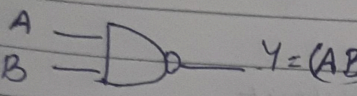
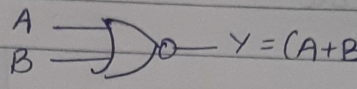
Bittoo

Date :
Page no.

What is universal gate?
A logic gate which can implement any Boolean func" without the need to use any other type of logic gate.
NOR gate & NAND gate are universal gates

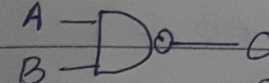



Logic gates are the basic building blocks of any digital system. It is an electronic circuit having one or more than one input and only one output.

The relationship b/w the input & output is based on a certain logic.

NAND Gate.
- Combination of NOT and AND gate.
- Produces High (1) when any ~~one~~ input is low (0).
- Produces low (0) when all inputs are High (1).
- Known as universal gate.




$C = \overline{A.B}$

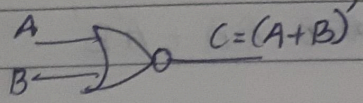
Truth Table

| A | B | (AB)' |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Date :
Page no.

## NOR Gate

- Combination of NOT and OR gate
- Produces High (1) when all inputs are low (0)
- Produces low (0) ~~o/p~~ when any input is low (0)
- Known as universal gate

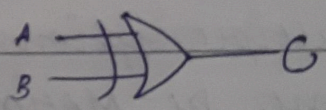



Truth Table

| A | B | Y = (A+B)' |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

## XOR Gate

- stands for exclusive OR gate
- It is a hybrid gate
- can have only 2 inputs
- used as a half-adder.




$C = A \oplus B$

$\Rightarrow C = A'B + AB'$

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

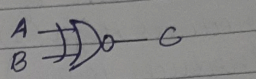
Truth Table

| A | B | $A \oplus B$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Date :
Page no.

## XNOR Gate

- stands for the exclusive NOR Gate
- It is a hybrid gate
- can have only 2 inputs at a time
- used as a sign bit comparator




$C = (A \oplus B)'$

$\Rightarrow AB + A'B'$

Truth Table

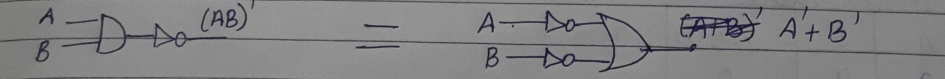
| A | B | C |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

## Need of a universal gate

- To minimize the number of gates in designing of a circuit.
- To have multiple options.
- To implement De-Morgan's principle.

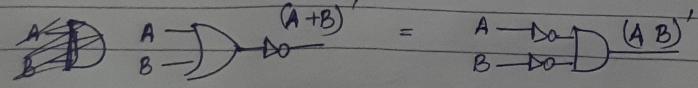
## De-Morgan's principle of universal gate.

- A NOR gate is equivalent to a bubbled AND gate.




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

- A NAND gate is equivalent to a bubbled OR gate

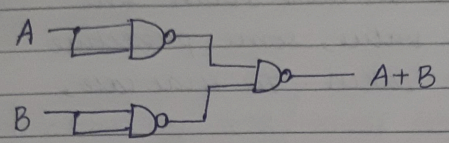



$(A+B)' = A'B'$

Bittoo

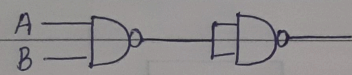
Date :
Page no.

Q. Construct a OR gate using NAND Gates



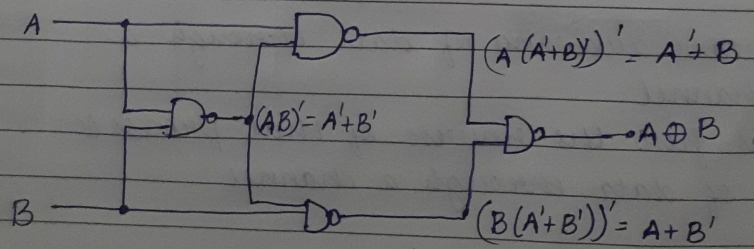

| A | B | Output |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Q. Construct AND Gate using NAND Gates




* Significance of Universal Gates
- They are called 'Universal Gates' because- they can realize all the binary operations
- All the basic logic gates can be derived from them.
- They have the following properties:
  a) They are not associative in nature
  b) They are commutative in nature.

Q. Construct a XOR Gate using NAND Gates



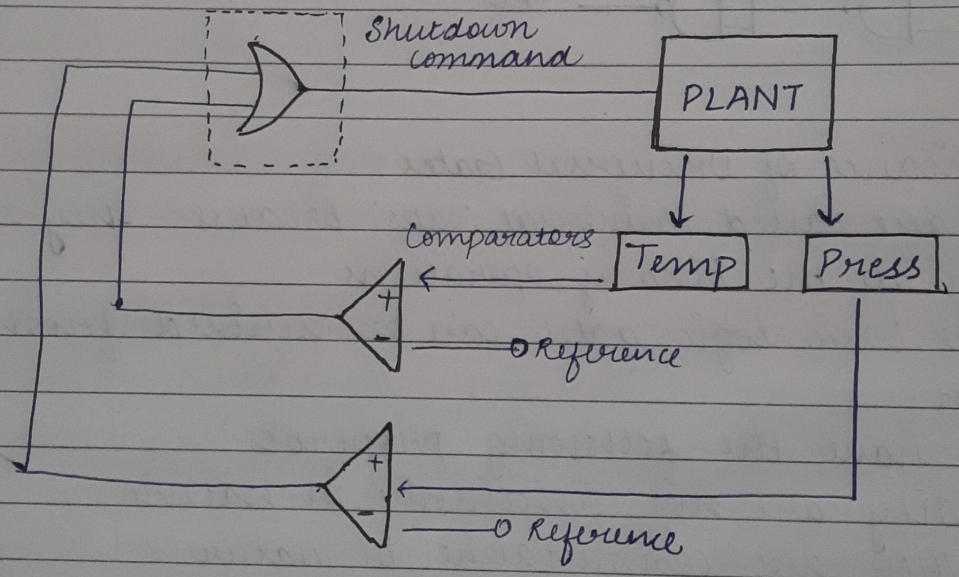
Bittoo TRADE MARK REGD.

Date :
Page no.

## Application of OR Gate in ~~two~~ Industrial Safety

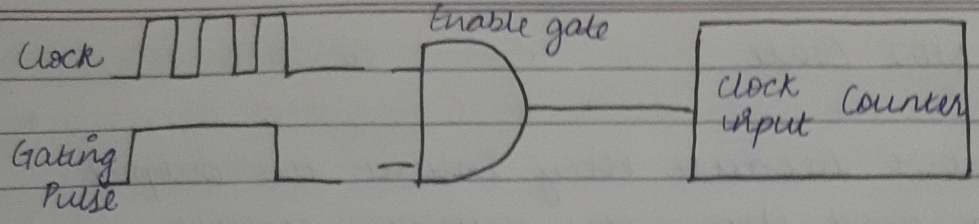
- If in an industrial plant one or more than one parameter exceeds the safe value, some protective measures are needed to be taken. In that case, the OR Gate is used.

- OR gate here is used to detect exceed of temp or press and produce command signal for the system to take required actions.




## Application of AND Gate

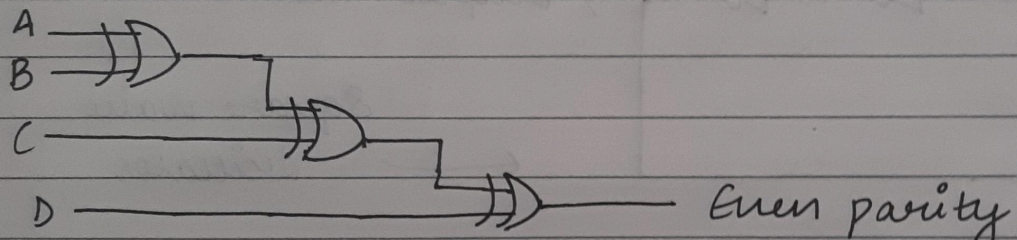
- Enable gate means allowance of data through a channel
- Inhibit gate is just the reverse of that process ie disallowance of data through a channel
- The enabling operation is shown here:



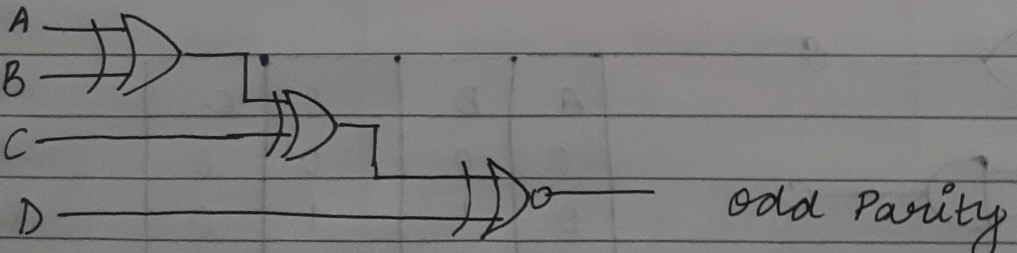

## Application of XOR / XNOR Gate

- These type of logic gates are used in generation of parity generation and checking units.
- The 2 diagrams show the even and odd parity generator circuits respectively for 4 data.

XOR even ~~parity~~ parity



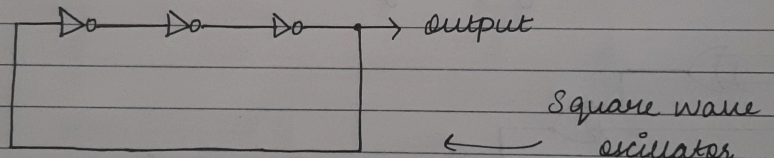

XNOR odd parity



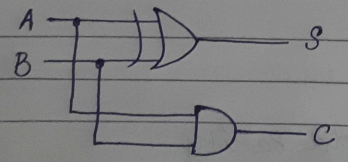
Date :
Page no.

## Application of NOT Gate

- A.K.A. ~~as~~ inverter because they invert the output given to them and show the reverse result.
- Now the CMOS inverters are commonly used to build square wave oscillators which are used for generating clock signals.
- The advantage of using this is they consume low power and their interfacing is very easy compared to ~~their~~ other logic gates.




Q construct a circuit to add 2 digits




| A | B | S | C |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |