25 Marks
ग्रह 10 710

①

1

Digital Electronics Principle

Number System

- → Decimal number system
- → Binary number system
- → Octal number system.
- → Hexadecimal number system.

→ All these number system are known as positional number system because each digit position is assigned with a particular weight.

→ The total number used in any number system is known as base or radix of that system and is denoted by r.

		r
→ Decimal →	0, 1, 2, 3, 4, 5, 6, 7, 8, 9 ,	10
Binary →	0, 1 ,	2
Octal →	0, 1, 2, 3, 4, 5, 6, 7 ,	8
Hexadecimal →	0, 1, 2 --- 8, 9, A, B, C, D, E, F ,	16

e.g. < 485.76 >

Decimal number system:

Suppose a number system having base y is is written as

$$(a_n\ a_{n-1}\ -\ a_2\ a_1\ a_0\ .\ b_1\ b_2\ b_3\ ---\ b_m)_y$$

$= \cancel{a_n \times y^{n}}$

$$= a_n y^n + a_{n-1} y^{n-1} ----- a_1 y^1 + a_0 y^0 + b_1 y^{-1} + b_2 y^{-2} --- b_m y^{-m}$$

$(5263.343)_{10}$ is calculated as,

$5\times10^{3}+2\times10^{2}+6\times10+3+3\times10^{-1}+4\times10^{-2}+3\times10^{-3}$

$= 5000+200+60+3+\frac{3}{10}+\frac{4}{100}+\frac{3}{1000}$

$= 5263+(.3+.04+.003)$

$= 5263+0.343$

$= 5263.343$

$$\begin{array}{r} .3 \\ .04 \\ .003 \\ \hline .343 \end{array}$$

Number Base Conversion :-

4 number system, 2 conversion at a time

${}^4P_2 = \frac{4!}{2!} = 12$ ways)

Decimal → binary, → octal, → hexa decimal

binary → decimal, → octal, → hexadecimal

Like that

(1) Decimal to binary

ex : 1 $(19)_{10} = (\_\_\_)_2$

Ans :

$$\begin{array}{l|l} 2 & 19 \\ \hline 2 & 9 \rightarrow 1 \\ \hline 2 & 4 \rightarrow 1 \\ \hline 2 & 2 \rightarrow 0 \\ \hline 2 & 1 \rightarrow 0 \\ \hline & 0 \rightarrow 1 \end{array}$$

$(19)_{10} = (10011)_2$ (Ans)

ex-2 :- $(0.625)_{10} = (\ \ )_2$

$$\begin{array}{r} .625 \\ \times\ \ 2 \\ \hline 1.250 \rightarrow 1 \\ 2 \\ \hline 0.500 \rightarrow 0 \\ 2 \\ \hline 1.000 \rightarrow 1 \end{array}$$

$(.625)_{10} = (0.101)_2$

$\therefore (0.625)_{10} = (0.101)_2$

Ex-3

$$(0.1875)_{10} = (\quad)_2$$

$$\begin{array}{r} 0.1875 \\ \times \quad 2 \\ \hline 0.3750 \rightarrow 0 \\ 2 \\ \hline 0.7500 \rightarrow 0 \\ 2 \\ \hline 1.5000 \rightarrow 1 \\ 2 \\ \hline 1.0000 \rightarrow 1 \end{array}$$

$$(0.1875)_{10} = (0.0011)_2$$

Ex-4 :

$$(26.375)_{10} = (\quad)_2$$

Consider integer part

$$\begin{array}{r|l} 2 & 26 \\ \hline 2 & 13 \rightarrow 0 \\ \hline 2 & 6 \rightarrow 1 \\ \hline 2 & 3 \rightarrow 0 \\ \hline 2 & 1 \rightarrow 1 \\ \hline & 0 \rightarrow 1 \end{array}$$

$$(26)_{10} = (11010)_2$$

$$\begin{array}{r} 0.375 \\ 2 \\ \hline 0.750 \rightarrow 0 \\ 2 \\ \hline 1.500 \rightarrow 1 \\ 2 \\ \hline 1.000 \rightarrow 1 \end{array}$$

$$(.375)_{10} = (.011)_2$$

$$(26.375)_{10} = (11010.011)_2$$

2) Decimal to octal conversion:-

ex-5

$$(8276)_{10} = (\quad)_8$$

$$
\begin{array}{l|ll}
8 & 8276 & \\
8 & 1034 & \rightarrow 4 \\
8 & 129 & \rightarrow 2 \\
8 & 16 & \rightarrow 1 \\
8 & 2 & \rightarrow 0 \\
 & 0 & \rightarrow 2
\end{array}
$$

$$(8276)_{10} = (20124)_8$$

ex-6

$$(5734)_{10} = (\quad)_8$$

$$
\begin{array}{l|ll}
8 & 5734 & \\
8 & 716 & \rightarrow 6 \\
8 & 89 & \rightarrow 4 \\
8 & 11 & \rightarrow 1 \\
8 & 1 & \rightarrow 3 \\
 & 0 & \rightarrow 1
\end{array}
$$

$$(5734)_{10} = (13146)_8$$

ex-7

$$(0.0078125)_{10} = (\quad)_8$$

$$
\begin{array}{r}
.0078125 \\
\times 8 \\
\hline
0.0625\cancel{000} \rightarrow 0 \\
\times 8 \\
\hline
0.5000\cancel{000} \rightarrow 0 \\
\times 8 \\
\hline
4.0000 \rightarrow 4
\end{array}
$$

$$(.0078125)_{10} = (0.004)_8$$

ex-8

$(112.01)_{10} = (\quad)_8$

$$\begin{array}{r|l} 8 & 112 \\ \hline 8 & 14 \rightarrow 0 \\ \hline 8 & 1 \rightarrow 6 \\ \hline & 0 \rightarrow 1 \end{array}$$

$(112)_{10} = (160)_8$

$$\begin{array}{r} .01 \\ \times \quad 8 \\ \hline 0.08 \rightarrow 0 \\ 8 \\ \hline 0.64 \rightarrow 0 \\ 8 \\ \hline 5.12 \rightarrow 5 \\ 8 \\ \hline 0.96 \rightarrow 0 \end{array}$$

$(0.01) \simeq (.0050)$

It will continue. Approximating to 4 digits.

$(112.01)_{10} = (160.0050)_8$.

3) <u>Decimal to Hexadecimal Conversion</u>

ex-9 $(84)_{10} = (\quad)_{16}$

$$\begin{array}{r|l} 16 & 84 \\ \hline 16 & 5 \rightarrow 4 \\ \hline & 0 \rightarrow 5 \end{array}$$

$(84)_{10} = (54)_{16}$

ex-10 $(0.02)_{10} = (\quad)_{16}$

$$
\begin{array}{r}
.02 \\
\times\ 16 \\
\hline
0.32 \\
16 \\
\hline
5.12 \\
16 \\
\hline
1.92 \\
16 \\
\hline
14.72
\end{array}
\qquad
\begin{array}{l}
\\ \\ \\
\to 0 \\ \\ \\
\to 5 \\ \\ \\
\to 1 \\ \\ \\
\to E
\end{array}
$$

⋮

$(.02)_{10} = (0.051E)_{16}$

ex-11

$(89726.001953125)_{10} = (\quad)_{16}$

$$
\begin{array}{r|l}
16 & 89726 \\
\hline
16 & 5607 \to E \\
\hline
16 & 350 \to 7 \\
\hline
16 & 21 \to E \\
\hline
16 & 1 \to 5 \\
\hline
 & 0 \to 1
\end{array}
$$

$(89726)_{10} = (15E7E)_{16}$

$$
\begin{array}{r}
.001953125 \\
16 \\
\hline
0.031250000 \\
16 \\
\hline
0.50000 \\
16 \\
\hline
8.0
\end{array}
\qquad
\begin{array}{l}
\\ \\ \\
\to 0 \\ \\ \\
\to 0 \\ \\ \\
\to 8
\end{array}
$$

$(.001953125)_{10} = (0.008)_{16}$

$\therefore (89726.001953125)_{10}$

$= (15E7E.008)_{16}$

Error BPUT $(BAD)_{16} = ?$ $(DAD)_{16}$

4) Binary to Decimal Conversion:-

$(11010.101)_2 = (\quad)_{10}$

$1\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2 + 0 + 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3}$

$= 16 + 8 + 0 + 2 + 0 + \frac{1}{2} + 0 + \frac{1}{8}$

$= 26 + (\frac{1}{2} + \frac{1}{8})$

$= 26 + (\frac{4+1}{8})$

$= 26 + \frac{5}{8}$

$= 26 + 0.625$

$= 26.625$

$\therefore (11010.101)_2 = (26.625)_{10}$

5) Octal to Decimal Conversion:-

$(230.1)_8 = (\quad)_{10}$

$\cancel{2\times8^{-2}}$ $2\times8^2 + 3\times8^1 + 0 + 1\times8^{-1}$

$= 128 + 24 + \frac{1}{8}$

$= 152 + .125$

$(230.1)_8 = (152.125)_{10}$

6/ Hexadecimal to decimal conversion

$$(AD0.C)_{16} = (\quad)_{10}$$

$A \times 16^2 + D \times 16^1 + 0 + C \times 16^{-1}$

$= 10 \times 256 + 13 \times 16 + \frac{12}{16}$

$= 2560 + 208 + \frac{3}{4}$

$= 2768 + .75$

$(AD0.C)_{16} = (2768.75)_{10}$

$(BAD)_{16} = (\quad)_{10}$

Ans: → 2989

7) Octal - to - Binary Conversion

Octal Digit	Binary equivalent
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

$$(765.234)_8 = (\quad)_2$$

7	6	5	.	2	3	4
↓	↓	↓		↓	↓	↓
(111	110	101	.	010	011	$100)_2$

$(765.234)_8 = (111\,110\,101.010\,011\,100)_2$

1) Binary - to - Octal Conversion :-

→ $(011\,001\,111\,010 \cdot 110\,001\,110)_2 = (\quad)_8$

← Start →

$(011\ 001\ 111\ 010 \cdot 110\ 001\ 110)_2$

$= (3\ 1\ 7\ 2 \cdot 6\ 1\ 6)_8$

→ $(11011 \cdot 1011)_2 = (\ ?\ )_8$

$(011\,011 \cdot 101\,100)_2 =$

$(3\ 3 \cdot 5\ 4)_8$

1) Hexadecimal to Binary

Hex digit	Binary equivalent
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

$(\cancel{ADF}$ $(ADFE.5CB)_{16} = (\;)_2$

$(1010\ 1101\ 1111\ 1110\ .\ 0101\ 1100\ 1011)_2$

10) Binary to Hexadecimal

$(1\ 001\ 011\ 010\ 101\ .\ 1011\ 000\ 001)_2 = (?)_{16}$

$0001\ 0010\ 1101\ 0101\ .\ 1011\ 0000\ 0100 =$

$(1\ 2\ D\ 5\ .\ B\ 0\ 4)_{16}$

11) Hex to octal

Step-1 → Hex - Binary

Step-2 → Binary - octal.

Ex: $(D03.F5)_{16} = (\ ?\ )_8$

$= (1101\ 0000\ 0011\ .\ 1111\ 0101)$

$(D03.F5) = (1\ 101\ 000\ 000\ 011\ .\ 111\ 101\ 010)$

$= (6403.752)_8$

12) Octal to Hex

Step-I → Octal - Binary

Step-II → Binary to Hex.

$(1375.25)_8$

$= (001\ 011\ 111\ 101.\ 010\ 101)$

$= (\underbrace{0010}\ \underbrace{1111}\ \underbrace{1101}\ .\ \underbrace{0101}\ \underbrace{0100})$

$= (2\ F\ D\ .\ 5\ 4)_{16}$

Some Important terms:-

Bit → Abbreviation used for binary digit i.e 0 or 1

Byte → A group of 8 bits.

Nibble → A group of 4 bits.

1 KB → 1 kilo Byte $= 2^{10}$ Bytes $= 1024$ bytes.

1 MB → 2^{20} Bytes $= 1024 \times 1024$ bytes

1 GB $= 2^{30}$ Byte $= 1024 \times 1024 \times 1024$ bytes.

4 G Bytes $= 4 \times 2^{30} = 2^2 \times 2^{30}$

$= 2^{32}$ Bytes.

Ex:- Convert the following number with indicated base to decimal.

$(4310)_5$

$= 4 \times 5^3 + 3 \times 5^2 + 1 \times 5 + 0$

$= 500 + 75 + 5$

$= (580)_{10}$

Ex :- What is the largest binary number that can be expressed with 12 bits? What is its decimal equivalent.

Ans :- $(1111\ 1111\ 1111)_2$ = The largest binary no.

Its decimal equivalent = $2^{12}-1 = 4096-1 = 4095$.

(4 bit → $2^4-1 = 16-1 = 15$)

Binary Addition :-

0+0 → 0

0+1 → 1

1+0 → 1

1+1 = 10
 ↑ carry

1+1+1 = 11
 ↑ carry

In binary (2 is 1 unit)

In decimal (10 is 1 unit)

ex :-

$$\begin{array}{r} 1011 \\ +\ 1111 \\ \hline 11010 \end{array} \quad \begin{array}{r} 11 \\ +15 \\ \hline 26 \end{array}$$

Compare with decimal addⁿ

$$\begin{array}{r} 1 \\ 879 \\ 246 \\ \hline 1125 \end{array}$$

Imp

$$\begin{array}{r} 1 \\ 11 \\ 1\,1\,1 \\ 1\,1\,1\,1 \\ \hline 11010 \end{array}$$

Verification :-

→ 1
→ 3
→ 7
→ 15
26

$$\begin{array}{l|ll} 2 & 26 & \\ 2 & 13 & \to 0 \\ 2 & 6 & \to 1 \\ 2 & 3 & \to 0 \\ 2 & 1 & \to 1 \\ & 0 & \to 1 \end{array}$$

$26 = (11010)_2$

Additional Concepts / Problems :-

A. Signed Binary Numbers

Positive integers (including zero) can be represented as unsigned numbers. However, to represent −ve integers, we need ~~additional~~ a notation for −ve values.

(i) Signed - Magnitude representation.

(ii) 1's Complement representation.

(iii) 2's Complement representation.

The User first fixes (assumes) whether he is using Signed number or Unsigned number.

If he is using signed number system

→ the leftmost bit represents the sign and rest of bits represent the number.

If the binary number is assumed to be unsigned, then the left most bit is the MSB of the number.

e.g 1) 0 1001 considered as 9 (unsigned binary)

or ,, ,, ,, +9 (Signed binary)

2) 1 1001 ,, ,, 25 (unsigned binary)

1 1001 ,, ,, −9 (signed binary)

This is because the left most bit (1) designates a −ve and other 4 bits represent binary 9.

8 bit representation of (−9)

Signed Magnitude representation: 1 0001001

↑ (For sign)

1's Complement representation :-

+9 → 0 0 0 0 1 0 0 1

−9 → 1 1 1 1 0 1 1 0

} → Making 1 to 0 & 0 to 1

2's Complement representation

[1's Complement representation + 1]

i.e 1 1 1 1 0 1 1 0

\+ 1

−9 → 1 1 1 1 0 1 1 1

in 2's complement method

The MSB '1' indicates it is −ve number in all the 3 way of representation,

Note:- The +9 has same representation in signed-magnitude, 1's Complement & 2's Complement representation i.e 00001001

Arithmatic addition of Signed numbers. (2's Complement form)

1) +6 → 0 0 0 0 0 1 1 0

\+13 → + 0 0 0 0 1 1 0 1

+19 → 0 0 0 1 0 0 1 1

2) −6

\+13

To find −6 in 2's Complement form

+6 → 0 0 0 0 0 1 1 0

1's → 1 1 1 1 1 0 0 1

2's → + 1

1 1 1 1 1 0 1 0

$\therefore$ $-6 \rightarrow$ 1 1 1 1 1 0 1 0

$+13 \rightarrow$ + 0 0 0 0 1 1 0 1

$+7$ $\cancel{1}$ 0 0 0 0 0 1 1 1

(1 is discarded as 8 bit representation)

3) $+6$
-13

To find -13 in 2's Complement form

$+13 \rightarrow$ 0000 1101

1's $\rightarrow$ 1111 0010

2's $\rightarrow$ +1
1111 0011

$+6 \rightarrow$ 0 0 0 0 0 1 1 0

$-13 \rightarrow$ + 1 1 1 1 0 0 1 1

-7 1 1 1 1 1 0 0 1

(Verification $\rightarrow$ $+7 \rightarrow$ 00000 111

1's $\rightarrow$ 11111 000

2's $\rightarrow$ +1
11111001

4) $-6 \rightarrow$ 1111 1010

$-13 \rightarrow$ + 1111 0011

-19 $\cancel{1}$ 1110 1101

[As derived earlier]

(Verification $\rightarrow$ $+19 \rightarrow$ 0001 0011

1's $\rightarrow$ 1110 1100

2's $\rightarrow$ + 1
1110 1101

2	19	
2	9	→ 1
2	4	→ 1
2	2	→ 0
2	1	→ 0
	0	1

Ex :- What is the largest binary number that can be expressed with 12 bits? What is its decimal equivalent.

Ans :- $(1111\ 1111\ 1111)_2$ = The largest binary no.

Its decimal equivalent = $2^{12}-1 = 4096-1 = 4095$.

$\langle$ 4 bit → $2^4-1 = 16-1 = 15$ $\rangle$

Binary Addition :-

0+0 → 0

0+1 → 1

1+0 → 1

1+1 = 10 ← carry

1+1+1 = 11 ← carry

In binary (2 is 1 unit)

In decimal (10 is 1 unit)

ex :-

$$\begin{array}{r} 1011 \\ +\ 1111 \\ \hline 11010 \end{array} \quad \begin{array}{r} 11 \\ +15 \\ \hline 26 \end{array}$$

Compare with decimal addⁿ

$$\begin{array}{r} 879 \\ 246 \\ \hline 1125 \end{array}$$

Imp

$$\begin{array}{r} 1 \\ 1\ 1 \\ 1\ 1\ 1 \\ 1\ 1\ 1\ 1 \\ \hline 1\ 1\ 0\ 1\ 0 \end{array}$$

Verification :-

→ 1

→ 3

→ 7

→ 15

26

2 | 26
2 | 13 → 0
2 | 6 → 1
2 | 3 → 0
2 | 1 → 1
0 → 1

$26 = (11010)_2$

Binary Subtraction :-

(i) Direct Subtraction

(ii) Complement subtraction.

Direct Subtraction :-

0 − 0 = 0

0 − 1 = 1 , with borrow 1.

1 − 0 = 1

1 − 1 = 0.

Ex −

```
  11001   → 25
− 10010   → 18
  -----     --
  00111      7
```

```
   335
 − 297
   ---
    38

  3001
− 1997
  ----
  1004
```

If

```
  10010
− 11001
  -----
```

It will come −ve.

So subtract from the bigger no, then attach a −ve sign.

```
  11001
  10010
  -----
  00111
```

10010 − 11001 = − (111)

(Generally not used. We use sign magnitude, 1's complement, 2's Complement method to represent −ve numbers).

①

$$\begin{array}{r} 1010.11 \\ -\ 0110.01 \\ \hline 100.10 \end{array} \left\{ \begin{array}{l} \rightarrow 10.75 \\ \rightarrow 6.25 \\ \hline \quad 4.50 \end{array} \right.$$

Negative number and their representation :-

Sign magnitude :-

In sign-magnitude representation the Most significant bit (MSB) is used to represent the sign.

If MSB is 0 → +ve

MSB is 1 → −ve

ex :

0101 → +5

1101 → −5

In a n bit system,

maxm +ve no. that can be represented → $+(2^{n-1}-1)$

" −ve " " " → $-(2^{n-1}-1)$

ex : 4 bit system,

Maxm +ve no → $+(2^3-1) = +7$

" −ve no → $-(2^3-1) = -7$

i.e −7 ---- 7

0	000	0
0	001	
0	010	
0	011	
0	100	
0	101	
0	110	
0	111	+7
1	000	−0
1	001	
1	010	
1	011	
1	100	
1	101	
1	110	
1	111	−7

Dis advantage :-

+0 → 0000

−0 → 1000

} +0 & −0 have different representation although +0 = −0 = 0.

One's Complement :-

→ In a binary number, if we replace 0 by 1 & each 1 by 0, we obtain another binary number which is 1's complement of the first binary number.

→ Both the numbers are complement to each other

→ If MSB, 0 → +ve no.
If MSB, 1 → −ve no.

→ But magnitude of −ve no is obtained by taking 1's complement of the number.

ex :

+7 → 0111

−7 → 1000

OR

To find 1's complement the no. Subtract from 1111.

$$\begin{array}{r} 1111 \\ -0111 \\ \hline 1000 \end{array}$$

← 7
← (−7)

Q → What is 1000 in 1's complement ?

Step 1 → Since MSB 1 → −ve number.

To find its magnitude

Step-2 → Take 1's complement

1000 → 0111 → 7.

∴ The no. is −7. (Ans)

→ Maxm +ve no → $+(2^{n-1}-1)$
" −ve no → $-(2^{n-1}-1)$

for 4 bit +7 to −7

Disadvantage :-

In 1's Complement representation, Zero has 2 different representation.

Actually +0 = -0, but in 1's Complement,

+0 → 0000

-0 → 1111

But in 2's Complement method, it has been corrected.

2's Complement representation :-

By adding 1 to one's complement of a binary number we get the 2's Complement of that binary number.

→ +5 → 0101

1's Complement → 1010

+ 1

2's " → 1011

(2's) -5 → 1011

Q/ What is the number which has 2's Complement representation 1011.

Ans : MSB = 1

So it is a −ve no.

1's Complement of magnitude will be

$$\begin{array}{r} 1011 \\ -\quad 1 \\ \hline 1010 \end{array}$$

Actual no → 0101

∴ 1010 It − (0101) i.e −5.

→ +0 → 0000

−0 → 0000 ↓ 1's complement

$$\begin{array}{r} 1111 \\ +\quad 1 \\ \hline \textcircled{1}\,0000 \end{array}$$ 2's complement.

+0 → +0 (both have value 0000)

→ Max^m +ve no → $+(2^{n-1}-1)$

Min → " −ve no → $-(2^{n-1})$

ex, 4 bit, −8 to +7

Single representation for '0' gives the advantage to represent −8.

Q What is decimal equivalent of a number, whose 2's complement is 1011.

Ans:- (−5)

	Sign, magnitude	1's complement	2's complement
1 →	0001	0001	0001
2 →	0010	0010	0010
3 →	0011	0011	0011
4 →	0100	0100	0100
5 →	0101	0101	0101
6 →	0110	0110	0110
7 →	0111	0111	0111
+0 →	0000	0000	0000
−0 →	1000	1111	0000
−1 →	1001	1110	1111
−2 →	1010	1101	1110
−3 →	1011	1100	1101
−4 →	1100	1011	1100
−5 →	1101	1010	1011
−6 →	1110	1001	1010
−7 →	1111	1000	1001
−8 →	X	X	1000.

1's complement

1's

$\xrightarrow[+1]{2's}$

same

−8 in 2's complement

+8 → 1000

1's complement → 0111

2's $\quad$ + 1

$\quad$ 1000

∴ −8 → 1000 in 2's complement method.

Advantage of 2's Complement

1) +0 & −0 i.e Zero has ~~sing~~ single representation

2) We can represent one extra number in 2's complement. (e.g. ⇒ −8 in 4 bit representation)

Complement Subtraction:-

(A) 1's Complement Subtraction

$$\begin{array}{r} -\ A \leftarrow \text{Minuend} \\ B \leftarrow \text{Subtrahend.} \\ \hline \end{array}$$

(i) Add 1's Complement of Subtrahend to the minuend.

(ii) If there is End-Around-Carry (EAC), Remove EAC & add it to the result.

(iii) If there is no EAC, then re complement it and attach a −ve sign.

Ex →

$$\begin{array}{r} 111 \\ -\ 100 \\ \hline \end{array}$$

Ans → (i) 1's Complement of 100 → 011

(ii)

$$\begin{array}{r} 111 \\ +\ 011 \\ \hline 1\,010 \\ 1 \\ \hline 011 \end{array}$$ (Ans)

ex:

$$\begin{array}{r} 1010 \\ -1101 \\ \hline \end{array}$$

(i) 1's complement of 1101 → 0010

(ii)

$$\begin{array}{r} 1010 \\ +\ 0010 \\ \hline 1100 \end{array}$$

(iii) Since no carry, recomplement it 0011 and attach a -ve sign.

$-(0011)$

2's Complement Subtraction :-

(i) Add 2's complement subtrahend to the minuend.

(ii) If there is End-Around - Carry (EAC), then carry will be discarded and answer is +ve

(iii) If no carry, then recomplement (2's) and attach a -ve sign.

ex :-

$$\begin{array}{r} 110011 \\ -100111 \\ \hline ? \end{array}$$

(i) 2's complement of 100111

1's → 011000

$$\begin{array}{r} +1 \\ \hline \end{array}$$

2's → 011001

(ii) $\quad 110011$
$+\ 011001$

discard ← 1 0 0 1 1 0 0

(iii) Ans is (001100).

ex :- $\quad 1010$
$-\ 1101$
?

(i) 2's complement of 1101

$1101 \xrightarrow{1's} 0010$
$+\ 1$
$2's \rightarrow 0011$

(ii) $\quad 1010$
$\quad 0011$
$\quad 1101$

No carry.

(iii) Again find 2's complement

1101 →
1's 0010
+ 1
2's 0011

Attach a -ve sign. -(0011)

(Ans)

9's Complement:-

The 9's complement of a decimal no. is obtained by subtracting each digit in the number from 9.

Decimal digit	9's complement
0	$\rightarrow 9-0=9$
1	$\rightarrow 9-1=8$
2	$\rightarrow 9-2=7$
3	$\rightarrow 9-3=6$
4	$\rightarrow 9-4=5$
5	$\rightarrow 9-5=4$
6	$\rightarrow 9-6=3$
7	$\rightarrow 9-7=2$
8	$\rightarrow 9-8=1$
9	$\rightarrow 9-9=0$

ex:- Find 9's complement of following decimal no.

(i) 43 (ii) 122 and (iii) 3267

(i)
$$\begin{array}{r} 99 \\ -43 \\ \hline 56 \end{array}$$
(Ans)

(ii)
$$\begin{array}{r} 999 \\ 122 \\ \hline 877 \end{array}$$
(Ans)

(iii)
$$\begin{array}{r} 9999 \\ 3267 \\ \hline 6732 \end{array}$$
(Ans)

9's complement Subtraction method :- (Same as 1's complement)

(i) Add 9's complement substrahend to the minuend.

(ii) If there is EAC, then carry will be added to the sum and answer is +ve.

(iii) If there is no carry, then recomplement (9's) and attach a -ve sign.

ex :-

$$\begin{array}{r} 48 \\ -\ 24 \\ \hline ? \end{array}$$

(i) 9's complement of 24 → $\begin{array}{r} 99 \\ 24 \\ \hline 75 \end{array}$

(ii)

$$\begin{array}{r} 48 \\ +\ 75 \\ \hline 1\,23 \\ 1 \\ \hline 24 \end{array}$$ (Ans)

ex :-

$$\begin{array}{r} 26 \\ -\ 38 \\ \hline ? \end{array}$$

(i) 38 → 9's complement $\begin{array}{r} 99 \\ -\ 38 \\ \hline 61 \end{array}$

(ii)

$$\begin{array}{r} 26 \\ 61 \\ \hline 87 \end{array}$$

(iii) Since no carry, recomplement it

9's Complement

$$\begin{array}{r} 99 \\ -\ 87 \\ \hline 12 \end{array}$$

Attach -ve sign. -12 (Ans)

10's Complement :-

The 10's complement of a decimal number is obtained by adding 1 to the 9's complement of that no.

Ex :- $\underline{327} \rightarrow$ 10's complement ?

$$\begin{array}{r} 999 \\ -\ 327 \\ \hline 672 \\ +\ \ \ 1 \\ \hline 673 \end{array}$$

673 (Ans).

Subtraction using 10's Complement : (Same as 2's Complement)

(i) Add 10's complement subtrahend to the minuend.

(ii) If there is EAC, then carry will be discarded and answer is +ve.

(iii) If there is no carry, then recomplement (10's) and answer is -ve.

Ex :

$$\begin{array}{r} 277 \\ -\ 436 \\ \hline ? \end{array}$$

Step (1)

10's complement

$$\begin{array}{r} 999 \\ -\ 436 \\ \hline 563 \\ +\ \ \ 1 \\ \hline 564 \end{array}$$

(ii) Add

$$\begin{array}{r} 277 \\ 564 \\ \hline 841 \end{array}$$

No carry, re complement

$$\begin{array}{r} 999 \\ -\ 841 \\ \hline 158 \\ +\ \ \ 1 \\ \hline 159 \end{array}$$

(iii) Attach -ve sign -159 (Ans).

ex:

$$\begin{array}{r} 436 \\ -\ 277 \\ \hline \end{array}$$

(1) 10's complement of 277.

$$\begin{array}{r} 999 \\ -\ 277 \\ \hline 722 \\ +\ \ \ 1 \\ \hline 723 \end{array}$$

(2) Add,

$$\begin{array}{r} 436 \\ 723 \\ \hline 1159 \end{array}$$

discard ← 1

(3) Ans. is 159.

①

Binary Multiplication :-

(a)

```
        1 1 0 1
        1 0 1 0
      ---------
        0 0 0 0
      1 1 0 1
    0 0 0 0
  1 1 0 1
-----------------
1 0 0 0 0 0 1 0
```

(Ans)

(b)

```
        1 1 . 0 1
      × 1 1 . 0 1
      -----------
          1 1 0 1
        0 0 0 0
      1 1 0 1
    1 1 0 1
  ---------------
  1 0 1 0 . 1 0 0 1
```

(Ans)

Binary Division -

1011011 ÷ 111

```
111 ) 1011011 ( 1101
      0111
      ----
       1000
       0111
       ----
        0011
           0
        -----
        00111
        00111
        -----
            0
```

1011011 ÷ 111 = 1101 (Ans)

①

Find x radix

(i) $(431)_x = (116)_{10} \rightarrow x = ?$

Ans:

$4 \cdot x^2 + 3 \cdot x + 1 = 116$

$\Rightarrow 4x^2 + 3x + 1 = 116$

$\Rightarrow 4x^2 + 3x - 115 = 0$

$$x = \frac{-3 \pm \sqrt{9 - 4 \cdot (4) \cdot (-115)}}{2 \cdot 4}$$

$$= \frac{-3 \pm 43}{8}$$

x should be +ve, $\frac{-3+43}{8} = \frac{40}{8} = 5$ (Ans).

(ii) Q. Determine the base numbers on each case for the following operations to be correct.

(i) $\frac{14}{2} = 5$ (ii) $\frac{54}{4} = 13$ (iii) $24 + 17 = 40$

Ans

i) $\frac{(14)_x}{(2)_x} = (5)_x$

$\Rightarrow \frac{1 \cdot x^1 + 4 \cdot x^0}{2x^0} = 5 \cdot x^0 = 5$

$\Rightarrow x + 4 = 2 \times 5 = 10$

$\Rightarrow x = 10 - 4 = 6$ (Ans)

(ii) $\dfrac{(54)_x}{(4)_x} = (13)_x$

$\Rightarrow \dfrac{5 \cdot x^1 + 4}{4 \cdot x^0} = 1 \cdot x^1 + 3$

$\Rightarrow 5x + 4 = 4(x+3) = 4x + 12$

$\Rightarrow x = 12 - 4 = 8$ (Ans).

(iii) $(24)_x + (17)_x = (40)_x$

$\Rightarrow 2 \cdot x^1 + 4 + 1 \cdot x^1 + 7 = 4 \cdot x^1 + 0 \cdot x^0$

$\Rightarrow 2x + 4 + x + 7 = 4x$

$\Rightarrow 3x + 11 = 4x$

$\Rightarrow x = 11$ (Ans).

— o —

(iv) $\sqrt{(14)_x} = (3)_x \Rightarrow x = ?$

Ans $\left[\sqrt{(14)_x}\right]^2 = \left[3_x\right]^2$

$\Rightarrow (14)_x = (3_x) \cdot (3_x)$

$\Rightarrow 1 \cdot x^1 + 4 \cdot x^0 = (3 \cdot x^0) \cdot (3 \cdot x^0)$

$\Rightarrow x + 4 = 3 \cdot 3 = 9$

$\Rightarrow \boxed{x = 5}$

verify

$(14)_5$

$= 1 \times 5^1 + 4 \cdot 5^0$

$= 5 + 4$

$= 9$

$\sqrt{9} = (3)_5$

Logic Gates & Boolean Algebra

↳ Perform logical operation.

→ Logic gates are the basic building blocks of digital system. Basically there are 7 gates are available in digital electronics, those are discussed in this section.

→ Boolean algebra named after its pioneer George Boole, is the algebra of logic which has 2 states (0 & 1). Any digital ckt can be described using this algebra.

→ Logic Gates:-

Logic gates are basic building blocks of any digital ckts. They can have one or more i/p and only one o/p ckt symbol.

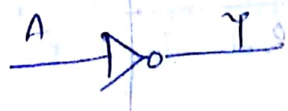
→ NOT, AND, OR } Basic gates

NAND, NOR } Universal gates

X-OR, X-NOR } Special gates

NOT Gate:- IC (7404) or 74LS04

→ One i/p one o/p.

→ 

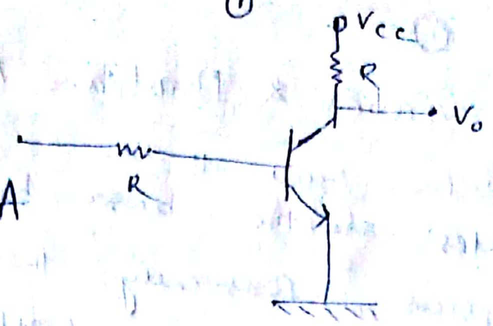
Truth table

A	$Y = \bar{A}$
0	1
1	0

→ $Y = \bar{A}$

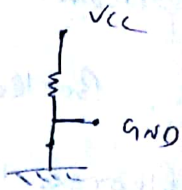
→ If i/p is low o/p will be high. If i/p is high o/p is low.

①



<u>If A = 1</u>, transistor conducts, o/p short circuited,

V_0 = GND. potential = 0.



If $\boxed{A = 1, \quad Y = 0}$

<u>If A = 0</u>, transistor is off,

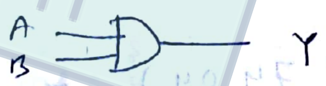
$V_0 = V_{cc} = 1$

$\boxed{A = 0, \quad Y = 1}$

AND gate :- (74 LS 08)

→ Many i/p → one o/p.

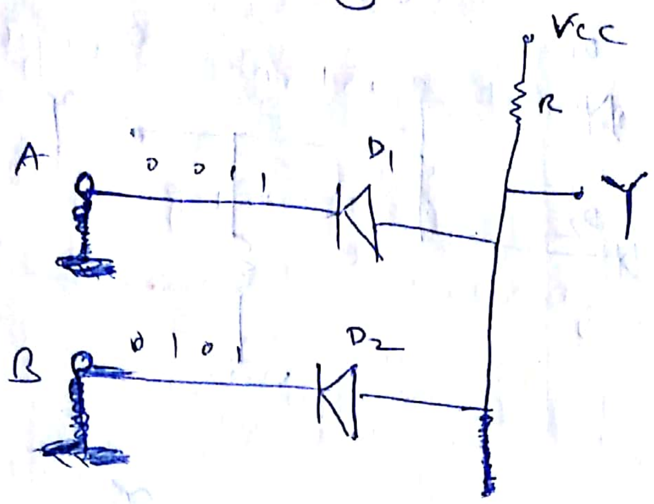
→ $Y = A \cdot B$

→ 

→

A	B	Y = A . B
0	0	0
0	1	0
1	0	0

→ The o/p of AND gate is high if all i/ps are high.



A	B	D1	D2	Y
0	0	on	on	0
0	1	on	off	0
1	0	off	on	0
1	1	off	off	1

OR Gate :- (74LS 32)

→ Many i/p → one o/p.

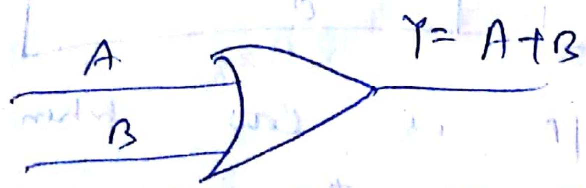
→ $Y = A + B$

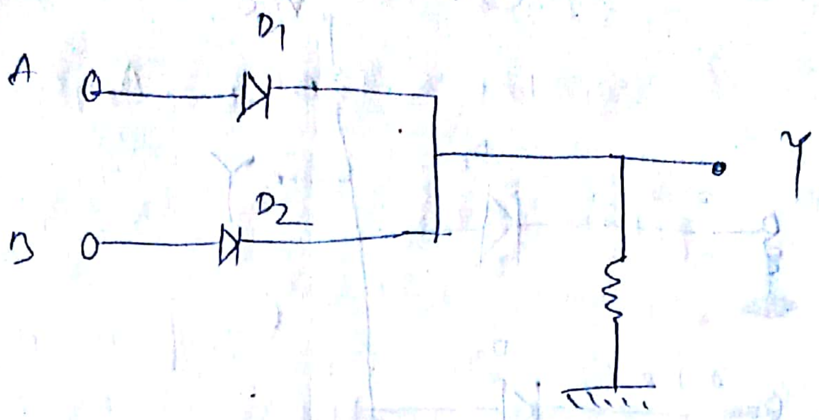
→

A	B	Y = A+B
0	0	0
0	1	1
1	0	1
1	1	1

→ The o/p of OR gate is high if any or all i/ps are high.

→





A	B	D_1	D_2	Y
0	0	off	off	0
0	1	off	on	1
1	0	on	off	1
1	1	on	on	1

NAND gate :- (74LS00)

→ Many i/p one o/p.

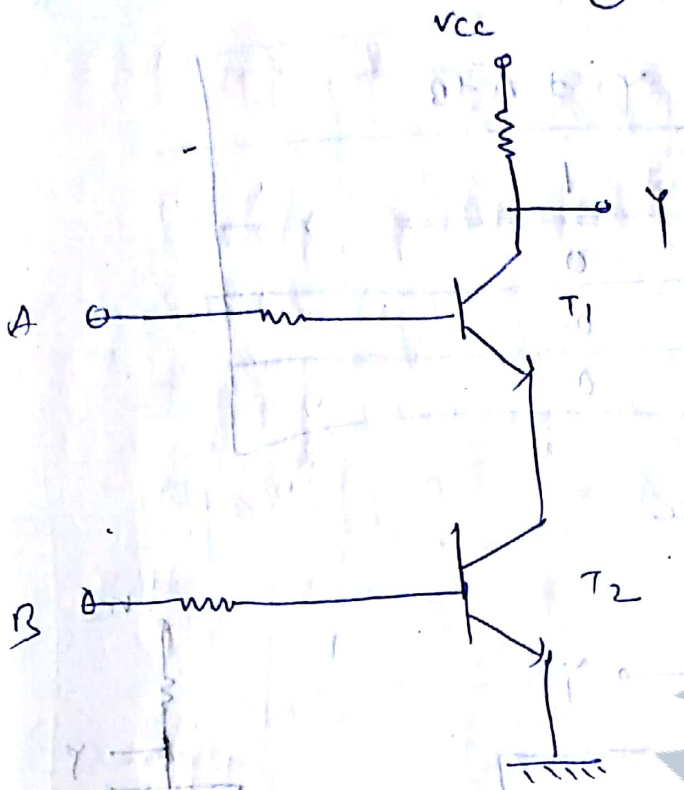
→ $Y = \overline{AB}$

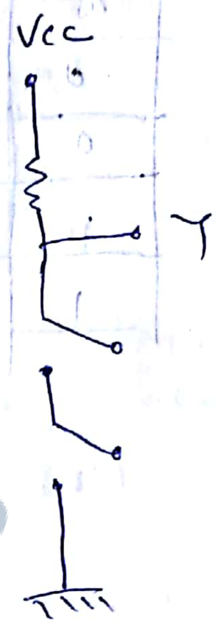
→ 

$Y = \overline{AB}$

A	B	$Y = \overline{AB}$
0	0	1
0	1	1
1	0	1
1	1	0

→ The o/p is low when all i/ps are high, otherwise the o/p is high.





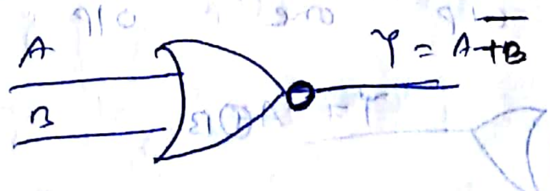
A	B	T_1	T_2	Y
0	0	off	off	1
0	1	off	on	1
1	0	on	off	1
1	1	on	on	0

NOR Gate (74 LS 02)

→ Many i/p ⟶ one o/p.

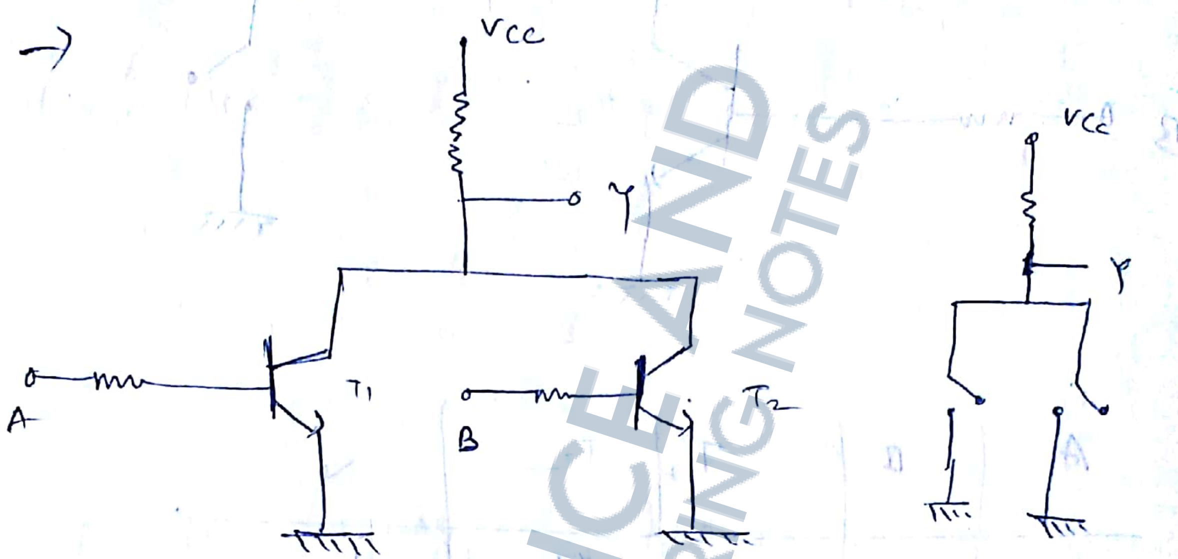
→ $Y = \overline{A+B}$

→



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

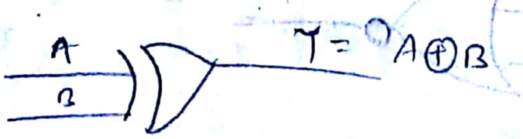
→



A	B	T1	T2	Y
0	0	off	off	1
0	1	off	on	0
1	0	on	off	0
1	1	on	on	0

EX-OR Gate :- (74 LS 86)

→ Many I/P one O/P

→ 

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

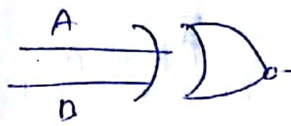
→ $\odot$

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

→ The gate produces o/p high when i/ps have odd no. of 1s.

EX-NOR (74LS266)

→ Many i/p one o/p

→ 

$Y = A \odot B$

→ $A \odot B = \cancel{AB + \bar{A}B}\ AB + \bar{A} \cdot \bar{B}$

→

A	B	$Y = A \odot B$
0	0	1
0	1	0
1	0	0
1	1	1

→ The o/p of X-NOR is high when i/p have even no. of zeros or even no. of 1s.

→ X-NOR gate is also known as 'Equivalence function'.

Universal Gates :-

→ NAND & NOR gates are called universal gates, because both can be used to implement any logic gate or any logic expression.

NAND as universal gate :-

(i) NOT gate

A	B	$\overline{AB}$
0	0	1
0	1	1
1	0	1
1	1	0

$Y = \overline{A}$

Boolean Algebra should be taught before showing universal gates. Because the o/p of any gate can be simplified by boolean algebra.

Boolean Algebra :- (Mathematics of of digital system)

✓ Duality principle :-

It states that any boolean expression has a dual form which can be obtained as following steps.

(a) Replace OR operation by AND operation.

(b) Replace AND operation by OR operation.

(c) Replace 1 by 0 & 0 by 1.

ex :- Find dual of $F = 1 . A + \bar{B} . 0 + C\bar{A} 1$

Ans :- $F = A(0+A) . (\bar{B}+1) . (C+\bar{A})$

Laws of Boolean Algebra

1) Commutative law

$A+B = B+A$

$A.B = B.A$

2) Associative law

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

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

3) Distributive law

Distributive law of addition

$A.(B+C) = AB + AC$

Distributive law of multiplication

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

Them s / Rules of Boolean Algebra :-

Them 1 (a) $0 . A = 0$ (AND)

(b) $1 + A = 1$ (OR)

Note :-

$\cdot \to$ Intersection

$+ \to$ Union.

$0 \to \phi$ = null set

$1 \to U$ = Universal set.

(b) is dual form of (a).

Them 2 (a) $1 \cdot A = A$ AND

$1 \cdot 0 = 0$
$1 \cdot 1 = 1$
$1 \cdot A = A$

(b) $0 + A = A$ OR

Them 3 (a) $A \cdot A = A$

(b) $A + A = A$

($\cdot$ → intersection
$+$ → union

Them 4 (a) $A \cdot \bar{A} = 0$

4 (b) $A + \bar{A} = 1$

Them 5 (a) $A + AB = A$

5 (b) $A \cdot (A+B) = A$

Proof

5(a) L.H.S $A + AB$

$= A(1+B)$

$= A \cdot 1$

$= A$ (R.H.S)

(b) $A \cdot (A+B)$

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

$= A + AB$

$= A(1+B)$

$= A \cdot 1$

$= A$

Them

6 (a) $A + BC = (A+B)(A+C)$

R.H.S

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

$= A.A + A.C + B.A + B.C$

$= A + \underbrace{A.C + B.A}_{A(B+C)} + B.C$

$= A[1+(B+C)] + BC$

$= A.1 + BC$ $\quad$ $\because 1+(B+C) = 1$

$= A+BC$

(b) $A.(B+C) = AB+AC$ (Distributive law)

7. $\overline{\overline{A}} = A$

Demorgan's Thm

(i) $\overline{A+B} = \overline{A}.\overline{B}$

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

i.e (i) The complement of sum of 2 or more ~~more~~ variables is equal to the product of the complement of the variable

(ii) The complement of the product of 2 or more variables is equal to the sum of the complement of of the variables.

The above thm can be proved by the help of truth table.

A	B	$\bar{A}$	$\bar{B}$	AB	A+B	$\overline{AB}$	$\bar{A}+\bar{B}$	$\overline{A+B}$	$\bar{A}\cdot\bar{B}$
0	0	1	1	0	0	1	1	1	1
0	1	1	0	0	1	1	1	0	0
1	0	0	1	0	1	1	1	0	0
1	1	0	0	1	1	0	0	0	0

From the above table, we proved that

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

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

SL No	Them^s.	Dual Rel^n
1	$A+0=A$	$A\cdot 1=A$
2	$A+\bar{A}=1$	$A\cdot\bar{A}=0$
3	$A+A=A$	$A\cdot A=A$
4	$A+1=1$	$A\cdot 0=0$
5	$\bar{\bar{A}}=A$	$\bar{\bar{A}}=A$
6	$A+B=B+A$	$A\cdot B=B\cdot A$
7	$A+(B+C)=(A+B)+C$	$A\cdot(BC)=(A\cdot B)\cdot C$
8	$A(B+C)=AB+AC$	$A+BC=(A+B)(A+C)$
9	$\overline{A+B}=\bar{A}\cdot\bar{B}$	$\overline{AB}=\bar{A}+\bar{B}$
10	$A+AB=A$	$A\cdot(A+B)=A$

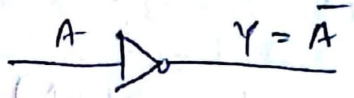
Universal gates:

→ NAND & NOR gates are called universal gates because both can be used to implement any logic gates or any logic expression.

NAND as universal gate: $\left[Y = \overline{AB} \right]$

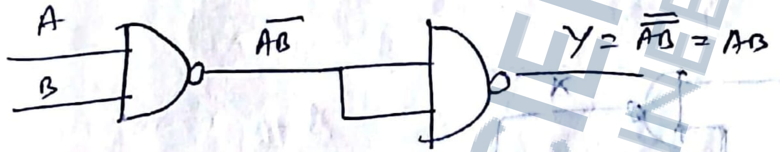
(i) NOT Gate

$$Y = \overline{A}$$

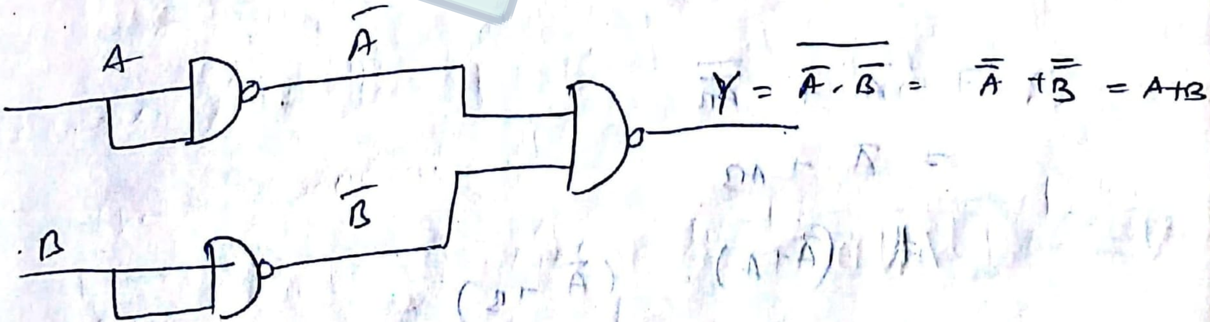


(ii) AND Gate:-

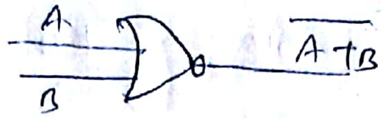
Trick:-

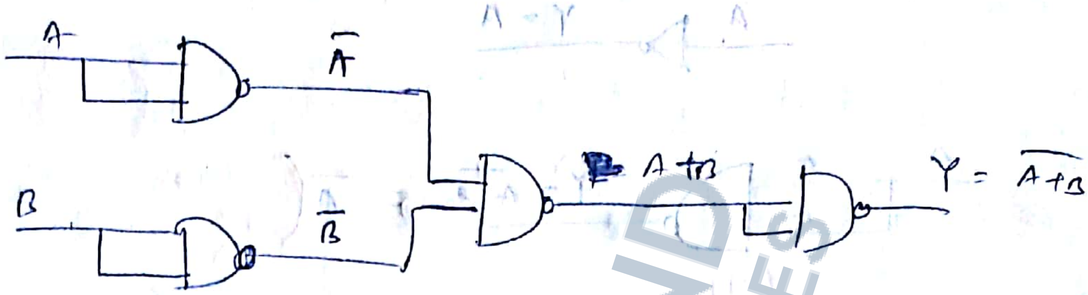


(iii) OR Gate

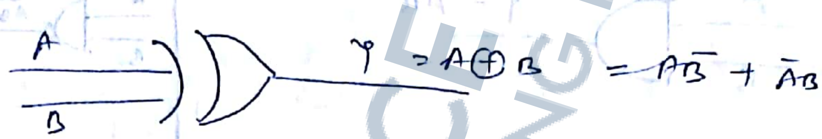


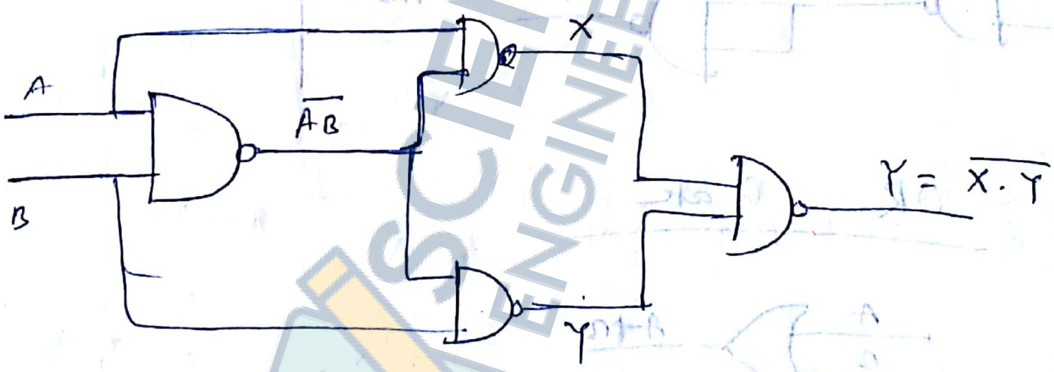
(iv) NOR gate





(v) EX-OR gate





$$X = \overline{A \cdot \overline{AB}}$$

$$= \bar{A} + \overline{\overline{AB}}$$

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

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

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

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

$$Y = \overline{\overline{AB} \cdot B}$$

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

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

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

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

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

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

$$Y = \overline{X\,Y}$$

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

$$= \overline{\overline{A}\,\overline{B} + \cancel{\overline{A}A}^{\,0} + \cancel{B\overline{B}}^{\,0} + AB}$$

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

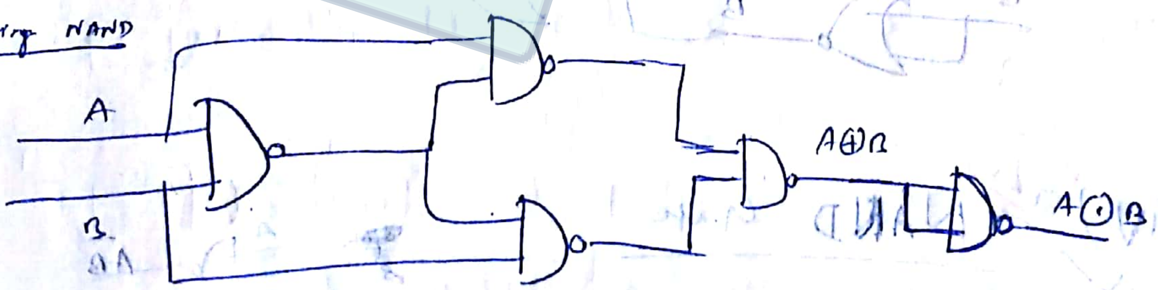
$$= \overline{A \odot B}$$

$$Y = A \oplus B$$

(Proved)

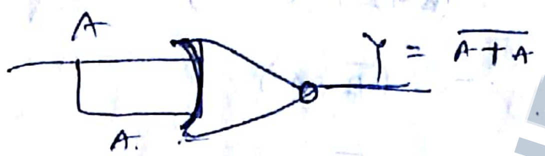
(vi) X-NOR gate

Using NAND



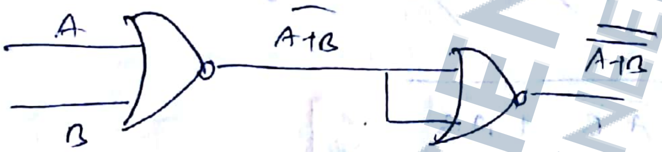
NOR as universal gate :-

(i) NOT gate :-



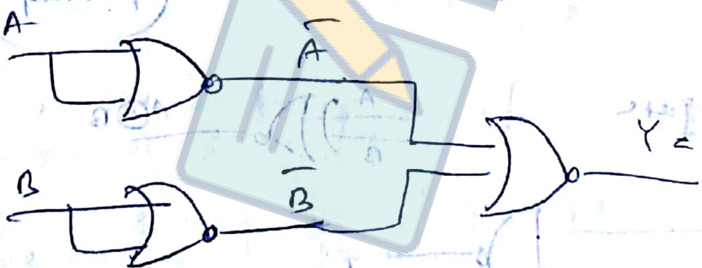
$Y = \overline{A+A} = \bar{A}$

(ii) OR Gate :-



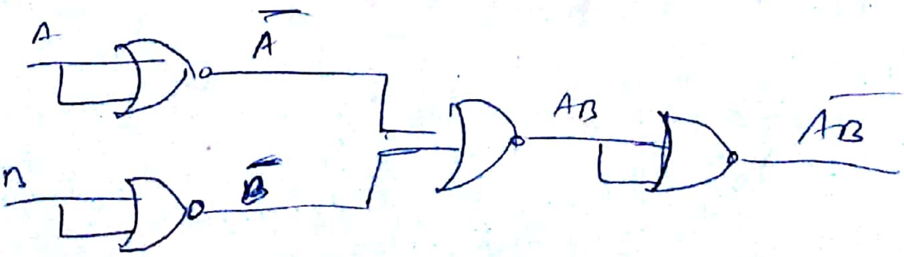
$\overline{\overline{A+B}} = A+B$

(iii) AND gate :-

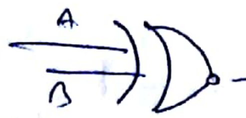


$Y = \overline{\bar{A}+\bar{B}} = \bar{\bar{A}} \cdot \bar{\bar{B}} = AB$

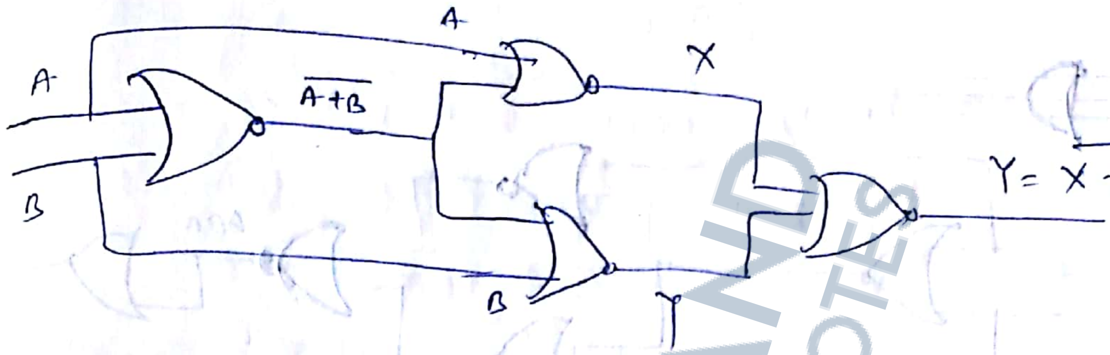
(iv) start NAND Gate



(v) Ex-NOR gate



$A \odot B = AB + \overline{A}\,\overline{B}$



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

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

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

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

$$= \overline{A}B$$

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

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

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

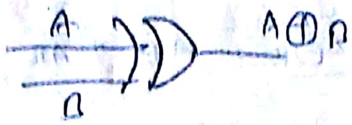
$$= \overline{B}A + \overline{B} \cdot B$$

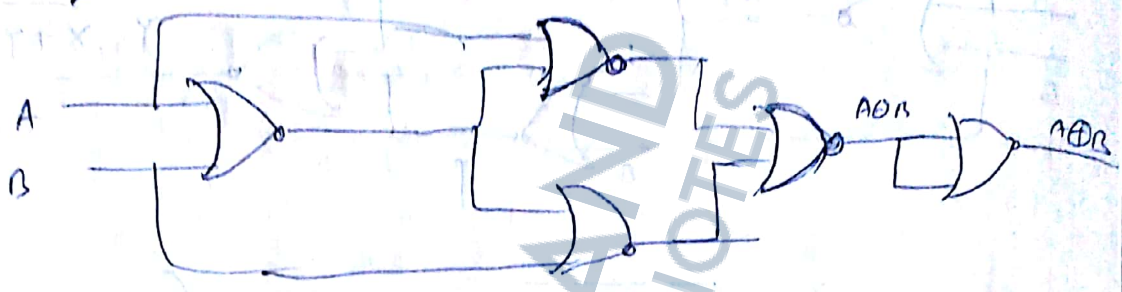
$$= \overline{B}A$$

$$Y = \overline{X+Y} = \overline{\overline{A}B + \overline{B}A} = \overline{A \oplus B} = A \odot B$$

(Proved)

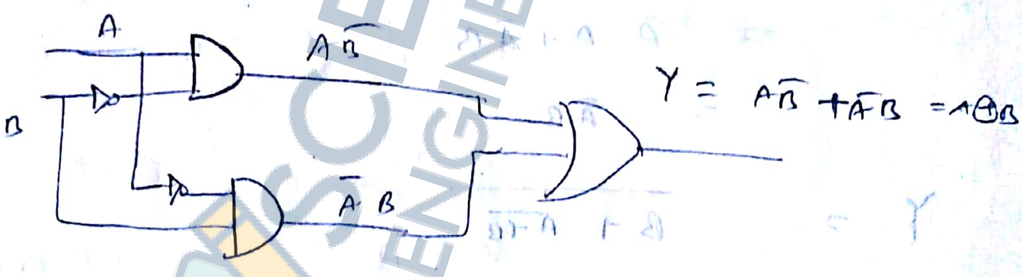
(vi) EX-OR Gate





Q. Draw X-OR gate using basic gates

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



Standard Forms of Boolean Expression:-

There are 2 forms of Boolean expressions;
These 2 forms are

(i) SOP (Sum of the Product) form.

(ii) POS (Product of the Sum) form.

SOP form

This is an expression in which AND terms (product) ORed (summed) together.

ex:- $F = \bar{A}BC + ABC + A\bar{B}C + BC$

POS form :-

This is an expression in which OR terms (sum) ANDed (product) together.

Ex:- $F = (A+B)(\bar{C}+D)(\bar{D}+A)$

Conversion of a General expression to SOP form

Any logic expression can be changed to SOP form by applying Boolean Algebra technique.

Ex-1, $A(B+CD) = AB + ACD$

Ex-2 $AB + B(CD+EF)$

$= AB + BCD + BEF$

Ex-3: $\overline{\overline{(A+B)} + C}$

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

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

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

The standard SOP form:- [Cannonical form]

A standard SOP expression is one in which all the variables in the domain appear in the each product term.

Step-I :- Multiply each non standard product term by a term made up of sum of a missing variable and its complement. This results in 2 product terms.

Step-2 :- Repeat step I, until all resulting product terms contain all variables in the domain in either complemented or uncomplemented form.

Ex ÷ $A\bar{B}C + \bar{A}\bar{B} + AB\bar{C}D$

We know
$$A + \bar{A} = 1$$
$$B + \bar{B} = 1$$
$$C + \bar{C} = 1$$
$$D + \bar{D} = 1$$

$\therefore$ $A\bar{B}C(D+\bar{D}) + \bar{A}\bar{B}(C+\bar{C}).(D+\bar{D}) + AB\bar{C}D$

~~$= A\bar{B}CD + A\bar{B}C\bar{D} +$~~

Consider

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

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

(3) $AB\bar{C}D \rightarrow$ It contains 4 variables.

$\therefore$ Finally,

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

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

Binary representation of a standard SOP form :-

Ex :- If $A\bar{B}.C.\bar{D} = 1$, what is $ABCD$?

Ans :- $A\bar{B}.C\bar{D} = 1$,

Since there product is 1, individual terms must be 1. If any one would have zero, then their product would be zero.

$A = 1$
$\bar{B} = 1 \Rightarrow B = 0$
$C = 1$
$\bar{D} = 1 \Rightarrow D = 0$

$\boxed{ABCD = 1010}$

Imp

A SOP expression (Total) is equal to 1 only if one or more of the product terms in the expressions is equal to 1.

(Because $A \cup 1 = 1$) [At least one expression = 1
ex: $0 \cup 1 = 1$, $0 \cup 1 \cup 1 = 1$]

The Standard POS form (Cannonical POS form)

A standard POS expression is one in which all the variables in the domain appear in each sum term in the expression.

Steps

1. Add to each nonstandard product term a term made up of the product of the missing variable and its complement. This results in 2 sum terms. (because we can add 0 to anything without changing its value).
2. Apply the rule $A+BC = (A+B)(A+C)$
3. Repeat step I until all resulting sum terms contain all variables in the domain in either complemented or uncomplemented form.

Ex: Convert the following Boolean expression into standard POS form.

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

Ans:

$$A+\bar{B}+C = \underbrace{A+\bar{B}+C}_{A}+\underbrace{D\bar{D}}_{BC}$$

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

$\because D\bar{D}=0$, we can add.

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

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

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

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

→ $(A+\overline{B}+\overline{C}+D)$ it contains all the variables.

∴ Finally

$(A+\overline{B}+C)(\overline{B}+C+\overline{D})(A+\overline{B}+\overline{C}+D)$ can be converted into standard POS form as:

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

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

Converting standard SOP to Standard POS

Step I :- Evaluate each product term in the SOP expression. That is, determine the binary numbers that represent the product term.

Step 2 :- Determine all of the binary number not included in the evaluation of step I.

Step 3 :- Write the equivalent sum term for each binary number form step 2 and expression in

POS form. 0

Ex:-

$\bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}C + ABC$

Ans: SOP

000 + 010 + 011 + 101 + 111

i.e. 0, 2, 3, 5, 7

So left 1, 4, 6

001, 100, 110

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

POS

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

(In POS form $0 \to A$, $1 \to \bar{A}$

In SOP form $0 \to \bar{A}$, $1 \to A$)

Q) Convert POS → SOP $(A+B+\bar{C})(\bar{A}+B+C)(\bar{A}+\bar{B}+C) \to$? SOP

Q) Convert $\bar{A}B + A\bar{B}$ = ? into POS.

→ (A POS expression is equal to 0 only if one or more of sum terms in the expression is equal to 0)

Ex: Determine the binary value of the variables for which following POS expression is equal to 0.

$(A+B+C+D)\,(A+\bar{B}+\bar{C}+D)\,(\bar{A}+\bar{B}+\bar{C}+\bar{D})$

Ans :- The total expression $= 0$, if any one of them is 0.

If $A+B+C+D = 0 \Rightarrow A=0,\ B=0,\ C=0,\ D=0$

If $A+\bar{B}+\bar{C}+D = 0 \Rightarrow A=0,\ B=1,\ C=1,\ D=0$

If $\bar{A}+\bar{B}+\bar{C}+\bar{D} = 0 \Rightarrow A=1,\ B=1,\ C=1,\ D=1$.

Min term :-

~~Min~~ Minterm is a product terms which contain all the variables either in true form or complement form.

$$F(A,B,C) = ABC + \bar{A}BC + B\bar{C} + A\bar{C}$$

min term.

N.B. - For n i/p variable, we have 2^n min terms.

$4 \rightarrow 2^4 \rightarrow 16$ minterms.

Minimal SOP form :- It is an SOP expression containing min product term which can't be further simplified.

Max term :-

Maxterm is sum term which contains all the variables given in the expression either in true form or in complement form.

ex. $F(A,B,C) = (A+B+C)(B+\bar{C})$

Minimal POS form :- It is a POS expression containing minimum sum term which can't be further simplified.

Min term & max term representation

For 3 i/p binary variable function, we have $2^3 = 8$ min terms as well as max terms.

x y z	min term (m_i)	Max term (M_i)
0 0 0	$\bar{x}\bar{y}\bar{z}$ (m_0)	$(x+y+z)$ M_0
0 0 1	$\bar{x}\bar{y}z$ (m_1)	$(x+y+\bar{z})$ M_1
0 1 0	$\bar{x}y\bar{z}$ (m_2)	$(x+\bar{y}+z)$ M_2
0 1 1	$\bar{x}yz$ (m_3)	$(x+\bar{y}+\bar{z})$ M_3
1 0 0	$x\bar{y}\bar{z}$ (m_4)	$(\bar{x}+y+z)$ M_4
1 0 1	$x\bar{y}z$ (m_5)	$(\bar{x}+y+\bar{z})$ M_5
1 1 0	$xy\bar{z}$ (m_6)	$(\bar{x}+\bar{y}+z)$ M_6
1 1 1	xyz (m_7)	$(\bar{x}+\bar{y}+\bar{z})$ M_7

Short

→ For n input variables we have 2^n min as well as max term.

→ Min term is a product term whose value equal to 1 i.e denoted by

$$\sum_{i=0}^{2^n-1} m_i = 1$$, i.e either of them 1.

e.g

$$f(A,B,C) = \Sigma(0,2,4,6)$$

i.e $= 000 + 010 + 100 + 110$

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

→ Find the truth table?

If given

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

→ Max term is a sum term whose value equal to 0, i.e denoted by

$$\prod_{i=0}^{2^n-1} M_i = 0$$, i.e either of them is 0.

Find the truth table.

Given

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

e.g

$$F(A,B,C) = \Pi(0,2,4,7)$$

$= (000)(010)(100)(111)$

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

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

→ Min term & Max term are complement to each other.

$$\Sigma(0,2,4,6) = \Pi(1,3,5,7)$$

→ Minimization,

Ex - 1 (Simplify the following expression)

(i) $AB\bar{C} + A\bar{B}C + ABC$

Ans :

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

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

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

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

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

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

$$= A[(1)((B + C))]$$

$$= A(B + C)$$

$$= AB + AC \quad \text{(Ans)}$$

→ (Minimized expression)

formula

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

(ii) $(A + \bar{B} + C)(\bar{A}B\bar{C}D)$

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

$$= 0 + 0 + 0$$

$$= 0$$

$A\bar{A} = 0$

$B\bar{B} = 0$

$C\bar{C} = 0$

(iii) $\bar{x}\bar{y}z + yz + xz$

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

$$= z[x + y + \bar{x}\bar{y}]$$

$$= z[(x + y) + \overline{x + y}]$$

~~$= Z \, [\ldots]$~~

$= Z \cdot [1]$

$= Z$ (Ans)

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

(iv)

$$AB + \bar{B}C + AC$$

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

$= AB + \bar{B}C + AC \cdot 1$

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

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

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

$= AB + \bar{B}C$

See $AB + \bar{B}C + AC = AB + \bar{B}C$,

one term is eliminated (i.e. $\underline{AC}$)

Trick :- See one term is complemented i.e. $\bar{B}C$ for ans. Keep the complemented term & its [redacted] corresponding uncomplemented term, eliminate the other one.

$AB + \bar{B}C + AC$

$\bar{B}C \longrightarrow$ Complemented term

$AB \longrightarrow$ Corresponding uncomplemented term.

Keep these two.

Ans $\rightarrow AB + \bar{B}C$.

$\cancel{= Z [\ldots]}$ ∵ $A + \bar{A} = 1$

$= Z \cdot [1]$

$= Z$ (Ans)

(iv)

$AB + \bar{B}C + AC$

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

$= AB + \bar{B}C + AC \cdot 1$

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

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

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

$= AB + \bar{B}C$

See $AB + \bar{B}C + AC = AB + \bar{B}C$,

one term is eliminated (i.e. $\underline{AC}$)

Trick :- See one term is complemented i.e $\bar{B}C$ for ans. Keep the complemented term & its corresponding uncomplemented term, eliminate the other one.

$AB + \bar{B}C + AC$

$\bar{B}C \rightarrow$ Complemented term

$AB \rightarrow$ Corresponding uncomplemented term.

Keep these two.

Ans $\rightarrow AB + \bar{B}C$.

(v) Try

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

Ans:

$$AB + \bar{A}C + BC \cdot 1$$

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

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

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

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

(v) Try

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

(vi) Prove that

$$ABCD + AB\overline{CD} + ABC\bar{D} + AB\bar{C}D + ABCDE + AB\bar{D}E = AB$$

Ans:-

$$ABCD + AB(\overline{CD}) + ABC\bar{D} + AB\bar{C}D + ABCDE + AB\bar{D}E$$

$$= ABC(D + \bar{D}) + AB(\bar{C} + \bar{D}) + AB\bar{C}D + ABCDE + AB\bar{D}E$$

$$= ABC + AB\bar{C} + AB\bar{D} + AB\bar{C}D + ABCDE + AB\bar{D}E$$

$$= ABC + AB\bar{C}(1 + D) + AB\bar{D}(1 + E) + ABCDE$$

$$= ABC + AB\bar{C} + AB\bar{D} + ABCDE$$

$= ABC(1+DE) + AB\bar{C} + AB\bar{D}$

$= ABC + AB\bar{C} + AB\bar{D}$

$= AB(C+\bar{C}) + AB\bar{D}$

$= AB + AB\bar{D}$

$= AB(1+\bar{D})$

$= AB$ (Ans)

Prove that

(ii) $\overline{AB+BC+CA} = \cancel{\bar{A}\bar{B}+\bar{B}\bar{C}+\bar{A}\bar{C}}$

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

Proof:

$\overline{AB+BC+CA}$

$= \overline{AB}\cdot\overline{BC}\cdot\overline{CA}$ (Demorgan's thm $\overline{A+B} = \bar{A}\cdot\bar{B}$)

$= \underbrace{(\bar{A}+\bar{B})\cdot(\bar{B}+\bar{C})}_{\text{multiply}}\cdot(\bar{C}+\bar{A})$ (Again Demorgan's thm $\overline{A\cdot B} = \bar{A}+\bar{B}$)

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

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

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

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

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

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

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

$A\cdot A = A$

$A+A = A$

$\rightarrow \overline{AB + BC + CA} = \bar{A}\bar{B} + \bar{B}\bar{C} + \bar{A}\bar{C}$ (proved)

$\because A + A = A$

Alternative Method for Converting SOP → POS

1) Take the complement of given SOP expression & expand using Demorgan's theorem.

2) Simplify the above expression using theorem of Boolean algebra.

3) Take once again complement of the simplified expression obtained in step 2, to get the POS form.

Ex: $F = A\bar{B} + \bar{A}B$

Step I $\quad \bar{F} = \overline{A\bar{B} + \bar{A}B}$

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

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

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

Step II $\quad \bar{F} = \cancel{\bar{A}A}^{0} + \bar{A}\bar{B} + AB + \cancel{B\bar{B}}^{0}$

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

Step III $\quad \bar{\bar{F}} = \overline{AB + \bar{A}\bar{B}}$

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

$A = A \cdot A$

$A = A + A$

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

$\Rightarrow \boxed{F = (\bar{A} + \bar{B})(A + B)}$ POS form

POS → SOP

Step 1 :- Expand the POS expression

Step 2 :- Simplify using theorems of Boolean algebra.

Ex : $F = (\bar{A}+\bar{B})(A+B)$

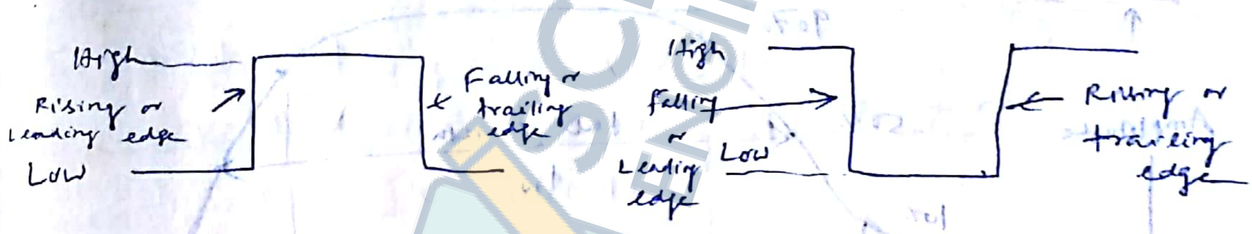
Step 1 :

$F = \cancel{\bar{A}.A}^{0} + \bar{A}.B + \bar{B}.A + \cancel{B\bar{B}}^{0}$

Step 2 :- $= \bar{A}B + A\bar{B}$ (SOP form)

Q) Design $A+B\bar{C}$ using NAND/NOR.

Digital waveform :-



(a) +ve going pulse (b) -ve going pulse.

→ A positive going pulse is generated when the voltage (current) goes from its normally low level to its high & then backs to low level.

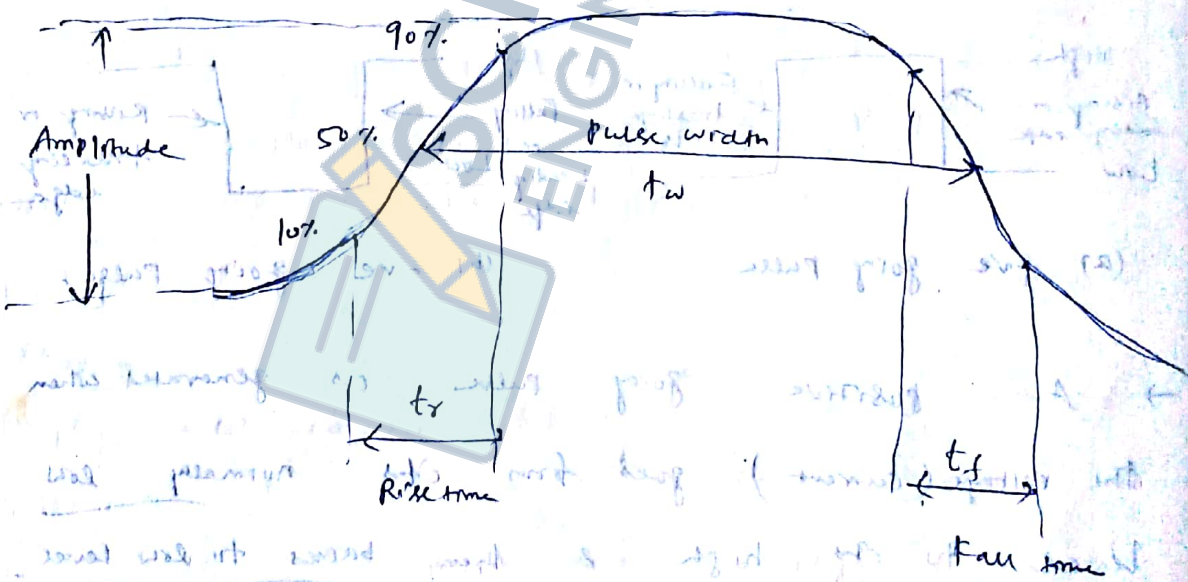
→ A -ve going pulse is generated when the voltage goes from its normally High level to its low level & back to high level

t_r (Rise time)

The time required for the pulse to go from its low level to high level, is called rise time (t_r). Practically, it is the measure of time from 10% of pulse amplitude to 90% of pulse amplitude.

t_f (Fall time)

The time required for the transition from High level to Low level is called the fall time (t_f). Practically it is the measure of time from 90% to 10% of pulse amplitude.



$$\text{Duty cycle} = \frac{t_w}{T} = \frac{\text{Pulse width}}{\text{Time period}}$$

$f = \frac{1}{T}$ i.e frequency $= \frac{1}{\text{Time period}}$

(c)

Pulse Width :-

It is a measure of the duration of the pulse and is often defined as the time interval between 50% point on the rising & falling edge.

20

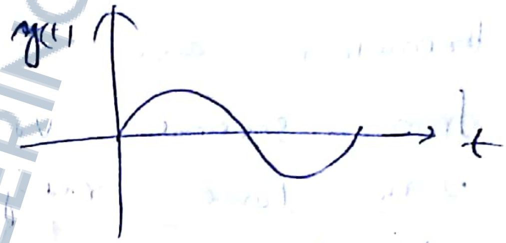
2010 BPUT

1) (a) Analog, digital, discrete time signal ?

Analog :-

The analog signal is that type of signal which varies continuously with certain interval of time.

e.g. $y(t) = \sin t$



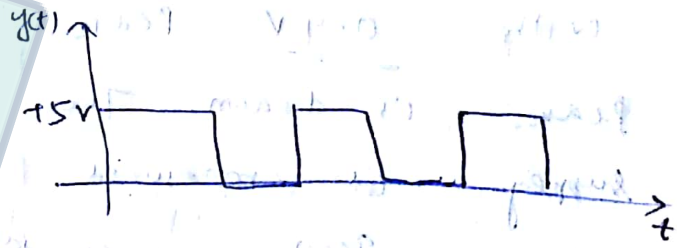
Digital :-

Digital signal is that type of signal which represented as sequence of numbers i.e magnitude at an constant of time.

e.g :- Pulse train shown.

High = +5V (called logic 1)

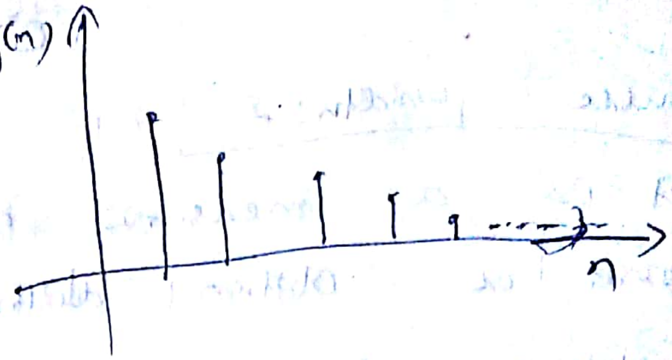
low = 0V (called logic 0)



Discrete-Time Signal :-

It is defined at certain specific values of time. The time constant need not be equidistant in practice. They are usually taken at equally ~~constant at~~ spaced intervals for computational convenience and mathematical help.

$$y(n) = \begin{cases} 0.2^n, & \text{for } n \geq 0 \\ 0, & \text{otherwise} \end{cases}$$



n = integer.

b) P-N junction diode made up of which material (Si, Ge, GaAs) will have highest thermal stability? Why?

Ans:- P-N junction diode made up of Si material will have highest thermal stability, because Silicon diodes operated in Avalanche breakdown are available with maintaining voltages from several volts to several hundred volts with power rating up to 50 watts. They can operates in high temperature.

Internal:- 4) An amplifier operating from ±5V supplies provides a 3.2V peak sine wave across 50Ω load; when provided with 0.4V peak i/p from which 2.0 mA peak is drawn. The avg current in each supply is measured to be 30 mA. Find voltage gain, current gain & power gain expressed in dB as well as supply power, amplifier dissipation & amplifier efficiency.

Logic Gates & Boolean Algebra

1) BPUT 2018 **Prove that**

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

Ans ⇒ $\overline{AB+AC} + A\bar{B}C$

$= \overline{AB}\cdot\overline{AC} + A\bar{B}C$ ∵ $\overline{A+B} = \bar{A}\cdot\bar{B}$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\overline{AB+AC} + A\bar{B}C$$

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

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

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

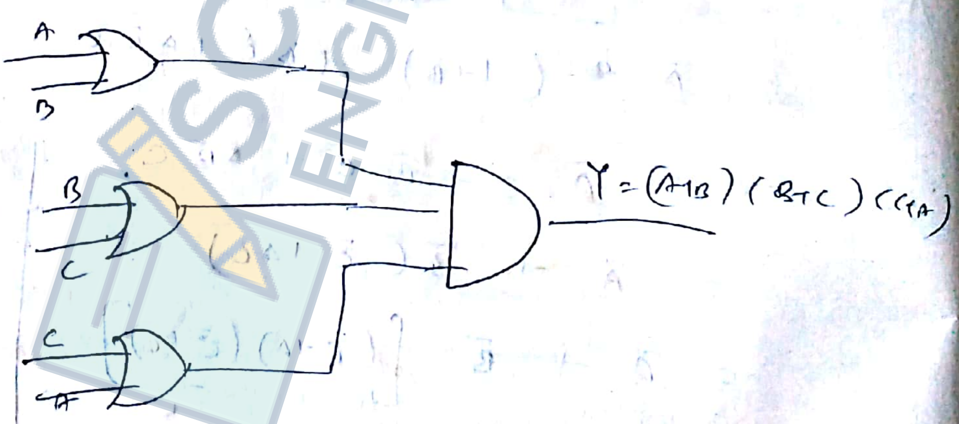
(Proved)

2) (a) Draw the logic diagram for the boolean expression

$$Y = (A+B).(B+C)\,(C+A)$$

(b) Simplify the above equation & draw the logic diagram for simplified expression.

Ans :-



(ii) $(A+B)(B+C)\,(C+A)$

$$= (AB+AC+B.B+BC).(C+A)$$

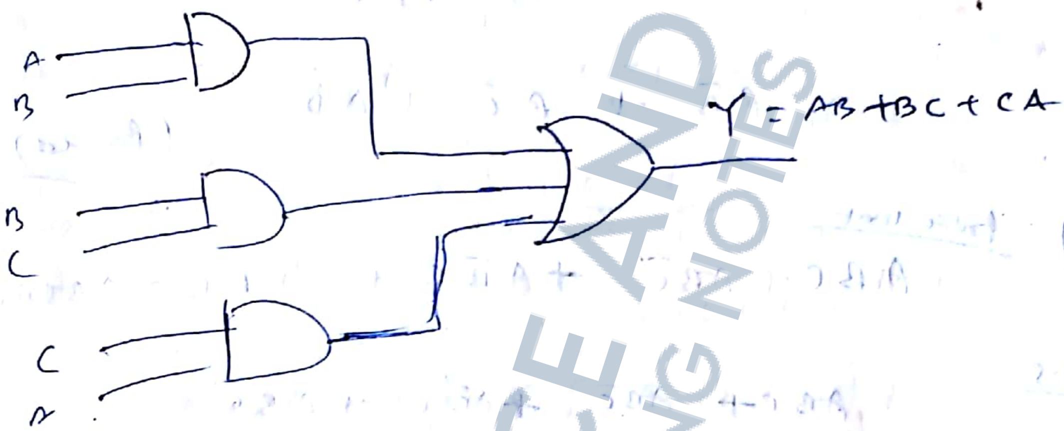
$$= (AB+AC+B+BC)\,(C+A)$$

$$= (B(1+A+C)+AC)\,(C+A)$$

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

$= BC + BA + A.C.C$ | $C.C = C$

$= AB+BC+AC$



3) Prove that

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

Proof

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

$= A\bar{B} + AB[\bar{D}+\bar{C}D]$

$= A\bar{B} + AB[(\bar{D}+\bar{C})(\bar{D}+D)]$

$= A\bar{B} + AB[(\bar{C}+\bar{D})]$

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

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

$= A[(\bar{B}+B)(\bar{B}+\bar{C})] + AB\bar{D}$

$= A[\bar{B}+\bar{C}] + AB\bar{D}$

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

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

$\Rightarrow$ $A\bar{B} + AB\bar{D} + AB\bar{C}D$

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

$= A\left[(\bar{B}+B)(\bar{B}+\bar{D})\right] + A\bar{C}$

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

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

4) Prove that

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

L.H.S

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

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

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

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

$= A\left[(B+\bar{B})(B+C)\right] + \bar{A}BC$

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

$= AB + AC + \bar{A}BC$

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

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

$= AB + C(A+B)$

$= AB + AC + BC$

$= R.H.S$ (Proved)

1 Find the complement of the function given below and Implement using logic gates.

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

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

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

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

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

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

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

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

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

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

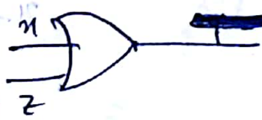
$$= \underbrace{x + yz + \bar{x}z}$$

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

$$= (x+z) + yz$$

$$= x + z(1+y)$$

$$\bar{F} = x+z$$



$\bar{F} = x+z$

6) Simplify the following expression

$F(x,y,z) = (x+y)\left[\overline{\bar{x}(\bar{y}+\bar{z})}\right] + \overline{xy} + \bar{x}.\bar{z}$

$(x+y)\left[\overline{\bar{x}(\bar{y}+\bar{z})}\right] + \overline{xy} + \bar{x}.\bar{z}$

$= (x+y)\left[\bar{\bar{x}} + \overline{(\bar{y}+\bar{z})}\right] + \overline{xy} + \bar{x}.\bar{z}$ $\quad \overline{AB} = \bar{A}+$

$= (x+y)(x + \bar{\bar{y}}.\bar{\bar{z}}) + \overline{xy} + \bar{x}.\bar{z}$

$= (x+y)(x+yz) + \overline{xy} + \bar{x}.\bar{z}$

$= (x.x + xyz + xy + yyz) + \overline{xy} + \bar{x}.\bar{z}$

$= (x + xy + xyz + yz) + \overline{xy} + \bar{x}.\bar{z}$

$= [x(1+y+yz) + yz] + \overline{xy} + \bar{x}.\bar{z}$

$= (x+yz) + \overline{xy} + \bar{x}.\bar{z}$

$= x + \bar{x}\bar{z} + yz + \overline{xy}$

$= (x+\bar{x})(x+\bar{z}) + yz + \overline{xy}$

$= x + \bar{z} + yz + \bar{x} + \bar{y}$

$= \bar{x} + \bar{y} + \bar{z} + yz$

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

$= \bar{x} + \bar{y} + \bar{z} + y$

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

7) If $\bar{A}B + A\bar{B} = C$

Then prove $\bar{A}C + A\bar{C} = B$

Proof : $A\bar{C} + \bar{A}C$

$$= A\left(\overline{\bar{A}B + A\bar{B}}\right) + \bar{A}\left(\bar{A}B + A\bar{B}\right) \qquad \because C = \bar{A}B + A\bar{B}$$

$$= A\left(\overline{\bar{A}B} \cdot \overline{A\bar{B}}\right) + \bar{A}\cdot\bar{A}\cdot B + \cancel{\bar{A}\cdot A\cdot B}^{\,0}$$

$$= A\left((\bar{\bar{A}} + \bar{B})(\bar{A} + \bar{\bar{B}})\right) + \bar{A}B$$

$$= A\left((A + \bar{B})(\bar{A} + B)\right) + \bar{A}B$$

$$= A\left[\cancel{A\bar{A}}^{\,0} + AB + \bar{B}\bar{A} + \cancel{\bar{B}B}^{\,0}\right] + \bar{A}B$$

$$= A\left[AB + \bar{A}\bar{B}\right] + \bar{A}B$$

$$= AAB + \cancel{A\bar{A}\bar{B}}^{\,0} + \bar{A}B$$

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

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

$$= B \quad \text{(Proved)}$$

8) Simplify the ~~one~~ logic expression & draw the logic diagram for the simplified expression.

$$(\bar{x} + xy\bar{z}) + (\bar{x} + xy\bar{z})(x + \bar{x}\bar{y}z)$$

$$(\bar{x} + xy\bar{z}) + (\bar{x} + xy\bar{z})(x + \bar{x}\bar{y}z)$$

$$= (\bar{x} + xy\bar{z})\left[1 + (x + \bar{x}\bar{y}z)\right]$$

$$= \underbrace{\bar{x}}_{A} + \underbrace{x}_{B}\underbrace{y\bar{z}}_{C}$$

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

$$= \cancel{\bar{x} + y\bar{z}}$$

$$= \bar{x} + y\bar{z} \quad \text{(Ans)}$$

9) Given $F = A(B+\bar{C}) + D$

express as

(a) Minimal SOP (ii) minimal POS

Ans:

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

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

The above expression is itself minimal SOP, since further reduction is not possible.

∴ (i) Minimal SOP $= AB + A\bar{C} + D$.

(ii)

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

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

$$= \overline{AB + A\bar{C}} \cdot \bar{D}$$

$$\Rightarrow \overline{F} = \overline{AB} \cdot \overline{A\bar{C}} \cdot \overline{D}$$

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

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

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

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

$$= [\bar{A}(1+C+\bar{B}) + \bar{B}C] \cdot \bar{D}$$

$$= [\bar{A} + \bar{B}C]\,\bar{D}$$

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

$$\overline{\overline{F}} = \overline{\bar{A}\bar{D} + \bar{B}C\bar{D}}$$

$$\Rightarrow F = \overline{\bar{A}\bar{D}} \cdot \overline{\bar{B}C\bar{D}} \qquad \Big|\ \overline{A+B} = \bar{A}\cdot\bar{B}$$

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

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

= Minimal POS form.

Q) Minimize the following expression

ex: $F = \Sigma(4,5,6,7)$

Ans:

P.T.O

$F = \Sigma(4, 5, 6, 7)$

$= \cancel{100 + 0}$

$= 100 + 101 + 110 + 111$

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

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

$= A\bar{B} + AB$

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

$\boxed{F = A}$

11) Convert following expression into POS form.

$F = \Sigma(0, 1, 2, 6)$

Ans : $F = \pi(3, 4, 5, 7)$

$(\because$

$F = m_0 + m_1 + m_2 + m_6$

$\cancel{\bar{F} = \overline{m_0 + m_1 + m_2 + m_6}}$

$\cancel{= \bar{m}_0 \cdot \bar{m}_1 \cdot \bar{m}_2 \cdot \bar{m}_6}$

$\bar{F} = m_3 + m_4 + m_5 + m_7$

$\bar{\bar{F}} = \overline{m_3 + m_4 + m_5 + m_7}$

$= \bar{m}_3 \cdot \bar{m}_4 \cdot \bar{m}_5 \cdot \bar{m}_7$

$F = M_3 \cdot M_4 \cdot M_5 \cdot M_7$

$F = 011 \cdot 100 \cdot 101 \cdot 111$

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

12) Simplify the following boolean expression

$F = \sum(0, 2, 4, 6)$

~~F = 0000 + 0010 + 0100 +~~

$F = 000 + 010 + 100 + 110$

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

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

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

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

$= \bar{C}$ (Ans).

13) Realize the following expression using
(a) Basic logic gates.
(b) Universal logic gates.

~~$F = ABC + \bar{A}\bar{B}\bar{C} + \bar{A}\bar{C}$~~

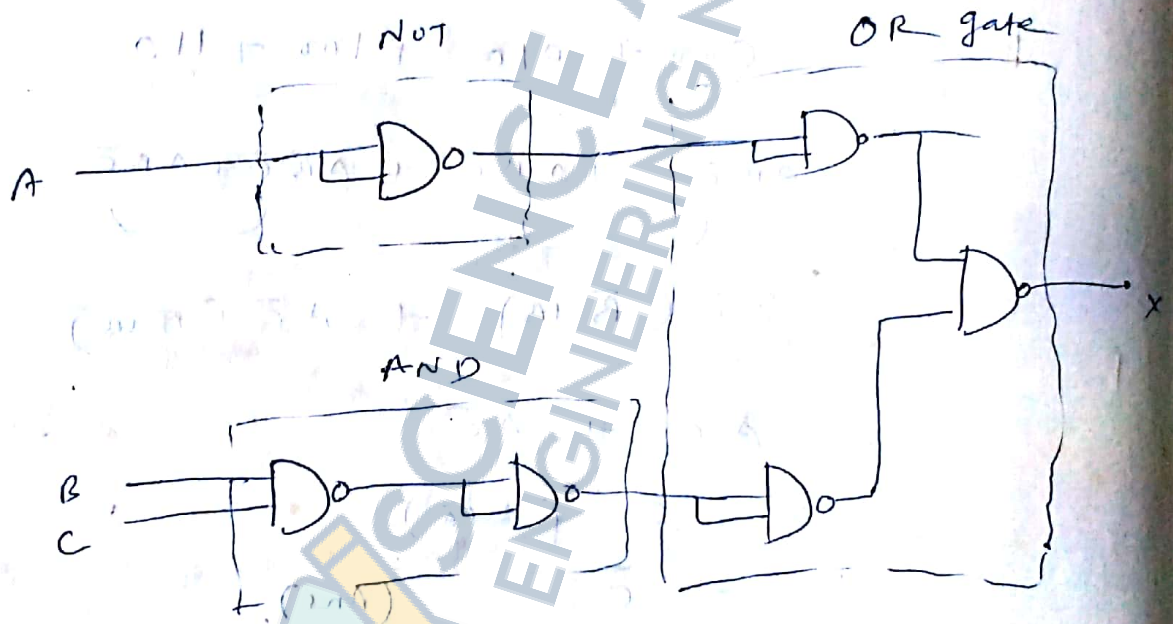
$F = \bar{A} + BC$

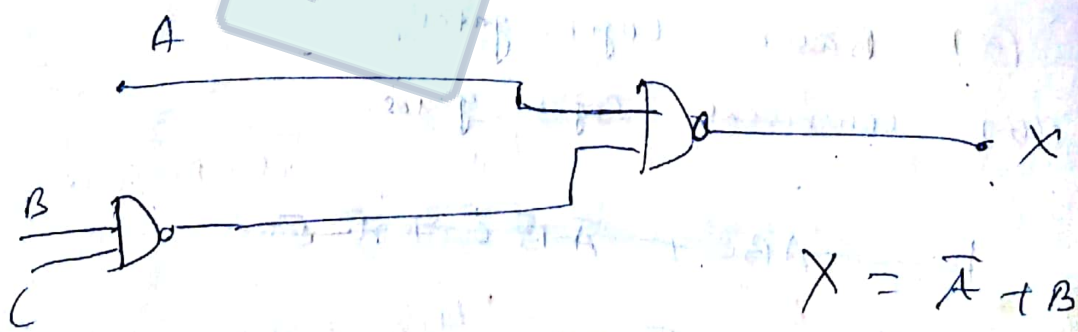
(a) using Basic logic gates:-

(i)

$$F = \bar{A} + BC$$

Replacing each gate by corresponding NAND gate

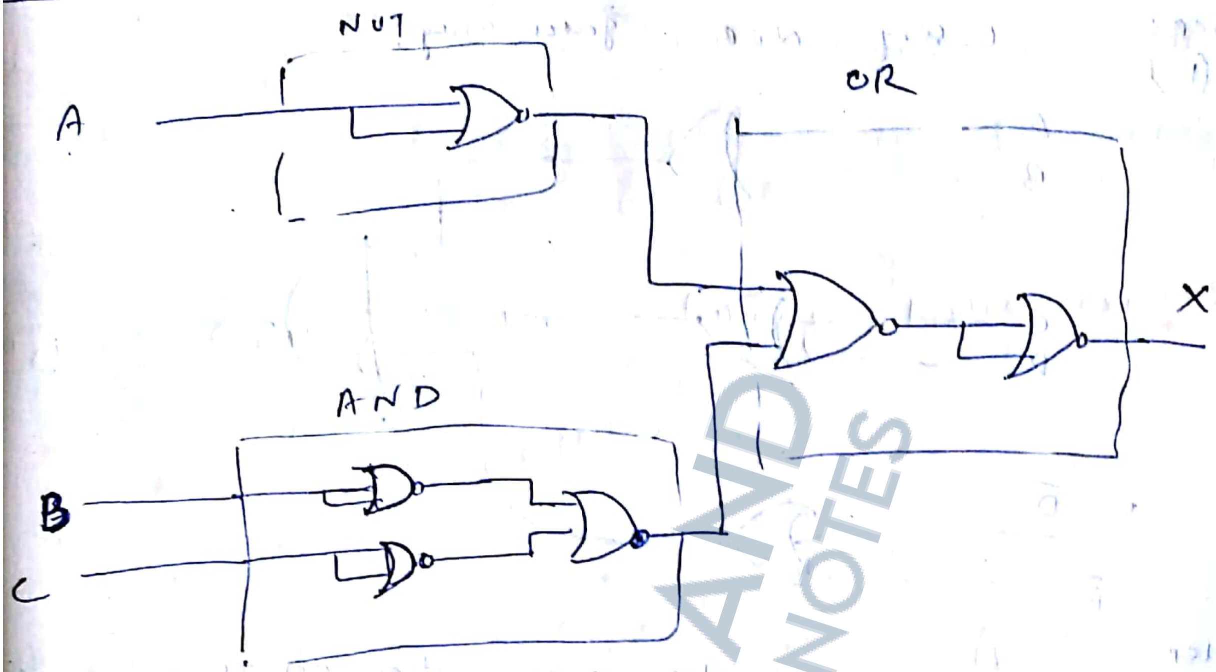




$$X = \bar{A} + BC$$

($\because$ 2 Not gates cancel each other $\bar{\bar{A}} = A$)

Replacing each gate by NOR gate



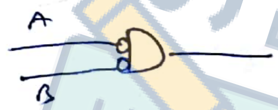
$$X = \bar{A} + BC$$

Note:-

→ Bubbled AND = NOR

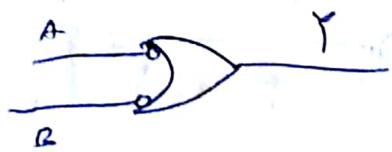
→ Bubbled OR = NAND

Bubbled AND :-



$Y = \bar{A} \cdot \bar{B} = \overline{A+B} = NOR$

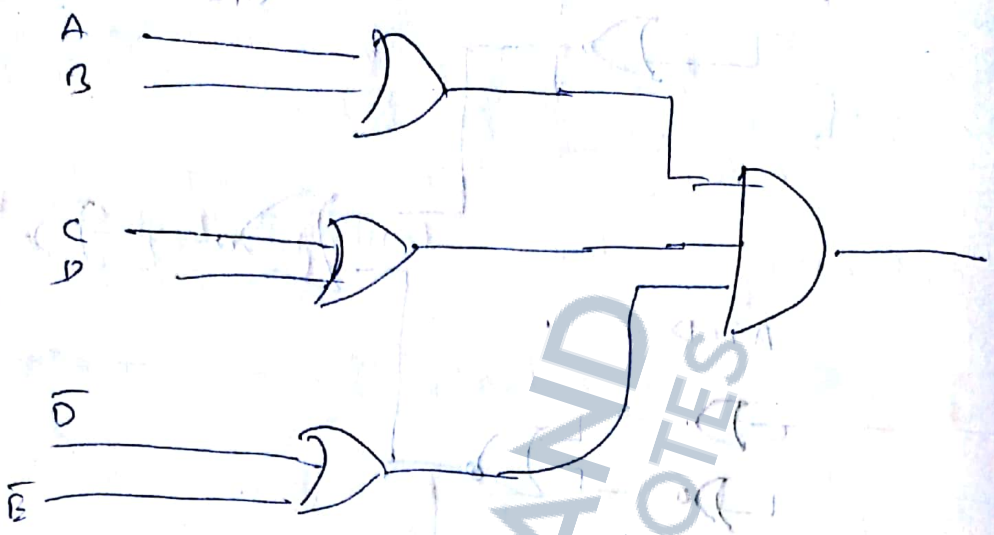
Bubbled OR



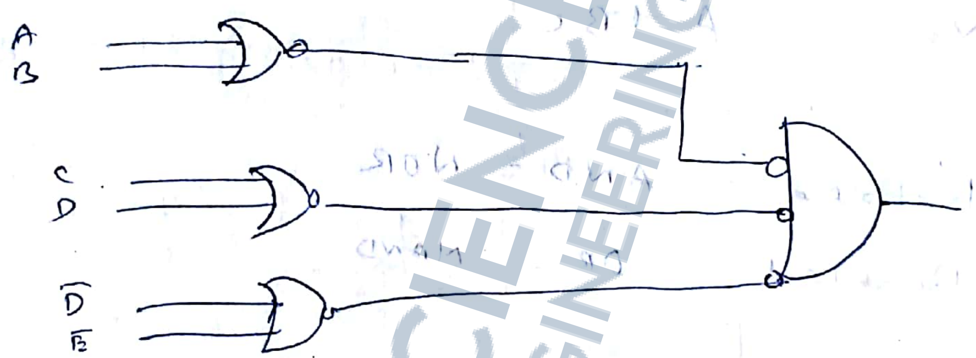
$Y = \bar{A} + \bar{B} = \overline{AB} = NAND.$

14) Implement $F = (A+B)\cdot(C+D)\cdot(\bar{D}+E)$

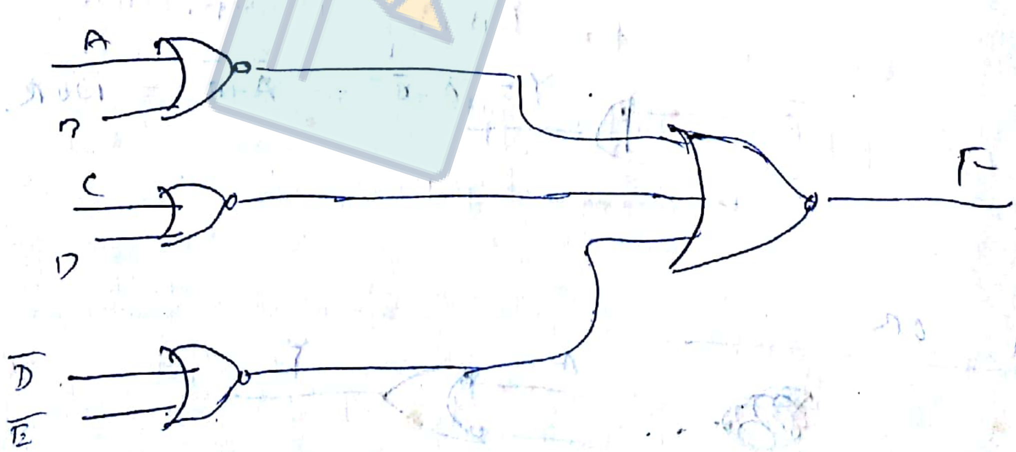
Step (1) Using NOR gates only!



Step (2) Adding ~~Not~~ 2 NOT gate does not change ($\bar{\bar{A}} = A$)



(3) Bubbled AND = NOR



Another Method of Using NAND and NOR

Q) Implement $A+B\bar{C}$ Using Minm NAND gates & Minimum NOR gates.

Ans :-

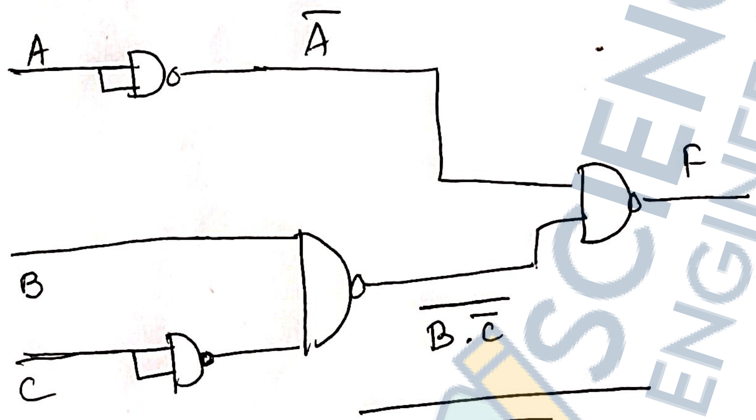
Minimum NAND gates

(express in product term bar)

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

$$F = \bar{\bar{F}} = \overline{\overline{A + B\bar{C}}}$$

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



$$F = \overline{\bar{A} \cdot \overline{B\bar{C}}}$$

Minimum NOR gates

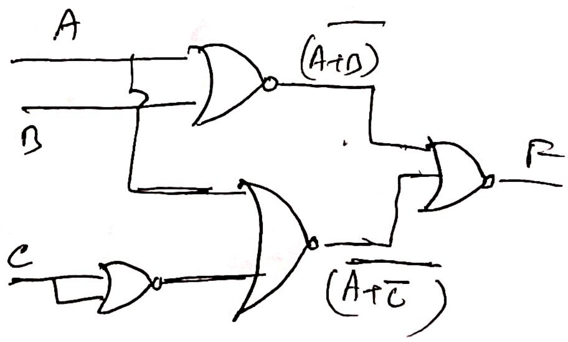
(express in sum term bar)

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

$$F = \bar{\bar{F}} = \overline{\overline{A + B\bar{C}}}$$

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

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



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

Digital Ckts. & Design

Digital ckts

- Combinational logic ckts
- Sequential logic ckts.

→ A Combinational ckt consists of logic gates whose o/p at any time are determined from present combination of i/ps.

→ A Combinational ckts performs an operation that can be specified logically by a set of Boolean functions.

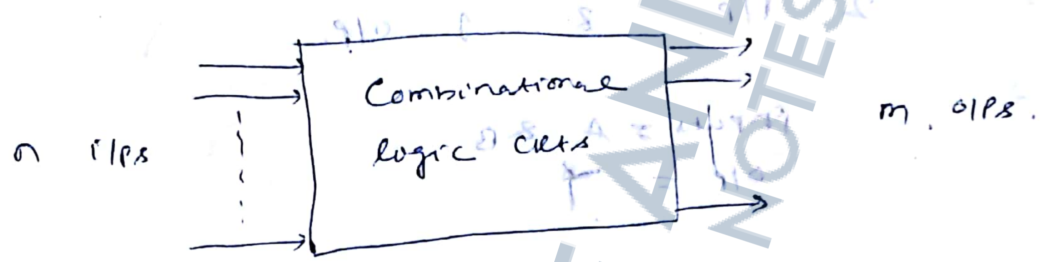
→ Sequential ckts employ storage elements in addition to logic gates. Their o/ps are a function of previous i/ps. As a consequence, the o/ps of a sequential ckts depend not only on present values of i/ps, but also past i/ps and the ckt. behavior must be specified by a time sequence of i/ps and internal states.

→ Comparison of Combinational & sequential logic ckts.

Combinational Logic ckt	Sequential Logic ckt
1) Its o/p depend on the present values of the i/p only.	1) Its o/p not only depend on the present values of i/ps, but also depends on past i/ps.
2) It does not have memory element.	2) It has memory element.
3) It does not have feedback path from o/p to i/p	3) It has feedback path from o/p to i/p.
4) It does not have a clock signal.	4) It has a clock signal
5) A combinational ckt consists of logic gates.	5) A sequential logic gate consists of combinational logic ckts and memory elements.
6) Its action does not depend on clock transition.	6) Its action depend on clock transition.
7) Its ckt is simpler than the ~~Combinational~~ Sequential logic ckts.	7) Its ckt is more complex than that of combinational logic ckts.
8) Ex → Adder, Subtractor, MUX, DeMux, Decoder, Encoder etc.	8) Ex → Flip-Flops, Counters, Shift-registers.

Combinational Logic Ckt :-

A Combinational Ckt. Consists of i/p variables, logic gates and o/p variables. The logic gates accept signal from the i/ps and generate signal to the outputs.



The n i/p binary variables come from an external source. The m o/p variables go to external destination.

Design procedures -

1) From the specification of the ckt, determine the required number of i/ps and o/ps.

2) Assign i/p & o/p variables.

3) Derive truth table that defines the required relationship between i/ps and o/ps.

4) Obtain the simplified expression for o/p as function of i/p variables.

5) Draw the logic diagram & verify the correctness of the design.

Ex :- 1) Design a 2 i/p digital ckt whose o/p is 1 if all i/ps are same. The o/p is 0 otherwise. Design the digital ckts.

(a) Using Basic gates b) Using NAND gates.

Ans :-

Step 1 :- 2 i/p & 1 o/p.

Step 2 :-

Inputs = A & B
o/p = Y

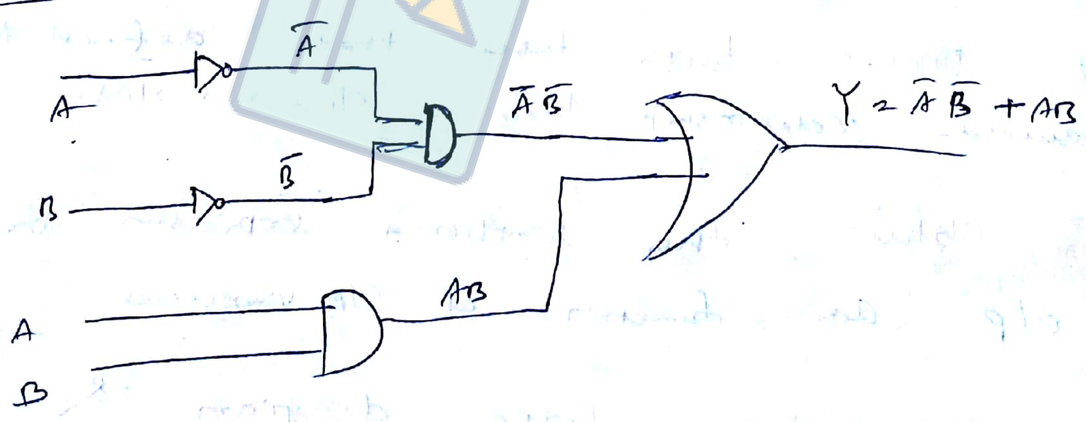
Step-3

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

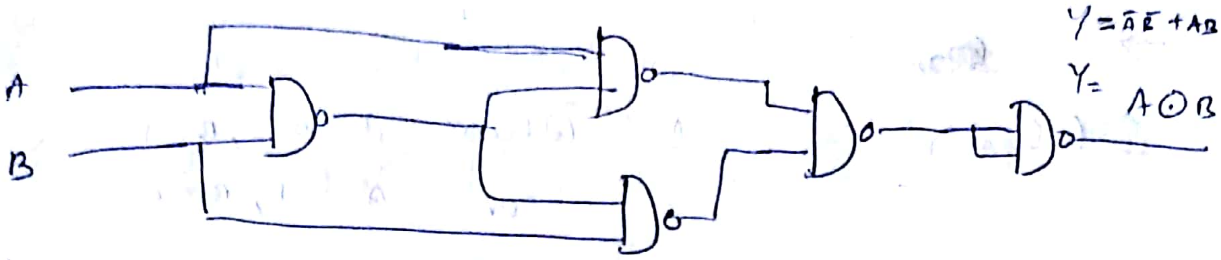
Step-4

$$Y = \bar{A}\bar{B} + AB \quad (\text{X-NOR})$$

Step-5 :-



(a) using Basic gates.



Basic Adders:-

→ The most basic arithmetic operation is addition of 2 binary digits.

0+0 = 0
0+1 = 1
1+0 = 1
1+1 = 10 carry

$$\begin{array}{r} 1 \\ +\,1 \\ \hline 10 \end{array}$$

→ The first three operation produces a sum of one digit and 4th one produces a carry.

→ A combinational ckt. that performs the addition of 2 bit is called half adder.

→ One that performs addition of 3 bit is called full adder.

Half Adder:-

→ Half-adder is a combinational logic ckt, which perform 2 bits addition.

→ The result produces a sum as well as carry.

→ The simplified expression can be directly obtained from the truth table.

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

⟨Truth table for half adder⟩

→ S (Sum) is 1 when $A = 0, B = 1$
or $A = 1, B = 0$

$$\boxed{\therefore S = \bar{A}B + A\bar{B} = A \oplus B}$$

C (Carry) is 1, when $A = 1, B = 1$

$$\boxed{\therefore C = AB}$$

Block Diagram :-

fig (a)

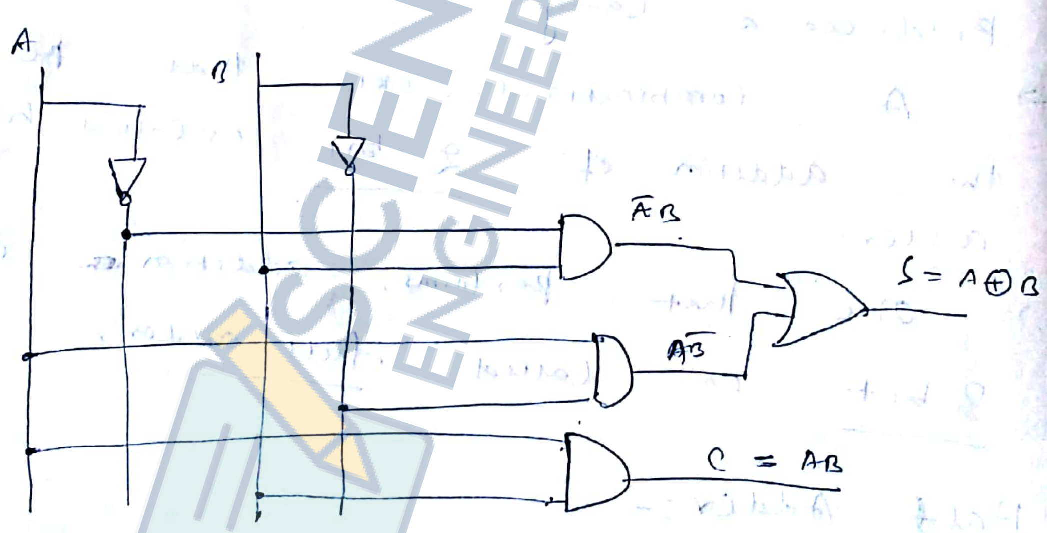
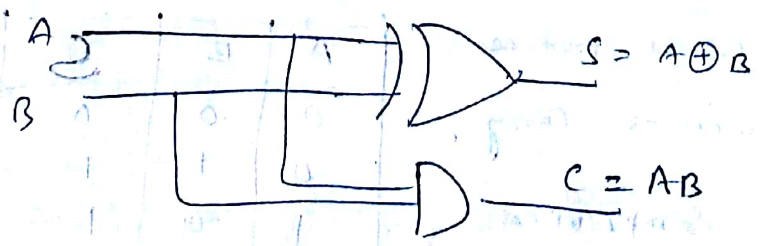


fig (b) Using Basic logic gates.



(c) Using EX-OR and AND gate

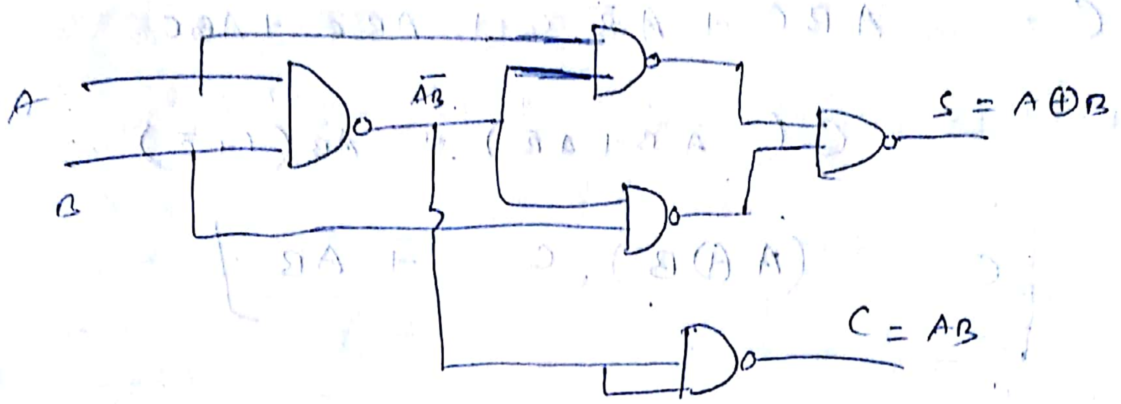


fig (4) :- Using NAND gates only.

Full Adder :- (Start - Remaining part of half adder)

Full adder is combinational logic ckt which perform 3 bit binary addition, which produces a sum as well as carry.

A	B	C	S (sum)	C (carry)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

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

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

$$= \bar{A}Y + A\bar{Y} \qquad \begin{array}{|l} B \oplus C = Y \\ B \odot C = \bar{Y} \end{array}$$

$$= A \oplus Y$$

$$\boxed{S = A \oplus B \oplus C}$$

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

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

$$\boxed{C = (A \oplus B).C + AB}$$

Or

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

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

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

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

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

$$= B(\underbrace{(A+\bar{A})}_{=1}(A+C)) + A\bar{B}C \qquad \because A+BC = (A+B)(A+C)$$

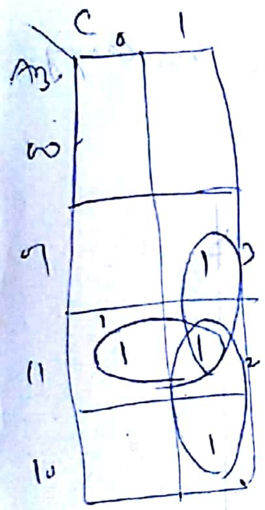
$$= B(A+C) + A\bar{B}C$$

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

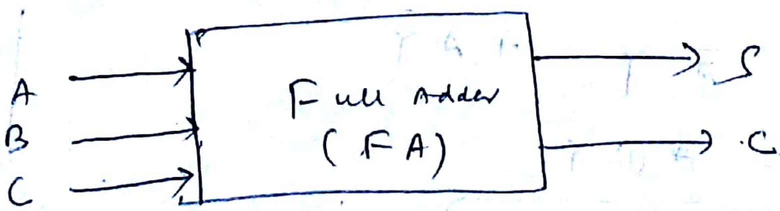
$$= AB + C(B + A\bar{B}) \qquad \because A+BC = (A+B)(A+C)$$

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

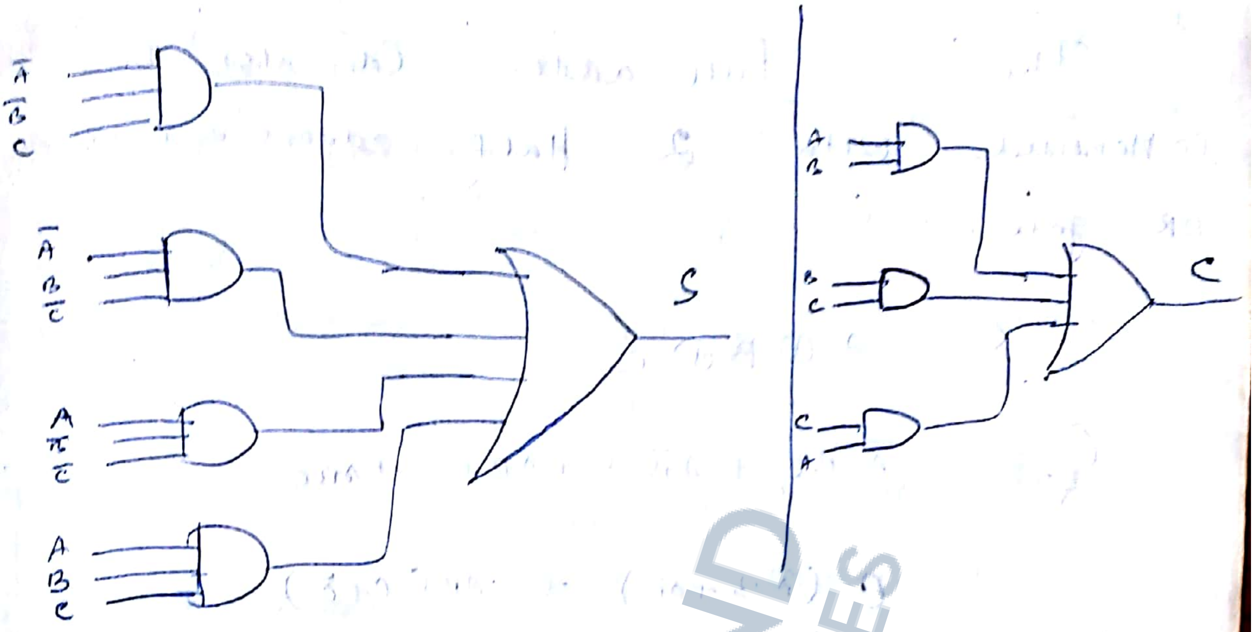
$$= AB + C(A+B)$$



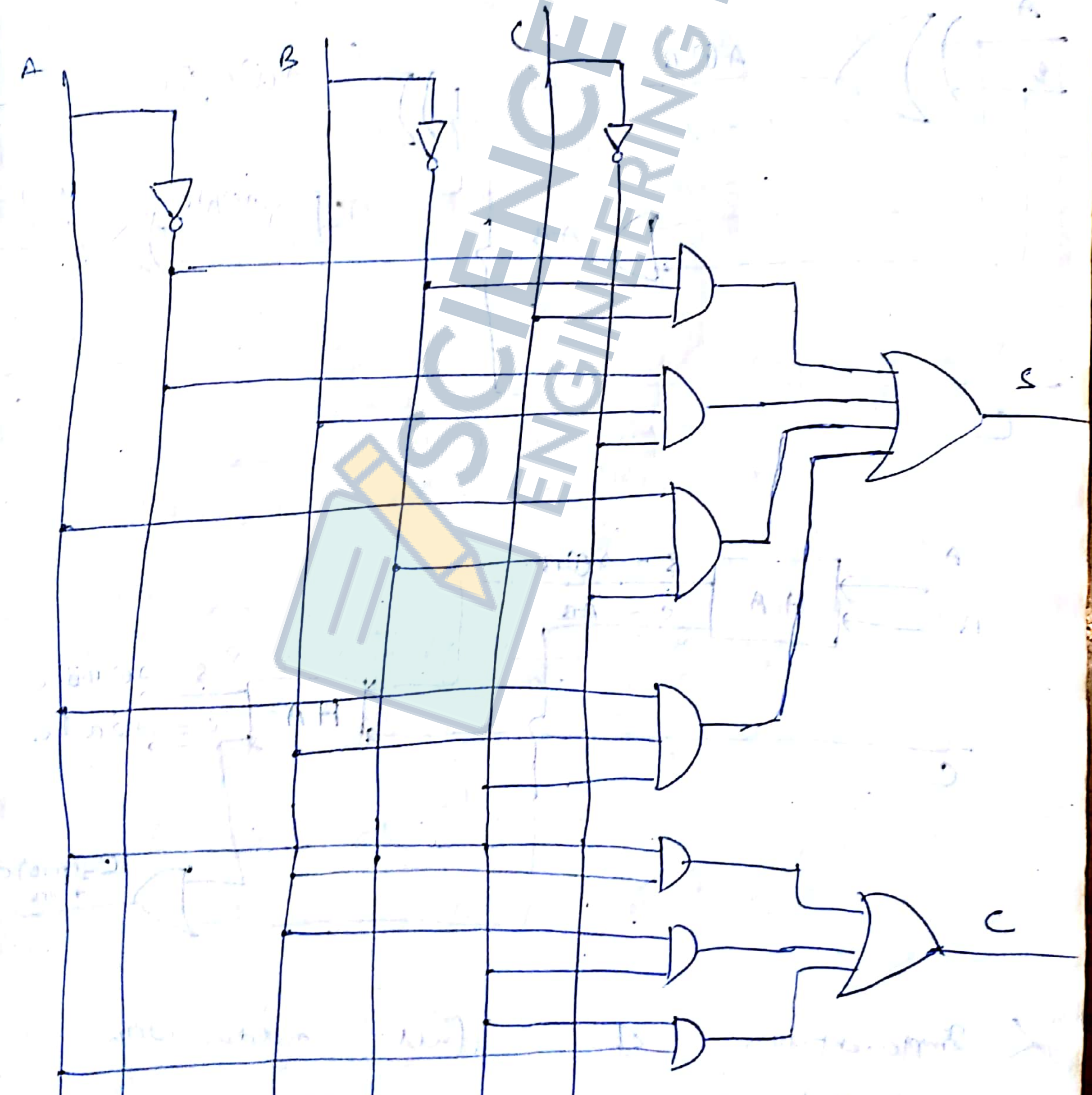
$$\boxed{C = AB + BC + AC}$$



(a) Block diagram.



(b) Implementation of full adder in sum of product (SOP)



Full Adder
(C) Using Basic logic gates.

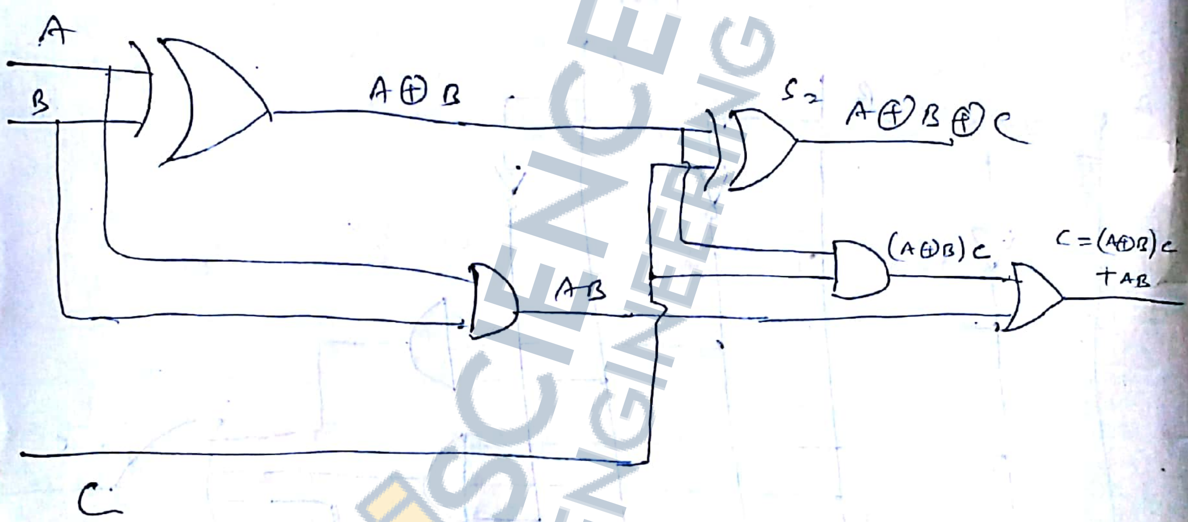
The full adder can also be implemented with 2 Half adders and one OR gate.

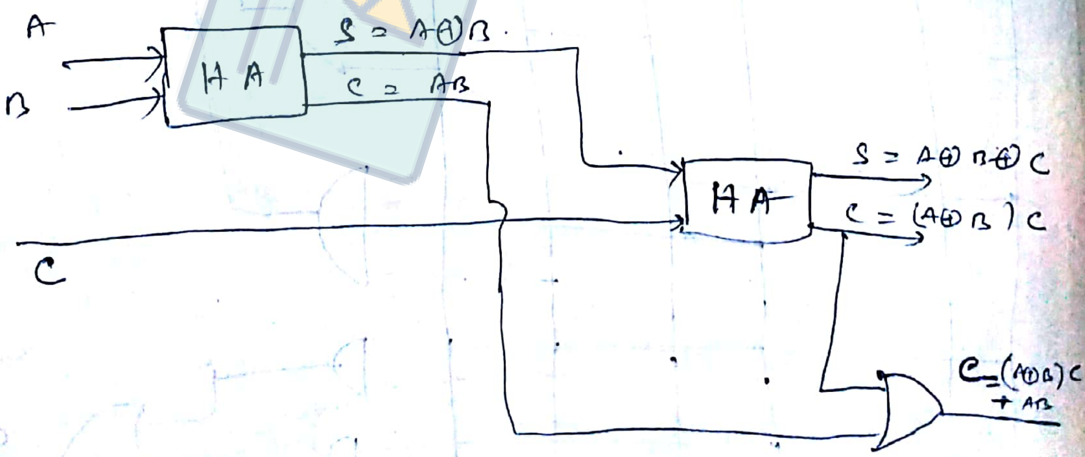
$$S = A \oplus B \oplus C$$

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

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

$$= C(A \oplus B) + AB$$





(Implementation of full adder with 2 Half adder an OR gate)

Basic Subtractor :-

→ The simple subtraction consists of 4 possible elementary operations:-

$0-0=0$, $0-1=1$, Borrow 1.

$1-0=1$, $1-1=0$

→ The combinational ckt that performs the ~~a difference of one digit and~~ the subtraction of 2 bit is called half subtractor.

→ one that performs the subtraction of 3 bits is called full subtractor.

→ 2 half subtractor can be used to complement a full subtractor.

Half Subtractor :-

· Half subtractor is a combinational ckt, which performs 2 bits subtraction, produces a difference as well as borrow.

Truth Table:-

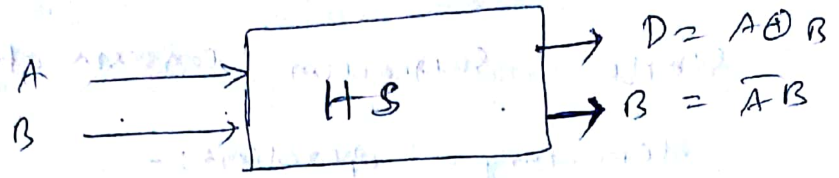
Difference Borrow

A	B	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

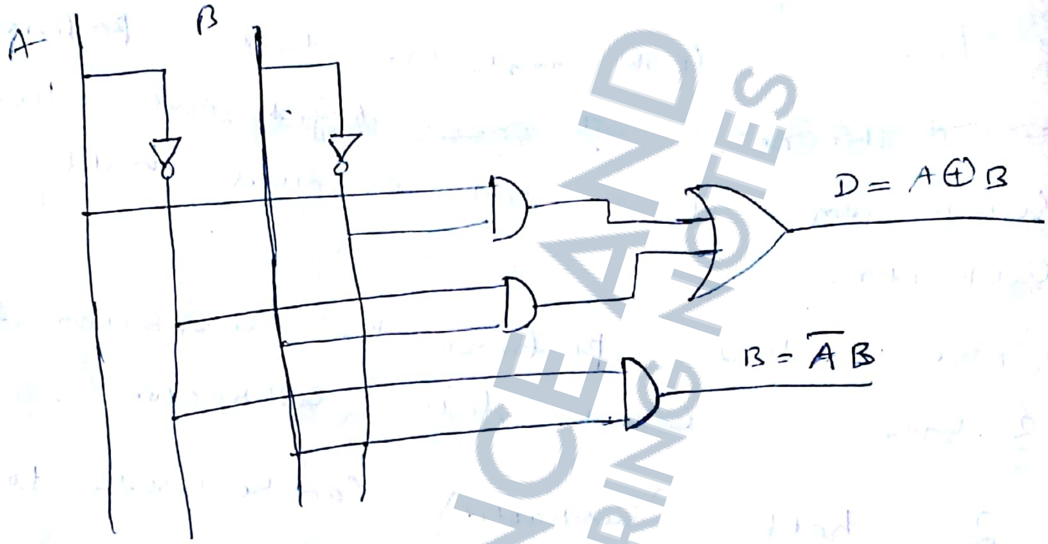
$$\begin{array}{r} 100 \\ -011 \\ \hline \end{array}$$

$\therefore$ $D = \bar{A}B + A\bar{B} = A \oplus B$

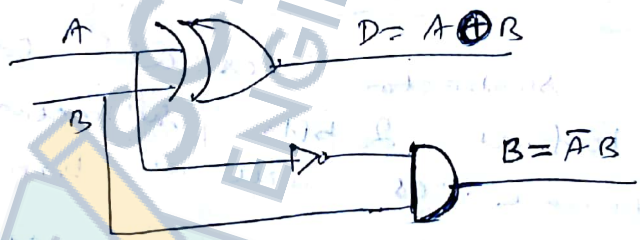
$B = \bar{A}B$
(Borrow)

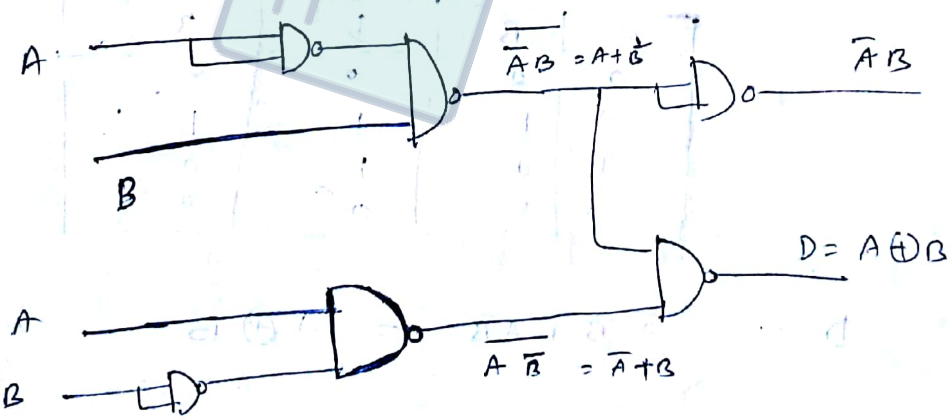


(a) Block diagram.



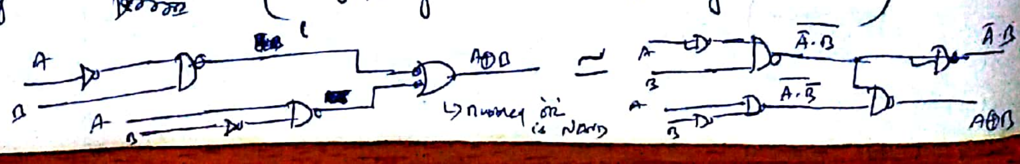
(b) using basic logic gates





(using NAND gate only)

HINTS: - using Bubbled OR



Proof:- $D = \overline{(A+\bar{B}) \cdot (\bar{A}+B)}$

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

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

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

$= A \oplus B$ (Proved)

Full Subtractor :-

Full subtractor is a combinational logic ckt which perform 3 bit binary subtraction, which produces a difference as well as a borrow.

Truth Table:-

A	B	C	D (Difference)	B (Borrow)
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

From the truth table :-

$D = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

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

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

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

$= A \oplus Y = A \oplus B \oplus C$

$B \oplus C = Y$

$B \odot C = \bar{Y}$

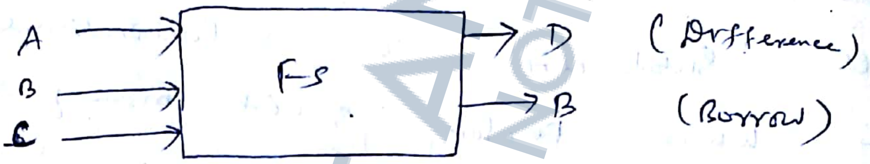
0

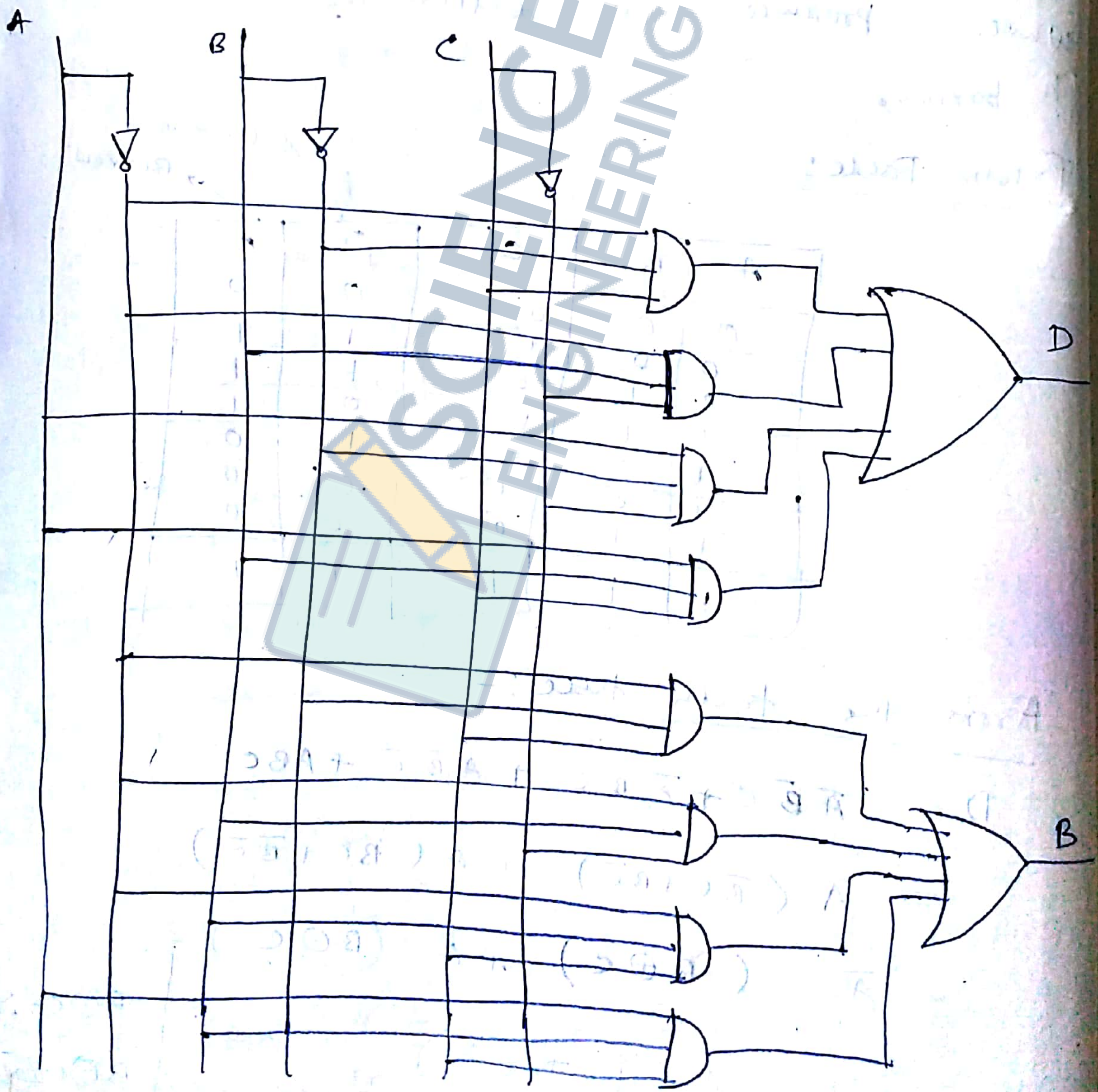
B (Borrow) $= \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + ABC$

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

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

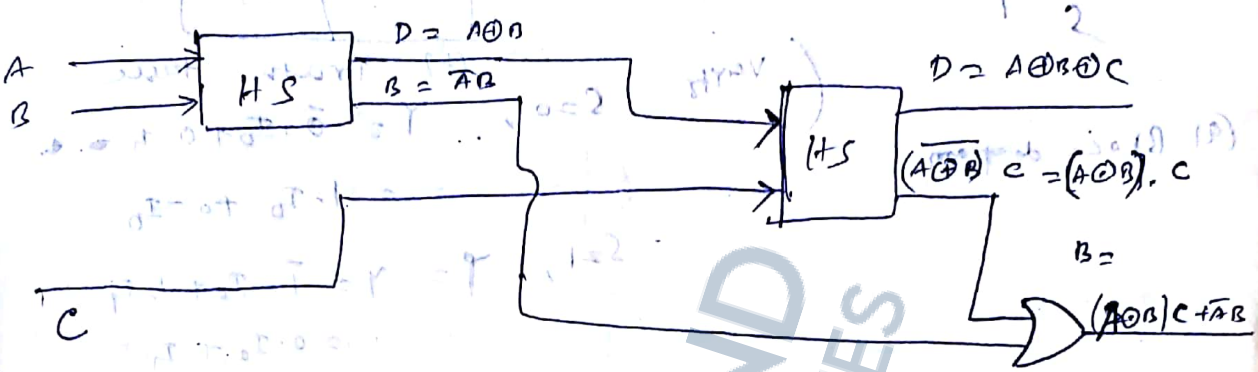
The Block diagram for a full subtractor is shown





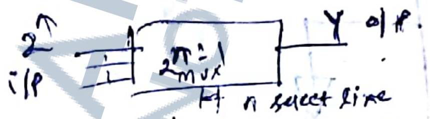
(Full Subtractor using Basic logic gates)

Full Substractor can also be realized with 2 half substractor and one OR gate.



2nd

Multiplexers :-



→ Multiplexer (MUX) is a Combinational logic ckt. that receives information from many i/ps and directs this information to the o/p. → (Ex:- Enable TV a Remote)

→ The i/p information that is selected is Controlled by a selector lines.

→ A MUX with 2^n i/ps requires n selector lines.

→ Mux is also sometimes known as many to one or it is known as data selector.

2 : 1 MUX

→ It has 2 i/ps and one o/p, so no. of selector line is one

→ Inputs are I_0 & I_1, o/p 'y' & selector line 's'.

①

$Y = \bar{S} I_0 + S I_1$

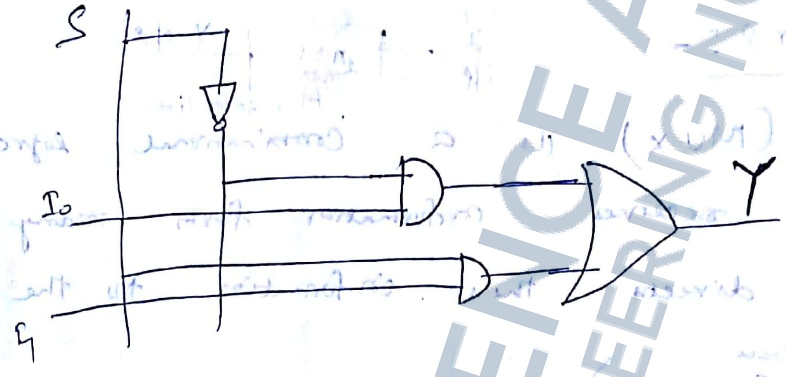
(a) Block diagram

S	Y
0	I_0
1	I_1

(b) Truth table

verify: $S=0$, $Y = \bar{0} \cdot I_0 + 0 \cdot I_1$
$= 1 \cdot I_0 + 0 = I_0$

$S=1$, $Y = \bar{1} \cdot I_0 + 1 \cdot I_1$
$= 0 \cdot I_0 + I_1$
$= I_1$



(c) Logic ckt.

4 : 1 MUX

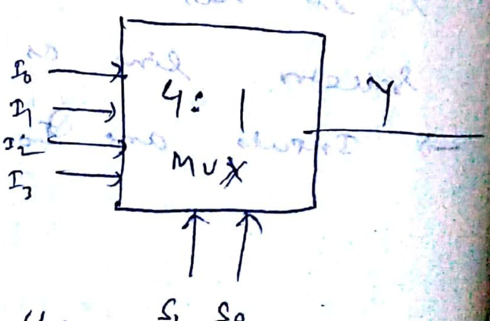
→ It has 4 i/ps and one o/p. So number of selector line is 2. ($\because 4 = 2^2$)

→ Inputs are I_0, I_1, I_2, I_3 and selector lines S_1, S_0.

→

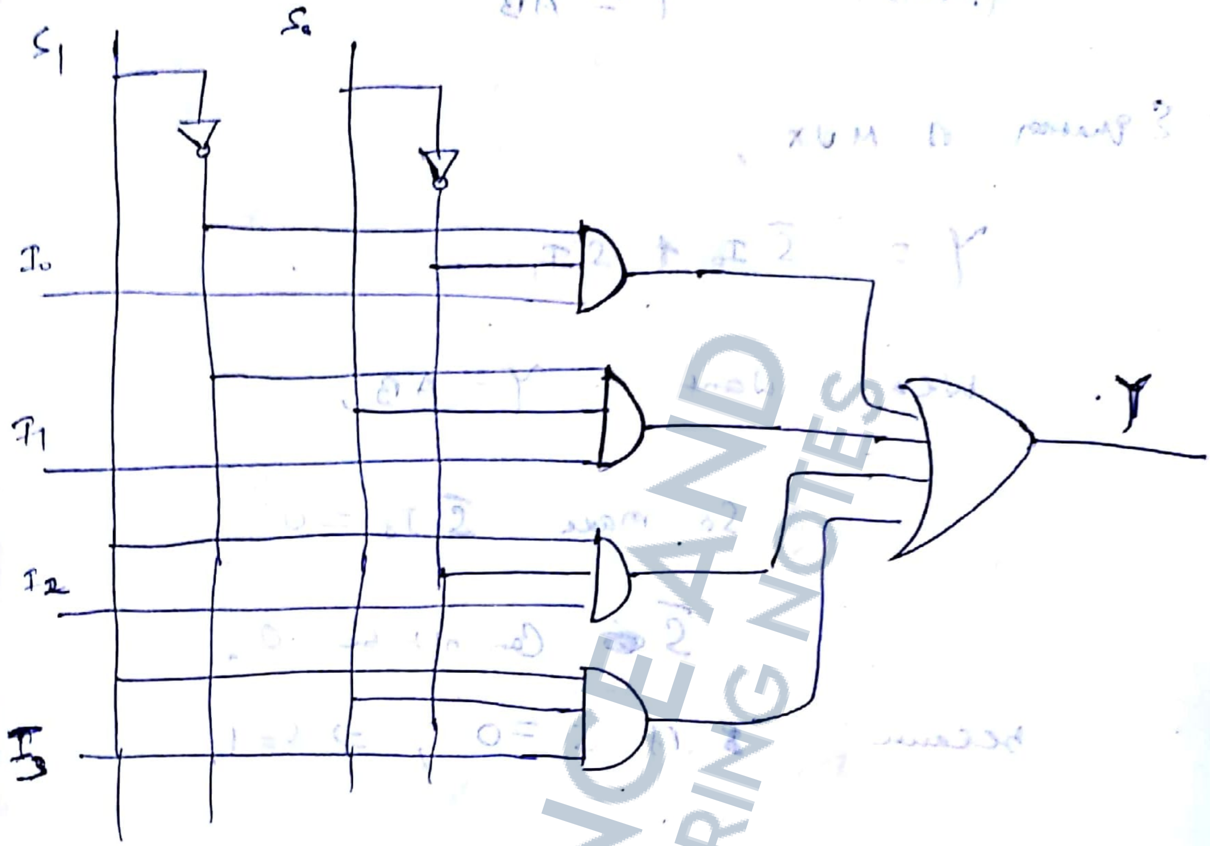
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

(a) Truth table



(b) Block diagram

$$Y = \bar{S}_1 \bar{S}_0 I_0 + \bar{S}_1 S_0 I_1 + S_1 \bar{S}_0 I_2 + S_1 S_0 I_3$$



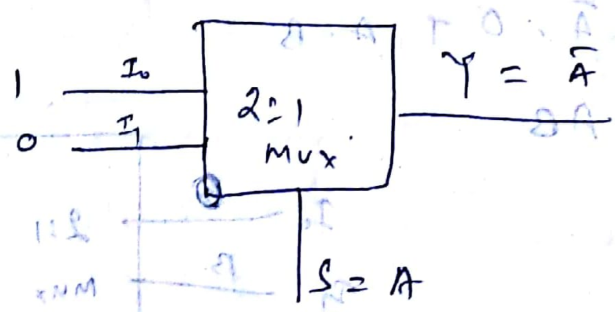
Ex-1 :- Implement Inverter $Y = \bar{A}$ using MUX

Ans :- $Y = \bar{S} I_0 + S I_1$

If $S = A$, $I_1 = 0$, $I_0 = 1$

$$Y = \bar{A} \cdot 1 + A \cdot 0$$

$$Y = \bar{A}$$



Ex-2

Implement 2 i/p AND gate using MUX

Ans :

(AND) $Y = AB$

Equation of MUX,

$$Y = \bar{S} I_0 + S I_1$$

We want $Y = AB$,

So make $\bar{S} I_0 = 0$

$\bar{S}$ Can n't be 0.

because if $\bar{S} = 0$, $\Rightarrow S = 1$

$$S I_1 = 1 . I_1 = I_1$$

But we want $S I_1 = AB$.

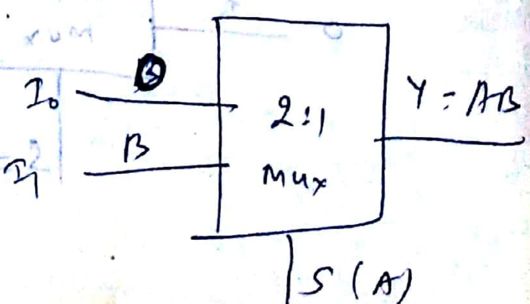
$\therefore$ since $\bar{S} \neq 0$, $I_0 = 0$ (because we want $\bar{S} I_0 = 0$)

Lets take
$S = A$
$I_1 = B$
$\bar{S} = \bar{A}$
$I_0 = 0$

$$Y = \bar{S} I_0 + S I_1$$

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

$$Y = AB$$



or

$S = B$

$I_1 = A$

$\bar{S} = \bar{B}$

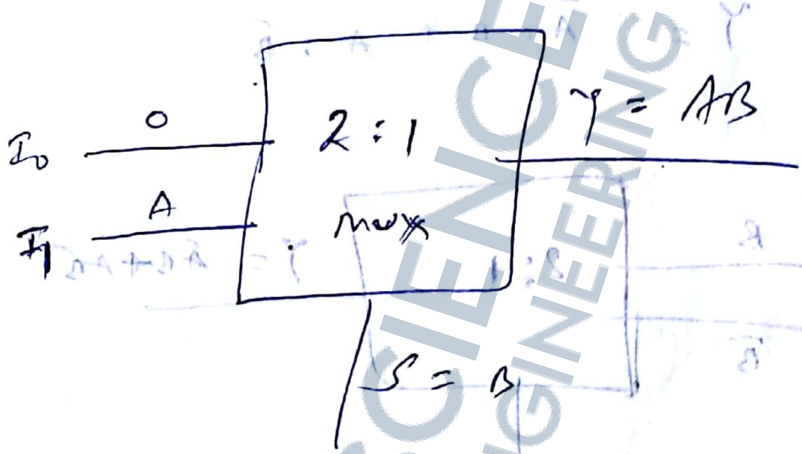
$I_0 = 0$

$$Y = \bar{S} I_0 + S I_1$$

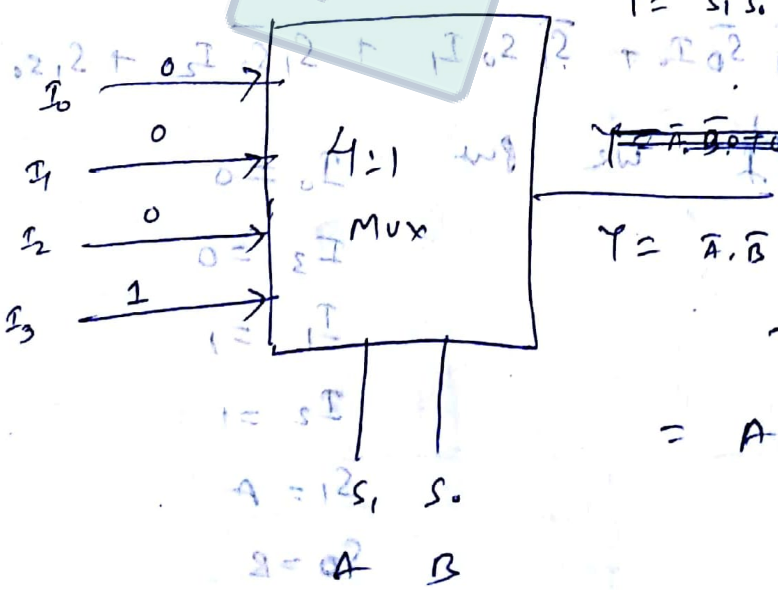
$$= \bar{B} \times 0 + B \cdot A$$

$$= BA$$

$$= AB$$



Using 4:1 mux



$$Y = \bar{S_1}\bar{S_0} I_0 + \bar{S_1} S_0 I_1 + S_1 \bar{S_0} I_2 + S_1 S_0 I_3$$

$$Y = \bar{A}.\bar{B}.0 + \bar{A}.B.0 + A.\bar{B}.0 + AB.1$$

$$= AB \text{ (AND gate)}$$

Ex-3 ⓪ 95

Implement 2 i/p X-OR gate using MUX.

Ans :

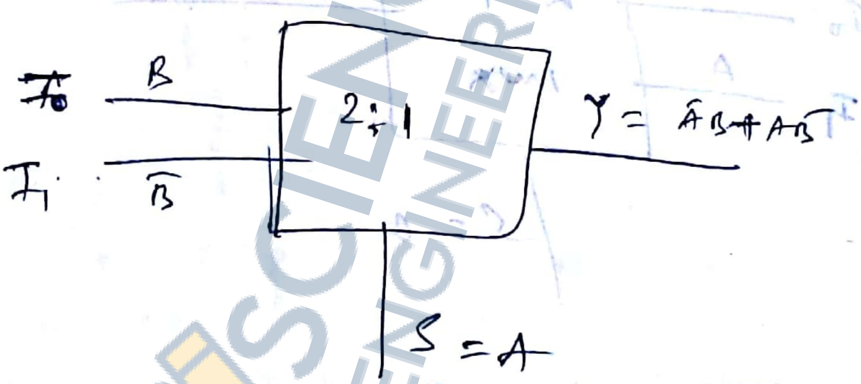
$$Y = \bar{A}B + A\bar{B}$$

But the eqn

$$Y = \bar{S} I_0 + S I_1$$

Let $S = A$, $I_0 = B$, $I_1 = \bar{B}$

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



Using 4:1 MUX

$$Y = \bar{S_1}\bar{S_0} I_0 + \bar{S_1} S_0 I_1 + S_1 \bar{S_0} I_2 + S_1 S_0 I_3$$

If we put $I_0 = 0$

$I_3 = 0$

$I_1 = 1$

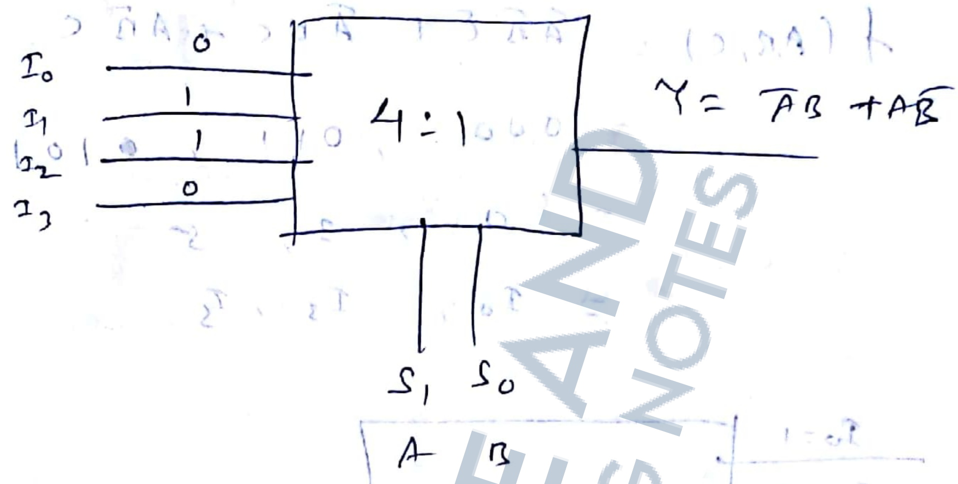
$I_2 = 1$

$S_1 = A$

$S_0 = B$

$$Y = 0 + \bar{A}B \cdot 1 + A\bar{B} \cdot 1 + 0$$

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



<u>XNOR</u> : Similar $Y = \bar{A}\bar{B} + AB$

$$Y = \bar{S_1}\bar{S_0} I_0 + \bar{S_1} S_0 I_1 + S_1 \bar{S_0} I_2 + S_1 S_0 I_3$$

If we put, $I_1 = 0$

$I_2 = 0$

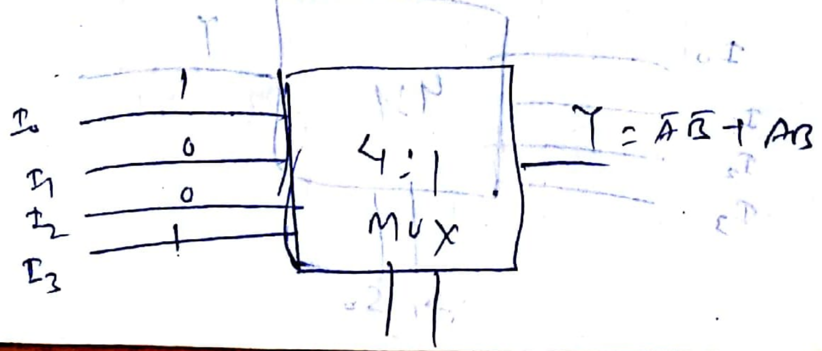
$I_0 = 1$

$I_3 = 1$

$S_1 = A$

$S_0 = B$

$$Y = \bar{A}\bar{B} \cdot 1 + 0 + 0 + AB \cdot 1$$



Ex :- 4 Implement

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

using 8:1 MUX

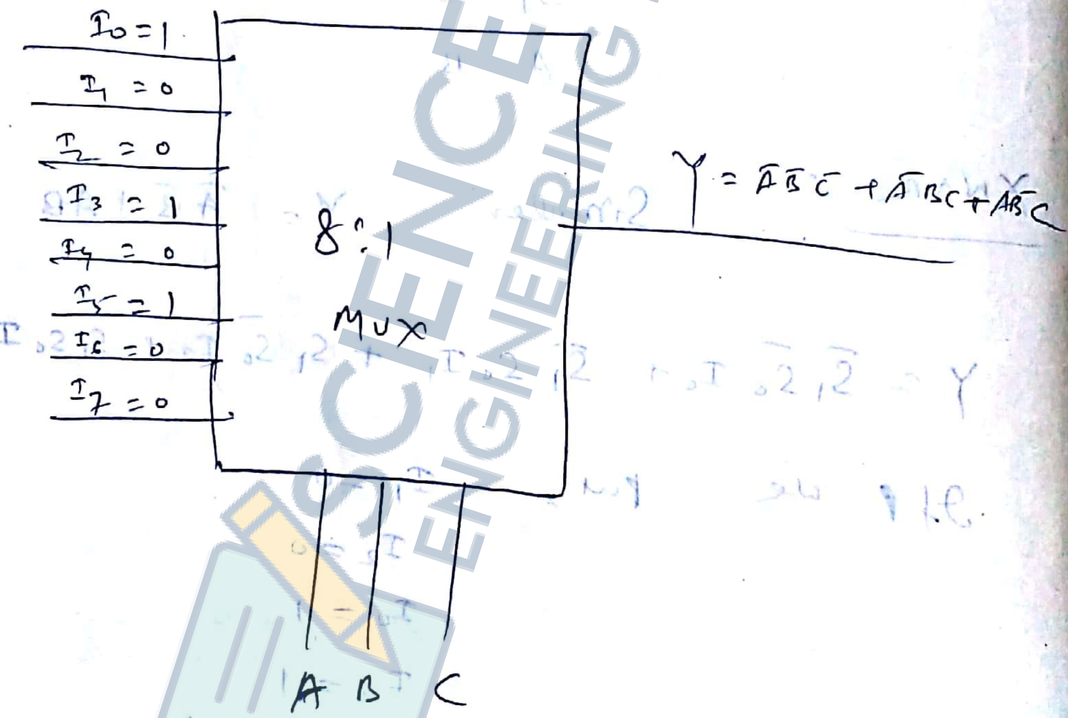
Ans :

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

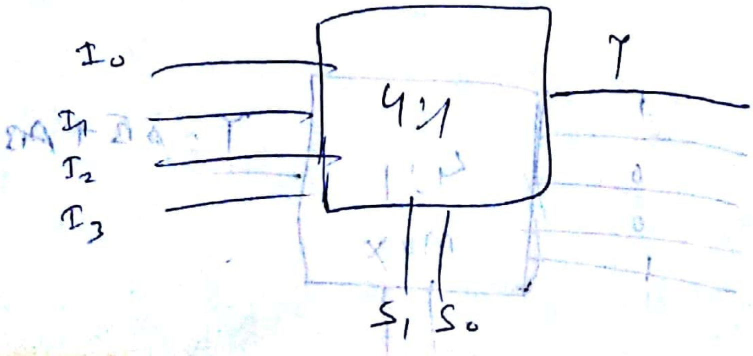
$$= 000, 011, 101$$

$$= 0, 3, 5$$

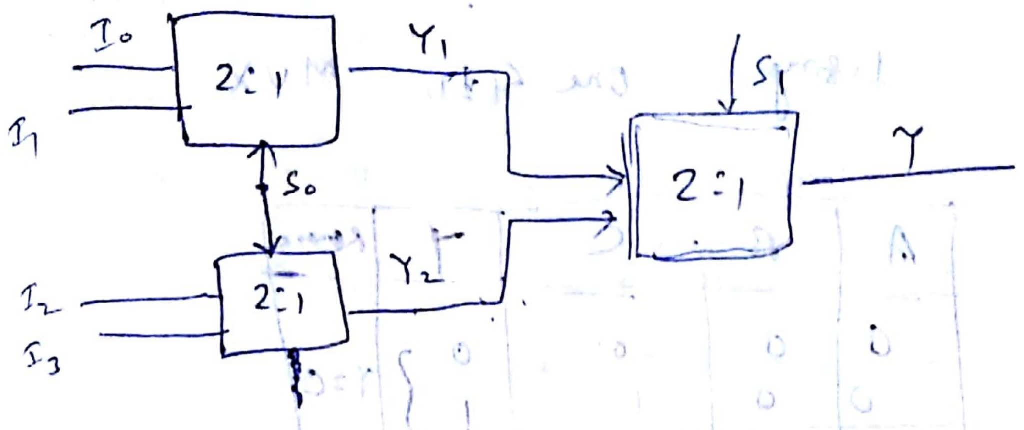
$$= I_0, I_3, I_5$$



Ex :- 5 Realize 4:1 MUX using 2:1 MUX



S_1	S_0	
0	0	Y_1
0	1	Y_1
1	0	Y_2
1	1	Y_2



< we have 4 i/p pins & 2 select pin. Use three 2:1 MUX to construct the functionality of 4:1 MUX >

S_1	S_0		Y
0	0	—	I_0
0	1	—	I_1
1	0	—	I_2
1	1	—	I_3

<u>HW)</u> 1) 8:1 mux using 4:1 mux
2) 8:1 " " 2:1 mux

BIUT 2019
Ex-5

Implement the following boolean function with a 4:1 MUX.

$$f(x, y, z) = \Sigma(1, 2, 6, 7)$$

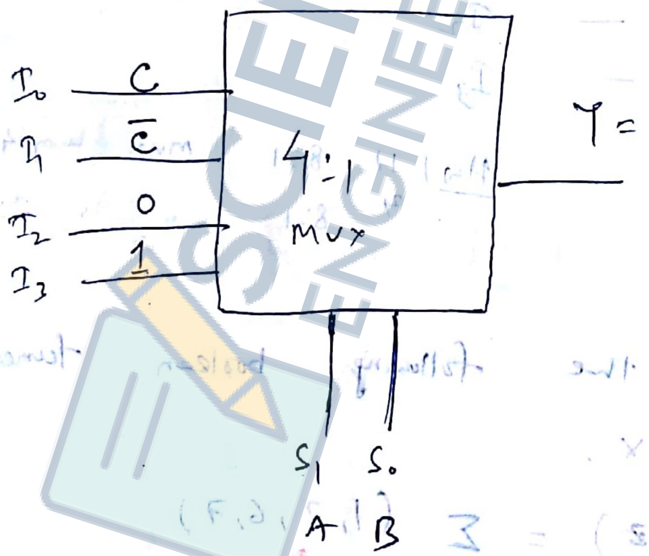
Ans: The truth table for the boolean function is given below

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

①

We have to implement the truth table using one 4:1 MUX :

A	B	C	Y	Remark
0	0	0	0	Y = C
0	0	1	1	
0	1	0	1	$Y = \bar{C}$
0	1	1	0	
1	0	0	0	Y = 0
1	0	1	0	
1	1	0	1	Y = 1
1	1	1	1	



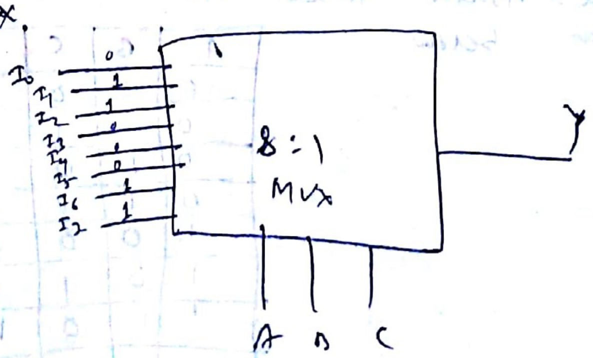
Using 8:1 mux

O/P is 1,

At 1, 2, 6, 7

i.e.

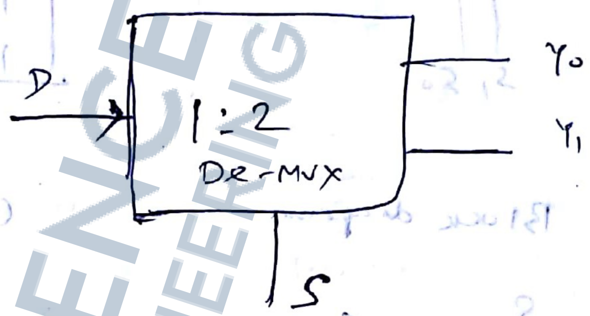
I_1, I_2, I_6, I_7



Demultiplexer

Demultiplexer (DE-MUX) is a combinational logic ckt that performs the reverse operation of MUX. It has one i/p & many o/ps. The i/p information is transmitted to a particular o/p line depending on the data available at the selector line.

1:2 DE-MUX



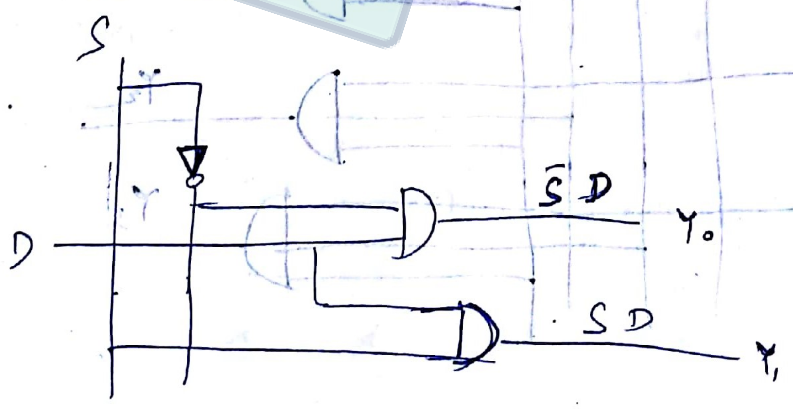
(a) Block diagram

S	Y_0	Y_1
0	D	0
1	0	D

$Y_0 = \bar{S} D$

$Y_1 = S D$

(b) Truth table



(c) Logic ckt.

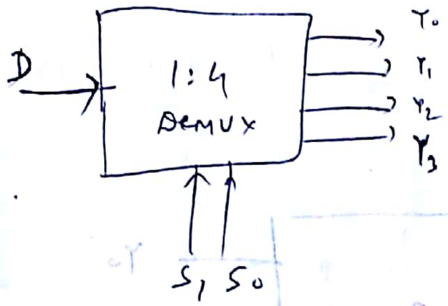
1 : 4 DE-MUX

Here 1 i/p & 4 o/p, so it has 2 Nos. selector line.

Input → D

o/p → Y_0, Y_1, Y_2, Y_3

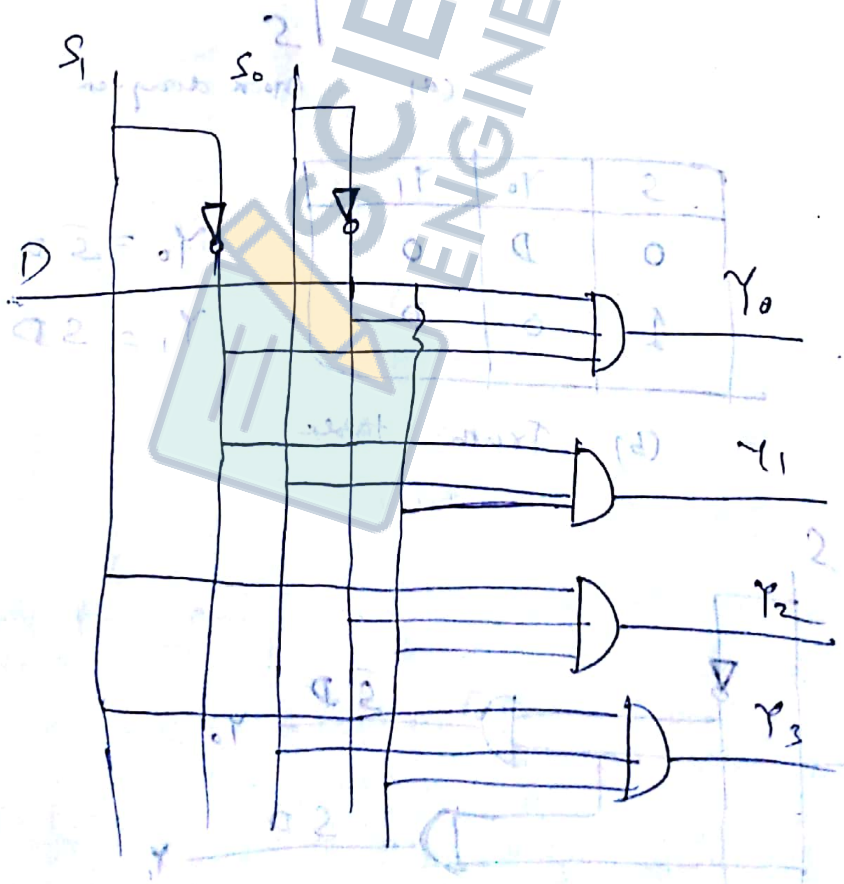
Selector line - S_1 & S_0



(a) Block diagram

S_1	S_0	Y_0	Y_1	Y_2	Y_3
0	0	D	0	0	0
0	1	0	D	0	0
1	0	0	0	D	0
1	1	0	0	0	D

(b) Truth table



(c) Logic ckt.

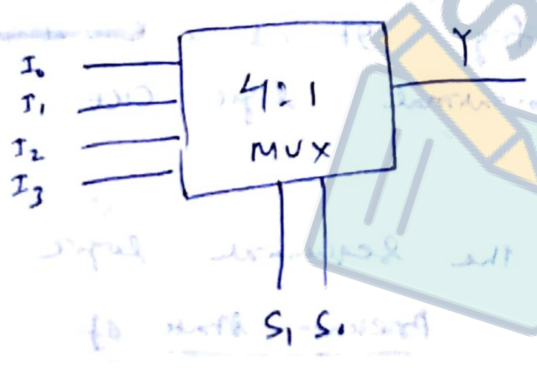
$$Y_0 = \bar{S_1}\,\bar{S_0}\,D$$

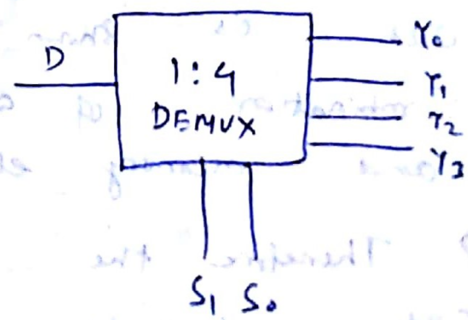
$$Y_1 = \bar{S_1}\,S_0\,D$$

$$Y_2 = S_1\,\bar{S_0}\,D$$

$$Y_3 = S_1\,S_0\,D$$

Multiplexer	Demultiplexer
1) It is a data selector.	1) It is a data distributor.
2) It has many i/p and single o/p.	2) It has single i/p and many o/p.
3) Parallel to serial converter.	3) Serial to parallel converter.
4) It has 2^n no of i/p, n select line and 1 o/p.	4) It has 1 i/p, n select line and 2^n o/p.
5) Ex:-	5) Ex:-

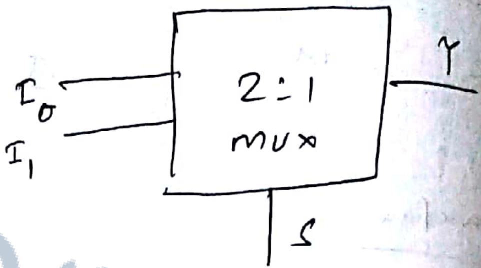




MUX as Universal gate

1) NOT gate using 2:1 MUX

A	Y
0	1
1	0



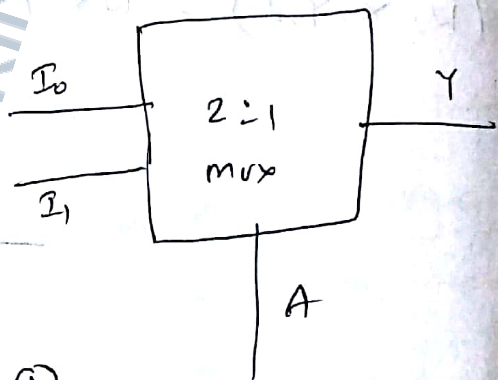
Step 1:-

We have one select line. A select line can take 2 values '0' or '1'. So lets take 'A' as select line because it has value 0 & 1.

Step 2:- When

$A=0, \ Y = I_0$

$A=1, \ Y = I_1$



A	Y
0	I_0
1	I_1

— (1)

Step 3:- Relation of Y with A

$$Y = \bar{A}$$

when $A=0, \ Y=1$

when $A=1, \ Y=0$.

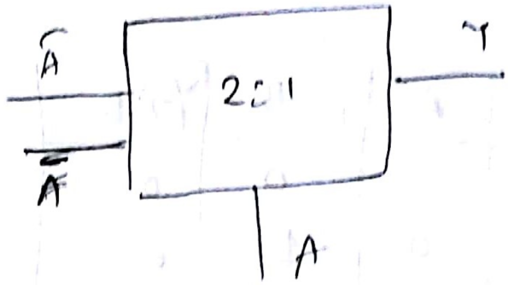
A	Y	Y
0	1	$\bar{A}$
1	0	$\bar{A}$

— (2)

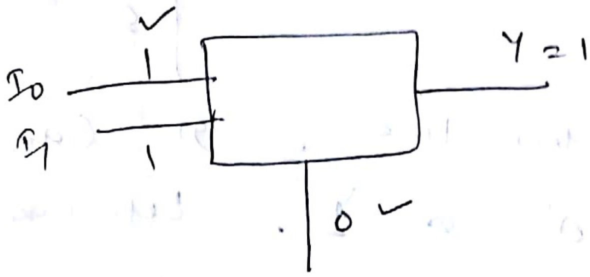
Equating ① & ②

$I_0 = \bar{A}$,

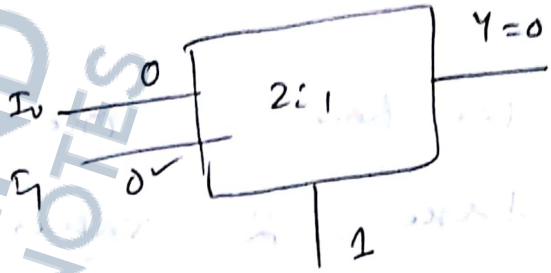
$I_1 = \bar{A}$



ex:- when $A = 0$

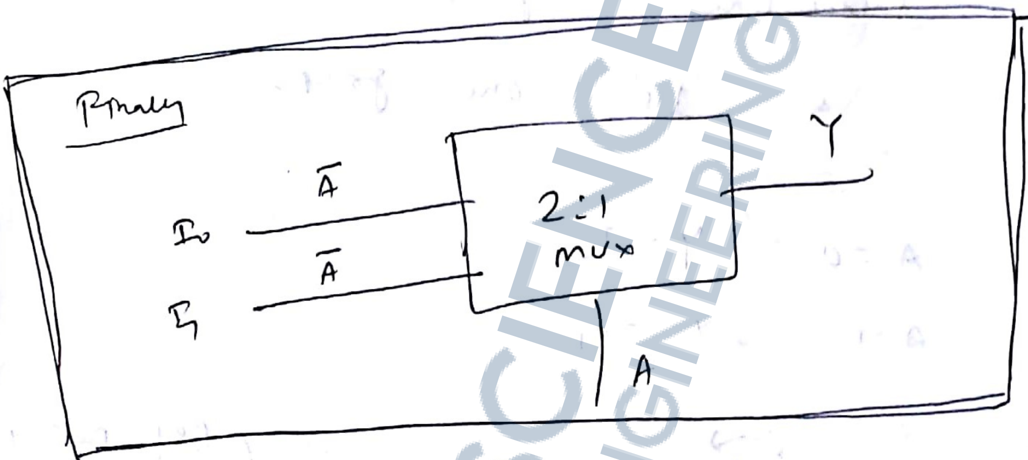


when $A = 1$



→ Answer 1

∴ Finally



~~AND gate~~

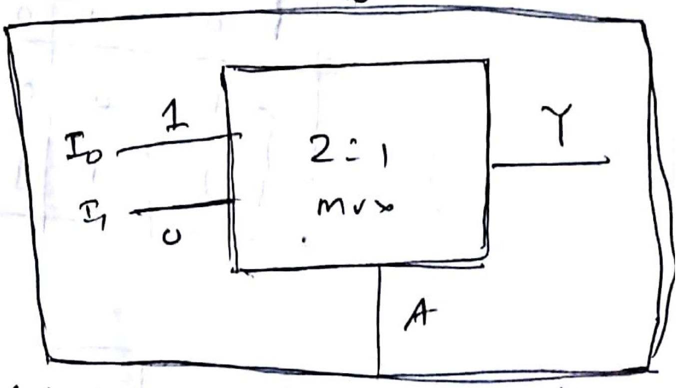
or

Answer 2

Equating ① & ②.

$I_0 = 1$

$I_1 = 0$



Verify

when $A = 0$, I_0 is selected $\Rightarrow Y = 1$.

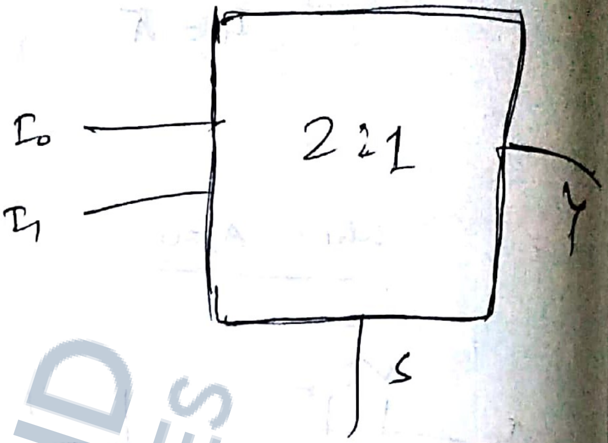
when $A = 1$, I_1 is selected $\Rightarrow Y = 0$

} Act as NOT gate

Verify; $Y = \bar{A} \cdot I_0 + A \cdot I_1 = \bar{A} \cdot 1 + A \cdot 0 = \bar{A}$

2) AND gate

A	B	Y=AB
0	0	0
0	1	0
1	0	0
1	1	1



We have one select line. It can take 2 values '0' or '1'. Lets take 'A' as select line group the '0's as one group & '1's one group.

For $A=0$, $Y=I_0$
$A=1$, $Y=I_1$

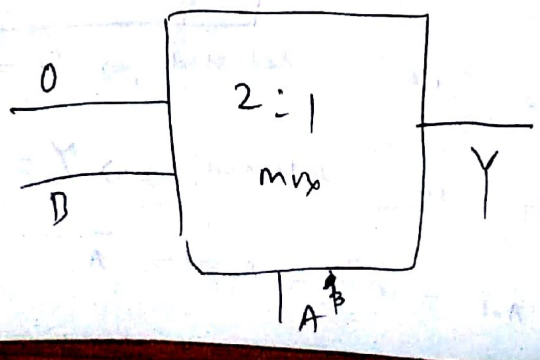
A	B	Y	Y	Remark Y (Rel^n Betn Y and B)
0	0	0	I_0	0
0	1	0		
1	0	0	I_1	B
1	1	1		

→ Y = 0, is always independent of B.

Y = B

$\therefore I_0 = 0$

$\therefore I_1 = B$



Verify

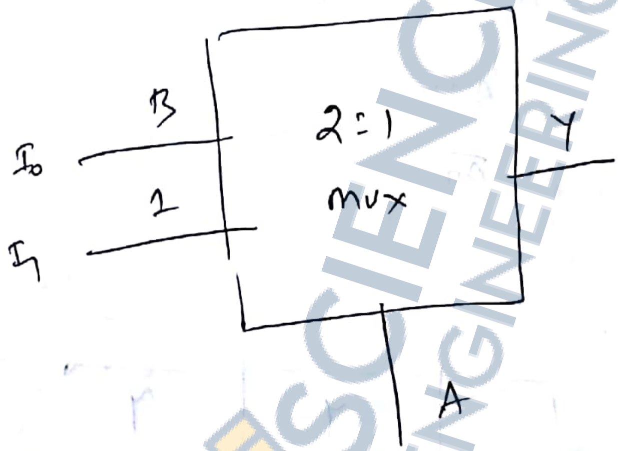
$$Y = \bar{S} I_0 + S I_1$$

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

3) OR gate $= AB$

A	B	Y	Y	Y
0	0	0	I_0	B
0	1	1		
1	0	1	I_1	1
1	1	1		

A = Select line



Verify ($Y = \bar{S} I_0 + S I_1$)

$$Y = \bar{A} \cdot B + A \cdot 1$$

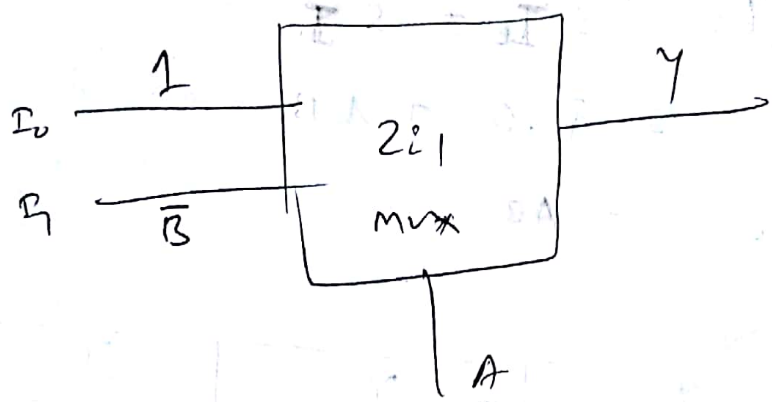
$$= A + \bar{A} B$$

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

$$Y = A + B$$

4) NAND gate

A	B	Y	Y	Y
0	0	1	I_0	1
0	1	1		
1	0	1	I_1	$\bar{B}$
1	1	0		



<u>Verify</u> :

$$Y = \bar{S}I_0 + SI_1$$

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

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

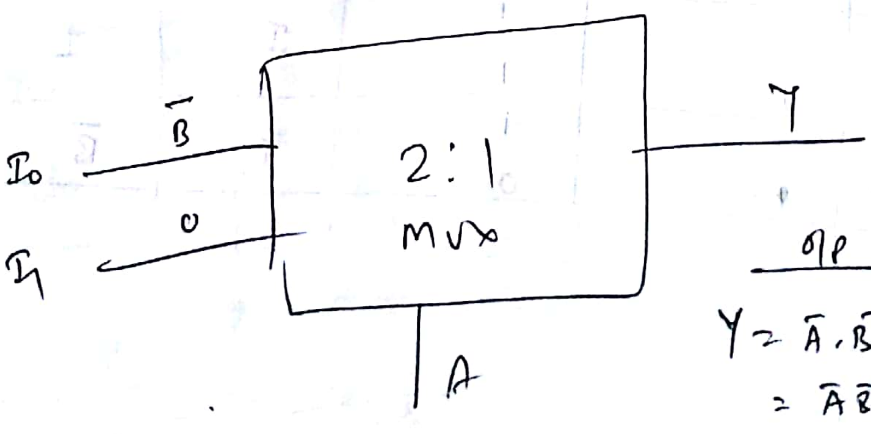
$$= (\bar{A}+A)(\bar{A}+\bar{B})$$

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

$$Y = \overline{AB}$$

5) <u>NOR</u>

A	B	Y	Y	Y
0	0	1	I_0	$\bar{B}$
0	1	0		
1	0	0	I_1	0
1	1	0		



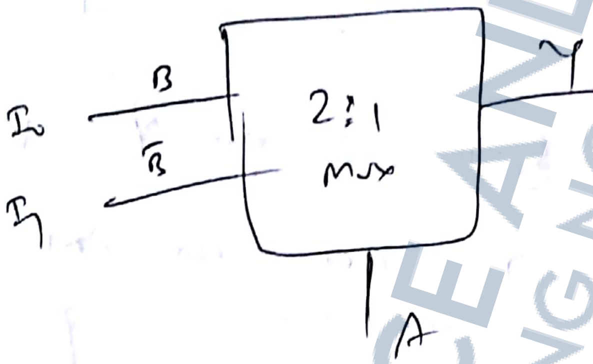
<u>op</u>

$$Y = \bar{A}.\bar{B} + A.0$$

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

6) Ex-OR

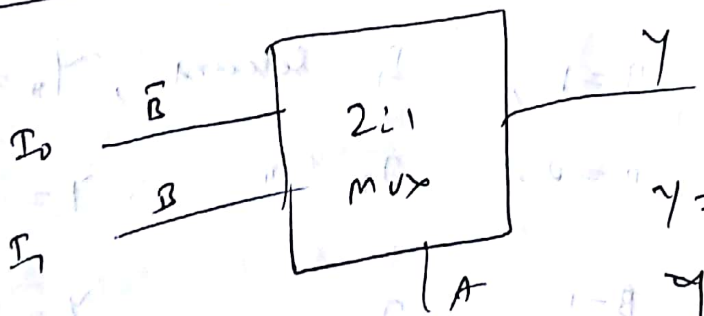
A	B	Y	Y	Y
0	0	0	I_0	B
0	1	1		$\bar{B}$
1	0	1	I_1	
1	1	0		



verify: $Y = \bar{A}.B + A\bar{B} = A \oplus B$.

7) Ex-NOR

A	B	Y	Y	Y
0	0	1	I_0	$\bar{B}$
0	1	0		
1	0	0	I_1	B
1	1	1		

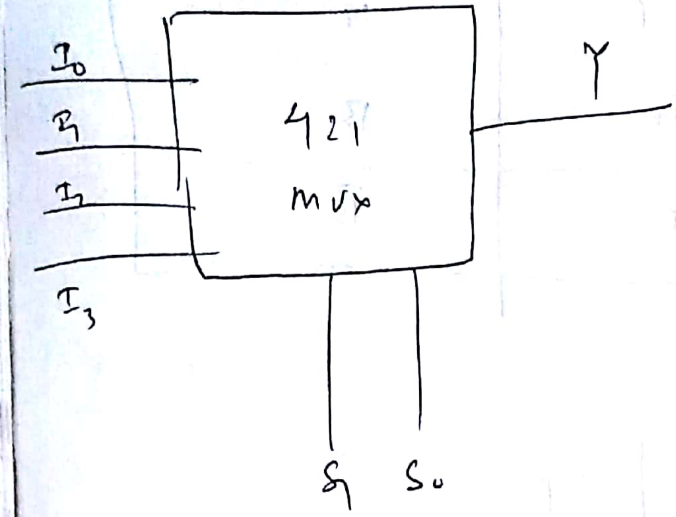


$Y = \bar{A}.\bar{B} + A.B$

$Y = A \odot B$

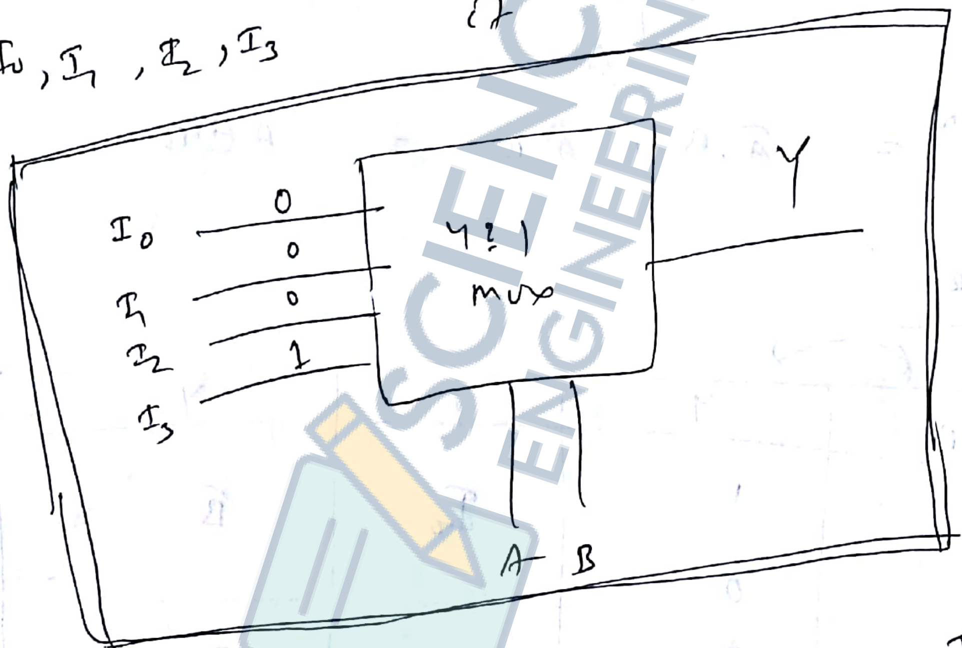
(verified)

87) Implement $Y = AB$ using 4:1 mux



A	B	Y	Y
0	0	0	I_0
0	1	0	I_1
1	0	0	I_2
1	1	1	I_3

If $S_1 = A$ & $S_0 = B$ are Y values to I_0, I_1, I_2, I_3 it act as AND gate



A	B
0	0

Ex. When $A=0, B=0$, I_0 is selecting its value is '0'.

When $A=0$, $B=1$, I_1 selected, $Y = 0$

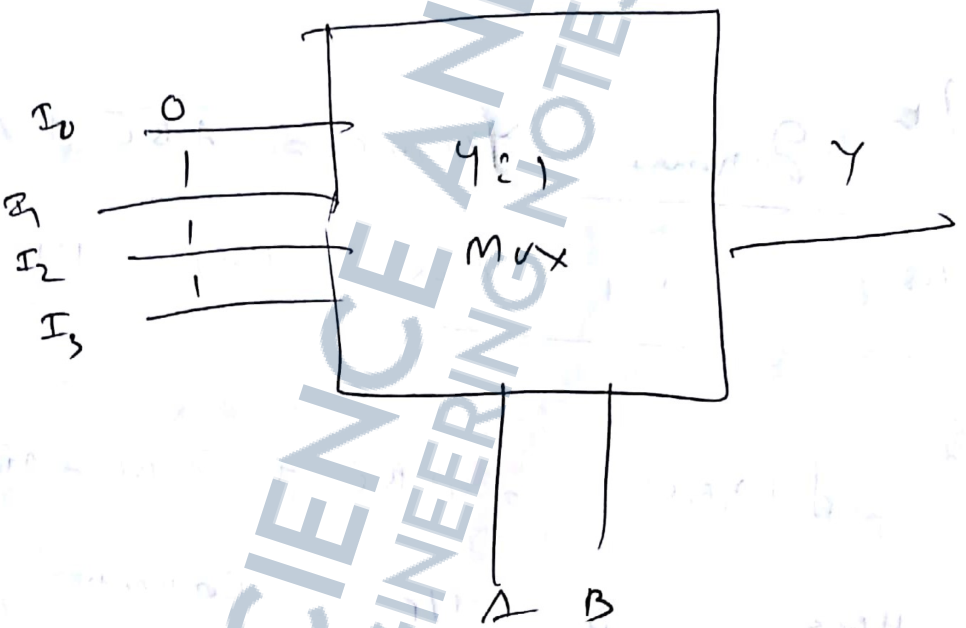
$A=1$, $B=0$, I_2 ", $Y=0$

$A=1$, $B=1$, I_3 ", $Y=1$.

∴ It behaves as AND gate.

7) Similarly other gates can be implemented using 4:1 MUX. With A, B select line & I_0, I_1, I_2, I_3 are truth table values of that gate.

Ex: A+B



Implementation of any function using MUX

→ First check what are the I/Ps ~~required~~ combinations required.

→ ~~If AND~~ (Ex:- AND gate: 4 combinations)

→ Then check what are the i/p line available.

If 2:1 MUX → 2 i/p line → I_0, I_1

If 4:1 MUX → 4 " " → I_0, I_1, I_2, I_3

→ If 2 i/p line then we have to make a group.

$A = \begin{cases} 0 \\ 0 \end{cases} \rightarrow 1$ group $\rightarrow I_0$

$A = \begin{cases} 1 \\ 1 \end{cases} \rightarrow 1$ group $\rightarrow I_1$

If 4 i/p line, for 4 possible i/p combination, no need of grouping just assign the corresponding truth table values to I_0, I_1, I_2, I_3.

Ex) 10) Implement $f(A,B,C) = \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}C$

using 8:1 mux, using 4:1 mux, 2:1 mux

Ans: $f(A,B,C) = \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}C$

Here 8 i/p combination, If we have to use a 8:1 mux, just we have to assign the truth table values.

A	B	C	Y
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	0
1	1	1	0

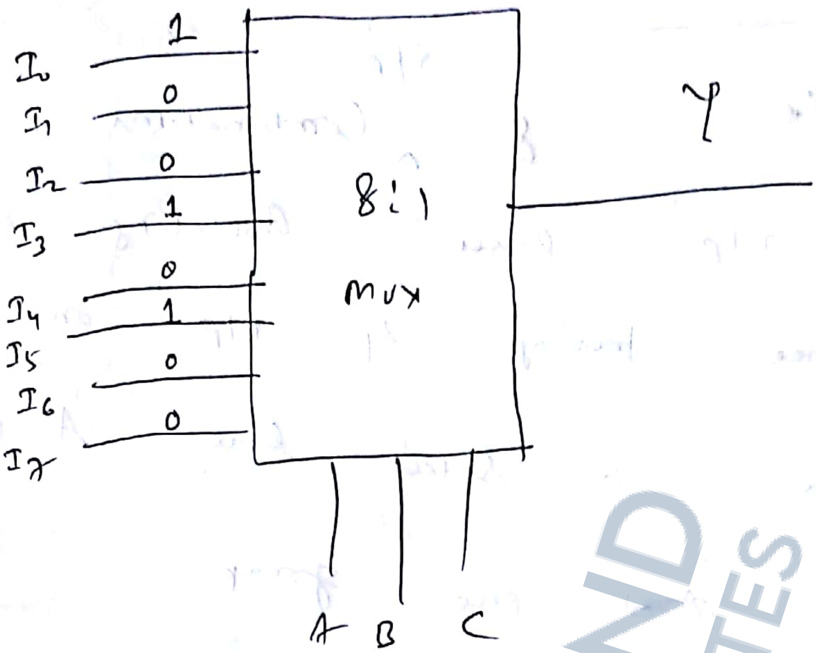
$Y = 000, 011, 101$

$= 0, 3, 5$

$= I_0, I_3, I_5$

$\therefore I_0, I_3, I_5$ will be 1 and rest will be zero

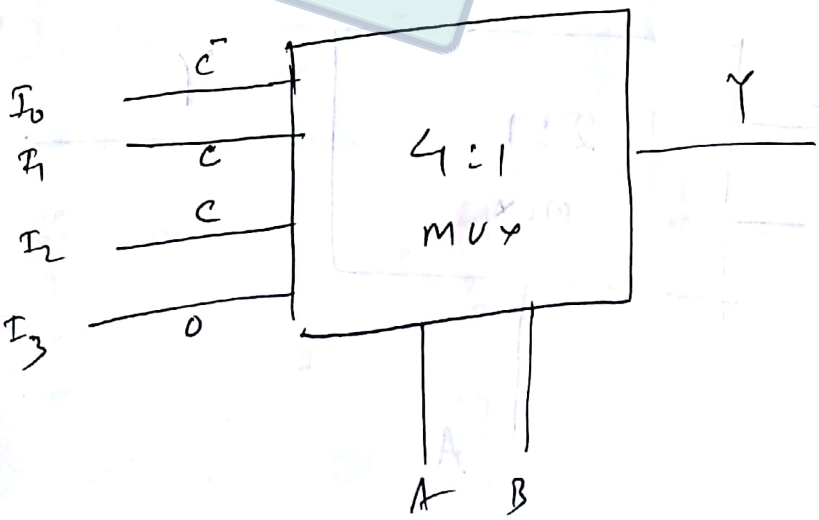
A, B, C are the select lines.



Using 4:1 MUX

A	B	C	Y	Y	Y
0	0	0	1	I_0	$\bar{C}$
0	0	1	0		
0	1	0	0	I_1	C
0	1	1	1		
1	0	0	0	I_2	C
1	0	1	1		
1	1	0	0	I_3	0
1	1	1	0		

8 i/p Combination we have 4 i/p line. So we have to make grouping. Lets say 'A' as the select line. 00, 01, 10, 11 4 groups.



Using 2:1 mux

We have 8 i/p combinations, but we have 2 i/p lines. Grouping is made ~~mani~~ taking 4 i/p one group.

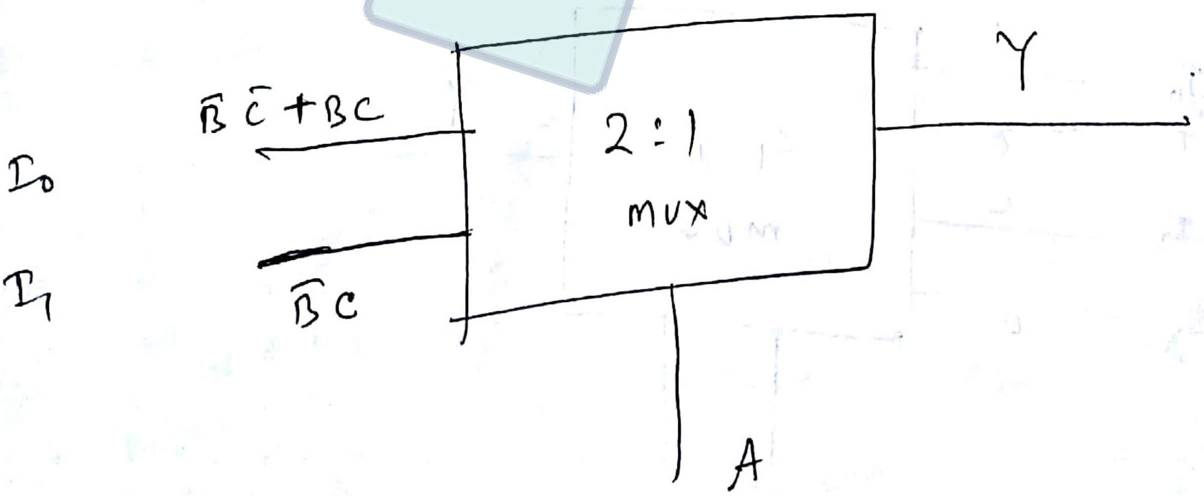
→ We have a select line 'A' (lets say)

→ A = 0, ~~A=~~ one group

A = 1, another group.

(Rel^n ~~of~~ of B and C with Y)

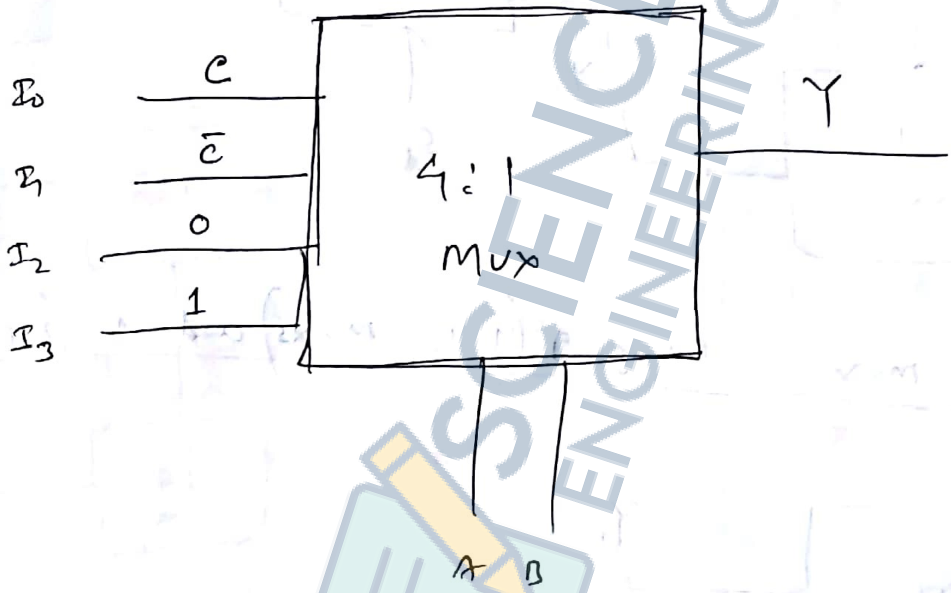
A	B	C	Y	Y	Y
0	0	0	1	I_0	$\bar{B}\bar{C} + BC$
0	0	1	0		
0	1	0	0		
0	1	1	1		
1	0	0	0	I_1	$\bar{B}C$
1	0	1	1		
1	1	0	0		
1	1	1	0		



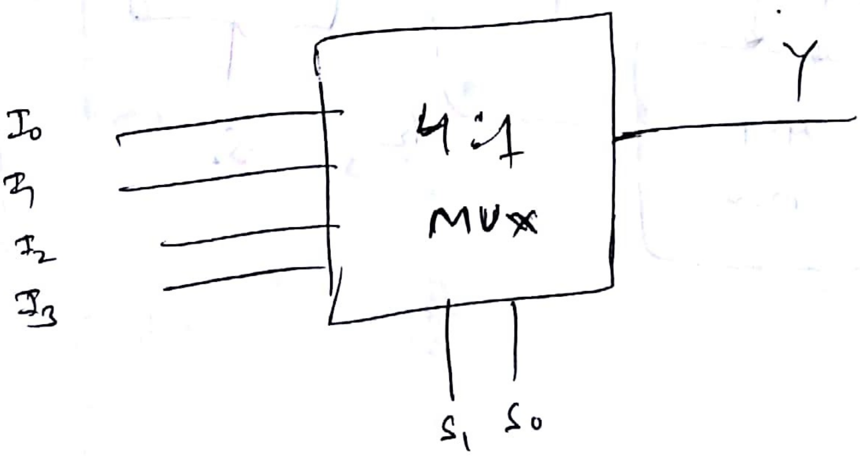
10) BPUT-2009

$f(A, B, C) = \Sigma(1, 2, 6, 7)$ Using 4:1 MUX.

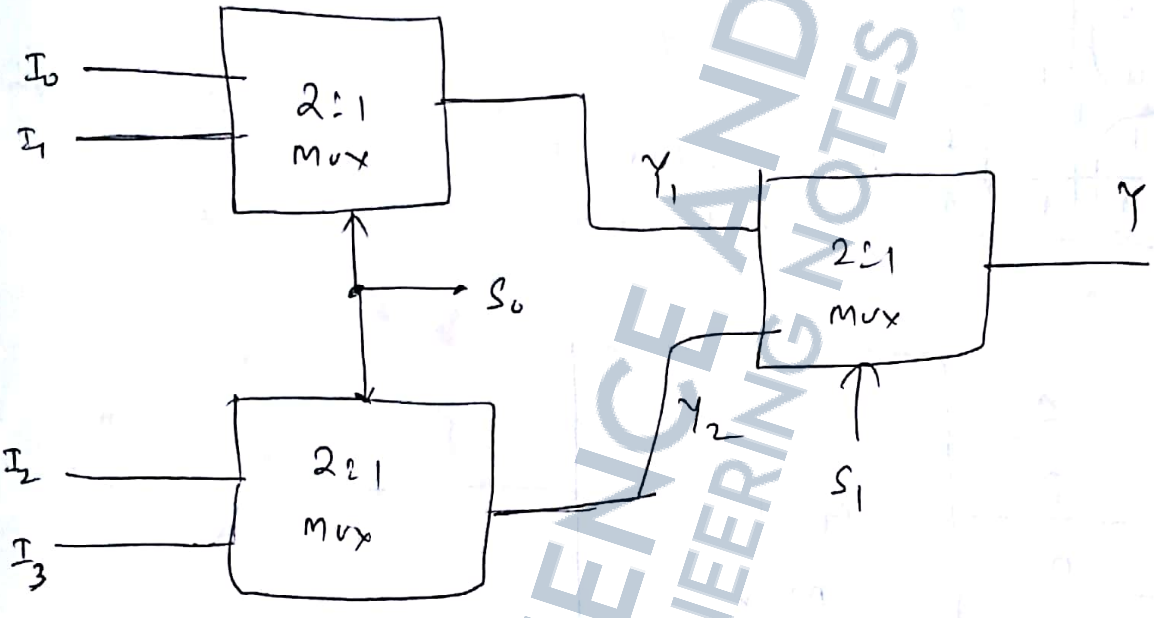
A	B	C	Y	Y	Y
0	0	0	0	I_0	$Y = C$
0	0	1	1		
0	1	0	1	I_1	$Y = \bar{C}$
0	1	1	0		
1	0	0	0	I_2	$Y = 0$
1	0	1	0		
1	1	0	1	I_3	$Y = 1$
1	1	1	1		



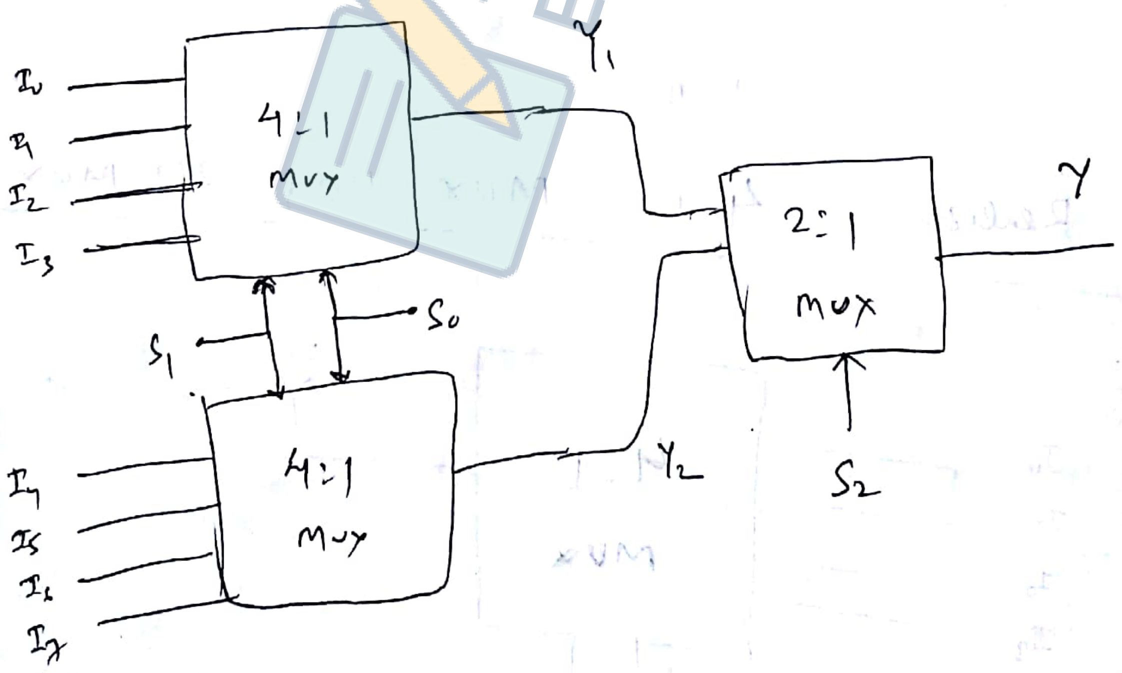
11) Realize 4:1 MUX using 2:1 MUX



S_1	S_0	Y	Y
0	0	I_0	Y_1
0	1	I_1	
1	0	I_2	Y_2
1	1	I_3	



12) 8:1 mux using 2 (4:1 muxs) and 1 (2:1 MUX)



8:1 Mux

S_2	S_1	S_0	Y
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

For

$S_2 = 0$, Y_1 is selected.

Y_1 depends on $S_1 S_0$

Lets check

$S_2 = 0$, $S_1 = 0$, $S_0 = 0$.

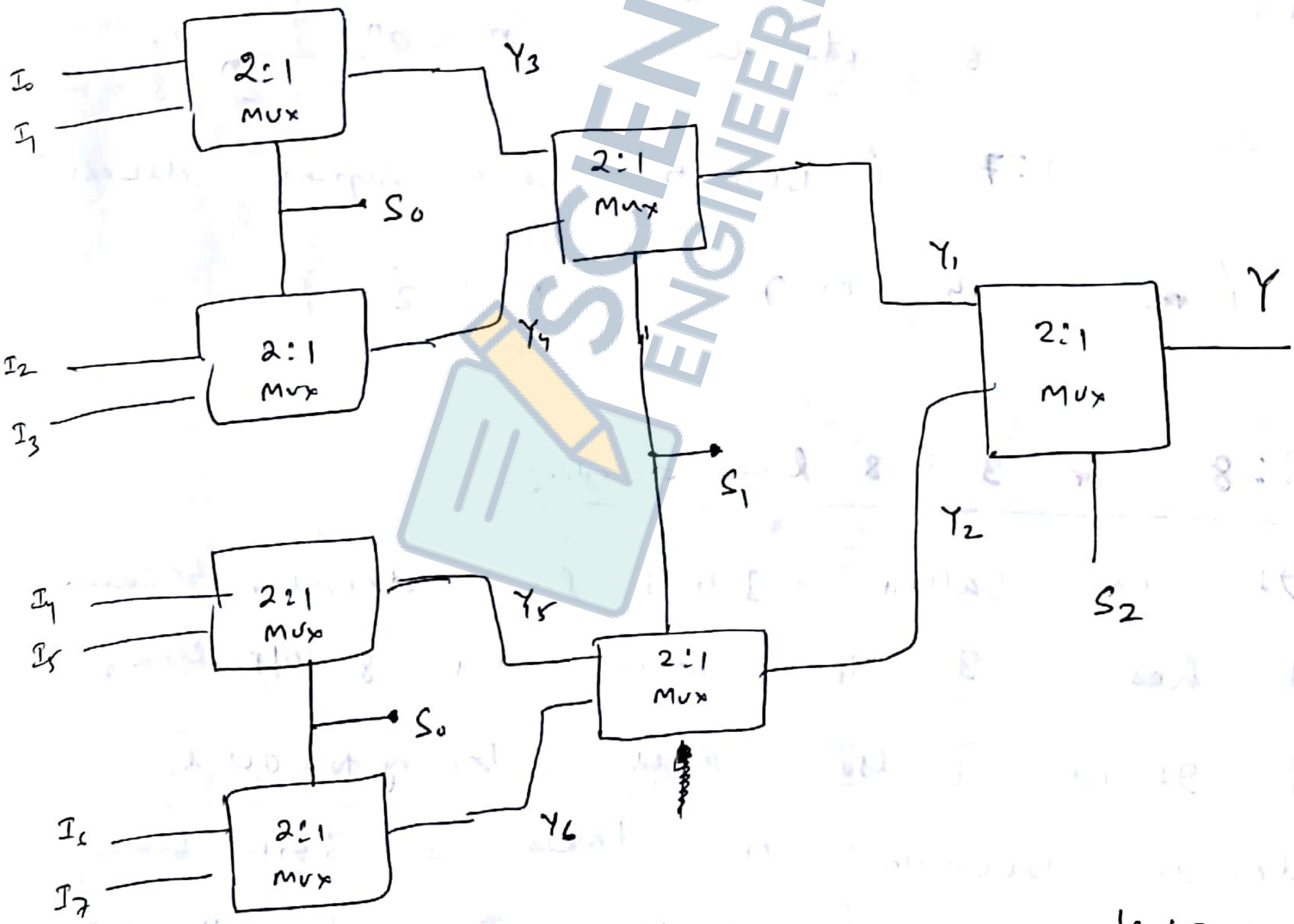
$S_1 = 0$, $S_0 = 0$, $Y_1 = I_0$

$Y_2 = I_4$

Again since $S_2 = 0$,

Y_1 is selected i.e I_0.

137 8:1 mux using 2:1 mux



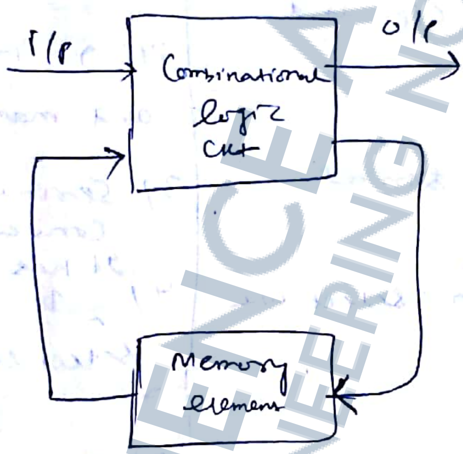
Total 2:1 mux required $= 4+2+1$

$= 7$

Sequential Logic ckt :-

→ In Sequential logic ckt the o/p not only depends upon the present i/p but also depends on the past i/p.

→ Although every digital system is likely to have Combinational ckts, most systems encountered in practice also include storage elements.



→ A block diagram of Sequential ckt is shown in fig. It is a Combination of a Combinational logic ckt and memory element.

→ Therefore the o/p of the Sequential logic ckt depends on the present state of i/p and state of the memory element.

→ The state of memory element in term depends on the previous state o/p, which in term depends on the previous state i/p.

→ The Basic Sequential logic ckts are Latch and flip-flops.

Latches :-

→ A latch is a type of temporary storage device that has 2 stable states (bistable)

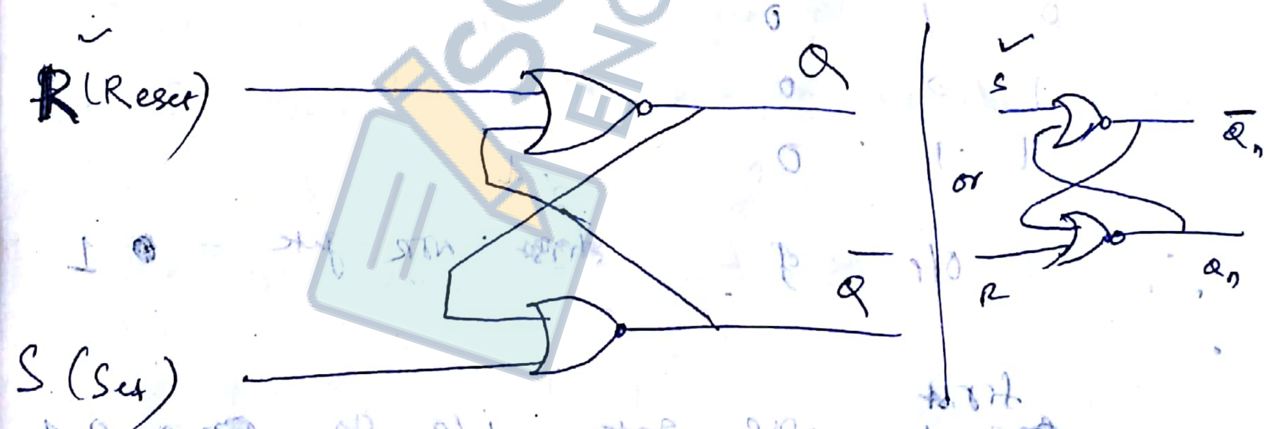
→ Two stable states are SET (1) and RESET (0). they (Latches) can retain either of these states indefinitely, making them useful as storage devices.

→ For storing 1 bit data, latch is used.

SR Latch :-

→ The SR Latch is a ckt with 2 cross-coupled NOR gates or two cross coupled NAND gates.

→ It has 2 i/ps labelled S for Set and R for Reset



Imp

S	R	Q_{n+1}	$\overline{Q}_{n+1}$	Remark
0	0	Q_n	$\overline{Q}_n$	No change
0	1	0	1	Reset
1	0	1	0	Set
1	1	0	0	Invalid

→ In this case, it acts as storage element

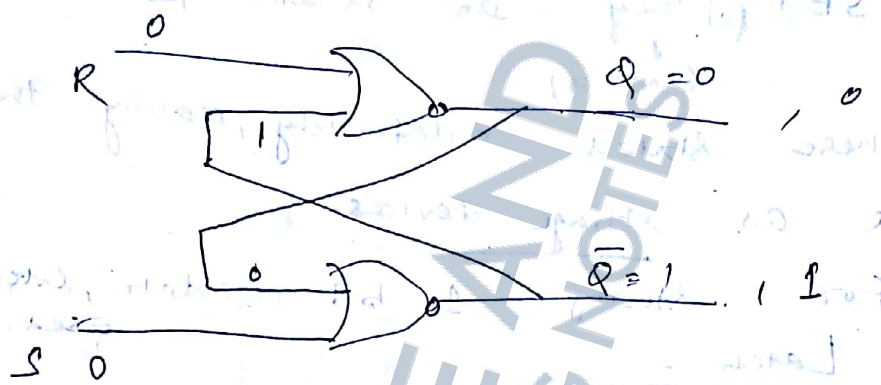
NOT used

Explanation :-

Let $\underline{S=0, R=0}$, $Q=0$

~~Since Q=0, $\bar{Q}=1$~~ ,

Now i/p for 2nd NOR gate is 0, 0



~~Q = 0~~

NOR gate

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

∴ O/p of 2nd NOR gate = 1

For first NOR gate i/p is 0, 1

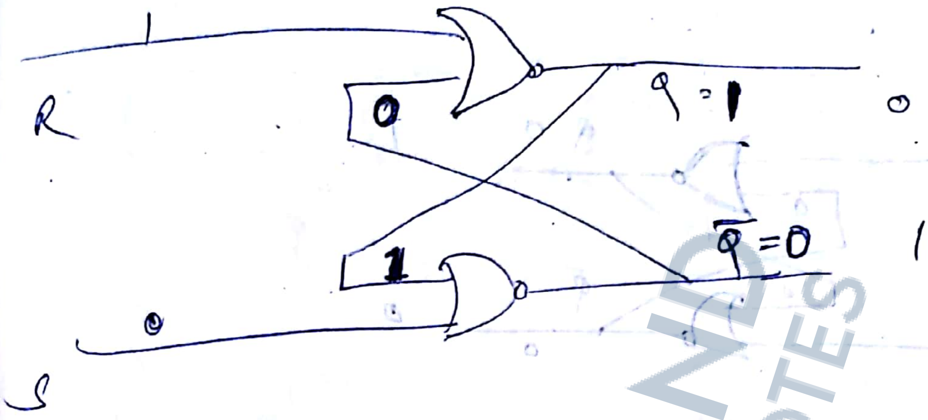
O/p will be 0.

∴ The o/p Remains unchanged

S	R	Q_{n+1}	$\bar{Q}_{n+1}$	Remarks
0	0	Q_n	$\bar{Q}_n$	No change

Case – II

$S = 0$, $R = 1$, ~~Q~~ $Q = 1$



Let $Q = 1$,

i/p for ~~2nd~~ 2nd NOR gate is $\boxed{1, 0}$.

o/p is 0.

Now i/p for ~~2nd~~ first NOR gate is 1, 0

(because o/p of 2nd NOR gate connected to first NOR gate)

o/p of first NOR gate is 0.

Now the 0 will be connected to 2nd NOR gate i.e 0, 0 i/p.

o/p of 2nd NOR will be 1.

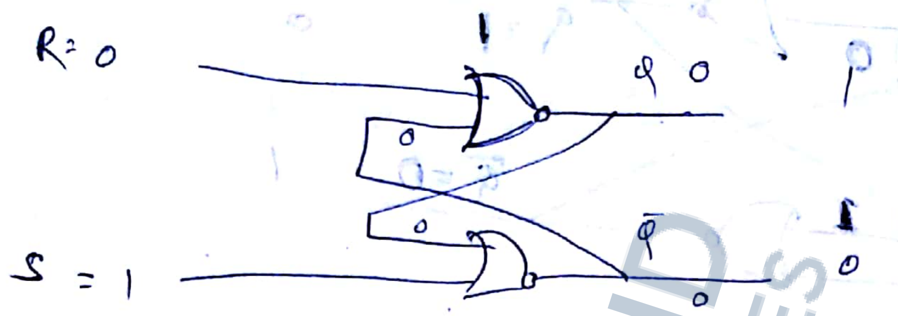
$Q = 0$, $\bar{Q} = 1$

Q is reset to 0.

S	R	Q_{n+1}	$\bar{Q}_{n+1}$	Remark
0	1	0	1	Reset.

Case - III

S = 1, R = 0, Q = 0



$\underline{Q = 0}$, i/p for 2nd gate is $\underline{0, 1}$

O/P of 2nd NOR gate is 0

Now, i/p for first gate is 0, 0

O/P of " " is $\boxed{1}$ Q

Now i/p for 2nd gate is $\underline{1, 1}$

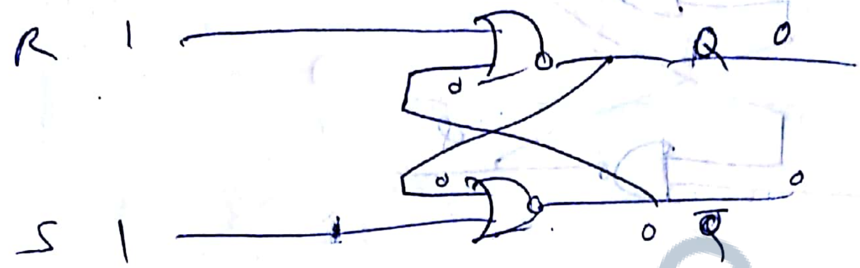
O/P of 2nd gate $\boxed{0}$ $\bar{Q}$

$\therefore$ Q is set to 1.

S	R	Q_{n+1}	$\bar{Q}_{n+1}$	Remark
1	0	1	0	Set

Case IV

$S = 1, \quad R = 1, \quad Q = 0$



$Q = 0$, i/p for 2nd NOR gate is 0,1.
o/p of 1 " " " is 0.

Now i/p for first NOR gate is 1,0
o/p of first NOR gate is 0

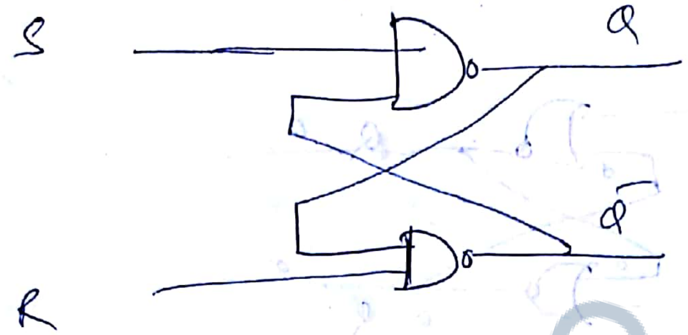
from now onwards the o/p & $\bar{\text{o/p}}$ both will remain at 0,0
which is Invalid.

Because $\bar{Q}$ should be complemented of Q, but both remain at 0,0.

S	R	Q_{n+1}	$\bar{Q}_{n+1}$	Remark
1	1	0	0	Invalid

Another explanation
when both o/ps are 0,0 and we give R=0, & S=0
then it should retain the value 0,0, as R=0, S=0 is
the requirement for a latch behaving as memory device.
But the o/p is 0, 1. which is not expected. So S=1, R=1
is never used. It is a unused or Invalid i/p.

SR Latch with NAND gates :-

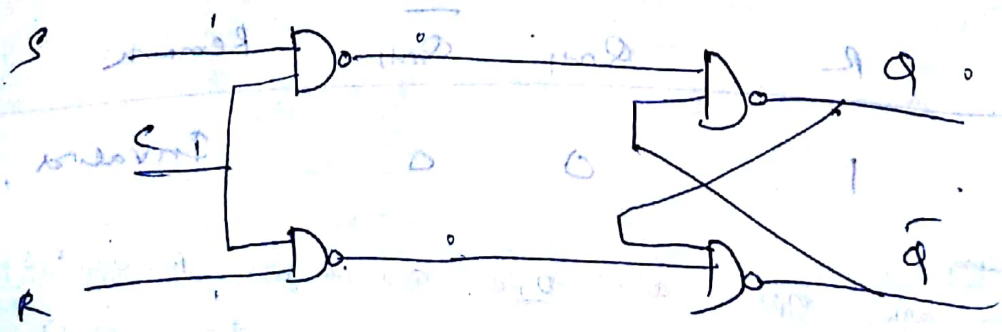


S	R	Q_{n+1}	$\bar{Q}_{n+1}$	Remark
0	0	1	1	Invalid
0	1	1	0	~~Reset~~
1	0	1	0	Set
1	1	Q_n	$\bar{Q}_n$	No change

Imp

S	R	Q_{n+1}	$\bar{Q}_{n+1}$	Remark
0	0	1	1	Invalid / Unused
0	1	1	0	Set
1	0	0	1	Reset
1	1	Q_n	$\bar{Q}_n$	No Change → Used as memory element.

SR Latch with Control I/P



C	S	R	Q_{n+1}	$\bar{Q}_{n+1}$	Remark
0	X	X	Q_n	$\bar{Q}_n$	No change
1	0	0	Q_n	$\bar{Q}_n$	No change
1	0	1	0	1	Reset
1	1	0	1	0	Set
1	1	1	1	1	Invalid

If C=0 (← Control Signal), whatever be the value of S & R (X, X), X → Don't care, the o/p remains unchanged.

(C=0, o/p of first 2, Nand gates is 1, 1) 'No change', i/ps for the ~~...~~ this ~~...~~ is ~~...~~ second 2 Nand gates)

D Latch:-

→ One way to eliminate the undesirable condition of indeterminate / invalid condⁿ in SR Latch is to ensure that S & R should never be equal.

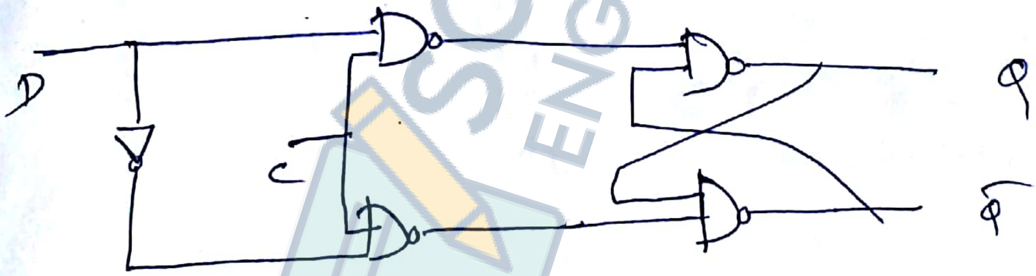
→ This is done by D latch.

→ It has 2 i/ps : D (Data)

C (Control)

2 o/p → Q & $\bar{Q}$

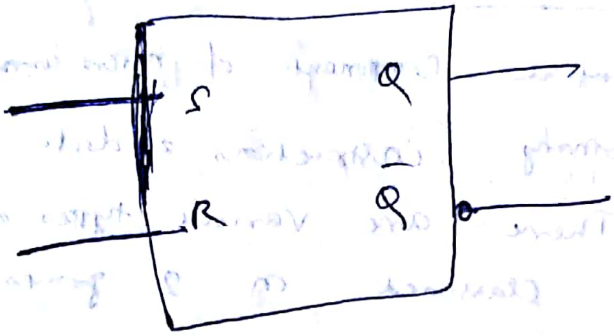
→ We have connected a NOT gate in between SR Latch to convert into D latch.



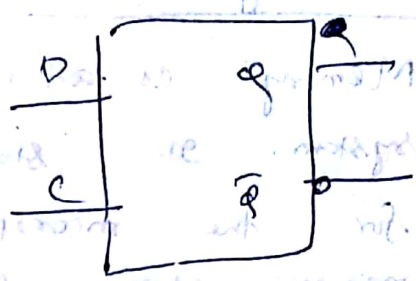
C	D	Q_{n+1}	$\overline{Q_{n+1}}$	Remark
0	X	Q_n	$\overline{Q_n}$	No change
1	0	0	1	'0' is Latched
1	1	1	0	'1' is Latched.

If Control is not enabled, it will remain as it is.

If Control is 1, $\begin{cases} D = 0, & Q = 0 \\ D = 1, & Q = 1 \end{cases}$

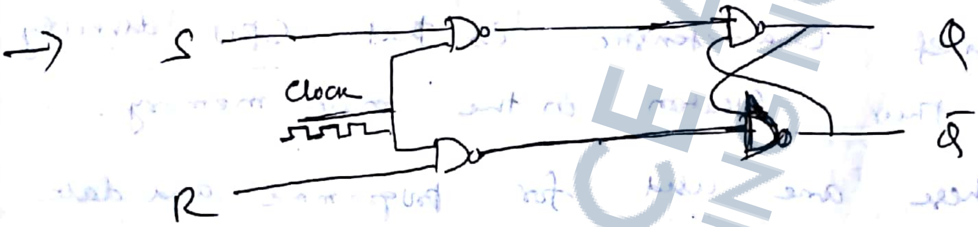


(SR Latch)



(D Latch)

Roots :- SR F/F



Clock	S	R	Q_{n+1} $\bar{Q}_{n+1}$	Remark
0	X	X	Q_n $\bar{Q}_n$	No change
↑	0	0	Q_n $\bar{Q}_n$	No change
↑	0	1	0 1	Reset
↑	1	0	1 0	Set
↑	1	1	1 1	Invalid

2-mark -2013 BPUT

Difference betn Latch & F/F

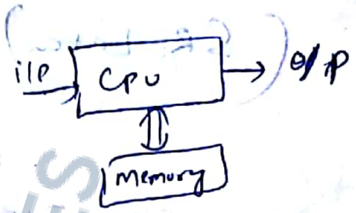
1) The difference betn latch and F/F is that latch does not have a clock signal, where as F/F always have a clock signal.

2) Latch is level triggering where as F/F is edge triggered.

Memory Classification:-

Memory is an essential component of micro computer system. It stores binary instructions & data for the microprocessor. There are various types of memory which can be classified in 2 groups such as

1. Primary Memory.
2. Secondary Memory.



Primary Memory:-

(writing a C-program)

The chief characteristic is that CPU directly accesses their location in the main memory.

These are used for programme and data storage, during computer operations. Total capacity of a primary memory varies from a few KB for small computing system to several MB/GB for large systems.

→ If a memory has n address lines then the memory capacity is 2^n bytes.

→ Basically the primary memory are divided into 2 groups

1. Read/write memory (RAM)
2. Read only memory (ROM)

Read/Write Memory (RAM)

The microprocessor can write into or read from this memory. It is popularly known as RAM (Random Access Memory)

It is used primarily for information that is likely to be altered, such as writing programs or receiving data.

→ This memory is volatile, meaning that when power is turned off, all the contents are destroyed.

→ Two types of R/W memories – static and dynamic. (SRAM / DRAM)

SRAM (Static RAM)

→ This memory is made up of flip-flop and it stores the bit as a voltage.

→ Each memory cell requires six transistors.

→ Therefore the memory chip has low density ~~and~~ but high speed.

→ This memory is more expensive & consumes more power than dynamic memory.

→ In high speed processor, SRAM known (as Cache memory), ~~are~~ included on the processor chip.

→ In addition, high speed cache memory is also included external to the processor to improve the performance of a system.

DRAM (Dynamic RAM)

→ This memory is made up of MOS transistor and it stores the bit as a charge.

→ The advantage of dynamic memory are that it has high density and low power consumption and is cheaper than static memory.

→ The disadvantage is that the charge (bit information) leaks, therefore stored information needs to be read and written again every few milliseconds.

→ This is called refreshing the memory, and it requires extra circuitry, adding to the cost of the system.

V. Imp

Static RAM	Dynamic RAM
1) It uses registers & latches / Flip flops for data storage	1) It uses capacitor for data storage.
2) The stored information are retained in it as long as power supply is ON.	2) It losses its content in a very short time even though the power supply is ON.
3) It does n't require refreshing ckt.	3) It requires refreshing ckt
4) It is used for small amount of data storage	4) It is used for large amount of data storage
5) It consumes more power and are more expensive	5) It consumes less power & less expensive
6) Faster	6) Slower
7) It stores the bit as a voltage.	7) It stores the bit as a charge

Both SRAM & DRAM are volatile.

ROM (Read-only-Memory)

- The ROM is a non volatile memory.
- It retains information even if the power is turned off.

Imp
- The ROM, like the RAM is random-access memory, but the term RAM traditionally means a random-access read/write memory.
- This memory is used for programs and data that need not to be altered.
- As the name suggests, the information can be read only, which means once a bit pattern is stored, it is permanent.

ROM are classified into 5 types, such as

(a) Masked ROM
(b) PROM
(c) EPROM
(d) EEPROM
(e) Flash Memory

a) Masked ROM :-

In this ROM, a bit pattern is permanently recorded by the masking and metallization process. Memory manufactures are generally equipped to do this process. It is expensive and specialized process. But economical for large production quantities.

PROM (Programmable Read only Memory) :-

→ This memory has nichrome and polysilicon wires arranged in a matrix

→ These wires can be functionally viewed as diodes or fuses. This memory can be programmed by the user with a special PROM programmer that selectively burns the fuses according to the bit pattern to be stored. This process is known as burning the PROM and the information is stored permanently.

< like Burning a Read only CD >

EPROM (Erasable Programmable Read only memory)

The information can be erased by exposing the chip to ultra violet light and the chip can be reprogrammed. Because the chip can be used many times, this memory is ideally suited for product development, experimental projects and college laboratories.

EE - PROM (Electrically Erasable PROM)

This memory is functionally similar to EPROM, ~~[illegible]~~ except that information can be altered by using electrical signals at the register level rather than erasing all the information. This has an advantage in field and remote control application.

Flash Memory :-

This is a variation of EE-PROM that is becoming popular. The major difference between the flash memory and EE PROM is the erase procedure. The EE-PROM can be erased at a register level, but the flash memory must be erased either in its entirely or at the sector (block) level. Faster than EEPROM.

Secondary Memory :-

If it is necessary to store more data than the maximum capacity of primary memory then secondary memory is employed.

→ The CPU can't directly access a secondary memory device. The secondary memories can be accessed only through I/O i.e input/output ports or in a serial format using proper software and hardware.

→ Magnetic tapes, floppy disks, CD-ROM, Hard-drive, Pen drive etc are common examples of secondary memory.

— o —