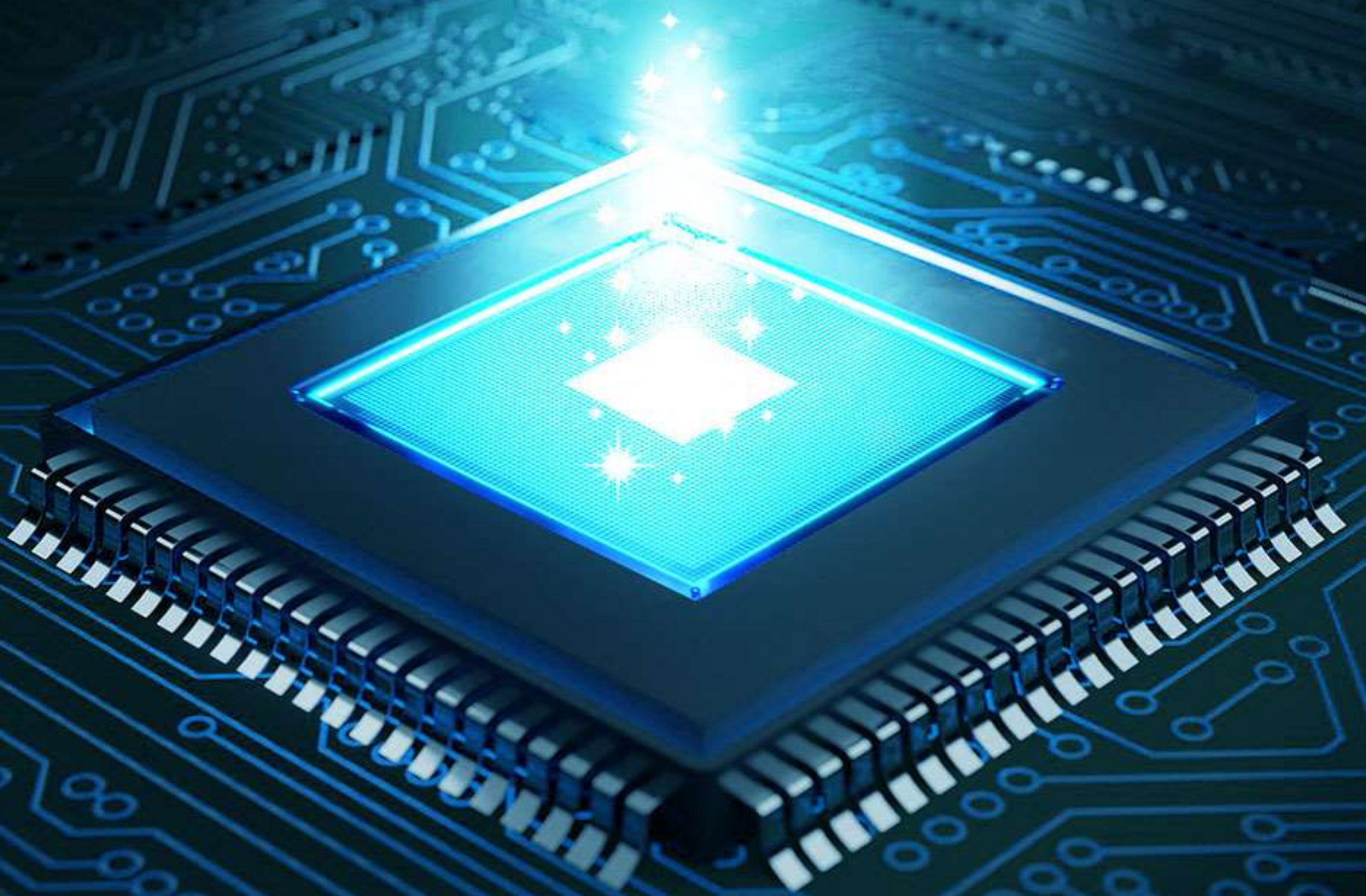
Digital Electronics and Logic Design



Unit I
Minimization Technique

Ms. Bhagyashri More, Assistant Professor ,Department of Computer Engineering, MESCOE, Pune

Agenda

01 **Logic Design Minimization Technique-** Minimization of Boolean function using K-map(up to 4 variables) and Quine Mc-Clusky Method

02 **Representation of signed number-** sign magnitude representation ,1's complement and 2's complement form

03 **Sum of product and product of sum**

04 **Minimization of SOP and POS using k-map**

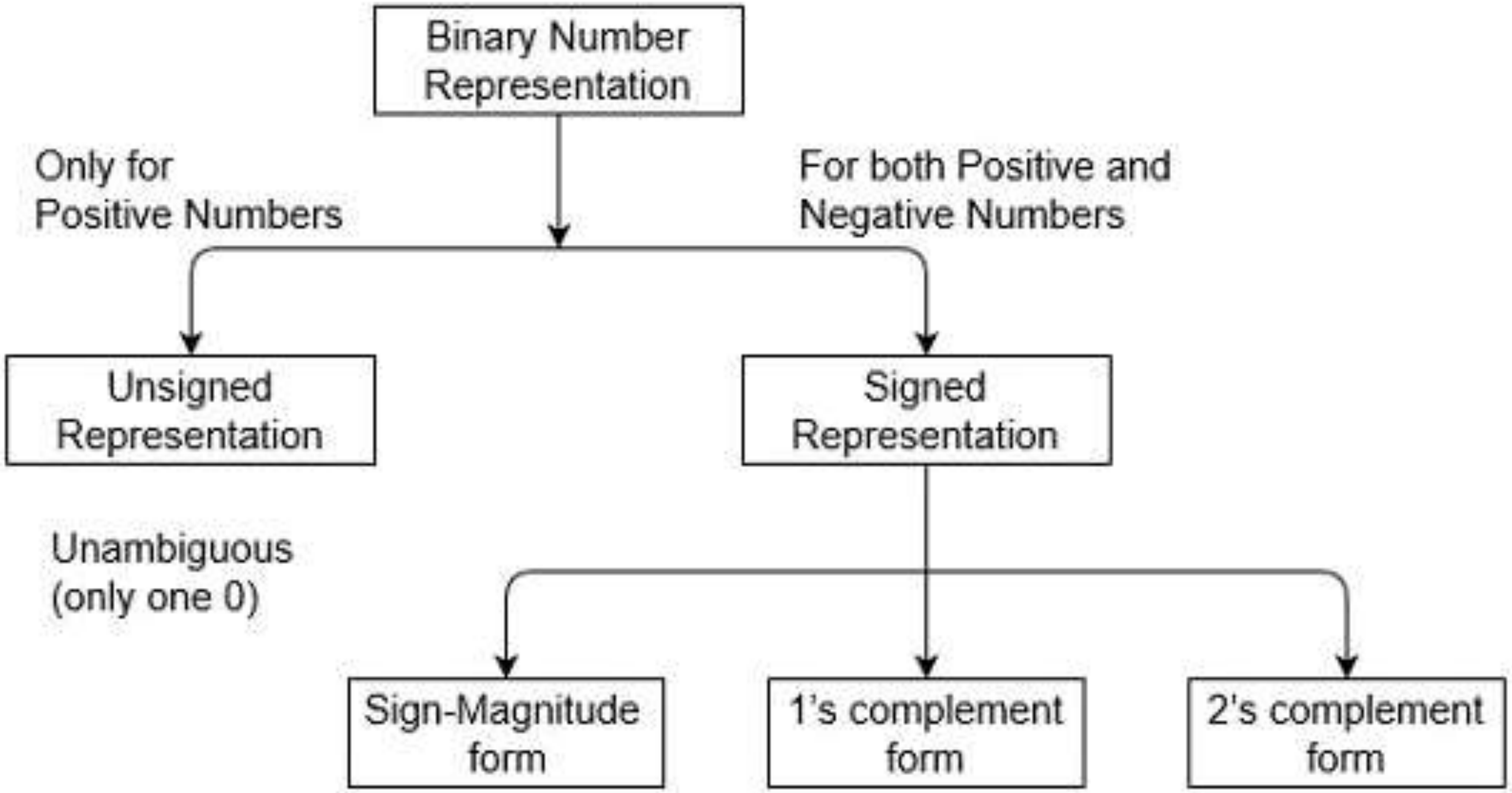
Representation of signed binary number

Signed Numbers
Signed numbers contain both sign and magnitude of the number. Generally, the sign is placed in front of number. So, we have to consider the positive sign for positive numbers and negative sign for negative numbers.

- There are three **types of representations** for signed binary numbers:

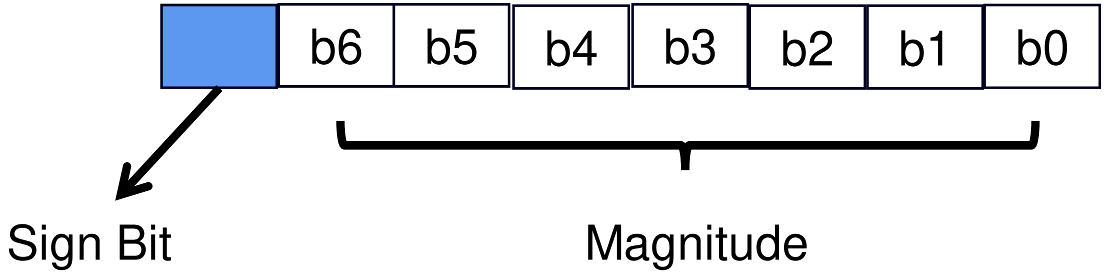
1. Sign-Magnitude form
2. 1's complement form
3. 2's complement form

Representation of signed binary number

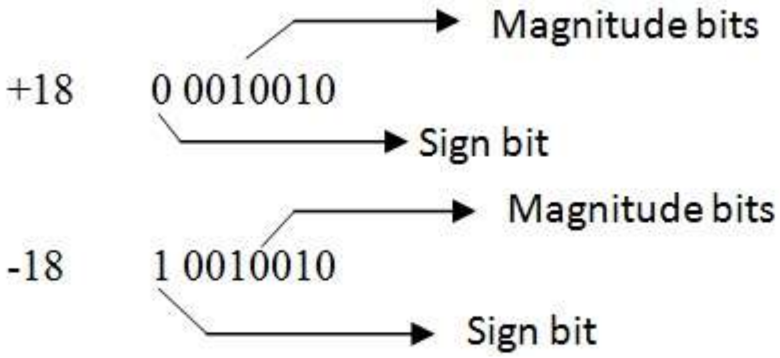


Representation of signed binary number

1. **Signed Magnitude:**
   Signed numbers contain sign flag, this representation distinguish positive and negative numbers. This technique contains both sign bit and magnitude of a number. For example, in representation of negative decimal numbers, we need to put negative symbol in front of given decimal number.



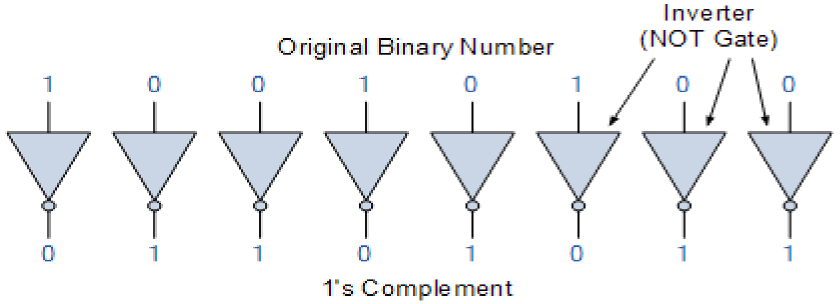
- Positive number is represented with ‘0’ at its most significant bit (MSB).
- Negative number is represented with ‘1’ at its most significant bit (MSB).



Representation of signed binary number

2. One's Complement of a Signed Binary Number

The 1's complement of a number is obtained by **complementing all the bits** of signed binary number. So, 1's complement of positive number gives a negative number. Similarly, 1's complement of negative number gives a positive number.



1. How to represent $(-33)_{10}$ in 1's complement form?

33 is represented as $(100001)_2$
In 8 bit notation, it is represented as $(0010\ 0001)_2$
Now, -33 is represented in one's compliment as $(1101\ 1110)_2$

2. How to represent $(-1)_{10}$ in 1's complement form?

1 is represented as $(001)_2$
In 8 bit notation, it is represented as $(0000\ 0001)_2$
Now, -1 is represented in one's compliment as $(1111\ 1110)_2$

Representation of signed binary number

3. Two’s Complement of a Signed Binary Number

To get 2’s complement of a binary number, simply invert the given number and add 1 to the least significant bit (LSB) of given result.

Example:

1. Find 2’s complement of binary number 10101110.

   Simply invert each bit of given binary number, which will be 01010001. Then add 1 to the LSB of this result, i.e., 01010001+1=01010010 which is answer.

Direct conversion

1. LSB with 0

10101101000

$\downarrow$

01010011000

2. MSB with 1

11001110011

$\downarrow$

00110001101

Signed binary number examples

Examples:

1. **Represent (-12) in,**

a. 8 bit signed magnitude form
b. 1's complement form
c. 2's complement form

2. Find out 2's complement of.

a. 1110101110101
b. 000101010010
c. 101010101000

Boolean Algebra Terminology

Boolean Algebra: It is mathematical system that defines a series of logical operations (AND, OR, NOT, etc) perform on set of variables (A,B,C, etc). Only two values(1 for high and 0 for low) are possible for the variable used in Boolean algebra.

Example:

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

- **Variables**
- **Constant**
- **Complement**
- **Literals**
- **Boolean Function**

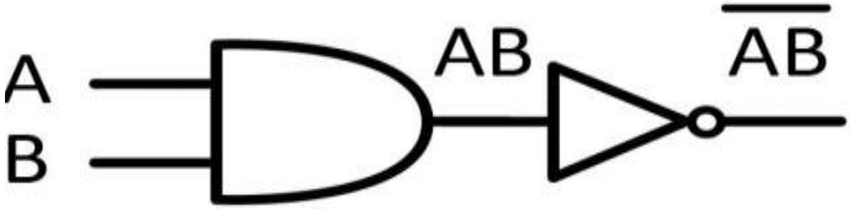
Boolean Expression = Variables + Constants + Boolean operations

We use Boolean Expression to describe Boolean Function.

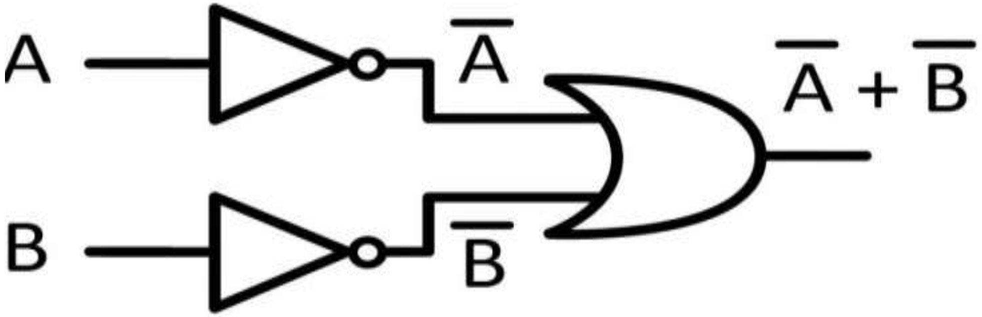
Properties of Boolean algebra

Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\bar{A} = 0$	$A + \bar{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan's law	$\overline{AB} = \bar{A} + \bar{B}$	$\overline{A + B} = \bar{A}\bar{B}$

Properties of Boolean algebra- DeMorgan's Theorem

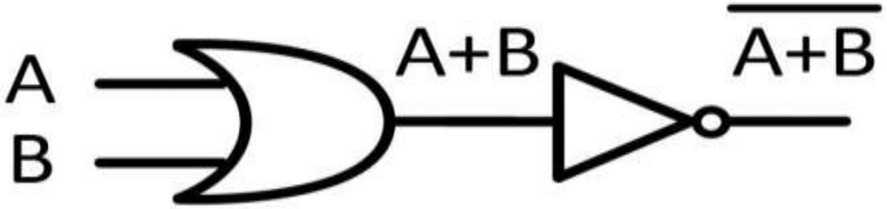


is equivalent to

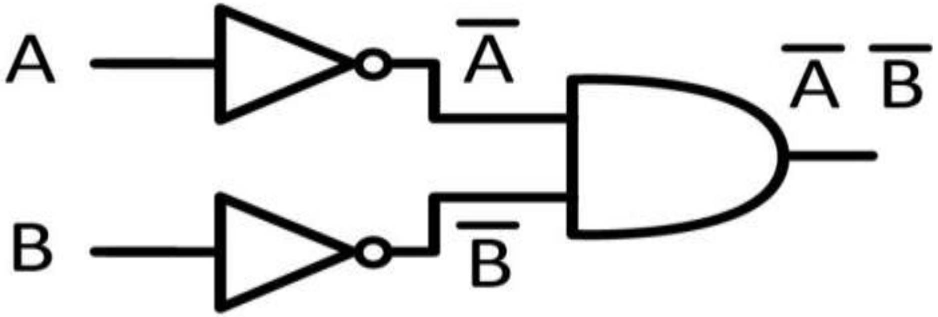


$$\overline{AB} = \overline{A} + \overline{B}$$

NAND = Bubbled OR



is equivalent to



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

NOR = Bubbled AND

Properties of Boolean algebra- DeMorgan's Theorem

Verification of De-Morgan's Theorem

Inputs		Truth Table Outputs For Each Term				
B	A	A.B	$\overline{A.B}$	$\overline{A}$	$\overline{B}$	$\overline{A} + \overline{B}$
0	0	0	1	1	1	1
0	1	0	1	0	1	1
1	0	0	1	1	0	1
1	1	1	0	0	0	0

Inputs		Truth Table Outputs For Each Term				
B	A	A+B	$\overline{A+B}$	$\overline{A}$	$\overline{B}$	$\overline{A} . \overline{B}$
0	0	0	1	1	1	1
0	1	1	0	0	1	0
1	0	1	0	1	0	0
1	1	1	0	0	0	0

Boolean algebra Examples

Examples:

- **Prove the following using De Morgans's Theorem**

1. **AB + CD = ((AB)' + (CD)')'**
2. **(A+B) . (C+D) = ((A+B)' + (C+D)')'**

- **Minimize the expression**

1. **Y = A'BCD' + BCD' + BC'D' + BC'D**
2. **F = A + A'B + AB'**
3. **X = A.(A'+B)**

Boolean Expression

Based on the structure of Boolean expression, it is classified into two categories.

1. Sum Of Product (SOP)
2. Product Of Sum (POS)

SOP	POS
F = AB + BC + AC product term **Standard / Canonical SOP** F(A,B,C) = ABC' + ABC + AB'C minterm Each individual product term in standard SOP is called minterm.	F = (A+B) *(A + B + C) * (C +D) sum term **Standard / Canonical POS** F = (A + B' + C) *(A' + B + C) * (A + B' + C) Maxterm Each individual sum term in standard POS is called maxterm.

Conversions of Canonical Forms

Conversion of SOP form to standard SOP form or Canonical SOP form

Step 1:

By multiplying each non-standard product term with the sum of its missing variable and its complement, which results in 2 product terms

Step 2:

By repeating the step 1, until all resulting product terms contain all variables

Example:

Convert the non standard SOP function F = x y + x z + y z

Sol:

$$
\begin{aligned}
F &= x\,y + x\,z + y\,z \\
&= x\,y\,(z + z') + x\,(y + y')\,z + (x + x')\,y\,z \\
&= x\,y\,z + x\,y\,z' + x\,y\,z + x\,y'\,z + x\,y\,z + x'\,y\,z \\
&= x\,y\,z + x\,y\,z' + x\,y'\,z + x'\,y\,z
\end{aligned}
$$

The standard SOP form is F = x y z + x y z' + x y' z + x' y z

Conversions of Canonical Forms

Conversion of POS form to standard POS form or Canonical POS form

Step 1:

By adding each non-standard sum term to the product of its missing variable and its complement, which results in 2 sum terms

Step 2:

Applying Boolean algebraic law, A + BC = (A + B) * (A + C)

Step 3:

By repeating the step 1, until all resulting sum terms contain all variables

Example:

F = (A’ + B + C) * (B’ + C + D’) * (A + B’ + C’ + D)

In the first term, the variable D or D’ is missing, so we add D*D’ = 0 to it. Then

(A’ + B + C + D*D’) = (A’ + B + C + D) * (A’ + B + C + D’)

Similarly, in the second term, the variable A or A’ is missing, so we add A*A’ = 0 to it. Then

(B’ + C + D’ + A*A’) = (A + B’ + C + D’) * (A’ + B’ + C + D’)

The third term is already in the standard form, as it has all the variables. Now the standard POS form equation of the function is

F = (A’ + B + C + D) * (A’ + B + C + D’) * (A + B’ + C + D’) * (A’ + B’ + C + D’) * (A + B’ + C’ + D)

Conversions of Canonical Forms

Convert of SOP form to standard SOP form or Canonical SOP form

Y= ABC + AB + BC’ + A

Z= ABC + ABD + BCD’ +AC’D’

X= A + B

Convert of POS form to standard POS form or Canonical POS form

Y= (A+B) (B+C) (A+C)

Z= (A+B+C+D) (A’+C’+D) (B+C’+D’)

M Notations: Minterm and Maxterm

SOP	POS
A variable is in complemented form, if its value is assigned to 0, and the variable is un-complimented form, if its value is assigned to 1.	In max term, each variable is complimented, if its value is assigned to 1, and each variable is un-complimented if its value is assigned to 0.

Variables			Min terms	Max terms
A	B	C	m_i	M_i
0	0	0	A' B' C' = m 0	A + B + C = M 0
0	0	1	A' B' C = m 1	A + B + C' = M 1
0	1	0	A' B C' = m 2	A + B' + C = M 2
0	1	1	A' B C = m 3	A + B' + C' = M 3
1	0	0	A B' C' = m 4	A' + B + C = M 4
1	0	1	A B' C = m 5	A' + B + C' = M 5
1	1	0	A B C' = m 6	A' + B' + C = M 6
1	1	1	A B C = m 7	A' + B' + C' = M 7

Standard Notation of SOP and POS

Example:

The SOP function,

1. F(A,B,C) = A’B’C’ + A’B’C + AB’C’ + ABC’
   0 1 4 6
   = $m_0 + m_1 + m_4 + m_6$
   F(A,B,C) = $\sum m\,(0, 1, 4, 6)$

2. F(A,B,C) = $\sum m\,(0, 2, 5, 6, 7)$
   = $m_0 + m_2 + m_5 + m_6 + m_7$
   = A’B’C’ + A’BC’ + AB’C + ABC’ + ABC

Exercise:

1. F = $\sum m\,(1,3,4,6,7)$
2. F = $\sum m\,(2,4,5)$
3. F = A’B’C + A’BC + AB’C + ABC’

The POS function,

1. F(A,B,C) = (A+B+C)*(A+B’+C)*(A+B’+C’)*(A’+B’+C)
   0 2 3 6
   = M0 * M2 * M3 * M6
   F(A,B,C) = Π M (0, 2, 3, 6)

2. F(A,B,C) = Π M (4,5,6,7)
   = M4 * M5 * M6 * M7
   = (A’+B+C) (A’+B+C’) (A’+B’+C) (A’+B’+C’)

Exercise:

1. F = Π M (1,3,5,7)
2. F = Π M (2,4,6)
3. F = (A+B+C) (A+B+C’) (A+B’+C) (A+B’+C’)

Construct SOP and POS From a Truth Table

For SOP Expression,

1. A circuit for a truth table with N input columns can use AND gates with N inputs, and each row in the truth table with a '1' in the output column requires one N-input AND gate.
2. Inputs to the AND gate are inverted if the input shows a '0' on the row, and not inverted if the input shows a '1' on the row.
3. All AND terms are connected to an M-input OR gate, where M is the number of '1' output rows.

The output of the OR gate is the function output.

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

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

Construct SOP and POS From a Truth Table

For POS Expression,

1. A circuit for a truth table with N input columns can use OR gates with N inputs, and each row in the truth table with a '0' in the output column requires one N-input OR gate.
2. Inputs to the OR gate are inverted if the input shows a '1' on the row, and not inverted if the input shows a '0' on the row.
3. All OR terms are connected to an M-input AND gate, where M is the number of '1' output rows.

The output of the AND gate is the function output

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

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

Construct SOP and POS From a Truth Table

Min and Max Term for given example

A	B	C	#	Minterm	Maxterm	F
0	0	0	0	$\overline{A}\cdot\overline{B}\cdot\overline{C}$	$A+B+C$	0
0	0	1	1	$\overline{A}\cdot\overline{B}\cdot C$	$A+B+\overline{C}$	1
0	1	0	2	$\overline{A}\cdot B\cdot\overline{C}$	$A+\overline{B}+C$	0
0	1	1	3	$\overline{A}\cdot B\cdot C$	$A+\overline{B}+\overline{C}$	1
1	0	0	4	$A\cdot\overline{B}\cdot\overline{C}$	$\overline{A}+B+C$	0
1	0	1	5	$A\cdot\overline{B}\cdot C$	$\overline{A}+B+\overline{C}$	1
1	1	0	6	$A\cdot B\cdot\overline{C}$	$\overline{A}+\overline{B}+C$	0
1	1	1	7	$A\cdot B\cdot C$	$\overline{A}+B+\overline{C}$	0

1. Write down SOP and POS expression from given truth tables

A	B	Y
0	0	**1**
0	1	**1**
1	0	**0**
1	1	**1**

A	B	C	Y
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	**0**

A	B	C	D	Y
0	0	0	0	**0**
0	0	0	1	**1**
0	0	1	0	**0**
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	**0**
1	1	0	0	**0**
1	1	0	1	**1**
1	1	1	0	**0**
1	1	1	1	**1**

2. From following expression draw the truth table,
   a. Y= A'B' + A'B + AB
   b. F= A'B'C' + A'B'C + ABC
   c. Y = (A+B) (A'+B)
   d. Y= (A+B'+C+D') (A+B'+C+D) (A+B'+C'+D') (A+B+C+D') (A'+B'+C+D') (A'+B'+C'+D') (A+B+C+D)

Agenda

01 **Logic Design Minimization Technique- Minimization of Boolean function using K-map(up to 4 variables) and Quine Mc-Clusky Method**

02 **Representation of signed number-** sign magnitude representation ,1's complement and 2's complement form

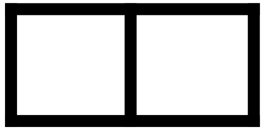
03 **Sum of product and product of sum**

04 **Minimization of SOP and POS using k-map**

Karnaugh Maps (K-map)

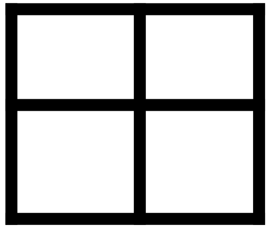
- A K-map is a collection of squares
- Each square represents a minterm
- The collection of squares is a graphical representation of a Boolean function
- Adjacent squares differ in the value of one variable
- Alternative algebraic expressions for the same function are derived by recognizing patterns of squares (corresponding to cubes)

1 variable



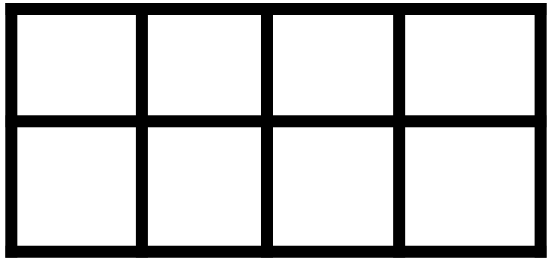
2 cells

2 variable



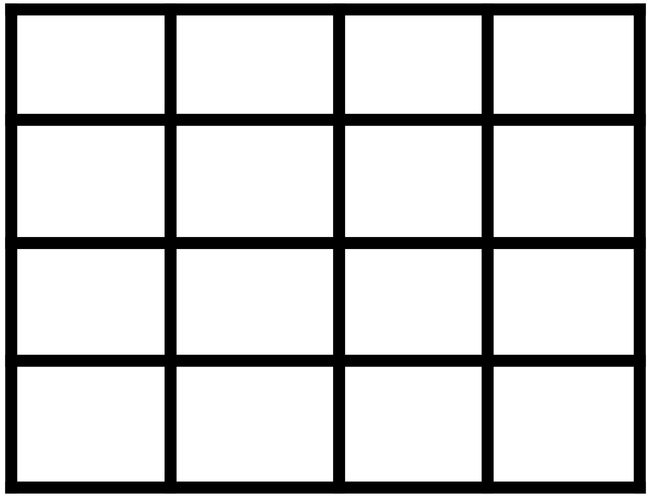
4 cells

3 variable



8 cells

4 variable



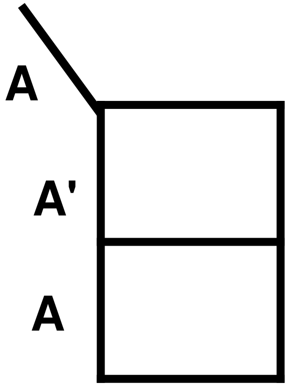
16 cells

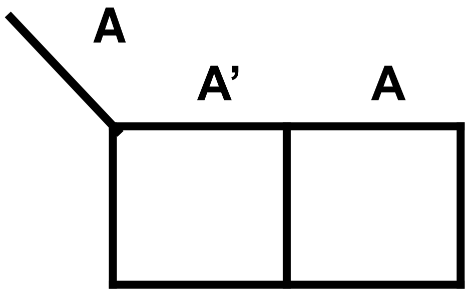
Karnaugh Maps (K-map)

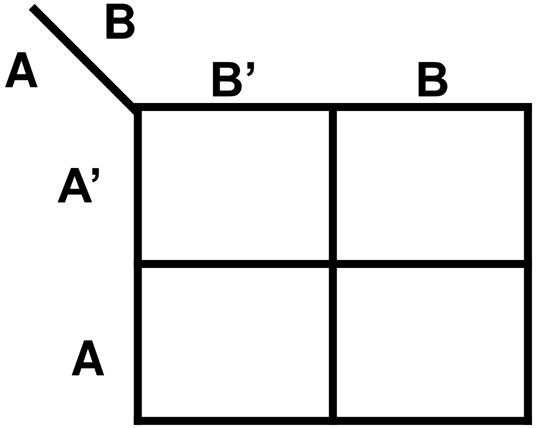
- Relation between number of variables and outline of kmap

Number of kmap cells = 2 ^ n ....n = no. of input variables

One variable = 2^1 = 2 cells (A)	Two variable = 2^2 = 4 cells (A,B)







Karnaugh Maps (K-map)

- Relation between number of variables and outline of kmap

Number of kmap cells = 2 ^ n ………….n = no. of input variables

Three variable = 2^3 = 8 cells (A,B,C)	Four variable = 2^4 = 16 cells (A,B,C,D)

A \ BC	B'C'	B'C	BC	BC'
A'				
A				

AB \ C	C'	C
A'B'		
A'B		
AB		
AB'		

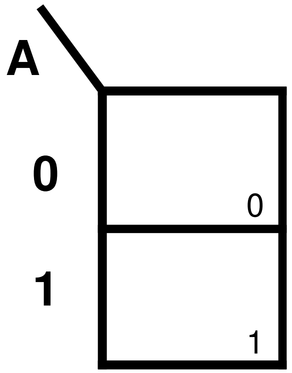
AB \ CD	C'D'	C'D	CD	CD'
A'B'				
A'B				
AB				
AB'				

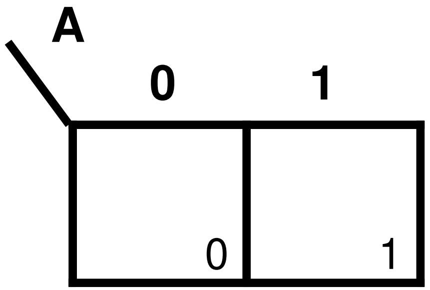
Karnaugh Maps (K-map)

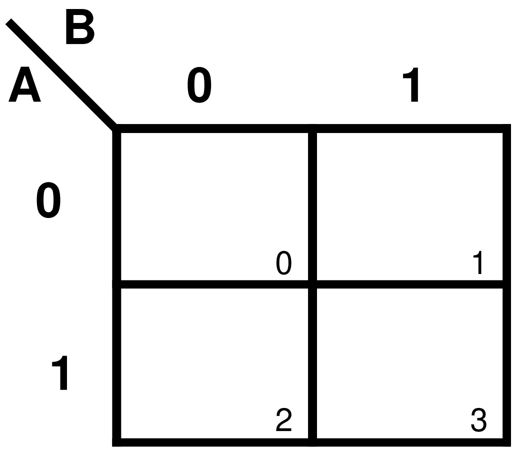
- **Relation between number of variables and outline of kmap**

Number of kmap cells = 2 ^ n ………….n = no. of input variables

One variable = 2^1 = 2 cells	Two variable = 2^2 = 4 cells







Karnaugh Maps (K-map)

- Relation between number of variables and outline of kmap

Number of kmap cells = 2 ^ n …………n = no. of input variables

Three variable = 2^3 = 8 cells (A,B,C)	Four variable = 2^4 = 16 cells (A,B,C,D)

A \ BC	00	01	11	10
0	0	1	3	2
1	4	5	7	6

AB \ C	0	1
00	0	1
01	2	3
11	6	7
10	4	5

AB \ CD	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

K-map for SOP

A	0	1
	A’	A

A	
0	A’
1	A

A \ B	0	1
0	A’B’	A’B
1	AB’	AB

A \ BC	00	01	11	10
0	A’B’C’	A’B’C	A’BC	A’BC’
1	AB’C’	AB’C	ABC	ABC’

AB \ C	0	1
00	A’B’C	A’B’C
01	A’BC’	A’BC
11	ABC’	ABC
10	AB’C’	AB’C

AB \ CD	00	01	11	10
00	A’B’C’D’	A’B’C’D	A’B’CD	A’B’CD’
01	A’BC’D’	A’BC’D	A’BCD	A’BCD’
11	ABC’D’	ABC’D	ABCD	ABCD’
10	AB’C’D’	AB’C’D	AB’CD	AB’CD’

K-map for POS

A	0	1
	A	A’

A	
0	A
1	A’

A \ B	0	1
0	A +B	A+B’
1	A’+B	A’+B’

A \ BC	00	01	11	10
0	A+B+C	A+B+C’	A+B’+C’	A+B’+C
1	A’+B+C	A’+B+C’	A’+B’+C’	A’+B’+C

AB \ C	0	1
00	A+B+C	A+B+C’
01	A+B’+C	A+B’+C’
11	A’+B+C	A’+B+C’
10	A’+B+C	A’+B+C’

AB \ CD	00	01	11	10
00	A+B+C+D			
01			A+B’+C’+D’	
11		A’+B’+C+D’		
10				A’+B+C’+D

Kmap Ploting from Truth Table

For 1 variable

A	Y
0	1
1	0

A	0	1
	1	0

A	
0	1
1	0

For 2 variables

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

A \ B	0	1
0	0	1
1	0	1

Kmap Ploting from Truth Table

For 3 variables

A	B	C	Y
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	**0**

A \ BC	00	01	11	10
0	1 $_0$	0 $_1$	0 $_3$	1 $_2$
1	1 $_4$	0 $_5$	0 $_7$	1 $_6$

AB \ C	0	1
00	1 $_0$	0 $_1$
01	1 $_2$	0 $_3$
11	1 $_6$	0 $_7$
10	1 $_4$	0 $_5$

Kmap Ploting from Truth Table

	A	B	C	D	Y
0	0	0	0	0	**0**
1	0	0	0	1	**1**
2	0	0	1	0	**0**
3	0	0	1	1	**1**
4	0	1	0	0	**0**
5	0	1	0	1	**0**
6	0	1	1	0	**1**
7	0	1	1	1	**1**
8	1	0	0	0	**0**
9	1	0	0	1	**0**
10	1	0	1	0	**1**
11	1	0	1	1	**0**
12	1	1	0	0	**0**
13	1	1	0	1	**1**
14	1	1	1	0	**0**
15	1	1	1	1	**1**

AB \ CD	**00**	**01**	**11**	**10**
00	0_{0}	1_{1}	1_{3}	0_{2}
01	0_{4}	0_{5}	1_{7}	1_{6}
11	0_{12}	1_{13}	1_{15}	0_{14}
10	0_{8}	0_{9}	0_{11}	1_{10}

Simplification of Boolean Expression using k-map

- It is based on combining or grouping the terms in the adjacent cells of the kmap.
- **Way of grouping**

- We can group 1,2,4,8,16,32…..number of 1's or 0's.
- We can not group 3,5,7,…. number of 1's or 0's.

1. **Pair** : A group of two adjacent 1's or 0's.
2. **Quad** : A group of four adjacent 1's or 0's.
3. **Octet** : A group of eight adjacent 1's or 0's.

Rules for k-map grouping

- In grouping no 0s allowed for SOP and no 1s allowed for POS.
- Diagonal grouping not allowed.
- Only power of 2 numbers of cells in each group.
- Group should be as large as possible.
- Every 1 or 0 must be in at least one group.
- Overlapping of groups allowed.
- Wrap around is allowed.
- Number of groups should be as few as possible.

K-map Grouping

1. **Pair** : A group of two adjacent 1’s.

A \ B	0	1
0	0	1
1	0	1

B

Y= B

A \ BC	00	01	11	10
0	0	1	0	0
1	0	1	1	1

B’C

AB

Y= B’C +AB

B’C’D’

BC’D

A’BC

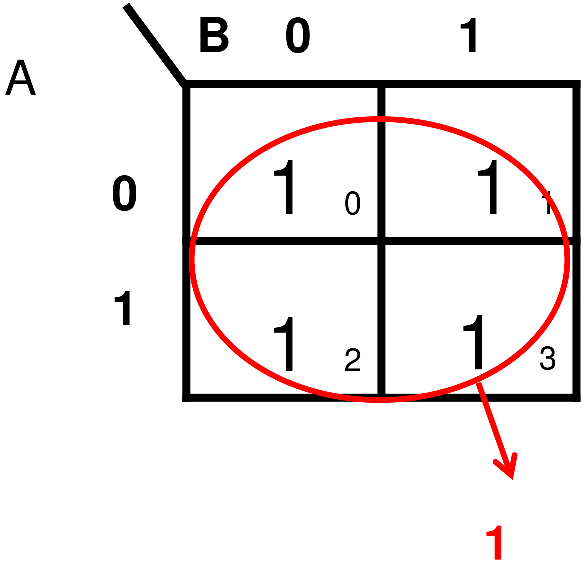
AB \ CD	00	01	11	10
00	1	0	0	0
01	0	1	1	1
11	0	1	0	0
10	1	0	0	1

AB’D’

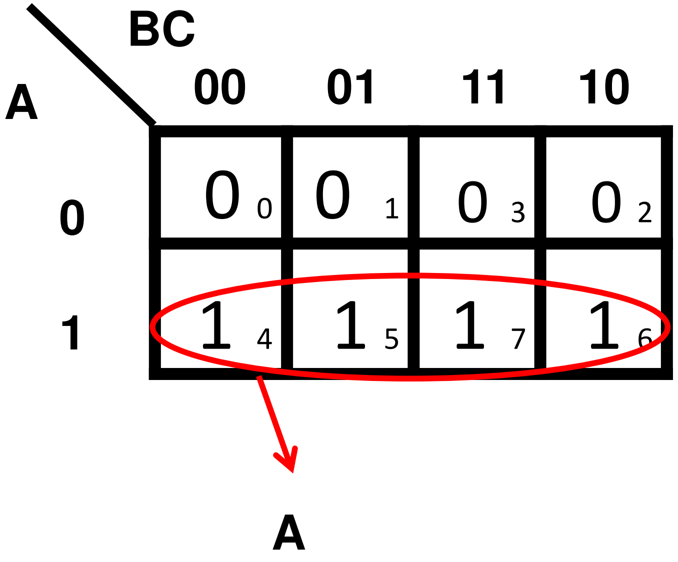
Y= AB’D’ + B’C’D’ + BC’D + A’BC

K-map Grouping

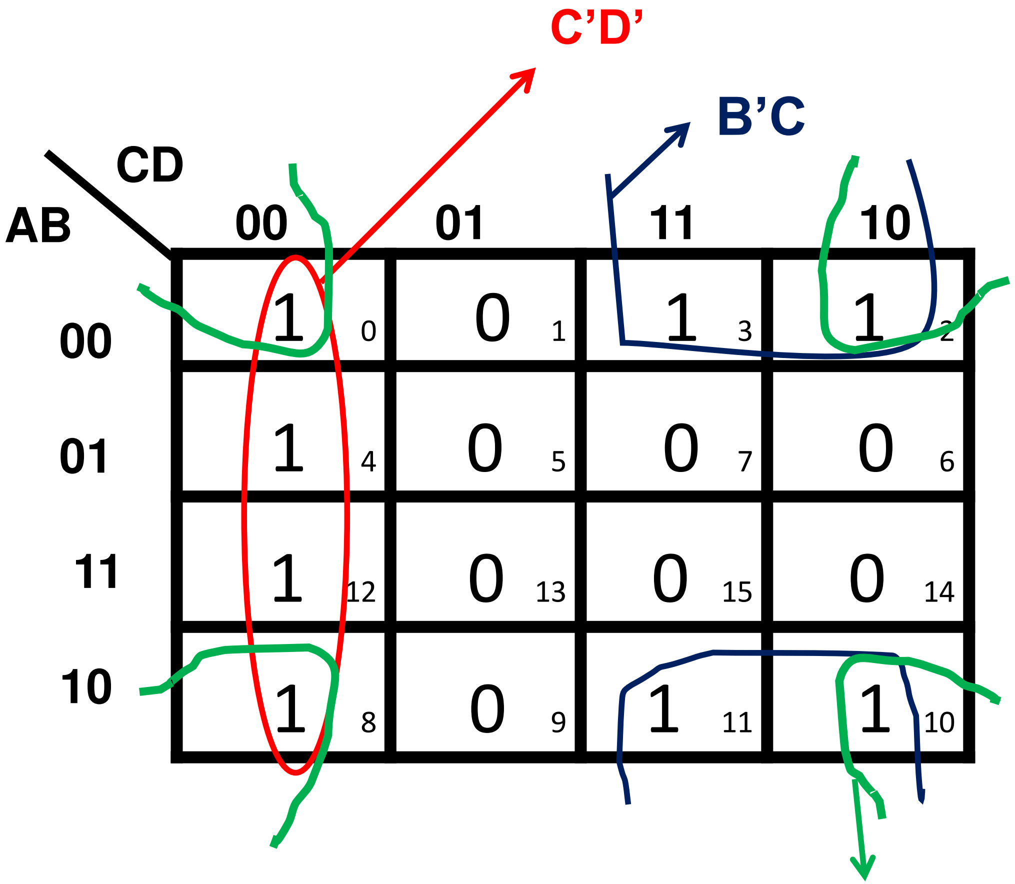
2. **Quad** : A group of four adjacent 1's.



Y = 1



Y = A



Y = C'D' + B'C + B'D'

B'D'

K-map Grouping

3. **Octet** : A group of eight adjacent 1's.

A \ BC	00	01	11	10
0	1 (0)	1 (1)	1 (3)	1 (2)
1	1 (4)	1 (5)	1 (7)	1 (6)

1

Y = 1

D'

C

AB \ CD	00	01	11	10
00	1 (0)	0 (1)	1 (3)	1 (2)
01	1 (4)	0 (5)	1 (7)	1 (6)
11	1 (12)	0 (13)	1 (15)	1 (14)
10	1 (8)	0 (9)	1 (11)	1 (10)

Y = C + D'

Minimization of SOP Expression

Example 1: Minimize the expression Y = A'B'C' + A'BC + AB'C' + AB'C using K-map method and draw the diagram.
Ans:

Y = A'B'C' + A'BC + AB'C' + AB'C

As it comprises 3 variables, we need a 3 variable K-map or 8 cells K-map. In the given expression the minterms are,

A'B'C' = 0 0 0 = m_0
A'BC = 0 1 1 = m_3
AB'C' = 1 0 0 = m_4
AB'C = 1 0 1 = m_5

The expression is ,

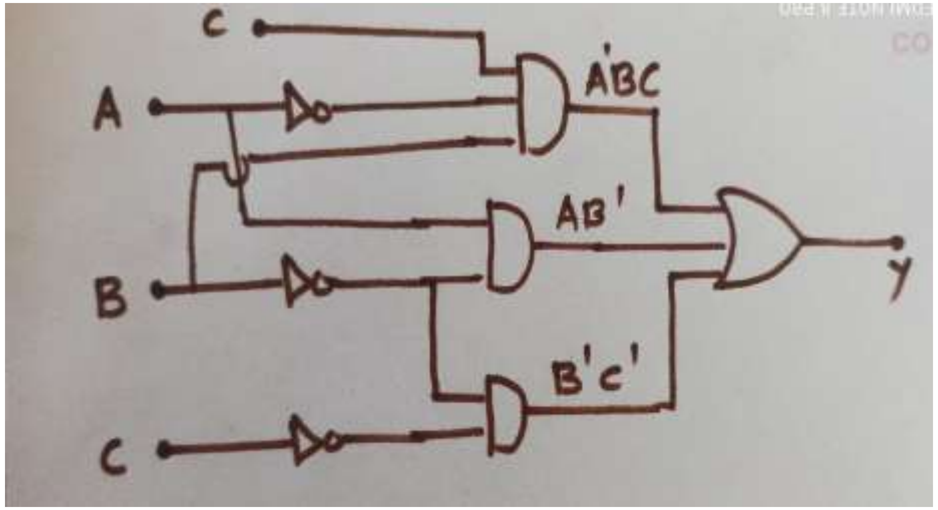
Y = $\sum$ m (0, 3, 4, 5)

B'C'

A'BC

A \ BC	00	01	11	10
0	1_0	0_1	1_3	0_2
1	1_4	1_5	0_7	0_6

AB'



Y = B'C' + AB' + A'BC

Minimization of SOP Expression

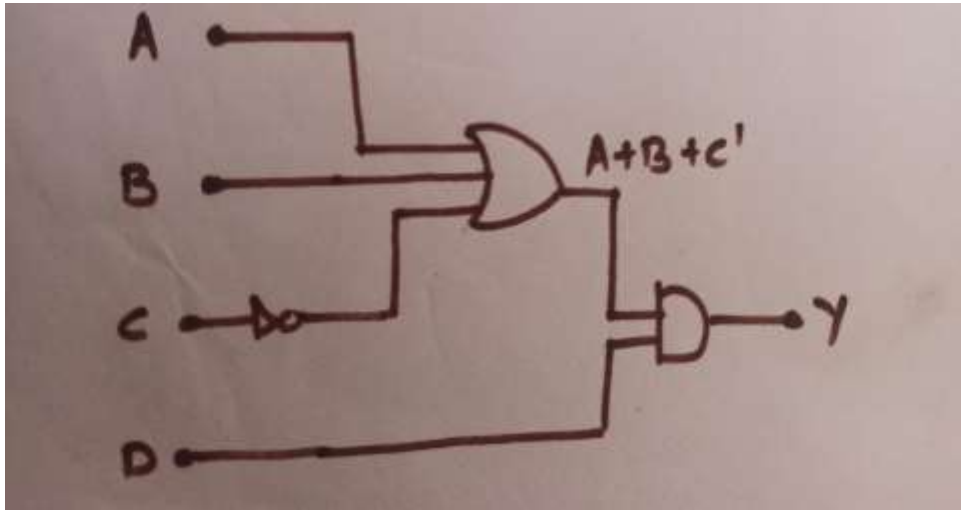
Example 2: Minimize the following expression using K-map method and draw the diagram.
Y = ∑ m (1,5,7,9,11,13,15)

Ans: **Y = ∑ m (1,5,7,9,11,13,15)**
Here we need a 4 variable K-map or 16 cells K-map.

C'D **BD** **AD**

AB \ CD	00	01	11	10
00	0 (0)	1 (1)	0 (3)	0 (2)
01	0 (4)	1 (5)	1 (7)	0 (6)
11	0 (12)	1 (13)	1 (15)	0 (14)
10	0 (8)	1 (9)	1 (11)	0 (10)

Y = C'D + BD + AD
= D (C' + B + A)



Minimization of SOP Expression

Example 1: Minimize the following expression using K-map method and draw the diagram.
$Y = \sum m\ (3,4,6,7)$

Example 2: Minimize the following expression using K-map method and draw the diagram.
$Y = \sum m\ (0,1,3,4,5,6,7,8,11,15)$

Don’t Care Condition

- **Input Combinations for which the values of the Boolean expression are not specified are called as Don’t care condition.**
- **It is indicated by ‘X’ or ‘d’ in the Truth Table.**
- **E.g. BCD to EX-3 Code Converter**

B3	B2	B1	B0	E3	E2	E1	E0
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

K-map for E3:

B3B2 \ B1B0	00	01	11	10
00	0 (0)	0 (1)	0 (3)	0 (2)
01	0 (4)	1 (5)	1 (7)	1 (6)
11	X (12)	X (13)	X (15)	X (14)
10	1 (8)	1 (9)	X (11)	X (10)

B2B1

B2B0

B3

E3 = B3 + B2B1 + B2B0

= B3 + B2 (B1 + B0)

Don’t Care Condition

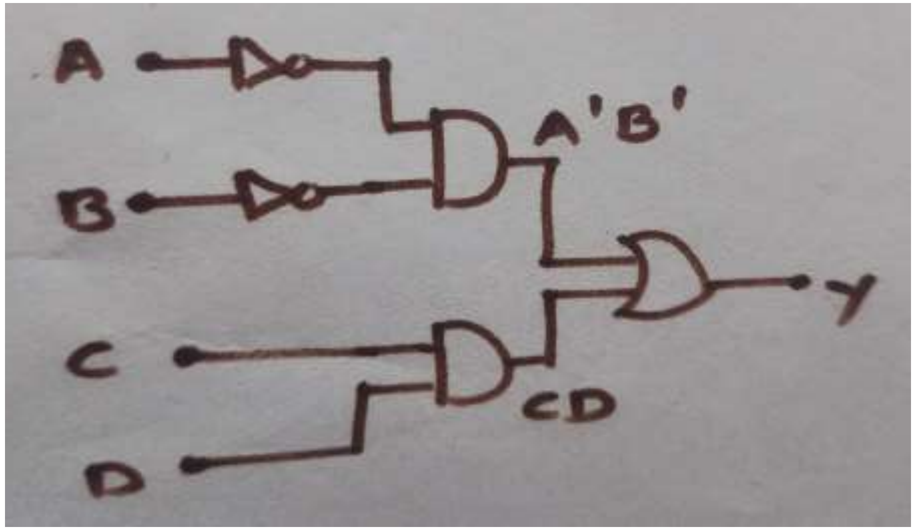
Example 1: Minimize the following logic function using K-map and realize using logic gates :

Y = ∑ m (1,3,7,11,15) + d(0,2,5) **[May 2018 - 4 Marks]**

A’B’ CD

AB \ CD	00	01	11	10
00	X 0	1 1	1 3	X 2
01	0 4	X 5	1 7	0 6
11	0 12	0 13	1 15	0 14
10	0 8	0 9	1 11	0 10

Y = A’B’ +CD



Exercise

Example 1: Minimize the following expression using K-map method and draw the diagram.
$Y = \sum m\ (0,2,5,8,11,15) + d(1,7,14)$ **[May 2016 4M]**

Example 2: Minimize the following expression using K-map method and draw the diagram.
$Y = \sum m\ (1,3,7,11,15) + d(0,2,5)$

Example 3: Minimize the following logic function using K-map and realize using logic gates :
$Y = \sum m\ (1,5,7,13,15) + d\ (0,6,12,14)$ **[May 2018 - 4 Marks]**

Minimization of POS Expression using K-map

Plot the K-map and place 0's corresponding to the max terms of the POS

Each cell in k-map represent sum term

Do the grouping of all 0's (Octet, Quad, Pair)

Check 0's that are not adjacent to any other 0; consider that single sum term

Compute the minimal expression by multiplying the sum term of all the group

Example 1: Minimize the expression using K-map,

$$Y = (A+B+C)\,(A+B'+C)\,(A+B'+C')\,(A'+B+C')$$

Ans:

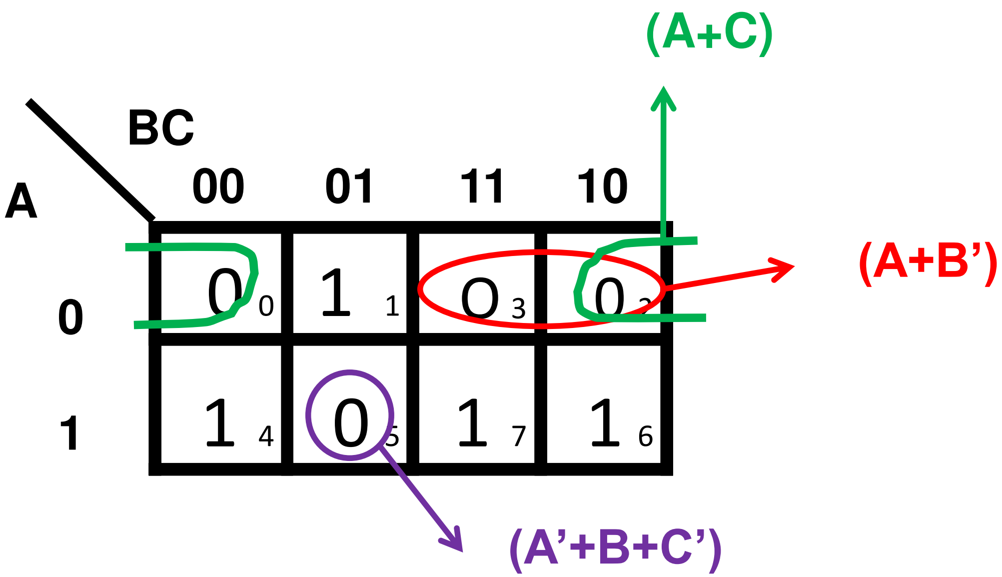
$(A+B+C) = M_0$

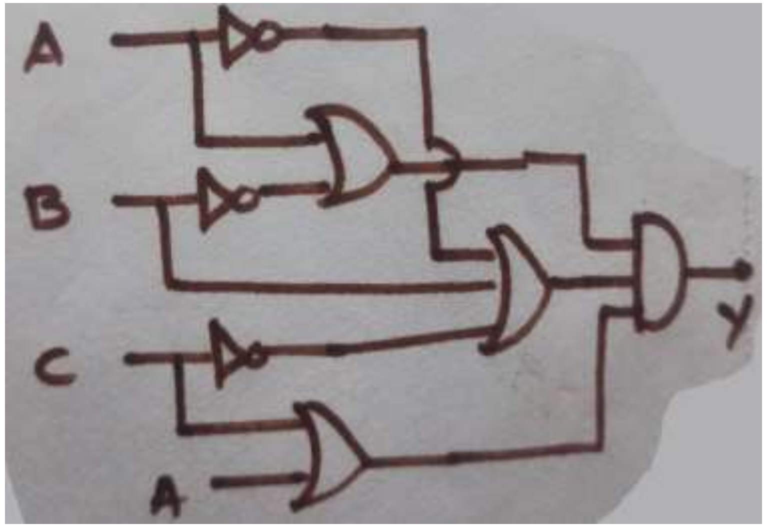
$(A+B'+C) = M_2$

$(A+B'+C') = M_3$

$(A'+B+C') = M_5$

$$Y = (A+C)\,(A+B')\,(A'+B+C')$$





Minimization of POS Expression using K-map

Example 2: Minimize the expression using K-map,

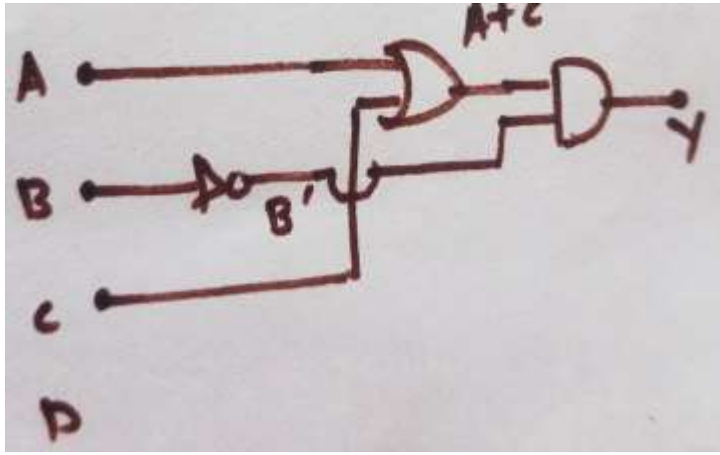
$$Y = \prod M(1,5,7,13,15) + d(0,4,6,12,14)$$

A+C

B'

AB \ CD	00	01	11	10
00	X 0	0 1	1 3	1 2
01	X 4	0 5	0 7	X 6
11	X 12	0 13	0 15	X 14
10	1 8	1 9	1 11	1 10

Y = (B') (A+C)



Exercise

Example 1: Minimize the following expression using K-map method and draw the diagram.
$Y = \prod M\ (1,2,3,7,10,11) + d(0,15)$

Example 2: Minimize the following expression using K-map method and draw the diagram.
$Y = \prod M\ (1,3,5,6,7,10,11) + d(2,4)$

Agenda

01 **Logic Design Minimization Technique-** Minimization of Boolean function using K-map(up to 4 variables) and Quine Mc-Clusky Method

02 **Representation of signed number-** sign magnitude representation ,1's complement and 2's complement form

03 **Sum of product and product of sum**

04 **Minimization of SOP and POS using k-map**

Quine Mc-Cluskey Method

- This method consists of two parts,
  - To find all the prime implicants
  - To identify the essential prime impliants from the obtained prime implicants to form a Minimized expression.

- **Prime Implicants (PI)** – A product term is called as a prime implicant if it cannot be combined with any other product term or group.
- **Essential Prime Implicants (EPI)** – It is prime implicant in which one or more min term are unique i.e. it contains at least one min term which is not contained in any other prime implicant.

Quine Mc-Cluskey Method

Example 1: Simplify the following function using Quine Mc-Cluskey minimization technique

Y= ∑ m (0,1,3,7,8,9,11,15) + d(2,4)

B’C’

CD

AB \ CD	00	01	11	10
00	1 $_{0}$	1 $_{1}$	1 $_{3}$	X $_{2}$
01	X $_{4}$	0 $_{5}$	1 $_{7}$	0 $_{6}$
11	0 $_{12}$	0 $_{13}$	1 $_{15}$	0 $_{14}$
10	1 $_{8}$	1 $_{9}$	1 $_{11}$	0 $_{10}$

Y = CD + B’C’

Quine Mc-Cluskey Method

Example 1: Simplify the following function using Quine Mc-Cluskey minimization technique

Y= $\sum$ m (0,1,3,7,8,9,11,15) +d(2,4)

Ans:

Step 1: List all the min term in binary form

Given Min term	A	B	C	D
m0	0	0	0	0
m1	0	0	0	1
m3	0	0	1	1
m7	0	1	1	1
m8	1	0	0	0
m9	1	0	0	1
m11	1	0	1	1
m15	1	1	1	1

Step 2: Arrange the min terms in group of 1's in their binary representation.

Group	Given Min term	A	B	C	D
1	m0	0	0	0	0
2	m1	0	0	0	1
	m8	1	0	0	0
3	m3	0	0	1	1
	m9	1	0	0	1
4	m7	0	1	1	1
	m11	1	0	1	1
5	m15	1	1	1	1

Quine Mc-Cluskey Method

Step 3: Compare every term of group n with each min term in the group n+1. If the binary representation of the terms differs by 1 bit in the same position, the two terms are combined. Put (✓) in front of min terms combined and '-' in the different position.

Group	Given Min term	A	B	C	D	
1	m0	**0**	**0**	**0**	**0**	✓
2	m1	**0**	**0**	**0**	**1**	✓
	m8	1	0	0	0	✓
3	m3	0	0	1	1	✓
	m9	1	0	0	1	✓
4	m7	0	1	1	1	✓
	m11	1	0	1	1	✓
5	m15	1	1	1	1	✓

Group	Given min term	A	B	C	D
1	m0-m1	0	0	0	-
	M0-m8	-	0	0	0
2	M1-m3	0	0	-	1
	m1-m9	-	0	0	1
	m8-m9	1	0	0	-
3	m3-m7	0	-	1	1
	M3-m11	-	0	1	1
	m9-m11	1	0	-	1
4	m7-m15	-	1	1	1
	m11-m15	1	-	1	1

Quine Mc-Cluskey Method

Step 4: Compare every term of group n with each min term in the group n+1.

Group	Given min term	A	B	C	D	
1	m0 – m1	0	0	0	-	✓
	m0 – m8	-	0	0	0	✓
2	m1 – m3	0	0	-	1	✓
	m1 – m9	-	0	0	1	✓
	m8 – m9	1	0	0	-	✓
3	m3- m7	0	-	1	1	✓
	m3 – m11	-	0	1	1	✓
	m9 – m11	1	0	-	1	✓
4	m7 – m15	-	1	1	1	✓
	m11 – m15	1	-	1	1	✓

Group	Given min term	A	B	C	D
1	M0-m1-m8-m9	-	0	0	-
	M0-m8-m1-m9	-	0	0	-
2	M1-m3-m9-m11	-	0	-	1
	M1-m9-m3-m11	-	0	-	1
3	M3-m7-m11-m15	-	-	1	1
	M3-m11-m7-m15	-	-	1	1

Quine Mc-Cluskey Method

Step 5: List the all prime implicants (all unchecked terms from step 2 to step 4)

Group	min term	A	B	C	D	
1	m0-m1-m8-m9	-	0	0	-	**B'C'**
	m0-m8-m1-m9	-	0	0	-	
2	m1-m3-m9-m11	-	0	-	1	**B'D**
	m1-m9-m3-m11	-	0	-	1	
3	m3-m7-m11-m15	-	-	1	1	**CD**
	m3-m11-m7-m15	-	-	1	1	

Quine Mc-Cluskey Method

Y= ∑ m (0,1,3,7,8,9,11,15)
Step 6: Select EPI from all PI

PI	Decimal no. for PI	Given Minterm							
		0	1	3	7	8	9	11	15
B'C'	0,1,8,9	X	X			X	X		
B'D	1,3,9,11,		X	X			X	X	
CD	3,7,11,15			X	X			X	X

1. Put X in each row under the min terms contained in a particular PI.
2. Encircled min term 0 and 8 are contained in only one PI i.e. B'C' and the min term 7 and 15 are contained by only PI CD
3. Hence B'C' and CD are EPI.

Y (A,B,C,D) = B'C' + CD

Exercise

Example 1: Simplify the following function using Quine Mc-cluskey method and draw the diagram.
$Y = \sum m\ (0,1,2,3,5,7,8,9,11,14)$ **[Dec 2018 6M]**

Example 2: Simplify the following function using Quine Mc-cluskey method and draw the diagram.
$Y = \sum m\ (0,2,3,6,7,8,10,12,13)$

Thank You…!!!