Simplifying Logic Circuits:

(i) Given a truth table, what is the corresponding Boolean Equation.

(ii) Given a Boolean equation, what is the simplest logic circuit for it?

Digital circuits are divided into two broad categories

1. combinational circuits, and
2. Sequential circuits.

In combinational circuits, the output at any instant of time depend upon the inputs presents at that instant of time. This means there is no memory unit in these circuits. There are other types of circuits in which the outputs at any instant of time depend upon the present inputs as well as past input/outputs. This means there are elements used to store past information. These elements are known as memory. Such circuits are known as sequential circuits.

The design requirements of combinational circuit may be specified in one of the following ways

1. A set of statement.
2. Boolean expression, and
3. Truth table.

The following methods can be used to simplify the Boolean functions.

1. Algebric method,
2. Karnaugh-map technique.
3. Quine-McCluskey method, and
4. Variable entered mapping (VEM) technique.

Logical functions are expressed in terms of logical variables. The values assumed by the logical functions as well as the logical variables are in the binary form. Any arbitrary logic function can be expressed in the following forms:

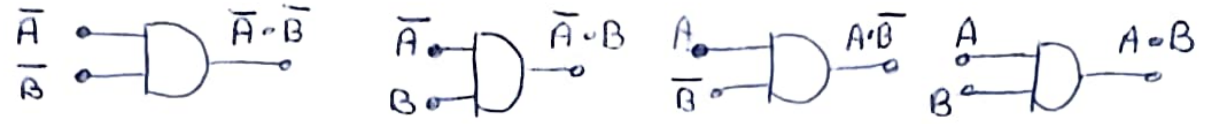
1. Sum-of-product form (SOP), and
2. Product-of-sum form (POS).

This does not mean that the logic function cannot be written in any other form. It can be written in various forms but the above two forms are conveniently suited in arriving at standard methods for designing the circuit will become ~~more~~ clear from the following discussions.

(i) Fundamental Products:

In digital systems, inputs/ signals are available in either complemented or uncomplemented form. For instance, if two variables A and B exist, their complements $\bar{A}$ and $\bar{B}$ normally are also present.

Four ~~variable~~ possible ways to AND gate (two input) signal that are in complemented or uncomplemented form.



(a) (b) (c) (d).

For figure (a) $Y = \bar{A}\cdot\bar{B}$ for Y to be 1 $\bar{A} = \bar{B} = 1$.
Equivalently, the output is 1 only when $A = B = 0$
From now on, we will abbreviate the foregoing input condition by

$A, B = 0, 0$

Fig (b). $Y = \bar{A}\cdot B$. The output can be equal to 1 only when $A = 0$ and $B = 1$.

Fig (c) $Y = A\cdot\bar{B}$, equal a 1 only for the input condition $A = 1$, $B = 0$.

Fig (d) $Y = A\cdot B$, can equal a 1 only when $A = B = 1$.

The table below summarizes the four possible ways of AND ing two inputs in complemented and uncomplemented form.

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

The four logical products are $\bar{A}\bar{B}$, $\bar{A}B$, $A\bar{B}$ and AB are called fundamental product, because they represents the basic ways to combine two inputs with an AND gate.

Fundamental product of three variables:

Variables = 3 = n, $2^3 = 8$, various combinations.

A	B	C	Fundamental product
0	0	0	$\bar{A}\,\bar{B}\,\bar{C}$
0	0	1	$\bar{A}\,\bar{B}\,C$
0	1	0	$\bar{A}\,B\,\bar{C}$
0	1	1	$\bar{A}\,B\,C$
1	0	0	$A\,\bar{B}\,\bar{C}$
1	0	1	$A\,\bar{B}\,C$
1	1	0	$A\,B\,\bar{C}$
1	1	1	$A\,B\,C$

Fundamental product of four variable. (n=4).
No of combinations $2^n = 16$

A	B	C	D	Fundamental product
0	0	0	0	$\bar{A}\,\bar{B}\,\bar{C}\,\bar{D}$
0	0	0	1	$\bar{A}\,\bar{B}\,\bar{C}\,D$
0	0	1	0	$\bar{A}\,\bar{B}\,C\,\bar{D}$
0	0	1	1	$\bar{A}\,\bar{B}\,C\,D$
0	1	0	0	$\bar{A}\,B\,\bar{C}\,\bar{D}$
0	1	0	1	$\bar{A}\,B\,\bar{C}\,D$
0	1	1	0	$\bar{A}\,B\,C\,\bar{D}$
0	1	1	1	$\bar{A}\,B\,C\,D$
1	0	0	0	$A\,\bar{B}\,\bar{C}\,\bar{D}$
1	0	0	1	$A\,\bar{B}\,\bar{C}\,D$
1	0	1	0	$A\,\bar{B}\,C\,\bar{D}$
1	0	1	1	$A\,\bar{B}\,C\,\bar{D}$
1	1	0	0	$A\,B\,\bar{C}\,\bar{D}$
1	1	0	1	$A\,B\,\bar{C}\,D$
1	1	1	0	$A\,B\,C\,\bar{D}$
1	1	1	1	$A\,B\,C\,D$

Sum of Product

Given a truth table one can find the Boolean expression for the output by ORing the fundamental products that produce 1 outputs. The following examples show you how.

Truth table.

A	B	Y	Fundamental Product
0	0	0	
0	1	0	
1	0	1	$\rightarrow A\bar{B}$
1	1	1	$\rightarrow AB$

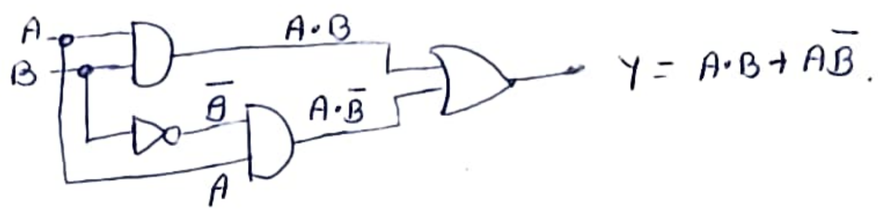
ORing the fundamental products for which the output is 1.

ie $Y = A\bar{B} + AB$ — This Boolean expression is sum of Product form

AND, OR, AND

logic circuit

$A\bar{B}$ is AND gate.
AB is AND gate
$+ \rightarrow$ is OR gate. so.



Example II : Three variables:

A	B	C	Y	Fundamental Product
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	$\bar{A}BC$
1	0	0	0	0
1	0	1	1	$A\bar{B}C$
1	1	0	1	$AB\bar{C}$
1	1	1	1	ABC

logic circuit for the given Truth table.

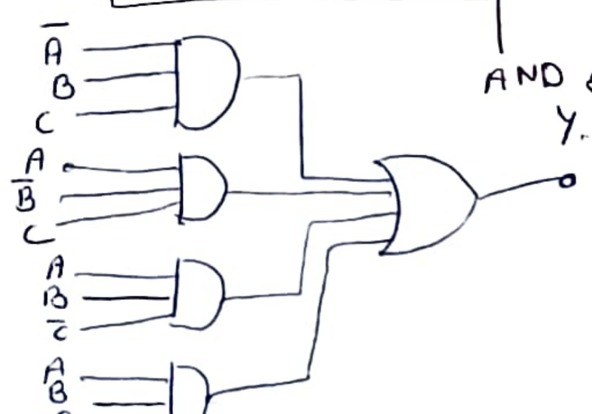
step I - Find the fundamental product corresponding to each 1 output in the truth table.

step II :- OR the fundamental product.

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

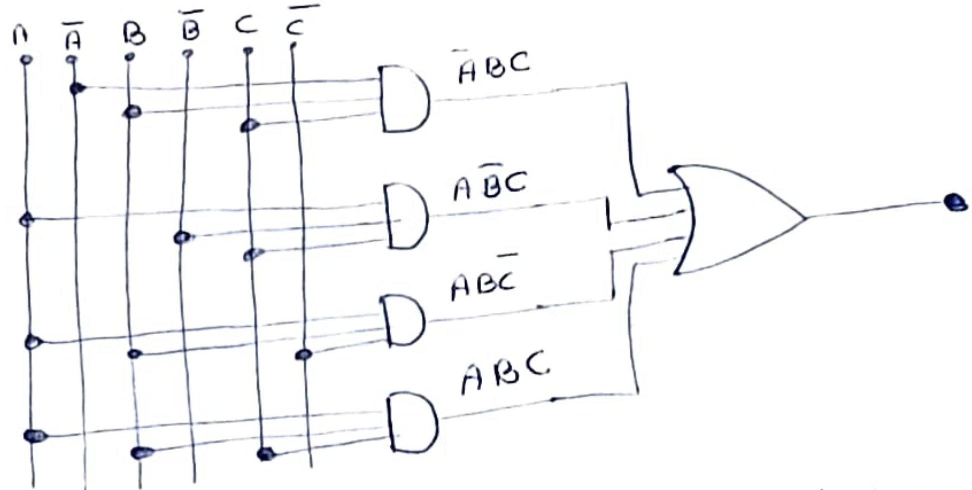
OR gate

AND gate



The Boolean expression so obtained is a logical sum of fundamental products. For this reason, this method of getting the Boolean expression is called the "sum-of-product" method.

sum of product logic diagram AND-OR network.



(ii) L Give a Boolean expression, what is logic circuit.

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

AND

OR.

In many ways logic circuit can be obtained.

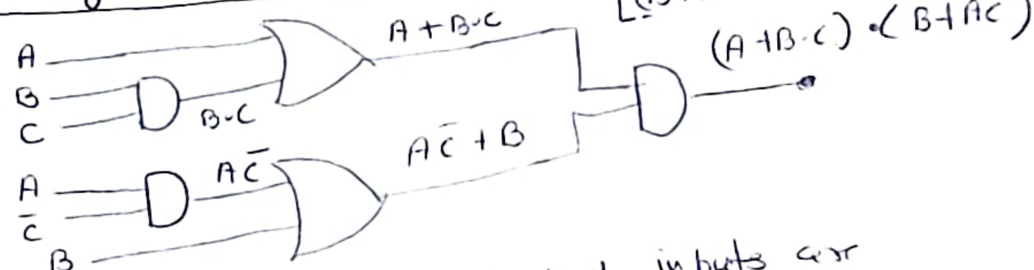
$B\cdot C \rightarrow$ AND gate

$A\cdot\bar{C}$ AND gate.

$\left[\begin{matrix}(A+BC) \\ (B+A\bar{C})\end{matrix}\right]$ OR gate

Sol

(I) using AND, OR gate,



Here we have assumed that inputs are available in complemented form also. So this is AND-OR realisation of the given Boolean expression.

(ii) Design the circuit with only one type of gates [Universal gate] NAND or NOR.

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

First convert it into sum of product (SOP) equation.

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

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

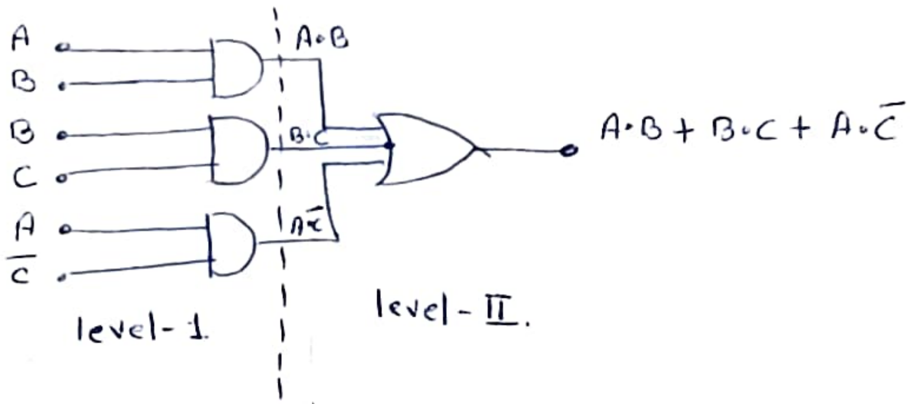
$$= A \cdot B + \underbrace{(A \cdot A)}_{A} \cdot \bar{C} + \underbrace{(B \cdot B)}_{B} \cdot C + A \cdot B \cdot \underbrace{(C \cdot \bar{C})}_{0}$$

$$Y = A \cdot B + B \cdot C + A\bar{C} \quad \text{——} \quad (2).$$

This equation is sum of product form.

This can be realized using AND-OR realization

This is known as two-level realization. The first level consists of AND gates and the second level level consists of OR gate, as shown is figure below.



Realization using NAND-NAND gates:

Apply De Morgan's theorem to eqution (2).

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

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

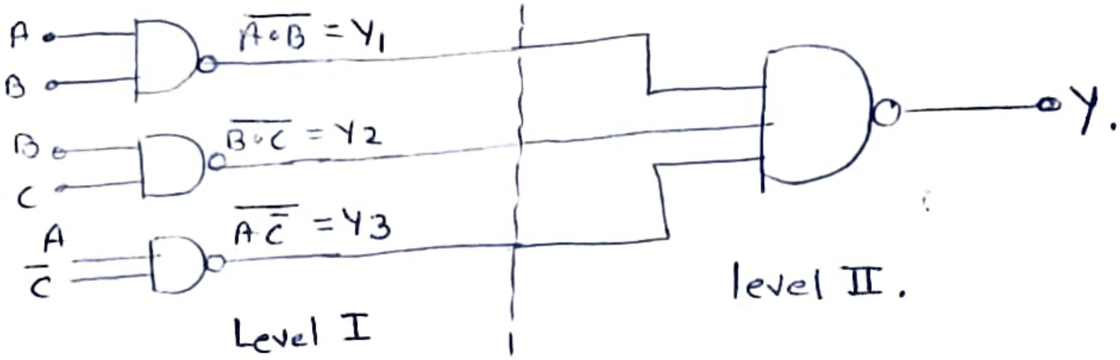
$$\bar{Y} = Y_1 \cdot Y_2 \cdot Y_3$$ where $Y_1 = \overline{A \cdot B}$, $Y_2 = \overline{B \cdot C}$, $Y_3 = \overline{A\bar{C}}$

Taking complements on both sides NAND Gate.

$$\bar{\bar{Y}} = \overline{Y_1 \cdot Y_2 \cdot Y_3}$$

$$Y = \overline{Y_1 \cdot Y_2 \cdot Y_3} \longleftarrow \text{NAND gate.} \quad \text{——} \quad (3).$$

The realization of equation (3) is given in figure.



using NOR gates only [Product of Sum].

$Y = (A+BC) \cdot (B+A\bar{C})$ — equation (1).

using the property

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

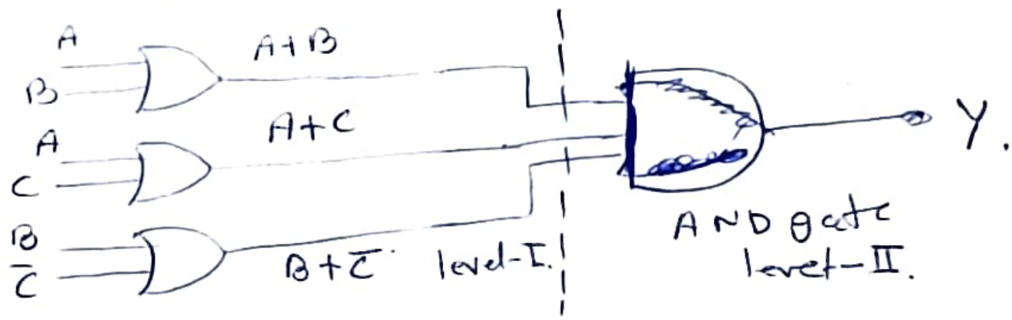
$(B+A\bar{C}) = (B+A) \cdot (B+\bar{C})$.

Equation (1) can be written as

$Y = (A+B) \cdot (A+C) \cdot (A+B) \cdot (B+\bar{C})$. OR gate.

$= (A+B) \cdot (A+C) \cdot (B+\bar{C})$ — (4). AND gate

Realization of eq. (4) using OR and AND gate is shown below.



use Demorgan's Theorem, we can write eq. 4 as

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

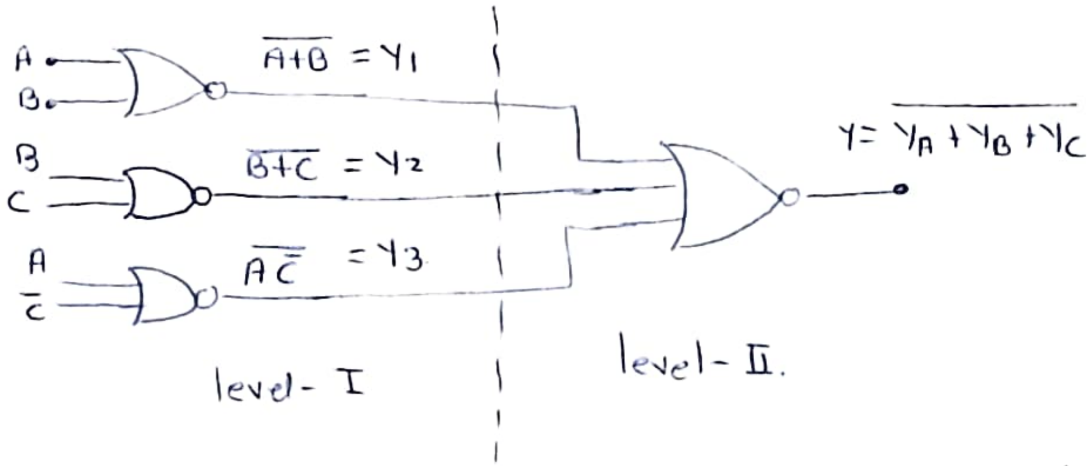
$= \overline{A+B} + \overline{A+C} + \overline{B+\bar{C}}$

$\bar{\bar{Y}} = \overline{Y_A + Y_B + Y_C}$. where $Y_A = \overline{A+B}$

$Y_B = \overline{A+C}$

$Y = \overline{Y_A + Y_B + Y_C}$ ← NOR gate, $Y_C = \overline{B+\bar{C}}$

(5).



POS equation can be designed using NOR gates only (only one type of gate - Universal gate NOR).

Minimization/simplification of ~~logic circuit~~/ Boolean expression.

consider eq. 2. [P.O.S] equation.

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

using Algebraic method.

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

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

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

$$= AB[1 + C] + A\bar{C} + A$$

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

$$= AB[C + \bar{C}] + A\bar{C} + BC.$$

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

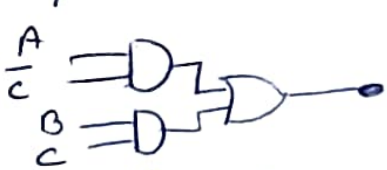
$$= A\bar{C}[1 + B] + BC[1 + A]$$

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

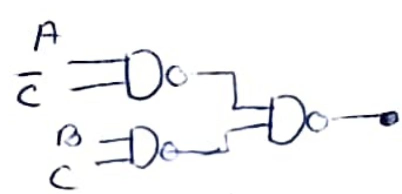
$Y = A\bar{C} + BC.$ —(6) SOP equation can be realised by AND-OR gate. using De morgan's theorem NAND gate can be used.

$$\bar{Y} = \overline{A\bar{C} + BC} = \overline{A\bar{C}} \cdot \overline{BC} = Y_1 \cdot Y_2, \quad Y_1 = A\bar{C}, \quad Y_2 = BC$$

$$Y = \bar{\bar{Y}} = \overline{Y_1 \cdot Y_2} \quad \text{— (7).}$$



Eq. (6).



Eq.(7)

Numbe rof Gates required.

(1) For eq. 7 = 3, 2-input NAND or 3, 2-input NOR gates

(2). For eq. 3 3, 2 input NAND & 1, 3-input NAND.
For eq. 5 3-2 input NOR and 1-3-input NOR

(3) For Eq. (4). OR-AND realization [POS]
3-2 input OR and 1-3 input AND gate.

For eq (2) AND-OR realization [S.OP]
3-2 input AND gate and 1-3 input OR gate.

(4). Eq. 1. 3-2-input AND gate
2-2-input OR gate.] Maximum Number of Gates, which

increases the propagation delay time, which is the same as decreases in speed of operation.

Eq. 3, 2, 5 For universal gates. only one type of gates NAND/NOR are required, which is convenient to use, when we use ICs because a number of similar gates are available in the same package.

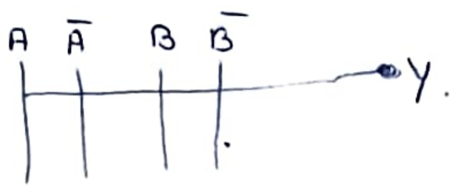
Eq. 7 :- Requires minimum number of gates. least propagation delay,.

Algebric simplification:

$Y = A\bar{B} + AB$ Example 1..

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

$Y = A$



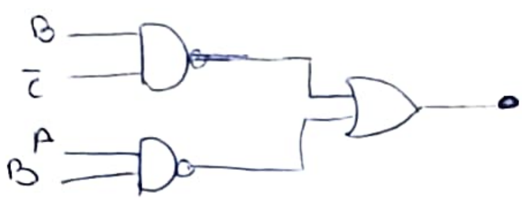
Ex. 2:- $Y = \bar{A}B\bar{C} + AB\bar{C} + ABC$

sol $Y = \bar{A}B\bar{C} + AB\bar{C} + AB\bar{C} + ABC$ $\boxed{X + X = X}$

↑add using the property $A + A = A$.

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

$= B\bar{C} + AB$



Example 2 $Y = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$

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

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

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

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

$= AB + BC + A\bar{B}C. \rightarrow AB + C[B + A\bar{B}]$

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