Practical No 3
Implement combinational circuits

Aim:

Design and implement combinational circuit based on the problem given and minimizing using K-maps.

Theory:

Sum of Products (SOP) and Product of Sums (POS) expressions are two standard forms used to represent Boolean functions. They are essential concepts in digital logic design, providing a systematic way to simplify and implement logic circuits

i) Sum of Products (SOP) Expression

SOP Definition:

- The SOP form is a Boolean expression where several product terms (ANDed variables) are summed (ORed) together.
- Each product term is called a minterm and corresponds to a unique combination of input variables that result in the function being true (1).

Example: Consider a Boolean function f(A,B,C) with the following truth table:

A	B	C	f
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	0
1	1	1	1

The SOP expression for this function is obtained by writing a product term for each row where the function output is 1

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

There is also one alternate way to represent the truth table using cell representation, which is as follows

$$f(ABC) = \sum m(0,2,4,7)$$

In order to implement SOP form expressions, AND-OR or NAND-NAND logic is used

ii) Product of Sums (POS) Expression

POS Definition:

- The POS form is a Boolean expression where several sum terms (ORed variables) are multiplied (ANDed) together.
- Each sum term is called a maxterm and corresponds to a unique combination of input variables that results in the function being false (0).

Example: Consider the Boolean function f(A,B,C) with the same truth table used in SOP form

A	B	C	f
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	0
1	1	1	1

The POS expression for this function is obtained by writing a product term for each row where the function output is 0

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

There is also one alternate way to represent the truth table using cell representation, which is as follows

$$f(ABC) = \Pi M(1,3,5,6)$$

In order to implement POS form expressions, OR-AND or NOR-NOR logic is used

K-maps:

Karnaugh Maps (K-maps) are a graphical tool used to simplify Boolean algebra expressions. They help visualize and minimize logical functions by organizing the truth values in a grid format. K-maps are particularly useful for simplifying Sum of Products (SOP) and Product of Sums (POS) expressions, making them easier to implement in digital circuits.

A K-map is a grid of cells, where each cell represents a minterm(m) or maxterm(M) of the function. The number of cells in a K-map is 2^n, (n is the number of variables).

The cells are arranged such that only one variable changes between adjacent cells, following Gray code order.

Prof. Ismail H. Popatia

Steps to Use K-maps for SOP and POS Simplification

Step 1: Construct the K-map
For SOP:
Place a 1 in the K-map cells corresponding to the minterms where the function is true.
For POS:
Place a 0 in the K-map cells corresponding to the maxterms where the function is false.

Step 2: Group the Cells
Group adjacent 1's (for SOP) or 0's (for POS) in powers of two (1, 2 (pair), 4(quad), 8(octet)).
Groups can wrap around the edges of the K-map.
Each group should be as large as possible to maximize simplification.

Step 3: Write the Simplified Expression and verification
For SOP:
i) For each group, write a product term (AND) by including the variables that do not change within the group.
ii) Sum (OR) all the product terms.
iii) Use either AND-OR logic or NAND-NAND logic to implement
iv) Verify the working by applying all the possible combinations of the inputs and comparing it with the truth table

For SOP:
i) For each group, write a sum term (OR) by including the variables that do not change within the group.
ii) AND all the sum terms.
iii) Use either OR-AND logic or NOR-NOR logic to implement
iv) Verify the working by applying all the possible combinations of the inputs and comparing it with the truth table

Prof. Ismail H. Popatia

Problem 1: Based on SOP function

Solve the following function using k-maps and implement using suitable gates

$$f(ABCD) = \sum m(0,1,2,3,6,7,14,15)$$

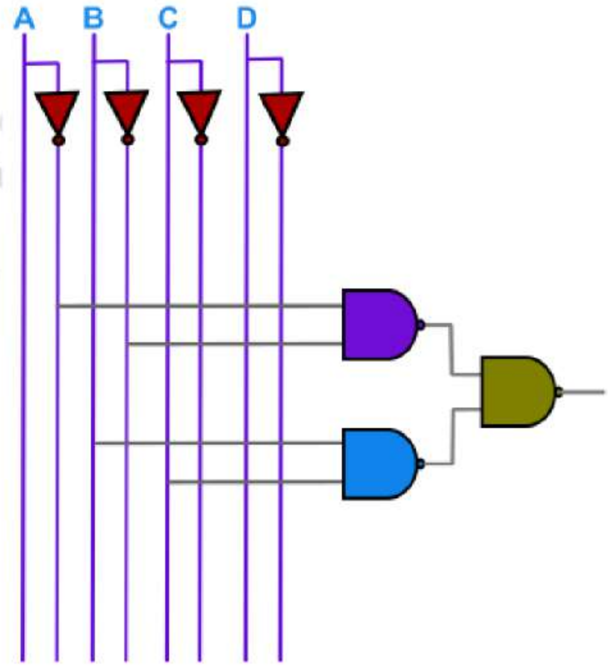
Solution: The given function is a SOP function, so we draw the SOP k-map as follows

	$\overline{A}.\overline{B}$	$\overline{A}.B$	$A.B$	$A.\overline{B}$
$\overline{C}.\overline{D}$	1	0	0	0
$\overline{C}.D$	1	0	0	0
$C.D$	1	1	1	0
$C.\overline{D}$	1	1	1	0

The above k-map has two quads, hence the equation of the k-map is

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

To implement the above expression we use NAND-NAND logic as follows



Prof. Ismail H. Popatia

Verification using Truth table:

Using a simulator we verify the above circuit using the following truth table

Inputs				Output
A	B	C	D	Y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

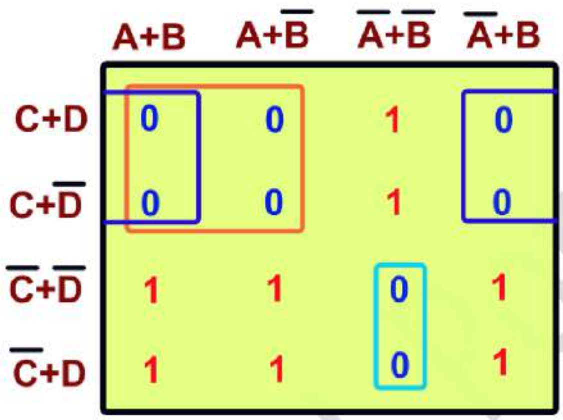
Prof. Ismail H. Popatia

Problem 2: Based on POS function

Solve the following function using k-maps and implement using suitable gates

f(ABCD) = ΠM(0,1,4,5,8,9,14,15)

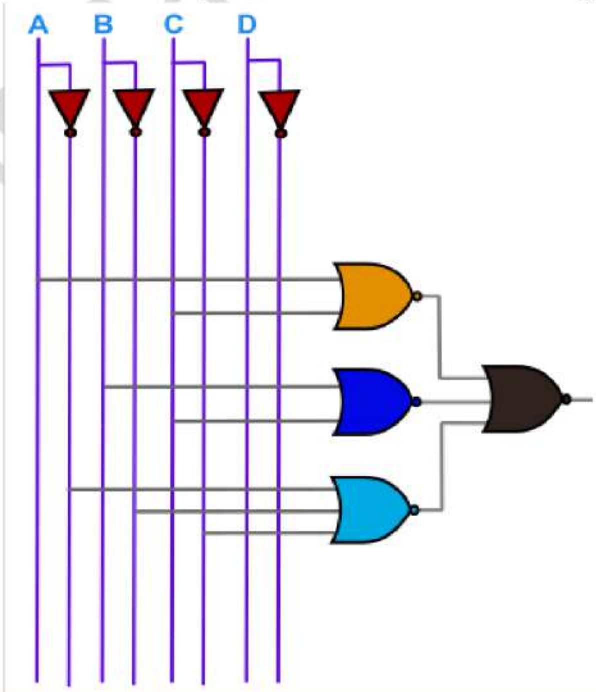
Solution: The given function is a POS function, so we draw the POS k-map as follows



The above k-map has two quads, hence the equation of the k-map is

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

To implement the above expression we use NOR-NOR logic as follows



Prof. Ismail H. Popatia

Verification using Truth table:

Using a simulator we verify the above circuit using the following truth table

Inputs				Output
A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	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	0
1	1	1	1	0

Result: The given SOP and POS functions were solved using k-maps and implemented.

For video demonstration of the given practical click on the link below or scan the QR-code

https://youtu.be/quzwiYmySo0

Prof. Ismail H. Popatia