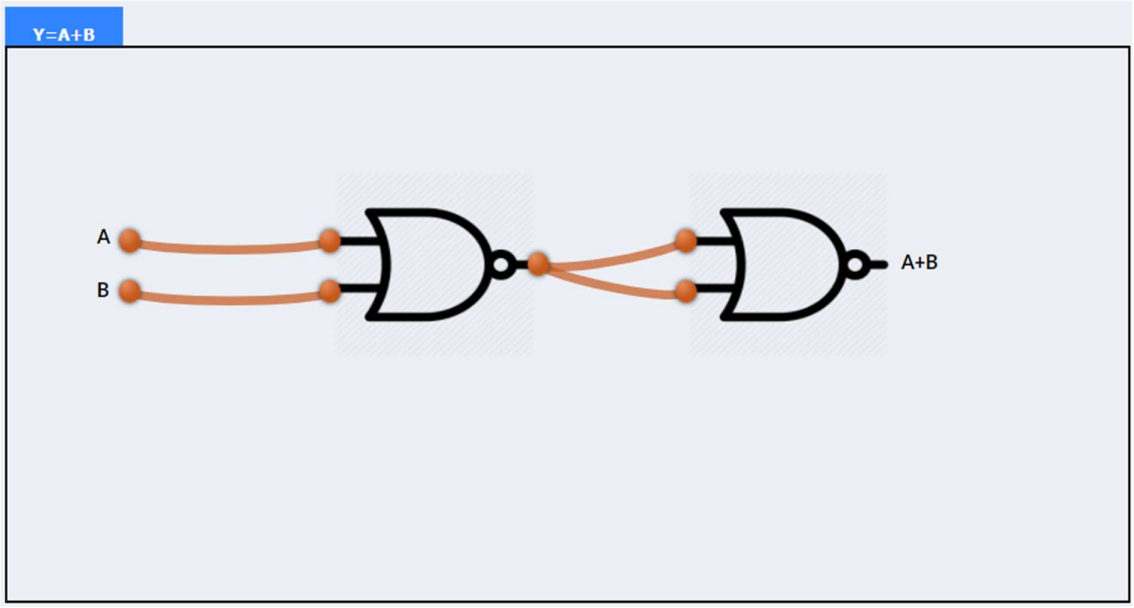
NOR as OR gate

THEORY : A NOR produces complement of OR gate. So, if the output of a NOR gate is inverted, overall output will be that of an OR gate.

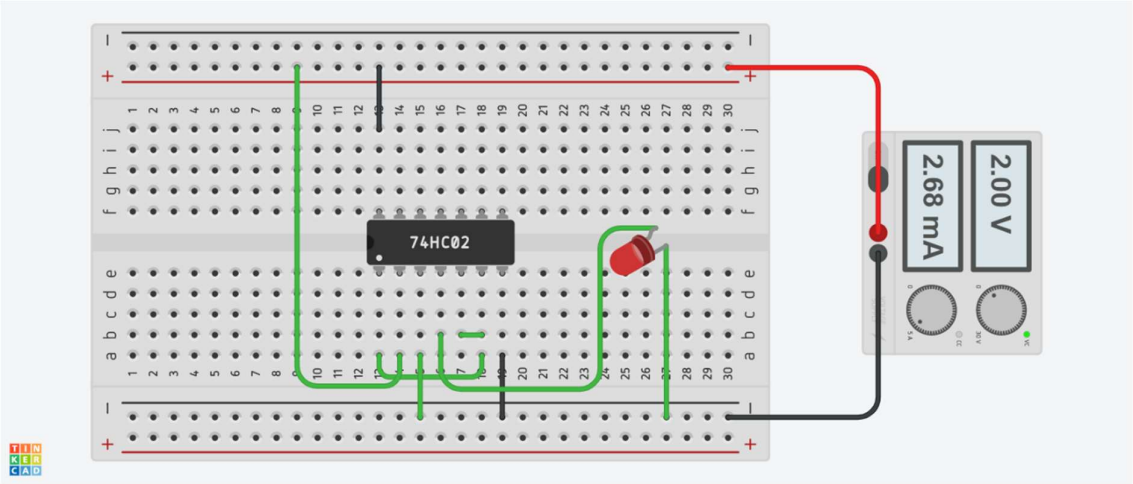
Y = ((A+B)')' Y = (A+B)

A	B	X = A+B
0	0	0
0	1	1
1	0	1
1	1	1

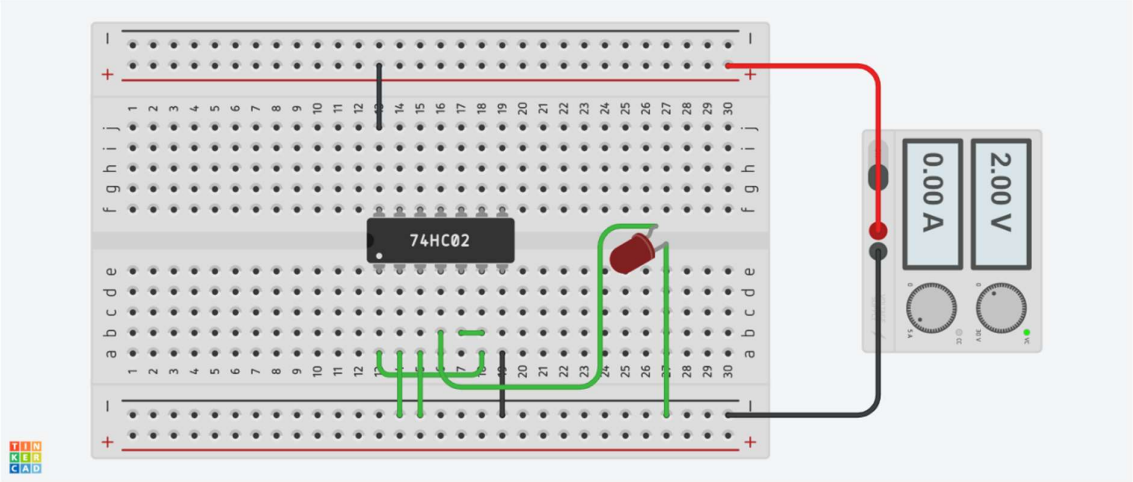
OUTPUT:



Tinkercad:



NOR as OR gate with inputs 1 and 0



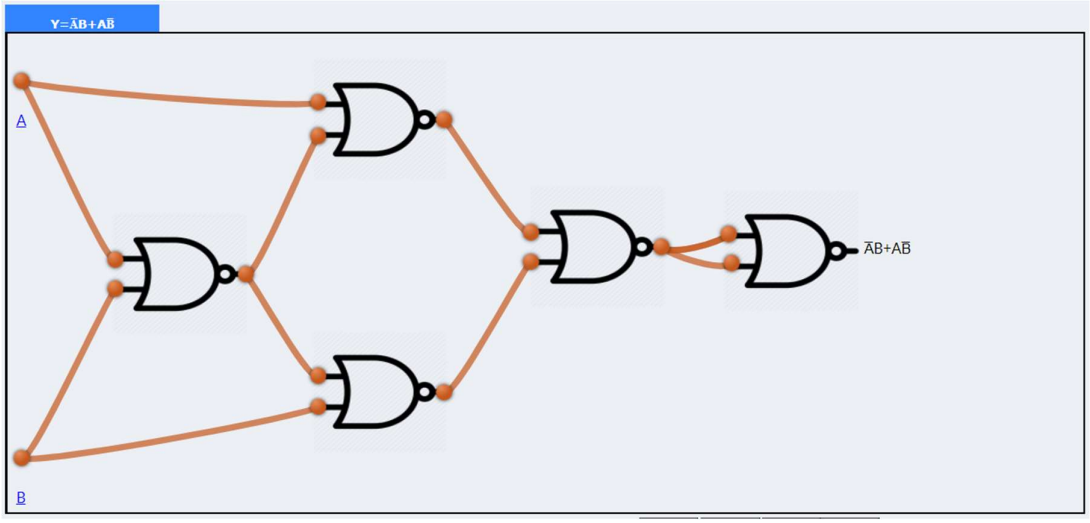
NOR as OR gate with inputs 0 and 0

NOR as X-OR gate

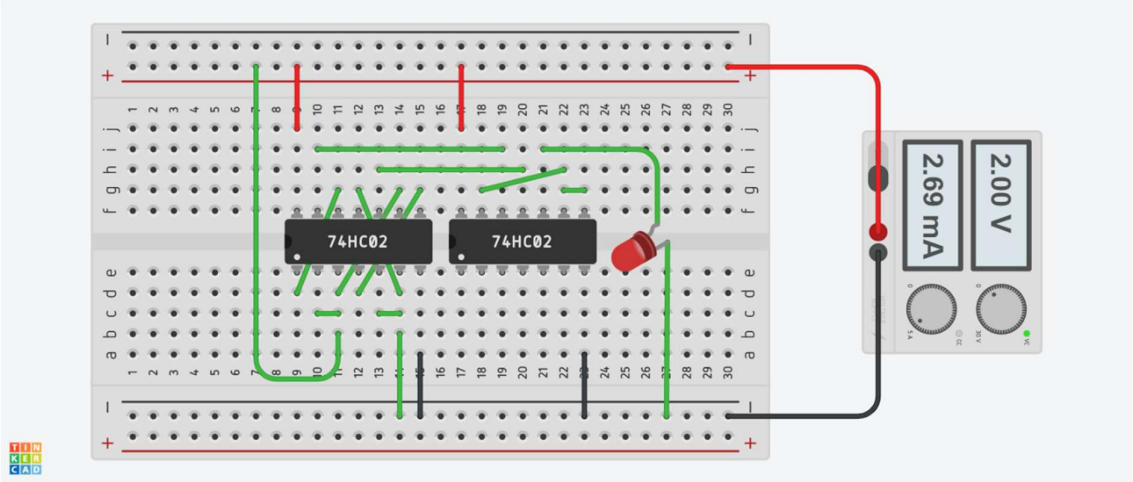
THEORY : Ex-OR gate is actually Ex-NOR gate followed by NOT gate. So give the output of Ex-NOR gate to a NOT gate, overall output is that of an Ex-OR gate. Y = A'B+ AB'

A	B	A **XOR** B
0	0	0
0	1	1
1	0	1
1	1	0

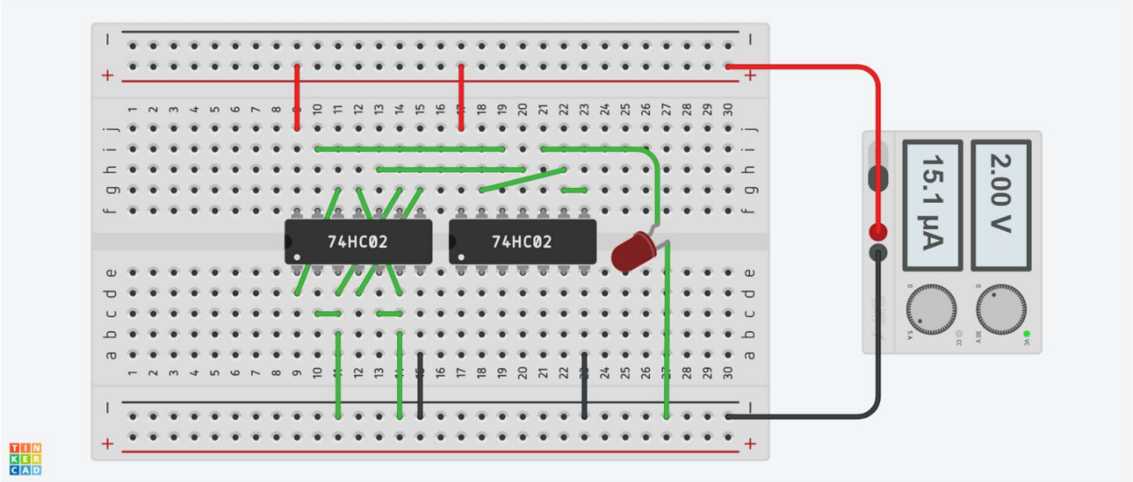
OUTPUT:



Tinkercad:



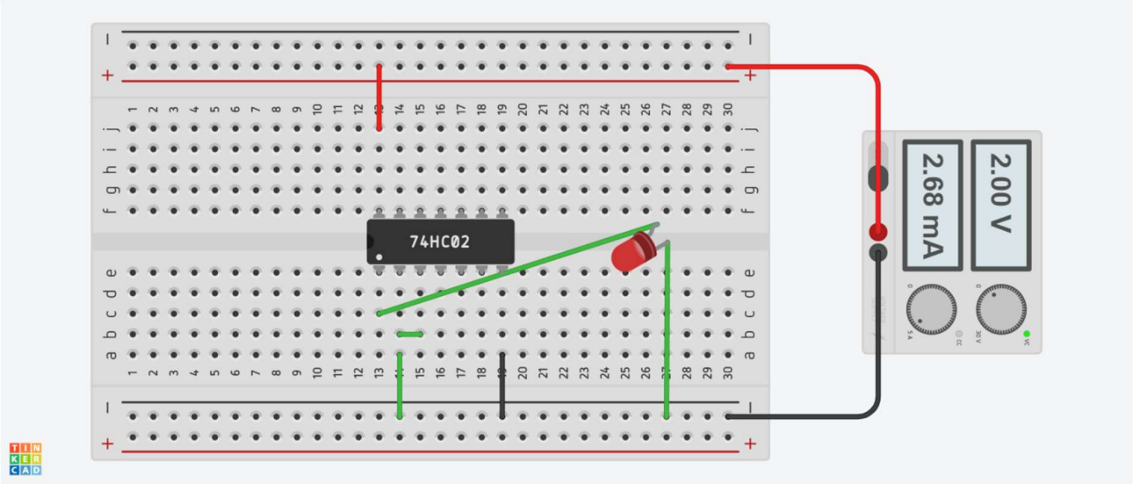
NOR as X-OR gate with inputs 1 and 0



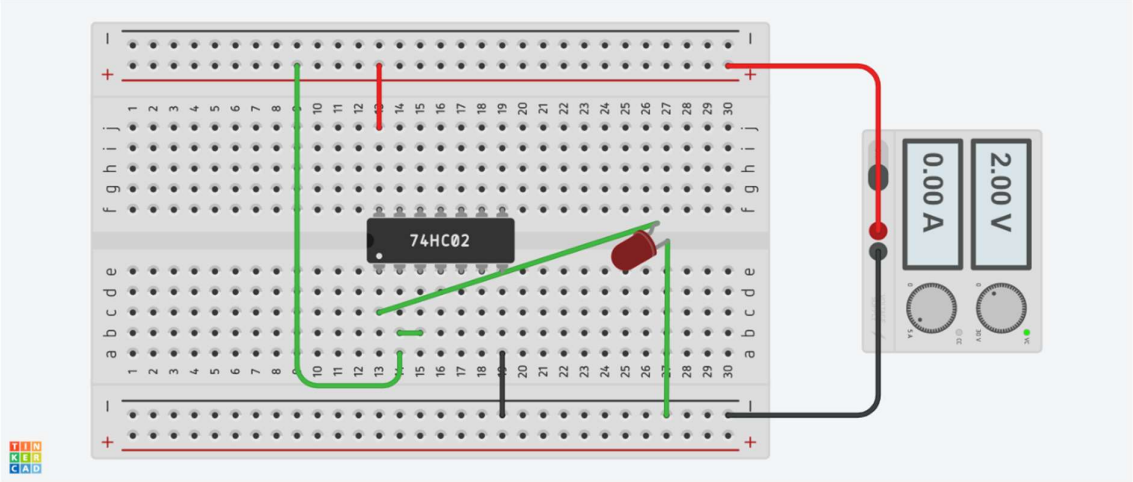
NOR as X-OR gate with inputs 0 and 0

NOR as NOT gate

Tinkercad:



NOR as NOT gate with input 0



NOR as NOT gate with input 1