

## ELECTRONIC PRINCIPLES AND DEVICES/UE24EC141A

### UNIT-3 Digital Electronics

#### UNIT-3

#### Digital Electronics

Logic gate: It is a basic building block of a circuit, used to make a large no. of electronic circuits.

- \* A circuit with any no. of inputs but only one o/p.
- \* Input and o/p can take values of 0 or 1 only.

#### 2 types of Gates

Basic gate: 3 types (AND, OR, NOT)

Derived gates :- (XOR, XNOR, NAND, NOR)

NAND & NOR are called universal gates because using these gates all other gates can be made using these gates.

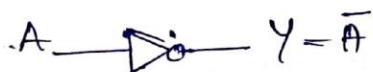
#### Truth Table:-

A table that gives an o/p for all possible i/p combinations.

Logic 0 = 0V (Low)

Logic 1 = 5V (High)

NOT gate: - A logic gate with only 1 i/p and 1 o/p. The o/p is inversely proportional to i/p.



Bar  $\rightarrow$  NOT operator

## Truth Table

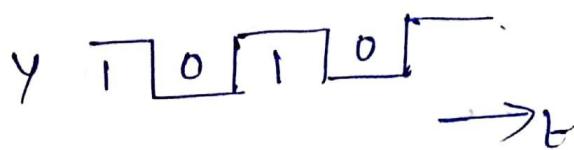
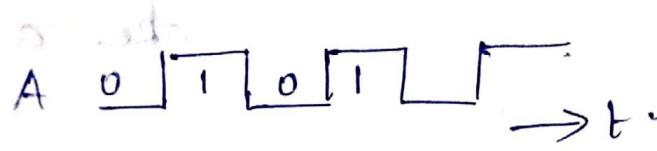
| <u>Input</u> | <u>Output</u> |
|--------------|---------------|
|--------------|---------------|

|   |   |
|---|---|
| A | Y |
|---|---|

|   |   |
|---|---|
| 0 | 1 |
|---|---|

|   |    |
|---|----|
| 1 | 0. |
|---|----|

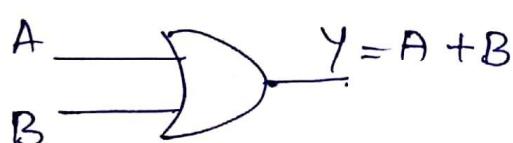
## Timing diagram



## OR gate

A logic circuit with 2 or more o/p's but 1 op.

## Symbol



$\rightarrow$  OR operator

If its o/p is high when any one of i/p is high.

## Truth Table

## Timing diagram

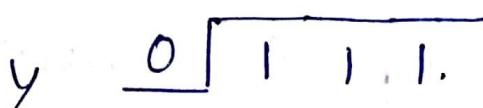
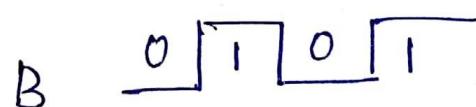
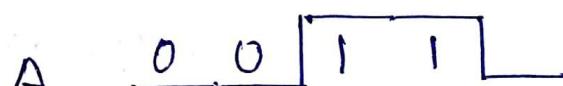
| <u>i/p</u> | <u>o/p</u> |
|------------|------------|
| A · B      | Y.         |

|     |   |
|-----|---|
| 0 0 | 0 |
|-----|---|

|      |   |
|------|---|
| 0 1. | 1 |
|------|---|

|      |   |
|------|---|
| 1. 0 | 1 |
|------|---|

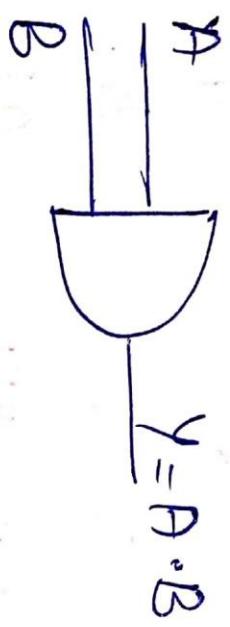
|     |    |
|-----|----|
| 1 1 | 1. |
|-----|----|



## AND gate

A logic circuit whose op is 1 when all the inputs are high.

Logic symbol



Truth Table

| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

00, 01, 10 → AND

operator

A    0    0    1    1

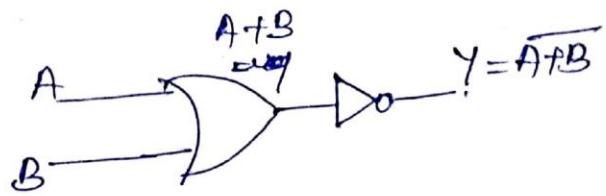
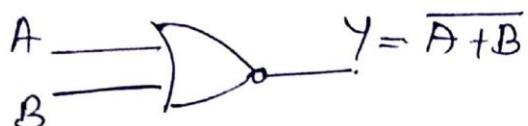
B    0    1    0    1

Y    0    0    0    1

Derived Gates [combination of two or more gates]

1. NOR = OR + NOT

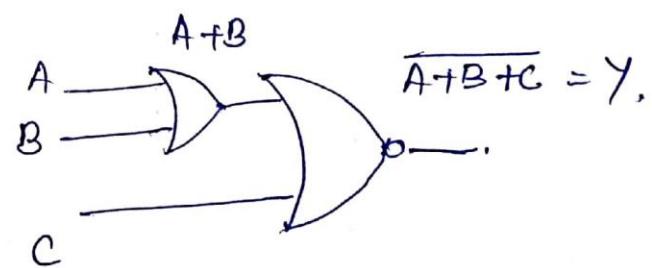
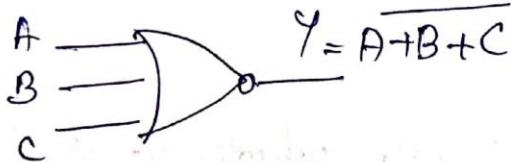
Logic symbol:



Truth Table

| A | B | $A+B$ | $\overline{A+B} = Y$ |
|---|---|-------|----------------------|
| 0 | 0 | 0     | 1                    |
| 0 | 1 | 1     | 0                    |
| 1 | 0 | 1     | 0                    |
| 1 | 1 | 1     | 0                    |

X. 3 o/p NOR gate.



XOR gate:

$$Y = A \oplus B. \quad (\oplus \rightarrow \text{XOR operation})$$

If both the o/p's are high or low, the o/p is low.

# Truth Table for ( $2^1$ ) $p$ )

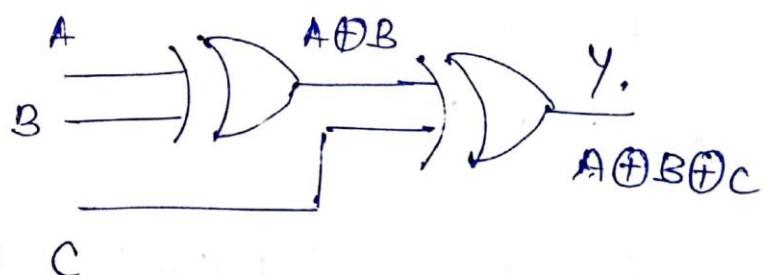
| $i/p$ | $o/p$ | $y_1 = A \oplus B$ |
|-------|-------|--------------------|
| 0 0   | 0     |                    |
| 0 1   | 1     |                    |
| 1 0   | 1     |                    |
| 1 1   | 0     |                    |

$$Y = A \oplus B = \bar{A}B + A\bar{B}$$

+  $3^1$   $i/p - A \oplus B \oplus C = Y.$

$i/p's$        $o/p$

| A | B | C | $o/p$ |
|---|---|---|-------|
| 0 | 0 | 0 | 0     |
| 0 | 0 | 1 | 1     |
| 0 | 1 | 0 | 1     |
| 0 | 1 | 1 | 0     |
| 1 | 0 | 0 | 1     |
| 1 | 0 | 1 | 0     |
| 1 | 1 | 0 | 0     |
| 1 | 1 | 1 | 1     |



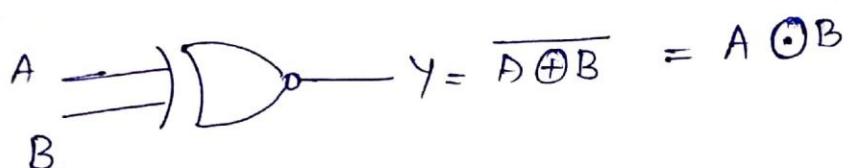
Logic gate whose o/p is 1  
when odd no of  $i/p$ 's are 1

- ODD FUNCTION GATE

## ELECTRONIC PRINCIPLES AND DEVICES/UE24EC141A

### UNIT-3 Digital Electronics

→ XNOR gate ( $\text{XOR} + \text{NOT}$ )



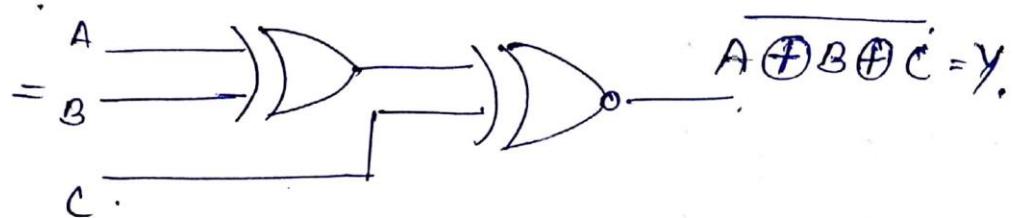
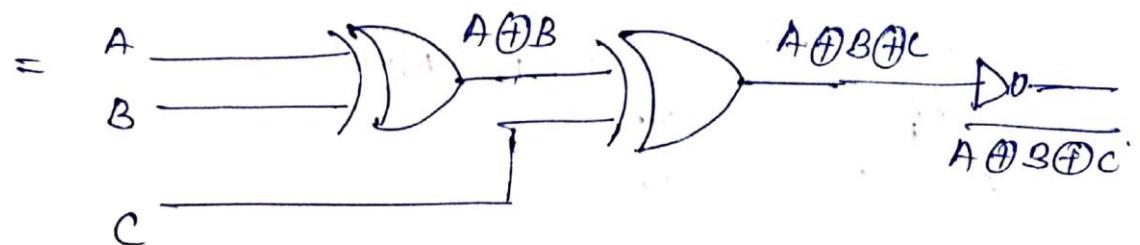
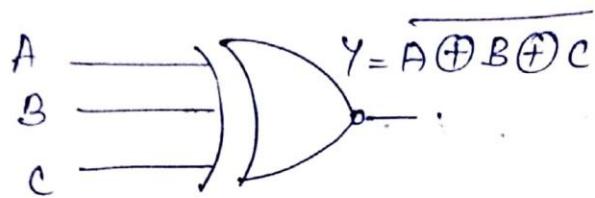
Truth Table. (2 i/p XNOR)

| i/p | O/P |   |
|-----|-----|---|
| A   | B   | Y |
| 0   | 0   | 1 |
| 0   | 1   | 0 |
| 1   | 0   | 0 |
| 1   | 1   | 1 |

3 i/p XNOR gate.

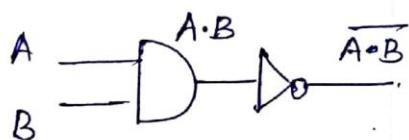
X

| i/p |   |   | O/P                                  |
|-----|---|---|--------------------------------------|
| A   | B | C | $Y = \overline{A \oplus B \oplus C}$ |
| 0   | 0 | 0 | 1                                    |
| 0   | 0 | 1 | 0                                    |
| 0   | 1 | 0 | 0                                    |
| 0   | 1 | 1 | 1                                    |
| 1   | 0 | 0 | 0                                    |
| 1   | 0 | 1 | 1                                    |
| 1   | 1 | 0 | 1                                    |

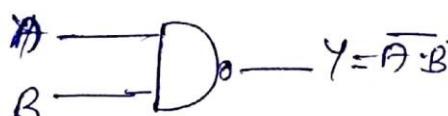


NAND gate :- [ AND + NOT ]

If both the o/p's are high o/p is low; otherwise o/p is high.



Logic symbol



Truth table

| a/p | o/p |   |
|-----|-----|---|
| A   | B   | Y |
| 0   | 0   | 1 |
| 0   | 1   | 1 |
| 1   | 0   | 1 |
| 1   | 1   | 0 |

Boolean Algebra. [Algebra of logic invented by George Boole]

It deals with variables that take only 2 values  
0 and 1.

Laws of Boolean Algebra.

1. Identity Law :-

$$A + 0 = A$$

$$A + 1 = 1$$

$$A \cdot 0 = 0$$

$$A \cdot 1 = A$$

2. Idempotent Law :-

$$A + A = A$$

$$A \cdot A = A$$

3. Complement Law :-

$$A + \bar{A} = 1$$

$$A \cdot \bar{A} = 0$$

4. Involution law :- (Inverse law)

$$\bar{\bar{A}} = A$$

Associative law :-

$$A + (B + C) = (A + B) + C$$

$$A \cdot (B \cdot C) = (A \cdot B) \cdot C.$$

6. Commutative law:-

$$A+B = B+A$$

$$A \cdot B = B \cdot A$$

7. Distributive Law:-

$$A \cdot (B+C) = A \cdot B + A \cdot C$$

$$A+(B \cdot C) = (A+B) \cdot (A+C)$$

8.  $A+A \cdot B = A(1+B) = A$

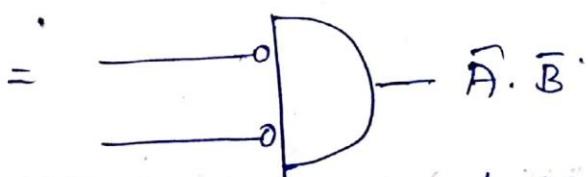
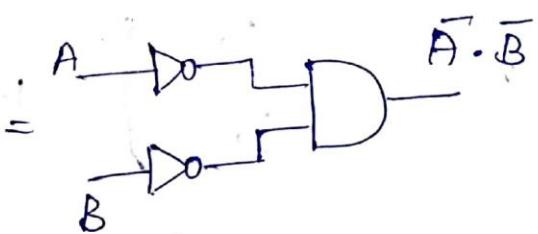
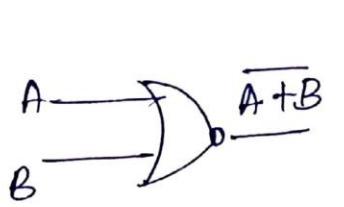
9.  $A+\bar{A}B = (A+B)(A+\bar{A}) = \underline{\underline{A+B}}$

DE MORGAN'S LAW

1.  $\overline{A+B} = \bar{A} \cdot \bar{B}$

The complement of OR sum is equal to the AND product of the complements.

Circuit Equivalent



A NOR gate is equivalent to a bubbled AND gate.

Proof.

| A | B | $\bar{A} + \bar{B}$ | $\bar{A}$ | $\bar{B}$ | $\bar{A} \cdot \bar{B}$ |
|---|---|---------------------|-----------|-----------|-------------------------|
| 0 | 0 | 1                   | 1         | 1         | 1                       |
| 0 | 1 | 0                   | 1         | 0         | 0                       |
| 1 | 0 | 0                   | 0         | 1         | 0                       |
| 1 | 1 | 0                   | 0         | 0         | 0                       |

$\uparrow$  both  
 $\downarrow$  are identical

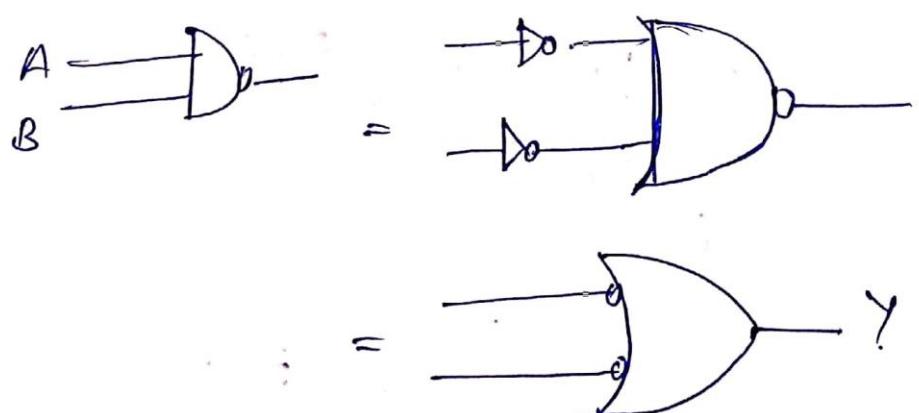
$$\therefore \bar{A} + \bar{B} = \bar{A} \cdot \bar{B}$$

\*  $\bar{A} + \bar{B} + \bar{C} = \bar{A} \cdot \bar{B} \cdot \bar{C}$

$$\bar{A} \cdot \bar{B} = \bar{A} + \bar{B}$$

The complement of AND product is equal to the OR sum of the components.

Circuit Equivalence :-



A NAND gate is equivalent to bubbled OR gate.



Proof.

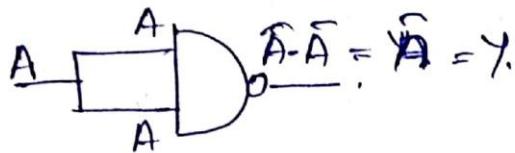
| A | B | $A \cdot B$ | $\overline{A \cdot B}$ | $\overline{A}$ | $\overline{B}$ | $\overline{A} + \overline{B}$ |
|---|---|-------------|------------------------|----------------|----------------|-------------------------------|
| 0 | 0 | 0           | 1                      | 1              | 1              | 1                             |
| 0 | 1 | 0           | 1                      | 1              | 0              | 1                             |
| 1 | 0 | 0           | 1                      | 0              | 1              | 0                             |
| 1 | 1 | 1           | 0                      | 0              | 0              | 1                             |

\* for 3 op's  $\overline{A \cdot B \cdot C} = \overline{A} + \overline{B} + \overline{C}$

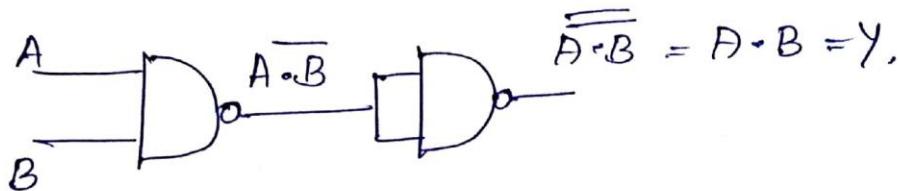
## UNIVERSAL GATES

NAND & NOR :- because all other gates can be obtained using these gates.

1. NAND as NOT  $\Rightarrow Y = \bar{A}$

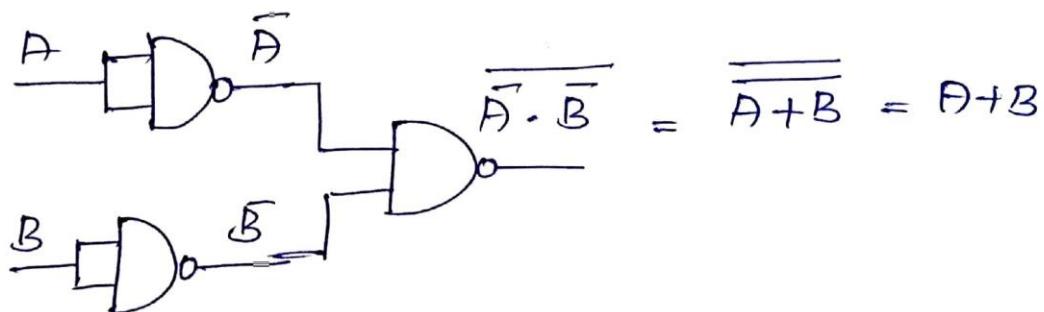


2. NAND as AND  $\Rightarrow Y = A \cdot B$

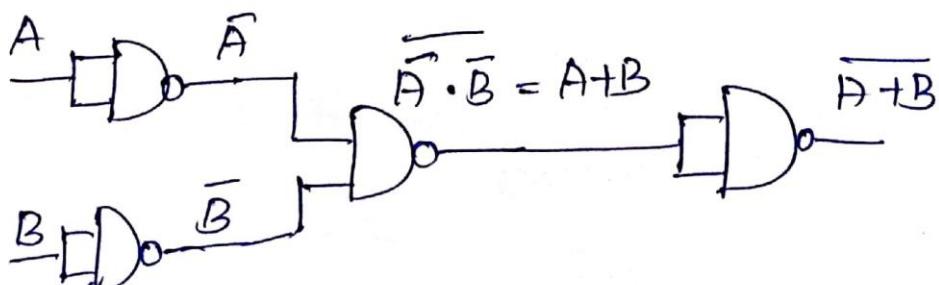


i.e.  $\text{NAND} + \text{NOT} = \text{AND}$ .

3. NAND as OR  $\Rightarrow Y = A + B$



4. NAND as NOR  $\Rightarrow Y = \bar{A} + \bar{B}$

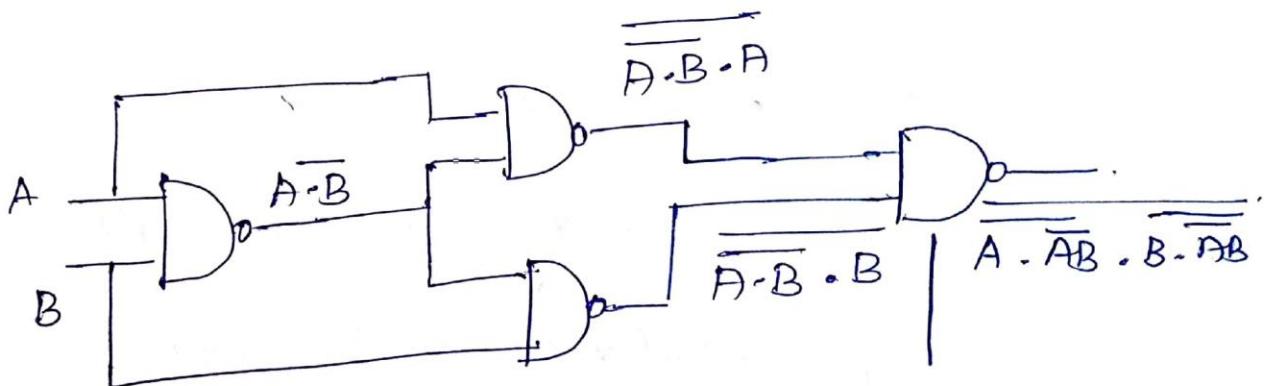


5. NAND as XOR

$$Y = A \oplus B = \bar{A}B + A\bar{B}$$

Proof

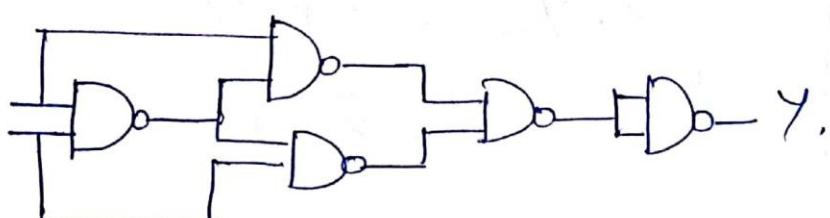
$$\begin{aligned}
 \bar{A}B + A\bar{B} &= \cancel{\bar{A}\bar{B}+B(\bar{A}+\bar{B})} + A(\bar{A}+\bar{B}) \\
 &= [B \cdot \cancel{\bar{A} \cdot B}] + [A \cdot \cancel{A \cdot \bar{B}}] \\
 &= [\cancel{B \cdot \bar{A} \cdot B}] + [\cancel{A \cdot \bar{A} \cdot \bar{B}}] \\
 &= [\cancel{\bar{B} \cdot A \cdot B}] + [\cancel{\bar{A} \cdot \bar{A} \cdot \bar{B}}] \\
 &= [\cancel{\bar{B} + A \cdot B}] + [\cancel{\bar{A} + A \cdot B}] \\
 &= (\bar{B} + AB) \cdot (\bar{A} + AB)
 \end{aligned}$$



6. NAND as XNOR

$$Y = \overline{A \oplus B}$$

$$\begin{aligned}
 Y = \overline{\bar{A}B + A\bar{B}} &= \overline{\bar{A}B \cdot A\bar{B}} = (A + \bar{B}) \cdot (\bar{A} + B) \\
 &= \underline{AB + \bar{A}\bar{B}}
 \end{aligned}$$



**ELECTRONIC PRINCIPLES AND DEVICES/UE24EC141A**  
**UNIT-3 Digital Electronics**

Conversion of Boolean Exp to NAND circuit :-

Method 1 :-

Step 1: Draw the circuit for boolean expression using basic gates.

Step 2: Replace each gate by its NAND equivalent

Step 3: Eliminate double inversions on a single line

$$Q \quad A = \overline{x\bar{y} + x\bar{y}\bar{z}}$$

$$A = \overline{\overline{x\bar{y}} + \overline{x\bar{y}\bar{z}}} = \overline{\overline{x\bar{y}}} \cdot \overline{\overline{x\bar{y}\bar{z}}}$$

$$= (\bar{x} + \bar{\bar{y}}) \cdot (\bar{\bar{x}} + \bar{\bar{y}} + \bar{\bar{z}})$$

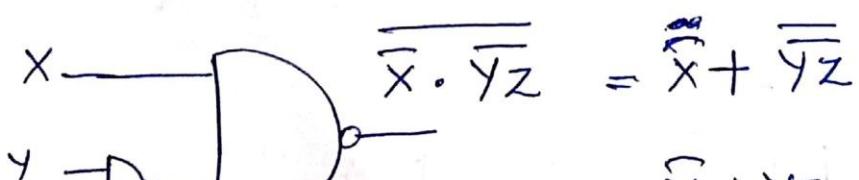
$$= (\bar{x} + y) \cdot (\bar{x} + \bar{y} + z)$$

$$= (\bar{x} \cdot \bar{x}) + (\bar{x} \cdot \bar{y}) + (\bar{x} \cdot z) + (\bar{x} \cdot y) + (y \cdot \bar{y}) + (y \cdot z)$$

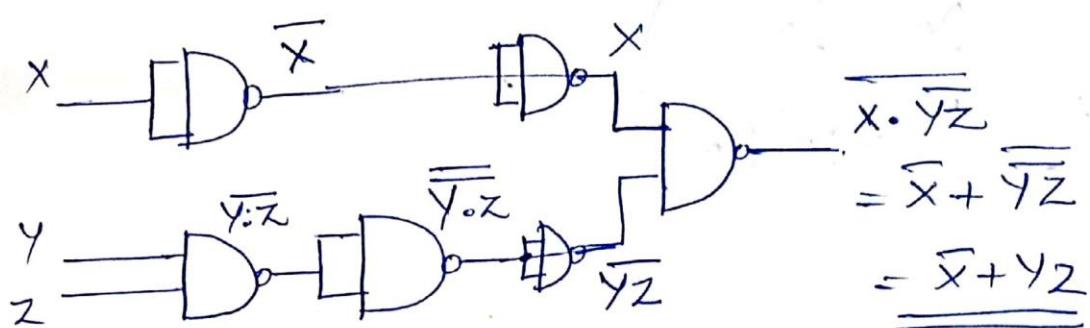
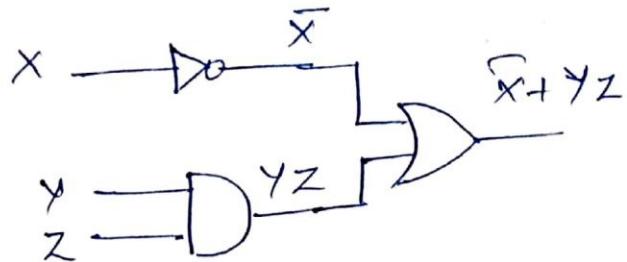
$$= \bar{x} + (\bar{x} \cdot \bar{y}) + (\bar{x} \cdot z) + (\bar{x} \cdot y) + 0 + yz$$

$$= \bar{x}(1+z) + \bar{x}(y+\bar{y}) + yz$$

$$= \underline{\bar{x} + 0 + yz}$$



### Method



### Method 2 :

Step 1: Draw the circuit for boolean exp using basic gates.

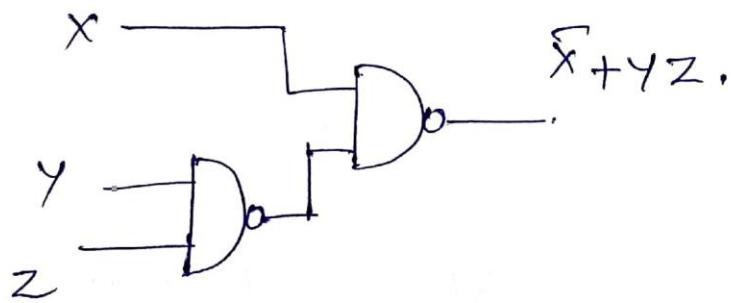
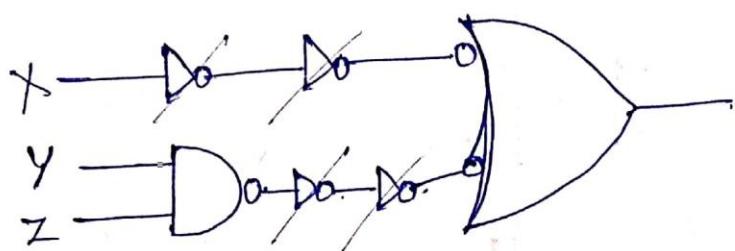
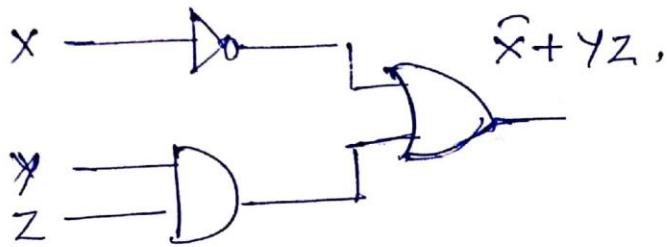
Step 2: Draw bubbles at the o/p of each AND gate and at each of the o/p's of the OR gate.

~~Add~~ Add a NOT gate to each line that received a bubble. (equal to no of bubbles)

Step 3: Eliminate double inversions on a single line.

Replace the bubbled OR gate by a NAND gate.

**ELECTRONIC PRINCIPLES AND DEVICES/UE24EC141A**  
**UNIT-3 Digital Electronics**



## Canonical Sum of Products (SOP)

- \* Each product term contains all literals.
- \* These product terms are called min-terms
- \* This is also called as sum of min terms form.

Example.  $\Rightarrow$  Truth Table

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

|   |   |   |   |
|---|---|---|---|
| 0 | 1 | 1 | 1 |
|---|---|---|---|

|   |   |   |   |
|---|---|---|---|
| 1 | 0 | 0 | 0 |
|---|---|---|---|

Fundamental Product  
for table truth table

$$\bar{A}BC \text{ (3)}$$

$$A\bar{B}C \text{ (5)}$$

$$AB\bar{C} \text{ (6)}$$

$$ABC \text{ (7)}$$

Steps:-

1. Locate each op '1' in truth table and write fundamental products.
2. OR the fundamental products.

$$Y = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

**ELECTRONIC PRINCIPLES AND DEVICES/UE24EC141A**  
**UNIT-3 Digital Electronics**

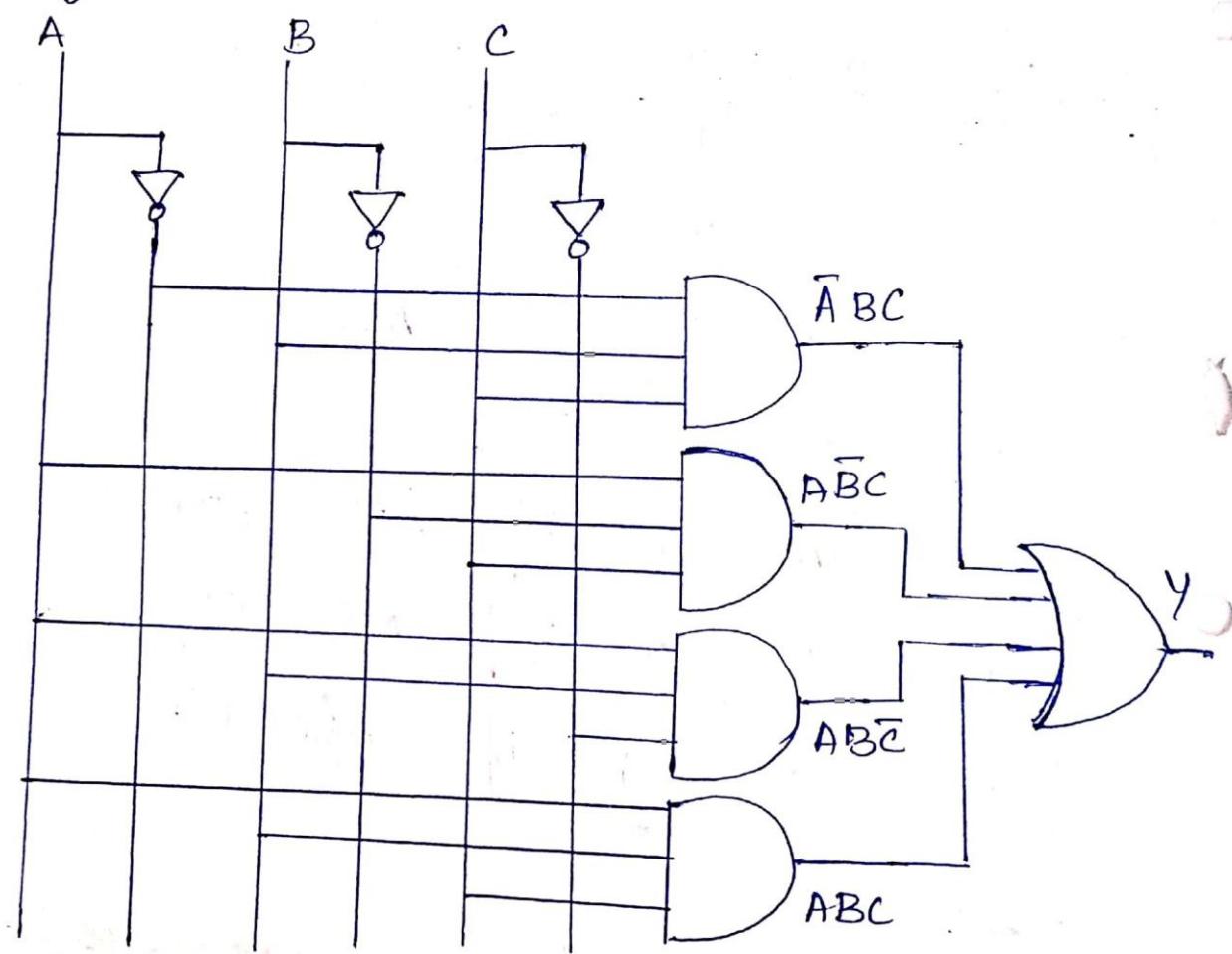
$$Y = F(A, B, C) = \sum m(3, 5, 6, 7)$$

$\Sigma$  — Symbolizes summation / OR operation performed on corresponding numbers

$Y$  is a fn of  $A, B, C$  (Boolean variables)

This kind of representation of truth table is known as canonical sum form.

Logic circuit :-



**ELECTRONIC PRINCIPLES AND DEVICES/UE24EC141A**  
**UNIT-3 Digital Electronics**

| A     | B | C | F |
|-------|---|---|---|
| 0     | 0 | 0 | 0 |
| 0     | 0 | 1 | 0 |
| <hr/> |   |   | T |
| 0     | 1 | 0 | 0 |
| 0     | 1 | 1 | 0 |
| <hr/> |   |   |   |
| 1     | 0 | 0 | 1 |
| <hr/> |   |   |   |
| 1     | 0 | 1 | 1 |
| <hr/> |   |   |   |
| 1     | 1 | 0 | 1 |
| <hr/> |   |   |   |
| 1     | 1 | 1 | 1 |

$\bar{A}B\bar{C}$

$A\bar{B}\bar{C}$

$A\bar{B}C$

$AB\bar{C}$

$ABC$

$$\begin{aligned}
 & \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC \\
 &= \bar{A}B\bar{C} + A\bar{B}(C + \bar{C}) + AB(C + \bar{C}) \\
 &= \bar{A}B\bar{C} + A\bar{B} + AB \\
 &= \bar{A}B\bar{C} + A(B + \bar{B}) = \bar{A}B\bar{C} + (A) \\
 & \quad B\bar{C} = X
 \end{aligned}$$

$$\Rightarrow A + \bar{A}X = A + X$$

$$= \underline{\underline{A + BC}}$$

$$A + B\bar{C} = A(B + \bar{B})(C + \bar{C}) + B\bar{C}(A + \bar{A})$$

**ELECTRONIC PRINCIPLES AND DEVICES/UE24EC141A**  
**UNIT-3 Digital Electronics**

$$\overline{B}C + B \Rightarrow \overline{B} + B \cdot \overline{B}C$$

$$1) Y = \overline{A}BC + A\overline{B}C + ABC\overline{C} + A\overline{B}\overline{C}$$

$$= \overline{A}BC + A\overline{B}C + AB(C + \overline{C}) = \overline{A}BC + A(\overline{B}C + B)$$

$$= \overline{A}BC + A(B + C) = \overline{A}BC + AB + AC$$

$$= B(\overline{A}C + A) + AC = B(A + C) + AC$$

$$= \underline{\underline{AB + BC + AC}}$$

$$2) Y = A + \overline{A}B + A\overline{B} = A(1 + \overline{B}) + \overline{A}B$$

$$= A + \overline{A}B = \underline{\underline{A + B}}$$

$$3) Y = AB + \overline{A}\overline{C} + A\overline{B}C(AB + C)$$

$$= AB + \overline{A} + \overline{C} + A\overline{B}C \cancel{AB} + A\overline{B}C \cdot C$$

$$= AB + \overline{A} + \overline{C} + A\overline{B}C$$

$$= AB + A\overline{B}C + \overline{A} + \overline{C}$$

$$= A(CB + \overline{B}C) + \overline{A} + \overline{C} = A(B + C) + \overline{A} + \overline{C}$$

$$= AB + AC + \overline{A} + \overline{C}$$

$$= AB + \overline{C} + \overline{A} + C = AB + \overline{A} + 1$$

$$4. \quad Y = \overline{AB} + A + \overline{B+C}$$

$$= \overline{A} + \overline{B} + A + \overline{B} \cdot \overline{C}$$

$$= 1 + \overline{B}(1 + \overline{C}) = 1 + \overline{B} = \underline{\underline{1}}$$

$$5. \quad Y = \overline{(AB+C)} \overline{(A+B+C)}$$

$$= \overline{(AB+\overline{C})} + \overline{(A+\overline{B}+C)}$$

$$\overline{AB+\overline{C}} = \overline{AB} \cdot \overline{\overline{C}} = (\overline{A} + \overline{B}) \cdot C$$

$$= \underline{\underline{\overline{AC} + \overline{BC}}}.$$

$$\overline{\overline{A+B}+C} = \overline{\overline{A} \cdot \overline{B} + C} = \overline{\overline{A} \cdot \overline{B}} \cdot \overline{C}$$

$$= (\widehat{\overline{A}} + \widehat{\overline{B}}) \cdot \overline{C}$$

$$= (A + B) \cdot \overline{C} = A\overline{C} + B\overline{C}$$

$$Y = \overline{AC} + \underline{\underline{\overline{BC} + A\overline{C} + B\overline{C}}}.$$

**ELECTRONIC PRINCIPLES AND DEVICES/UE24EC141A**  
**UNIT-3 Digital Electronics**

$$Q) Y = (A + \overline{B}C) (\overline{A} + B + C) (A + B)$$

$$= (\overline{A} + \overline{B}C) + (\overline{A} + B + C) + (\overline{A} + B)$$

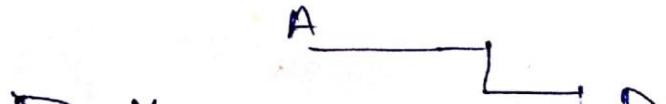
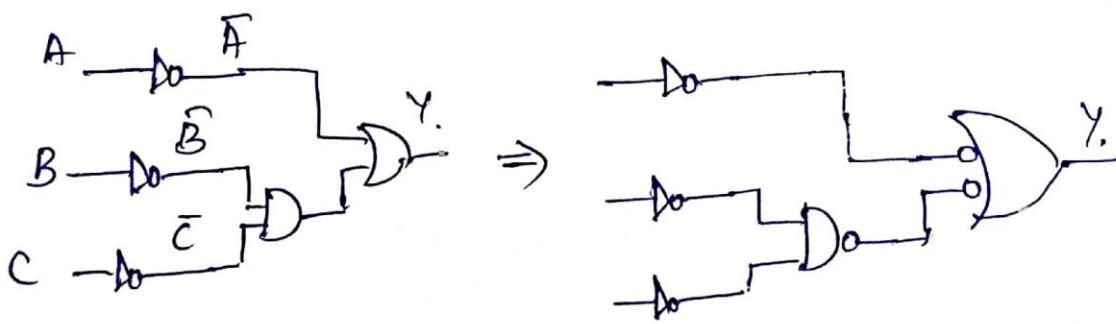
$$= (\overline{A} \cdot \overline{B}C)$$

$$\begin{aligned}\overline{A + \overline{B}C} &= \overline{\overline{A}} \cdot \overline{\overline{B}C} = \overline{A} \cdot (\overline{\overline{B}} + \overline{C}) \\ &= \overline{A} \cdot (B + \overline{C}) \\ &= \overline{A}B + \overline{A}\overline{C}\end{aligned}$$

$$\overline{\overline{A} + B + C} = \overline{\overline{A}} \cdot \overline{\overline{B}} \cdot \overline{\overline{C}} = A\overline{B}\overline{C}$$

$$(\overline{A + B}) = \overline{A} \cdot \overline{B}$$

$$\begin{aligned}Y &= AB + \overline{A}\overline{C} + A\overline{B}\overline{C} + \overline{A} \cdot \overline{B} \\ &= \overline{A}(B + \overline{B}) + \overline{C}(A\overline{B} + \overline{A}) \\ &= \overline{A} + \overline{C}(\overline{A} + \overline{B}) \\ &= \overline{A} + \overline{A}\overline{C} + \overline{B}\overline{C} = \overline{A}(1 + \overline{C}) + \overline{B}\overline{C} \\ &= \overline{A} + \underline{\overline{B}\overline{C}}\end{aligned}$$



$$Q \quad Y = \overline{(A + \bar{B}C)} \cdot \overline{(\bar{A} + \bar{B} + \bar{C})} \cdot \overline{(\bar{A} + B)}$$

$$= \overline{(A + \bar{B}C)} + \overline{(\bar{A} + \bar{B} + \bar{C})} + \overline{(\bar{A} + B)}$$

$$\overline{(A + \bar{B}C)} = \bar{A} \cdot \bar{B}C = \bar{A}(\bar{B} + \bar{C})$$

$$= \bar{A}(B + \bar{C}) = \underline{\bar{A}B + \bar{A}\bar{C}}$$

$$\overline{\bar{A} + \bar{B} + \bar{C}} = \bar{\bar{A}} \cdot \bar{\bar{B}} \cdot \bar{\bar{C}} = \underline{\underline{ABC}}$$

$$\overline{\bar{A} + B} = \bar{\bar{A}} \cdot \bar{B} = A \cdot \bar{B}$$

$$Y = \bar{A}B + \bar{A}\bar{C} + A\bar{B}C + A\bar{B}$$

$$= \bar{A}B + \bar{A}\bar{C} + AC\bar{B}C + \bar{B}) = \bar{A}B + \bar{A}\bar{C} + A(\bar{B} + C)$$

$$= \bar{A}B + \bar{A}\bar{C} + \underline{\underline{AB + AC}}$$

$$Q \quad Y = (A + B' + C)(A' + B + C)(A + B')$$

$$= (A\bar{A}' + AB + AC + \bar{A}\bar{B}' + B\bar{B}' + \bar{B}\bar{C}' + \bar{A}C + BC + C\bar{C}')$$

$$= (AB + AC + \bar{A}\bar{B}' + \bar{B}\bar{C}' + \bar{A}C + BC) (A + \bar{B})$$

$$= \cancel{AB + \bar{C} + 0 + \bar{B}\bar{C}'} + \dots$$

$$= AB + \bar{C} + 0 + A\bar{B}\bar{C} + 0 + \cancel{ABC} + 0 + A\bar{B}\bar{C} +$$

$$= \bar{A}\bar{B} + \bar{B}\bar{C} + \bar{A}\bar{B}C + 0$$

$$= AB(1+C) + \bar{C}(1+A\bar{B}) + \bar{A}\bar{B}(1+C) + \bar{B}\bar{C}(1+A)$$

$$= AB + \bar{C} + \bar{A}\bar{B} + \bar{B}\bar{C} = AB + \bar{A}\bar{B} + \bar{C}(1+\bar{B})$$

$$= AB + \bar{A}\bar{B} + \bar{C}$$

## Half Adder

- \* An adder is a digital circuit that performs addition of numbers.
- \* Half adder :- It is a combinational circuit that adds two binary digits called augend and addend and produces two ops sum and carry.

### Truth Table :-

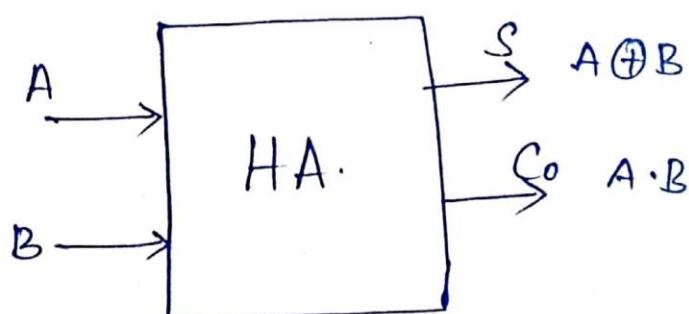
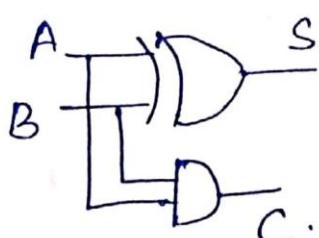
| <u>Inputs</u> |   | <u>outputs</u> |           |
|---------------|---|----------------|-----------|
| A             | B | Sum(S)         | Carry(Co) |
| 0             | 0 | 0              | 0         |
| 0             | 1 | 1              | 0         |
| 1             | 0 | 1              | 0         |
| 1             | 1 | 0              | 1         |

↓ → AND operation

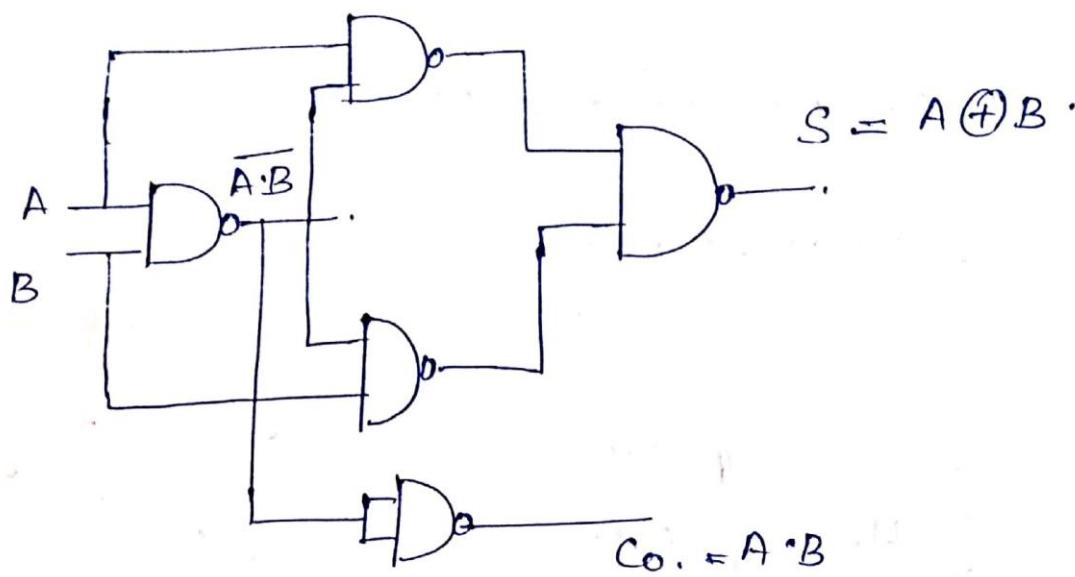
XOR  
operation

$$\therefore S = A \oplus B$$

$$C_0 = A \cdot B$$



Half Adder circuit using NAND gates.



### Full Adder.

Full adder is a combinational arithmetic circuit that add three bits and produce sum (bit) and carry (bit) as the o/p.

### Truth Table.

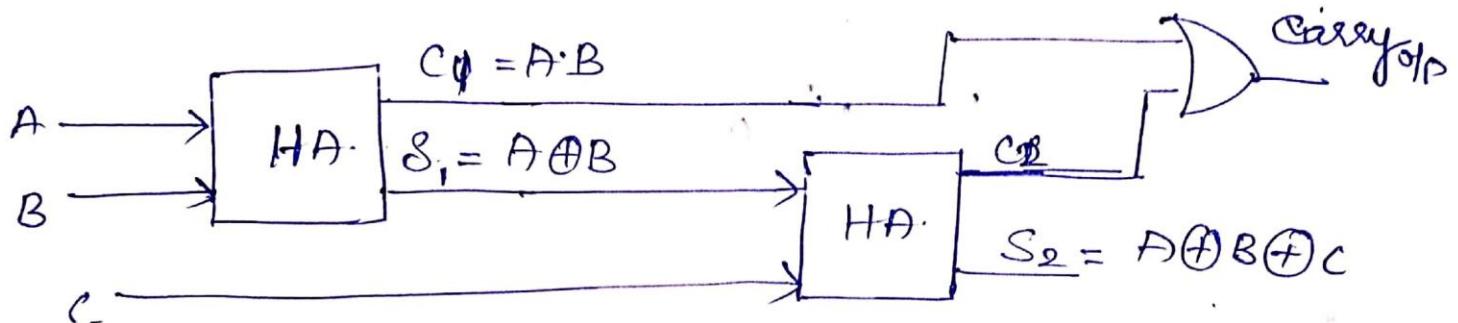
| A | B | Ci  | Sum<br>(S)                  | Carry           |  |
|---|---|-----|-----------------------------|-----------------|--|
| 0 | 0 | 0 → | 0                           | 0               |  |
| 0 | 0 | 1 → | 1   $\bar{A}\bar{B}\bar{C}$ | 0               |  |
| 0 | 1 | 0 → | 1   $\bar{A}B\bar{C}$       | 0               |  |
| 0 | 1 | 1 → | 0                           | 1 → $\bar{A}BC$ |  |
| 1 | 0 | 0 → | 1   $A\bar{B}\bar{C}$       | 0               |  |
| 1 | 0 | 1 → | 0                           | 1 → $A\bar{B}C$ |  |
| 1 | 1 | 0 → | 0                           | 1 → $ABC$       |  |
| 1 | 1 | 1 → | 1   $ABC$                   | 1 → $ABC$       |  |

$$\text{Sum} = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC = A \oplus B \oplus C$$

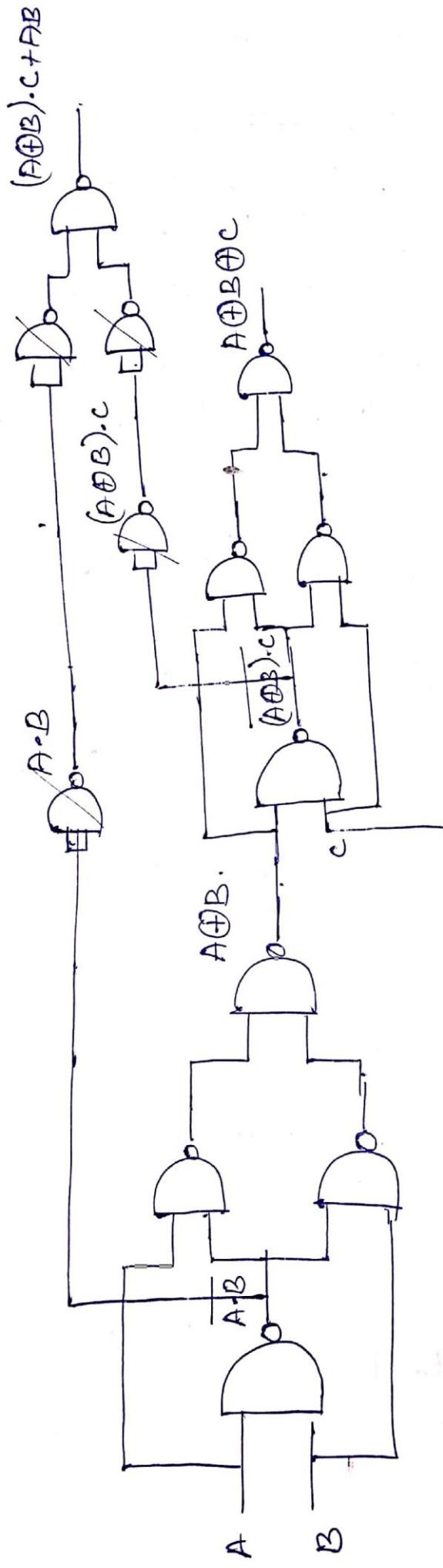
$$\text{Carry} = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

$$= \cancel{\bar{A}C} (\bar{A}B + A\bar{B})C + AB(C + \bar{C})$$

$$= \underline{(A \oplus B) \cdot C} + \underline{AB}$$

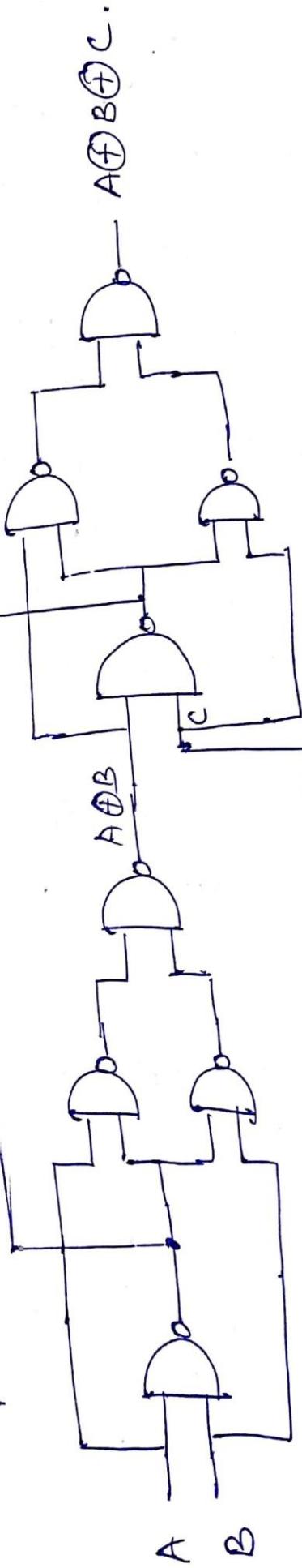


NAND gate



C

Simplified circuit :-



C

## Multiplexer (Mux) data selector

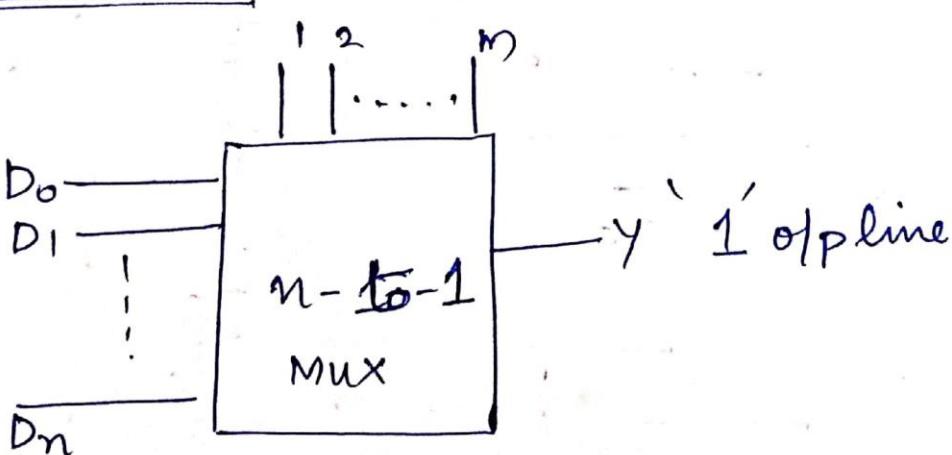
It is a device that selects between several analog or digital input signals and forwards it to a single output line.

- \* Multiple input  $\rightarrow$  single output switch
- \* Control signals are used to move data from one of the inputs to the o/p. It is also called as ~~data~~ selected lines.
- \* Each combination of select-lines shifts one data to an o/p.
- \* The circuit has ~~n~~ 'n' input signals and 'm' select / control lines. and 'one' o/p signal
- \* 'm' control /select lines can select at most  $2^m$  input signals.

$$n \leq 2^m$$

### General circuit

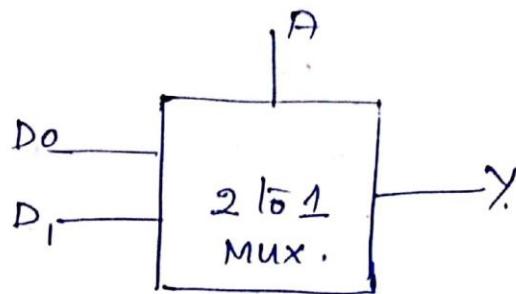
'm' select-lines



"n" i/p lines

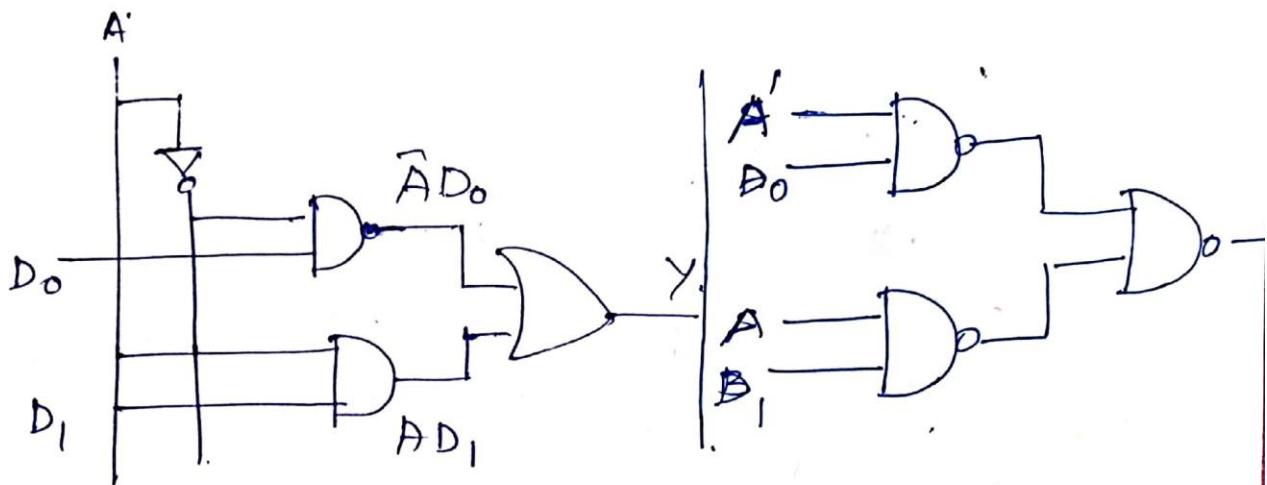
2 to 1 Mux.

| A | $Y(D_0)$ |
|---|----------|
| 0 | $D_0$    |
| 1 | $D_1$    |



$$Y = \bar{A} D_0 + A D_1$$

$$\begin{aligned} A = 0 & \quad Y = D_0 \\ A = 1 & \quad Y = D_1. \end{aligned}$$

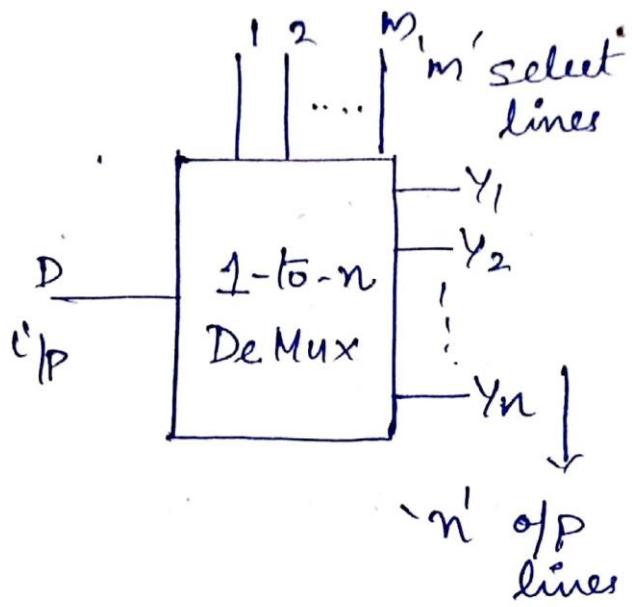


De-Multiplexer. De-Mux (Data distributor)

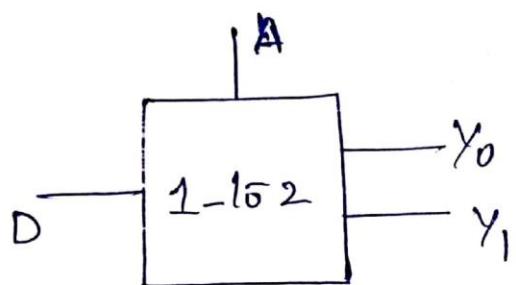
- \* De-multiplex means one to many.
- \* It is the process of taking the o/p's from one op and tying the same over ~~many~~ several o/p's.
- \* De Mux is a logic circuit that receives a signal on a single line i/p and ties it's the same under over one of several ( $2^P$ ) o/p lines.
- \* The ckt has one i/p signal and 'm' select lines. The select lines determines to which o/p the data o/p will be connected.

### Truth Table

| Select line | o/p.  |       |
|-------------|-------|-------|
| A           | $y_0$ | $y_1$ |
| 0           | D     | 0     |
| 1           | 0     | D.    |



### 1-to-2 Demux

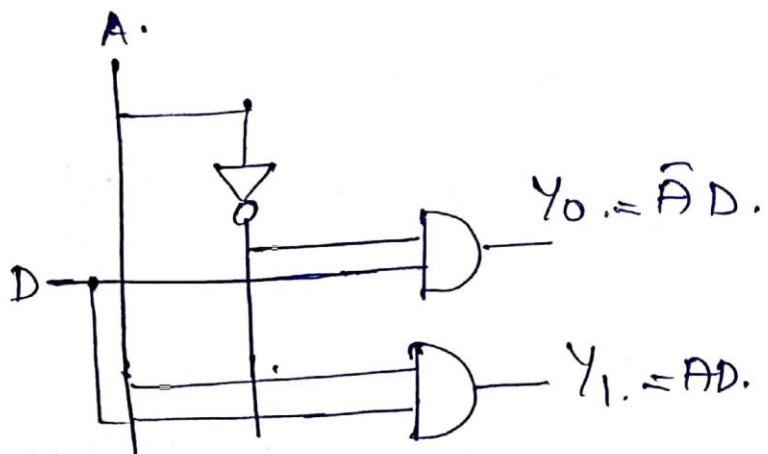


Logic

$$y_0 = \bar{A}D$$

$$y_1 = AD$$

### Logic circuit



As the serial data is changed to parallel data, the demux is also called as distributor or ~~pass~~ serial to parallel converter.

## Consensus Theorem

$$\bar{A}B + AC + BC = \bar{A}B + AC.$$

Proof :-

$$\begin{aligned}\bar{A}B + AC + BC &= \bar{A}B + AC + (\bar{A} + \bar{A})BC \\ &= \bar{A}B + AC + ABC + \bar{A}BC \\ &= \bar{A}(B + BC) \\ &= \bar{A}B(1+C) + AC(1+B) \\ &= \bar{A}B + AC\end{aligned}$$

Q Simplify  $Y = \bar{A}B + \bar{B}\bar{C} + BC + A\bar{B} + AC$  using  
consensus theorem

$$\begin{aligned}\bar{A}B + \bar{B}\bar{C} + BC + A\bar{B} + AC &= \bar{A}B + \bar{B}\bar{C} + (BC + \bar{B}A + AC) \\ &= \bar{A}B + \underline{\bar{B}\bar{C}} + BC + \underline{\bar{B}A}\end{aligned}$$

## Duality Theorem

A boolean equation can be obtained from another boolean equation by

→ changing  $+$  →  $\cdot$  and  $\cdot$  →  $+$

→ complementing '1' or '0' in the exp.

If the original equation is true, then its dual is also true.

**ELECTRONIC PRINCIPLES AND DEVICES/UE24EC141A**  
**UNIT-3 Digital Electronics**

eg: 1)  $A+1 = 1$  true

dual  $A \cdot 0 = 0$

2)  $A(B+C) = A \cdot B + A \cdot C$

$$A+BC = (A+B) \cdot (A+C)$$

Write dual of  $\overline{AB} + A + \overline{B+C} = 1$  and Prove that the dual is also true.

$$\overline{A \cdot B} + A + \overline{B+C} = 1$$

$$(\overline{A+B}) \cdot A \cdot (\overline{B+C}) = 0$$

$$\begin{aligned}
 (\overline{A+B}) \cdot A \cdot (\overline{B+C}) &= \overline{A} \cdot \overline{B} \cdot A \cdot (\overline{B+C}) \\
 &= \underline{0} = \text{RHS}.
 \end{aligned}$$

Hence dual is also true.



## Combinational and Sequential circuits.

### \* Combinational circuits:-

They are defined as the time independent circuits which do not depends upon previous inputs to generate any output.

### \* Sequential circuits:-

They are those which are dependent on clock cycles and depends on present as well as past inputs to generate any output.

# Latch and Flip Flop

1

Latch :- It is an electronic logic circuit that has two inputs and one output. The two inputs are set and reset. Latch circuits can be either active high or active low. It is level triggered and has a feed back path to retain the information.

Flip Flop :- It is an electronic circuit with two stable states that can be used to store binary data. The stored data can be changed by applying varying inputs. It is edge triggered.

## Differences between Latch and Flip Flop

| Latch  | Flip Flop   |
|--|---|
| * Latches do not require clock signal  | Flip flops have clock signals   |
| * Asynchronous device  | Synchronous device.   |
| * Latches are transparent device.<br>[Op changes immediately if c'p changes when they are enabled] | A transition from L to H or H to L of the clock signal will cause the flip flop to either change its output or retain it depending on the input signal. |
| * Latch is level sensitive device  | Flip flop is edge sensitive device.   |
| * Latches are simpler to design as there is no clock signal.                                       | Flip flops are more complex as they have clock signal and delay in the clock reaching each FF must be minimum/negl.                                     |

**ELECTRONIC PRINCIPLES AND DEVICES/UE24EC141A**  
**UNIT-3 Digital Electronics**

2

| <u>Latch</u>   | <u>Flip Flop</u>  |
|--|---|
| * The operation of a latch is faster as they do not have to wait for any clock signal. | Flip flops are comparatively slower than latches due to clock signal. |
| * The power requirement of a latch is less.  | Power requirement of a flip flop is more.                             |
| * A latch works based on enable signal.  | A flip flop works base on clock signal.                               |

### Types of latches and Flip flops

There are basically four main types of latches and flip flops. The major differences in these flip flops/latches are the number of inputs it have and how they change state.

1. SR flip flop ( Set - Reset FF)
2. D flip flop ( Data FF)
3. JK flip flop ( Jack Kilby FF)
4. T flip flop ( Toggle FF)

## R S Flip Flop (Reset - Set Flip Flop)

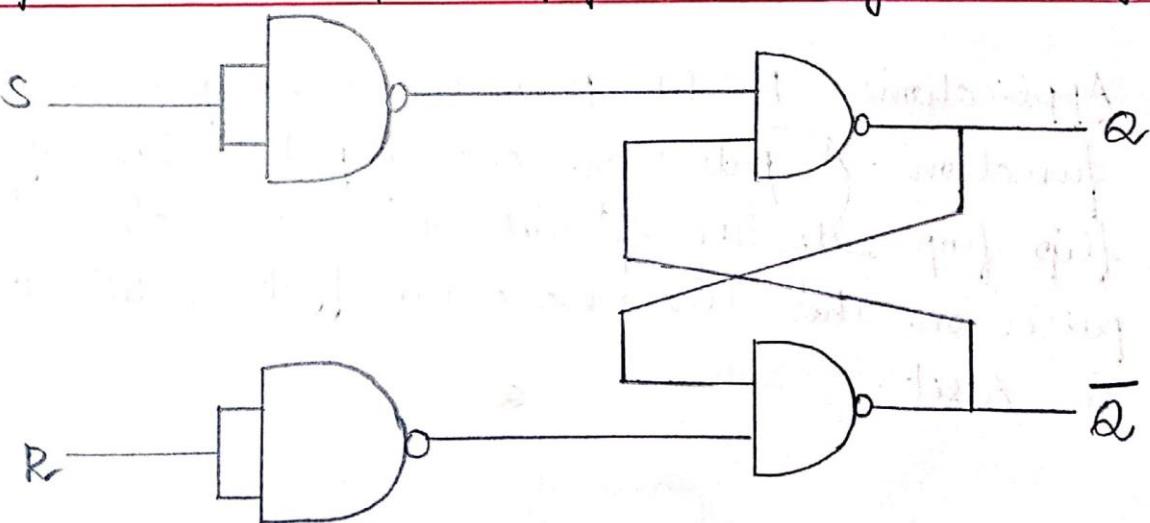
3

An SR(or RS) flip flop is an arrangement of logic gates that maintains a stable output even after the inputs are turned off.

### Set - Reset flip flop operation:-

The set - reset type flip flop is triggered to a high state at  $Q$  by the set signals and holds that value until reset to a low by a signal at the reset input. This can be implemented as a NAND gate latch as a clocked version.

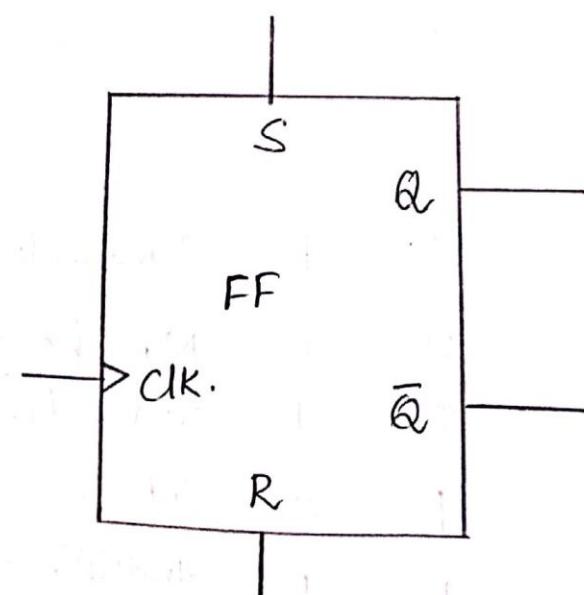
### Representation of SR flip flop using NAND gates



| S | R | Comments      |
|---|---|---------------|
| 0 | 0 | No change     |
| 0 | 1 | clear (Reset) |
| 1 | 0 | Set Q         |
| 1 | 1 | Indeterminate |

| Initial Condition | Inputs |     | Final o/p       | Comments      |
|-------------------|--------|-----|-----------------|---------------|
| $Q$               | $S$    | $R$ | $Q(t+1)$        |               |
| 0                 | 0      | 0   | 0               | No change     |
| 0                 | 0      | 1   | 0 ( $\bar{Q}$ ) | Reset         |
| 0                 | 1      | 0   | 1               | Set $Q$       |
| 0                 | 1      | 1   | ???             | Indeterminate |
| 1                 | 0      | 0   | 1               | No change     |
| 1                 | 0      | 1   | 0               | clear (Reset) |
| 1                 | 1      | 0   | 1               | Set $Q$       |
| 1                 | 1      | 1   | ???             | Indeterminate |

Applications :- RS FF provide a simple switching function. A pulse on one input line of the flip flop sets the circuit in one state. Further pulses on this line have no effect until the R-S is reset.



## JK Flip Flop (Jack Kilby Flip Flop)

5

The JK flip flop is basically a gated SR flip flop with the addition of a clock input circuitry that prevents the illegal or invalid output condition that can occur when both inputs S and R are equal to logic level '1'.

The JK flip flop is called as universal flip flop because it can be configured to work as an SR flip flop, a D flip flop or T flip flop

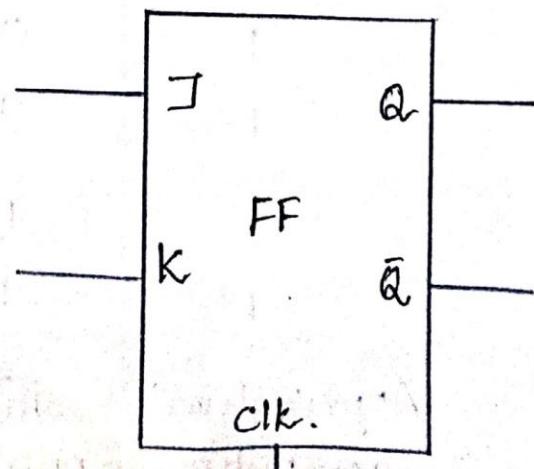
### JK flip flop operation:-

The sequential operation of JK flip flop is same as for the RS flip flop with the same SET and RESET input. The difference is that the JK flip flop does not have the invalid input states of the RS ~~FF~~ FF.

Truth Table

| S | R | Comments  |
|---|---|-----------|
| 0 | 0 | No change |
| 0 | 1 | 0         |
| 1 | 0 | 1         |
| 1 | 1 | Toggle    |

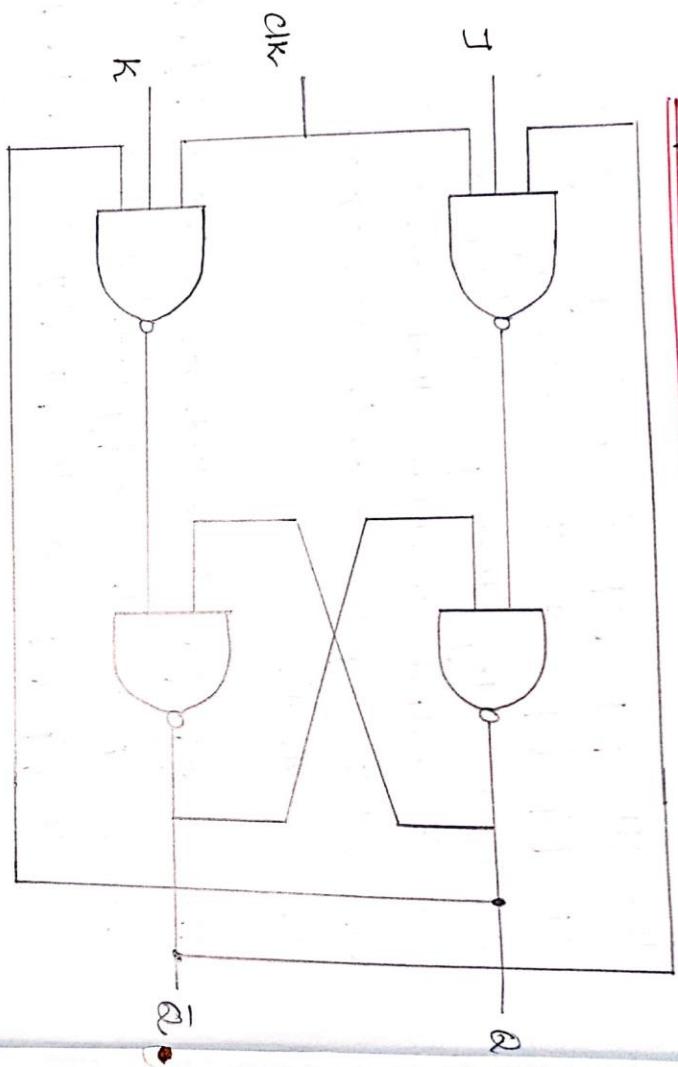
Symbol



## ELECTRONIC PRINCIPLES AND DEVICES/UE24EC141A

### UNIT-3 Digital Electronics

Representation of JK flip flop using NAND gate



Truth Table .

| Initial Condition | J    | K    | Output |
|-------------------|------|------|--------|
| Q(t)              | J(t) | K(t) | Q(t+1) |
| 0                 | 0    | 0    | 0      |
| 0                 | 0    | 1    | 1      |
| 0                 | 1    | 0    | 1      |
| 0                 | 1    | 1    | 0      |
| 1                 | 0    | 0    | 0      |
| 1                 | 0    | 1    | 1      |
| 1                 | 1    | 0    | 1      |
| 1                 | 1    | 1    | 0      |

Com

NC  
Res

Set

Toggle

NC

Res

Set

Toggle

Application : Shift Registers, counters , plm and computer applications.

## Race Around Condition:

It is an undesirable situation that occurs when a device or system attempts to perform two or more operations at the same time.

### Race around condition in SR flip flop

When the S and R inputs of an SR flip flop is at logical 1 and then the input is changed to any other condition, then the output becomes unpredictable and this is called the race around condition.

### Race around condition in JK flip flop.

When both inputs are high (J&K) the output switches between 1 and 0 (toggle condition) continuously, this is known as race around condition.

### How to avoid Race Around Condition.

#### SR & JK flip flop

If the clock on or high time is less than the propagation delay of the flip flop, then racing can be avoided. This is done by using edge triggering rather than level triggering.



**ELECTRONIC PRINCIPLES AND DEVICES/UE24EC141A**  
**UNIT-3 Digital Electronics**

6. If the flip flop is made to toggle over one clock period then racing can be avoided.

## D flip flop (Data flip flop)

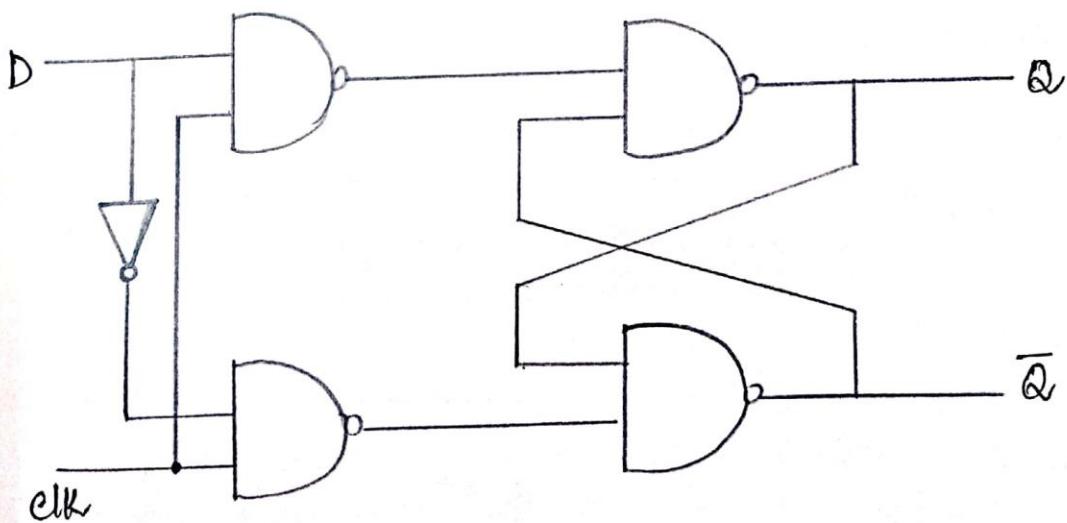
9

A 'D' type flip flop is a clocked flip flop which has two stable states. It operates with a delay in input by one clock cycle. This flip flop stores the value that is on the data line. It can be considered as a basic memory cell.

### Data flip flop operation:-

The D flip flop tracks the input, making transitions with match those of the input D. It can be made from a set/reset (SR) flip flop by tying the set to the reset through an inverter.

### Representation of D flip flop



| S | R | Q(t+1) |
|---|---|--------|
| 0 | 0 | 0      |
| 0 | 1 | 1      |
| 1 | 0 | 0      |
| 1 | 1 | 1      |

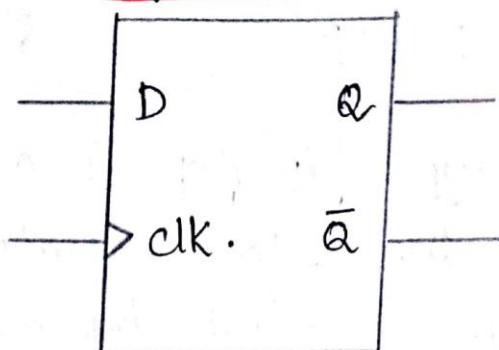
| Int Cond | D | Q | Q-bar |
|----------|---|---|-------|
| 0        | 0 | 0 | 1     |
| 0        | 1 | 1 | 0     |
| 1        | 0 | 0 | 1     |
| 1        | 1 | 1 | 0     |

ELECTRONIC PRINCIPLES AND DEVICES/UE24EC141A  
UNIT-3 Digital Electronics

10

| S | Q |
|---|---|
| 0 | 0 |
| 1 | 1 |

Symbol.



Applications :- Bounce elimination switch, data storage, data transfer, latch, registers, counter, frequency division, memory etc.

## T flip flop (Toggle Flip Flop)

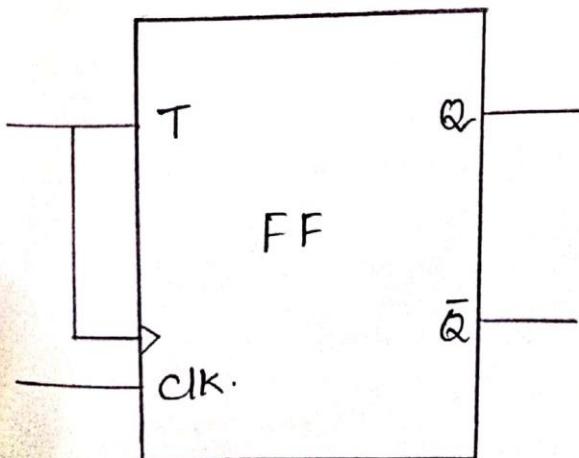
11

The toggle flip flop changes its output on each clock edge, giving an output which is half the frequency of the signal to the T input. It can be made from a JK flip flop by tying both of its inputs high.

### Toggle flip flop operation :-

T flip flop is modified form of JK flip flop making it to operate in toggling region. Whenever the clock signal is Low, the input is never going to affect the output state. The clock has to be high for the inputs to get active. Thus, T flip flop is a controlled Bi stable latch where the clock signal is the control signal.

### Representation of T flip flop



| Q <sub>n</sub> | T | Q <sub>(n+1)</sub> |
|----------------|---|--------------------|
| 0              | 0 | 0                  |
| 0              | 1 | 1                  |
| 1              | 0 | 1                  |
| 1              | 1 | 0                  |

12.

| T | Q           |
|---|-------------|
| 0 | $Q_n$       |
| 1 | $\bar{Q}_n$ |

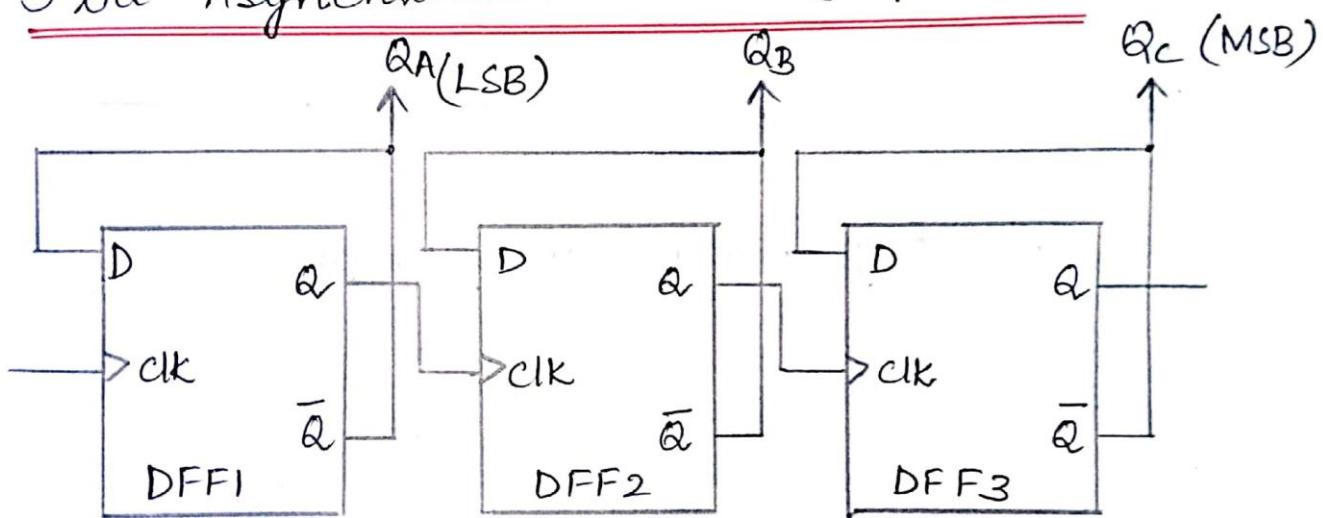
Application :- Counters and control circuits

## Counters :-

A digital circuit which is used for counting pulses is known as counter. It is a sequential circuit which is a group of flip flops with a clock signal applied. Counters are of two types.

- \* Asynchronous or ripple counters
- \* Synchronous counters

### 3 bit Asynchronous Counter (up counter)



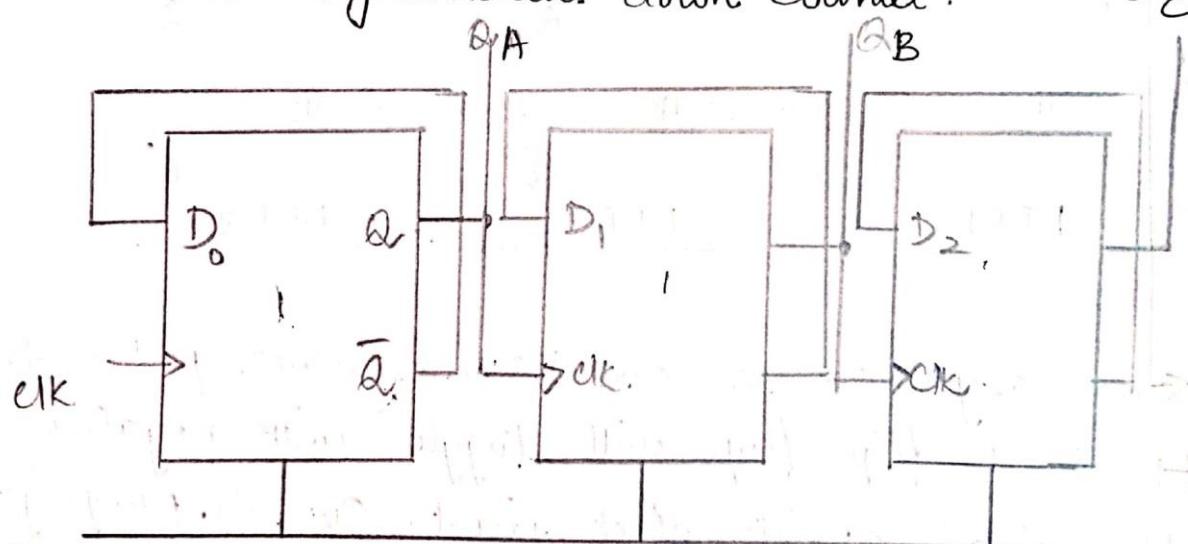
In asynchronous counter, a clock pulse drives FF. Each flip flop will toggle with negative transition at its clock input. Overall propagation delay time is the sum of individual delays.

3 bit asynchronous counter consists of 3 flip flops.

- ④ Asynchronous counter is the counter in which first flip flop is clocked by an external clock and all subsequent flip flops are clocked by the output of the preceding flip flop.

| COUNT UP (Mode) |                |                |                |
|-----------------|----------------|----------------|----------------|
| Status          | Q <sub>C</sub> | Q <sub>B</sub> | Q <sub>A</sub> |
| 0               | 0              | 0              | 0              |
| 1               | 0              | 0              | 1              |
| 2               | 0              | 1              | 0              |
| 3               | 0              | 1              | 1              |
| 4               | 1              | 0              | 0              |
| 5               | 1              | 0              | 1              |
| 6               | 1              | 1              | 0              |
| 7               | 1              | 1              | 1              |

3 bit Asynchronous down Counter :-



Reset

| clk. | Q <sub>C</sub> | Q <sub>B</sub> | Q <sub>A</sub> |
|------|----------------|----------------|----------------|
| ↑    | 1              | 1              | 1              |
| ↑    | 1              | 1              | 0              |
| ↑    | 1              | 0              | 1              |
| ↑    | 1              | 0              | 0              |
| ↑    | 0              | 1              | 1              |
| ↑    | 0              | 1              | 0              |
| ↑    | 0              | 0              | 1              |

**ELECTRONIC PRINCIPLES AND DEVICES/UE24EC141A**  
**UNIT-3 Digital Electronics**

## Shift Registers

15

Shift Registers are sequential logic circuits, capable of storage and transfer of data. They are made up of flip flops which are connected in such a way that the output of one flip flop could serve as the input of other flip flop depending on the type of shift registers created.

There are four types of shift registers :-

1. SISO — Serial In serial Out
2. SIPO — Serial In parallel out
3. PI'SO — Parallel In serial out
4. PIPD — Parallel In parallel out

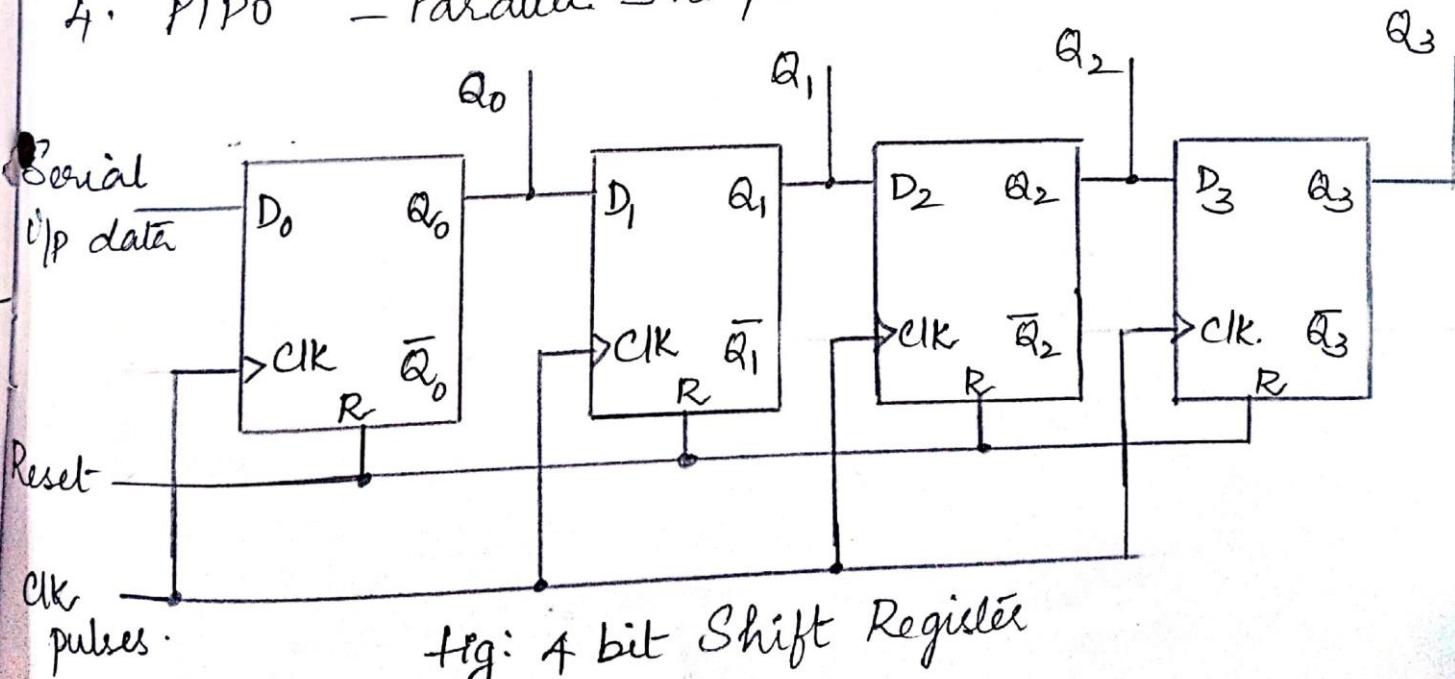


fig: 4 bit Shift Register

16

O/P is 1011.

| Clk.          | $Q_0$ | $Q_1$ | $Q_2$ | $Q_3$ |
|---------------|-------|-------|-------|-------|
| $\uparrow$ -1 | 1     | 0     | 0     | 0     |
| $\uparrow$ -2 | 1     | 1     | 0     | 0     |
| $\uparrow$ -3 | 0     | 1     | 0     | 0     |
| $\uparrow$ -4 | 1     | 0     | 1     | 1     |
| $\uparrow$ -5 | 0     | 1     | 0     | 1     |
| $\uparrow$ -6 | 0     | 0     | 1     | 1     |
| $\uparrow$ -7 | 0     | 0     | 0     | 0     |
| $\uparrow$ -8 | 0     | 0     | 0     | 1     |

O/P. 1011