UNIT - 1

Introduction :- 'Computer' is a machine that can store and process information. Most computers rely on a binary system that uses two variables 0 and 1 to complete tasks such as storing data calculating algorithms and displaying information.

Organization :- Group of people who work together and to reach a goal by using proper system.

System :- A set of things working together to accomplish particular Goal.

Ex :- School system, college system and railway system.

Number system :- Represent data in digital form.

Binary Number System :- A number is made up of a collection of digits and it has two parts.

ⓐ Integer part
ⓑ Fraction part

Both are separated by a radix point (.). The number is represented as

$$\underbrace{d_{n-1}\ d_{n-2}\ ----\ d_1\ d_0}_{\text{Integer part}}\ \underset{\text{Radix point}}{\downarrow}\ \underbrace{d_{-1}\ d_{-2}\ ---\ d_{-m}}_{\text{Fractional Part.}}$$

Number systems are classified as

(a) Binary Number system
(b) Decimal Number System
(c) Octal Number system
(d) Hexa decimal Number System.

(a) The Binary number system is a radix-2. This is represented interms of 0 and 1. The radix point is known as the binary point and 0 and 1 is a binary bits.

(b) The decimal number system is a radix-10. These are 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9 (0-9). The radix point is known as decimal point.

(c) Octal number system is a radix 8. They are 0, 1, 2, 3, 4, 5, 6, and 7 (0-7). The radix point is known as Octal point.

(d) The Hexadecimal number system is a radix 16. They are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15. The radix point is known as Hexadecimal point. The decimal equivalent of A, B, C, D, E and F are 10, 11, 12, 13, 14 and 15.

Radix :- Number of symbols presented in a respective number system is called Radix.

Ex:- 1 0 1 1 0 1 0

(Most significant bit) MSB

LSB (Least Significant bit)

Decimal number system to Binary number system conversion :-

① $(25)_{10} = (?)_2$

$$\begin{array}{r|l} 2 & 25 \\ 2 & 12 - 1 \\ 2 & 6 - 0 \\ 2 & 3 - 0 \\ & 1 - 1 \end{array}$$

From Bottom to top

$$\boxed{\therefore (25)_{10} = (11001)_2}$$

(Successive Division method)

② $(72)_{10} = (?)_2$

$$\begin{array}{r|l} 2 & 72 \\ 2 & 36 - 0 \\ 2 & 18 - 0 \\ 2 & 9 - 0 \\ 2 & 4 - 1 \\ 2 & 2 - 0 \\ & 1 - 0 \end{array}$$

From Bottom to Top

$$\boxed{\therefore (72)_{10} = (1001000)_2}$$

⇒ Fractional part :-

③ $(0.25)_{10} = (?)_2$

Top to Bottom

$0.25 \times 2 = 0.5 \rightarrow 0$
$0.5 \times 2 = 1.0 \rightarrow 1$
$0.0 \times 2 = 0.0 \rightarrow 0$
$0.0 \times 2 = 0 \rightarrow$ ignore it

whenever we get repeated numbers just ignore it

$$\boxed{\therefore (0.25)_{10} = (0.010)_2}$$

④ $(0.8125)_{10} = (?)_2$

$0.8125 \times 2 = 1.6250 \rightarrow 1$
$0.6250 \times 2 = 1.250 \rightarrow 1$
$0.250 \times 2 = 0.50 \rightarrow 0$
$0.50 \times 2 = 1.0 \rightarrow 1$
$0.0 \times 2 = 0.0 \rightarrow 0$

Top to Bottom

$$\boxed{\therefore (0.8125)_{10} = (0.11010)_2}$$

⑤ $(10.625)_{10} = (?)_2$

$$\begin{array}{r|l} 2 & 10 \\ \hline 2 & 5 - 0 \\ \hline 2 & 2 - 1 \\ \hline & 1 - 0 \\ \hline \end{array}$$

(1010)

$0.625 \times 2 = 1.250 \rightarrow 1$

$0.250 \times 2 = 0.50 \rightarrow 0$

$0.50 \times 2 = 1.0 \rightarrow 1$

$0.0 \times 2 = 0.0 \rightarrow 0$

(1010)

$$\boxed{\therefore (10.625)_{10} = (1010.1010)_2}$$

⑥ $(25.125)_{10} = (?)_2$

$$\begin{array}{r|l} 2 & 25 \\ \hline 2 & 12 - 1 \\ \hline 2 & 6 - 0 \\ \hline 2 & 3 - 0 \\ \hline & 1 - 1 \\ \hline \end{array}$$

(11001)

$0.125 \times 2 = 0.250 \rightarrow 0$

$0.250 \times 2 = 0.50 \rightarrow 0$

$0.5 \times 2 = 1.0 \rightarrow 1$

$0.0 \times 2 = 0.0 \rightarrow 0$

(0010)

$$\boxed{\therefore (25.125)_{10} = (11001.0010)_2}$$

Binary Number system to Decimal number system conversion :-

① $(1101)_2 = (\ )_{10}$ (Successive multiplication method)

1	1	0	1
2^3	2^2	2^1	2^0

$1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 4 + 0 + 1 = 13$ $\boxed{\therefore (1101)_2 = (13)_{10}}$

② $(101011)_2 = (?)_{10}$

1	0	1	0	1	1
2^5	2^4	2^3	2^2	2^1	2^0

$1\times2^5 + 0\times2^4 + 1\times2^3 + 0+2^2 + 1\times2^1 + 1\times2^0 = 32+0+8+0+2+1$
$= 43$

$\boxed{\therefore (101011)_2 = (43)_{10}}$

③ $(101.10)_2 = (?)_{10}$

1	0	1	.	1	0
2^2	2^1	2^0		2^{-1}	2^{-2}

$1\times2^2 + 0\times2^1 + 1\times2^0. + 1\times2^{-1} + 0\times2^{-2}$

$= 4+0+1.+0.5+0$

$= 5+0.5 = 5.5$

$\boxed{\therefore (101.10)_2 = (5.5)_{10}}$

④ $(11010.010)_2 = (?)_{10}$

1	1	0	1	0	.	0	1	0
2^4	2^3	2^2	2^1	2^0	.	2^{-1}	2^{-2}	2^{-3}

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

$= 16+8+0+2+0.+0+0.25+0$

$= 26+0.25 = 26.25$

$\boxed{\therefore (11010.010)_2 = (26.25)_{10}}$

⑤ $(10010)_2 = (?)_{10}$

1	0	0	1	0
2^4	2^3	2^2	2^1	2^0

$1\times2^4 + 0\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0$
$= 16+0+0+2+0 = (18)_{10}$

⑥ $(011.01)_2 = (?)_{10}$

0	1	1	.	0	1
2^2	2^1	2^0	.	2^{-1}	2^{-2}

$0\times2^2 + 1\times2^1 + 1\times2^0. + 0\times2^{-1} + 1\times2^{-2}$
$= 0+2+1.+0+0.25 = (3.25)_{10}$

- **Binary codes**
  - Numeric codes
    - Weighted codes
      - Binary
      - BCD
        - 8421
        - 2421
        - 3321
        - 4221
        - 5211
        - 5311
        - 5421
        - 6311
        - 7421
        - 74-2-1
        - 84-2-1
        - 642-3
    - Non-weighted codes
      - Excess-3
      - Gray
      - Five bit BCD codes
    - Self complementing codes
      - 2421
      - 5211
      - XS-3
      - 642-3
      - 84-2-1
    - Sequential codes
      - 8421
      - XS-3
    - Error correction & Detection
      - Parity
      - Hamming
    - Reflective codes
      - Gray code
    - cyclic codes
      - Gray code
  - Alphanumeric code
    - ASCII
    - EBCDIC
    - Hollerith

→ Binary coded decimal Numbers (BCD) :-

→ BCD code uses four bits to represents the decimal numbers. i.e (0-9). Each single decimal number can be represented by a four bit pattern.

→ 8421 is also Known as Natural BCD.

Ex:- 8421, 2421, 3321, 4221, 5211, 5311, 5421, 6314, 7421
84-2-1, 642-3.

→ Representation of BCD code

Ex:- ① 12 (Each digit is represented by four bits)
0001 0010

Ex:- ② 14 (Each digit is indicated by group of 4-bits)
0001 0100

There are 6 unused states in BCD (1010, 1011, 1100, 1101, 1110, 1111)
10, 11, 12, 13, 14, 15.

Decimal number		BCD
14	—	0001 0100
234	—	0010 0011 0100
239.56	—	0010 0011 1001 . 0101 0110
653.96	—	0110 0101 0011 . 1001 0110

* Represent 356 in BCD format

ans:- 3 5 6
0011 0101 0110.

⇒ Pure binary representation :-

Ex:- 14 — 8 4 2 1
1 1 1 0
$\underbrace{\quad\quad}$ Only 4-bits

Ex:- 12 - 8 4 2 1
1 1 0 0
$\underbrace{\quad\quad}$ Only 4-bits

In BCD

14
↓ ↓
0001 0100
$\underbrace{\quad\quad\quad}$ 8-bits required

12
↓ ↓
0001 0010
$\underbrace{\quad\quad\quad}$ 8-bits required.

Q) To represent 12 in binary what are the minimum no. of digits we required.

Binary → 8 4 2 1
1 1 0 0
only 4-bits

Q) To represent 12 in BCD what are the minimum no. of digits we required.

BCD → 1 2
↓ ↓
0001 0010
8-bits.

Note :-
From the above concept, we can conclude one thing that BCD is simple to represent decimal Numbers but some times it takes more no. of bits. So, it occupies memory. Arithmetic operations are more complex than the Binary.

Ex:-

92 in Binary representation

64 32 16 8 4 2 1
1 0 1 1 1 0 0

92 in BCD
↓ ↓
1001 0010

→ **Weighted codes** :- The weighted codes are those which obey the position weighting principle. Each position of the number represents a specific weight.

Positive weighted codes
- 8421
- 2421
- 3321
- 4221

Negative weighted codes
- 84-2-1
- 74-2-1

Ex:-

Decimal	8421	2421	3321	4221
0	0000	0000	0000	0000
1	0001	0001	0001	0001
5	0101	1010 0101	1010 0110	1001 0111
7	0111	1101 0111	1101	1101 1011

All these are positive weighted codes

Ex:-

Decimal	84-2-1	74-2-1
0	0 0 0 0	0 0 0 0
5	1 0 1 1	1 0 1 0
7	1 0 0 1	1 0 0 0
9	1 1 1 1	1 1 1 0

→ **Non-weighted codes** → Excess-3 code.
→ Gray code

→ **Self-complementing code** :- It is said to be Self-complementing if the code word of the 9's complement of N i.e 9-N can be obtained from the code word of N by interchanging all the 0's and 1's.

Ex:- 2421, 5211, 642-3, 84-2-1 & Excess-3.

2+4+2+1	5+2+1+1	6+4+2-3	8+4-2-1
=9	=9	=9	=9

Decimal digital	8421	2421	Excess-3
0	0000	0000	0011
1	0001	0001	0100
2	0010	0010	0101
3	0011	0011	0110
4	0100	0100	0111
5	0101	1011	1000
6	0110	1100	1001
7	0111	1101	1010
8	1000	1110	1011
9	1001	1111	1100

Ex:- ①

$$\overset{8421}{5 \text{ in Excess-3 is } 5+3=8=1000}$$

It can obtained by adding 3 to that binary number

1000 $\xrightarrow[\text{complement}]{\text{1's}}$ 0111 [EX-3 code of decimal number 4)

4 is the 9's complement of 5 (9-5=4)

∴ It is a selfcomplementing code

Ex:- ②

4 in 2421 = $\overset{2421}{0100}$

0100 $\xrightarrow{\text{1's complement}}$ 1011 (this is 2421 code for decimal number 5)

5 is the 9's complement of 4.

∴ It is a self-complementing code.

Ex:- ③

BCD code for 6 is $\overset{8421}{0110}$

0110 $\xrightarrow{\text{1's complement}}$ 1001 (this is BCD code for decimal number 9)

9 is not the 9's complement of 6.

∴ BCD is not a self complementing code.

Ex:- ④ pure Binary (8421) code for 7 is 0111

0111 $\xrightarrow{\text{1's complement}}$ 1000 = 8 (Binary code of decimal 8)

8 is not the 9's complement of 7

∴ Binary code (8421) is not a self complementing code.

⇒ Cyclic codes :- cyclic codes are those in which each successive code word differs from the preceding one in only 1-bit position. cyclic codes are also called as unit distance codes Ex:- Gray code.

* Gray code is also called as Reflective code. Reflective code means in 8421 code 0-7 is the mirror image of 8-15. Gray code is not a sequence code. that's why we can't do arithmetic operation by using this code.

Ex:-

Decimal NO.	Binary		Gray
0	0000	⟶	0000
1	0001	⟶	0001
2	0010	⟶	0011
3	0011	⟶	0010
4	0100	⟶	0110
5	0101	⟶	0111
6	0110	⟶	0101
7	0111	⟶	0100
8	1000	⟶	1100
9	1001	⟶	1101
10	1010	⟶	1111
11	1011	⟶	1110
12	1100	⟶	1010
13	1101	⟶	1011
14	1110	⟶	1001
15	1111	⟶	1000

ⓐ convert $(1010)_2$ to gray code

Sol 1010

↓

ans 1111

ⓑ convert $(0110)_2$ to gray code

0110

↓

ans 0101

X-OR truth table

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

→ Alphanumeric codes :– These are the codes which represent alphanumeric information i.e letters of the alphabet and decimal numbers as a sequence of 0's and 1's.

Eg :- ASCII, EBCDIC codes

→ ASCII → American standard code for information interchange

→ EBCDIC → Extended binary coded decimal interchange code.

→ Alphanumeric codes consists of numbers as well as alphabetic character.

→ It contains 26 Alphabets with capital & small letters, Numbers (0-9), punctuation marks and other symbols.

→ ASCII code is a 7-bit code and more commonly used worldwide. ∴ $2^7 = 128$ symbols are used to represent Infor'n.

→ EBCDIC code is a 8 bit code and used in large IBM computers. ∴ $2^8 = 256$ symbols are used

IBM → International Business machine.

32 → space	42 → *	61 → =
33 → !	43 → +	62 → >
34 → "	44 → ,	63 → ?
35 → #	45 → -	64 → @
36 → $	46 → .	65–90 (A–Z) capital letters
37 → %	47 → /	91 → [
38 → &	48 → 57 → (0-9)	92 → \
39 → (	58 → :	93 →]
40 →)	59 → ;	94 → ^
	60 → <	95 → _ (underscore)

96 → \`

97–122 (a-z) small Letters   0–31 → Character control

123 → {

124 → | (vertical bar)

125 → }

126 → ~

Note :- Binary – Decimal – ASCII (Basic phenamena to do ASCII problems).

Ex:- 1001001 1000001 1001101 → 73 65 77 → I A M

Ex:- 1001010 1001100 → ~~74~~ ~~76~~ → J L

→ ASCII codes are used in micro computer (or) personal computer

→ EBCDIC codes are used in Large computers.

→ Hollerith code :- This code is used in system to represent alphanumeric information.

→ It consists of 80 columns and 12 rows

→ It is a 12-bit code.

Ex:- 1010111 space 1000010 space 1000111 space 1000011

→ 87 32 66 32 ~~71~~ 32 67

→ W B G C

→ <u>Error correcting codes</u> :- Codes which allow Error detection and correction are called Error correcting codes.

Eg :- Hamming code.

→ Hamming code is a specific type of error correcting code that allows the detection and correction of single bit transmission errors. Hamming code algorithm can solve only single bite issues. These are used in satellite communication.

<u>Ex</u>:- Encode the data (or) message bits 0011 into the 7-bit even parity Hamming code.

<u>sol</u> Given message = 0011

Number of message bits $M = 4$

Number of parity bits required is calculated using the formula

$$2^P \geq m+p+1$$
$$2^P \geq 4+p+1$$
$$2^3 \geq 4+3+1$$
$$8 \geq 8$$

Decimal no	2^2	2^1	2^0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Number of parity bits $p = 3$

Total no of bits $m+p = 4+3 = 7$

7	6	5	4	3	2	1
			2^2		2^1	2^0
m_4	m_2	m_2	P_3	m_1	P_2	P_1
0	0	1	P_3	1	P_2	P_1

$P_1 = 1\ 3\ 5\ 7 = P_1 110$

$= 0110$ ($\because P_1 = 0$; to maintain the even parity)

$\therefore P_1 = 0.$

$P_2 = 2, 3, 6, 7 = P_2\,1\,0\,0$ To become the even parity ($\therefore P_2 = 1$)

$= 1100$

$\therefore P_2 = \phi$

$P_3 = 4, 5, 6, 7 = P_3\,1\,0\,0$ to become the even parity ($\therefore P_3 = 1$)

$= 1100$

$\therefore P_3 = 1$

Error position = By combining the parity bits

$P_3P_2P_1 = \cancel{P_1P_2P_3} = 0110 = (6)_{10}$

Error is located at 3^{rd} position

Total message bits = 0 0 1 1 1 1 0

After correcting = 0111110.

Ex: Generate hamming code for the message 1110

Sol

$2^p \geq p+m+1$ p = parity bits, m = message bits

$2^p \geq p+4+1$

$2^p \geq p+5$ p should be atleast 3 to satisfy the condition

$2^3 \geq 3+5 \therefore 8 \geq 8$ (true)

(the code may be any length the process will be same)

1	2	3	4	5	6	7
2^0	2^1		2^2			
P_1	P_2	m_1	P_3	m_2	m_3	m_4
P_1	P_2	1	P_3	1	1	0

(For even parity)

$P_1 \Rightarrow 1, 3, 5, 7 \rightarrow P_1\,110 \rightarrow 0110$ ($P_1 = 0$)

$P_2 \rightarrow 2, 3, 6, 7 \rightarrow P_2\,110 \rightarrow 0110$ ($P_2 = 0$)

$P_3 \rightarrow 4, 5, 6, 7 \rightarrow P_3\,110 \rightarrow 0110$ ($P_3 = 0$)

Total message bits

$= 0\ 0\ 1\ 0\ 1\ 1\ 0$

Odd parity :-

$P_1 \Rightarrow 1,3,5,7 \rightarrow P_1 110 \rightarrow 1110\ (P_1 = 1)$

$P_2 \Rightarrow 2,3,6,7 \rightarrow P_2 110 \rightarrow 1110\ (P_2 = 1)$

$P_3 \Rightarrow 4,5,6,7 \rightarrow P_3 110 \rightarrow 1110\ (P_3 = 1)$

Total message bits

$= P_1\ P_2\ m_1\ P_3\ m_2\ m_3\ m_4$

$= 1\ 1\ 1\ 1\ 1\ 1\ 0$

$\Rightarrow$ Error correction in Hamming code

Q3 A (7,4) hamming code is recieved as 1110000 determine the corrected code when even and odd parity.

Sol

1 2 3 4 5 6 7

1 1 1 0 0 0 0

To ensure that error is there are not

(even parity)

$E_1 \rightarrow 1,3,5,7 \rightarrow 1100 \rightarrow$ to make it even parity $E_1 = 0$.

$E_2 \rightarrow 2,3,6,7 \rightarrow 1100 \Rightarrow E_2 = 0$

$E_3 \rightarrow 4,5,6,7 \rightarrow 0000 \rightarrow E_3 = 0$

Error $= E_3 E_2 E_1 = 000$ (0^{th} position)

Odd parity

$E_1 \rightarrow 1,3,5,7 \rightarrow 1100 \rightarrow E_1 = 1$ (to make it odd parity)

$E_2 \rightarrow 2,3,6,7 \rightarrow 1100 \rightarrow E_2 = 1$ (to make it odd parity)

$E_3 \rightarrow 4,5,6,7 \rightarrow 0000 \rightarrow E_3 = 1$

Error $= E_3 E_2 E_1 = 111$ in 7^{th} position error is there

Corrected code may be 111000<u>1</u>

Q Determine the single error-correcting code for the information code 10111 for odd parity

Sol Given message bit
$m = 10111$.
By using trail and error method we should find parity bits

$\boxed{2^P \geq n+P+1}$ $\quad$ $\because n = m$

$2^1 \geq 5+1+1$ $\quad$ Let $p=1$
$2 \geq 7$ ✗
$2^2 \geq 5+2+1$ $\quad$ Let $p=2$
$2^2 \geq 8$ ✗
$2^3 \geq 5+3+1$ $\quad$ Let $p=3$
$8 \geq 9$ ✗
$2^4 \geq 5+4+1$ $\quad$ Let $p=4$
$16 \geq 10$ ✓

So, we need 4 parity bits.
we should take parity bits always powers of 2.
$2^0=1$; $2^1=2$, $2^2=4$, $2^3=8$
$2^4=16$; $2^5=32$ and so on.
Find the value to the parity

1 2 3 4 5 6 7 8 9
P_1 P_2 m_1 P_2 m_2 m_3 m_4 P_4 m_5
P_1 P_2 1 P_3 1 1 0 P_4 1

$\because m = 1\ 0\ 1\ 1\ 1$
$m_5\ m_4\ m_3\ m_2\ m_1$

Step ①

Bit destination	m_5	P_8	m_4	m_3	m_2	P_4	m_1	P_2	P_1
Bit Location	9	2^3 8	7	6	5	2^2 4	3	2^1 2	2^0 1
Information bits	1001	1000	0111	0110	0101	0100	0011	0010	0001
Parity bits	1	?	0	1	1	?	1	?	?
Received code	1	0	0	1	1	1	1	1	0

$P_1 \rightarrow P_1 m_1 m_2 m_4 m_5 = P_1\,1101$
To make it become odd
we kept $p=0$; so $\boxed{P_1 = 0}$

$P_2 \rightarrow P_2 m_1 m_3 m_4 = P_2\,110$ To make it become odd
we kept $P_2 = 1$; so $\boxed{P_2 = 1}$

$P_4 \rightarrow P_4 m_2 m_3 m_4 = P_4\,110$ To make it odd
we kept $P_4 = 1$; $\boxed{P_4 = 1}$

$P_8 \rightarrow P_8 m_5 = P_8\,1$ To make it odd
we kept $P_8 = 0$; $\boxed{P_8 = 0}$

Error position $\quad P_8 P_4 P_2 P_1$
$= 0110 = 6^{th}$ position

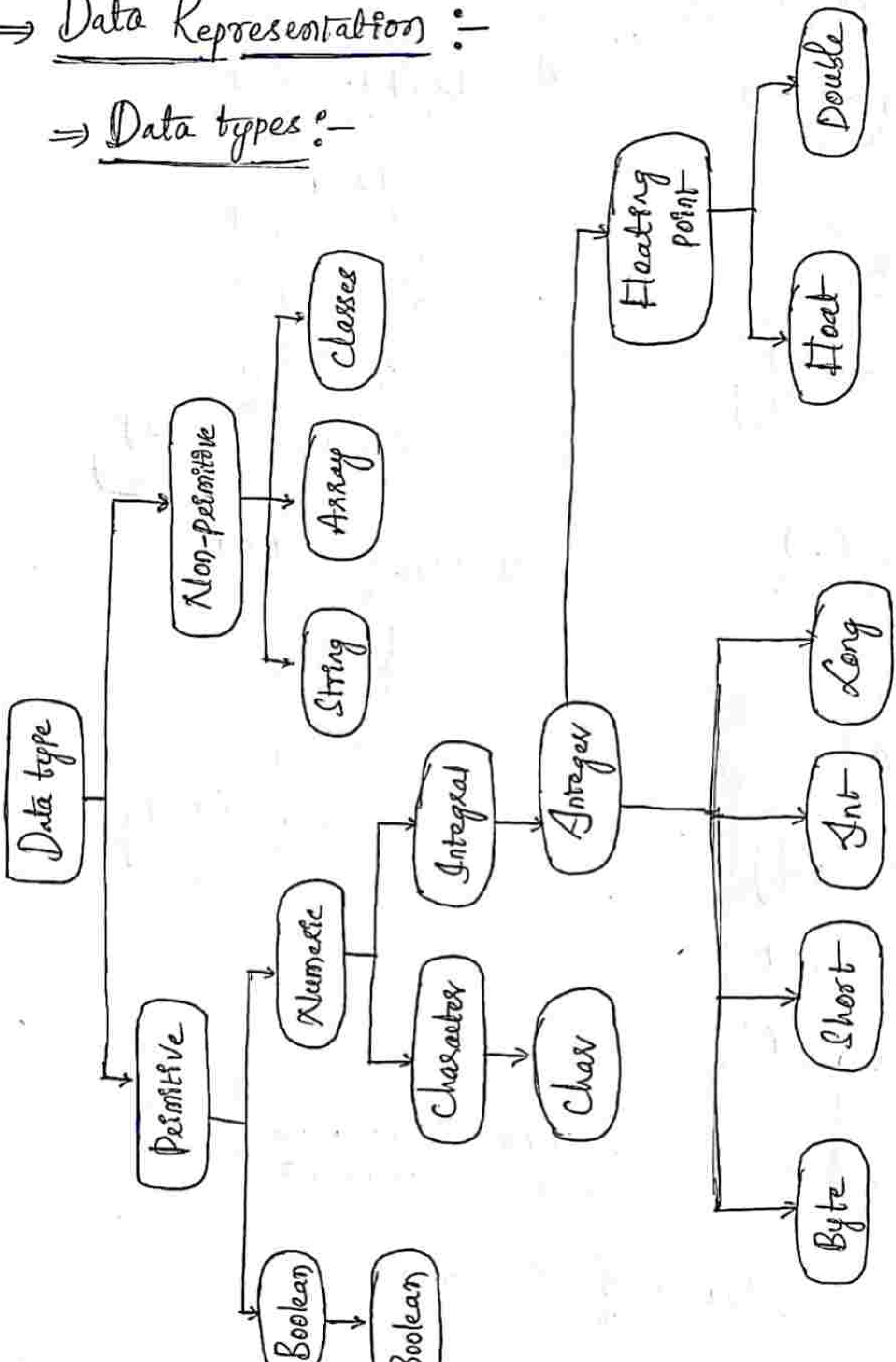
This is 9 bit hamming code

9 8 7 6 5 4 3 2 1
1 0 0 1 1 1 1 1 0 ✓
error in 6^{th} position

$\boxed{100011110}$ ✓

⇒ Data Representation :-

⇒ Data types :-



Decimal to octal :-

① $(20)_{10} = (?)_8$

$$\begin{array}{r|l} 8 & 20 \\ \hline & 2-4 \end{array}$$

$$\boxed{\therefore (20)_{10} = (24)_8}$$

② $(1234)_{10} = (?)_8$

$$\begin{array}{r|l} 8 & 1234 \\ \hline 8 & 154-2 \\ \hline 8 & 19-2 \\ \hline & 2-3 \end{array}$$

$$\boxed{\therefore (1234)_{10} = (2322)_8}$$

③ $(183)_{10} = (?)_8$

$$\begin{array}{r|l} 8 & 183 \\ \hline 8 & 22-7 \\ \hline & 2-6 \end{array}$$

$$\boxed{\therefore (183)_{10} = (267)_8}$$

④ $(145)_{10} = (?)_8$

$$\begin{array}{r|l} 8 & 145 \\ \hline 8 & 18-1 \\ \hline & 2-2 \end{array}$$

$$\boxed{\therefore (145)_{10} = (221)_8}$$

Fractional part :-

① $(27.625)_{10} = (?)_8$

$$\begin{array}{r|l} 8 & 27 \\ \hline & 3-3 \end{array}$$

$0.625 \times 8 = 5.000 \rightarrow 5$

$0.000 \times 8 = 0.000 \rightarrow 0$

$$\boxed{\therefore (27.625)_{10} = (33.50)_8}$$

② $(3287.5100098)_{10} = (?)_8$

Sol Integer part

$$\begin{array}{r|l} 8 & 3287 \\ \hline 8 & 410-7 \\ \hline 8 & 51-2 \\ \hline & 6-3 \end{array}$$

$0.5100098 \times 8 = 4.0800 \rightarrow 4$

$0.0800 \times 8 = 0.640 \rightarrow 0$

$0.640 \times 8 = 5.125 \rightarrow 5$

$0.125 \times 8 = 1.000 \rightarrow 1$

$$\boxed{\therefore (3287.5100098)_{10} = (6327.4051)_8}$$

③ $(20.5)_{10} = (?)_8$

$$\begin{array}{r|l} 8 & 20 \\ \hline & 2-4 \end{array}$$

Fractional part

$0.5 \times 8 = 4.0 \rightarrow 4$

$0.0 \times 8 = 0.0 \rightarrow 0$

$$\boxed{\therefore (20.5)_{10} = (24.40)_8}$$

④ $(60.7)_{10} = (?)_8$

$$\begin{array}{r|l} 8 & 60 \\ \hline & 7-4 \end{array}$$

$0.7 \times 8 = 5.6 \rightarrow 5$

$0.6 \times 8 = 4.8 \rightarrow 4$

$0.8 \times 8 = 6.4 \rightarrow 6$

$0.4 \times 8 = 3.2 \rightarrow 3$

$0.2 \times 8 = 1.6 \rightarrow 1$

$$\boxed{\therefore (60.7)_{10} = (74.54631)_8}$$

⇒ Decimal to Hexadecimal :– (H = 16)

① $(20)_{10} = (?)_H$

$$\begin{array}{r|l} 16 & 20 \\ \hline & 1-4 \end{array}$$

$\therefore \boxed{(20)_{10} = (14)_H}$

② $(1234)_{10} = (?)_H$

$$\begin{array}{r|l} 16 & 1234 \\ \hline 16 & 77-2 \\ \hline & 4-13 \text{ (or) } D \end{array}$$

$\therefore \boxed{(1234)_{10} = (4D2)_{16}}$

③ $(20.5)_{10} = (?)_H$

$$\begin{array}{r|l} 16 & 20 \\ \hline & 1-4 \end{array}$$

$0.5 \times 16 = 8.0 \longrightarrow 8$

$0.0 \times 16 = 0.0 \longrightarrow 0$

$\therefore \boxed{(20.5)_{10} = (14.8)_{16}}$

④ $(675.625)_{10} = (?)_{16}$

$$\begin{array}{r|l} 16 & 675 \\ \hline 16 & 42-3 \\ \hline & 2-10 \text{ (or) } A \end{array}$$

$0.625 \times 16 = 10.000 \longrightarrow 10 \text{ (or) } A$

$0.000 \times 16 = 0.000 \longrightarrow 0$

$\therefore \boxed{(675.625)_{10} = (2A3.A)_{16}}$

⟹ Binary to octal :-

To convert binary to octal, starting from binary point make group of 3 bits and write its equivalent

① $(101)_2 = (?)_8$

$\overset{421}{101} \rightarrow 5$

$\boxed{\therefore (101)_2 = (5)_8}$

② $(1101)_2 = (\ )_8$

$\underbrace{001}_{1}\underbrace{101}_{5} = 15$

$\boxed{\therefore (1101)_2 = (15)_8}$

③ $(10.11001)_2 = (?)_8$

$\leftarrow 10 . 11001 \rightarrow$

$\underbrace{010}_{2} . \underbrace{110}_{6}\underbrace{010}_{2}$

$\boxed{\therefore (10.11001)_2 = (2.62)_8}$

④ $(01101010110.11)_2 = (?)_8$

$\leftarrow 011010110 . 11 \rightarrow$

$\underbrace{011}_{3}\ \underbrace{010}_{2}\ \underbrace{110}_{6} . \underbrace{110}_{6}$

$\boxed{\therefore (011010110.11)_2 = (326.6)_8}$

⑤ $(1101101.01101)_2 = (\ )_8$

$\underbrace{001}_{1}\underbrace{101}_{5}\ \underbrace{101}_{5} . \underbrace{011}_{3}\underbrace{010}_{2}$

append zero. append zero

$\boxed{\therefore (1101101.01101)_2 = (155.32)_8}$

⇒ Binary to Hexadecimal :-

To convert binary to Hexadecimal, Group 4-bits of binary and write its equivalent hexadecimal digit.

① $(1101011011)_2 = (?)_{16}$

$$\underbrace{0011}_{3}\ \underbrace{0101}_{5}\ \underbrace{1011}_{11\ (or)\ B}$$

$$\boxed{\therefore (1101011011)_2 = (35B)_{16}}$$

② $(1101011011 \cdot 110101)_2 = (\ )_{16}$

$$\underbrace{0011}_{3}\ \underbrace{0101}_{5}\ \underbrace{1011}_{11\ (or)\ B} \cdot \underbrace{1101}_{13\ (or)\ D}\ \underbrace{0100}_{4}$$

$$\boxed{\therefore (1101011011 \cdot 110101)_2 = (35B \cdot D4)_{16}}$$

③ $(10100110101111)_2 = (?)_{16}$

$$\underbrace{0010}_{2}\ \underbrace{1001}_{9}\ \underbrace{1010}_{10\ (or)\ A}\ \underbrace{1111}_{15\ (or)\ F}$$

$$\boxed{\therefore (10100110101111)_2 = (29AF)_{16}}$$

④ $(100101011 \cdot 101110)_2 = (\ )_{16}$

$$\underbrace{0001}_{1}\ \underbrace{0010}_{2}\ \underbrace{1011}_{11\ (or)\ B} \cdot \underbrace{1011}_{11\ (or)\ B}\ \underbrace{1000}_{8}$$

$$\boxed{\therefore (100101011 \cdot 101110)_2 = (12B \cdot B8)_{16}}$$

⟹ Octal to other Number systems :-

⟹ Octal to Decimal :-

① $(24)_8 = (?)_{10}$

2	4
8^1	8^0

$2\times8^1+4\times8^0$

$= 16+4 = 20$

$\boxed{\therefore (24)_8 = (20)_{10}}$

② $(24.4)_8 = (?)_{10}$

2	4	.	4
8^1	8^0	.	8^{-1}

$= 2\times8^1+4\times8^0.+4\times8^{-1}$

$= 16+4.+4\times\frac{1}{8^2}$

$= 16+4.+0.5 = 20.5$

$\boxed{\therefore (24.4)_8 = (20.5)_{10}}$

③ $(6327.4051)_8 = (?)_{10}$

6	3	2	7	.	4	0	5	1
8^3	8^2	8^1	8^0	.	8^{-1}	8^{-2}	8^{-3}	8^{-4}

$= 6\times8^3+3\times8^2+2\times8^1+7\times8^0.+4\times\frac{1}{8}$
$+0\times\frac{1}{8^2}+5\times\frac{1}{8^3}+\frac{1}{8^4}\times1$

$= 3072+192+96+7.+0.5100098$

$= (3367.5100098)_{10}$

④ $(1234.342)_8 = (\ )_{10}$

1	2	3	4	.	3	4	2
8^3	8^2	8^1	8^0	.	8^{-1}	8^{-2}	8^{-3}

$1\times8^3+2\times8^2+3\times8^1+4\times8^0.+3\times\frac{1}{8}+4\times\frac{1}{8^2}$
$+2\times\frac{1}{8^3}$

$= 512+128+24+4+.0.375+0.062+0.003$

$= (668.440)_{10}.$

⇒ **Octal to Binary :-** To convert octal to binary first replace each octal digit by its 3-bit binary equivalent.

① $(15)_8 = (?)_2$

$1 \rightarrow \overset{4\,2\,1}{001}$

$5 \rightarrow 101$

$\boxed{\therefore (15)_8 = (001101)_2}$

② $(736)_8 = (?)_2$

$7 \rightarrow \overset{4\,2\,1}{111}$

$3 \rightarrow 011$

$6 \rightarrow 110$

$\boxed{\therefore (736)_8 = (111011110)_2}$

③ $(563)_8 = (?)_2$

$5 \rightarrow 101$

$6 \rightarrow 110$

$3 \rightarrow 011$

$\boxed{\therefore (563)_8 = (101110011)_2}$

④ $(725)_8 = (\ )_2$

$7 \rightarrow 111$

$2 \rightarrow 010$

$5 \rightarrow 101$

$\boxed{\therefore (725)_8 = (111010101)_2}$

⑤ $(326)_8 = (?)_2$

$3 \rightarrow 011$

$2 \rightarrow 010$

$6 \rightarrow 110$

$\boxed{(326)_8 = (011010110)_2}$

⑥ $(452)_8 = (\ )_2$

$4 \rightarrow 100$

$5 \rightarrow 101$

$2 \rightarrow 010$

$\boxed{\therefore (452)_8 = 100101010}$

⇒ Octal to Hexadecimal :- There is no direct conversion available for octal to hexadecimal. to convert octal number into a hexadecimal number by converting octal to binary then binary to hexadecimal (or) octal to decimal then decimal to hexadecimal.

Note :- $(\ )_8 \rightarrow (\ )_2 \rightarrow (\ )_{16}$
(or)
$(\ )_8 \rightarrow (\ )_{10} \rightarrow (\ )_{16}$

① $(356.63)_8 = (\ )_{16}$

Step① Octal to binary

Step② Binary to Hexadecimal

3 5 6 . 6 3
↓ ↓ ↓ ↓ ↓
011 101 110 . 110 011

$\underbrace{0000}_{0}\ \underbrace{1110}_{E=14}\ \underbrace{1110}_{E=14}\ .\ \underbrace{1100}_{C=12}\ \underbrace{1100}_{C=12}$

$\boxed{\therefore (356.63)_8 = (0EE.CC)_{16}}$

② $(247.52)_8 = (\ )_{16}$

Step① Octal to binary

Step② Binary to Hexadecimal

2 4 7 . 5 2
↓ ↓ ↓ ↓ ↓
010 100 111 . 101 010

$\underbrace{0000}_{0}\ \underbrace{1010}_{10 \text{ (or) } A}\ \underbrace{0111}_{7}\ .\ \underbrace{1010}_{10 \text{ (or) } A}\ \underbrace{1000}_{8}$

$\boxed{\therefore (247.52)_8 = (0A7.A8)_{16}}$

⇒ Hexadecimal to other Number system :-

⇒ Hexadecimal to binary :-

① $(2F9A)_{16} = (\ )_2$

2 → 0010
F → 1111
9 → 1001
A → 1010

$\boxed{\therefore (2F9A)_{16} = (0010111110011010)_2}$

(2) $(6A3)_{16} = (\ )_2$

$6 \rightarrow 0110$
$A \rightarrow 1010$
$3 \rightarrow 0011$

$\boxed{\therefore (6A3)_{16} = (011010100011)_2}$

(3) $(58C)_{16} = (\ )_2$

$5 \rightarrow 0101$
$8 \rightarrow 1000$
$C \rightarrow 1100$

$\boxed{\therefore (58C)_{16} = (010110001100)_2}$

(4) $(7DE3)_{16} = (\ )_2$

$7 \rightarrow 0111$
$D \rightarrow 1101$
$E \rightarrow 1110$
$3 \rightarrow 0011$

$\boxed{\therefore (7DE3)_{16} = (0111110111100011)_2}$

$\Rightarrow$ Hexadecimal to Decimal :-

(1) $(3A.2F)_{16} = (\ )_{10}$

3	A	.	2	F
16^1	16^0	.	16^{-1}	16^{-2}

$3 \times 16^1 + \overset{(A)}{10} \times 16^0 . + 2 \times 16^{-1} + 15 \times 16^{-2}$

$= 48 + 10 + \frac{2}{16} + \frac{15}{16^2}$

$\boxed{\therefore (3A.2F)_{16} = (58.1836)_{10}}$

(2) $(5E.7A)_{16} = (\ )_{10}$

5	E	.	7	A
16^1	16^0	.	16^{-1}	16^{-2}

$5 \times 16^1 + 14 \times 16^0 . + 7 \times \frac{1}{16^1} + 10 \times \frac{1}{16^2}$

$= 90 + 14 + 0.43 + 0.03$

$= (104.46)_{10}$

$\boxed{\therefore (5E.7A)_{16} = (104.46)_{10}}$

⟹ Hexadecimal to octal conversion :-

No direct conversion available, to convert hexadecimal to octal first convert given hexadecimal number into decimal / binary then into octal system.

Note :-

$$\boxed{\begin{array}{c} (\;)_{16} \rightarrow (\;)_{2} \rightarrow (\;)_{8} \\ \text{(or)} \\ (\;)_{16} \rightarrow (\;)_{10} \rightarrow (\;)_{8} \end{array}}$$

① $(B9F.AE)_{16} = (\;)_8$

✓ Hexadecimal to binary

✓ Binary to octal

$$\begin{array}{c|c|c|c|c|c} B & 9 & F & . & A & E \\ 1011 & 1001 & 1111 & . & 1010 & 1110 \end{array} = \underbrace{101}_{5}\;\underbrace{110}_{6}\;\underbrace{011}_{3}\;\underbrace{111}_{7}\;.\;\underbrace{101}_{5}\;\underbrace{011}_{3}\;\underbrace{100}_{4}$$

$$\boxed{\therefore (B9F.AE)_{16} = (5637.534)_8}$$

② $(A8C.BC7)_{16} = (\;)_8$

✓ Hexadecimal to binary

✓ Binary to octal

$$\begin{array}{c|c|c|c|c|c|c} A & 8 & C & . & B & C & 7 \\ 1010 & 1000 & 1100 & . & 1011 & 1100 & 0111 \end{array} =$$

$$\underset{1}{001}\;\underset{2}{010}\;\underset{4}{100}\;\underset{3}{011}\;\underset{4}{100}\;.\;\underset{5}{101}\;\underset{7}{111}\;\underset{0}{000}\;\underset{7}{111}$$

$$\boxed{\therefore (A8C.BC7)_{16} = (12434.5707)_8}$$

(16)

Complement of Numbers:

(or) $(r-1)$'s complement and r's complement

complements are used in systems to simplify the subtraction operation base (radix) r system there are two useful types of complements, r's complement (Radix complement) and $(r-1)$'s complement (Diminished Radix complement).

⇒ $(r-1)$'s complement :-

For a given Number 'N' have the no. of digits 'n' belonging to 'r' number system, then $(r-1)$ complement is given by $\boxed{(r^n-N)-1}$

⇒ r's complement :-

For a given number 'N' have the no. of digits 'n' belonging to 'r' number system, then r's complement is given by $\boxed{r^n-N}$

Binary (2) → 1's complement
Binary (2) → 2's complement

Octal (8) → 7's complement
Octal (8) → 8's complement

Decimal (10) → 9's complement
Decimal (10) → 10's complement

Hexadecimal (16) → 15's complement
Hexadecimal (16) → 16's complement

⇒ **1's and 2's complements :-**

The 1's complement of a binary number is obtained complementing all its bits, that is by replacing all 0's by 1's and all 1's by 0's.

Ex:- ① 0101101000 → (Given Binary Number)
1010010111 → (1's complement form)

The 2's complement of a binary number is obtained by adding '1' to its 1's complement.

Ex:- (1)

0101101000 → (Given binary number)
1010010111 → (1's complement form)
1 (adding 1)
111

1010011000 → 2's complement

⇒ **Binary addition**

0+0 = 0
0+1 = 1
1+0 = 1
1+1 = 1 0 → sum
↓
carry

1+1+1 = 1 1
↓ ↓
carry sum

(2) 0110101011 → Given binary number
1001010100 → 1's complement form
+1 → adding '1'

1001010101 → 2's complement form.

⇒ 1's complement Arithmetic :-

→ In 1's complement subtraction, add the 1's complement of the subtrahend to the minuend.

→ If there is a carryout, bring the carry around and add it to the LSB. This is called "End around carry"

→ Look at the sign bit (MSB). If this is a '0', the result is positive and is in true binary. If MSB is '1', the result is negative and is in its 1's complement form. Take its 1's complement and put -ve sign to get magnitude in binary.

Ex ①

① Subtract 14 from 25 using 8-bit 1's complement ~~arithmetic~~ Method.

Sol Normally

$14 \rightarrow 0000 1110$

1's complement $\rightarrow 1111 0001$

$$
\begin{array}{rcr}
25 & \longrightarrow & 0001 1001 \\
-14 & \longrightarrow & 1111 0001 \\
\hline
+11 & & \boxed{1}\ 0000 1010 \\
 & & 1 \\
\hline
 & & 0000 1011
\end{array}
$$

End around carry

$1 \rightarrow$ (adding of End around carry)

The MSB is a '0'. so the result is positive and is in pure binary. Therefore, the result is $0000 1011 = +(11)_{10}$

(2) Add −25 to +14 using 8-bit 1's complement method

+14 → 00001110
−25 → 11100110
−11   11110100 → No carry

25 → 00011001
1's complement → 11100110

⇒ There is no carry. The MSB is a '1'. So, the result is negative and is in its 1's complement form. Take 1's complement indicate −ve sign

⇒ The complement of 11110100 is −00001011. The result is $-(11)_{10}$

(3) Add −25 to −14 using 8-bit 1's complement method.

−25 → 11100110 (1's complement)
−14 → 11110001 (1's complement)
−39 $\boxed{1}$11010111.

25 → 00011001 (normal form)
14 → 00001011 (normal form)

End around carry

11010111
1 (adding end around carry)
11011000

MSB
00100111 → 1's complement

→ The MSB is '1' so the result is negative and we should find 1's complement of above answer. The 1's complement of 11011000 is 00100111 therefore the result is −39.

−ve sign.

(1) Add +25 to +14 using 8-bit 1's complement arithmetic.

$$
\begin{array}{lcr}
+25 & \rightarrow & 00011001 \\
+14 & \rightarrow & 00001110 \\
\hline
+39 & & 00100111 \\
\hline
\end{array}
$$

There is no carry. The MSB is '0'. So, the result is positive and is in pure binary. Therefore, the result is $+39_{10}$.

Ex:- (2)

1) Subtract 20 from 36 using 8-bit 1's complement form

$$
\begin{array}{lcrl}
36 & \rightarrow & 00100100 & \\
-20 & \rightarrow & 11101011 & (1^{st}\text{ complement}) \\
\hline
+16 & & \boxed{1}00001111 & \\
 & & 1 & (\text{adding of end around carry}) \\
\hline
 & & 00010000 & \\
\hline
\end{array}
$$

$36 \rightarrow 00100100$

$20 \rightarrow 00010100$

1 End around carry

MSB is '0'

⇒ The MSB is zero. The result is positive and it is in True binary form.

(2) Add +36 to +20 using 8-bit 1's complement form

$$
\begin{array}{lcr}
+36 & \rightarrow & 00100100 \\
+20 & \rightarrow & 00010100 \\
\hline
 & & 00111000 \\
\hline
\end{array}
$$

MSB is 0

32 16 8 4 2 1

$+36 \rightarrow 00100100$

$+20 \rightarrow 00010100$

MSB is zero. the result is positive and it is in true binary form

(3) Add −36 to −20 using 8-bit 1's complement form.

36 → 00100100
20 → 00010100

−36 → 11011011 → 1's complement
−20 → 11101011

−56 $\boxed{1}$ 11000110 → adding of end around carry

end around carry

11000111

MSB = 1

⇒ MSB is 1 the result is negative and it is in 1's complement form. To get the correct result take 1's complement to the result and put −ve sign before the result.

$11000111 \xrightarrow{\text{1's complement}} -00111000 = -(56)_{10}$

(4) Add −36 to +20 using 8-bit 1's complement form

−36 → 1's comple → 11011011
+20 → Normal → 00010100

−16 11101111

MSB = 1

36 → 00100100
20 → 00010100

→ the MSB is '1'. the result is negative and it is in 1's complement form.

→ Take 1's complement to the result and put −ve sign before the result

$11101111 \rightarrow -00010000 = (-16)_{10}$

→ **2's complement :-** In 2's complement subtraction, add 2's complement of the subtrahend to the minuend. If there is a carryout, ignore it. If the MSB is 'zero' the result is positive and is in true binary form. If MSB is '1' the result is negative and is in its 2's complement form.

Ex:-

① Subtract 14 from 25 using 8-bit 2's complement arithmetic

$$
\begin{array}{rl}
+14 = & 0000\,1110 \text{ (normal format)} \\
& 1111\,0001 \text{ (1's complement)} \\
& \qquad\quad 1 \text{ (2's complement)} \\
\hline
& 1111\,0010 \rightarrow \text{2's complement form}
\end{array}
$$

$$
\begin{array}{rl}
25 \rightarrow & 00011001 \\
-14 \rightarrow & 11110010 \\
\hline
& \boxed{1}\,00001011
\end{array}
$$

Ignore carry   MSB

→ There is a carry ignore it. The MSB is '0'. So, the result is positive and is in normal binary form. Therefore, the result is $+00001011 = +11_{10}$

② Add −25 to +14 using 8-bit 2's complement arithmetic

+25 → 00011001

−25 → 11100110 → (1's complement form)

$$\begin{array}{r} 11100110 \\ 1 \\ \hline 11100111 \end{array}$$ → 2's complement form

+14 → 00001110

−25 → 11100111

$$\begin{array}{r} 00001110 \\ 11100111 \\ \hline 11110101 \end{array}$$ (No carry)

msB = 1

There is no carry, the msB is '1'. So, the result is negative and is in 2's complement form. The magnitude is 2's complement of 11110101, that is 00001010

$$\begin{array}{r} 00001010 \\ 1 \\ \hline 00001011 \end{array} = (-11)_{10}$$

(13) Imp * Given the two binary numbers X = 1010100 and Y = 1000011, perform subtraction (a) X−Y (b) Y−X using 2's complements

X = 1010100   Y = 1000011 → subtrahend

0111100 → 1's complement

$$\begin{array}{r} 0111100 \\ 1 \\ \hline 0111101 \end{array}$$ → 2's complement

① Find 2's complement of subtrahend

② Add subtrahend to the minuend.

$X = 1010100$

$Y = 0111101 \rightarrow$ 2's complement of Y

$\boxed{1}\,0010001$

Discard carry

MSB = 0

The MSB = 0 the result is positive and it is in true binary form.

(b) Y−X

Y = 1000011 X = 1010100

0101011 → 1's complement

1

0101100 → 2's complement

Y = 1000011

0101100 → 2's complement of x

1101111

MSB = 1

There is no carry. And the MSB is '1' so the answer is 2's complement form. So find 2's complement of the result to get the correct answer

1101111

0010000 → 1's complement

1

0010001 → correct answer 2's complement form.

Imp Given the two binary numbers X = 1010100 and Y = 1000011, perform the subtraction (a) X−Y and (b) Y−X using 1's complement

(a) X = Y

X = 1010100

= 0111100 → 1's complement of Y

1111

$\boxed{1}$0010000

End around carry

1

0010001

MSB = 0 so, it is in true binary form

(b) Y − X

Y = 1000011

X = 0101011 → 1's complement of Y

11

1101110

MSB = 1

There is no carry. And the MSB is 1 so the result is in 1's complement form. So, find 1's complement of answer

1's complement of 1101110 is −0010001

⇒ 9's and 10's complement :-

→ In 9's complement subtraction just follow the below rules

① Find the 9's complement of subtrahend and Add 9's complement of subtrahend to minuend.

② If there is a carry it indicates that the answer is +ve then add carry to the LSD of the result to get true answer.

③ If there is no carry, it indicates that the answer is negative and the result obtained is its 9's complement

Q) Find the 9's complement of the following decimal numbers

① 3465

Sol
9999
3465
(6534)
↓
(9's complement of 3465)

By using formula

$(r^n - N) - 1$
$(10^4 - 3465) - 1 = 6534$

② 782.54

Sol
999.99
782.54
217.45

③ 4526.075

Sol
9999.999
4526.075
5473.924

9's complement Method of Subtraction :-

Q Subtract the following numbers using 9's complement method .

① 745.81 — 436.62

Step ①

$$\begin{array}{r} 999.99 \\ 436.62 \\ \hline 563.37 \\ \hline \end{array}$$

→ 9's complement of 436.62

Step ②

$$\begin{array}{r} 745.81 \\ 563.37 \\ \hline \boxed{1}\,309.18 \\ 1 \\ \hline 309.19 \\ \hline \end{array}$$

(adding end around carry)

→ final answer.

carry indicates the answer is +ve

② 436.62 — 745.81

Step ①

$$\begin{array}{r} 999.99 \\ 745.81 \\ \hline 254.18 \\ \hline \end{array}$$

→ 9's complement of 745.81

Step ②

$$\begin{array}{r} 436.62 \\ 254.18 \\ \hline 690.80 \end{array}$$

there is no carry, so it indicating that the answer is negative

so, take 9's complement of the intermediate result and put a minus sign before it.

$$\begin{array}{r} 999.99 \\ 690.80 \\ \hline -309.19 \\ \hline \end{array}$$

$\rightarrow$ therefore the answer is $\underline{-309.18}$.

⇒ 10's complement method of subtraction

The 10's complement of a decimal number is obtained by adding a '1' to its 9's complement.

Q. Find the 10's complement of the following decimal number.

① 3465 ② 782.54 ③ 4526.075

Sol

$$\begin{array}{r} 9999 \\ 3465 \\ \hline 6534 \\ 1 \\ \hline 6535 \\ \hline \end{array}$$

6534 → 9's complement

6535 → 10's complement

Sol

$$\begin{array}{r} 999.99 \\ 782.54 \\ \hline 217.45 \\ 1 \\ \hline 217.46 \\ \hline \end{array}$$

217.45 → 9's complement

217.46 → 10's complement

$$\begin{array}{r} 9999.999 \\ 4526.075 \\ \hline 5473.924 \\ 1 \\ \hline 5473.925 \\ \hline \end{array}$$

10's complement method of subtraction :-

1. To perform decimal subtraction using 10's complement method, obtained the 10's complement of the subtrahend and add it to the minuend.

2. If there is a carry, ignore it. the presence of the carry indicates that the answer is positive.

3. If there is no carry, it indicates the answer is negative and the result obtained in its 10's complement form and put negative sign infront of the answer.

① 745.81 − 436.62

step ①

$$\begin{array}{r} 999.99 \\ 436.62 \\ \hline 563.37 \\ 1 \\ \hline 563.38 \end{array}$$

→ 10's complement form

step ②

$$\begin{array}{r} 745.81 \\ 563.38 \\ \hline \boxed{1}309.19 \end{array}$$

Ignore the carry

carry indicates result is positive

② 436.62 − 745.81

step ①

$$\begin{array}{r} 999.99 \\ 745.81 \\ \hline 254.18 \\ 1 \\ \hline 254.19 \end{array}$$

step ②

$$\begin{array}{r} 436.62 \\ 254.19 \\ \hline 690.81 \end{array}$$

→ no carry answer is −ve

step ③

$$\begin{array}{r} 999.99 \\ 690.81 \\ \hline 309.18 \end{array}$$

−309.19

15's complement Method :-

Q Find the 15's complement of the following numbers

(a) 6A36

Sol

$$\begin{array}{r} FFFF \\ (-)\ 6A36 \\ \hline 95C9 \\ \hline \end{array} \quad (or) \quad \begin{array}{cccc} 15 & 15 & 15 & 15 \\ 6 & A & 3 & 6 \\ \hline 9 & 5 & C & 9 \\ \hline \end{array}$$

(b) 9AD.3A

$$\begin{array}{rcccccc} & 15 & 15 & 15 & . & 15 & 15 \\ (-) & 9 & A & D & . & 3 & A \\ \hline & 6 & 5 & 2 & . & C & 5 \\ \hline \end{array} \rightarrow \text{(15's complement)}$$

⟹ 15's complement method of subtraction

(a) 69B − C14

Step(1) 15's complement of (−C14)

$$\begin{array}{rccc} & 15 & 15 & 15 \\ (-) & C & 1 & 4 \\ \hline & 3 & E & B \\ \hline \end{array} \rightarrow \text{(15's complement of (−C14))}$$

Step(2) 69B − C14 = 69B + (15's complement of (−C14))

$$\begin{array}{rccc} & 6 & 9 & B \\ (+) & 3 & E & B \\ \hline & A & 8 & 6 \\ \hline \end{array} \qquad \begin{array}{l} (22)_{10} = (16)_H \\ (24)_{10} = (18)_H \end{array}$$

There is no carry, result is (−) ve

Step ③ 15's complement of intermediate result is given by

$$\begin{array}{ccc} 15 & 15 & 15 \\ A & 8 & 6 \\ \hline 5 & 7 & 9 \\ \hline \end{array}$$

∴ Final result is $-(579)_{16}$

ⓑ $-69B + C14$ (or) $C14 - 69B$

Step ① 15's complement of (−69B)

$$\begin{array}{cccc} & 15 & 15 & 15 \\ - & 6 & 9 & B \\ \hline & 9 & 6 & 4 \\ \hline \end{array} \rightarrow \text{15's complement}$$

Step ②

$C14 - 69B = C14 + (\text{15's complement of } (-69B))$

$$\begin{array}{cccc} & C & 1 & 4 \\ (+) & 9 & 6 & 4 \\ \hline 1 & 5 & 7 & 8 \\ \hline \end{array}$$ $(21)_{10} = (15)_{16}$

carry

There is a carry, so the result is +ve

$$\begin{array}{r} 578 \\ 1 \text{ (End around carry)} \\ \hline 579 \end{array}$$

∴ Final result $= +(579)_{16}$

16's complement method :-

First find the 15's complement and then add 1

Q Find the 16's complement of the following number.

(a) A8C

15's complement is given by

$$\begin{array}{rrr} 15 & 15 & 15 \\ (-)\ A & 8 & C \\ \hline 5 & 7 & 3 \end{array} \rightarrow \text{15's complement}$$

$$\text{16's complement} \rightarrow \begin{array}{r} 573 \\ (+)\ 1 \\ \hline 574 \end{array}$$

⇒ 16's complement method of subtraction :-

Find the 16's complement subtraction of the following numbers :

(a) C9B − C14

Step (1) 15's complement of (−C14)

$$\begin{array}{rrr} 15 & 15 & 15 \\ C & 1 & 4 \\ \hline 3 & E & B \end{array} \rightarrow \text{15's complement}$$

$$\text{16's complement} \rightarrow \begin{array}{r} 3EB \\ 1 \\ \hline 3EC \end{array}$$

Step ②:-

$C9B - C14 = C9B + (16\text{'s complement of } (-C14))$

$$\begin{array}{r} C\,9\,B \\ +\;3\,E\,C \\ \hline ①\,0\,8\,7 \end{array}$$

carry

$(23)_{10} = (17)_{16}$

$(24)_{10} = (18)_{16}$

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

there is a carry ignore it. Since the carry is 1. the result is (+)ve.

∴ Final result is $+(087)_{16}$.

(B) $2A4.2D - 3B2.3C$

step ①:- 15's complement of $(-3B2.3C)$

$$\begin{array}{r} 15\;15\;15.\;15\;15 \\ -\;3\;\;B\;\;2\;.\;\;3\;\;C \\ \hline C\;\;4\;\;D\;.\;\;C\;\;3 \end{array}$$

→ 15's complement

16's complement is given by,

$$\begin{array}{r} C\,4\,D\,.\,C\,3 \\ 1 \\ \hline C\,4\,D\,.\,C\,4 \end{array}$$

→ 16's complement

step②.

$2A4.2D - 3B2.3C = 2A4.2D + (16\text{'s complement of } (-3B2.3C))$

$$\begin{array}{r} 2A4.2D \\ C4D.C4 \\ \hline EF1.F1 \end{array}$$

→ intermediate result

there is no carry. So the result is (−)ve

Step③ 15's complement of intermediate result is given by

15 15 15 . 15 15
E F 1 . F 1

1 0 E . 0 E → 15's complement
(+) 1

1 0 E . 0 F → 16's complement

∴ Final result is $-(0E.0F)_{16}$

⇒ 7 and 8's complements:

Subtract the following numbers using 7's complement method.

ⓐ 234.65 − 135.74

Sol 7's complement of (−135.74) is given by

777.77
135.74

642.03 → 7's complement

∴ 234.65 + (7's complement of (−135.74))

234.65
642.03

① 076.70 (Carry ⇒ result is positive)

carry

$(8)_{10} = (10)_8$

$$\begin{array}{r} 076.70 \\ (+)\ 1 \\ \hline 76.71 \\ \hline \end{array}$$ (End around carry)

$\therefore$ Result is $+76.71$

(B) $135.74 - 236.65$

7's complement of (-236.65) is given by,

$$\begin{array}{r} 777.77 \\ 236.65 \\ \hline 541.12 \\ \hline \end{array} \rightarrow \text{7's complement}$$

$\therefore 135.74 - 236.65 = 135.74 + (\text{7's complement of } (-236.65))$

$$\Rightarrow \begin{array}{r} 135.74 \\ 541.12 \\ \hline 677.06 \\ \hline \end{array}$$

$541.12 \rightarrow$ (7's complement)

$677.06 \rightarrow$ Intermediate result

There is no carry. Hence the final result is $(-)$ve.

Final result is the 7's complement of the above intermediate result

$$\begin{array}{r} 777.77 \\ (-)\ 677.06 \\ \hline 100.71 \\ \hline \end{array}$$

$\therefore$ Final result is -100.71

Subtract the following using 8's complement method :

ⓐ 246.31 − 162.45

7's complement of (−162.45) is given by,

$$\begin{array}{r} 777.77 \\ 162.45 \\ \hline 615.32 \end{array} \rightarrow \text{(7's complement)}$$

$$\begin{array}{r} (+)\ 1 \\ \hline 615.33 \end{array} \rightarrow \text{(8's complement)}$$

∴ 246.31 − 162.45 = 246.31 + (8's complement of (−162.45))

$$\Rightarrow \begin{array}{r} 246.31 \\ 615.33 \\ \hline 1\ 063.64 \end{array}$$

carry

There is a carry. Hence the result is (+)ve and ignore the carry.

∴ Final result is +63.64

ⓑ 162.45 − 246.31

8's complement of (−246.31) is given by,

$$\begin{array}{r} 777.77 \\ (-)\ 246.31 \\ \hline 531.46 \end{array} \rightarrow \text{(7's complement)}$$

$$\begin{array}{r} (+)\ 1 \\ \hline 531.47 \end{array} \rightarrow \text{(8's complement)}$$

$\therefore$ $162.45 - 246.31 = 162.45 +$ (8's complement of (-246.31))

$$\Rightarrow \begin{array}{r} 162.45 \\ 531.47 \\ \hline 714.14 \\ \hline \end{array} \rightarrow \text{(Intermediate result)}$$

there is no carry. Hence the final result is (-)ve. Final result is the 8's complement of the above intermediate result

$$\begin{array}{r} 777.77 \\ 714.14 \\ \hline 063.63 \\ (+)\ 1 \\ \hline 063.64 \\ \hline \end{array}$$

$063.63 \rightarrow$ 7's complement

$063.64 \rightarrow$ 8's complement

$\therefore$ Final result is $-\underline{63.64}$

⇒ Floating point Representation :-

The goal of floating point representation is represent a large range of numbers.

Ex :- Given the number -123.154×10^5

Sign = − (Negative)

Mantissa = 123.154

Exponent = 5

Base = 10 (decimal)

Eg :-

① Distance b/w two Planet = 5.9×10^{12} m

② mass of electron = 9.1×10^{-28} gm.

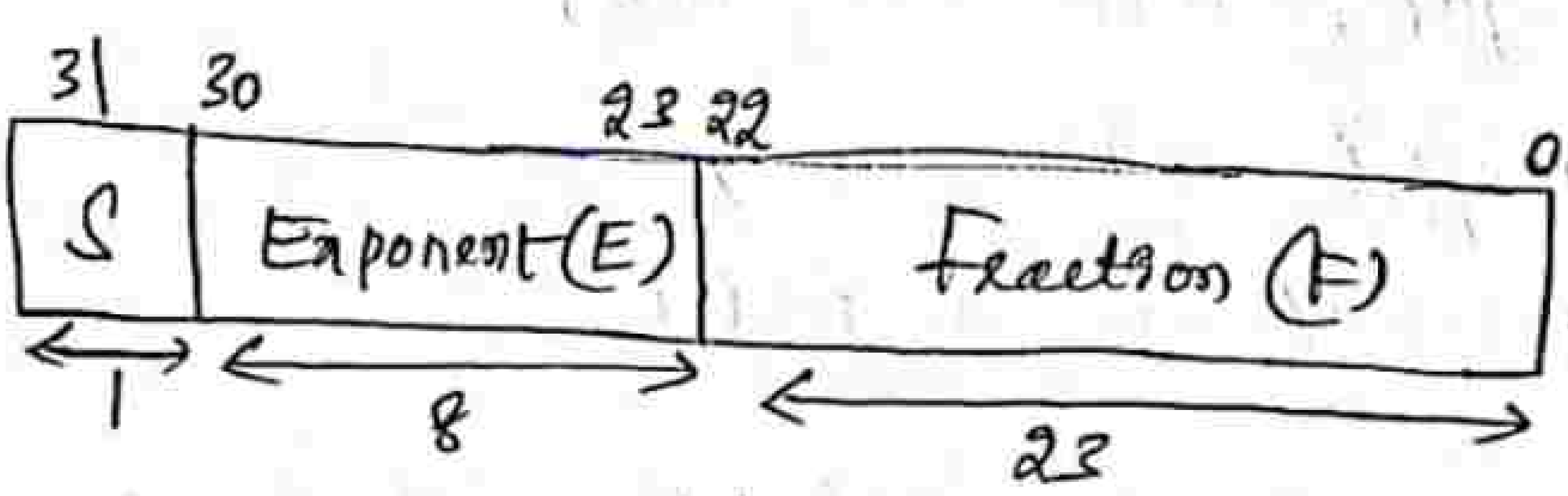
Ex :- Given the number 732.136×10^7

Sign = + (positive)

mantissa = 732.136

Exponent = 7

Base = 10 (Decimal)



32-bit single-precision Floating point Number

Ex :-

$4.2 * 10^8$ → Exponent

4.2 → Mantissa, 10 → Base

∴ Only the mantissa and exponent are stored. The base is implied (Already Known). It will save the memory.

Ex:-

$(11.8)_{10} = (1011.11001\ldots)_2$

$= (1.01111001)_2 * 2^3$

$= (1.01111001)_2 * 2^{(11)_2}$

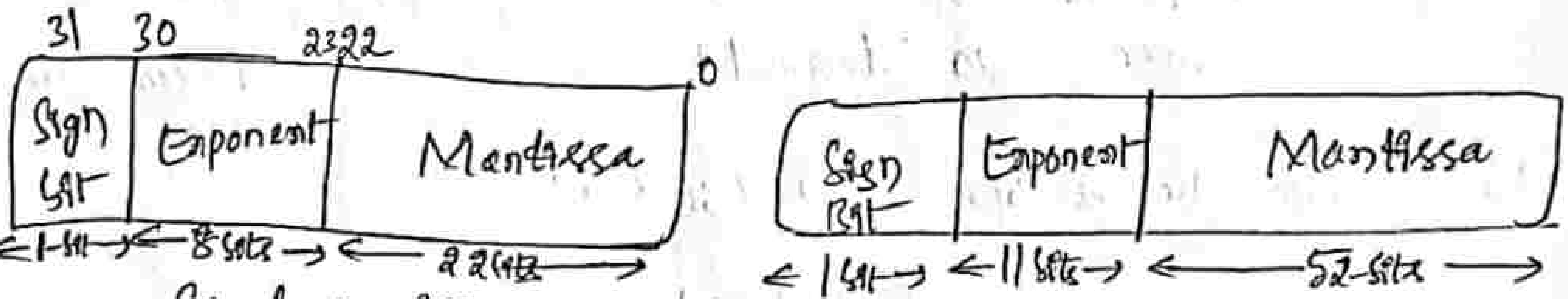
<u>Decimal representation</u>

$12345 = 1.2345 \times 10^4$ → Exponent

1.2345 ↓ mantissa (or) Significant

10 ↓ Base

⇒ We will represent floating point numbers in single precision and double precision formats. They are shown below



<u>Single precision</u>
<u>32-bits</u>

<u>Double precision</u>
<u>64-bits</u>

(IEEE 754 standard)

* 1 bit for the sign (positive (or) Negative)
8 bit for the range (exponent field)
23 bit for the precision (fraction field)

$$\begin{cases} N = (-1)^s \times 1.\text{fraction} \times 2^{\text{exponent}-127} & 1 \le \text{exponent} \le 254 \\ N = (-1)^s \times 0.\text{fraction} \times 2^{\text{exponent}-126}, & \text{exponent} = 0. \end{cases}$$

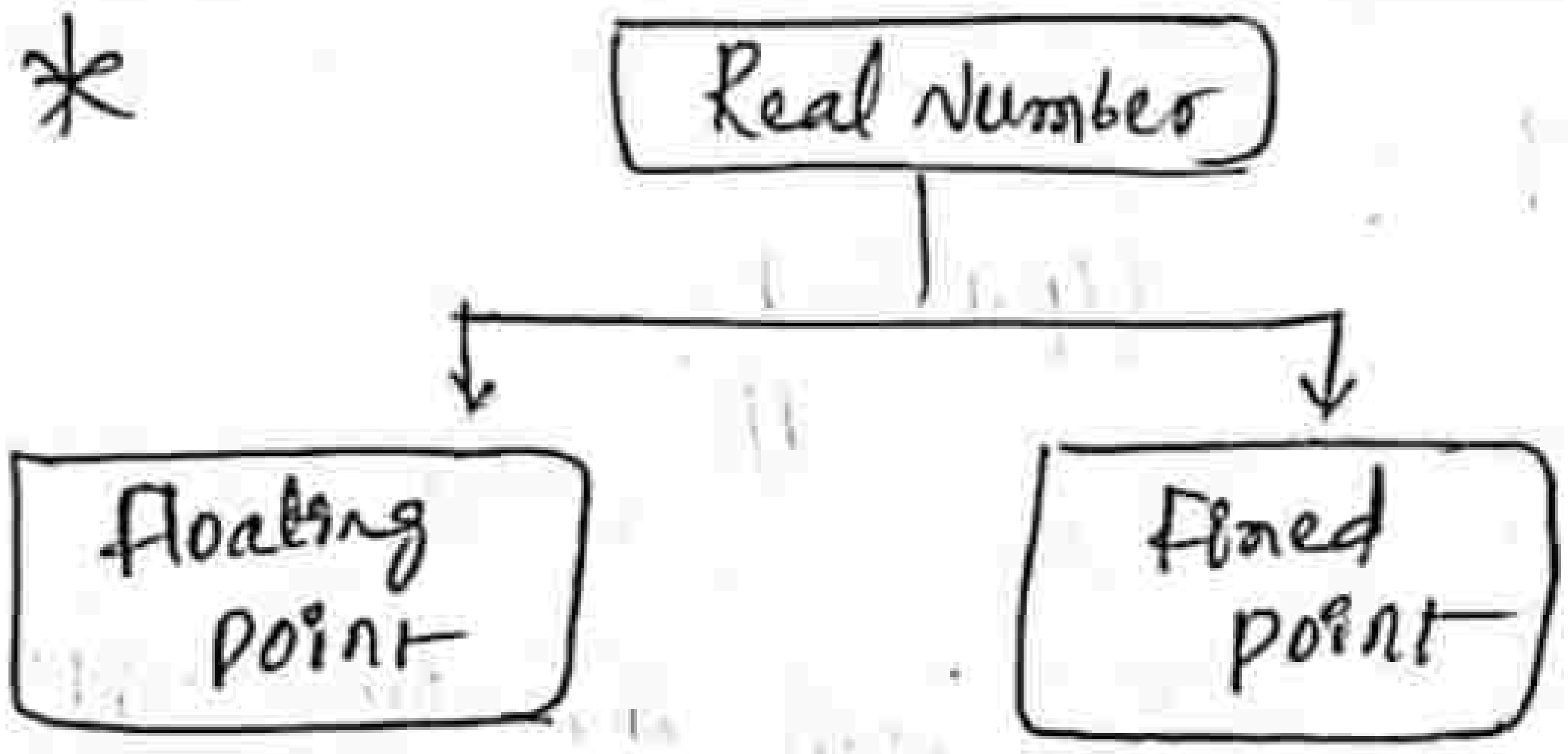
* $\text{Value} = (-1)^s \times (1+F) \times 2^{E-127}$ (Or)

$$\boxed{X = (-1)^s \times 2^{E-1024} \times 1.M}$$ ↑ double precision

$$\boxed{N = (-1)^s \times 2^{E-127} \times 1.M}$$ ↑ single precision

Fixed point Representation	Floating point Representation
① A representation of real data type for a number that has a fixed number of digits after the radix point.	① A formulaic representation of real numbers as an approximation so as to support a trade off between range and precision.
② Used to represent a limited range of values.	② Used to represent a wide range of values.
③ High performance	③ High performance
④ Less flexible	④ More flexible

*



* Limitation of floating points

✓ Size of mantissa is fixed.

Sol Use a floating point format with a larger mantissa.
double (53 bits) long double (64 bits)

problems:-

Q what is the 111.11 in decimal
ⓐ 7.75
ⓑ 21
ⓒ 7.375
ⓓ 15.25

Q what is 8.5 in binary
ⓐ 11111111.11111
ⓑ 1000.01
ⓒ 0.100011
ⓓ 1000.10.

Ex: −114.625 represent in binary

Sol

128	64	32	16	8	4	2	1	.	0.5	0.25	0.125
0	1	1	1	0	0	1	0	.	1	0	1

$64+32+16+2 = \underline{114}$ ↑ $0.5 + \cancel{0.25} + 0.125$

$= 01110010.101$

$= 1.1100101 01 \times 2^6$

$127 + 6 = \underline{\underline{133}}$

133 in binary

10000101

$\therefore\ 1\ \underbrace{10000101}\ \underbrace{1100101 01.}$

Sign bit

Ex:

Sign	Exponent	Mantissa
0	1001010	11101000

$\underbrace{0100}_{4}\ \underbrace{1010}_{10 \to A}\ \underbrace{1110}_{14 \to E}\ \underbrace{1000}_{8} = (4AE8)_H$

Ex: $0.000110011001100110011001100_2$ represent in floating point in 32-bits.

Sol $1.10011001100110011001100 \times 2^{-4}$

128	64	32	16	8	4	2	1
0	1	1	1	1	0	1	1

$= \underline{\underline{132}}$

Exponent $= -4 + 127 = 123$

Sign bit $= 0$

Mantissa $= 10011001100110011001100$

S (1 bit)	Exponent 8 bits	mantissa (23 bits)
0	01111011	10011001100110011001100.

Fixed point Representation :-

⇒ Representation of signed binary numbers :-

Positive numbers can be represented by unsigned numbers however to represent negative numbers, we need notation for negative numbers.

There are two types of numbers.

(1) Unsigned numbers
(2) Signed numbers

(1) Unsigned numbers :- There is no specific for sign representation. The numbers without positive (or) negative signs are known as unsigned numbers. The unsigned numbers are always positive numbers.

(2) Signed numbers :- There is a specific bit for sign representation. In signed numbers, the number may be positive (or) negative. Different formats are used for representation signed binary numbers. They are

1. Signed magnitude representation
2. 1's complement representation
3. 2's complement representation

1. Sign magnitude representation :-

In signed magnitude form, an additional bit called the 'sign bit' is placed infront of the number. If the sign bit is a '0', the number is positive. If it is a '1', the number is negative.

→ For example :-

$\boxed{0}\underbrace{101001}_{\text{magnitude}} = +41$

Sign bit

$\boxed{1}\underbrace{101001}_{\text{magnitude}} = -41$

Sign bit

In sign magnitude representation the 'MSB' represents the 'sign' and remaining bits represents the magnitude.

2. 1's complement representation :-

In 1's complement representation the positive number remain unchanged. 1's complement representation of negative number can be obtained by the 1's complement of the binary number.

$0\,110011 = +51$; MSB = 0 for +ve

Sign bit

$1\,001100 = -51$; MSB = 1 for −ve

Sign bit

② 2's complement representation :-

In 2's complement representation, the positive numbers remain unchanged, 2's complement representation of negative numbers can be obtained by

1. Find the 1's complement of the number
2. To find 2's complement of the number adding '1' to its 1's complement number.

$\underset{\text{Sign bit}}{0}\,\underbrace{110011}_{\text{magnitude}} = +51$

$\underset{\text{Sign bit}}{1}\,\underbrace{001101}_{\text{magnitude}} = -51$ [In sign 2's complement form]

Number system	+9	−9
Unsigned	+1001	−1001
Sign magnitude	Sign bit 0 1001	Sign bit 1 1001
Sign 1's complement form	0 1001	1 0110
Sign 2's complement form	0 1001	1 0111

Decimal	Sign magnitude form	Sign 1's complement form	Sign 2's complement form
+7	0 111	0 111	0 111
+6	0 110	0 110	0 110
+5	0 101	0 101	0 101
+4	0 100	0 100	0 100
+3	0 011	0 011	0 011
+2	0 010	0 010	0 010
+1	0 001	0 001	0 001
+0	0 000	0 000	0 000
−0	1 000	1 111	—
−1	1 001	1 110	1 111
−2	1 010	1 101	1 110
−3	1 011	1 100	1 101
−4	1 100	1 011	1 100
−5	1 101	1 010	1 011
−6	1 110	1 001	1 010
−7	1 111	1 000	1 001

Q Represent +51 and −51 in sign magnitude, sign 1's complement and sign 2's complement representation.

Sol

	+51	−51
Sign magnitude	0 110011 (sign bit)	1 110011 (sign bit)
Sign 1's complement	0 110011 (sign bit)	1 001100 (sign bit)
Sign 2's complement	0 110011 (sign bit)	1 001101 (sign bit)

Q Represent +43 and −43 in sign magnitude, sign 1's complement and 2's complement representation

Sol

	+43	−43
Sign magnitude	0 101011 (sign bit)	1 101011 (sign bit)
Sign 1's complement	0 101011 (sign bit)	1 010100 (sign bit)
Sign 2's complement	0 101011 (sign bit)	1 010101 (sign bit)

⇒ Combinational Circuits :-

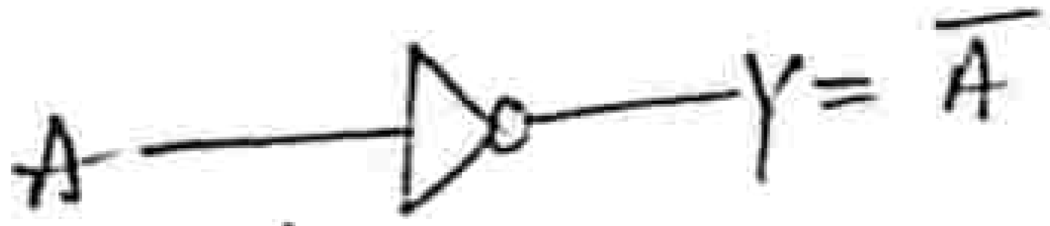
⇒ **Boolean Expressions :-** Boolean Algebra is a division of mathematics which deals with operations on logic values and incorporates binary variable. Boolean algebra was invented by great mathematician George Boole in 1854.

→ Minimization of logic expressions can be done by using boolean theorems and laws.

→ Boolean algebra, Karnaugh map (K-map) are used for boolean minimization.

→ The main moto of this concept is to make information simpler, cheaper and low cost.

⇒ **Logic Gates :-** ① Not gate (or) Inverter :-



Symbol

Truth table

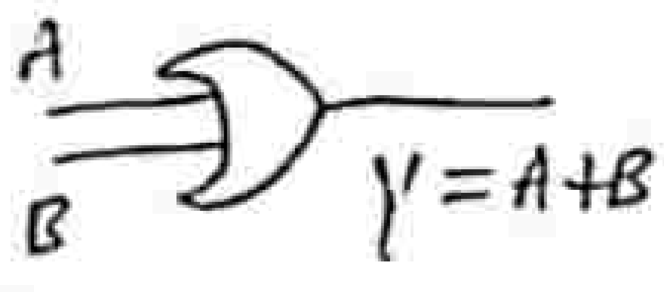
A	$Y=\overline{A}$
0	1
1	0

② AND Gate :-

A	B	Y = AB
0	0	0
0	1	0
1	0	0
1	1	1

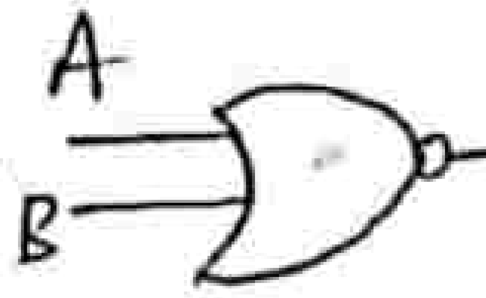
∴ these 3 gates are primary gates

③ OR Gate :-



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

⇒ **NOR Gate :-**



$Y = \overline{A+B}$

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

⇒ **NAND Gate :-**

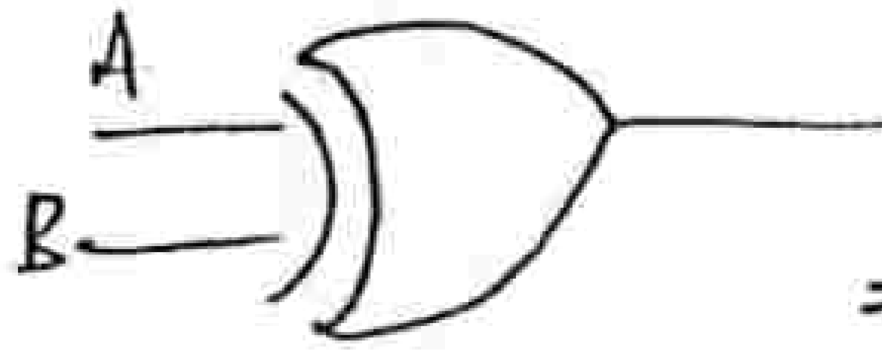
$Y = \overline{AB}$

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

∴ The above two gates are universal gates

Special Gates :-

⇒ **EX-OR gate :-**

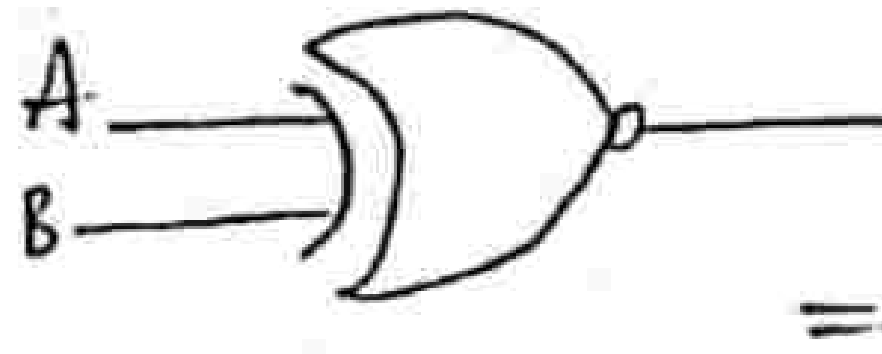


$Y = A \oplus B$

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

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

⇒ **EX-NOR gate :-**



$Y = A \odot B$

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

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

⇒ **Laws of Boolean Algebra :-**

AND operation	OR operation	NOT operation
1) $0 \cdot 0 = 0$	5) $0+0=0$	9) $\overline{0} = 1$
2) $0 \cdot 1 = 0$	6) $0+1=1$	10) $\overline{1} = 0$
3) $1 \cdot 0 = 0$	7) $1+0=1$	
4) $1 \cdot 1 = 1$	8) $1+1=1$	

⟹ Complement Law

$\overline{0} = 1$

$\overline{1} = 0$

If $A = 0$ then $\overline{A} = 1$

If $A = 1$ then $\overline{A} = 0$

$\overline{\overline{A}} = A$

AND Law

$A \cdot 0 = 0$

$A \cdot 1 = A$

$A \cdot A = A$

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

proof

$= A \cdot A$

$= A \cdot A + 0$

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

$= A(A + \overline{A}) = A(1) = \underline{A}$

⟹ OR Law :-

$A + 0 = A$

$A + 1 = 1$

$A + \overline{A} = 1$

$A + A = A$

proof

$A + A = A$

$(A + A)\,1$

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

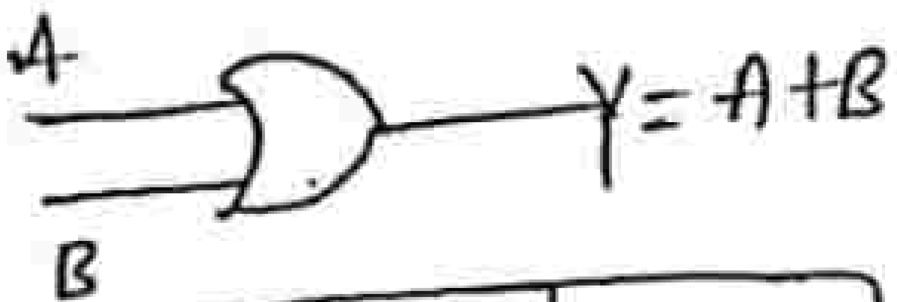
$A + A \cdot A + A \cdot \overline{A}$

$= A + A \cdot A + 0$

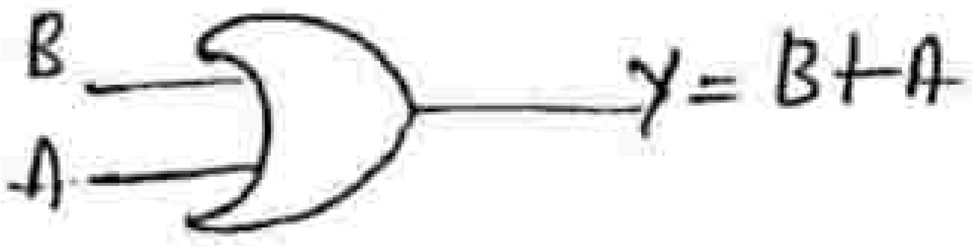
$= A(1 + A) = A \quad (1 + A = 1)$

⟹ Commutative Law :-

(1) $A + B = B + A$

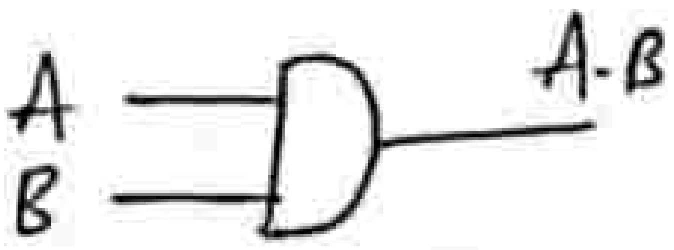


A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1



B	A	B+A
0	0	0
0	1	1
1	0	1
1	1	1

Law ②:- $A \cdot B = B \cdot A$

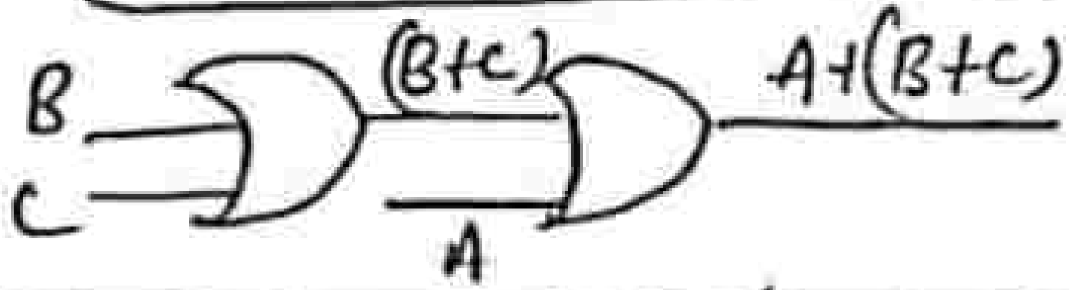


A	B	A·B
0	0	0
0	1	0
1	0	0
1	1	1

B	A	B·A
0	0	0
0	1	0
1	0	0
1	1	1

⇒ Associative Law :-

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

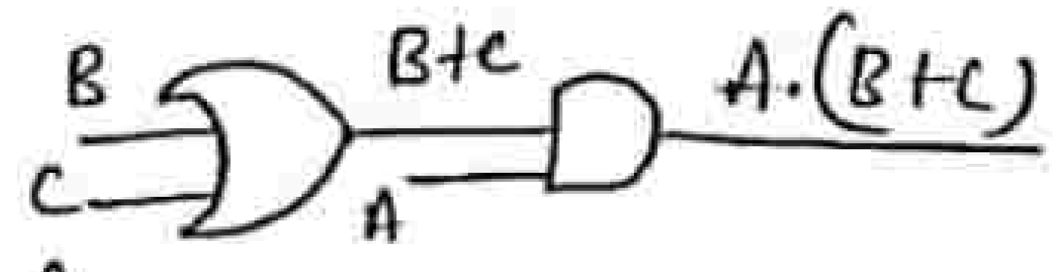


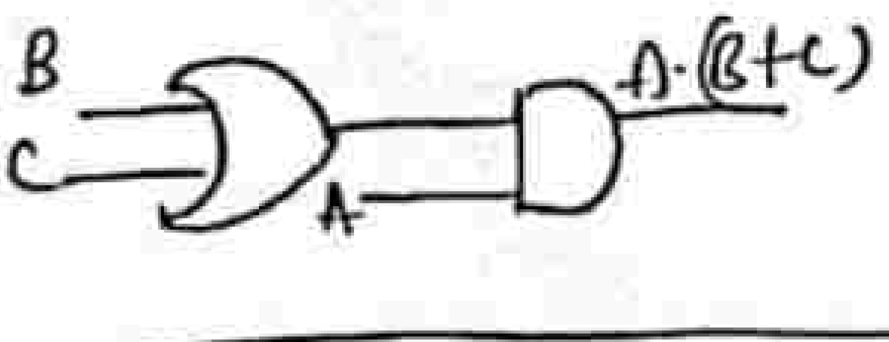
A	B	C	(B+C)	A+(B+C)
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	0	0	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

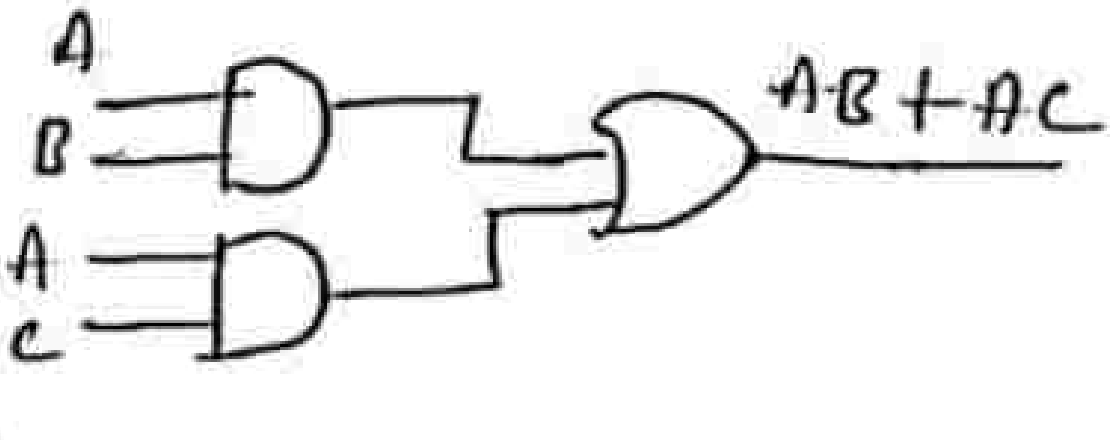
A	B	C	(A+B)	(A+B)+C
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

⇒ Distributive Law :-

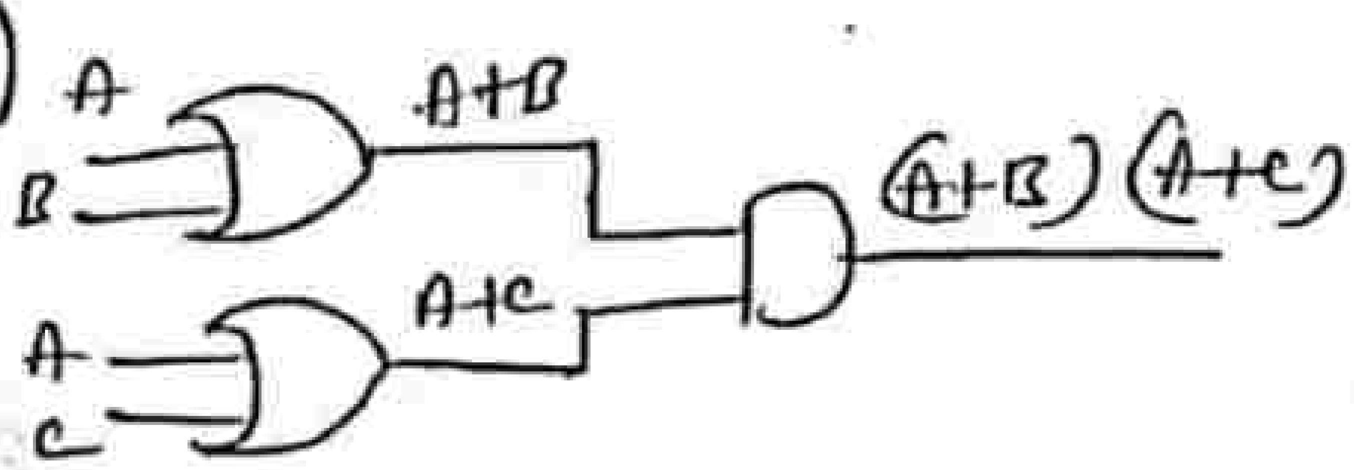
① $\boxed{A \cdot (B+C) = AB + AC}$

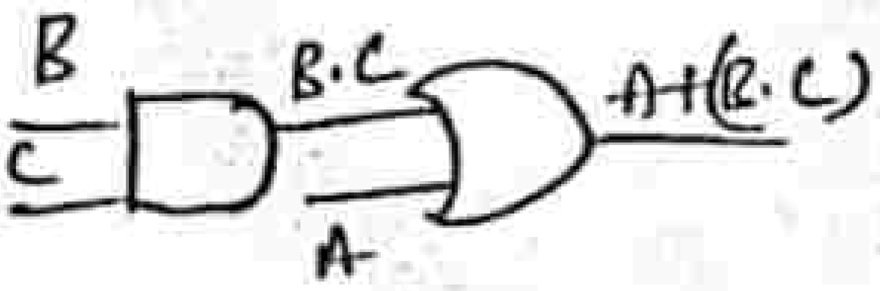






② $\boxed{A+(B \cdot C) = (A+B)(A+C)}$





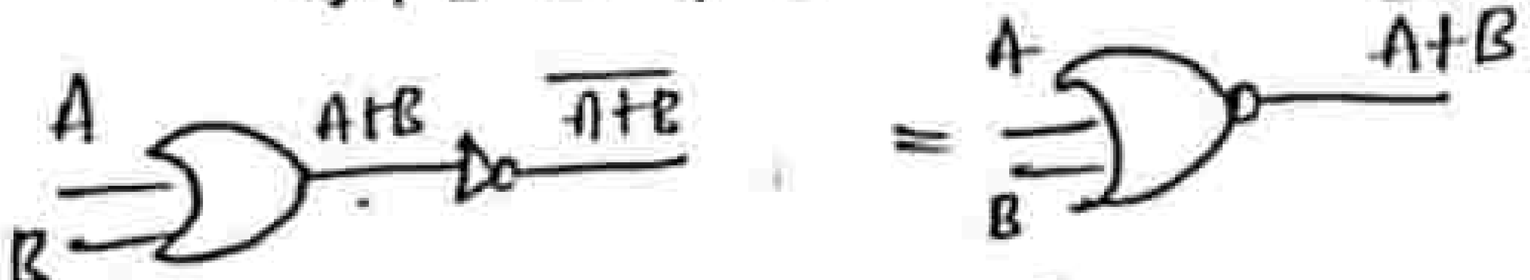
⟹ Consensus Theorem :-

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

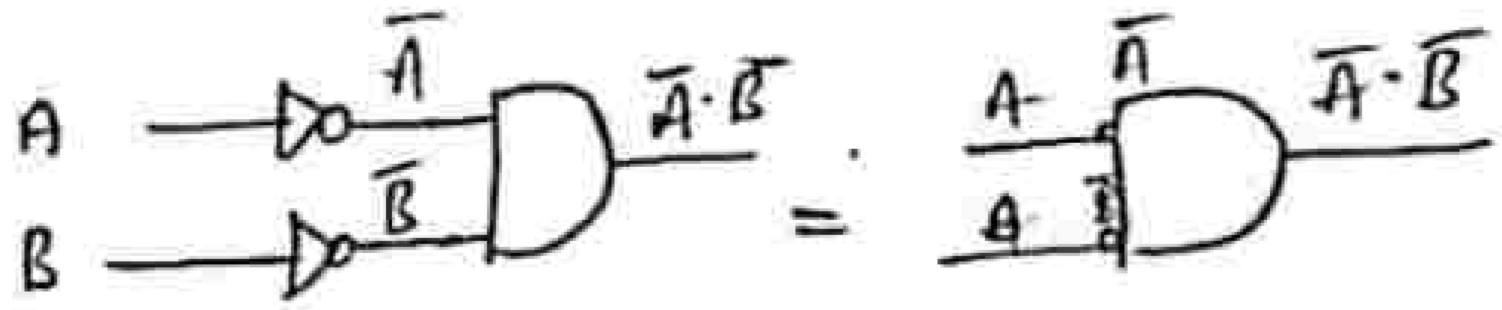
$$
\begin{aligned}
L.H.S &= A\cdot B + \bar{A}C + BC \\
&= AB + \bar{A}C + BC(A+\bar{A}) \qquad \therefore A+\bar{A}=1 \\
&= AB + \bar{A}C + BCA + BC\bar{A} \qquad 1+C=1 \\
&= AB(1+C) + \bar{A}C(1+B) \qquad 1+B=1 \\
&= AB(1) + \bar{A}C(1) \\
&= AB + \bar{A}C \\
&= R.H.S \qquad \therefore \boxed{L.H.S = R.H.S}
\end{aligned}
$$

⟹ Demorgan's Theorem :-

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

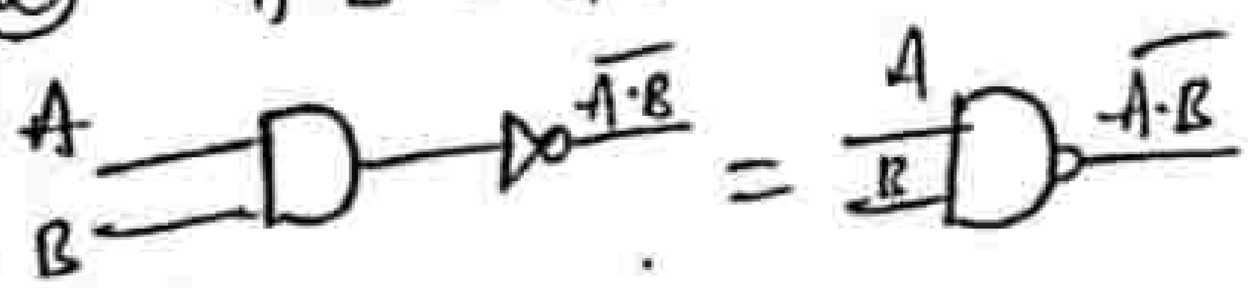


A	B	A+B	$\overline{A+B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

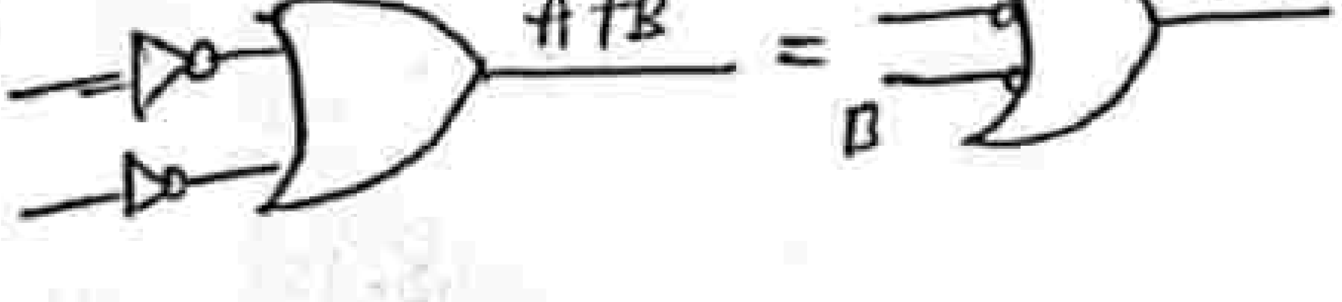


A	B	$\bar{A}$	$\bar{B}$	$\bar{A}\cdot\bar{B}$
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0.
1	1	0	0	0

② $\overline{A\cdot B} = \bar{A}+\bar{B}$



A	B	A·B	$\overline{A\cdot B}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0



A	B	$\bar{A}$	$\bar{B}$	$\bar{A}+\bar{B}$
0	0	1	1	1
0	1	1	0	1
1	0	0	1	1
1	1	0	0	0

⇒ **Duality** :– when changing from one logic system to another system ; 0 becomes 1 and 1 becomes 0. An AND gate becomes OR gate and OR gate becomes an AND gate.

⇒ **complement** :– If a boolean Identity is given, We should change '+' sign '·' sign and '·' sign to '+' sign. Variables $(\bar{A}, B)$ also complemented.

Note :– ① $\underbrace{AB+AB+AB+AB}_{\text{parenthesis}} = AB$ (Same no of variables are repeated we may considered single term)

② $A\cdot B\cdot \bar{B} = A\cdot 0 = 0$; ③ $ABC\bar{C} = AB\cdot 0 = 0$

③ $AB\bar{C}+AB\bar{C}+AB\bar{C}D$

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

④ $AB\bar{C}D + AB\bar{C}\bar{D}$

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

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

① $f = A+B\left[AC+(B+\bar{C})D\right]$

$= A+B\left[AC+BD+\bar{C}D\right]$

$= A+\left[ABC+\underbrace{BBD}+B\bar{C}D\right]$ | $\because BBD = BD$ repeated terms ; we can consider as single.

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

$(\because 1+BC = 1 ;\ 1+\bar{C} = 1)$

$\boxed{f = A+BD}$

② $f = \left(\overline{A+\bar{B}C}\right)\cdot\left(A\bar{B}+ABC\right)$

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

$= \bar{A}\ BC\ (A\bar{B}+ABC)$ | $\therefore A\bar{A}=0$; $B\bar{B}=0$

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

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

$= 0\ 0 + 0$ $\boxed{f=0}$ Hence the boolean expression has been reduced by boolean theorems.

Q Write the duality for the following functions

① $\bar{A}B + \bar{A}B\bar{C} + \bar{A}BCD + \bar{A}B\bar{C}\bar{D}E$

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

② $\bar{x}yz + x\bar{y}\bar{z} + xyz + xy\bar{z}$

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

Q Find the complements of the following expressions

① $AB + A(B+C) + \bar{B}(B+D)$

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

② $\bar{B}\bar{C}D + (\overline{B+C+D}) + \bar{B}\bar{C}\bar{D}E$

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

⇒ Karnaugh Map (K-map) Representation :-

① Sum of product (SOP) Ex:- $\bar{A}B + A\bar{B}$

② product of sum (POS) Ex:- $(A+B)(\bar{A}+C)$

① Sum of product (SOP) :- this is also called as disjunctive Normal form (DNF). variables present in this variables are called 'minterms' ($m_0, m_1, m_2 \dots$)

Ex:- $f(A,B,C) = m_1 + m_2 + m_3 + m_5 = \Sigma m(1,2,3,5)$

⇒ Standard SOP form :- (SSOP) · It is also called as Disjunctive Canonical form (DCF)

② product of sum (POS) :- this is also called as conjunctive Normal form (CNF). variables present in this form is called Maxterm ($M_1, M_2, M_3 \dots$)

Ex:- $F(A,B,C) = \Pi(M_1, M_2, M_6, M_7)$
$= \Pi M(1, 2, 6, 7)$

⟹ <u>Standard POS form</u> :- This form is also called as conjunctive Canonical form (CCF)

Ex:- $f(A,B,C) = (\bar{A}+\bar{B})(A+B)$

$= (\bar{A}+\bar{B}+C\cdot\bar{C})(A+B+C\cdot\bar{C})$ $\quad | \because C\cdot\bar{C}=0.$

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

Ex:- Convert <u>SOP to standard SOP form</u>

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

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

$= ABC + A\bar{B}C + ABC + AB\bar{C} + ABC + \bar{A}BC$ Repeated ABC product is there. we should write only one

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

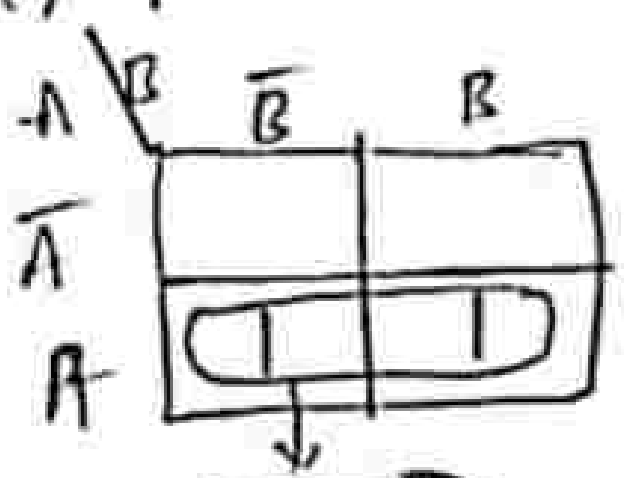
Decimal no.	A	B	C	minterms	Maxterms
0	0	0	0	$\bar{A}\bar{B}\bar{C}$ (m_0)	$A+B+C$ (M_0)
1	0	0	1	$\bar{A}\bar{B}C$ (m_1)	$A+B+\bar{C}$ (M_1)
2	0	1	0	$\bar{A}B\bar{C}$ (m_2)	$A+\bar{B}+C$ (M_2)
3	0	1	1	$\bar{A}BC$ (m_3)	$A+\bar{B}+\bar{C}$ (M_3)
4	1	0	0	$A\bar{B}\bar{C}$ (m_4)	$\bar{A}+B+C$ (M_4)
5	1	0	1	$A\bar{B}C$ (m_5)	$\bar{A}+B+\bar{C}$ (M_5)
6	1	1	0	$AB\bar{C}$ (m_6)	$\bar{A}+\bar{B}+C$ (M_6)
7	1	1	1	ABC (m_7)	$\bar{A}+\bar{B}+\bar{C}$ (M_7)

Rules for K-map minimization:-

1. Either group zeros & Ones
2. Diagonal mapping is not allowed
3. Only powers of 2, no. of cells in each group (i.e 2, 4, 6, 8 ....)
4. Group should be as large as possible
5. Overlapping is allowed.

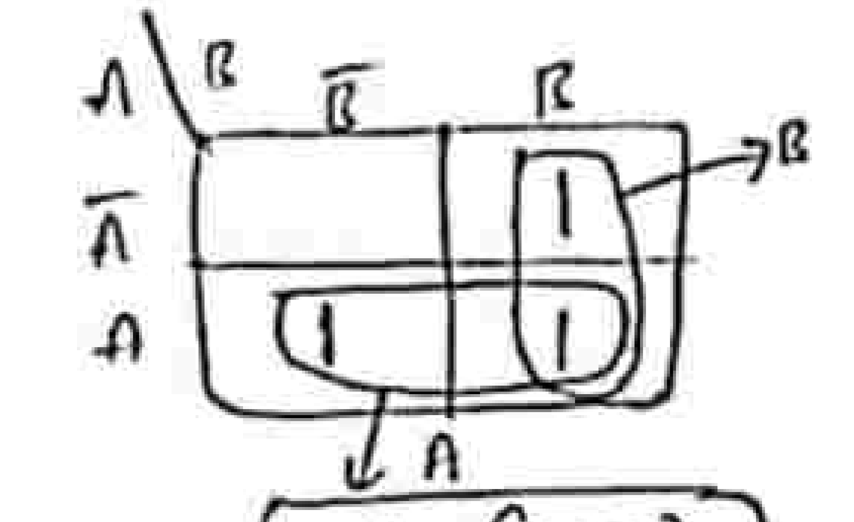
Q Problems on K-map representation — 2-variable K-map (SOP)

i, $f = A\bar{B} + AB$



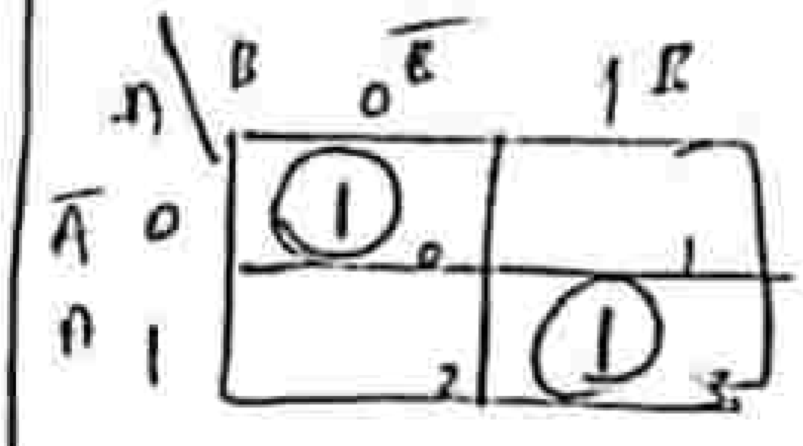
$\boxed{f = A}$

ii, $f = A\bar{B} + AB + \bar{A}B$



$\boxed{f = (A+B)}$

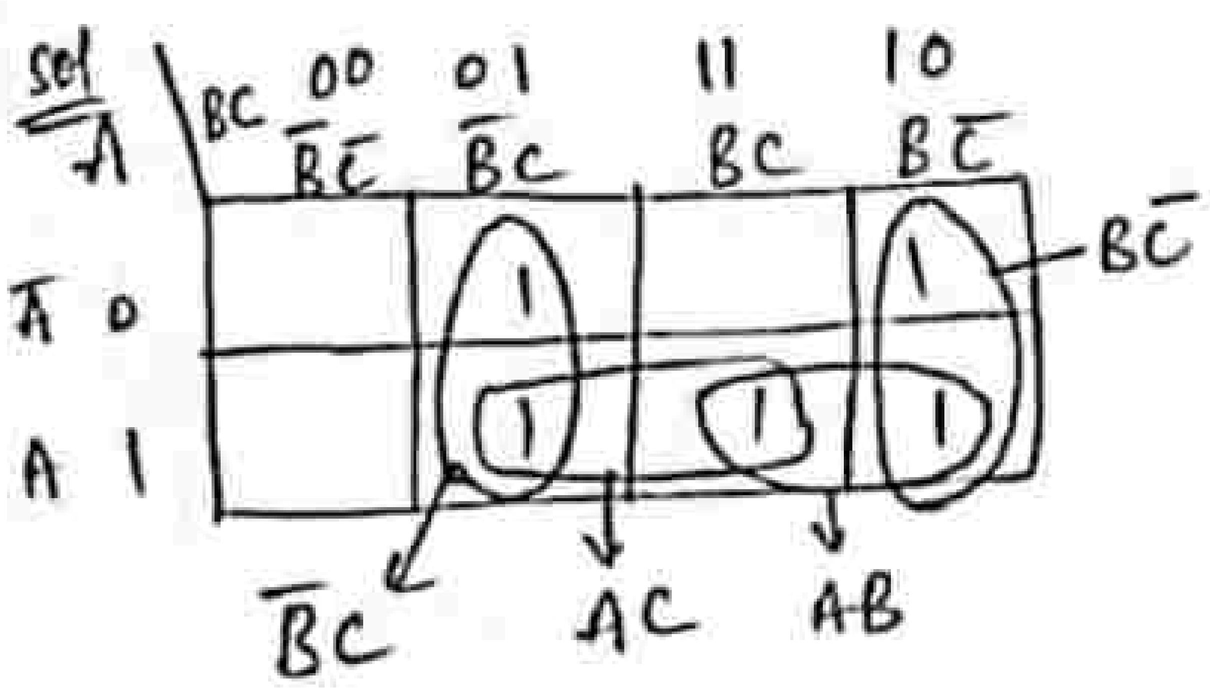
iii, $f(A,B) = (0,3)$



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

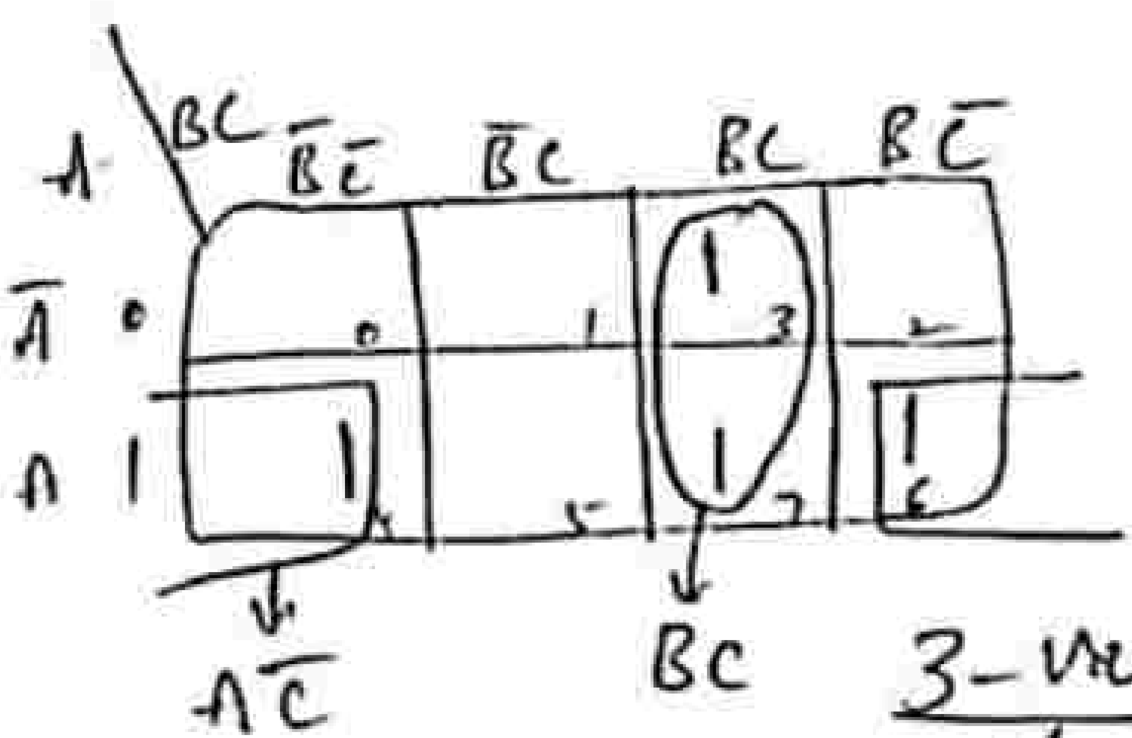
Q $f = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + AB\bar{C} + ABC$

Sol



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

$f(A,B,C) = \sum m(3,4,6,7)$

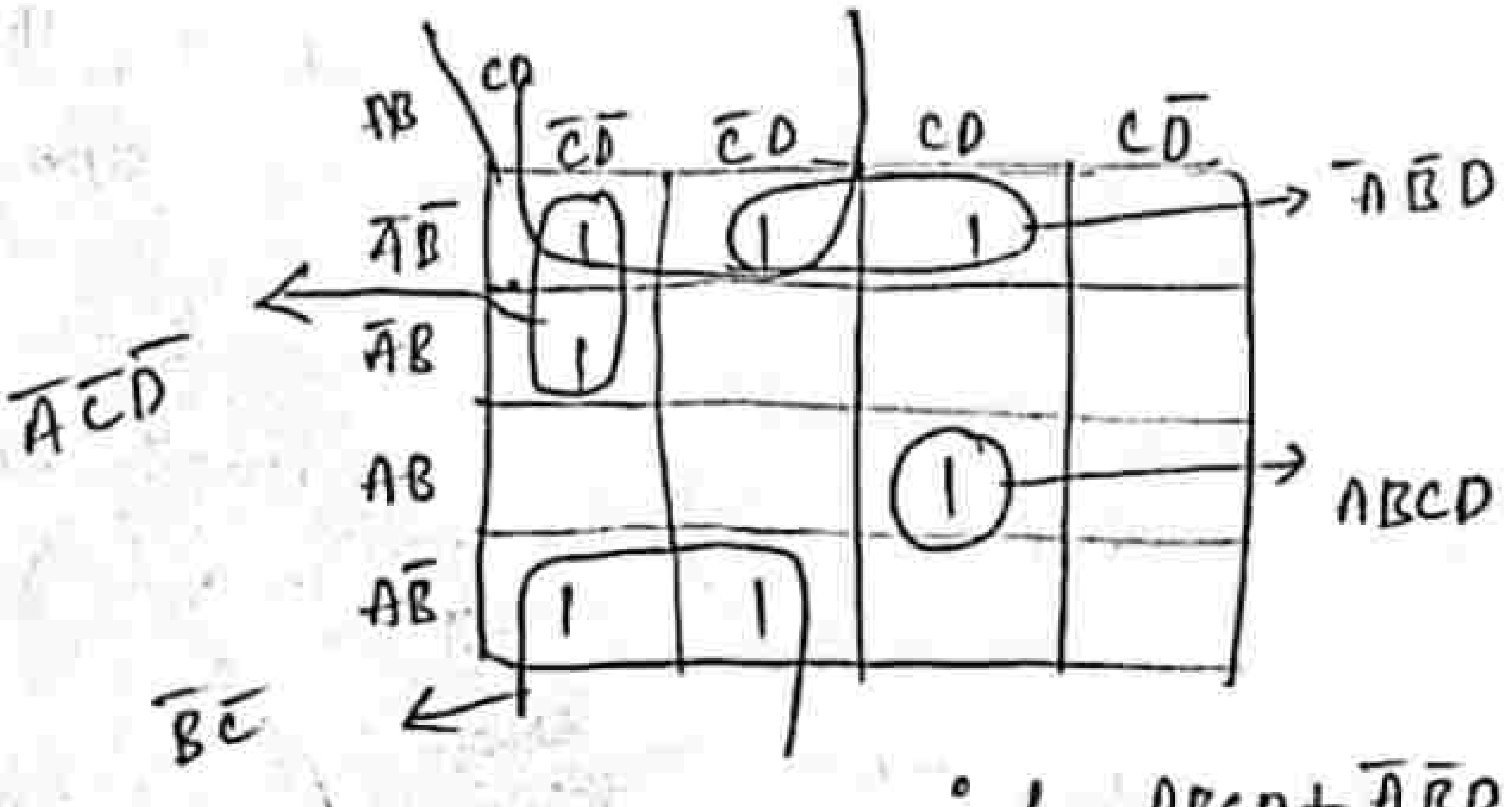


$f = BC + A\bar{C}$

3-variable K-map

Q Reduce the below expression using 4 variable K-map

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



$\therefore f = ABCD + \bar{A}\bar{B}D + \bar{A}\bar{C}\bar{D} + \bar{B}\bar{C}$

Q. Convert the following in the SSOP form and calculate the minterms

ⓐ $F(A,B) = \bar{A}B + B$

Sol Given $F(A,B) = \bar{A}B + B$

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

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

$= \bar{A}B + AB + \bar{A}B$ / $\because \bar{A}B + \bar{A}B = \bar{A}B$ (If same digits are more than two then it become one)

$= \bar{A}B + AB$

$= 01 \quad 11$

$= m_1 + m_3$

$= \Sigma m(1,3)$

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

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

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

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

$110 \quad 101 \quad 111 \quad 110$

$6 \quad 5 \quad 7 \quad 3$

$= m_6 + m_5 + m_7 + m_3$

It should be in proper order

$\therefore m_3 + m_5 + m_6 + m_7$

$\Rightarrow \Sigma m(3,5,6,7)$

Q. Convert the following in the SPOS form and calculate the maxterms.

ⓐ $F(A,B) = A(\bar{A}+\bar{B})$

$= A+0\,(\bar{A}+\bar{B})$

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

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

$00 \quad 01 \quad 11$

$= M_0\ M_1\ M_2$

$= \Pi M(0,1,2)$

ⓑ $f(A,B,) = A(\bar{A}+\bar{B})B$

$= A+0\,(\bar{A}+\bar{B})\,B+0$

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

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

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

$= (A+B)(A+\bar{B})(\bar{A}+B)$ / $\because (A+B)(A+B) = (A+B)$, $(\bar{A}+B)(\bar{A}+B) = (\bar{A}+B)$

$00 \quad 01 \quad 10$

$M_0 \cdot M_1 \cdot M_2 = \Pi M(0,1,2)$

Note:- K-map consist of a no. of squares. Each one of the square is cell. To do K-map minimization the expression should be in SOP form (or) POS form.

This is extremely useful and extensively used in the minimization of function of 2-variable K-map, 3-variable K-map, 4-variable K-map and so on.

Q Simplify, $f(A,B,C,D) = \Sigma m(0,1,3,7,11,15)$

$AB \backslash CD$	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	1	1	
$\bar{A}B$			1	
AB			1	
$A\bar{B}$			1	

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

$\therefore f = \bar{A}\bar{B}\bar{C} + CD$

Q $f(A,B,C,D) = \Sigma m(0,5,1,2,3,7,8,10)$

$AB \backslash CD$	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	1	1	1
$\bar{A}B$		1	1	
AB				
$A\bar{B}$	1			1

$\bar{B}\bar{D}$ $\quad$ $\bar{A}D$

$\therefore f = \bar{B}\bar{D} + \bar{A}D.$

⇒ Minimization of Boolean Expression with help of POS by using K-map

$A \backslash B$	0 = B	1 = $\bar{B}$
$0 = A$	$A+B$	$A+\bar{B}$
$1 = \bar{A}$	$\bar{A}+B$	$\bar{A}+\bar{B}$

2-variable

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

3-variable

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

Q $f(A,B) = (A+B)(\bar{A}+B)$

	B	$\bar{B}$
A	0	
$\bar{A}$	0	

$B = f$

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

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

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

$(A+B)$

Q $f = \Pi M(0,2,3,4,5,6,7,9,12,8)$

$A+B \backslash C+D$	$C+D$	$C+\bar{D}$	$\bar{C}+\bar{D}$	$\bar{C}+D$
$A+B$	0		0	0
$A+\bar{B}$	0	0	0	0
$\bar{A}+\bar{B}$	0			
$\bar{A}+B$	0	0		

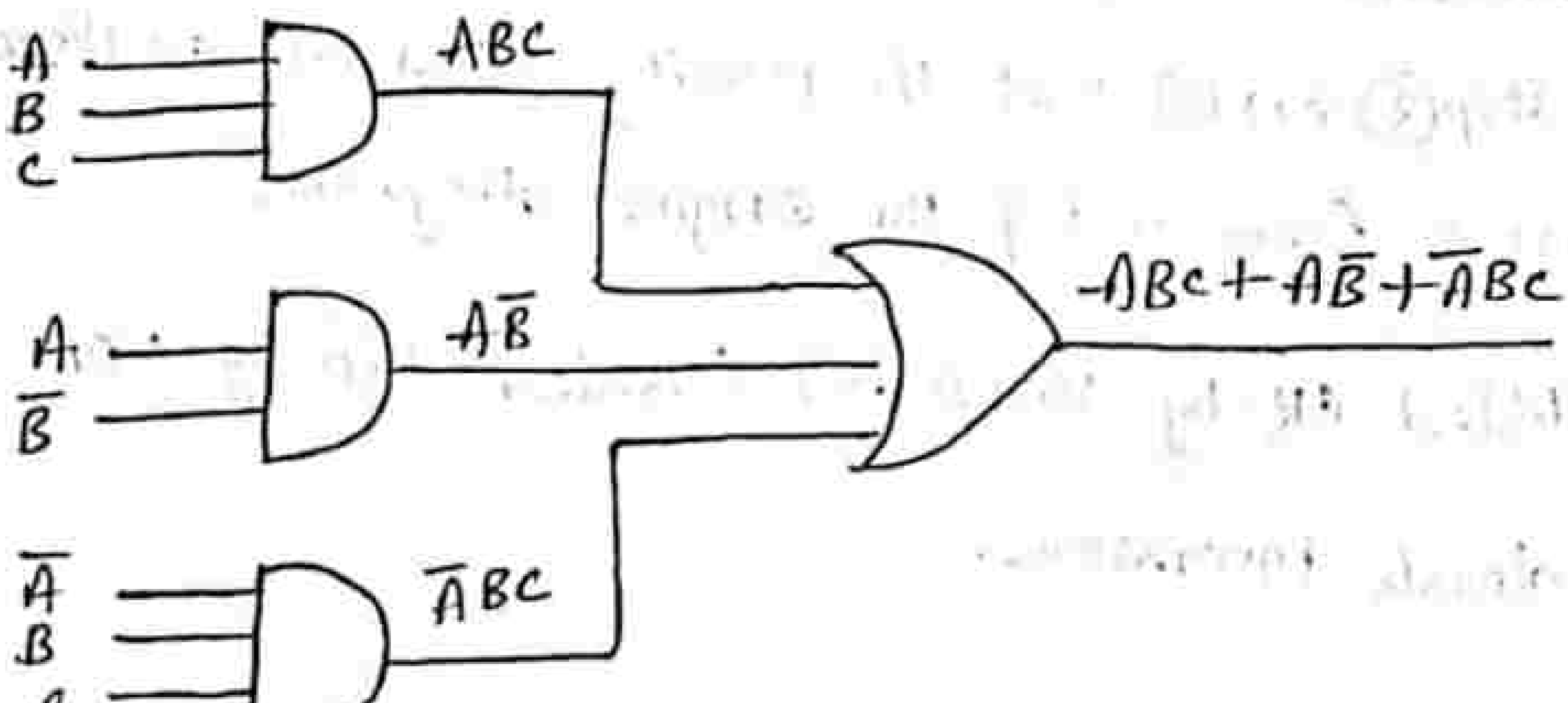
$(C+D)$ $\quad$ $(\bar{A}+B+C)$ $\quad$ $(A+\bar{B}+C)$ $\quad$ $(A+\bar{C})$

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

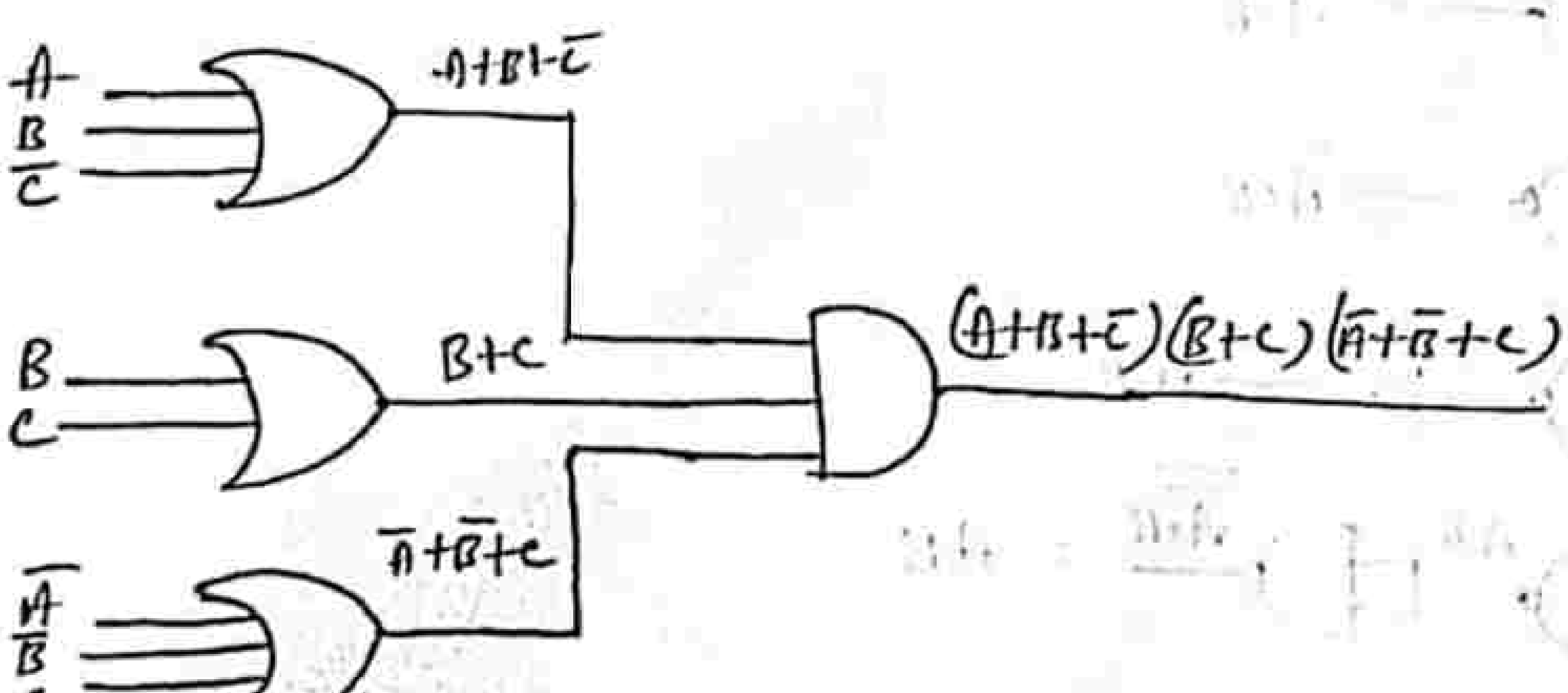
⇒ NAND and NOR Implementation :—

In the design of digital circuits, the minimal boolean expressions are usually obtained in Sop-form (or) pos-form. Sometimes the minimal expressions may also be expressed in hybrid form.

For example $f = ABC + A\bar{B} + \bar{A}BC$
Given example is in Sop-form. So, Sop expression can be implemented by using AND/OR logic as shown below.



The form of given expression is $f = (A+B+\bar{C})(B+C)(\bar{A}+\bar{B}+C)$.
This can be implemented using OR/AND logic as shown below.



The procedure to convert an AOI logic to NAND logic (or) NOR logic is given below. (127)

(1) Draw the circuit in AOI logic

(2) If NAND hardware is chosen, add a circle at the output of each AND gate and at the inputs to all the OR gates.

(3) If NOR hardware is choosen, add a circle at the output of each OR gate and at the inputs to all the AND gates.

(4) Add (or) subtract an inverter on each line that received a circle in step (2) (or) (3) that the polarity of signals on those lines remains from that of the original diagram.

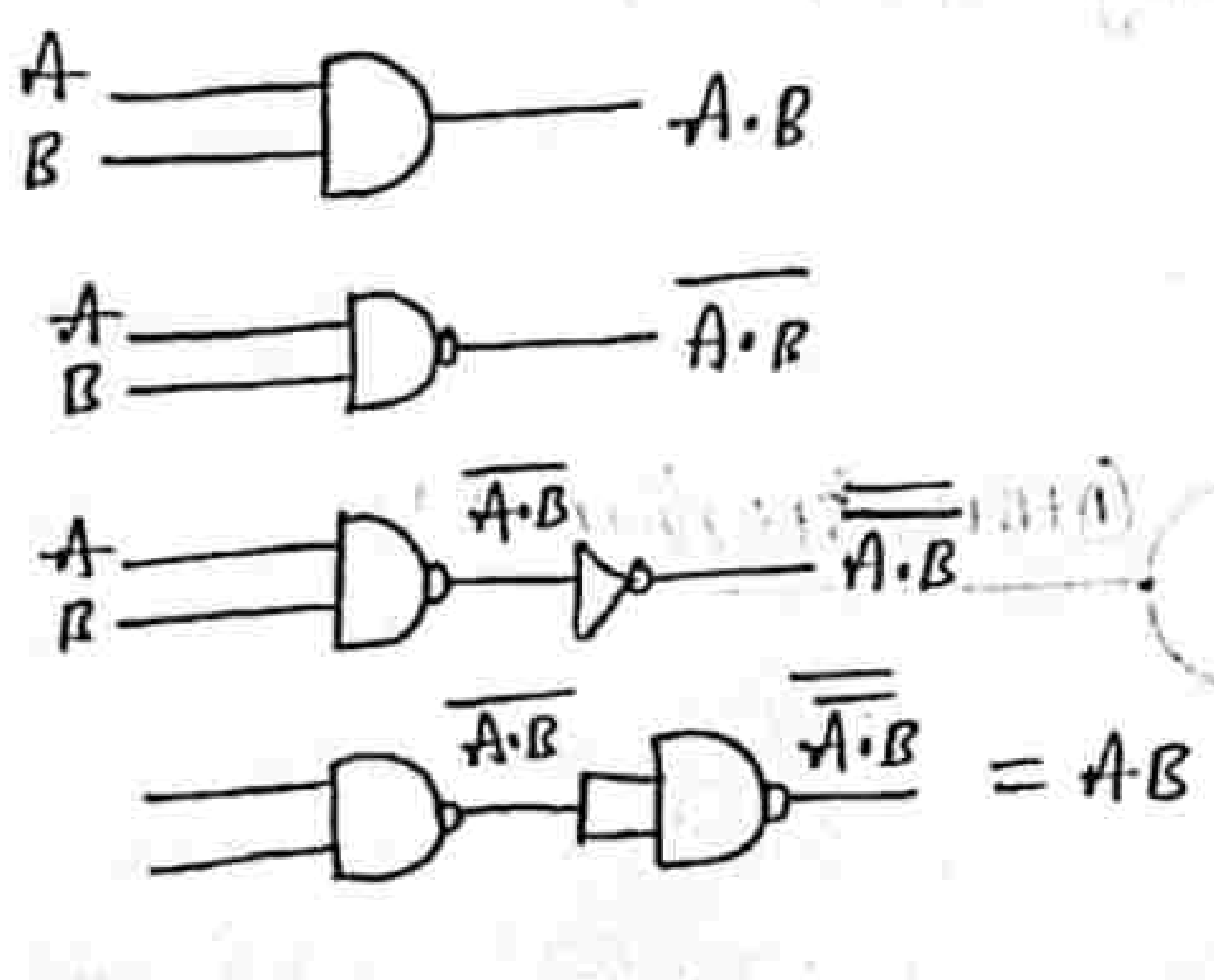
(5) Replace bubbled OR by NAND and bubbled AND by NOR.

(6) Eliminate double inversions.

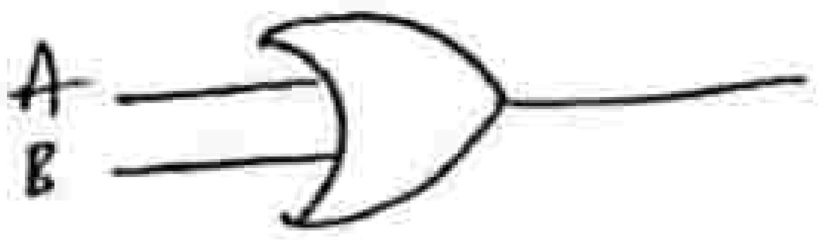
(1) <u>Implementation of AND gate using NAND Gate</u> :—

(1)

(2)



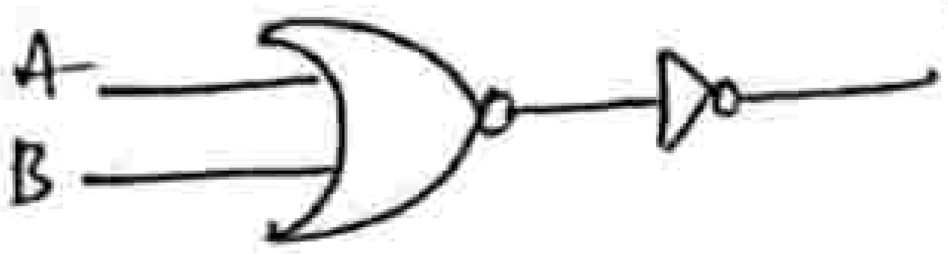
④ OR gate using NOR :

① 

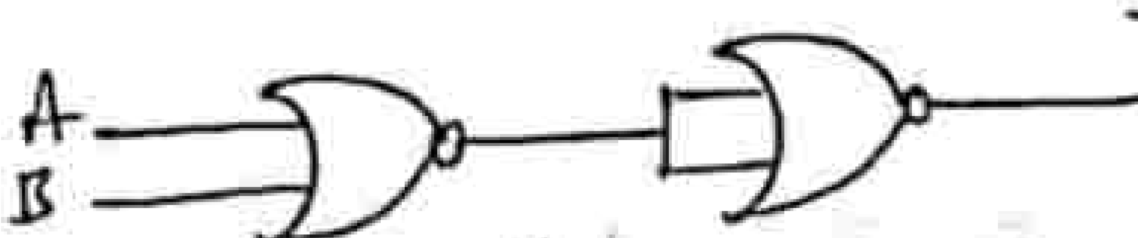
$A+B$

② 

$\overline{A+B}$

③ 

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

④ 

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

⑤ Realization of NOT gate using NAND & NOR

① 

$\overline{A}$

② 

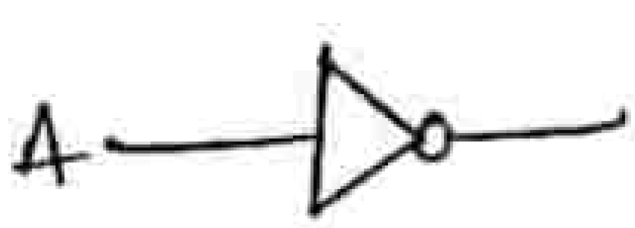
$\overline{A}$

③ 

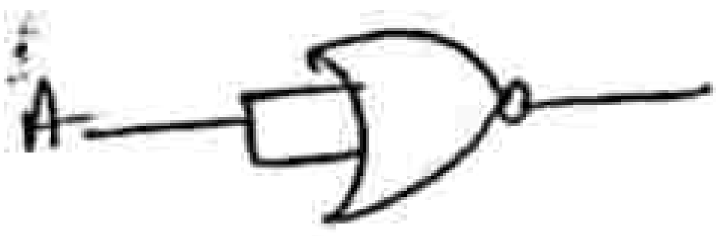
$\overline{A \cdot A} = \overline{A}$

single i/p AND gate.

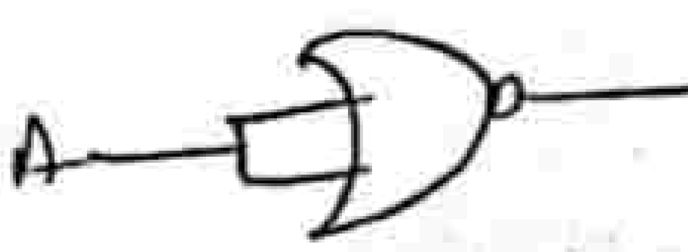
⇒ Using NOR



$\overline{A}$



$\overline{A}$

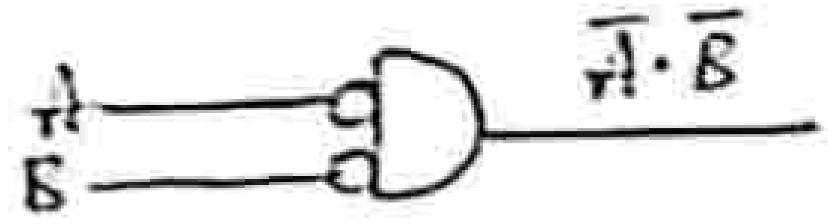


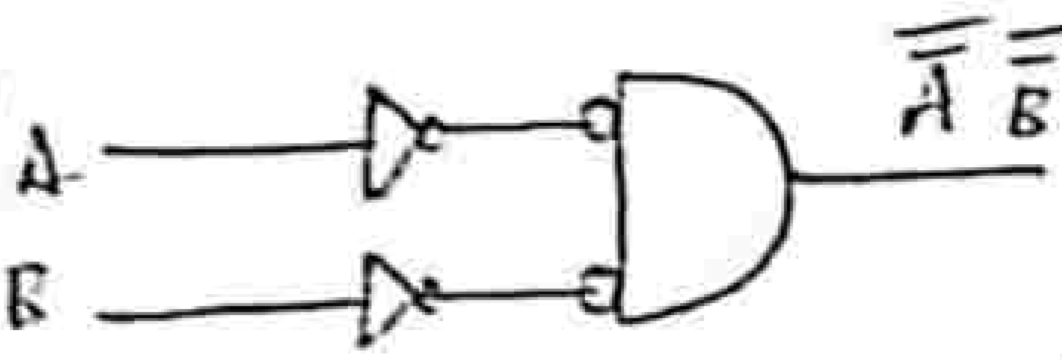
$\overline{A+A} = \overline{A}$

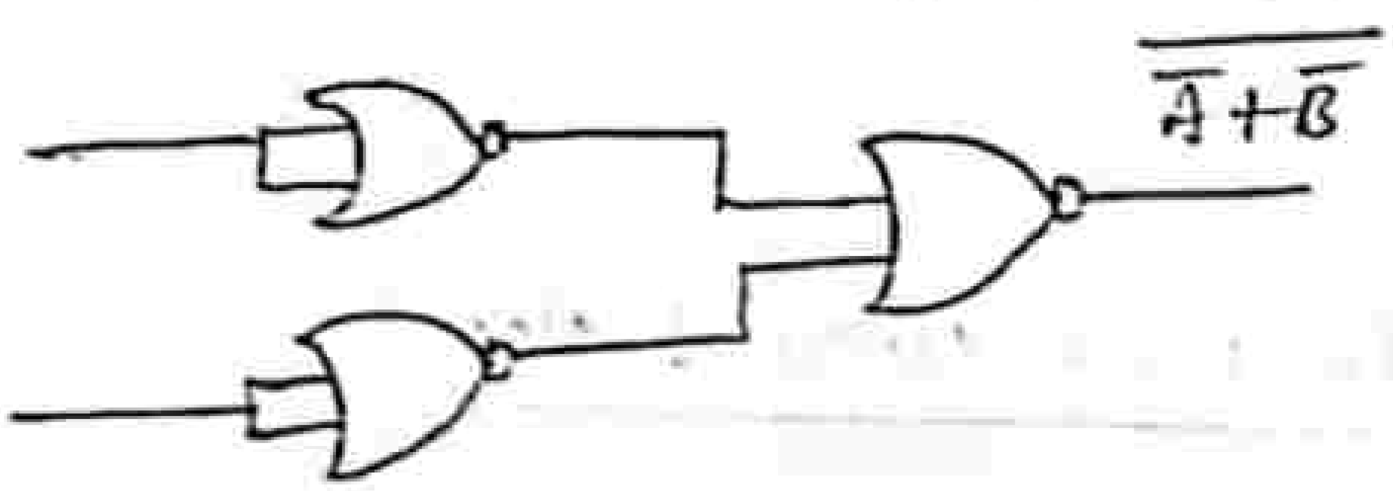
Single i/p NOR gate.

② Realization of AND gate Using NOR Gate :-

①

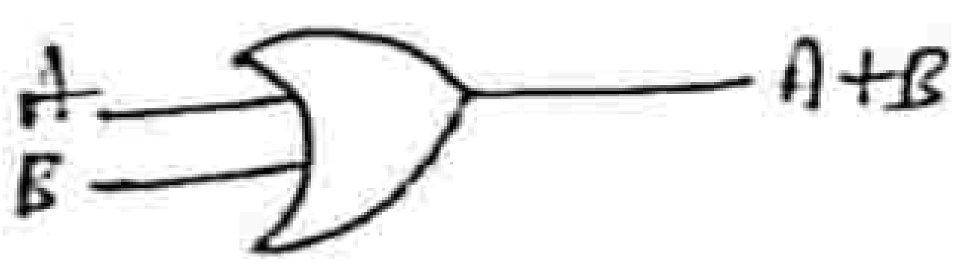

②


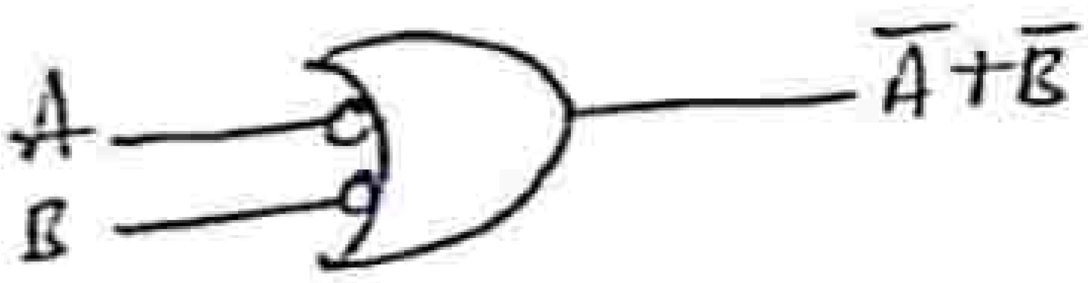
③


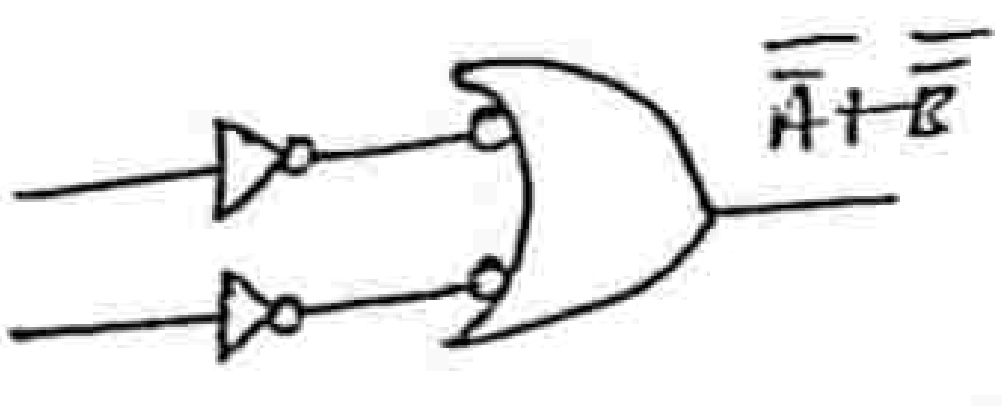
④


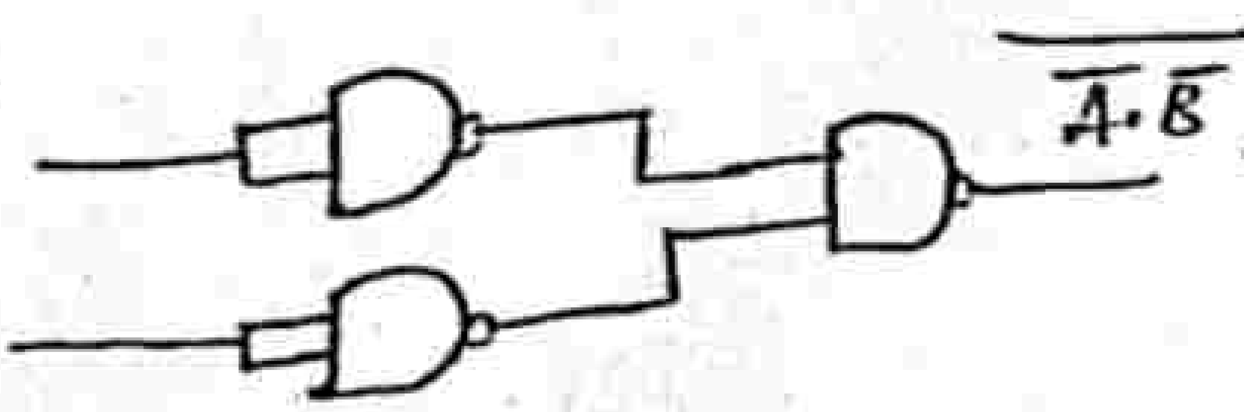
$= \bar{\bar{A}}\cdot\bar{\bar{B}} = A\cdot B$

③ Realization of OR gate Using NAND & NOR gates :

①


②


③


④


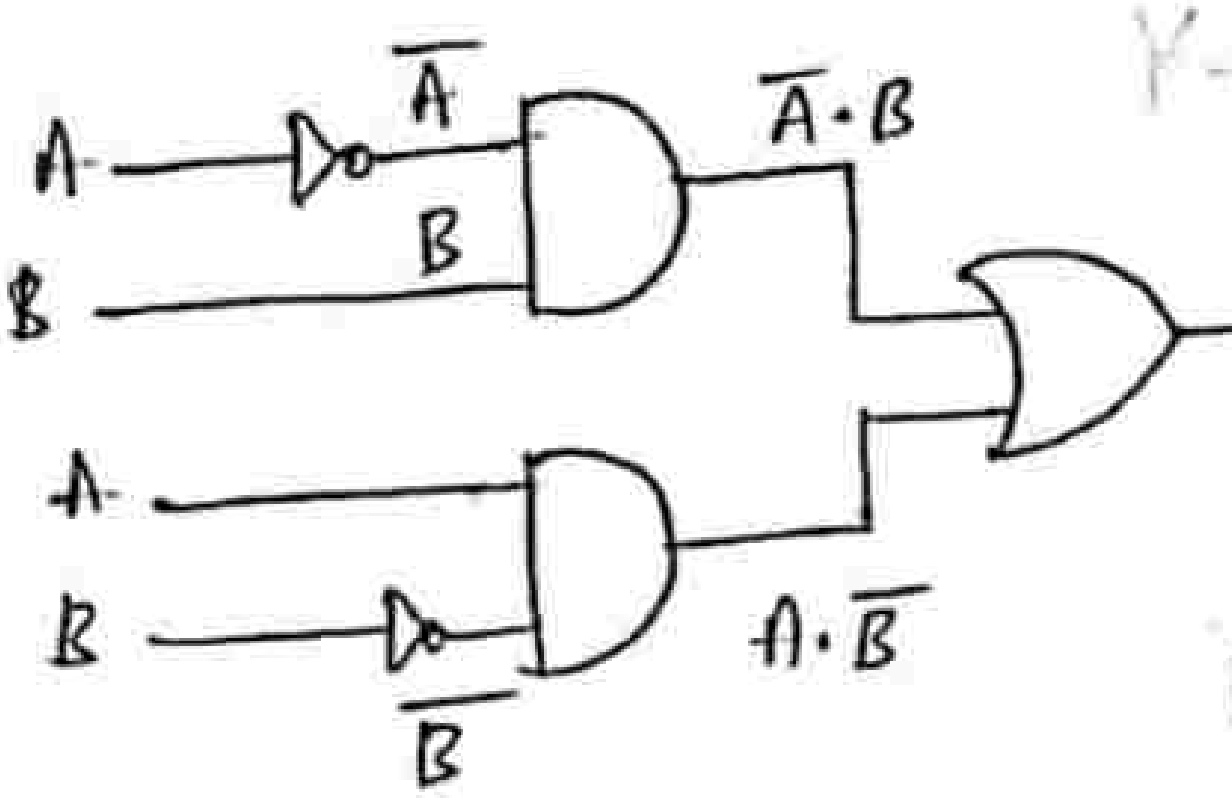
$= \bar{\bar{A}}+\bar{\bar{B}} = A+B$

Realization of EX-OR gate using NAND & NOR using NAND gate :-

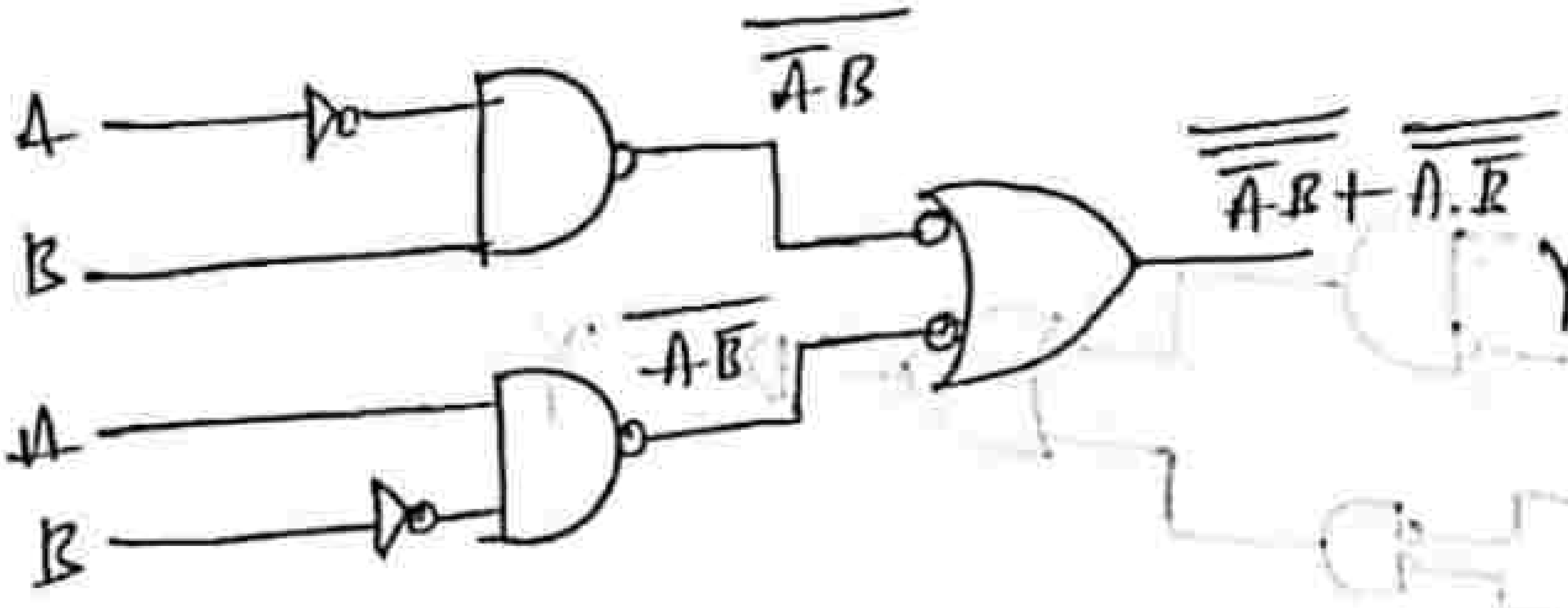
(130)

① 

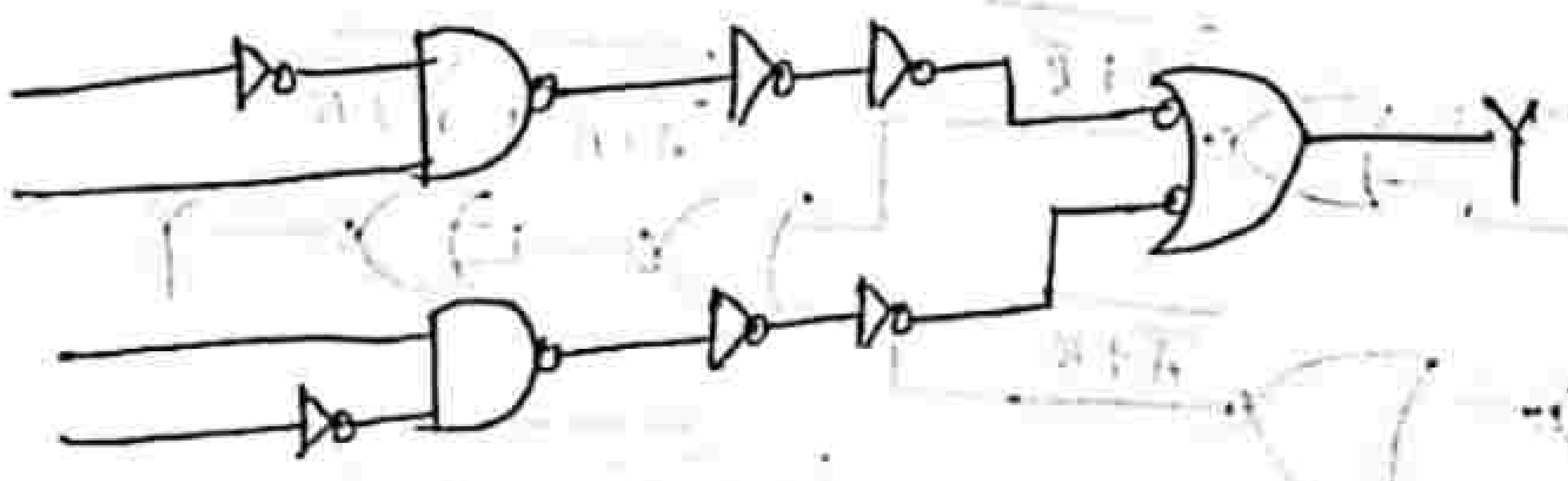
$Y = \overline{A}B + A\overline{B} = A \oplus B$

② 

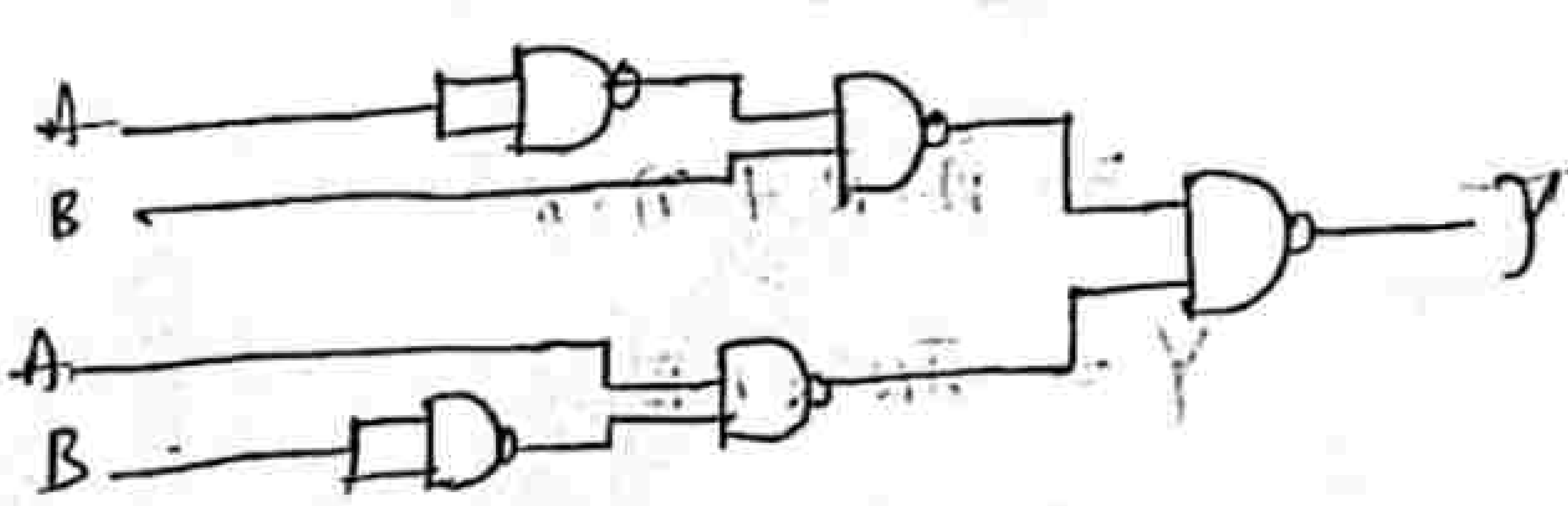
$Y = \overline{A} \cdot B + A \cdot \overline{B}$

③ 

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

④ 

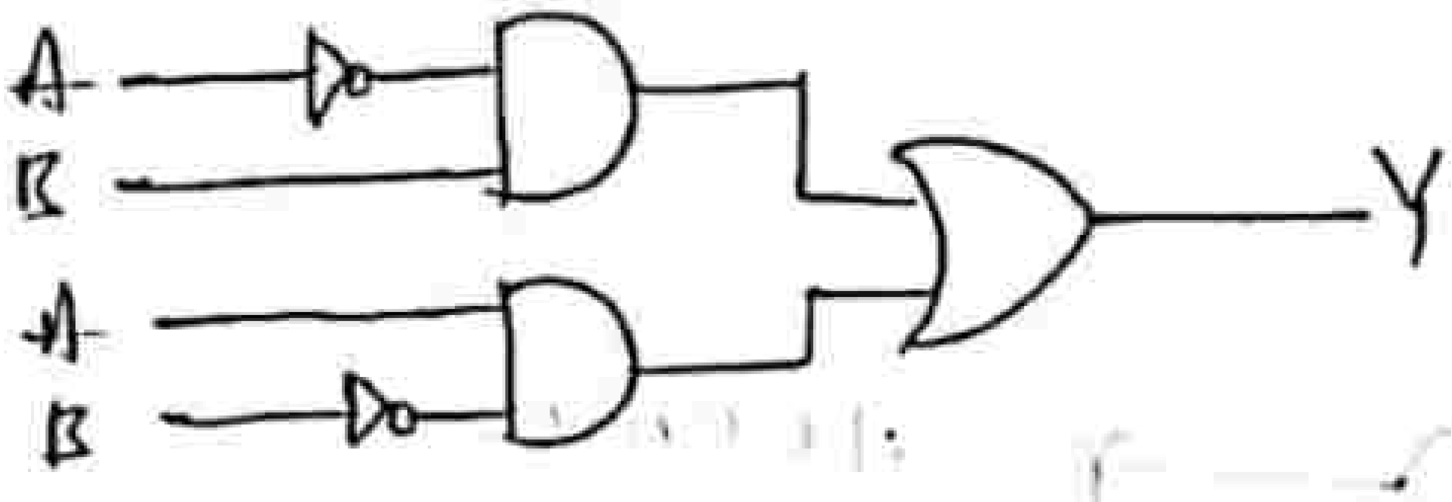
⑤ Instead of not gate place single i/p NAND gate and, Instead of bubbled OR gate place NAND gate

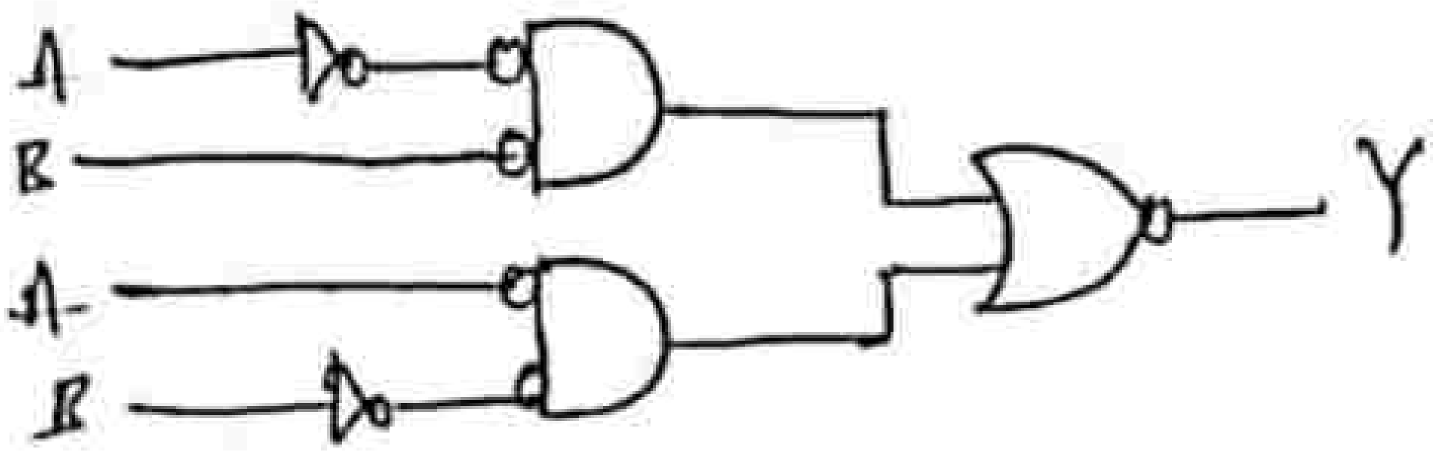


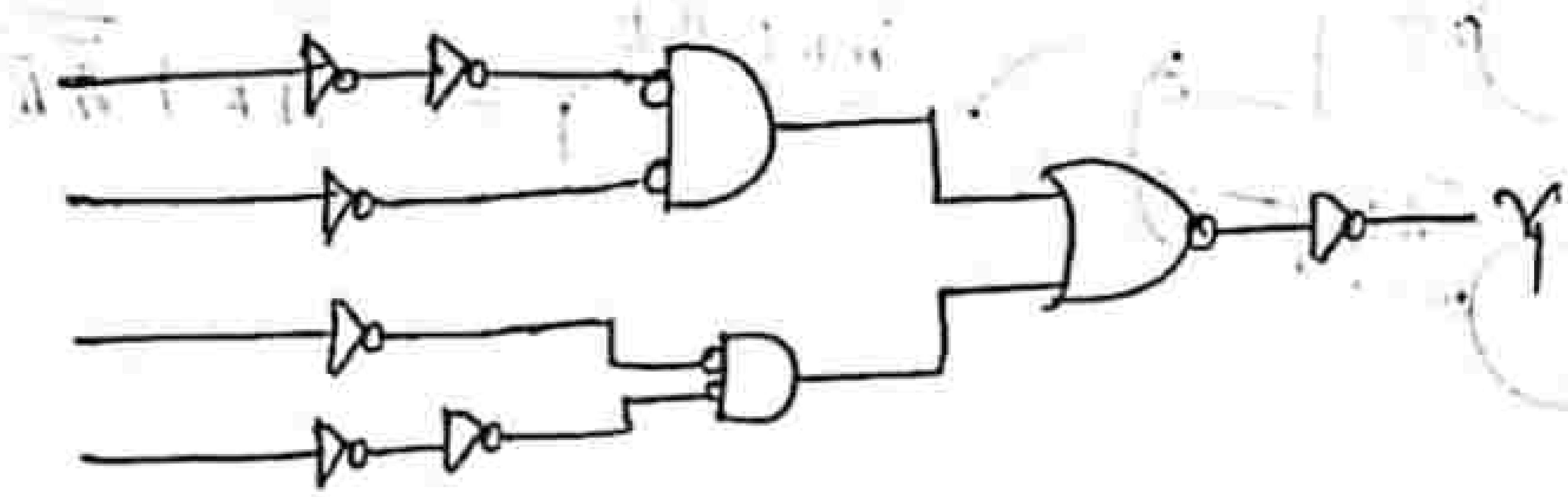
⇒ Using NOR Gate :-

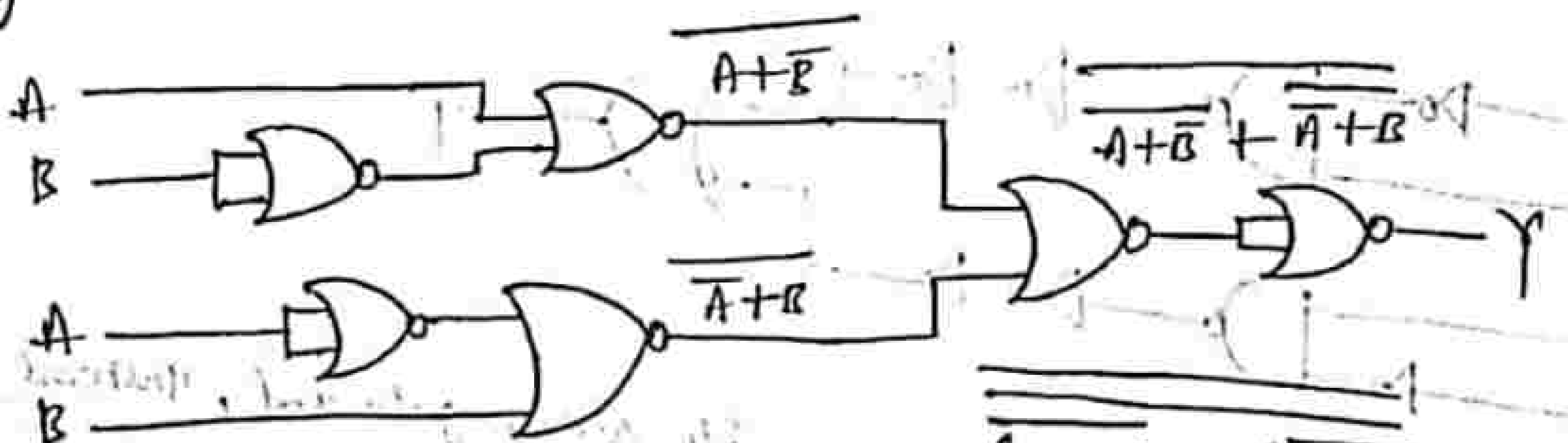
① 

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

② 

③ 

④ 

⑤ 

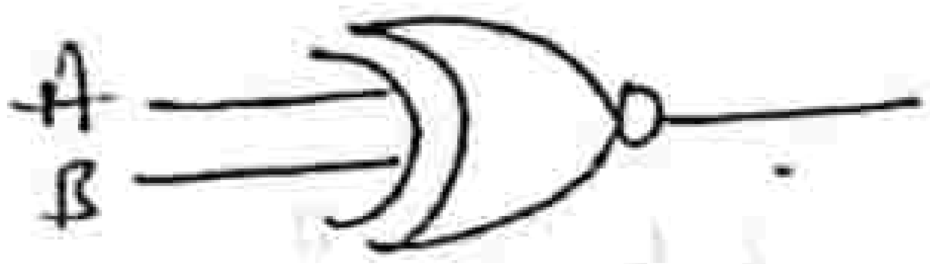
$$= \overline{\overline{\overline{(A+\bar{B})} + \overline{(\bar{A}+B)}}}$$

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

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

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

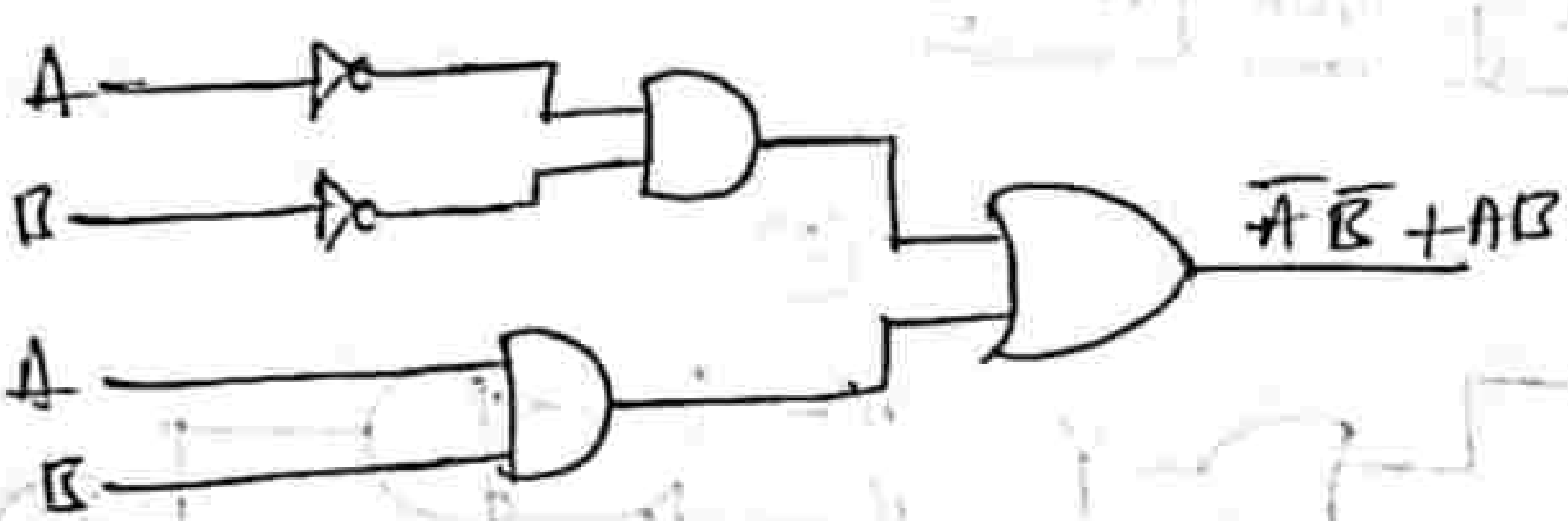
⟹ Realization of X-NOR gate Using NAND & NOR :-



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

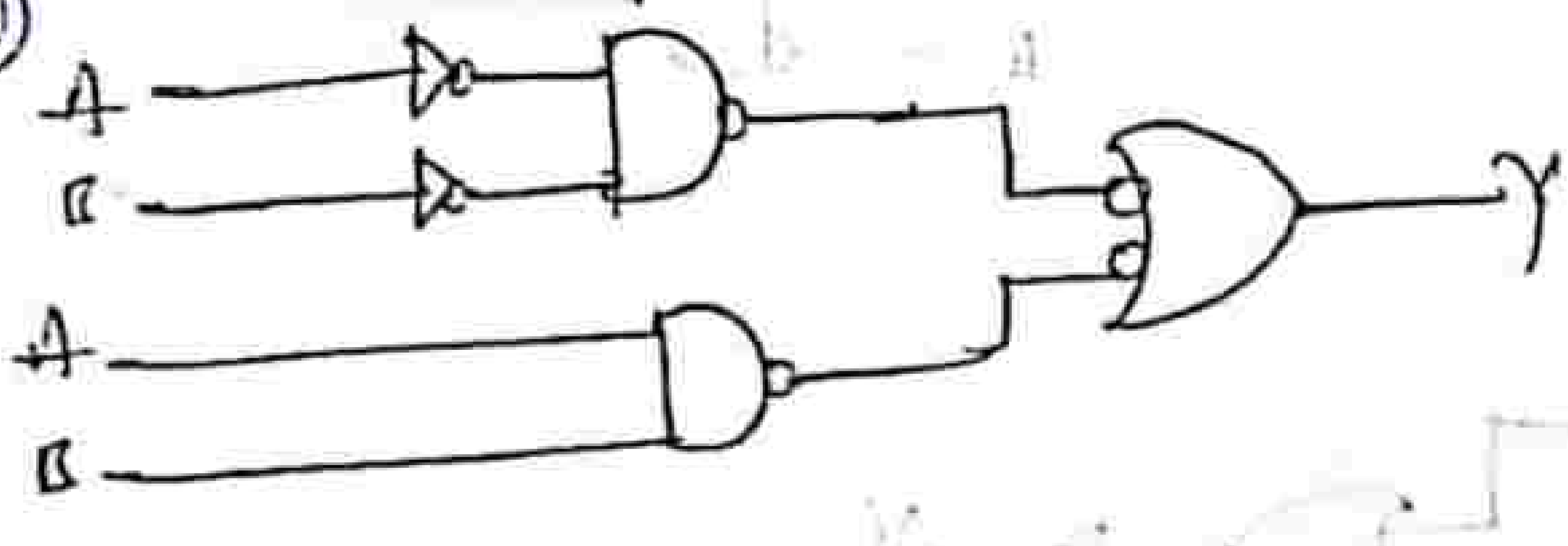
⟹ Using NAND Implementation :-

(1)

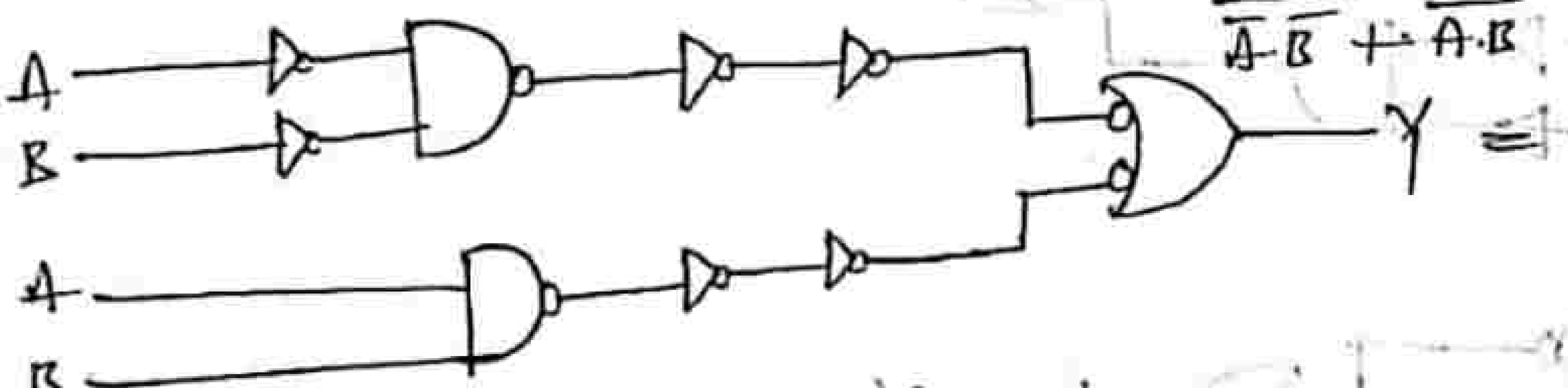


$Y = \bar{A}\bar{B}+AB$

(2)

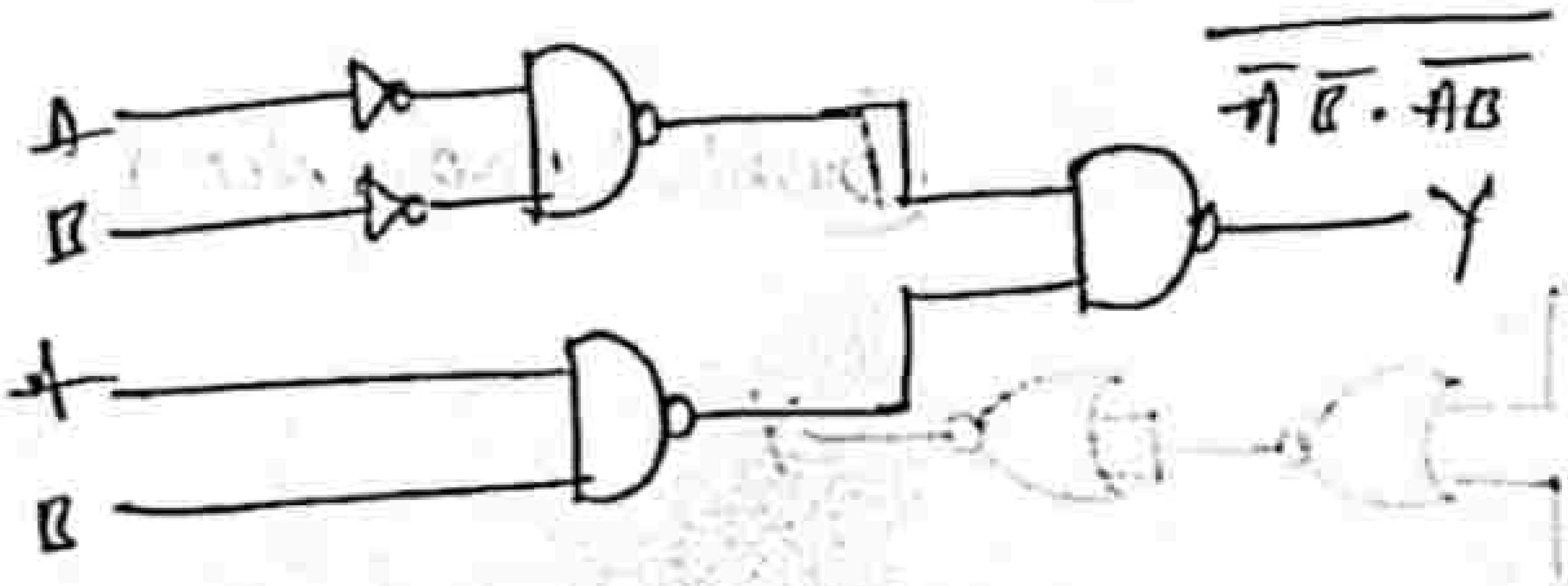


(3)

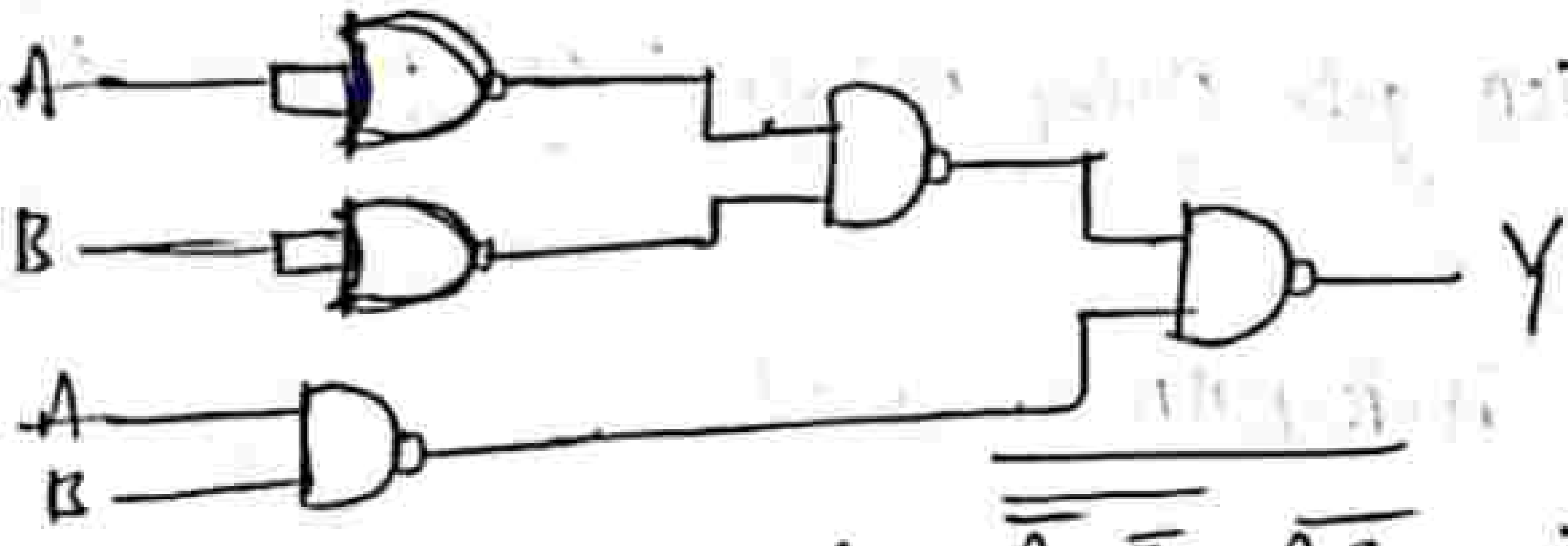


$Y = \bar{A}\bar{B}+AB$

(4)



(Bubbled OR gate = NAND gate)

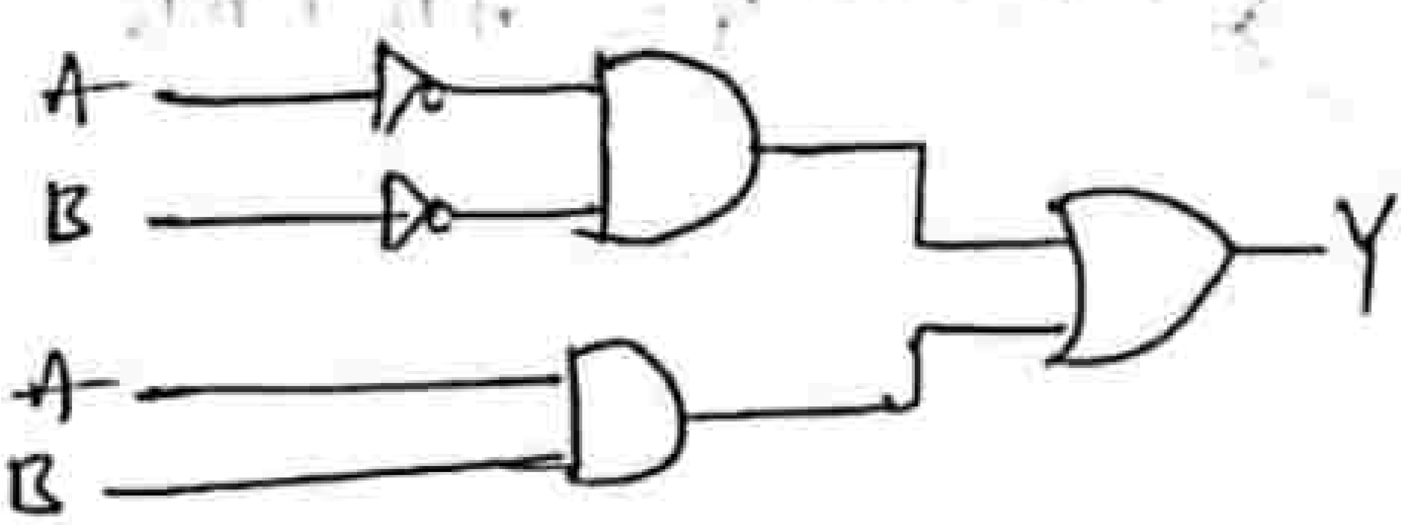


$\overline{\bar{A}\cdot\bar{B}\cdot\bar{A}\cdot\bar{B}}$

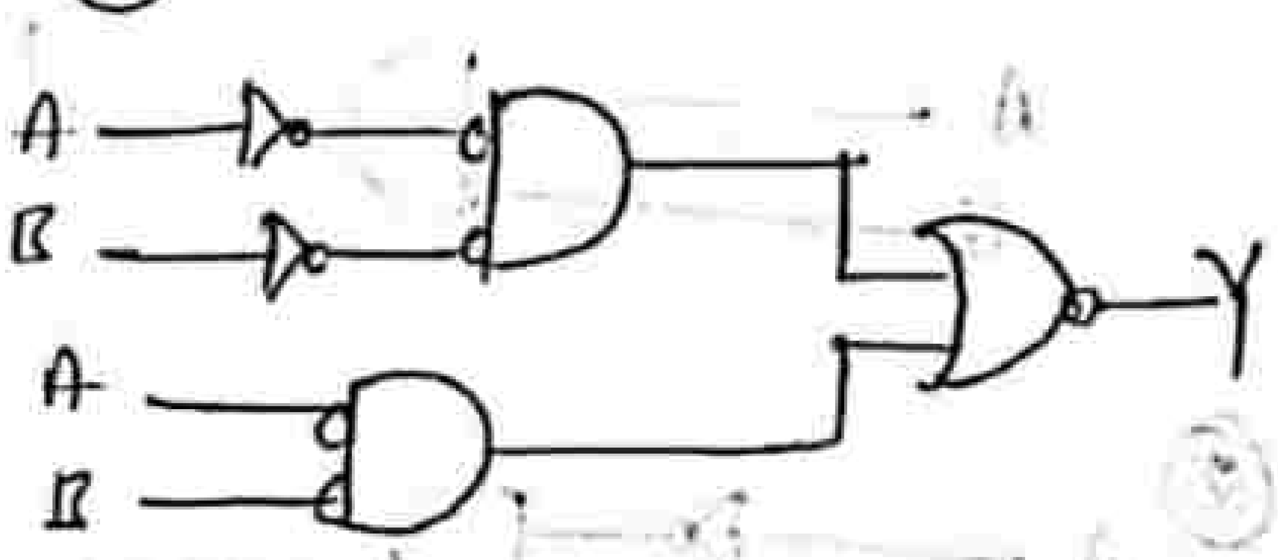
$$Y = \overline{\overline{\bar{A}\cdot\bar{B}}\ \overline{AB}} = \overline{\overline{\bar{A}\cdot\bar{B}}} + \overline{\overline{AB}}$$
$$= \bar{A}\cdot\bar{B} + A\cdot B$$

⇒ X-NOR Gate using NOR gate :-

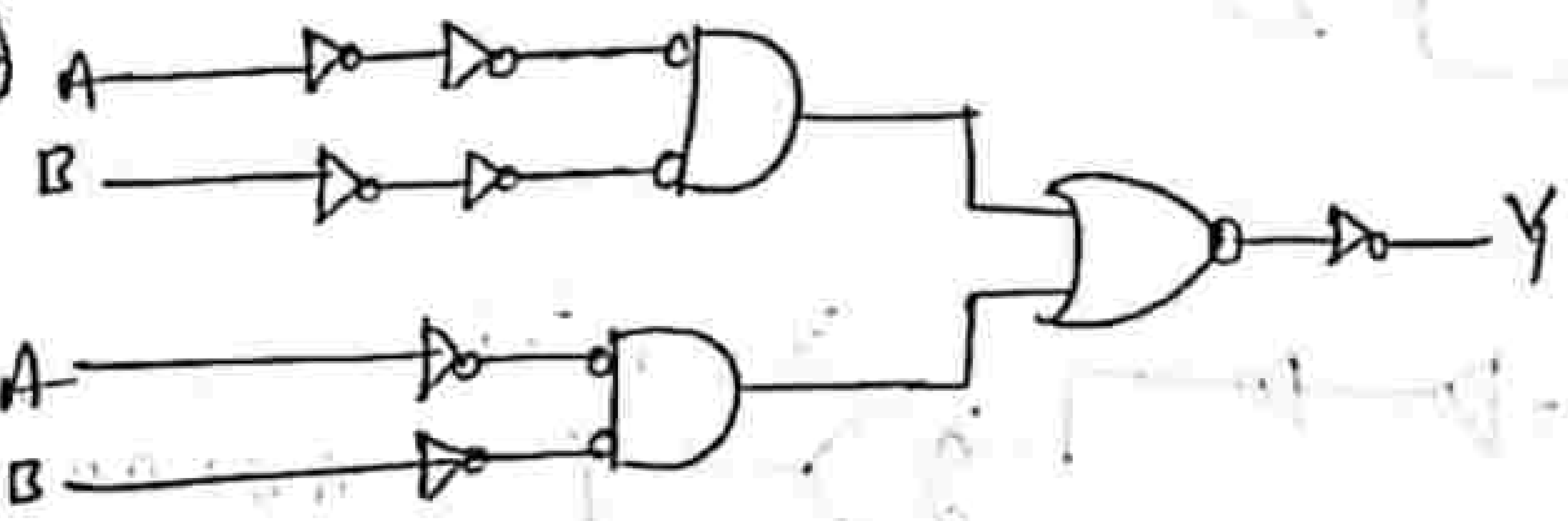
①



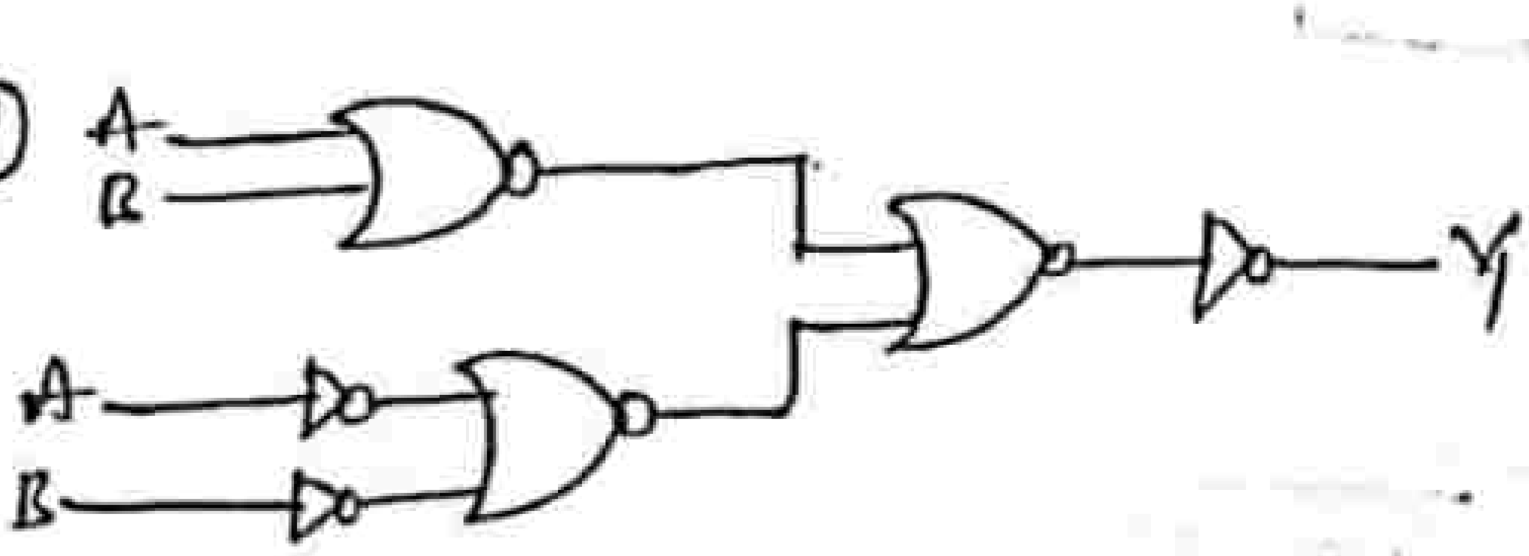
②



③

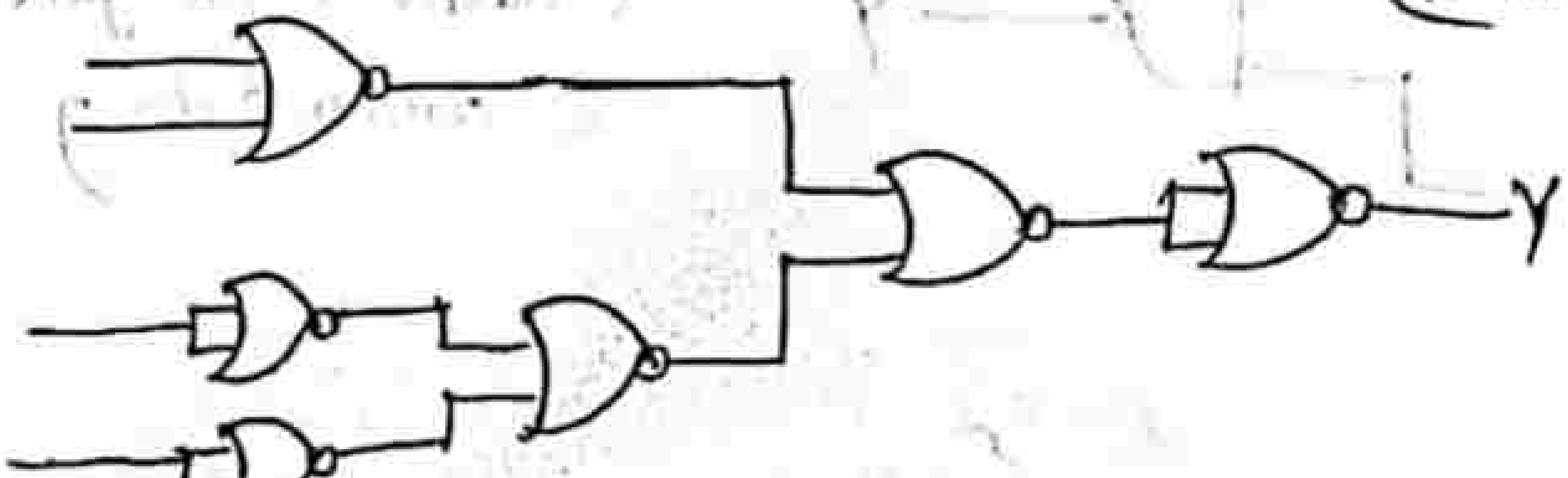


④



(bubbled AND = NOR)

⑤



⇒ Implement the following functions using NAND Gates :—

(134)

ⓐ $F_1 = A(B+CD) + \overline{BC}$

ⓑ $F_2 = w\bar{x} + \bar{x}y(z+\bar{w})$

Sol ⓐ $F_1 = A(B+CD) + \overline{BC}$

$= AB + ACD + \overline{BC}$

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

$= \underline{AB + \bar{B}} + \underline{\bar{C} + ACD}$

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

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

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

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

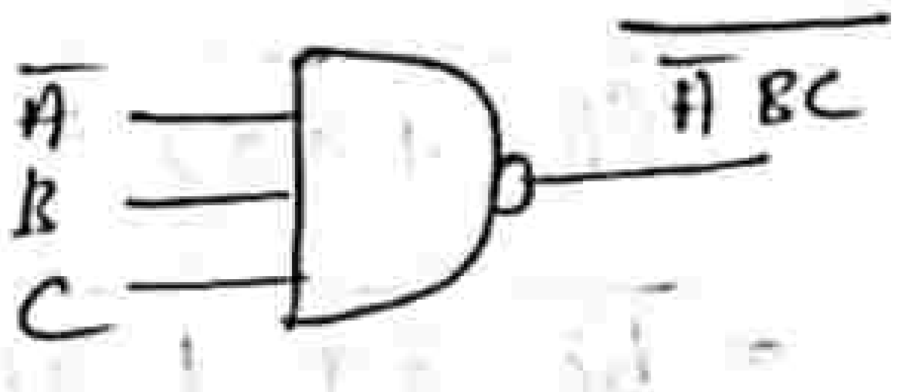
$= \overline{\bar{A}\cdot B\cdot C}$

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

$\bar{C} + ACD = \bar{C} + AD$

Redundant Law. R.L.R

$\therefore 1+D = 1$

∴ 

ⓑ $F_2 = w\bar{x} + \bar{x}y(z+\bar{w})$

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

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

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

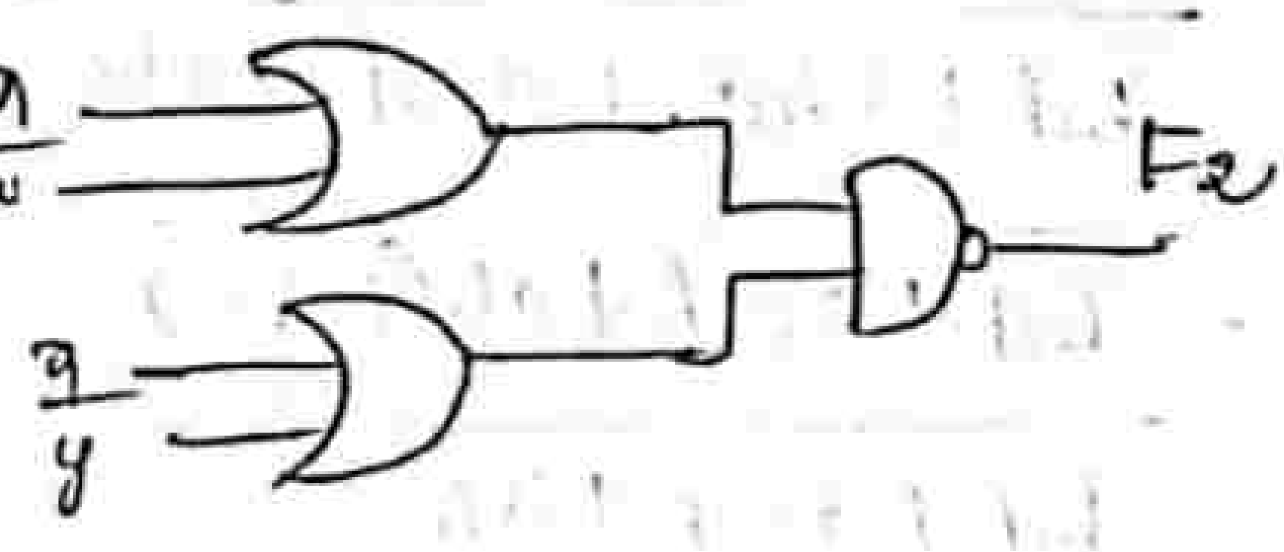
$= \bar{x}w + \bar{x}y + \bar{x}yz$

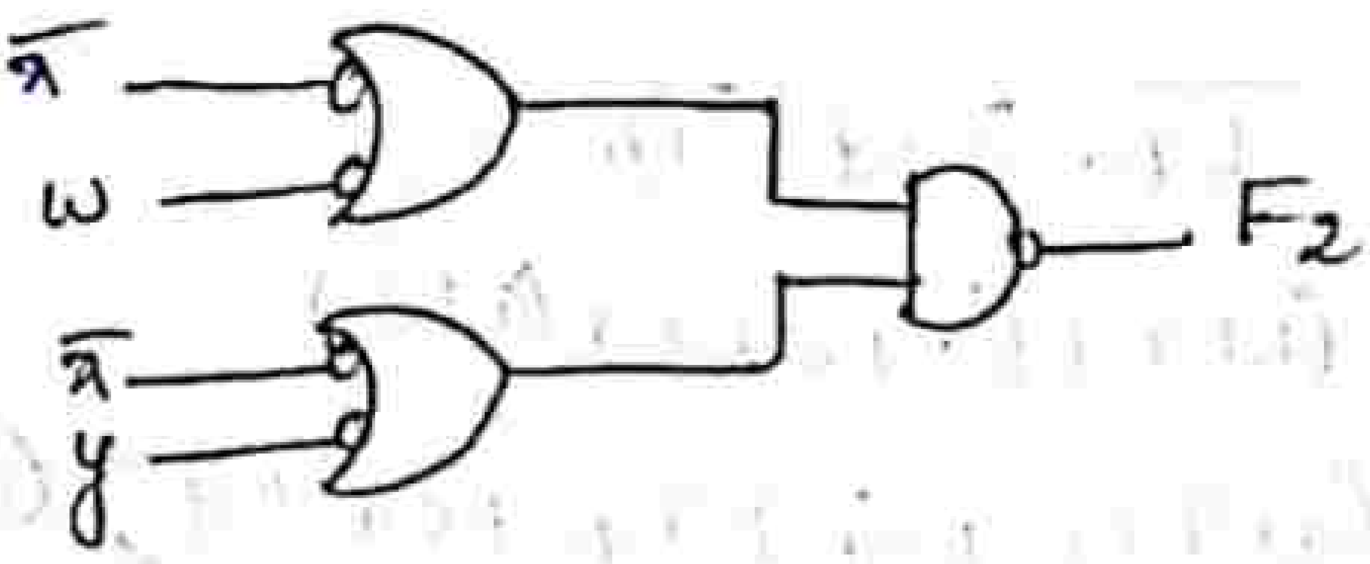
$= \bar{x}w + \bar{x}y(1+z)$

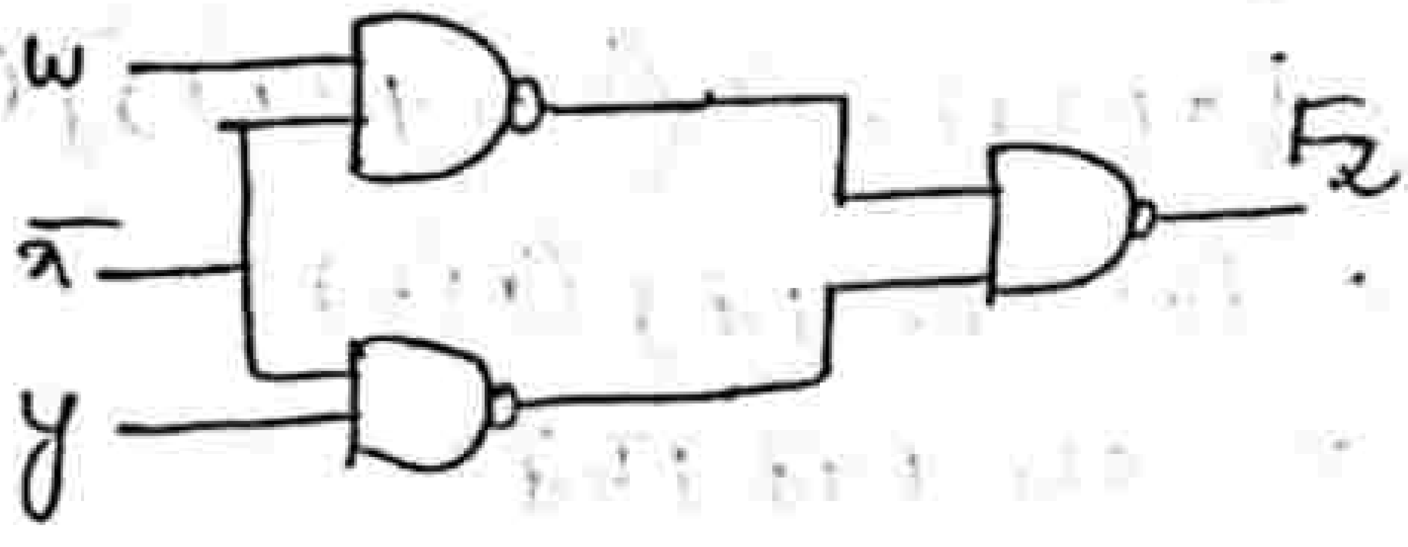
$= \bar{x}w + \bar{x}y \quad = \overline{\overline{\bar{x}w + \bar{x}y}}$

$= \bar{x}(w+y) \quad = \overline{\overline{\bar{x}w}\cdot\overline{\bar{x}y}}$

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







⟹ Find the complement of the following boolean function & Reduce them to minimum no. of literals. (135)

ⓐ $(b\bar{c}+\bar{a}d)(a\bar{b}+c\bar{d})$

ⓑ $(\bar{b}d+\bar{a}b\bar{c}+acd+\bar{a}bc)$

Sol ⓐ $\overline{(b\bar{c}+\bar{a}d)(a\bar{b}+c\bar{d})}$

$= \overline{(b\bar{c}+\bar{a}d)}+\overline{(a\bar{b}+c\bar{d})}$

$= \overline{b\bar{c}}\cdot\overline{\bar{a}d}+\overline{a\bar{b}}\cdot\overline{c\bar{d}}$

$= (\bar{b}+c)(a+\bar{d})+(\bar{a}+b)(\bar{c}+d)$

$= a\bar{b}+\bar{b}\bar{d}+ac+c\bar{d}+\bar{a}\bar{c}+\bar{a}d+b\bar{c}+bd.$

$= a\bar{b}+ac+\bar{b}\bar{d}+c\bar{d}+\bar{a}\bar{c}+\bar{a}d+b\bar{c}+bd$

$= 1$

ⓑ $\overline{\bar{b}d+\bar{a}b\bar{c}+acd+\bar{a}bc}$

$= \overline{\bar{b}d+acd+\bar{a}b(\bar{c}+c)}$

$= \overline{\bar{b}d+acd+\bar{a}b}$

$= \overline{\bar{b}d}\cdot\overline{acd}\cdot\overline{\bar{a}b}$

$= (b+\bar{d})(\bar{a}+\bar{c}+\bar{d})(a+\bar{b})$

$= (\bar{a}b+b\bar{c}+b\bar{d}+\bar{a}\bar{d}+\bar{c}\bar{d}+\bar{d})(a+\bar{b})$

$= [\bar{a}b+b\bar{c}+\bar{d}(b+\bar{a}+\bar{c}+1)](a+\bar{b})$

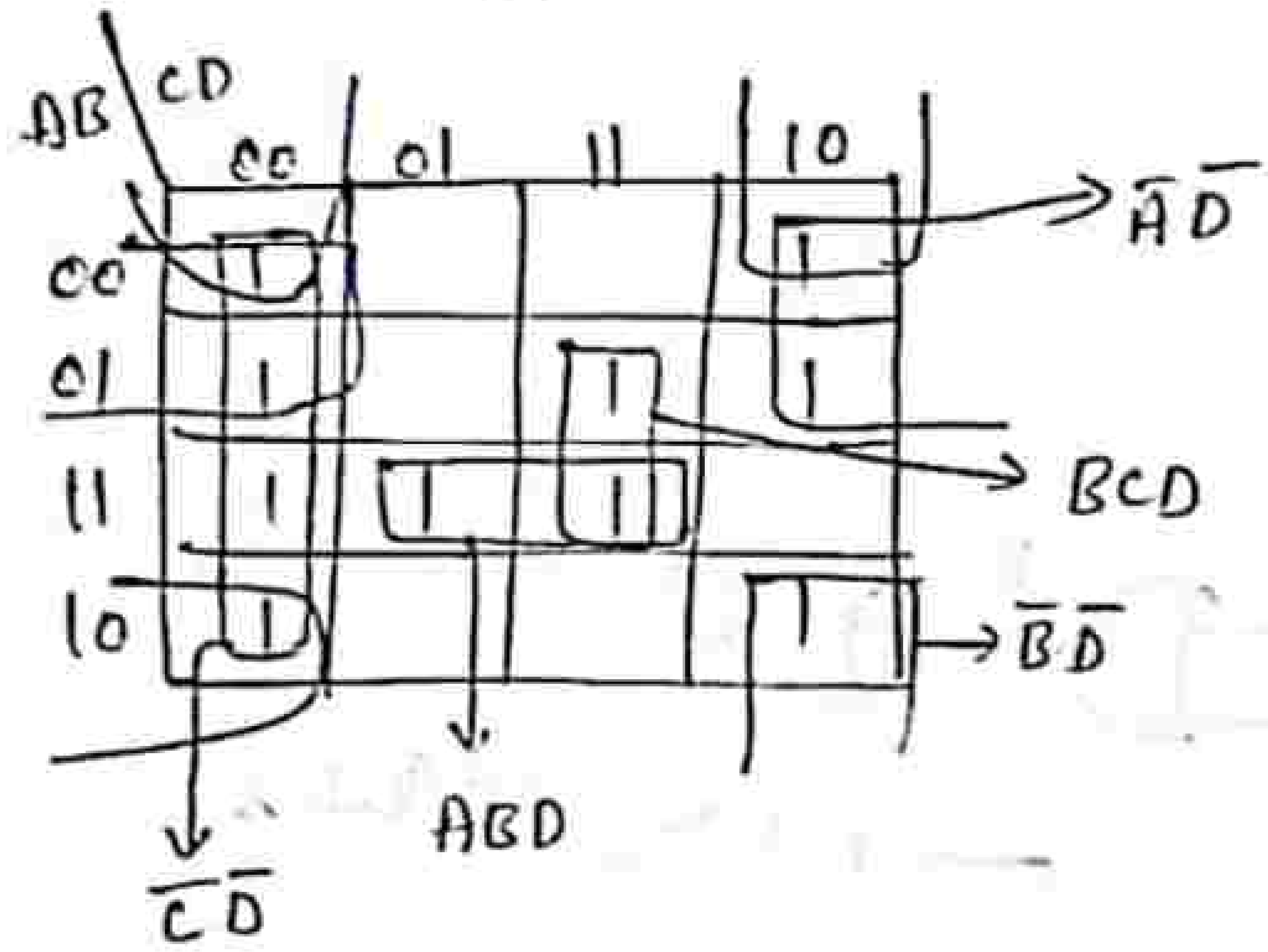
$= (\bar{a}b+b\bar{c}+\bar{d})(a+\bar{b})$

$= ab\bar{c}+a\bar{d}+\bar{b}\bar{d}$

Reduce using mapping the following expression and implement the real minimal expression in universal logic (136)

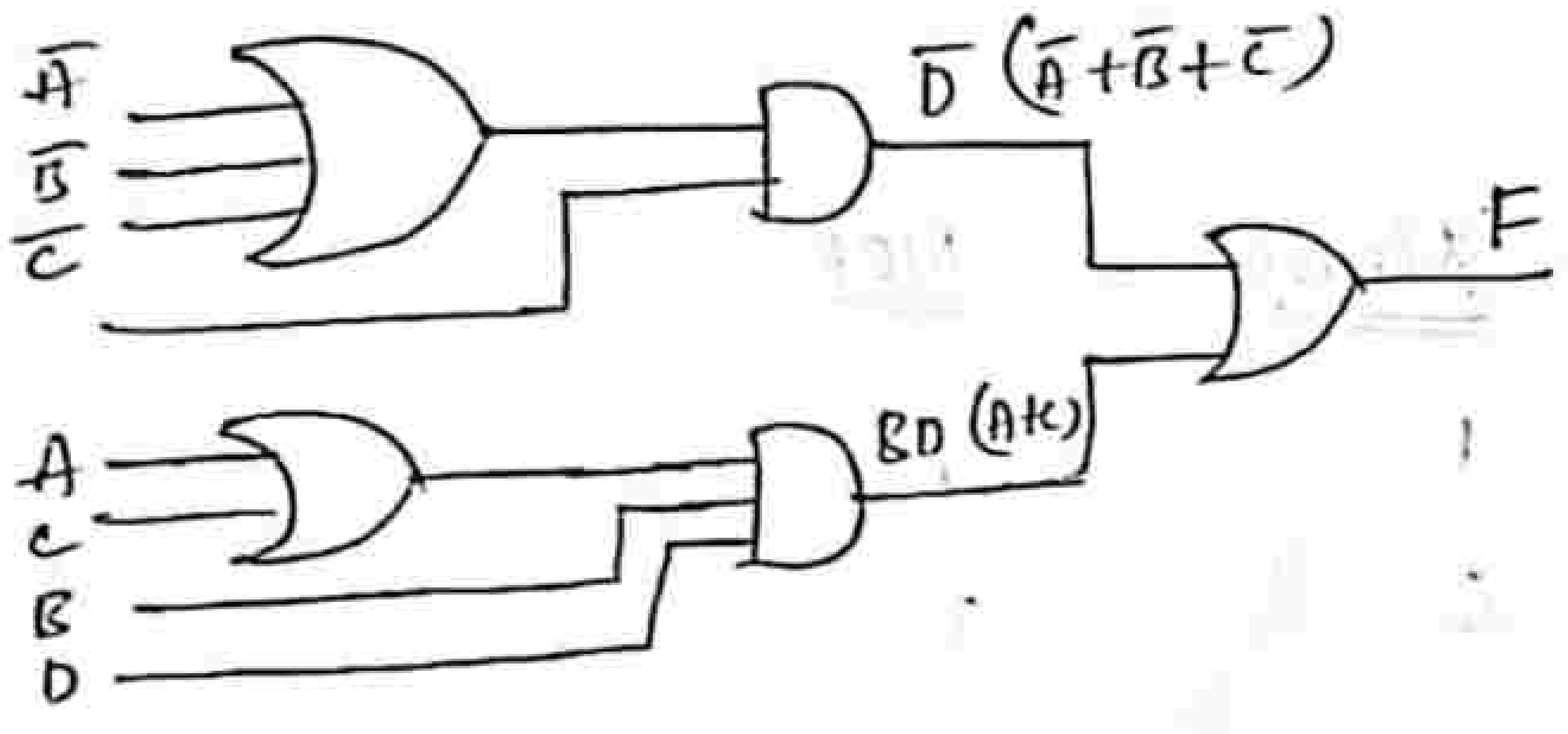
$$F = \Sigma m(0, 2, 4, 6, 7, 8, 10, 12, 13, 15)$$

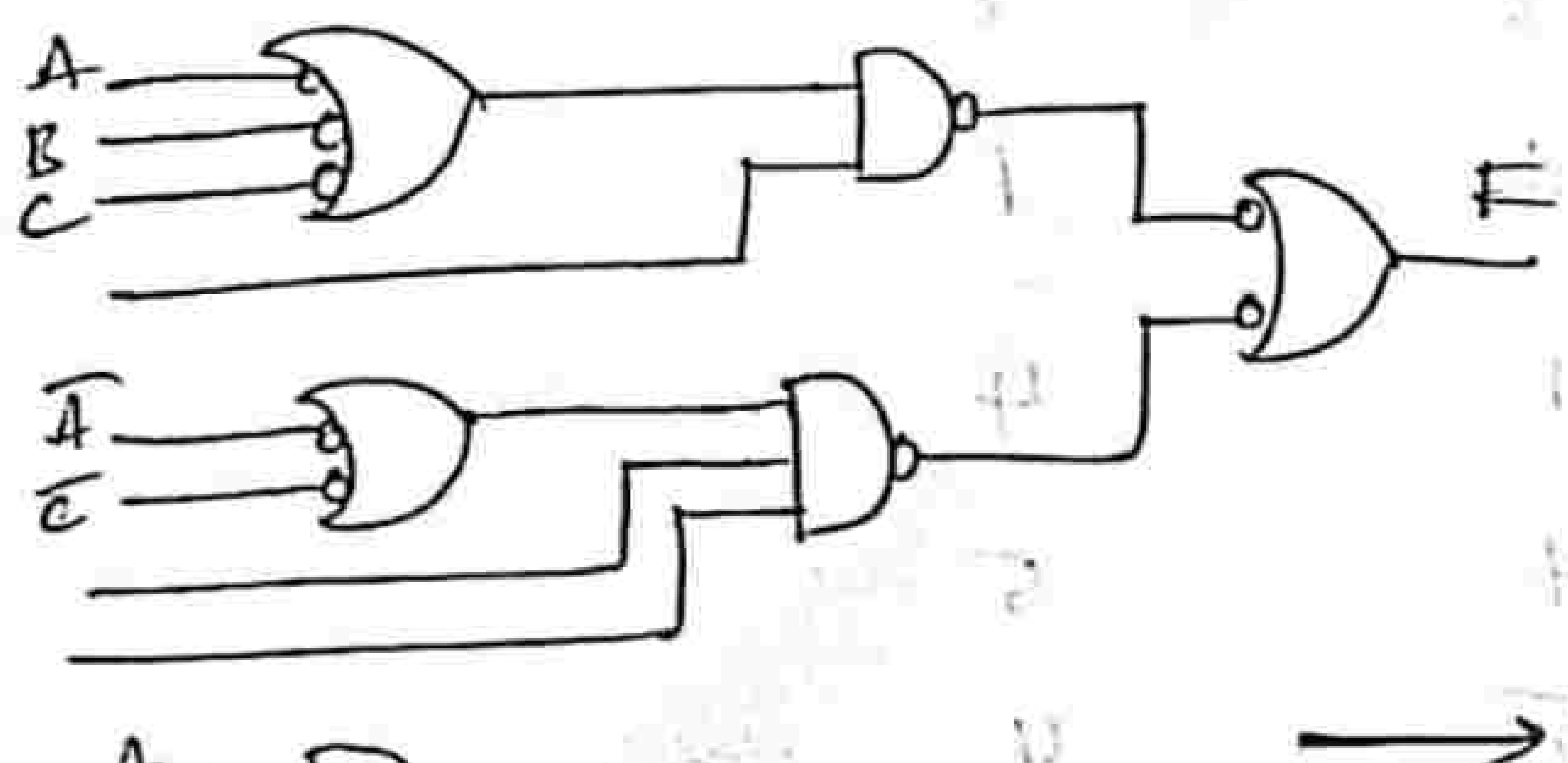
Sol

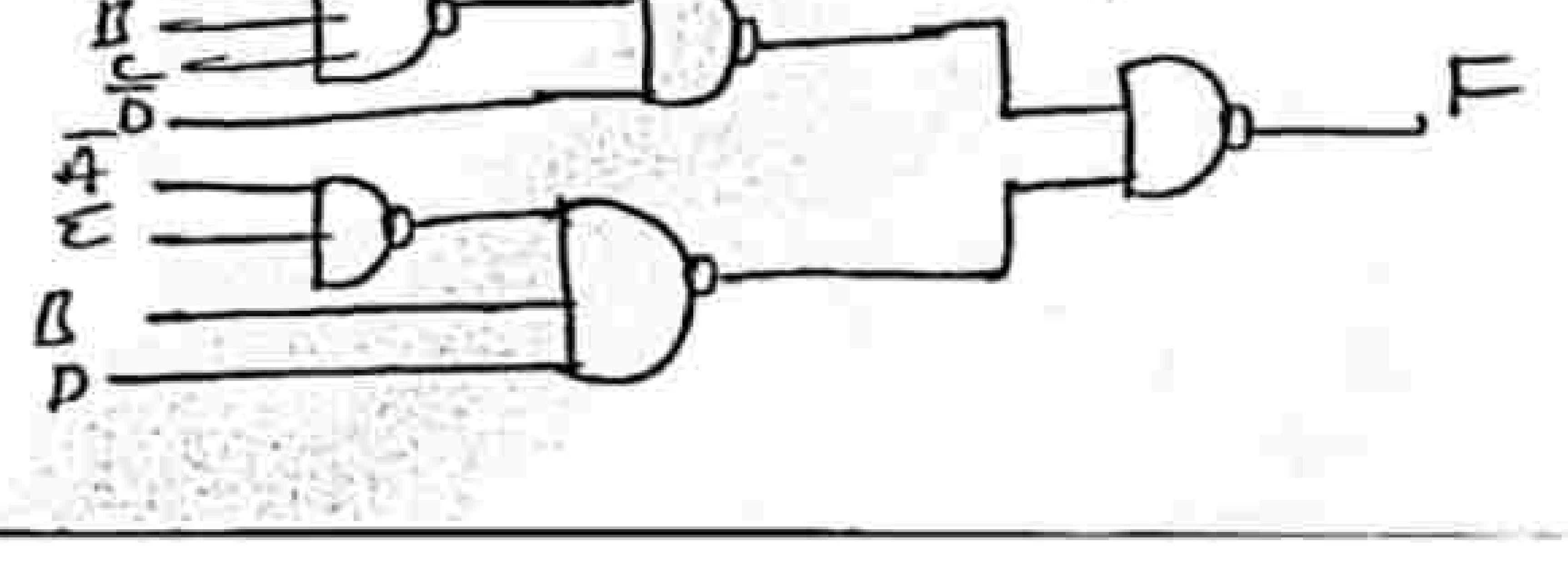


$$F = \bar{C}\bar{D} + \bar{A}\bar{D} + \bar{B}\bar{D} + ABD + BCD$$

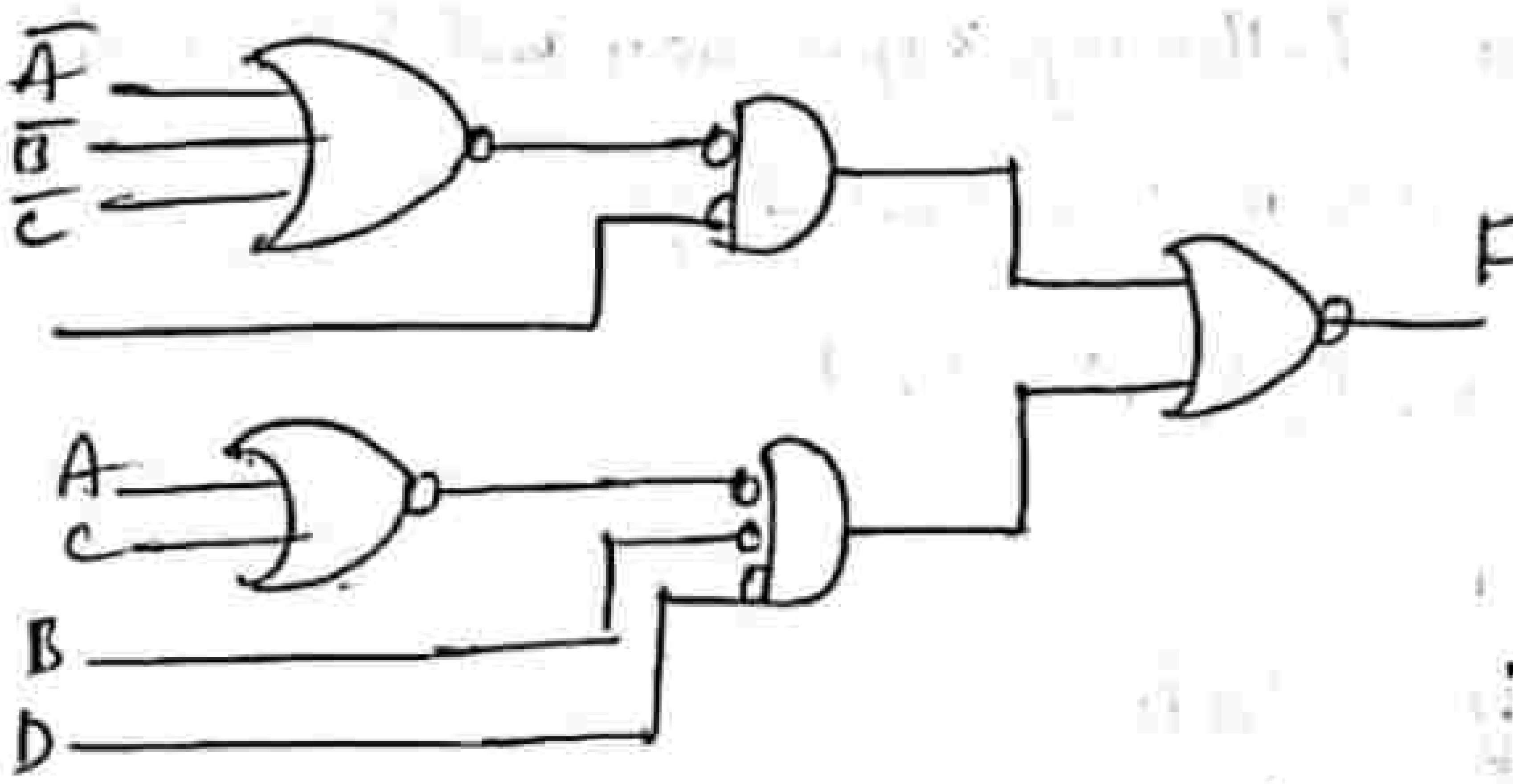
$$= \bar{D}(\bar{A} + \bar{B} + \bar{C}) + BD(A + C)$$

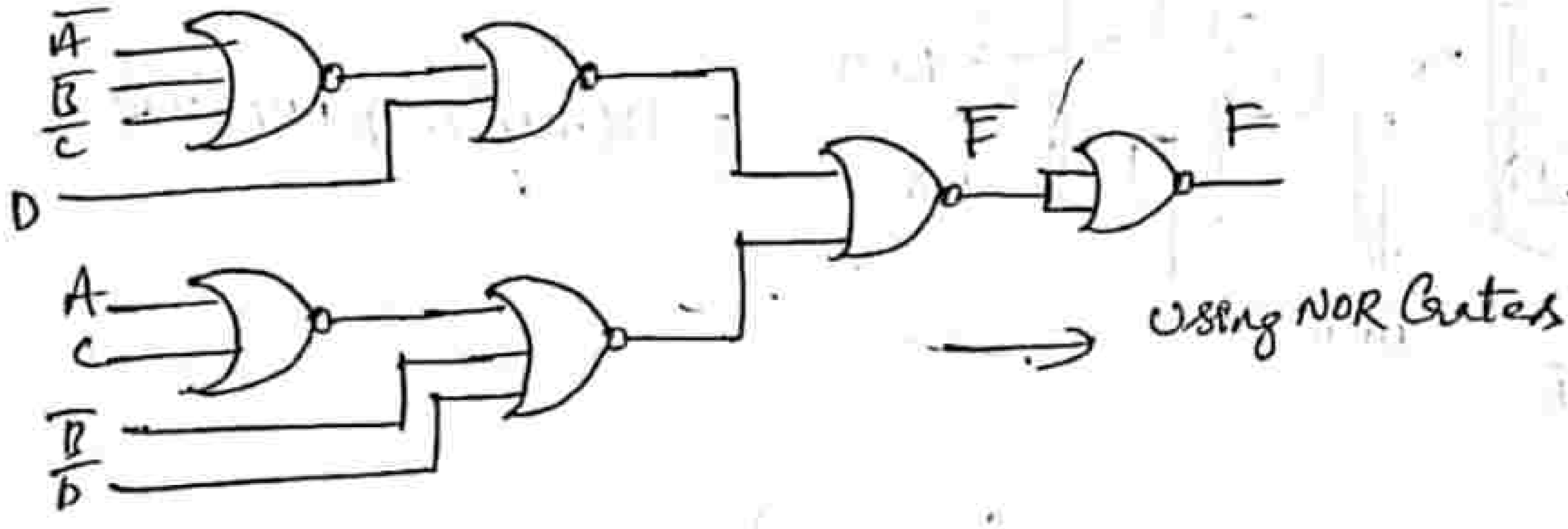






→ (Using NAND gates)



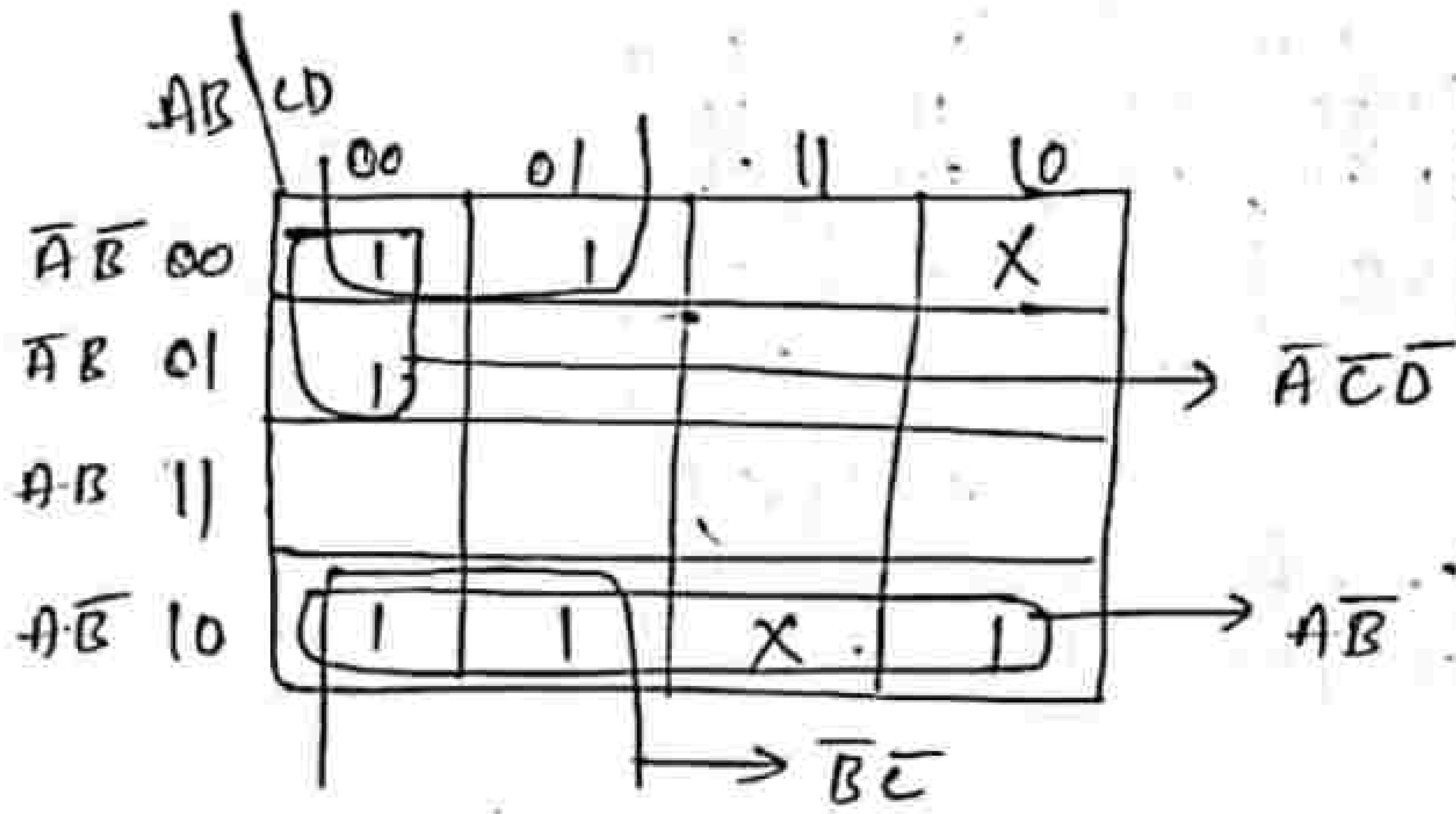


	NAND	NOR
NOT	1	1
AND	2	3
OR	3	2
NOR	4	1
NAND	1	4
X-OR	4	5
X-NOR	5	4

Q. Reduce the following expression using K-map and implement it using NAND gate. (138)

$\pi M(3,5,6,7,12,13,14,15) + d(2,11)$

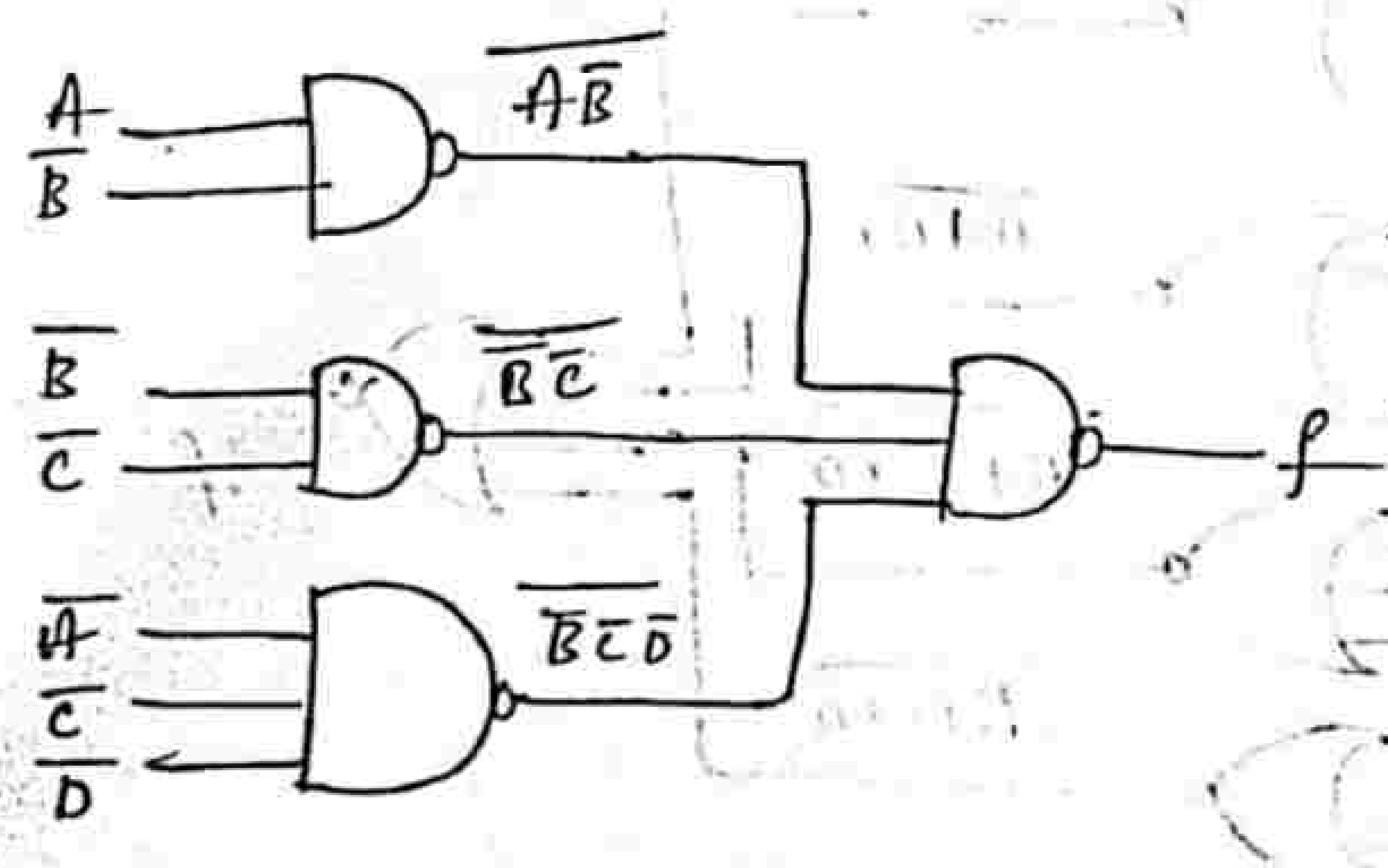
$\Rightarrow \Sigma m(0,1,4,8,9,10) + d(2,11)$



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

W.K.T $f = \overline{\overline{A\bar{B} + \bar{B}\bar{C} + \bar{A}\bar{C}\bar{D}}}$

$f = \overline{\overline{A\bar{B}} \cdot \overline{\bar{B}\bar{C}} \cdot \overline{\bar{A}\bar{C}\bar{D}}}$



Q. Minimize the expression using K-map and implement it using NOR gate (139)

$$f(A,B,C,D) = \Sigma m(1,4,7,10,11,12,13) + \Sigma d(5,14,15)$$

$$= \Pi M(0,2,3,6,8,9) + \Sigma d(5,14,15)$$

		AB \ CD	00 ($C+D$, $\bar{C}\bar{D}$)	01 ($C+\bar{D}$, $\bar{C}D$)	11 ($\bar{C}+\bar{D}$, CD)	10 ($\bar{C}+D$, $C\bar{D}$)
$A+B$	$\bar{A}\bar{B}$	00	0		0	0
$A+\bar{B}$	$\bar{A}B$	01		X		0
$\bar{A}+\bar{B}$	AB	11			X	X
$\bar{A}+B$	$A\bar{B}$	10	0	0		

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

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

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