CHAPTER: 5

Combinational Logic Analysis

BASIC COMBINATIONAL LOGIC CIRCUITS :-

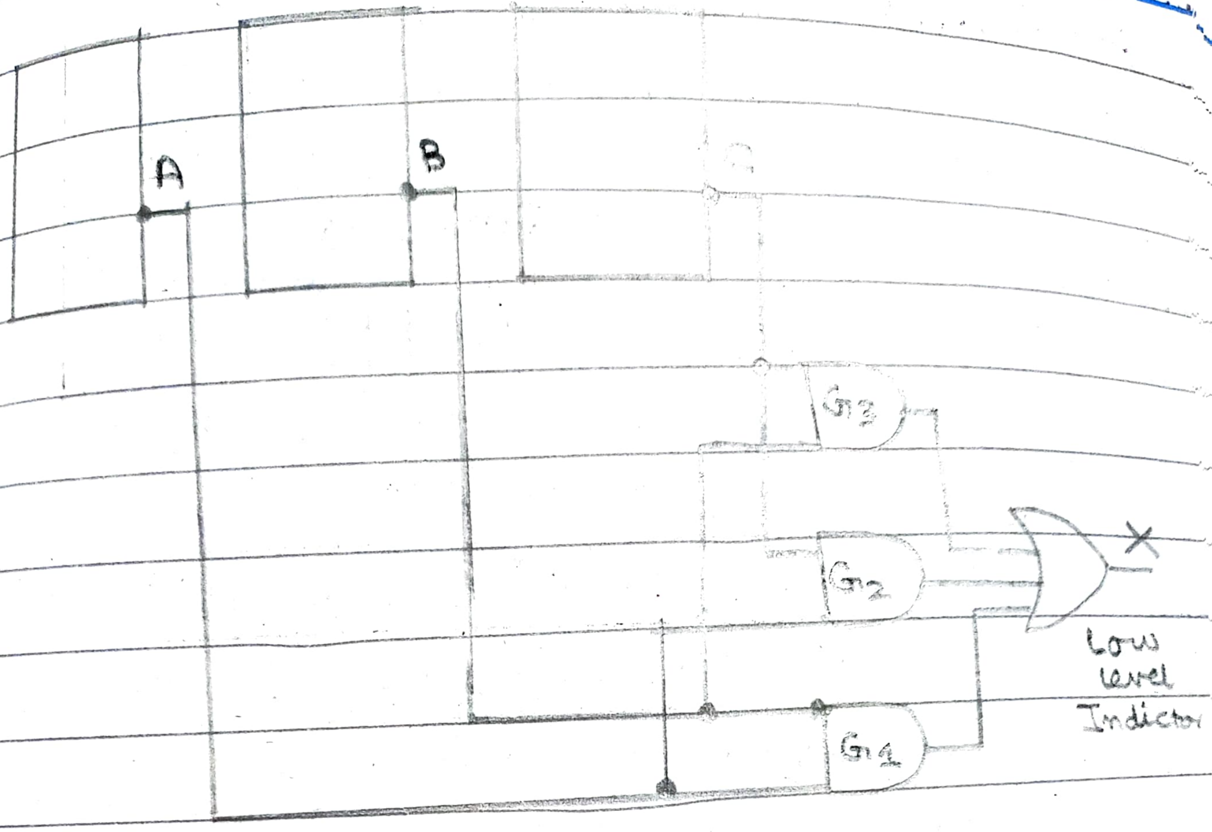
Example 5-1

In a certain chemical processing plant, a liquid chemical is used in a manufacturing process. The chemical is stored in three different tanks. A level sensor in each tank produces a HIGH voltage when the level of chemical in the tank drops below a specified point.

Design a circuit that monitors the chemical level in each tank and indicates when the level in any two of the tanks drops below the specified point.

Solution:

The AND-OR circuit in figure has inputs from sensors on tanks A, B and C. The AND gate G_1 checks the levels in tanks A and B, gate G_2 checks tanks A and C, and gate G_3 checks tanks B and C. When the chemical level in any two of the tanks gets ~~too~~ too low, one of the AND gates will have HIGHs on both of its inputs, causing its output to be HIGH and so the final output X from the OR gate is HIGH. This HIGH input is then used to activate an indicator such as lamp or audible alarm,



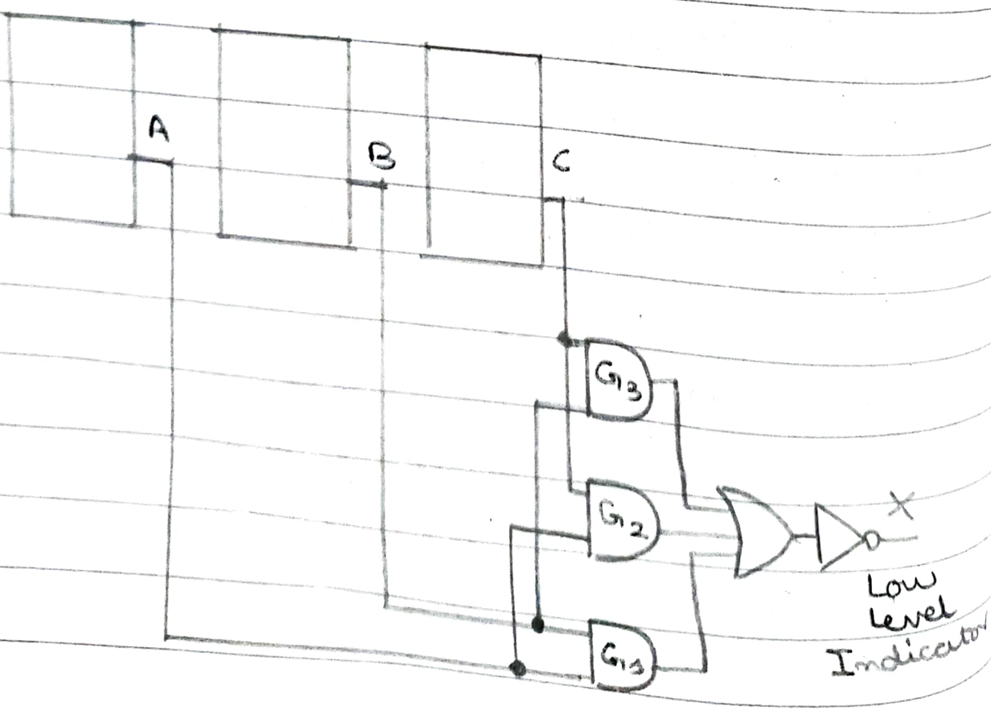
Example 5-2

The sensors in the chemical tanks of example 5-1 are being replaced by a new model that produces a LOW voltage instead of a HIGH voltage when the level of the chemical in the tank drops below a critical point.

Modify the circuit to operate with the different input levels and still produce a HIGH output to activate the indicator when the level in any two of the tanks drops below the critical point. Show the logic diagram.

Solution :

The AND-OR invert circuit has inputs from the sensors on tanks A, B and C. The AND gate G1 checks the levels in tanks A and B, gate G2 checks tanks A and C and gate G3 checks tanks B and C. When the chemical level in any two of the tanks gets too low, each AND gate will have a LOW on at least one input, causing its output to be LOW and thus the final output X from the inverter is HIGH. This HIGH output is then used to activate an indicator

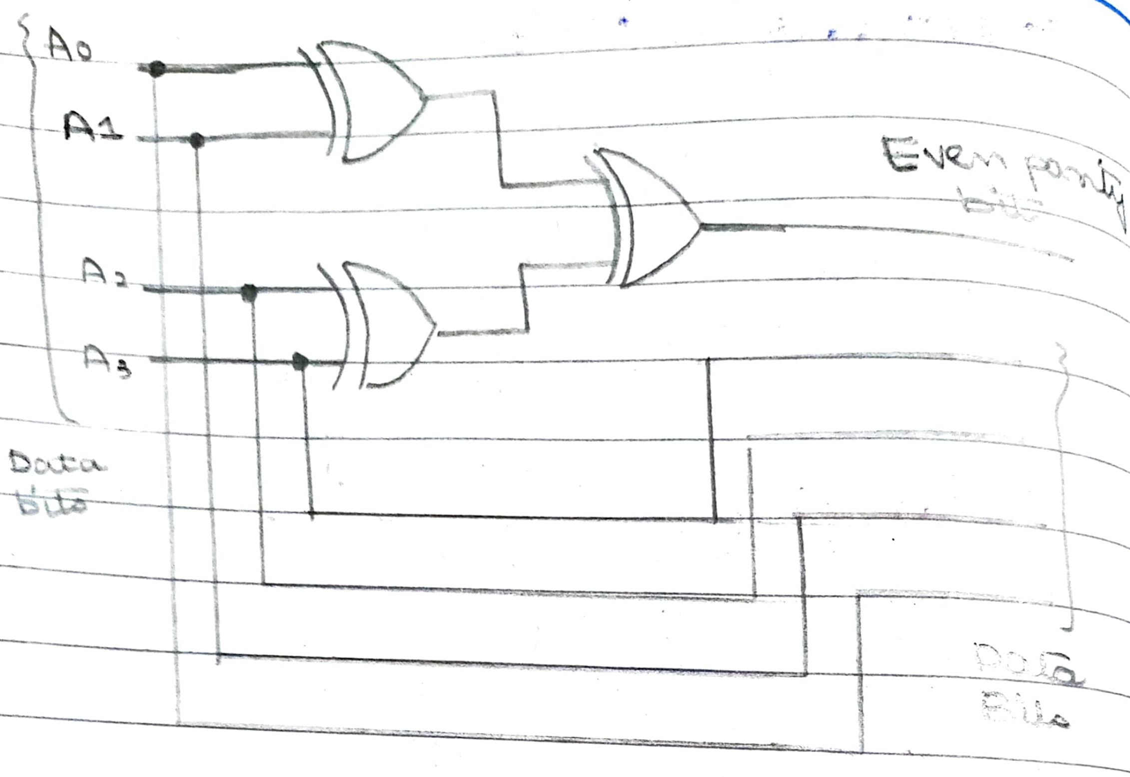


Example 5-3

Use exclusive-OR gates to implement an even-parity code generator for an original 4-bit code.

Solution:

A parity bit is added to a binary code in order to provide error detection. For even parity, a parity bit is added to the ~~so~~ original code to make the total number of 1s in the code even. The circuit in figure produces a 1 output when there is an odd number of 1s on the inputs in order to make the total number of 1s in the output code even. A 0 output is produced when there is an even number of 1s on the inputs.

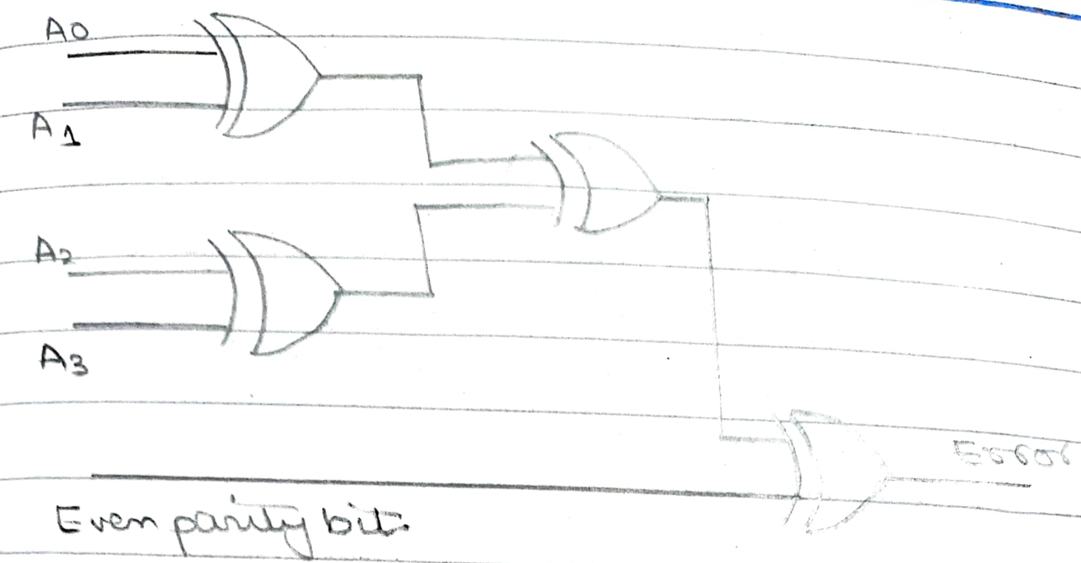


Example 5-4

Use exclusive-OR gates to implement an even parity checker for the 5-bit code generated by circuit in example 5-3.

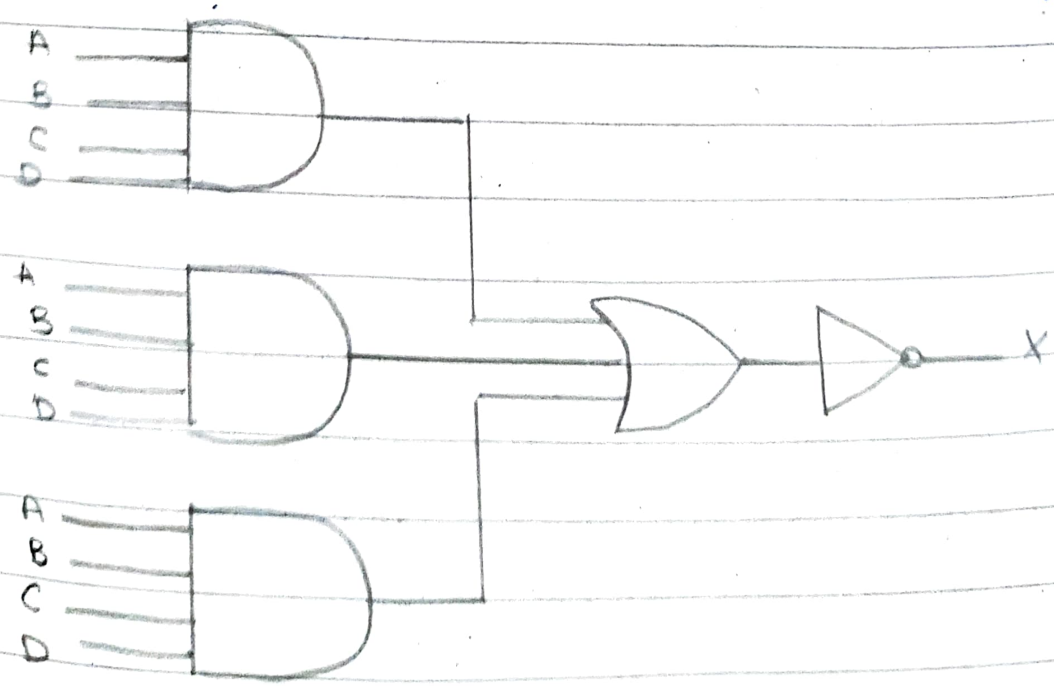
Solution:

The circuit in figure produces a 1 output when there is an error in the five-bit code and a 0 when there is no error.

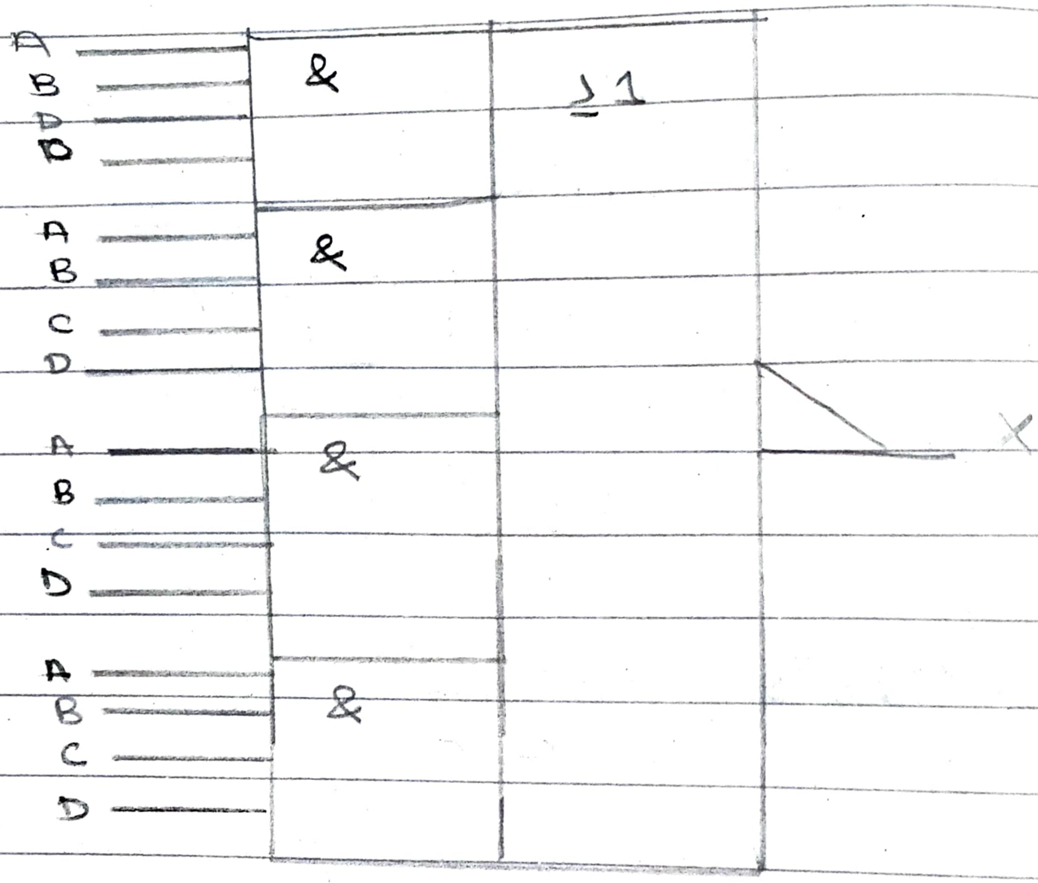


Section 5-1:

1. Draw ANSI distinctive shape logic diagram for 4-wide, 3-input AND-OR-Invert circuit. Also draw the ANSI standard rectangular online symbol.

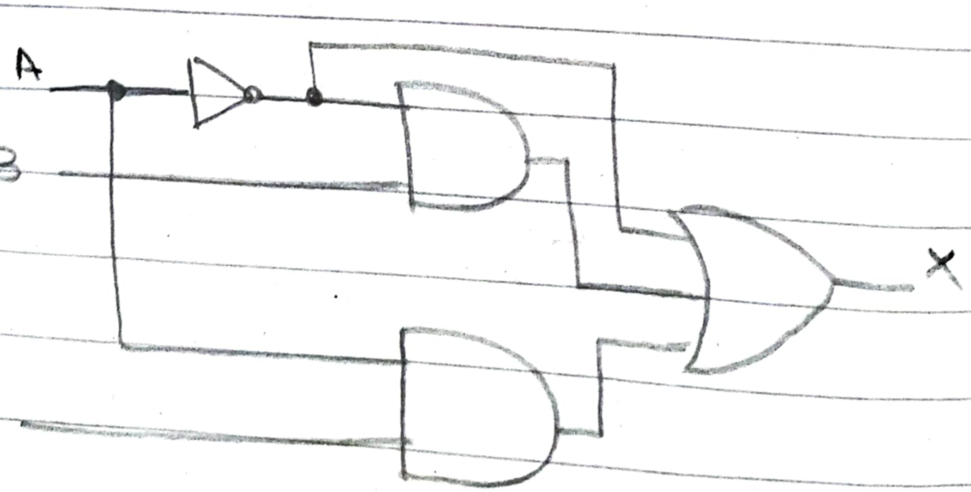


Rectangular outline.



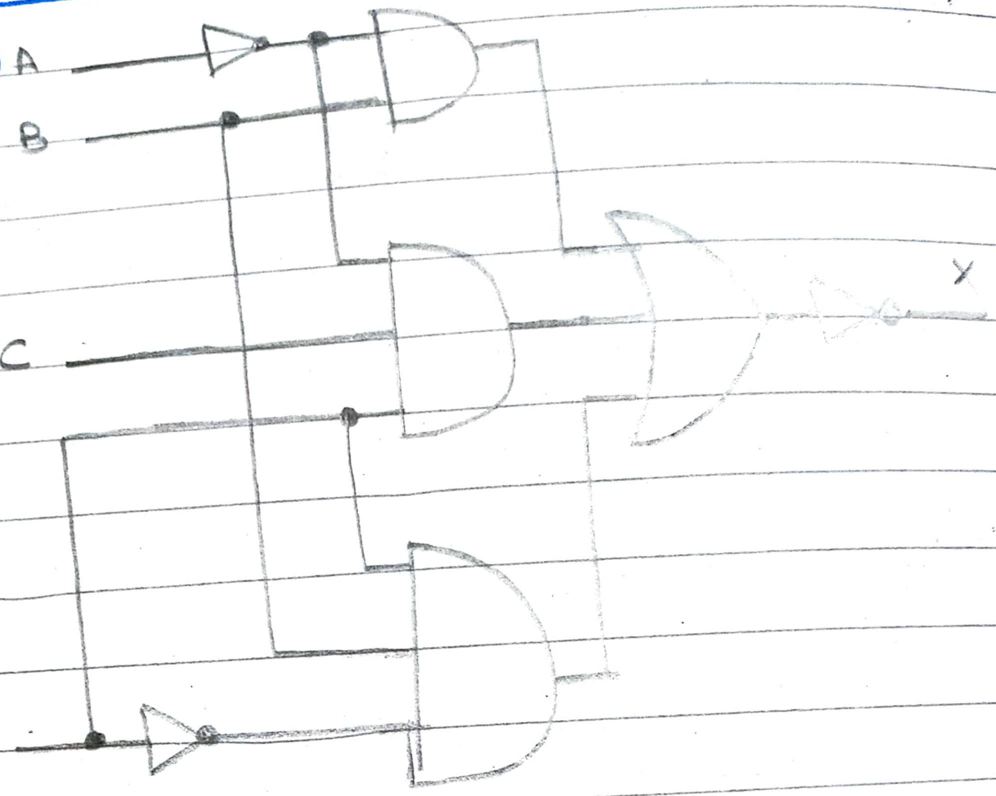
2. Write output expression for each circuit.

a)



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

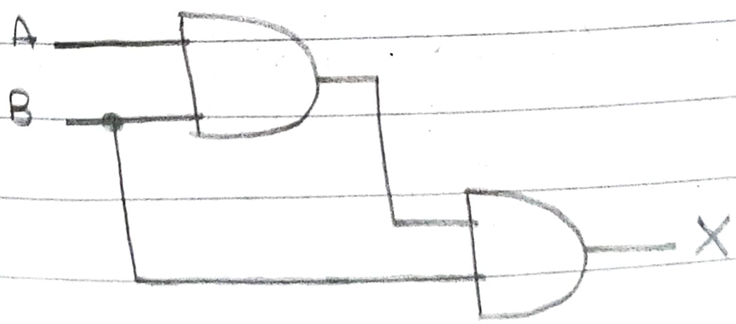
b)



$$X = \overline{\overline{A}B + \overline{A}CD + DB\overline{D}}$$

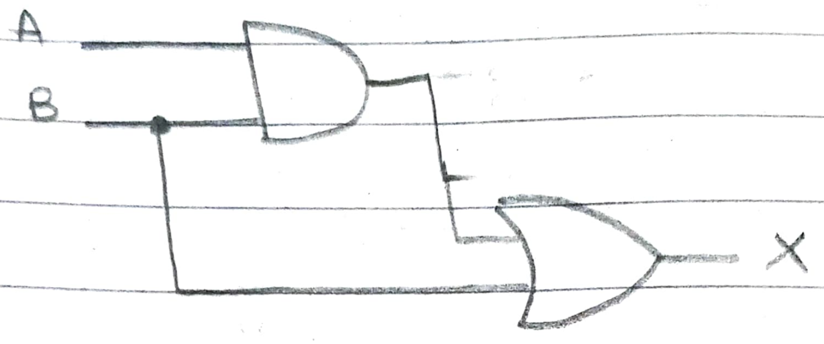
3. Write output expression for each circuit as it appears.

a)



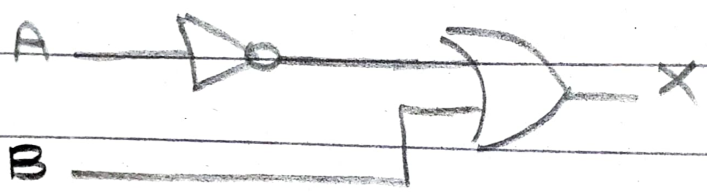
$$X = ABB$$

b)



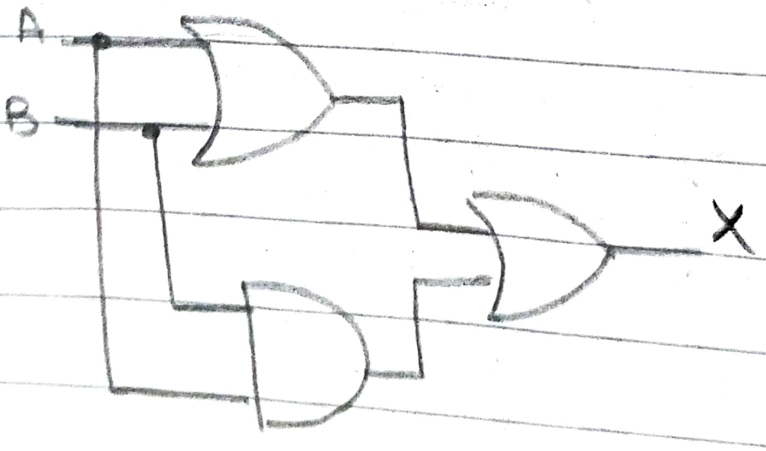
$X = AB + B$

c)



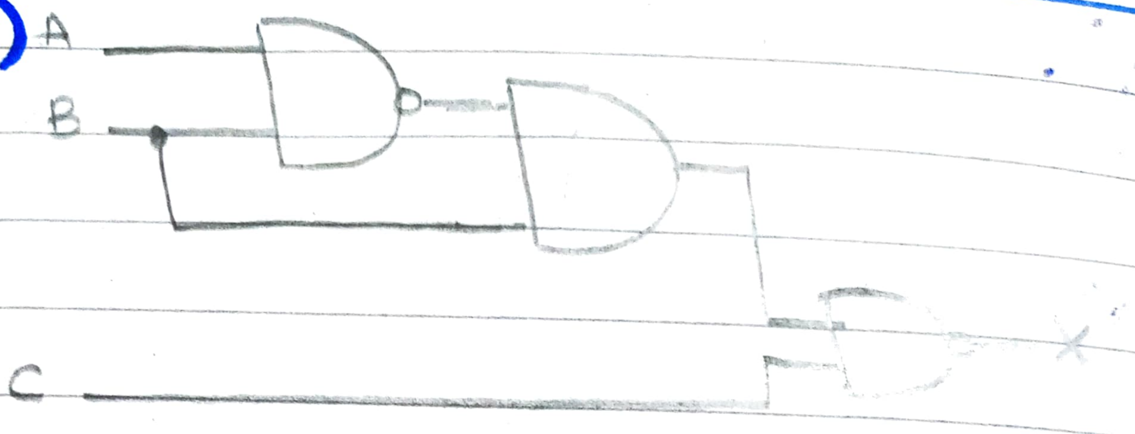
$X = \overline{A} + B$

d)



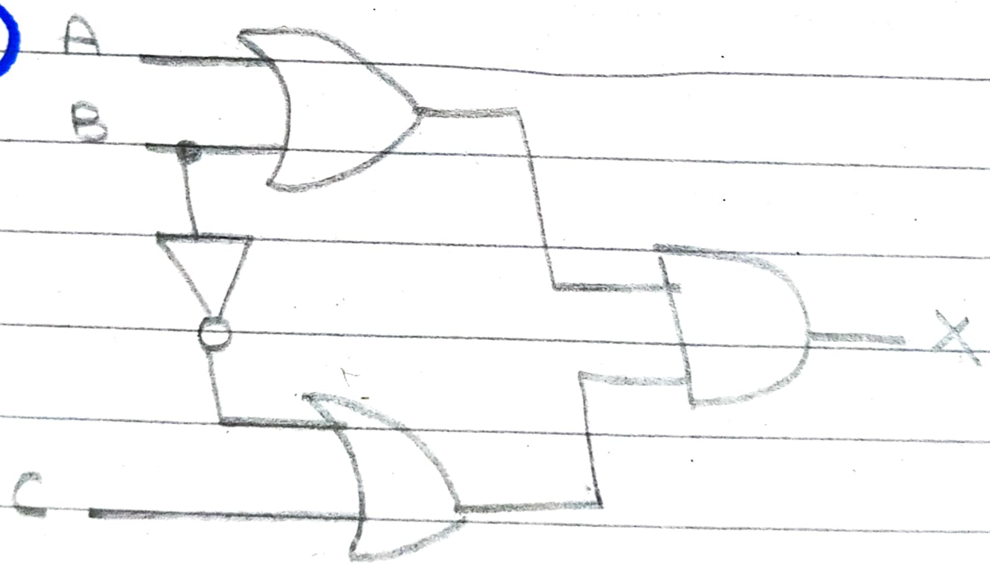
$X = (A+B) + AB$

e)



$$X = \overline{\overline{AB}BC}$$

f)



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

5. Develop truth table for each circuit in figure.

a) $X = ABB$

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

b) $X = AB + B$

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

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

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

d) $X = (A+B) + AB$

A	B	A+B	AB	X
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

e) $X = \overline{\bar{A}BC}$

A	B	C	$\bar{A}$	X
0	0	0	1	1
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	0	1

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

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

IMPLEMENTING COMBINATIONAL LOGIC:-

Example 5-5

Design a logic circuit to implement the operation specified in truth table.

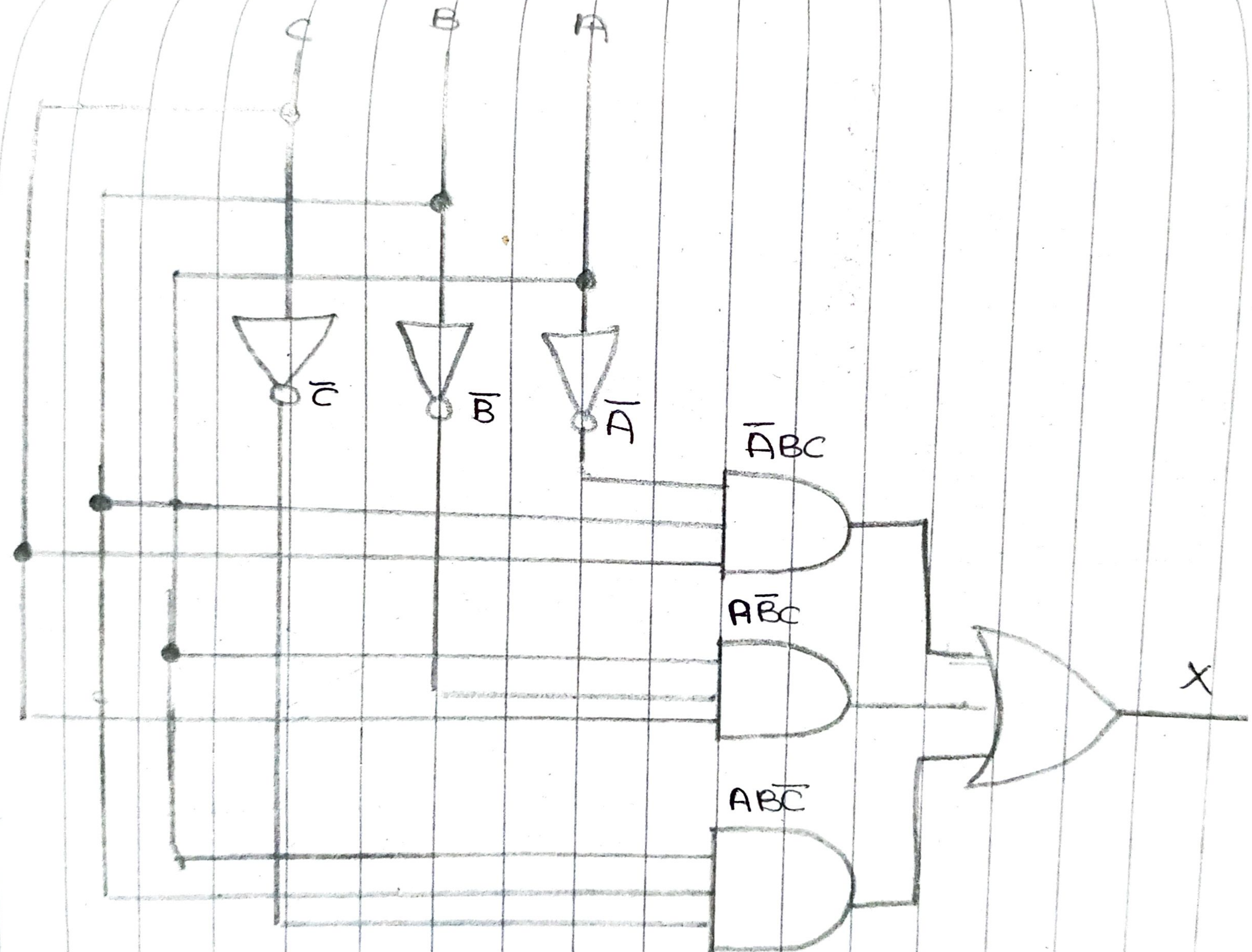
Inputs			Output	
A	B	C	X	Product Term
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\bar{A}BC$
1	0	0	0	
1	0	1	1	$A\bar{B}C$
1	1	0	1	$AB\bar{C}$
1	1	1	0	

Solution:

Notice that $X = 1$ for only three of the input conditions. Therefore, the logic expression is:

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

The logic gates required are three inverters, 3-input AND gates and one 3-input OR gate.



C
B
A
$\overline{C}$
$\overline{B}$
$\overline{A}$
$\overline{A}BC$
$A\overline{B}C$
$AB\overline{C}$
X

Example 5-6

Develop a logic circuit with four input variables that will only produce a 1 output when exactly three input variables are 1.

Solution:

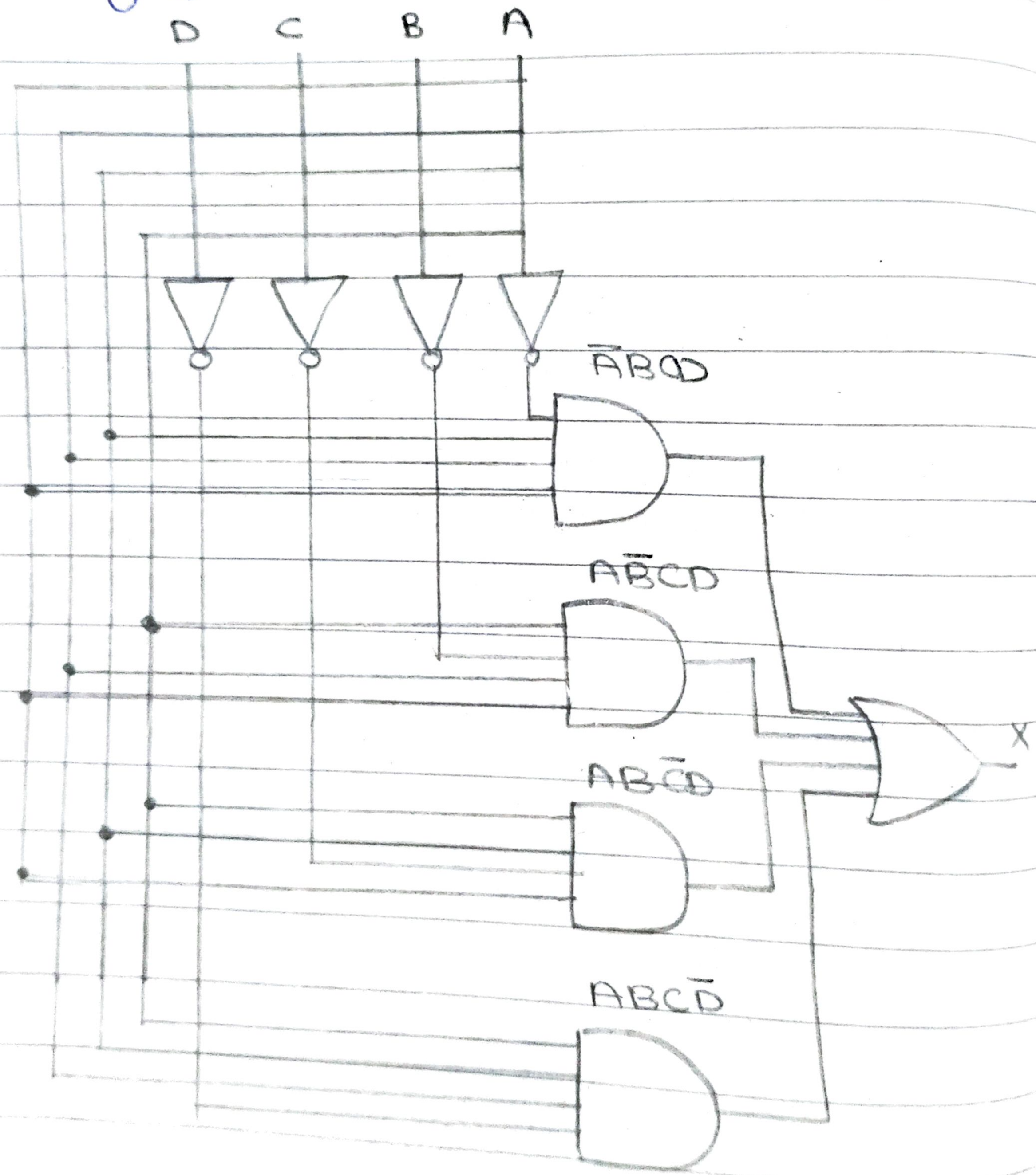
Out of sixteen possible combinations of four variables the combinations in which there are exactly three 1s are listed in table along with the corresponding product term for each.

A	B	C	D	Product Term
0	1	1	1	$\overline{A}BCD$
1	0	1	1	$A\overline{B}CD$
1	1	0	1	$AB\overline{C}D$
1	1	1	0	$ABC\overline{D}$

The product terms are OR-ed to get the following expression.

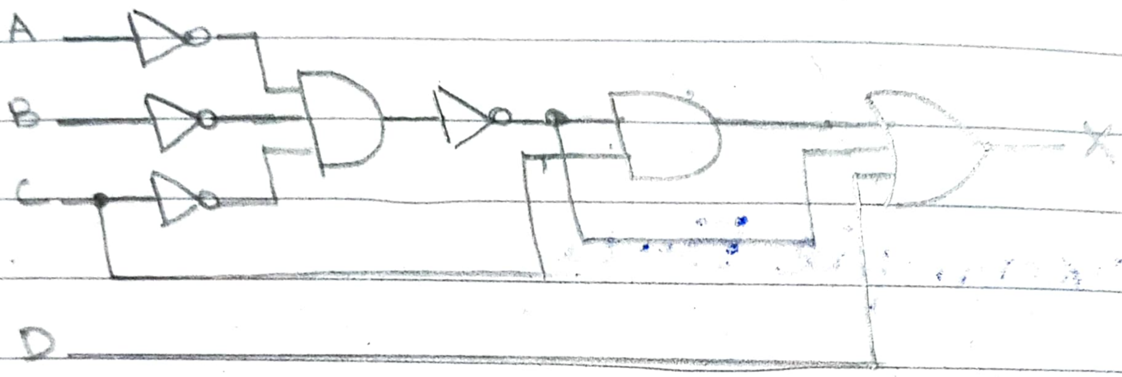
$$X = \overline{A}BCD + A\overline{B}CD + AB\overline{C}D + ABC\overline{D}$$

This expression is implemented in figure with AND-OR logic.



Example 5-7

Reduce the combinational logic circuit in figure to a minimum form.



Solution:

The expression for output of the circuit is:

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

Applying DeMorgan's theorem and Boolean algebra.

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

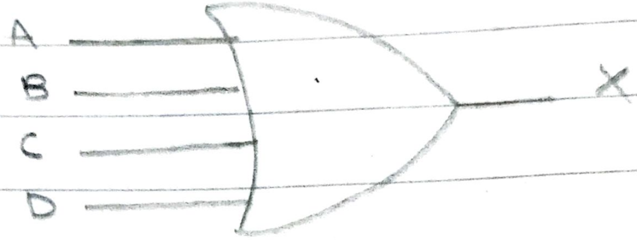
$$= AC + BC + CC + A + B + C + D$$

$$= AC + BC + C + A + B + \cancel{C} + D$$

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

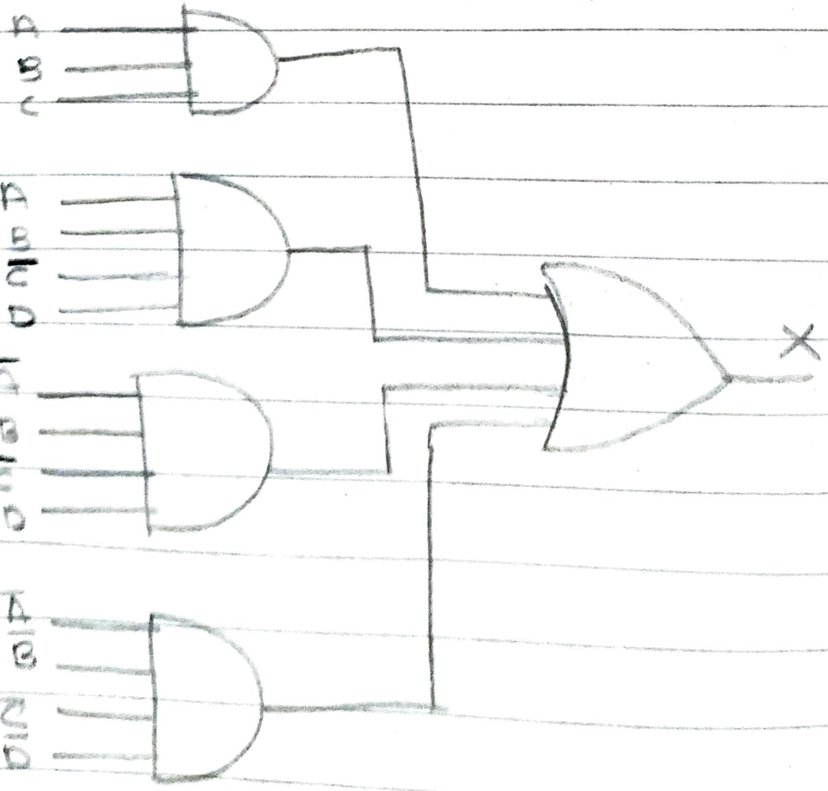
$$X = A + B + C + D$$

The simplified circuit is a 4-input OR gate.



Example 5-8

Minimize the combinational logic circuit in figure. Inverters for the completement variables are not shown.



Solution :

The output expression is

$$X = A\overline{B}\,\overline{C} + AB\overline{C}\,\overline{D} + \overline{A}\,\overline{B}\,\overline{C}D + \overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$$

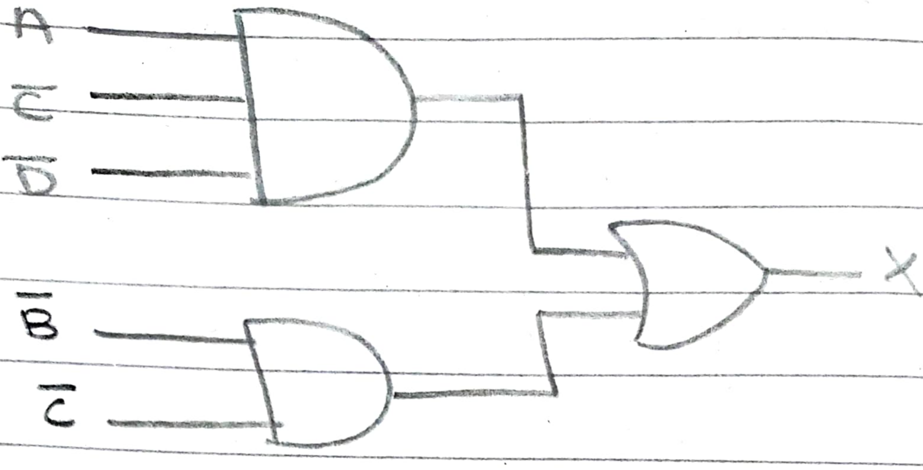
Expanding the first term to include the missing variables D and $\overline{D}$.

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

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

This expanded SOP expression is mapped and simplified on the Karnaugh map in figure. The simplified implementation is shown. Inverters are not shown.

AB \ CD	00	01	11	10
00	1	1		
01				
11	1			
10	1	1		

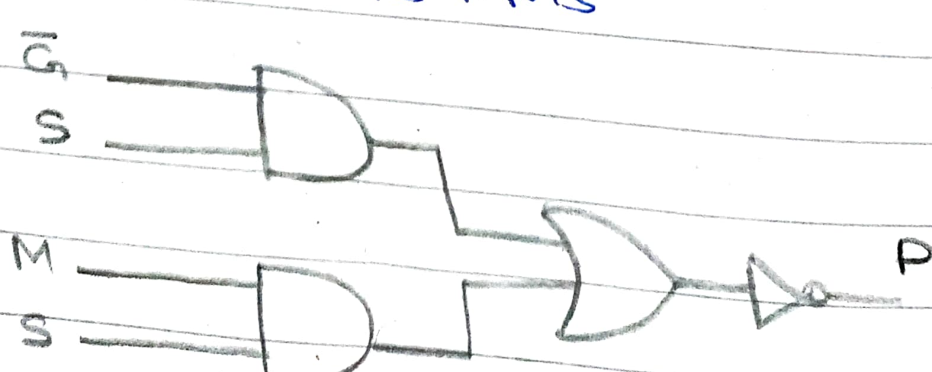


Section 5-2

8. Develop an AND-OR-Invert logic circuit for a power drive which switches on (logic 1) when the guard is in place (logic 1) and switches off (logic 0) when the motor is too hot (logic 0)

Let G = guard, S = switch, M = Motor temperature, P = Power.

$$\Rightarrow P = \overline{\overline{G}S + MS}$$

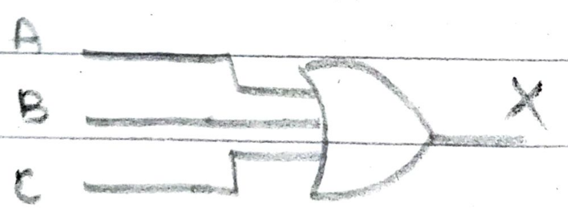


9. An AOI (AND - OR Invert) logic chip has two 4-input AND gates connected to a 2-input NOR gate. Write the Boolean expression for the circuit.

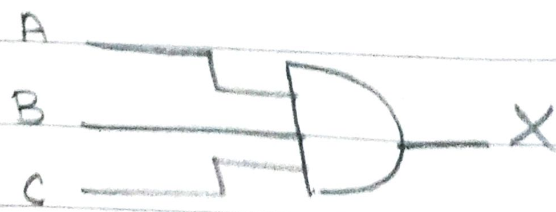
$$X = \overline{ABCD + EFGH}$$

10. Use AND gates, OR gates or combinations of both to implement the following expressions as stated:

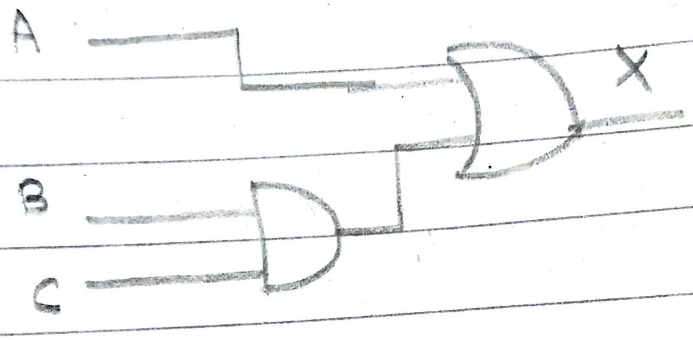
a) $X = A + B + C$



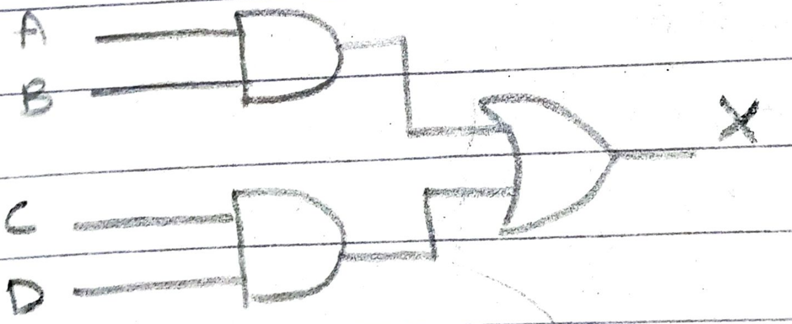
b) $X = ABC$



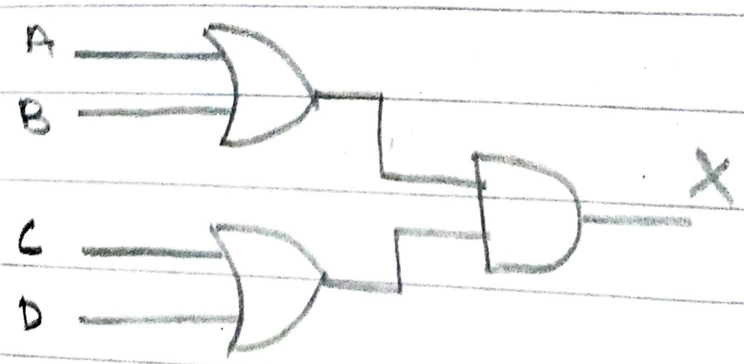
c) $X = A + BC$



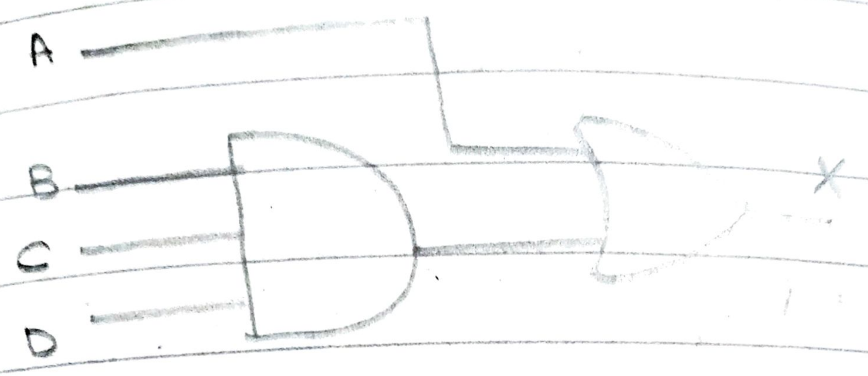
d) $X = AB + CD$



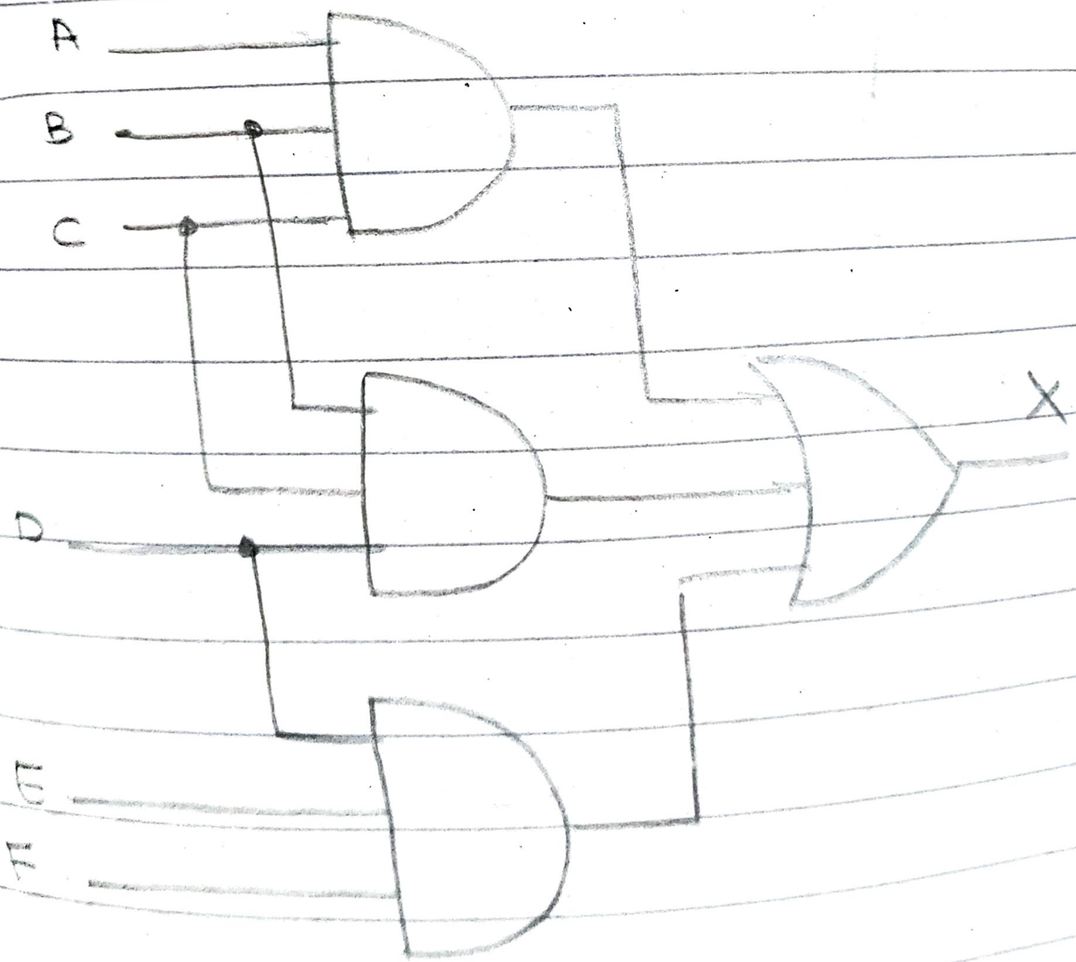
e) $X = (A+B)(C+D)$



f) $X = A + BCD$

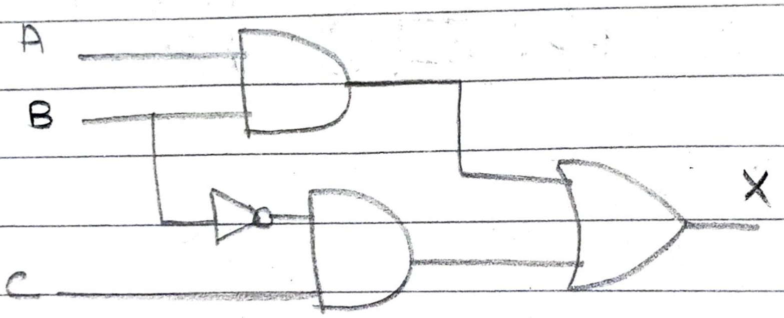


g) $X = ABC + BCD + DEF$

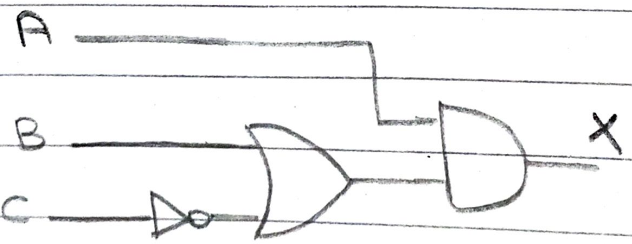


11. Uses AND gates, OR gates and invertors as needed to implement the following logic expressions as stated:

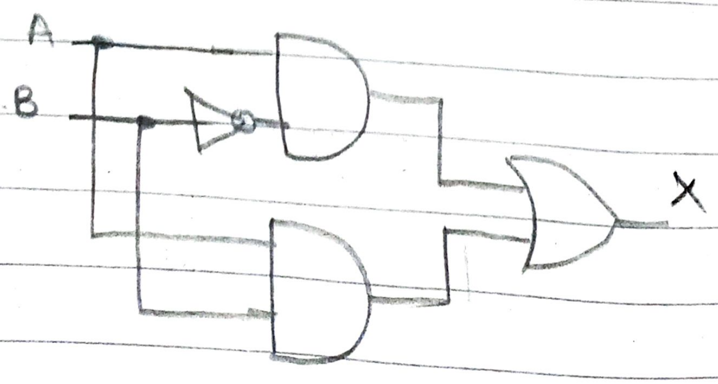
a) $X = AB + \overline{B}C$



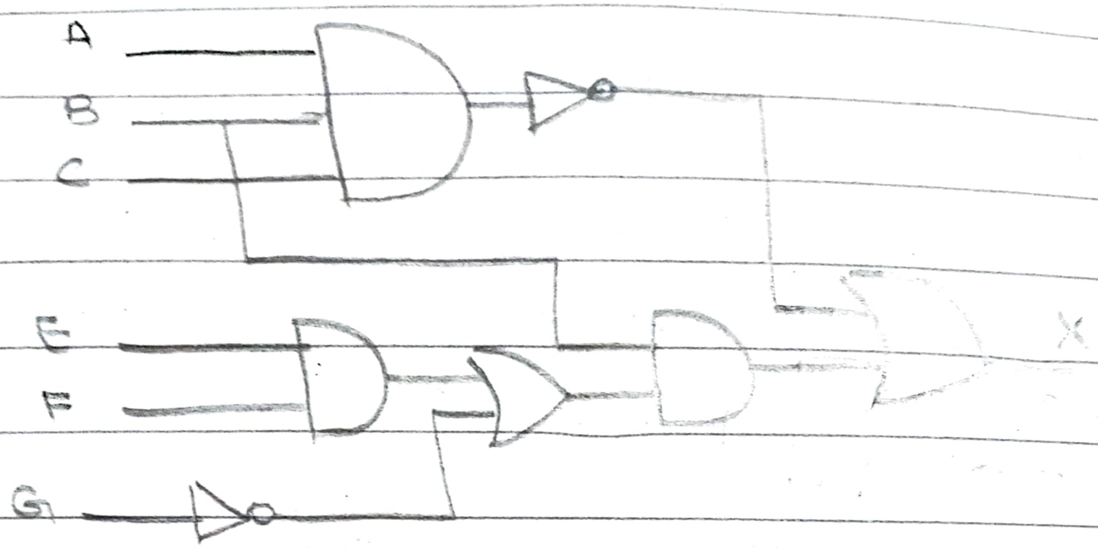
b) $X = A(B + \overline{C})$



c) $X = A\overline{B} + AB$



d) $X = \overline{ABC} + B(EF + \overline{G})$



13. Implement a logic circuit for the truth table.

Inputs			Output
A	B	C	X
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	1

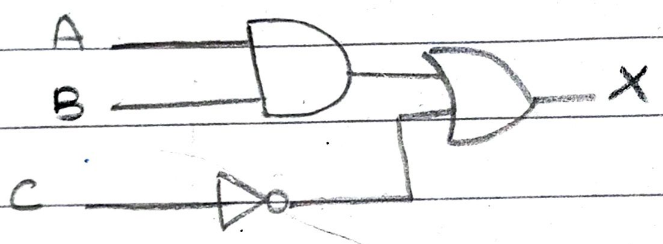
$$X = \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + A\overline{B}\overline{C} + AB\overline{C} + ABC$$

AB \ C	0	1
00	1	
01	1	
11	1	1
10	1	

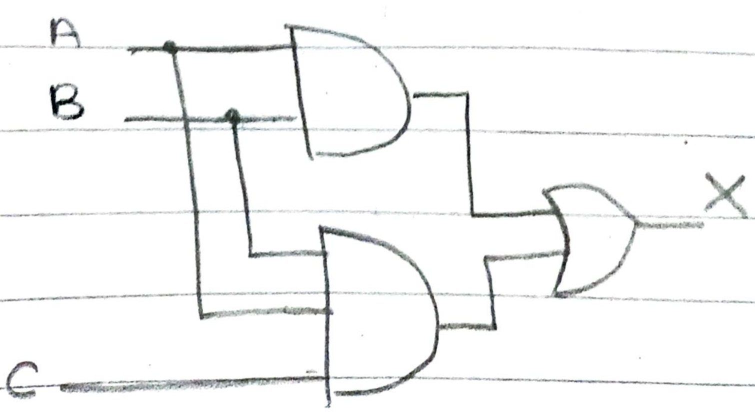
$\rightarrow AB$

$\overline{C}$

$X = AB + \overline{C}$



15. Simplify the circuit as much as possible and verify that the simplified circuit is equivalent to the original by showing that the truth tables are identical.

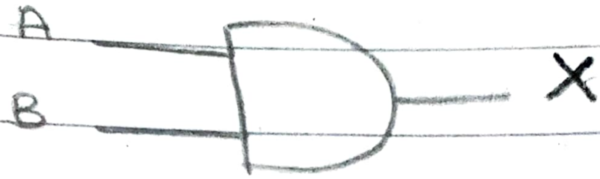


$$X = AB + ABC$$
$$= AB(1+C)$$
$$= AB$$

A	B	C	AB	ABC	X
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	1	0	1
1	1	1	1	1	1

The value of X does not change what value of C is. It is depend on AB.

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



17. Minimize the gates required to implement the function in each part of Problem 11 in SOP form.

a) $X = AB + \overline{B}C$

AB \ C	0	1
00		1
01		
11	1	1
10		1

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

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

No simplification possible.

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

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

AB \ C	0	1
00		
01		
11		1
10		1

$$X = AC$$

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

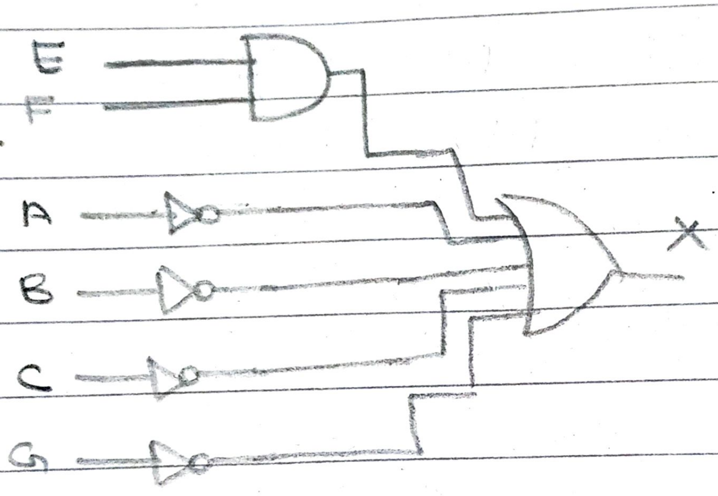
$$\Rightarrow X = AB + A\bar{B}$$
$$= A(B+\bar{B})$$
$$= A(1)$$
$$= A$$

No simplifications.

d) $X = \overline{ABC} + B(EF + \overline{G})$

$\Rightarrow X = \overline{A} + \overline{B} + \overline{C} + BEF + B\overline{G}$

$= \overline{A} + \overline{B} + \overline{C} + EF + \overline{G}$



e) $X = A(BC(A+B+C+D))$

$\Rightarrow X = A(BC(A+B+C+D))$

$= A(ABC + BBC + BCC + BCD)$

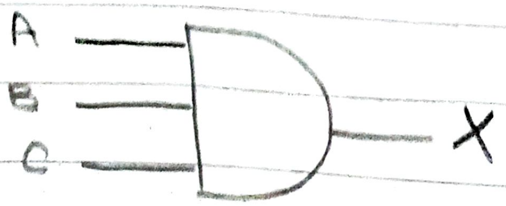
$= AABC + ABC + ABC + ABCD$

$= ABC + A\cancel{B}C + A\cancel{B}C + ABCD$

$= ABC + ABCD$

$= ABC(1 + D)$

$= ABC$



18. Minimize the gates required to implement the functions in each part of problem 12 in SOP form.

a) $X = \overline{A}B + CD + (\overline{A+B})(ACD + \overline{BE})$

$\Rightarrow X = \overline{A}B + CD + \overline{A}\,\overline{B}(ACD + \overline{B} + \overline{E})$

$= \overline{A}B + CD + \overline{A}\,\overline{B}ACD + \overline{A}\,\overline{B}\,\overline{B} + \overline{A}\,\overline{B}\,\overline{E}$

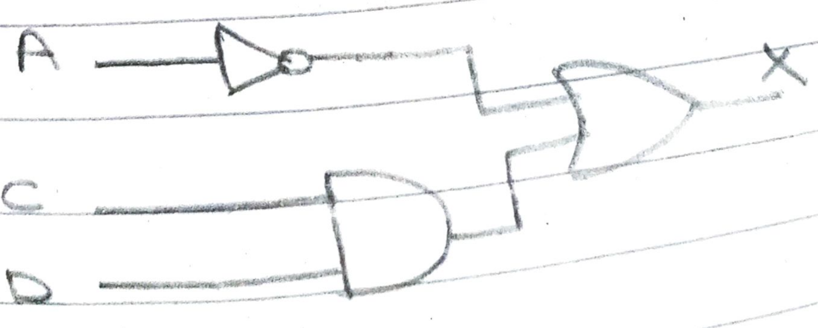
$= \overline{A}B + CD + 0 + \overline{A}\,\overline{B} + \overline{A}\,\overline{B}\,\overline{E}$

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

$= \overline{A} + CD + \overline{A}\,\overline{B}\,\overline{E}$

$= \overline{A}(1 + \overline{B}\,\overline{E}) + CD$

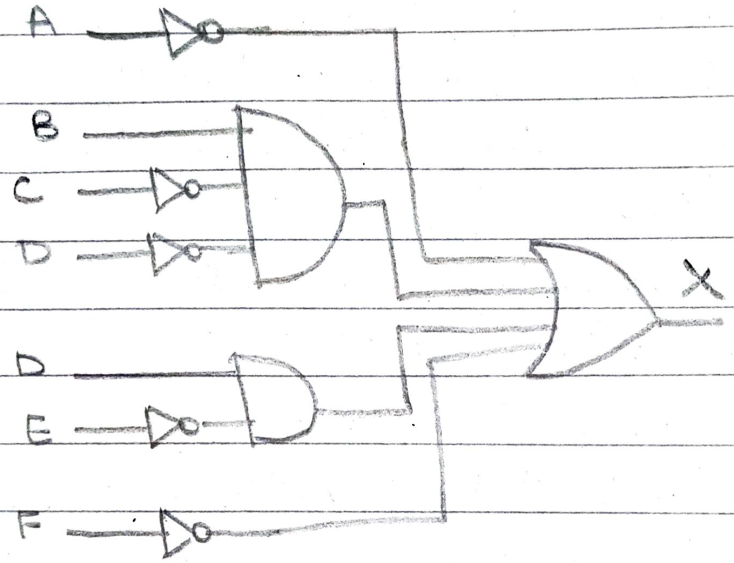
$\Rightarrow X = \overline{A} + CD$



b) $X = AB\overline{CD} + D\overline{E}F + \overline{AF}$

$\Rightarrow X = AB\overline{C}\overline{D} + D\overline{E}F + \overline{A} + \overline{F}$

$\Rightarrow X = B\overline{C}\overline{D} + D\overline{E} + \overline{A} + \overline{F}$



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

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

$\Rightarrow X = \overline{A}B + \overline{A}\overline{C}D + \overline{A}\overline{C}E$