Logic Gates

- Logic gates are the main part of a digital syst.

- They are a block of hardware that produces signals of binary 1 and 0 when input logic requirements are satisfied.

- Each gate has a distinct graphic symbol, and its operation can be described by means of algebric expressions.

7 types of logic gates

AND OR XOR NOT NAND NOR ~~NXOR~~ XNOR

- The relationship b/w the I/O binary variables for each gate can be represented in tubular form by a truth table.

- Each gate has one / two binary input variables designated by A and B and one binary output variable designated by x.

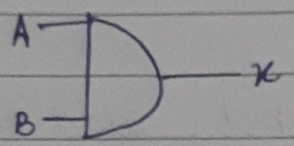
* Logic gates are the basic building blocks of any digital system. It is an electronic ~~system~~ circuit having one or more than one input and only one output. The rel^n b/w the input and output is based on a certain logic.

- Logic gates are small electronic devices that perform a boolean function with 2 inputs & provide 1 output.

* Input devices are set of instructions that are given to the comp. Output is the result that comes after processing the inputs. The main funcⁿ of the input device is to enter the data in the comp.

① **AND Gate** - It is an electronic circuit which gives a high output only if all its inputs are high.

The AND operator is represented by a dot (.)



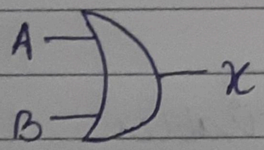
Truth table:

A	B	x
0	0	0
0	1	0
1	0	0
1	1	1

Algebraic funcⁿ $x = AB$

② **OR Gate** - It is an electronic circuit which gives a high output if one or more of its inputs are high.

The OR operator is a plus sign (+)



Truth table:

A	B	x
0	0	0
0	1	1
1	0	1
1	1	1

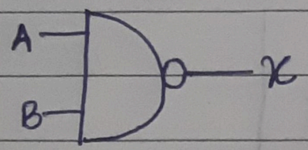
Algebraic funcⁿ $x = A+B$

③ **NOT Gate** - It is an electronic circuit which produces an inverted version of the input as its output.

It is also known as an Inverter.

(4) **NAND Gate** - The NOT-AND gate is ~~an~~ equal to an AND gate followed by a NOT gate.

The Nand gate gives a high output if any of the inputs are low.
The NAND gate is represented by an AND gate with a small circle on the output. The small circle represents inversion.



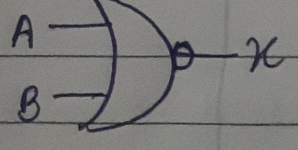
Algebric func^n $x = (AB)'$

Truth table:

A	B	x
0	0	1
0	1	1
1	0	1
1	1	0

(5) **NOR Gate** - The NOT-OR gate which is equal to an OR gate followed by a NOT gate.

The NOR gate gives a low output if any of the inputs are high.
The NOR gate is represented by an OR gate with a small circle on the output. The small circle represents inversion.



Algebric func^n $x = (A+B)'$

Truth table:

A	B	x
0	0	1
0	1	0
1	0	0
1	1	0

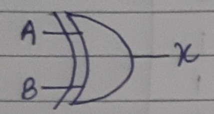
Date :
Page no.

⑥ <u>Exclusive OR / XOR Gate</u> - It is a circuit which will give a high output if one of its inputs is high but not both of them.

The XOR operation is represented by an encircled plus sign $\oplus$

• The output will be high only when both inputs are different

<u>XOR Gate</u>:

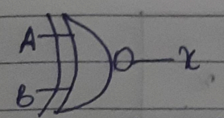


$x = A \oplus B$
or
$x = A'B + AB'$

Truth table:

A	B	x
0	0	0
0	1	1
1	0	1
1	1	0

⑦ <u>Exclusive NOR / XNOR / Equivalence Gate</u> - It is a circuit that does the inverse operation to the XOR gate. It will give a low output if one of its input is high but not both of them.
• The small circle represents inversion.



$x = (A \oplus B)'$
or
$x = A'B' + AB$

Truth table:

A	B	x
0	0	1
0	1	0
1	0	0
1	1	1

☆ NAND as a Universal Gate.

The ~~Nat~~ NAND gate is a special type of logic gate in the digital logic circuit.

⇒ The NAND gate is the universal gate. This means that all the basic gates such as AND, OR and NOT gate can be constructed using a NAND gate. It is the combination of the NOT-AND gate. The output state of the NAND gate will be low only when all the ~~of~~ inputs are high.

Simply, this gate returns the complement result of the AND gate.

The logic or boolean expression of NAND gate is the complement of the logical multiplication of inputs denoted by a single dot

$x = (A . B)'$

The value of x will be true (1) when any one of the input is set to 0.

Types of NAND Gates.

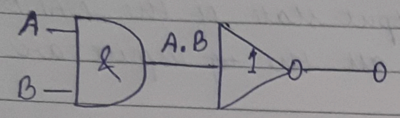
- 2 input NAND gate
- 3 input NAND gate
- Multi input NAND gate

Date :
Page no.

① The 2 input NAND Gate.

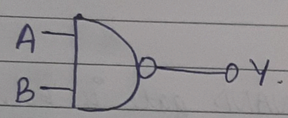
This is the simple formation of NAND gate. In this, there are only 2 input values and one output value.

There are $2^2 = 4$ possible combinations of inputs.



$Y = \overline{A.B}$

Or



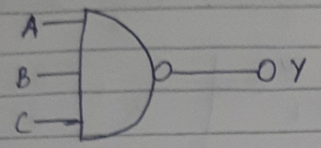
Truth table:

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

② The 3 input NAND Gate.

This gate has 3 inputs. The boolean expression of the logic NAND gate is defined by the binary operation dot (.).

There are $2^3 = 8$ possible combinations of input



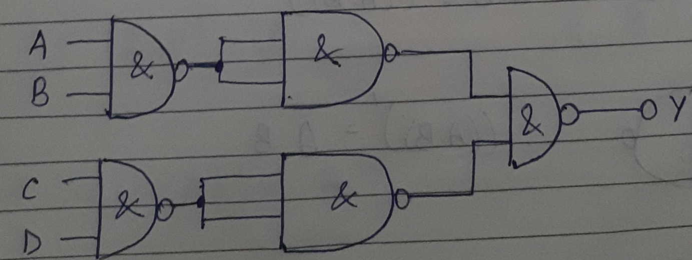
A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

③ The Multi-input NAND Gate.

We can form an 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.

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

Y = (A NAND B) NAND (C NAND D)



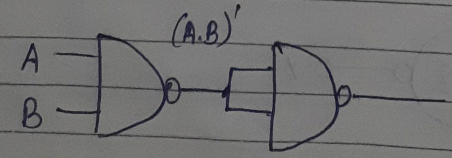
Date :
Page no.

Truth table :

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
0	0	0	1	0	0	1
0	0	0	1	0	1	1
0	0	0	1	1	0	1
0	0	0	1	1	1	1
0	0	1	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	1	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

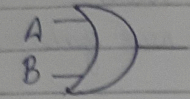
★ AND gate using NAND Gate.

Involution law : $A.B = ((A.B)')'$



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

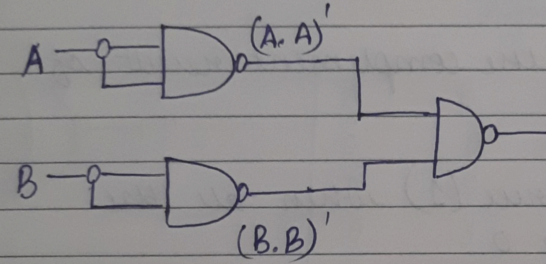
☆ OR gate using NAND gate.



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

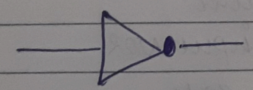
$= [(A+B)]'$ De-Morgan's law

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

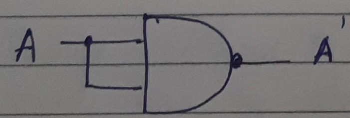


$[(A.A)' . (B.B)']' = A+B$.

☆ NOT gate using NAND gate.



(Desired Gate)



Idempotency law.

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

$\therefore (A.A)' = A'$.

★ NOR Gate as a Universal Gate.

The NOR gate is also a universal gate which means we can also form all basic gates using the NOR gate.

The NOR gate is a combination of NOT-OR gate.

The output state of the NOR gate will be high only when all of the inputs are low.

Simply, this gate returns the complement result of the OR gate.

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

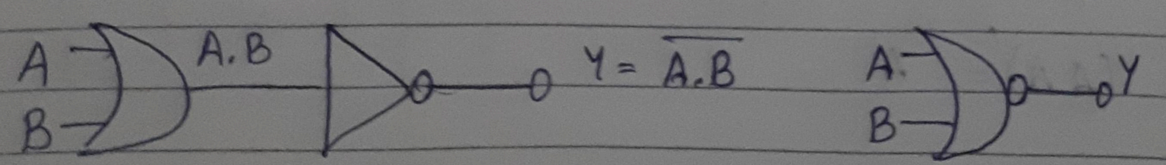
The value of Y will be true (1) when all the inputs are ~~stated~~ set to 0.

Types of NOR Gates

- 2 input NOR gate
- 3 input NOR gate
- Multi input NOR gate.

① The 2 input NOR gate.

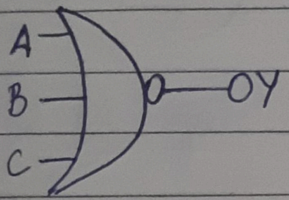
There are only 2 input values and 1 output value. There are $2^2 = 4$ possible combinations of inputs.



Truth table:

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

(2) The 3 input NOR gate - This gate has 3 inputs. There are $2^3 = 8$ possible combinations of inputs.



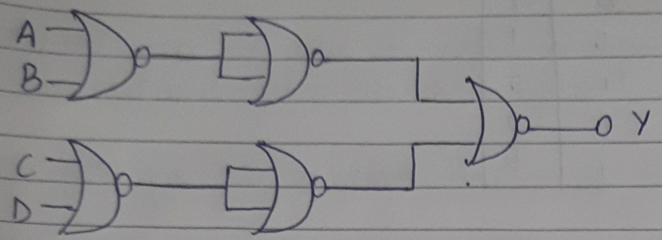
A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

(3) The Multi-input NOR Gate - We can also form the n-input NOR Gate. If the number of inputs required is odd, any 'unused' input can be held low by directly connecting it to the power supply using low "suitable" pull resistors.

$$Y = ((A+B)' + (C+D))'$$

Y = A NOR B NOR C NOR D

Date :
Page no.



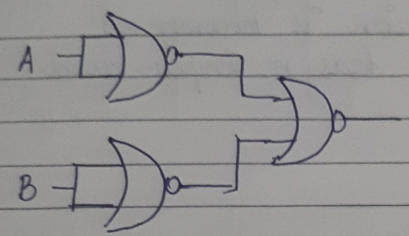
Truth table:

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

★ AND gate using NOR gate.

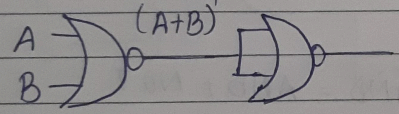
$A.B = [(A.B)']' = [A'+B']'$

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



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

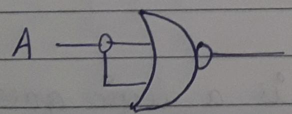
★ OR gate using NOR gate



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

★ NOT gate using NOR gate.

Idempotency law $(A+A=A)$



A' $\quad (A+A)' = (A)'$

★ Logic gates applications

Logic gates are used in microcontrollers, microprocessors, electronic and electrical project circuits, and embedded system applications.

These are the important digital devices, mainly based on the boolean function

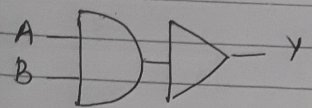
Date :
Page no.

OR ; AND ; NOT — Basic gates

NAND ; NOR — Universal gates

XOR, XNOR — Process IOP gates

NAND gate: A not-And operation is known as NAND operation. It has n input and one output



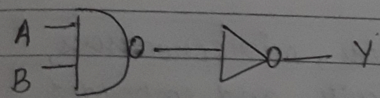
$Y = (AB)' = A' + B'$

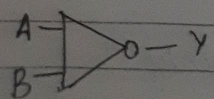
TT: A	B	$(AB)'$
0	0	1
0	1	1
1	0	1
1	1	0

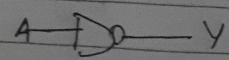
NAND = AND + NOT

~~NO~~

Universal Gate: A universal gate is a logic gate using which we make other types of logic gates.
NAND and NOR are the universal gates.

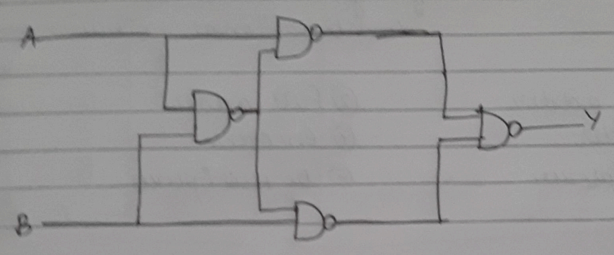




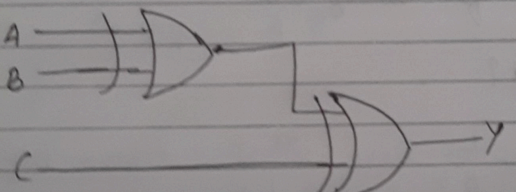


Date :
Page no.

$Y = A \oplus B = A.B + A'.B'$

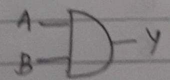


XOR gate with 3 inputs



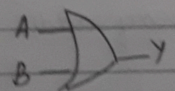
7/03/23

AND



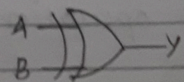
$Y = A.B$

OR



$Y = A+B$

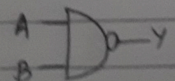
XOR



$Y = A'.B + A.B'$

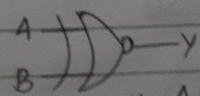
$Y = A \oplus B$

NAND



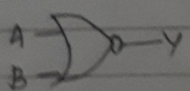
$Y = (A.B)'$

XNOR



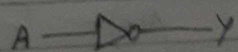
$Y = A.B + A'B'$

NOR



$Y = (A+B)'$

NOT



$Y = A'$

An electronic circuit in which the output depends upon the present combination of the input.

Or

Combinational circuits — combination of 2 or more logic gates is called combinational circuits.

6 types
1. Half adder
2. Full adder
3. Decoder
4. Encoder
5. Multiplexer
6. De multiplexer

---

1. **Half Adder** – It is used to add 2 bits/ digits. It has 2 inputs and 2 outputs.
The first output is for sum (S)
The second output is for carry (C).

Q How 2 bits are added?

0 + 0 = 0 0	0 + 1 = 0 1	1 + 0 = 0 1	1 + 1 = 1 0
C S	C S	C S	C S.

All four operations are done by the half adder (H.A)

It is made up of 2 logic gates

1. XOR Gate
2. AND Gate.