

Date

K-maps provide systematic method for simplifying Boolean expressions and if properly used, also provide simplest SOP & POS expression possible.

K-maps can be used for expressions with 2, 3 or 4/5 variables but we will discuss only 3/4 variable situations.

① \Rightarrow for 3 variables, 8 cells are required by
2③ \Rightarrow no. of variable \Rightarrow 8 cells.

Now creating cells.

AB \ C	0	1
00		
01		
11		
10		

- Ones are read as true variables
- Zeros are read as complemented variables.

By the
format
of code.
gray 1-bit
(changing
at a time.)

* K-maps simplify the combinational logic by grouping cells and eliminating variables that change.

Scenario-based-understanding:-

Let random values on the T.T in which have 2 groups (1 vertical & 1 horizontal)

→ P.T.O

Date

Now, the horizontal group C changes from 0 \rightarrow 1 and $\bar{A}B$ remains same.

AB	C	0	1
00	(1)	0	
01	(1)	1	
11		0	0
10		0	0

Also Now the vertical group; B changes from 0 \rightarrow 1 so its eliminated and C remains same.

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

(11) \Rightarrow 4-variable map

(its)

It has adjacent cell on each of the four boundaries;

AB	CD	00	01	11	10
00	(1)				
01		(1)			
11			(1)		
10				(1)	

→ Each of the cell is diff. only by one variable from an adj. cell.

→ Grouping follows the rules given in the text.

→ Read each group by eliminating any variable that changes across a boundary.

group 2.

Group 1 is in two part, supposedly across boundaries.

Group 2 is in 1 part with 4 entries.

For group 1

For group 2.

B & C changes across boundary. ($=$)

$$\therefore X = \bar{A}\bar{D} + \bar{A}D$$

Date

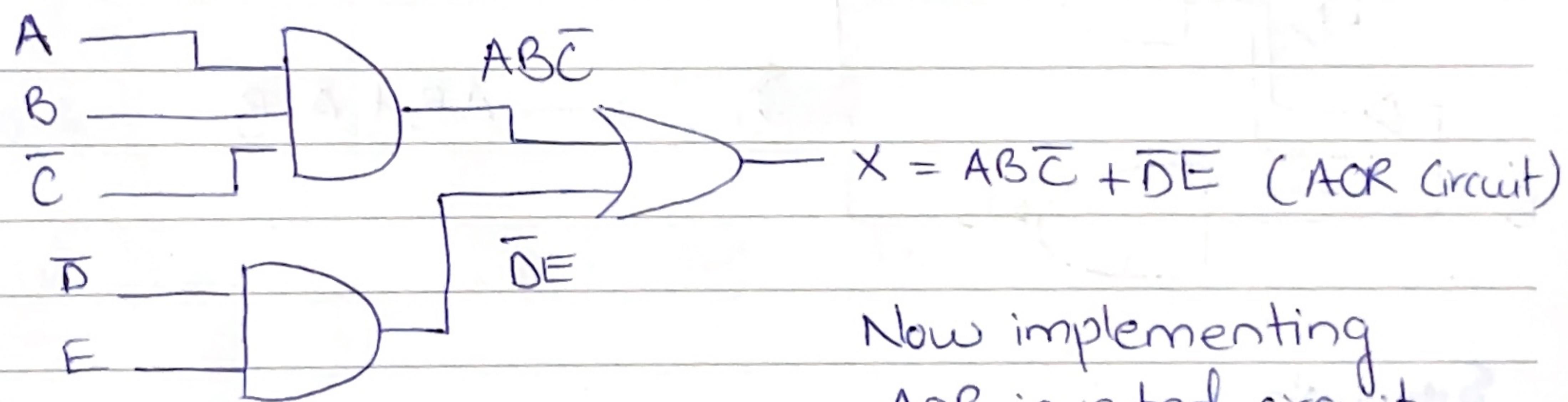
COMBINATIONAL LOGIC:-

Combination of logic gates to produce a specified output for certain specified combinations of input variables.

SOP & POS are basic forms of combinational logic.
↳ (AND-OR combination)

When output of SOP form is inverted, the circuit is called AND-OR Inverted Circuit. AOI configuration lend itself to product of sum (POS) implementation.

Example:-



Now implementing
AOR inverted circuit.

$$\begin{aligned} &= \overline{ABC} + \overline{DE} \\ &= (\overline{ABC})(\overline{DE}) \text{ Demorgan's} \\ &= (\overline{A} + \overline{B} + C)(D + \overline{E}) \text{ POS.} \end{aligned}$$

Date _____

L EXCLUSIVE OR. (XOR)

The output is high when both A & B disagree. T.T:-

Boolean Expression.

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

inputs

Input output

A B X

0 0 0

0 1 1

1 0 1

1 1 0

Step 1.

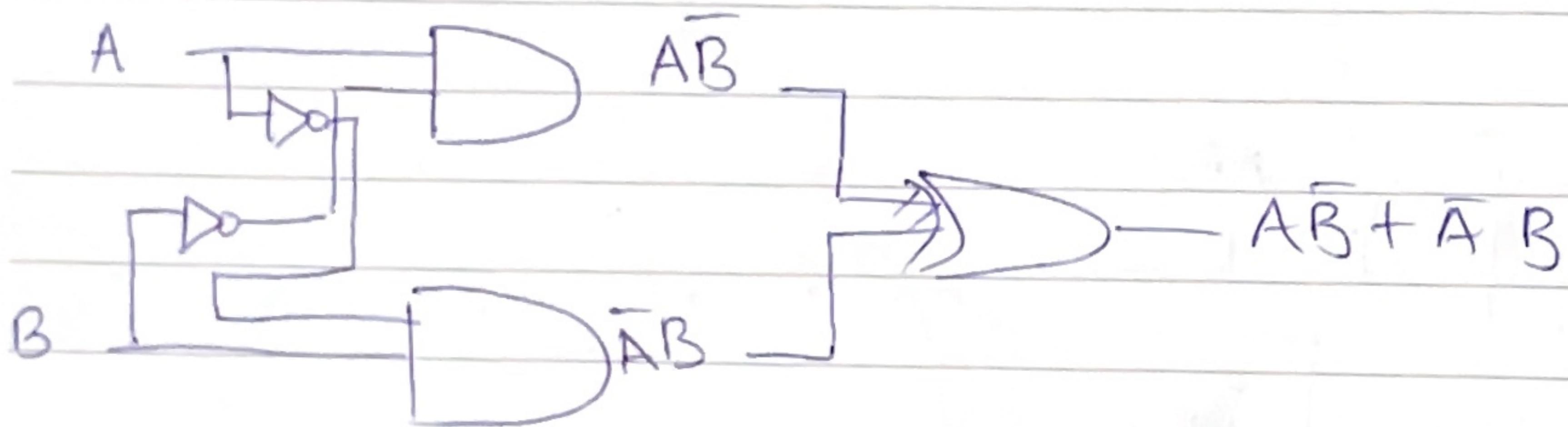
form a TT.

Step 2.

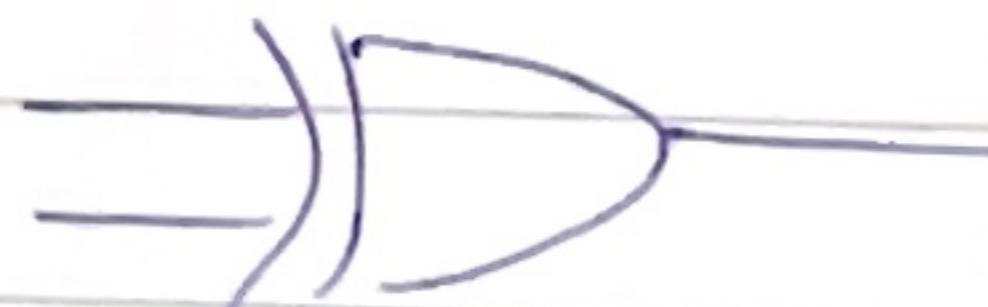
Write the product term for each output 1

Step 3.

Combine all product terms with OR.



Symbols :-



- (i) Distinctive shape.
- (ii) Rectangular outline.



Date

EXCLUSIVE (XNOR)

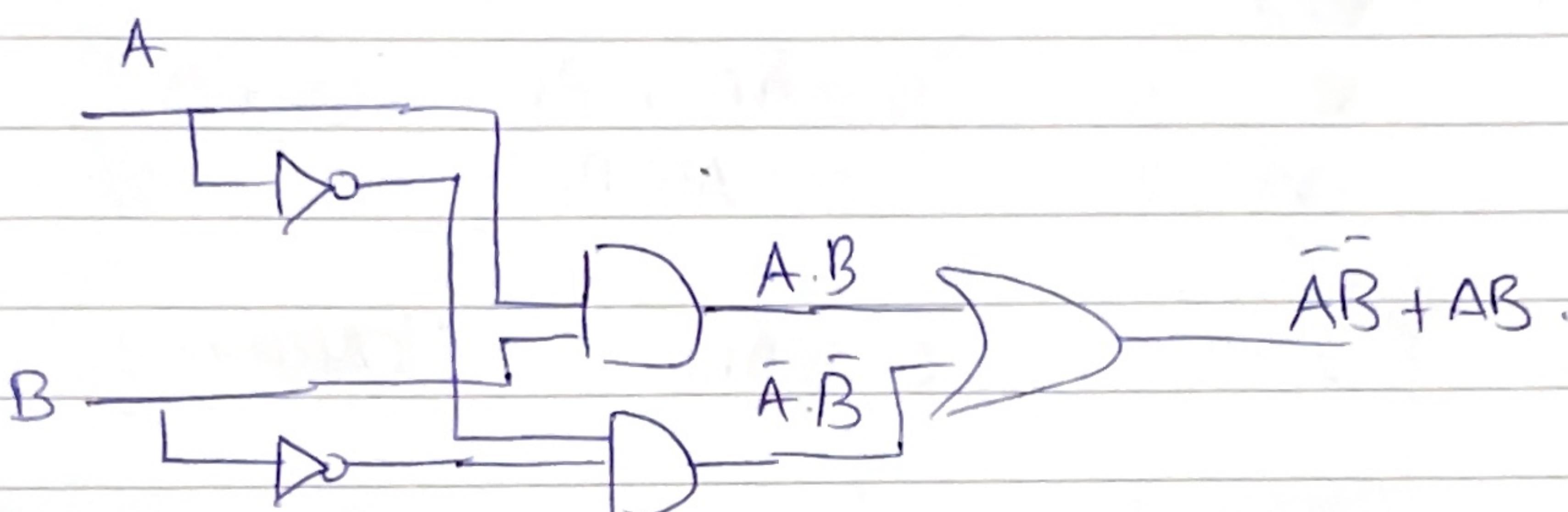
The output is high when both inputs agree.

Truth Table :-

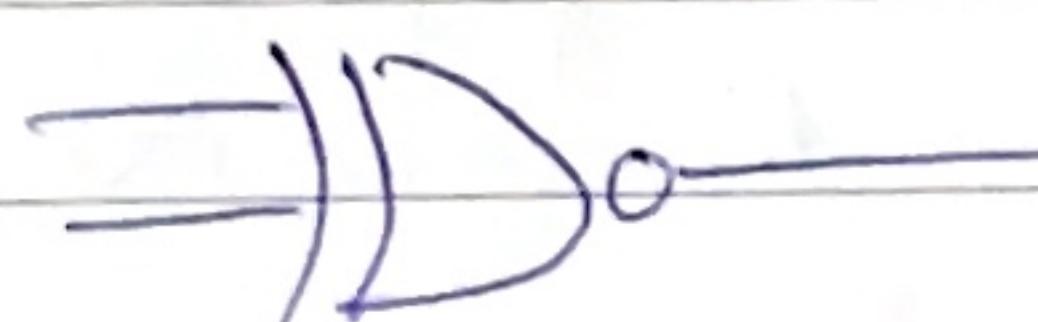
A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

Boolean Expression .

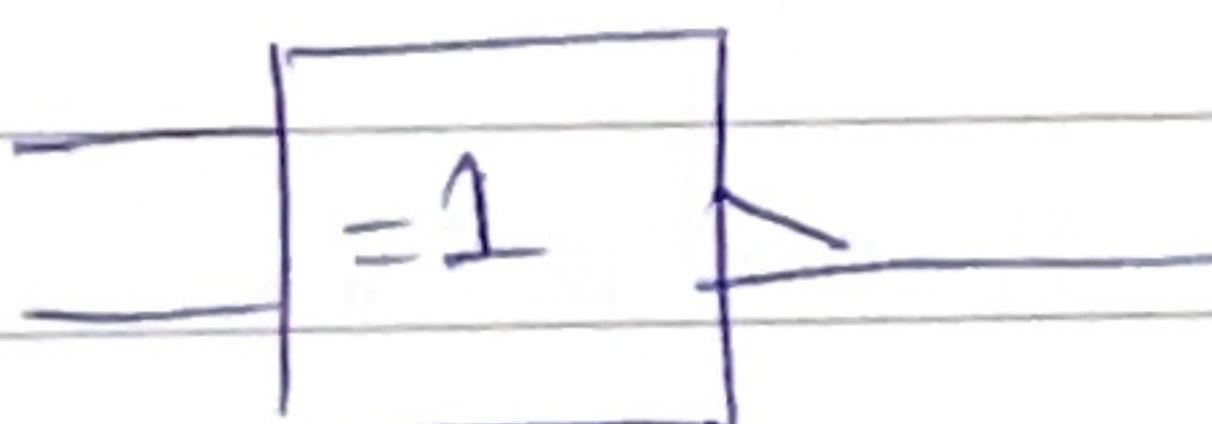
$$X = \bar{A}\bar{B} + AB$$



Symbols .



(i) Distinctive Symbol.



(ii) Rectangular Outline.

TOP

Date A=B :

A LB : $\bar{A}B$ Magnitude of Bit Compared.

A B A-B A>B A<B .

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

A>B: $A\bar{B}$

0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
0	0	1	0	0

K-MAP

GROUPING PRACTICES & RULES:-

(RULES)

(i) MAXIMIZE the size of the group but it should be in order of 2's exponent (i.e 2^n).

(ii) Minimize the no. of groups.

(iii) 'Don't Care' can be used in group but don't use it needlessly. — (a).

* NOTE :- 'DON'T CARE' is not our first Priority.

Analyze the correct use of don't care to follow every Rule.

1	1	1	
1	1	1	X
1	1	1	X
X	X		

Correct.

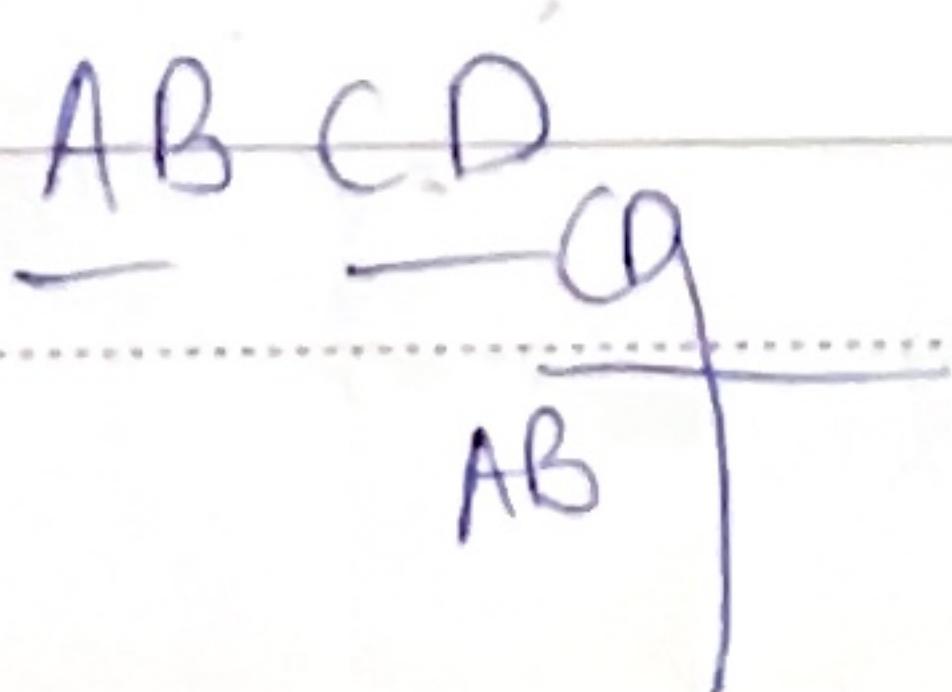
1	1	1	
0	1	1	X
1	1	1	X
X	X		

Incorrect.

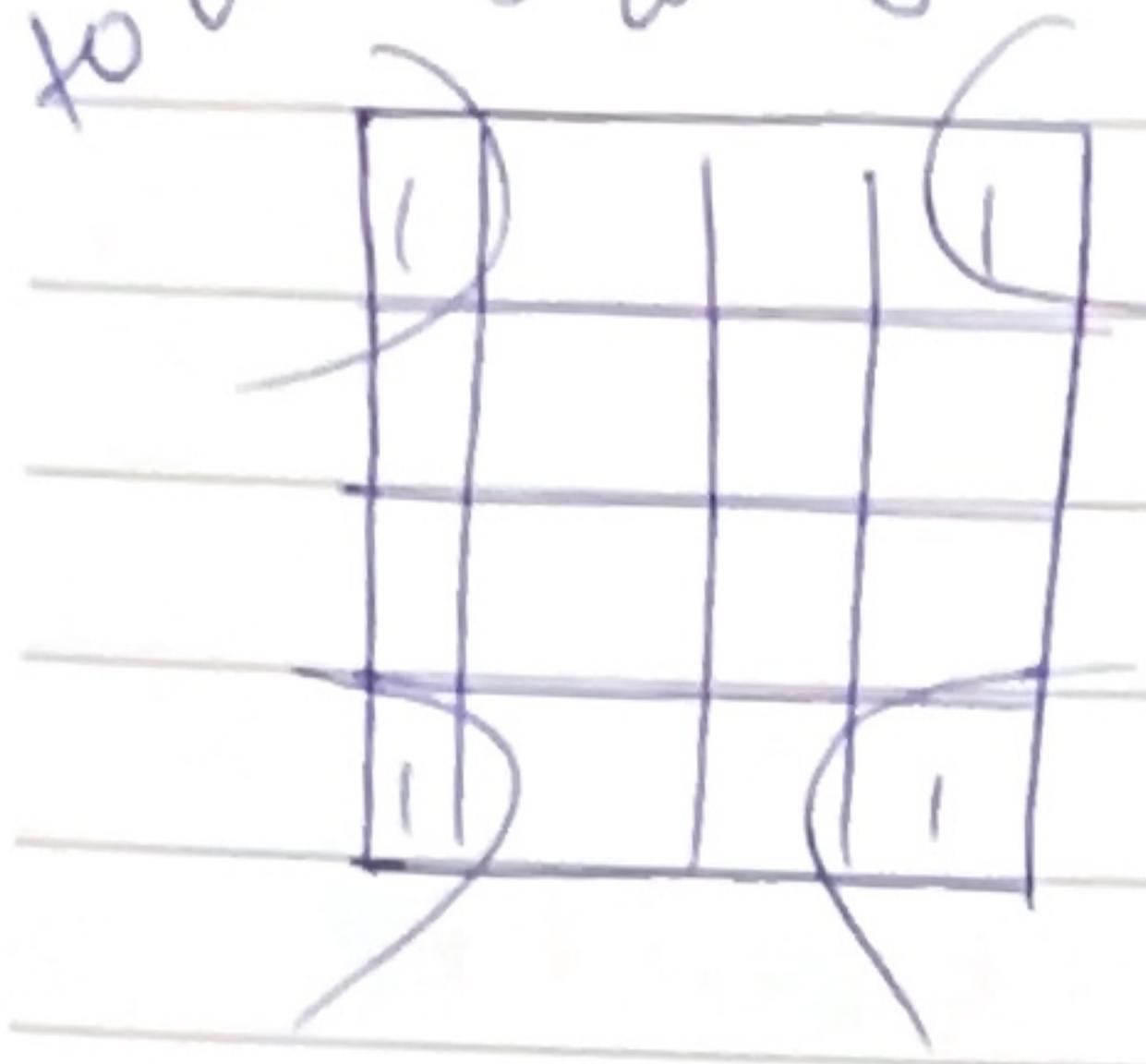
* for filling k-maps;

Start with the most Significant Bit (MSB)

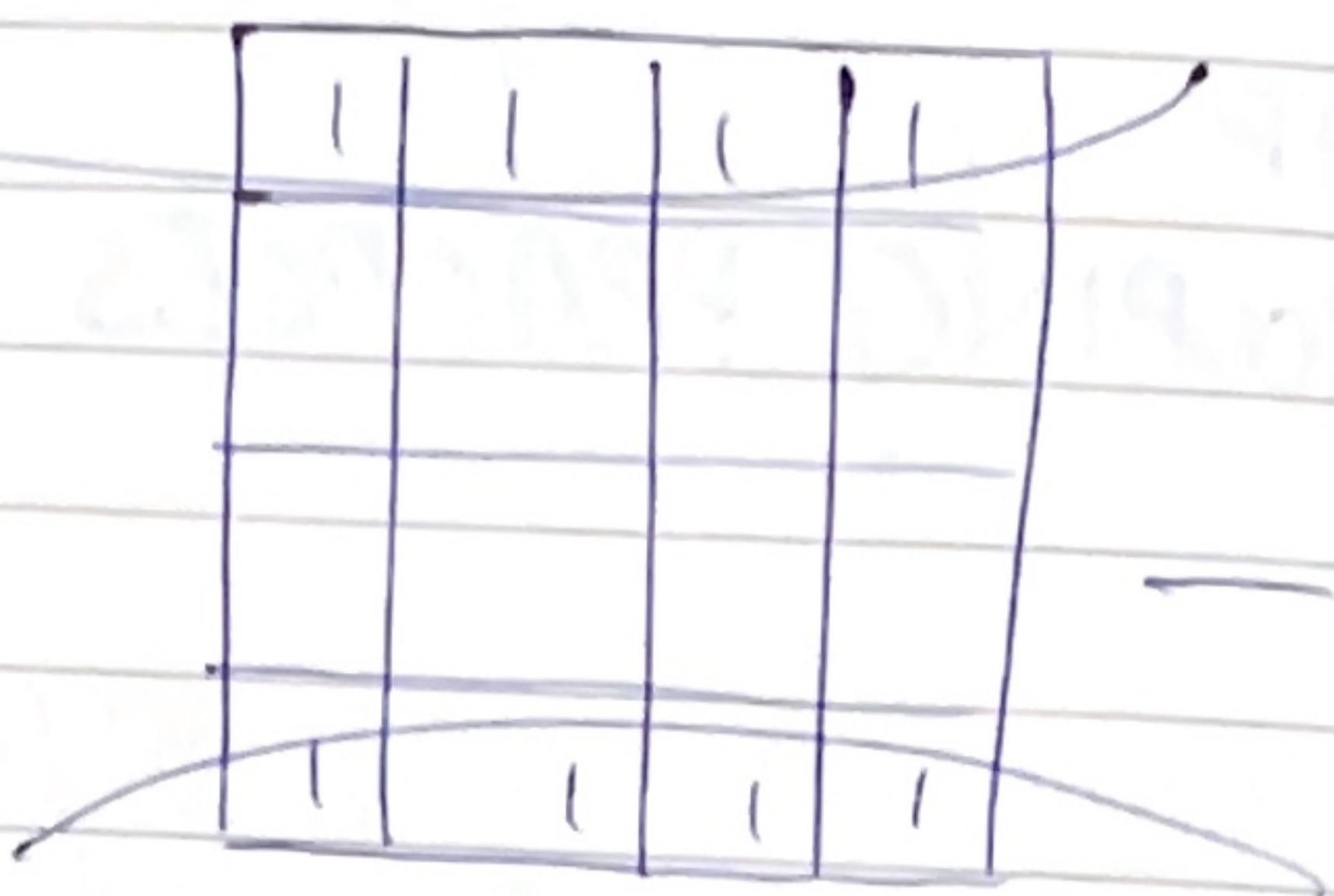
2nd to MSB, 2nd LSB, LSB.
last



Remember
to use gray lock
along axis.

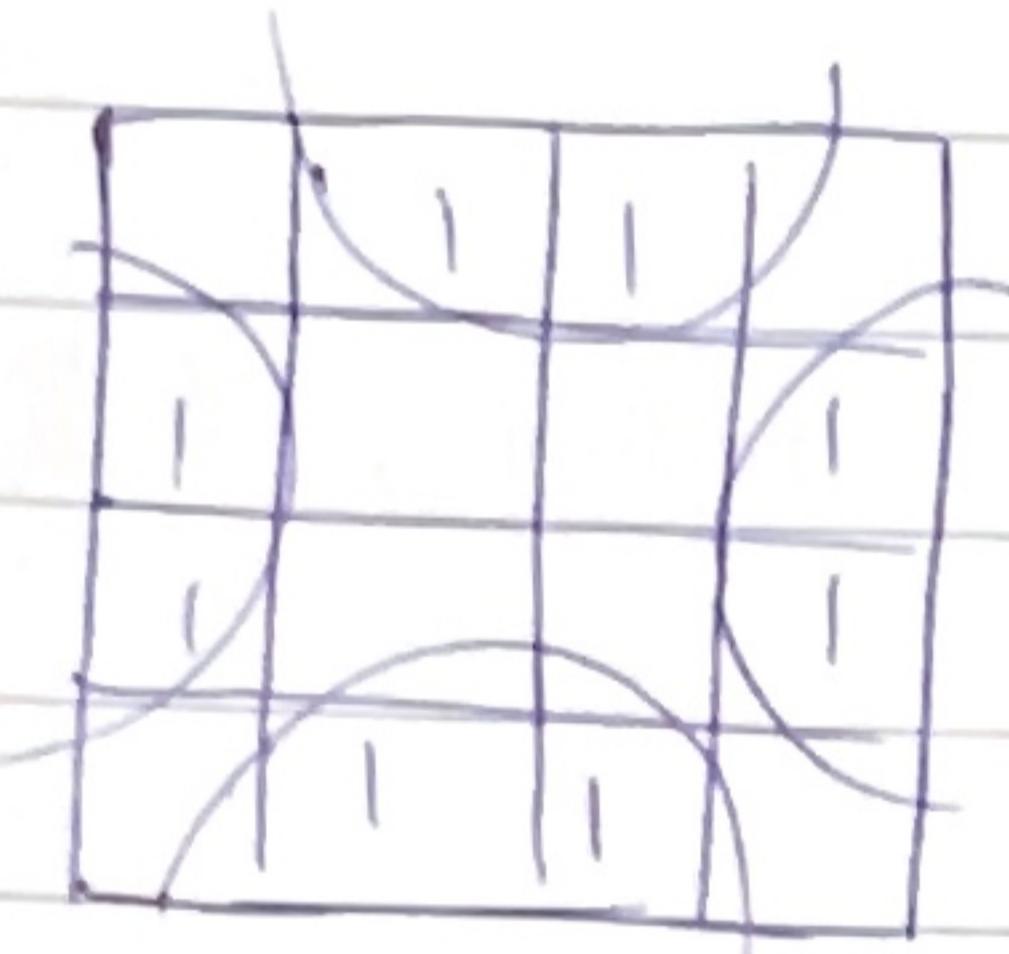


—?

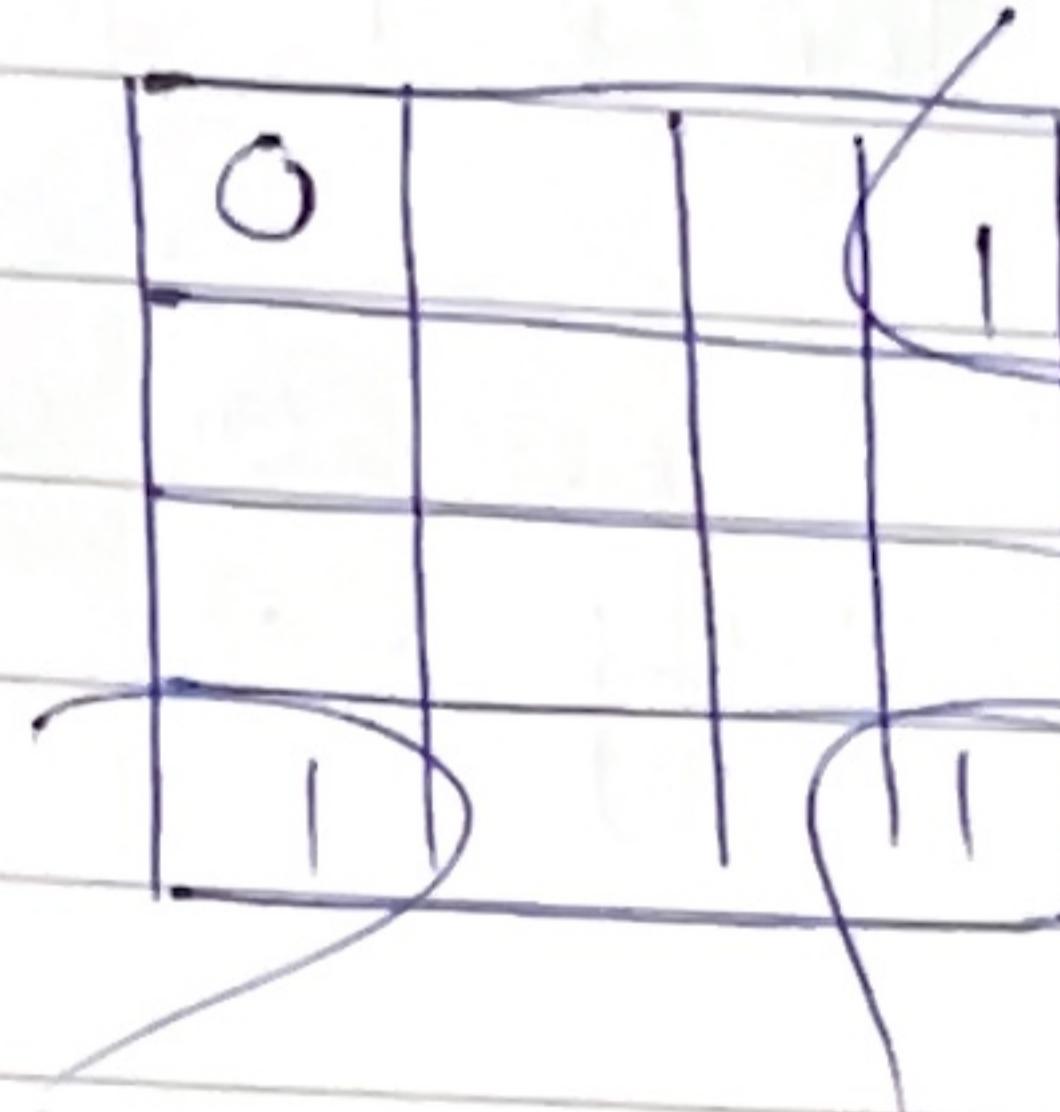


—?

Is it 1 group?

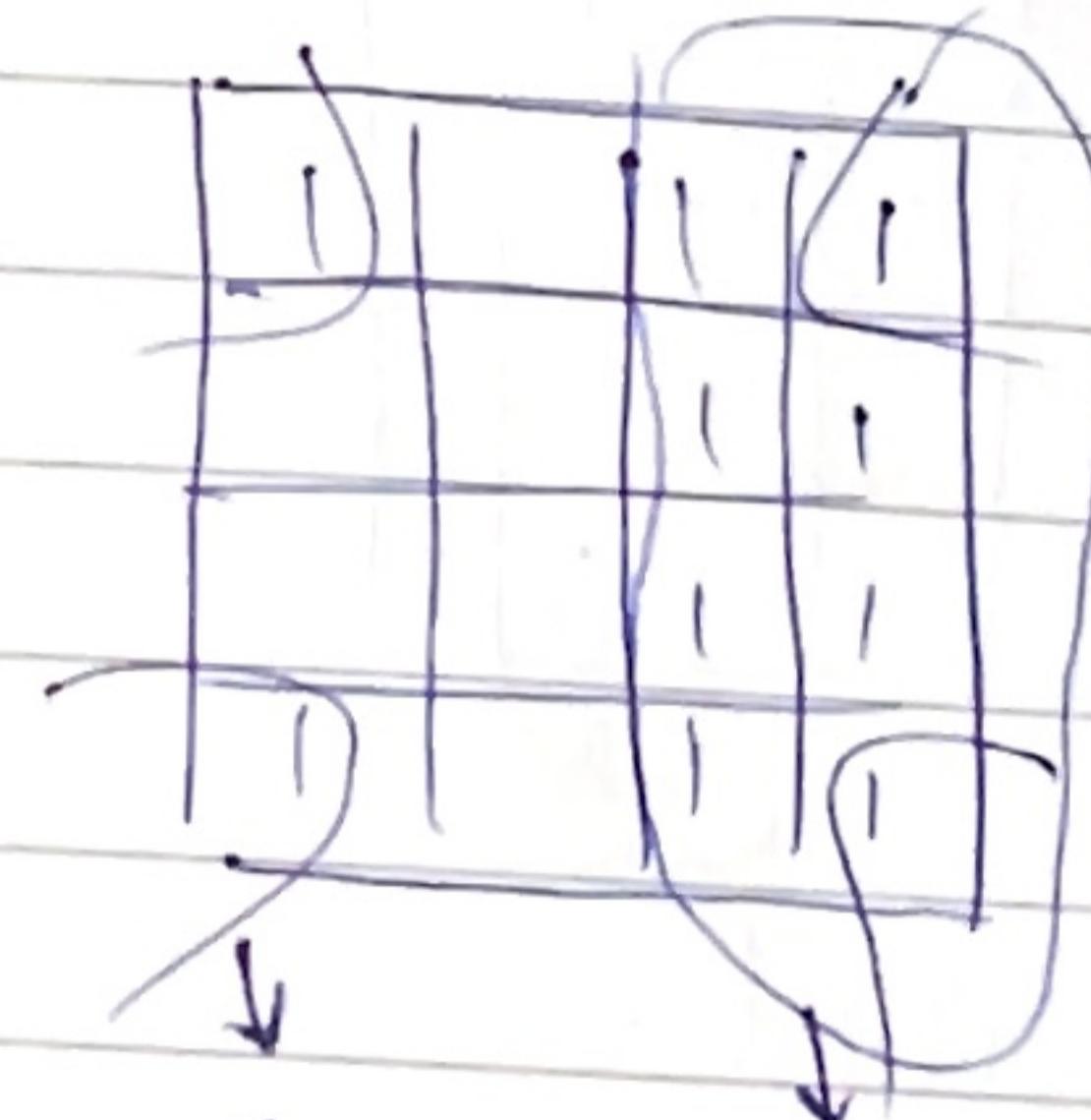
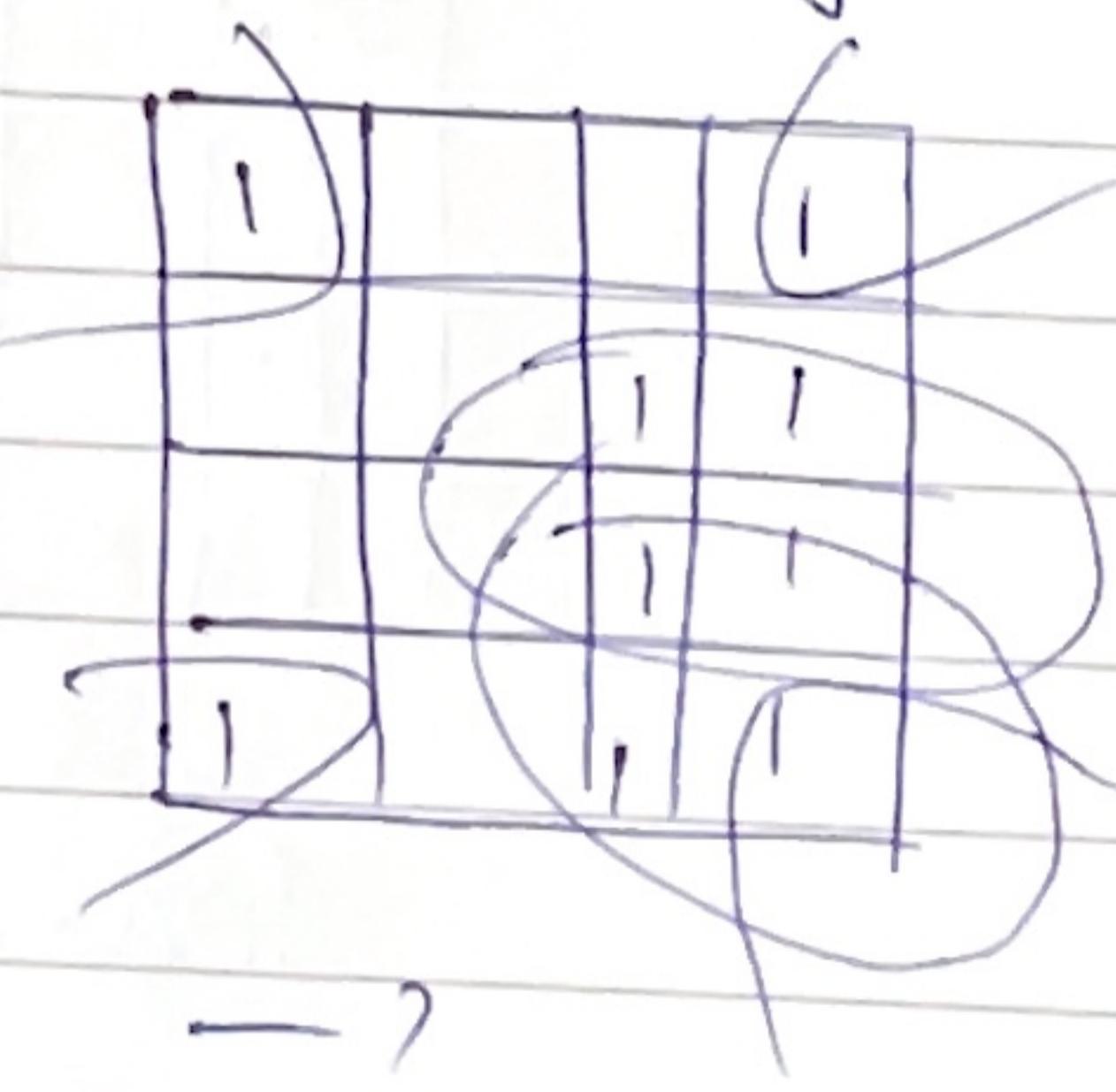
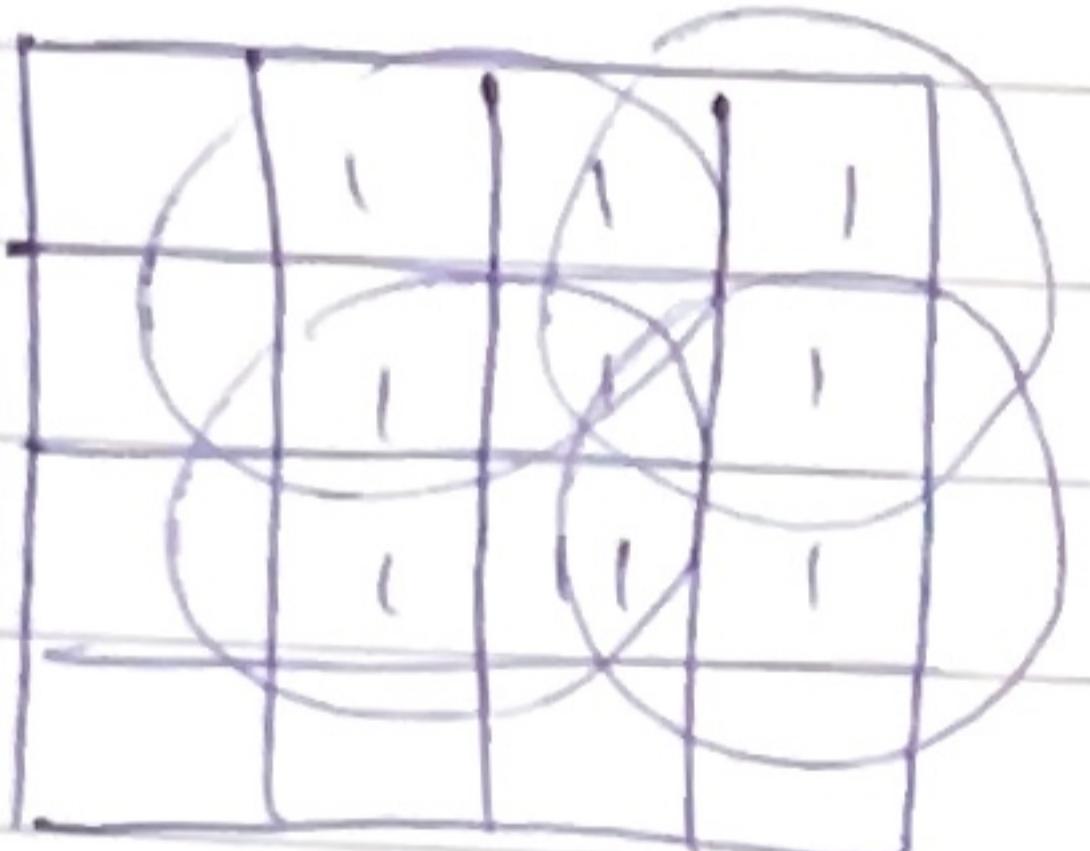


—?



—?

* Grouping cannot be done diagonally.

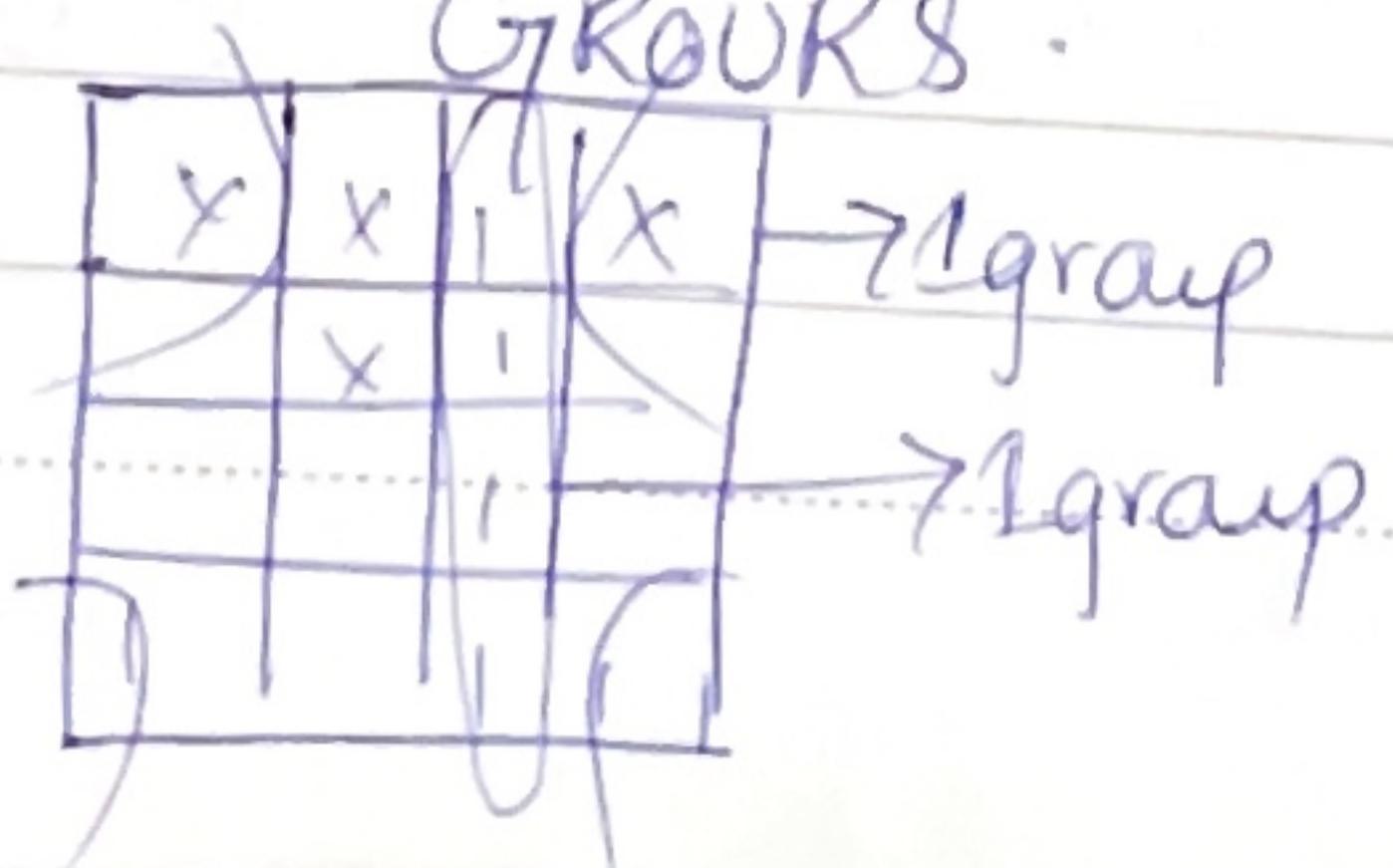


1 group? 1 group.

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

8 groups.

* DON'T CARE CAN BE USED TO
GROUP IS OR MAXIMIZE NO. OF
GROUPS.



Reference
to Original Notes



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

$$Cout \rightarrow (A+B)Cin + AP.$$

Cin	A	B	Σ	Cout
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

Cin/A	B	0	1
00	0	0	1
01	0	1	1
11	1	1	1
10	0	1	1
01	1	0	1

Cin/A	B	0	1
00	0	0	1
01	0	1	0
11	0	0	1
10	1	0	0

$$\Sigma = \overline{A}B\overline{C}in + \overline{A}\overline{B}\overline{C}in + ABC + \overline{ABC}in.$$

