Name - Gungan Pandey Student ID - 21712077

Discuss the working of NAND gate with the help of circuit diagram and touth table.

The NdAND gate is the universal gate. It means all the basic gates such as AND, OR, and NOT gates ~~are~~ can be constructed using a NAND gate. The NAND gate is the combination of the NOT-AND gate.

The output state of the NAND gate will be low only when all the inputs are high. Simply, this gate returns the complement result of the AND gate.

The logic or Boolean enpression for the NAND gate is the complement of logical multiplication of inputs denoted by a single dot as

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

The value of Y will be toue when any one of the input is set to 0.

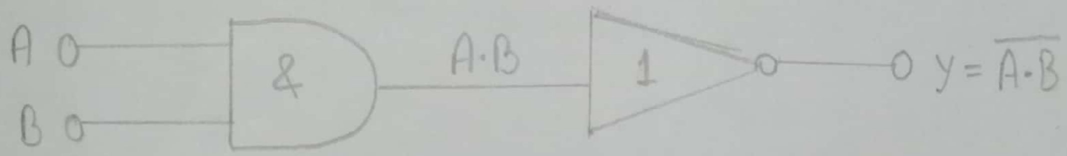
Types of Digital Logic AND Gate

The NAND gate is also classified into three types based on the inputs it takes. These are the following types of AND gate:

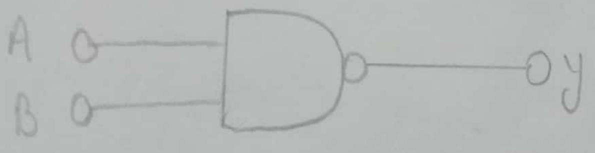
The 2-input NAND Gate

This is the simple formation of the NAND gate. In this type of NAND gate, there are only two input values and an output values. There are $2^2 = 4$ possible combinations of inputs.

Logic Design



2- Input "AND" gate plus a "NOT" gate



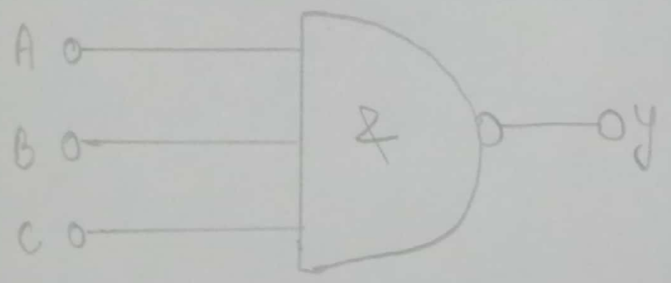
2- Input NAND gate.

Input		Output
A	B	y
0	0	1
0	1	1
1	0	1
1	1	0

The 3-input NAND Gate

The 3-input NAND gate has three inputs.
The Boolean expression of the logic ~~NAD~~ · NAND get is defined as the binary operation dot (.).
The NAND gate can be cascaded together to form any number of individual inputs.
There are $2^3 = 8$ possible combinations of inputs.

Logic Design



3 - Input NAND gate

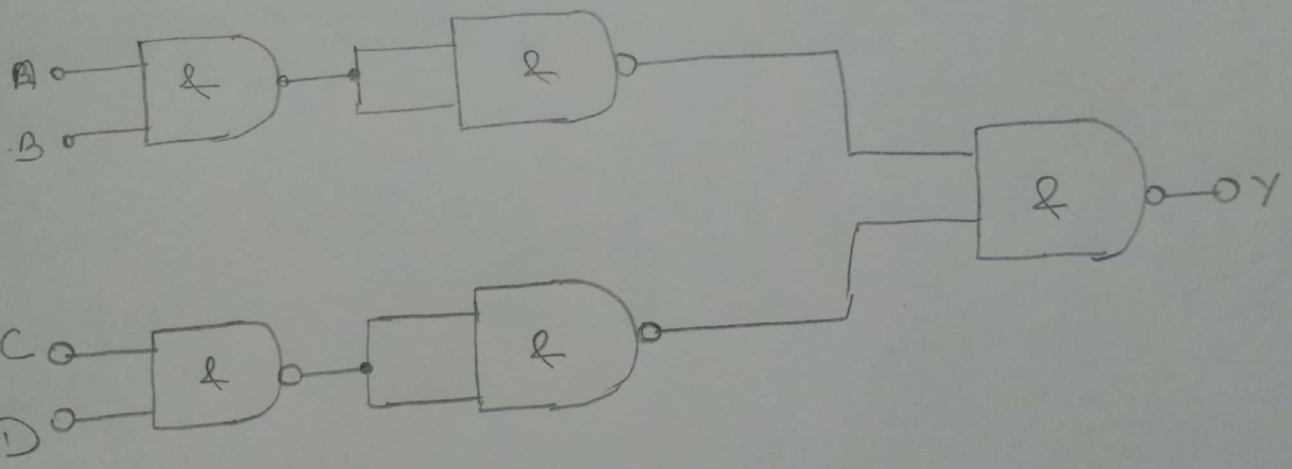
Truth Table.

Input			Output
A	B	C	y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

The Multi-input NAND Gate

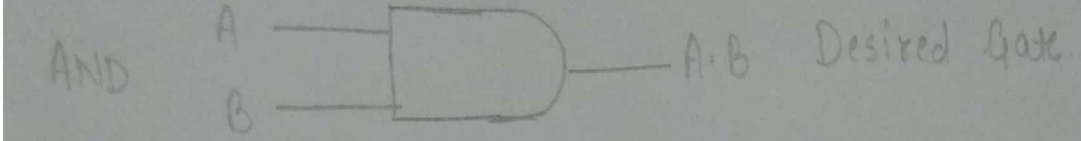
Just like AND, NOT, and OR gate, we can also form n-input NAND gate. If the number of inputs required is odd, any "unused" input can be held high by directly connecting it to the power supply using high "suitable" pull-up resistors. There is following expression of the 4-input NAND gate.

$$Y = ((A \cdot B) \cdot (C \cdot D))'$$

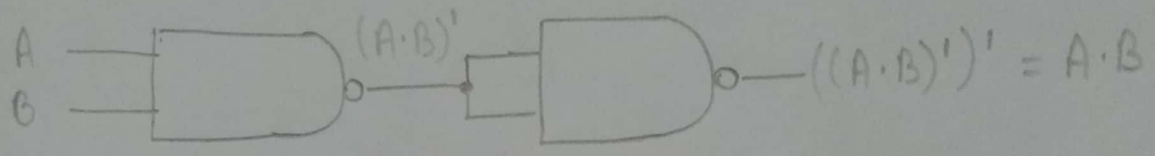


Input						Output
A	B	C	D	E	f	y
0	0	0	0	0	0	1
0	0	0	0	0	1	1
0	0	0	0	1	0	1
0	0	0	0	1	1	1
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
1	1	1	1	0	0	1
1	1	1	1	0	1	1
1	1	1	1	1	0	1
1	1	1	1	1	1	0

AND gate using NAND gate

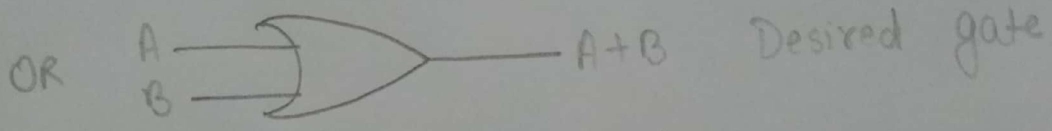


$A \cdot B = ((A \cdot B)')'$ Involution law



The Derivation of the AND Gate

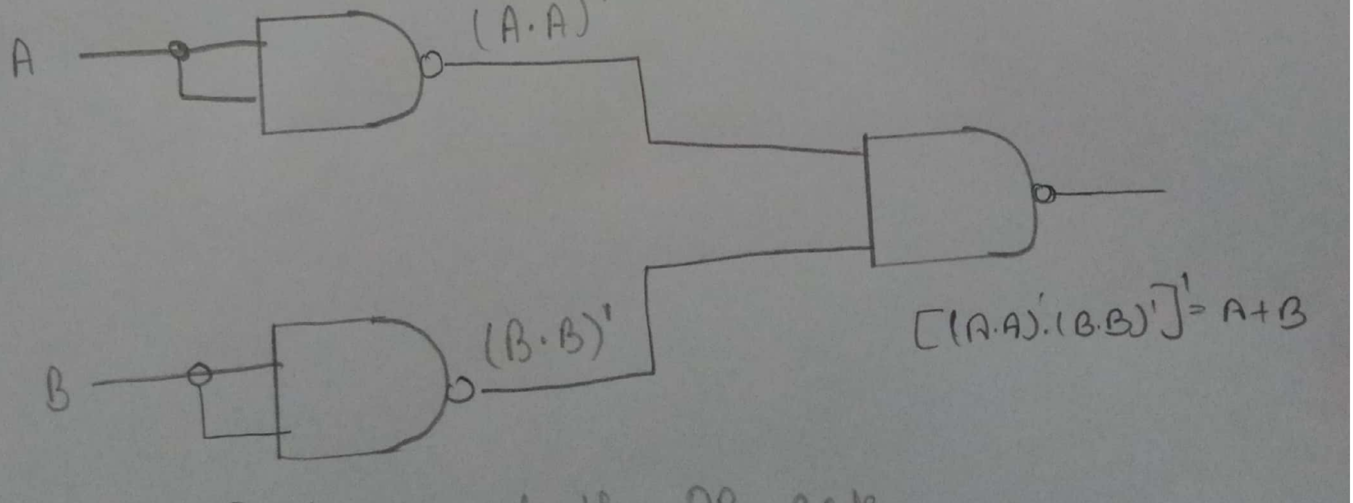
OR gate using NAND gate.



$A + B = [(A+B)]'$ Involution law

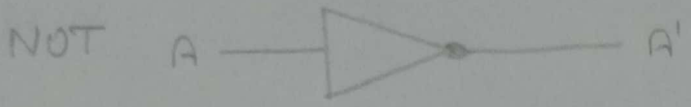
$= [(A+B)]'$ De-Morgan's law

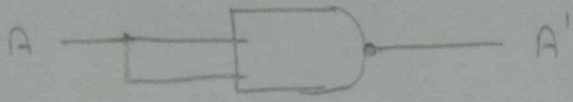
$A+B = [(A.A)'.(B.B)']'$ Idempotency law



The Derivation of the OR gate

NOT gate using NAND gate





$(A.A)' = (A)'$

$$\boxed{\therefore\ (A.A)' = A'}$$