DLD Lab-04

Karnaugh Map

Instructor: Mazhar Iqbal

EL227 – Digital Logic Design-Lab

SEMESTER SPRING 2021

FEBRUARY 9, 2022

NATIONAL UNIVERSTIY OF COMPUTER AND EMERGING SCIENCES, FAST- PESHAWAR CAMPUS
Department Of Computer Science

Contents

1. Objectives:

3/4 variable Boolean function simplification using K-map in SOP (sum of products) and POS (product of sums) forms

Implementation of simplified expressions using AND, OR and NOT gates

Implementation of simplified expressions using NAND and NOR gates

2. Outcomes:

Students should be able to

Simplify, implement and verify 3/4 variable Boolean expressions in SOP and POS forms,

and using only NAND and only NOR gates.

3. Equipment Required:

- DEV-2765E Trainer Board/ Multisim 14.2 /Logic.ly
- 7404 hex NOT (Inverter) gate IC
- 7408 quad 2-input AND gate IC
- 7432 quad 2-input OR gate IC
- 7400 quad 2-input NAND gate ICs
- 7402 quad 2-input NOR gate ICs
- Multisim / Logicly

1- Simplification of 3 variable Boolean function using Kmap

In the lab there are 2 ACs and a door . There is a monitoring room as well . If any of the AC is on , AND the door is open (i.e logic one for the door) , then an output signal (i.e logic 1) is issued to the monitoring room , otherwise output remains zero .

Make a truth table corresponding to the above scenario, use A1 and A2 for ACs and D for door and S for output signal , simplify the Boolean function into

(a) sum-of-products form and (b) product-of-sums form

1. Min term(product)

Minterm is product of boolean variables either in normal form or complemented form. In Minterm, we look for the functions where the output results in “1”

Row	X	Y	Z	F	***Minterm***
0	0	0	0	F(0,0,0)	$X' \cdot Y' \cdot Z'$
1	0	0	1	F(0,0,1)	$X' \cdot Y' \cdot Z$
2	0	1	0	F(0,1,0)	$X' \cdot Y \cdot Z'$
3	0	1	1	F(0,1,1)	$X' \cdot Y \cdot Z$
4	1	0	0	F(1,0,0)	$X \cdot Y' \cdot Z'$
5	1	0	1	F(1,0,1)	$X \cdot Y' \cdot Z$
6	1	1	0	F(1,1,0)	$X \cdot Y \cdot Z'$
7	1	1	1	F(1,1,1)	$X \cdot Y \cdot Z$

2. Max term(SUM)

Maxterm is sum of boolean variables either in normal form or complemented form. In Maxterm we look for function where the output results in "0"

Row	X	Y	Z	F	Maxterm
0	0	0	0	F(0,0,0)	X + Y + Z
1	0	0	1	F(0,0,1)	X + Y + Z′
2	0	1	0	F(0,1,0)	X + Y′ + Z
3	0	1	1	F(0,1,1)	X + Y′ + Z′
4	1	0	0	F(1,0,0)	X′ + Y + Z
5	1	0	1	F(1,0,1)	X′ + Y + Z′
6	1	1	0	F(1,1,0)	X′ + Y′ + Z
7	1	1	1	F(1,1,1)	X′ + Y′ + Z′

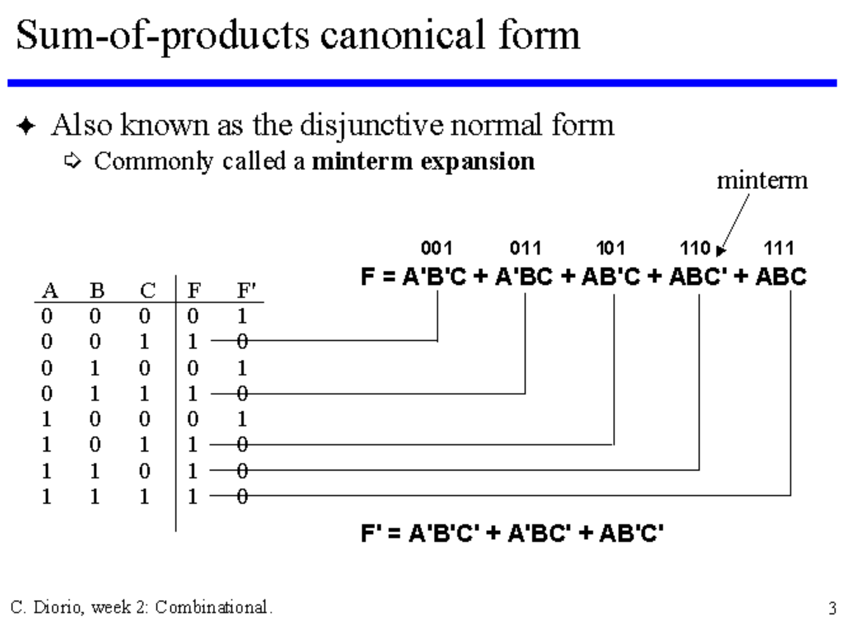
3. Min Term & Max term

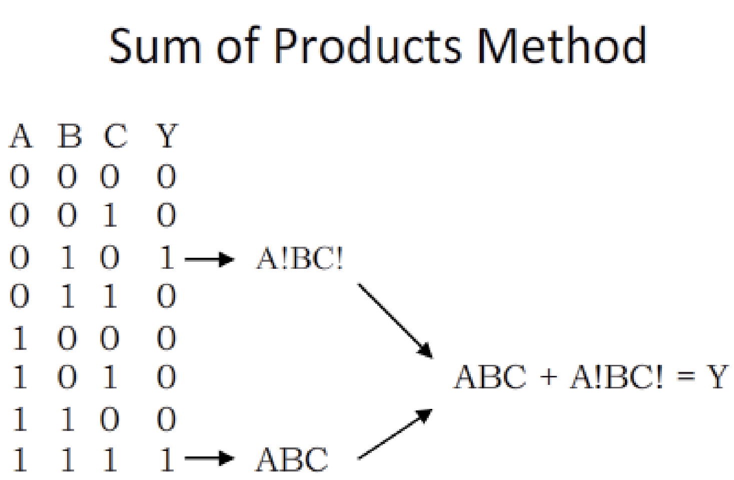
			Minterms	Maxterms
X	*Y*	*Z*	*Product Terms*	*Sum Terms*
0	0	0	$m_0 = \overline{X}\cdot\overline{Y}\cdot\overline{Z} = \min(\overline{X},\overline{Y},\overline{Z})$	$M_0 = X+Y+Z = \max(X,Y,Z)$
0	0	1	$m_1 = \overline{X}\cdot\overline{Y}\cdot Z = \min(\overline{X},\overline{Y},Z)$	$M_1 = X+Y+\overline{Z} = \max(X,Y,\overline{Z})$
0	1	0	$m_2 = \overline{X}\cdot Y\cdot\overline{Z} = \min(\overline{X},Y,\overline{Z})$	$M_2 = X+\overline{Y}+Z = \max(X,\overline{Y},Z)$
0	1	1	$m_3 = \overline{X}\cdot Y\cdot Z = \min(\overline{X},Y,Z)$	$M_3 = X+\overline{Y}+\overline{Z} = \max(X,\overline{Y},\overline{Z})$
1	0	0	$m_4 = X\cdot\overline{Y}\cdot\overline{Z} = \min(X,\overline{Y},\overline{Z})$	$M_4 = \overline{X}+Y+Z = \max(\overline{X},Y,Z)$
1	0	1	$m_5 = X\cdot\overline{Y}\cdot Z = \min(X,\overline{Y},Z)$	$M_5 = \overline{X}+Y+\overline{Z} = \max(\overline{X},Y,\overline{Z})$
1	1	0	$m_6 = X\cdot Y\cdot\overline{Z} = \min(X,Y,\overline{Z})$	$M_6 = \overline{X}+\overline{Y}+Z = \max(\overline{X},\overline{Y},Z)$
1	1	1	$m_7 = X\cdot Y\cdot Z = \min(X,Y,Z)$	$M_7 = \overline{X}+\overline{Y}+\overline{Z} = \max(\overline{X},\overline{Y},\overline{Z})$

4. Sum of Product (Min Term)

The SOP (Sum of Product) is the methods for deducing a particular logic function. Conversely, SOP produces a logical expression comprised of the OR of the multiple AND terms

A Sum-Of-Products Boolean expression is exactly a set of Boolean terms added together, each term being a product of a combination of Boolean variables. To get the Sum of Products form from a truth table, OR together all of the min terms which has a value of 1.





5. Product of Sum (Max term)

The POS (Product of Sum) is the methods for deducing a particular logic function. Conversely, POS produces a logical expression comprised of the AND of the multiple OR terms

A Product of Sum Boolean expression is exactly a set of Boolean terms product together, each term being a added of a combination of Boolean variables. To get the Product of Sum form from a truth table, product together all of the max terms which has a value of 0.

Product-of-sums canonical form

- Formally: The conjunctive normal form
  - Also called a **maxterm expansion**
  - $F(A,B,C) = \Pi M(0,2,4)$

maxterm

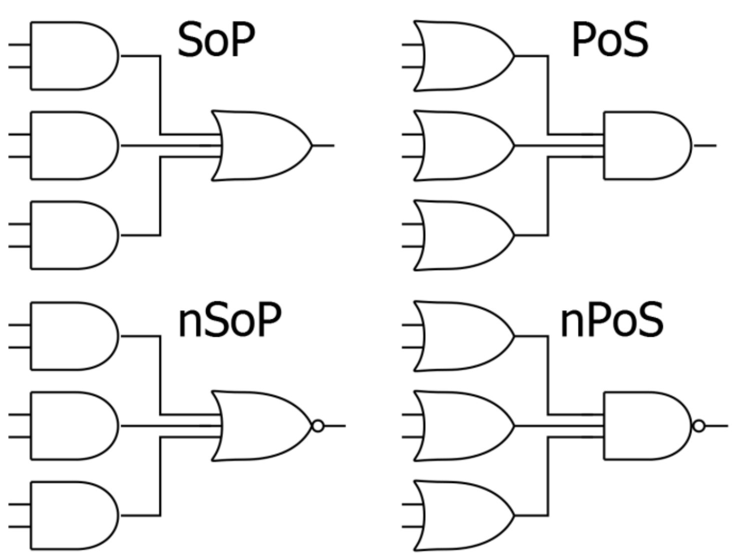
$$F = \underbrace{(A + B + C)}_{000}\ \underbrace{(A + B' + C)}_{010}\ \underbrace{(A' + B + C)}_{100}$$

A	B	C	F	F'
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

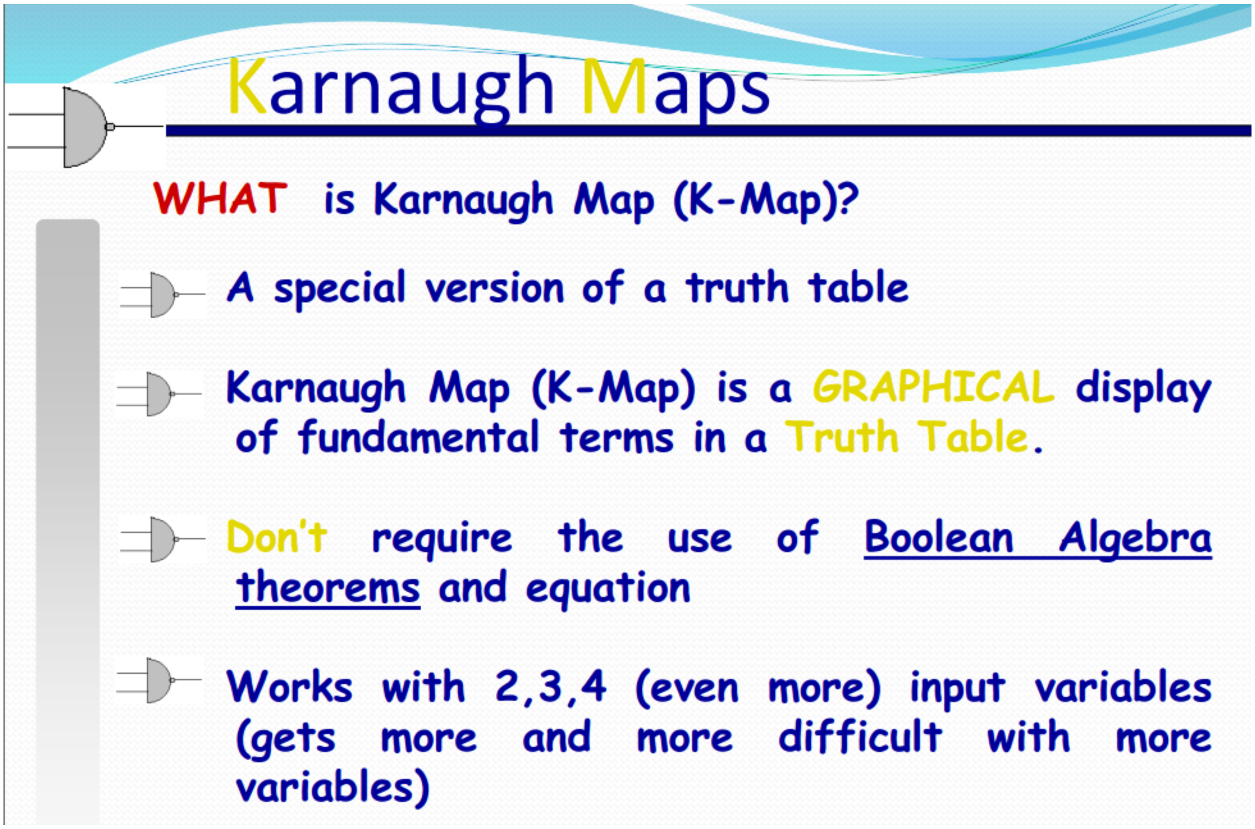
$$F' = (A+B+C')(A+B'+C')(A'+B+C')(A'+B'+C)(A'+B'+C')$$

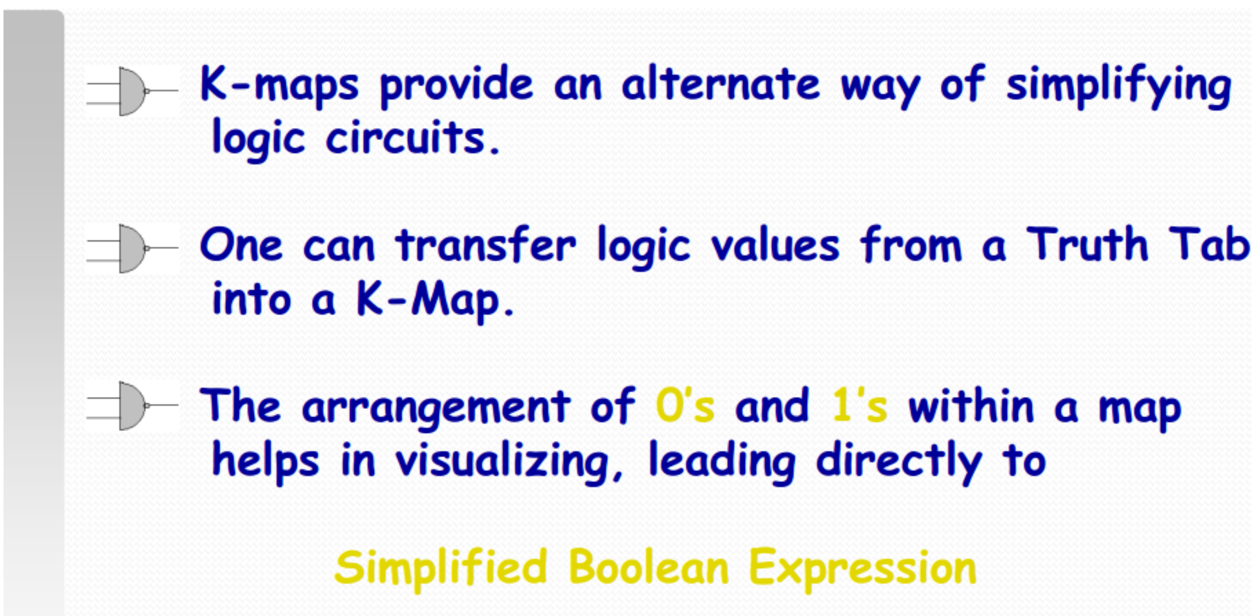
Difference between SOP and POS :

S.No.	SOP	POS
1.	A way of representing boolean expressions as sum of product terms.	A way of representing boolean expressions as product of sum terms.
2.	SOP uses minterms. Minterm is product of boolean variables either in normal form or complemented form.	POS uses maxterms. Maxterm is sum of boolean variables either in normal form or complemented form.
3.	It is sum of minterms. Minterms are represented as 'm'	It is product of maxterms. Maxterms are represented as 'M'
4.	SOP is formed by considering all the minterms, whose output is HIGH(1)	POS is formed by considering all the maxterms, whose output is LOW(0)
5.	While writing minterms for SOP, input with value 1 is considered as the variable itself and input with value 0 is considered as complement of the input.	While writing maxterms for POS, input with value 1 is considered as the complement and input with value 0 is considered as the variable itself.



6. Karnaugh Map





Drawing a Karnaugh Map (K-Map)

- K-map is a rectangle made up of certain number of SQUARES
- For a given Boolean function there are 2^N squares where N is the number of variables (inputs)
- In a K-Map for a Boolean Function with 2 Variables *f(a,b)* there will be $2^2=4$ squares
- Each square is different from its neighbour by ONE Literal
- Each SQUARE represents a MAXTERM or MINTERM

Karnaugh maps consist of a set of 2^2 squares where 2 is the number of variables
in the Boolean expression being minimized.

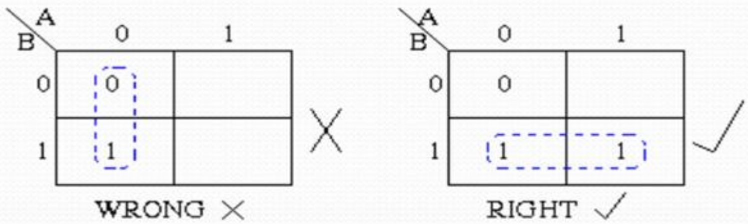
A	B	F	Minterm	Maxterm
0	0	0	A'B'	A + B
0	1	1	A'B	A + B'
1	0	1	A B'	A' + B
1	1	1	A B	A' + B'

Truth Table

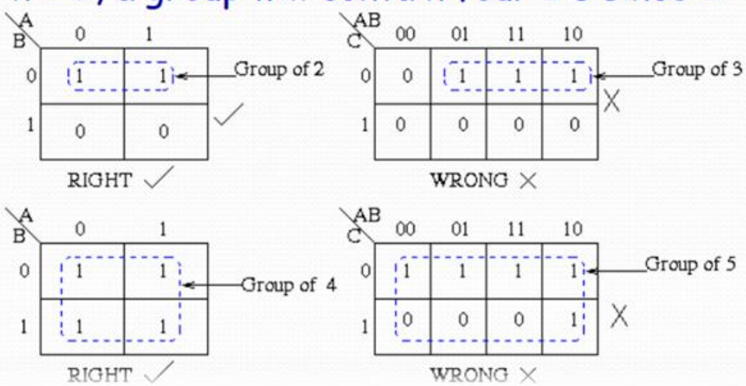
A \ B	0	1
0	0	1
1	1	1

2 Variable K-Map

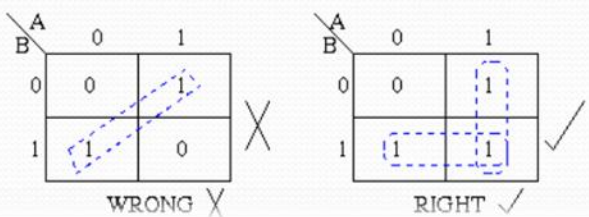
- **Groups may not include any cell containing a zero**



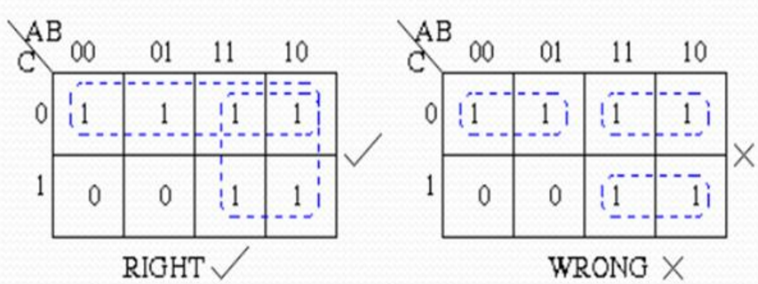
- Groups must contain 1, 2, 4, 8, or in general 2^n cells.
- That is if n = 1, a group will contain two 1's since $2^1 = 2$.
- If n = 2, a group will contain four 1's since $2^2 = 4$.



Groups may be horizontal or vertical, but not diagonal.

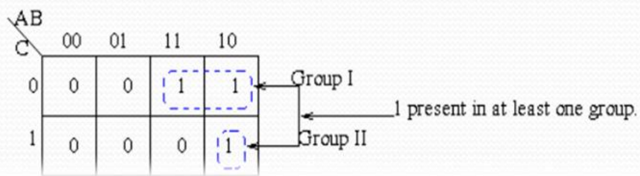


- Each group should be as large as possible.

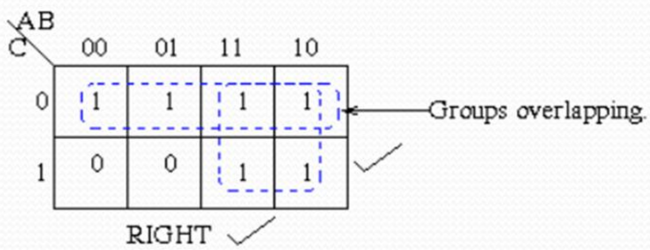


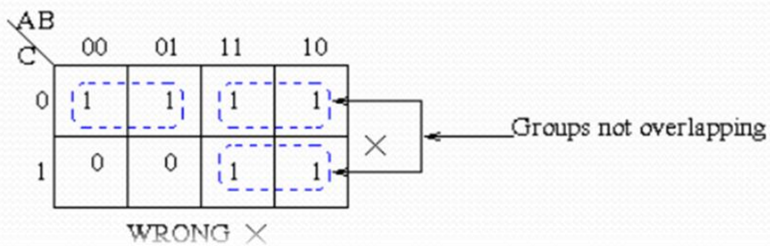
(Note that no Boolean laws broken, but not sufficiently minimal)

- Each cell containing a 1 must be in at least one group.

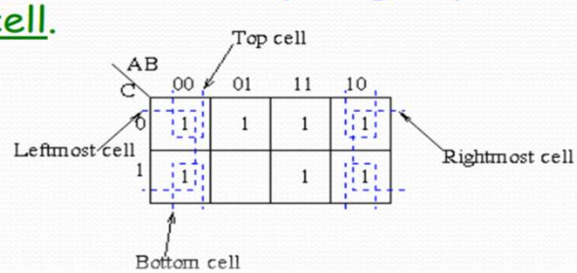


- Groups may overlap.

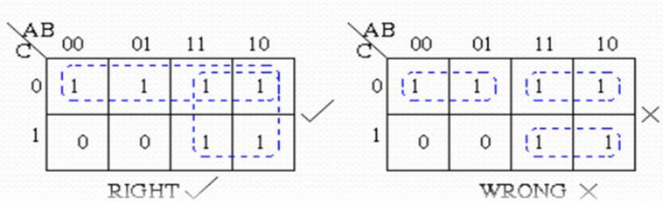




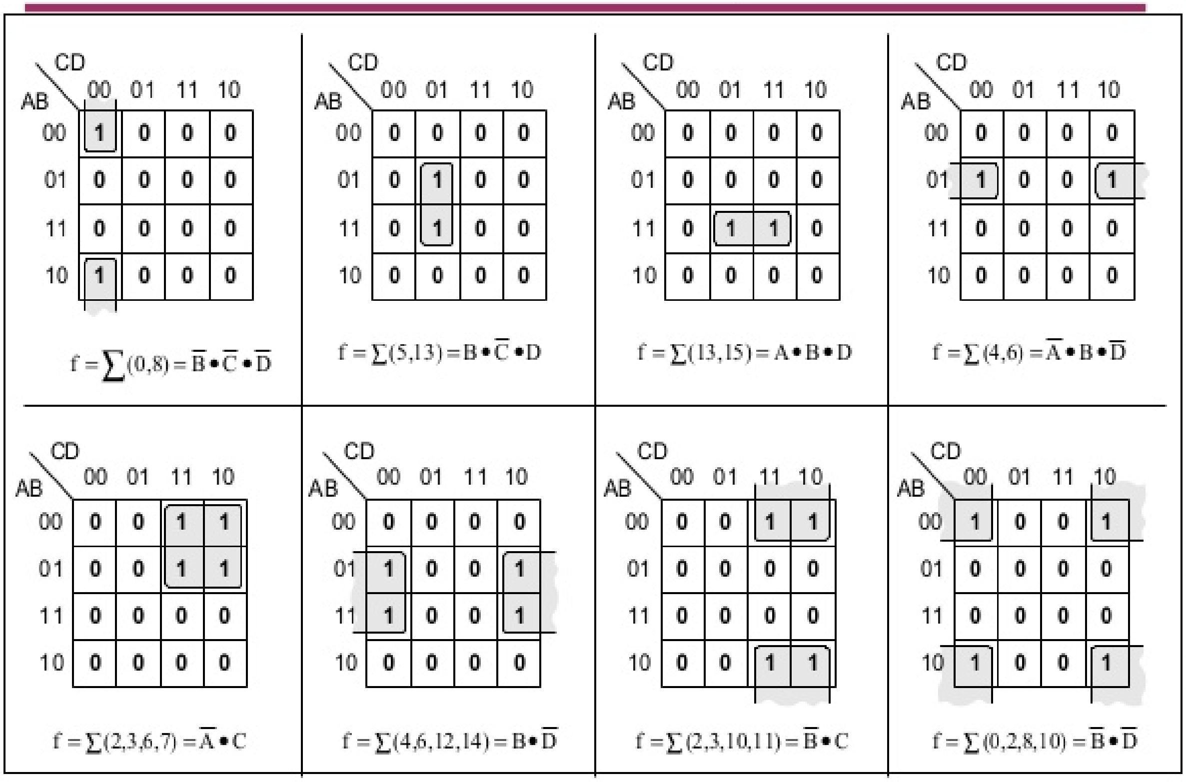
- Groups may wrap around the table.
- The leftmost cell in a row may be grouped with the rightmost cell and
- The top cell in a column may be grouped with the bottom cell.



- There should be as few groups as possible, as long as this does not contradict any of the previous rules.



Four-Variable K-Maps



7. Lab Task

Given the truth table, find the simplified SOP and POS form.

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Consider the following Boolean expression for the remaining experiments

$$F(x, y, z) = xy + yz + xy'z$$

3. Make truth table for the expression and implement the circuit using AND, OR, and NOT Gates.
4. Reduce the expression using 3-variable K-Map. Make the truth table for the reduced expression. Implement the circuit using AND, OR, and NOT Gates. Verify that the results of experiment 3 and 4 are same.
5. Provide NAND-ONLY implementation for the reduced expression from Exp 4. Prove that the results from Exp 4 and 5 match.
6. Provide NOR-ONLY implementation for the reduced expression from Exp 4. Prove that the results from Exp 4 and 6 match.

- F1 = A'B'+AB'C+A'BC+AB'+C+AB'+C'+AB
- F2 = AB'C'D+AC'D+C'D+C'D+A'B+AD
- F3 = AB+A'C'D+A'BD+AB'C'
- F4 = C + S where C = xy + yz and S = C' (x + y) + xyz
- F5 = F3 + F1 (F4)

Your Task is

1. Design Truth tables for all the above Boolean Functions.
2. Simplify the given functions using KMAP.