EXPERIMENT 1

Dtd: 17/12/21

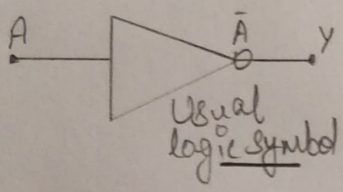
Using NAND, a universal gate, to build other gates.

Submitted by → (Section B)
Anushtan Saxena
S20210010027

(I) AIM → Make a NOT gate using NAND:

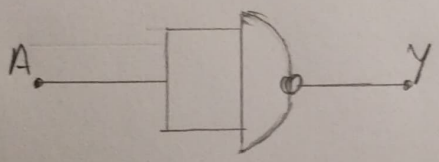
SOFTWARE USED → Logisim.

PROCEDURE →



Usual logic symbol

Boolean expression for a NOT gate ≡ $Y = \bar{A}$



NOT gate using NAND

Verification :-

case(i) → When A = 0, $Y = (\overline{A * A}) = (\overline{0 \cdot 0}) = 1$ which is correct.

case(ii) → When A = 1, $Y = (\overline{1 \cdot 1}) = 0$ which is also correct.

Truth table →

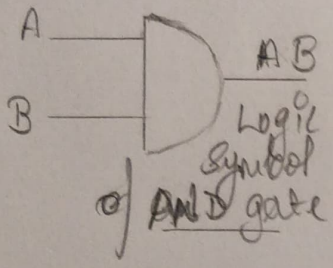
A	Y = Ā
1	0
0	1

CONCLUSION → The easiest gate to produce, this NOT gate is simulated in logisim and is verified with the inputs shown in the truth table.

(II) AIM → Make an AND gate using NAND.

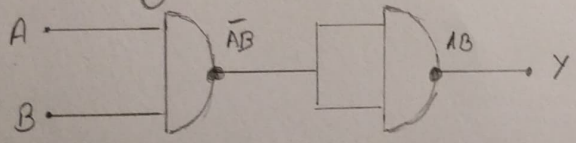
SOFTWARE USED → Logisim

PROCEDURE →



Boolean expression for AND gate ⇒ $Y = AB$

→ AND gate using NAND →



VERIFICATION:

case(i) → When A=0, B=0, Y = 0.0 = 0
Verified through logisim.

case(ii) → When A=1, B=1,
Y = 1.1 = 1
Verified through logisim.

case(iii) → when A=0, B=1
Y = 0.1 = 0
Verified through logisim

Truth table →

A	B	Y
1	1	1
1	0	0
0	1	0
0	0	0

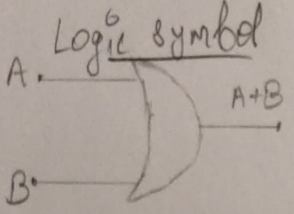
CONCLUSION → The AND gate produced is verified in logisim through simulation with the given truth table.

(III) AIM → Make an OR gate using NAND:

SOFTWARE USED → Logisim

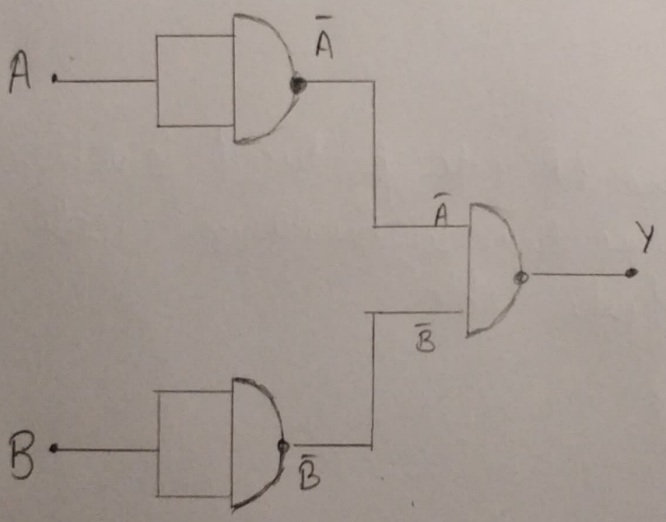
PROCEDURE →

Logic symbol



Boolean expression for OR gate ⇒ $Y = A + B$

→ OR gate using NAND →



Verification →

Truth table:

A	B	Y
1	1	1
1	0	1
0	1	1
0	0	0

case (i) → When A=0, B=0,

$Y = 0+0 = 0$

which is verified through logisim.

case (ii) → When A=1, B=0,

$Y = 1+0 = 1$,

verified through logisim.

case (iii) → When A=1, B=1,

$Y = 1+1 = 1$,

verified through logisim.

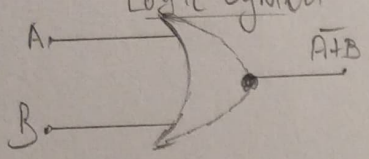
CONCLUSION → The OR gate produced is verified in logisim through simulation with the given truth table.

(IV) AIM → Make a NOR gate using NAND :

SOFTWARE USED → Logisim.

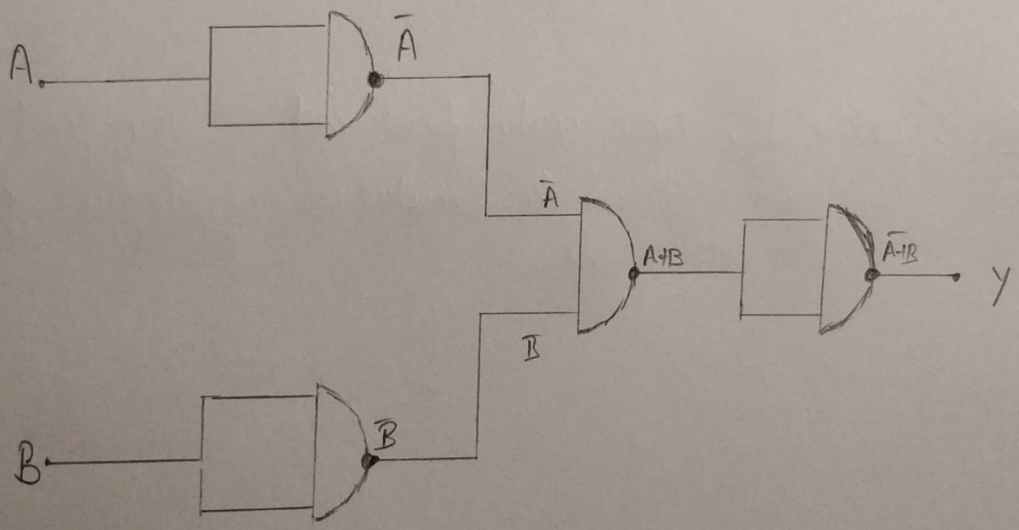
PROCEDURE →

Logic Symbol



Boolean expression for NOR gate ⇒ $Y = \overline{A+B}$

→ NOR gate using NAND →



Verification :-

Truth table :

A	B	Y
1	1	0
1	0	0
0	1	0
0	0	1

Case (i) → When A=0, B=0,

$$Y = \overline{0+0} = 1$$

which is verified through logisim.

Case (ii) → When A=1, B=0,

$$Y = \overline{1+0} = 0,$$

verified through logisim.

Case (iii) → When A=1, B=1,

$$Y = \overline{1+1} = 0,$$

verified through logisim

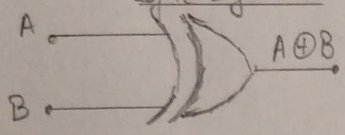
CONCLUSION → The NOR gate produced is verified in logisim through simulation with the given truth table.

(V) AIM → Make an XOR gate using NAND gate :

SOFTWARE USED → Logisim

PROCEDURE →

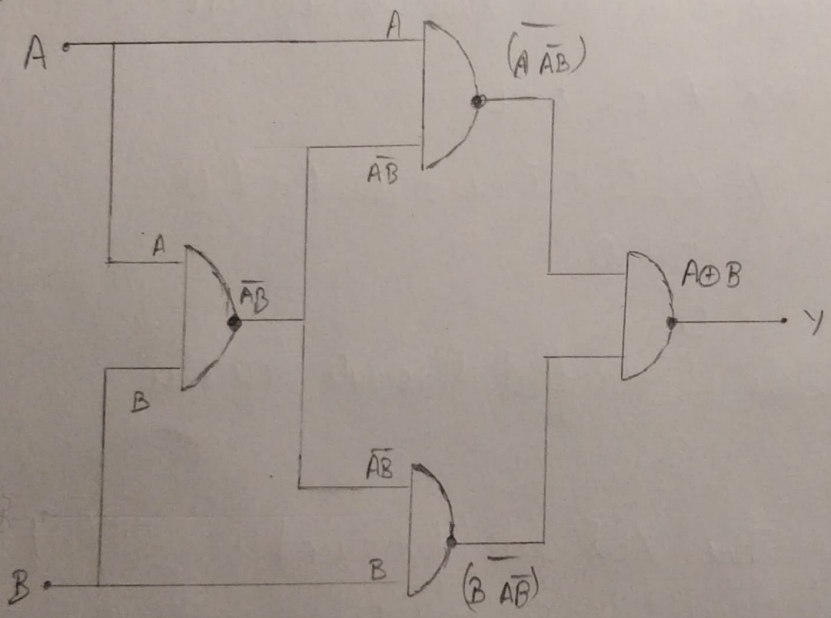
Logic Symbol



Boolean expression for XOR gate ⇒ $Y = A\bar{B} + \bar{A}B$

$= A \oplus B$

→ XOR gate using NAND →



Verification →

Truth table :

A	B	Y
1	1	0
1	0	1
0	1	1
0	0	0

Case(i) → When A = 1, B = 1,

$$Y = 1.\bar{1} + \bar{1}.1 = 1.0 + 0.1$$
$$= 0 + 0 = 0$$

which is verified through logisim.

Case(ii) → When A = 0, B = 1,

$$Y = 0.\bar{1} + \bar{0}.1$$
$$= 0 + 1 = 1$$

which is verified through logisim.

Case(iii) → When A = 0, B = 0,

$$Y = 0.\bar{0} + \bar{0}.0 = 0.1 + 1.0$$
$$= 0 + 0$$
$$= 0$$

which is verified through logisim.

CONCLUSION → The XOR gate produced is verified in logisim through simulation with the given truth table.