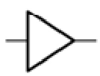
Digital Logic & Circuit Design

Boolean Algebra and Logic Gates

Prepared by:
Nilesh Patidar and Shiraz Husain

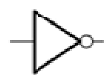
Logic Gates

YES



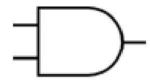
INPUT	OUTPUT
A	
0	0
1	1

NOT



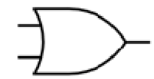
INPUT	OUTPUT
A	
0	1
1	0

AND



INPUT		OUTPUT
A	B	
0	0	0
1	0	0
0	1	0
1	1	1

OR

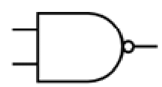


INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	1

XOR

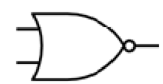
INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	0

NAND



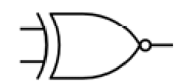
INPUT		OUTPUT
A	B	
0	0	1
1	0	1
0	1	1
1	1	0

NOR



INPUT		OUTPUT
A	B	
0	0	1
1	0	0
0	1	0
1	1	0

XNOR



INPUT		OUTPUT
A	B	
0	0	1
1	0	0
0	1	0
1	1	1

Boolean Postulates and Laws

Identity:	$X + 0 = X$	Dual: $X \bullet 1 = X$
Null:	$X + 1 = 1$	Dual: $X \bullet 0 = 0$
Idempotent:	$X + X = X$	Dual: $X \bullet X = X$
Involution:	$(X')' = X$	
Complementarity:	$X + X' = 1$	Dual: $X \bullet X' = 0$
Commutative:	$X + Y = Y + X$	Dual: $X \bullet Y = Y \bullet X$
Associative:	$(X+Y)+Z=X+(Y+Z)$	Dual: $(X \bullet Y) \bullet Z=X \bullet (Y \bullet Z)$
Distributive:	$X \bullet (Y+Z)=(X \bullet Y)+(X \bullet Z)$	Dual: $X+(Y \bullet Z)=(X+Y) \bullet (X+Z)$
Uniting:	$X \bullet Y+X \bullet Y'=X$	Dual: $(X+Y) \bullet (X+Y')=X$

Boolean Postulates and Laws(Cont..)

Absorption:	$X+X\bullet Y=X$	Dual: $X\bullet(X+Y)=X$
Absorption(#2):	$(X+Y')\bullet Y=X\bullet Y$	Dual: $(X\bullet Y')+Y=X+Y$
De-Morgan's:	$(X+Y+\ldots)'=X'\bullet Y'\bullet\ldots$	Dual: $(X\bullet Y\bullet\ldots)'=X'+Y'+\ldots$
Duality:	$(X+Y+\ldots)^D=X\bullet Y\bullet\ldots$	Dual: $(X\bullet Y\bullet\ldots)^D=X+Y+\ldots$

Multiplying and factoring :$(X+Y)\bullet(X'+Z)=X\bullet Z+X'\bullet Y$

Dual: $X\bullet Y+X'\bullet Z=(X+Z)\bullet(X'+Y)$

Consensus: $(X\bullet Y)+(Y\bullet Z)+(X'\bullet Z)= X\bullet Y+X'\bullet Z$

Dual: $(X+Y)\bullet(Y+Z)\bullet(X'+Z)=(X+Y)\bullet(X'+Z)$

Summary

	Property	Dual Property
Identity	$x + 0 = x$	$x \cdot 1 = x$
Complement	$x + x' = 1$	$x \cdot x' = 0$
Null	$x + 1 = 1$	$x \cdot 0 = 0$
Idempotence	$x + x = x$	$x \cdot x = x$
Involution	$(x')' = x$	
Commutative	$x + y = y + x$	$x\, y = y\, x$
Associative	$(x + y) + z = x + (y + z)$	$(x\, y)\, z = x\, (y\, z)$
Distributive	$x\,(y + z) = xy + xz$	$x + yz = (x + y)(x + z)$
Absorption	$x + xy = x$	$x(x + y) = x$
Simplification	$x + x'y = x + y$	$x(x' + y) = xy$
De Morgan	$(x + y)' = x'\, y'$	$(x\, y)' = x' + y'$

Proving theorems

- Example 1: Prove the uniting theorem-- $X \bullet Y + X \bullet Y' = X$

Distributive	$= X \bullet Y + X \bullet Y' = X \bullet (Y + Y')$
Complementary	$= X \bullet (1)$
Identity	$= X$

- Example 2: Prove the absorption theorem-- $X + X \bullet Y = X$

Identity	$X + X \bullet Y = (X \bullet 1) + (X \bullet Y)$
Distributive	$= X \bullet (1 + Y)$
Null	$= X \bullet (1)$
Identity	$= X$

Proving theorems

- Example 3: Prove the consensus theorem
  (XY)+(YZ)+(X'Z)= XY+X’Z

Solution:

- Complementarity XY+YZ+X'Z = XY+(X+X')YZ + X’Z
- Distributive = XY+XYZ+X'YZ+X’Z
- ***Use absorption {AB+A=A}***

  = XY+X'YZ+X'Z
- Rearrange terms = XY+X'ZY+X’Z
- ***Use absorption {AB+A=A}***

  = XY+X'Z

Example: Simplification

$Z = A'BC + AB'C' + AB'C + ABC' + ABC$	
$= A'BC + AB'(C' + C) + AB(C' + C)$	distributive
$= A'BC + AB' + AB$	complementary
$= A'BC + A(B' + B)$	distributive
$= A'BC + A$	complementary
$= BC + A$	absorption

Simplification of Boolean Equation

- Reduce the Expression- $A + B\,(AC + (B + \bar{C})D)$

1. Identity Element	x + 0 = x
2. Complementation	x + x'=1
3. Idem potency	x + x = x
4. Null Law	x + 1 = 1
5. Involution	(x')' = x
6. Commutative	x + y = y + x
7. Associative	x + (y + z) = (x + y) + z
8. Distributive	x + yz = (x + y)(x + z)
9. De Morgan	(x + y)' = x' y'
10. Absorption	x + xy = x
11. Simplification	x + x'y = x + y
12. Consensus	xy + x'z +yz = xy + x'z

Simplification of Boolean Equation

- Simplify the Expression- $(B + BC)(B + \bar{B}C)(B + D)$

1. Identity Element	x + 0 =x
2. Complementation	x + x'=1
3. Idem potency	x + x = x
4. Null Law	x + 1 = 1
5. Involution	(x')' = x
6. Commutative	x + y = y + x
7. Associative	x + (y + z) = (x + y) + z
8. Distributive	x + yz = (x + y)(x + z)
9. De Morgan	(x + y)' = x' y'
10. Absorption	x + xy = x
11. Simplification	x + x'y = x + y
12. Consensus	xy + x'z +yz = xy + x'z

Simplification of Boolean Equation

- Simplify the Expression- $(B + BC)(B + \bar{B}C)(B + D)$

1. Identity Element	x + 0 = x
2. Complementation	x + x' = 1
3. Idem potency	x + x = x
4. Null Law	x + 1 = 1
5. Involution	(x')' = x
6. Commutative	x + y = y + x
7. Associative	x + (y + z) = (x + y) + z
8. Distributive	x + yz = (x + y)(x + z)
9. De Morgan	(x + y)' = x' y'
10. Absorption	x + xy = x
11. Simplification	x + x'y = x + y
12. Consensus	xy + x'z + yz = xy + x'z

Simplification of Boolean Equation

- Simplify the Expression- F= A'B(C'D' + C'D) + AB (C'D' +C'D) + AB'C'D

1. Identity Element	x + 0 = x
2. Complementation	x + x' = 1
3. Idem potency	x + x = x
4. Null Law	x + 1 = 1
5. Involution	(x')' = x
6. Commutative	x + y = y + x
7. Associative	x + (y + z) = (x + y) + z
8. Distributive	x + yz = (x + y)(x + z)
9. De Morgan	(x + y)' = x' y'
10. Absorption	x + xy = x
11. Simplification	x + x'y = x + y
12. Consensus	xy + x'z + yz = xy + x'z

Logic Diagram

- Implement the following function using logic gates-
  $F = \overline{AB} + AC + \bar{B}$

Logic Diagram

- Implement the following function using logic gates-
- $F = (P'Q' + PQ)'R + (P'Q+PQ')R'$

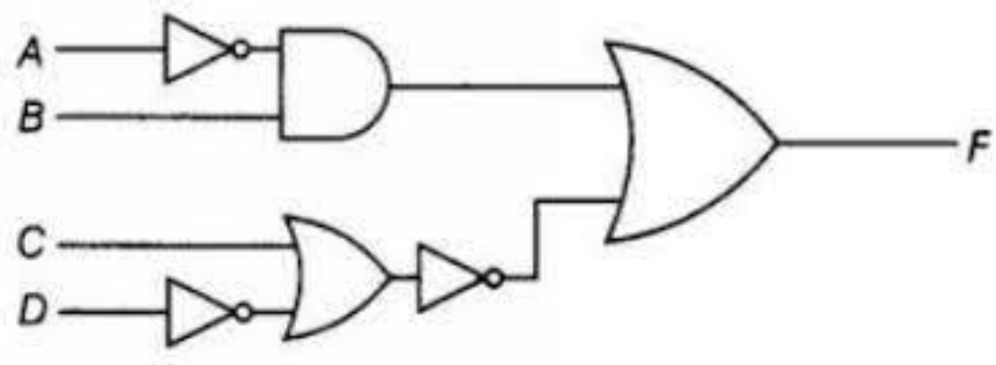
Logic Diagram

- Minimize the following Boolean Function & also draw the simplified logic diagram

  $Y = (AB+C)(AB+D)$

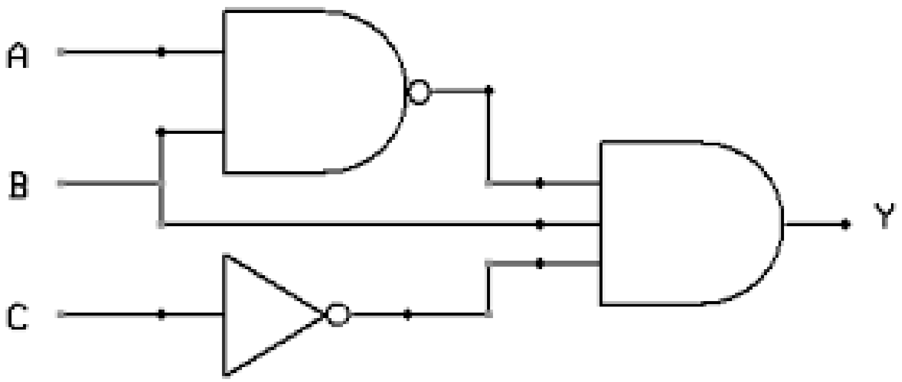
Logic Diagram

- Minimize the following Boolean Function & also draw the simplified logic diagram



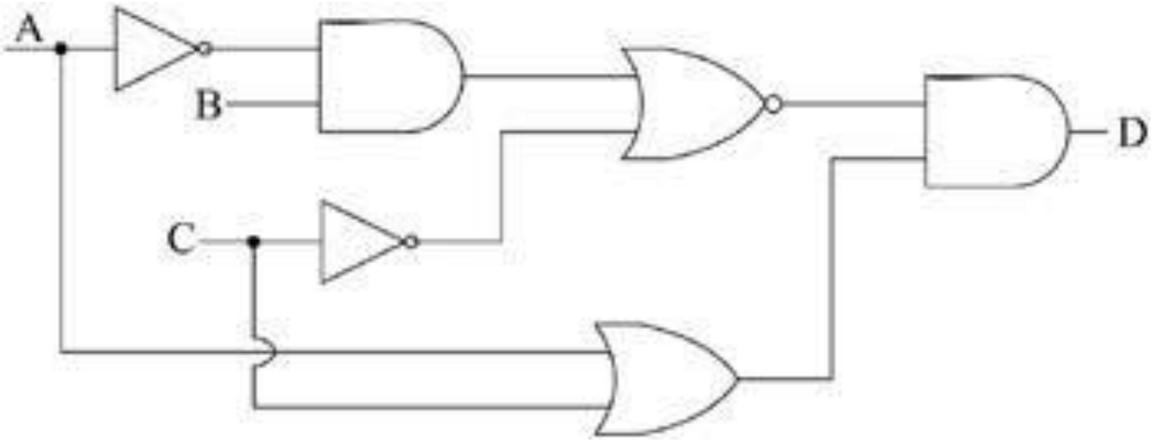
Logic Diagram

Obtain the Boolean expression



Logic Diagram

- Simplify the given logic diagram.

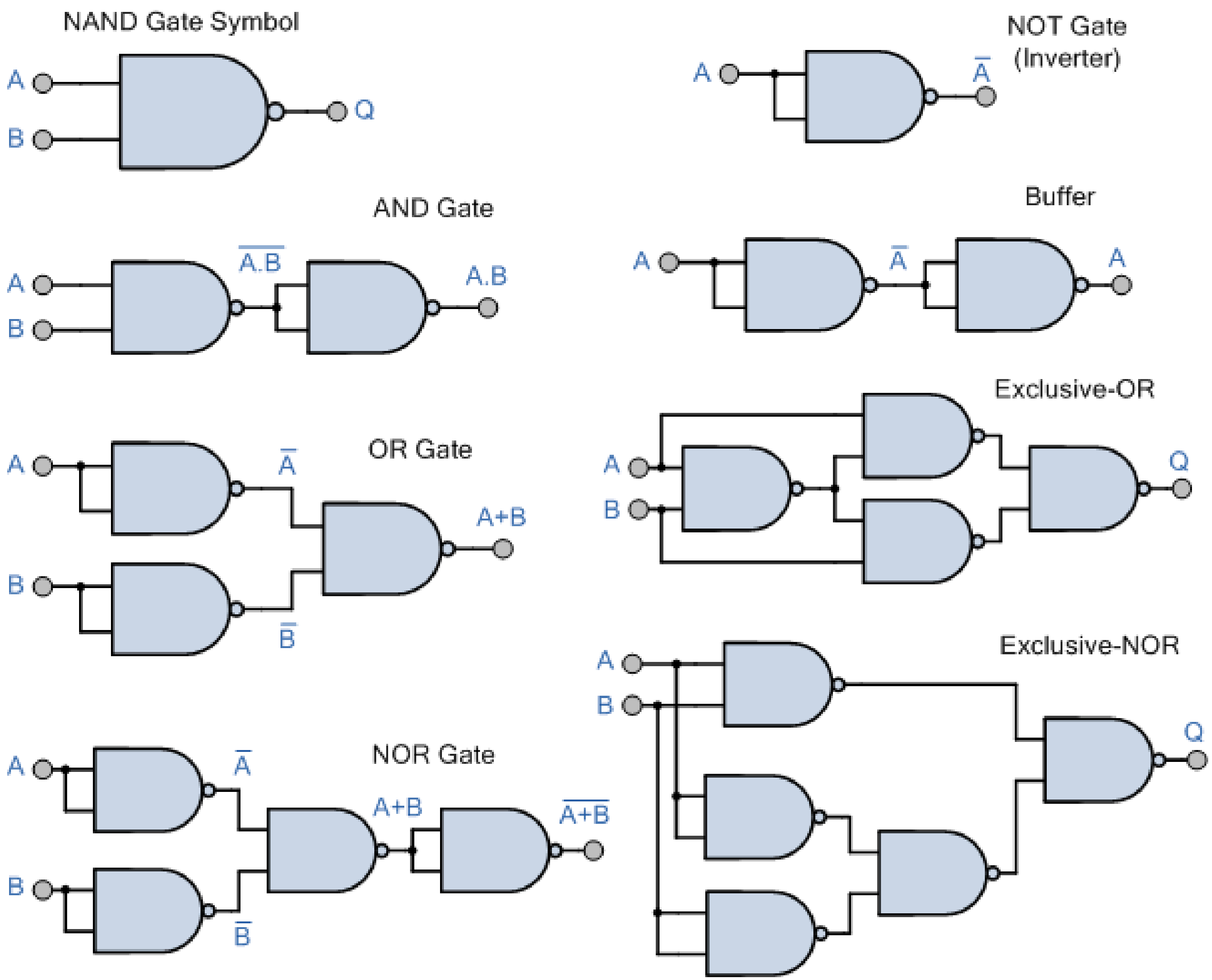


Duality

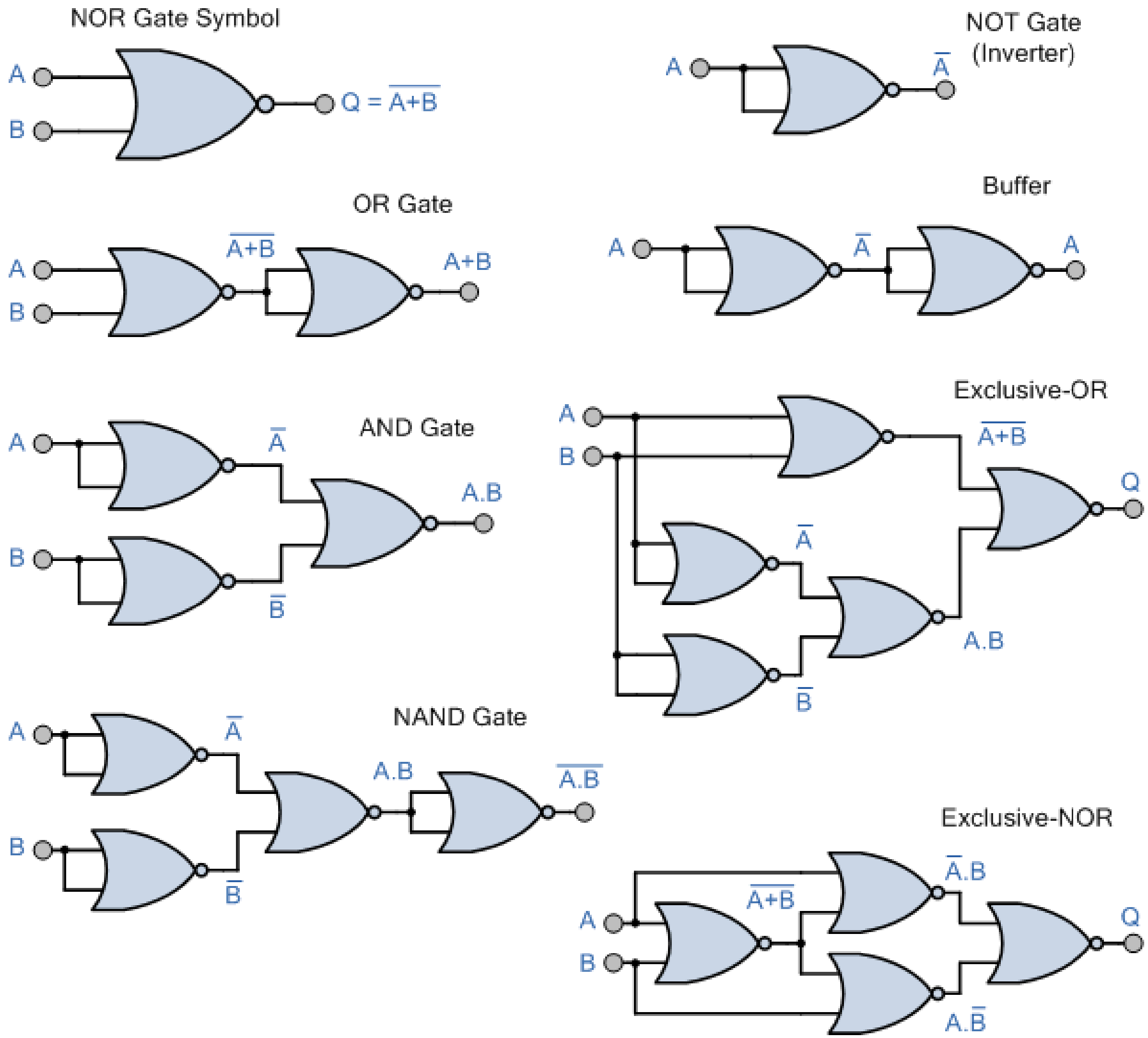
- It states that in a two valued Boolean algebra, the dual of an algebraic expression can be obtained simply by interchanging OR and AND operators and by replacing 1's by 0's and 0's by 1's.

- Find the dual of the following Boolean expressions:
  i. $A+AB = A$
  ii. $A + A'B = A + B$
  iii. $A + A' = 1$
  iv. $(A+B)(A+C) = A+BC$

Universal Gate: NAND Gate



Universal Gate: NOR gate



Bubble Pushing Method- NOR Gate

- Step 1: Implement the given Boolean expression in AOI form.
- Step 2: Add a bubble at input of each AND gate and output of each OR gate.
- Step 3: Connect a NOT gate on each wire where a bubble is added in Step 2.
- Step 4: Cancel out 2 NOT gates that are connected in series.
- Step 5: Replace NOT gate with NOR gate equivalent.
- Step 6: Replace bubbled AND with NOR gate.

Implementation using NOR Gate

- Implement the following function using NOR gates only
- $A(B'+C) + (BC)$

Implementation using NOR Gate

- Implement the following function using NOR gates only
- $F = (A'+B+C)(A+B)D$

Implementation using NOR Gate

- Implement the following function using NOR gates only
- *F= (C+BD’).(B’CA+CD)*

Bubble Pushing Method- NAND Gate

- Step 1: Implement the given Boolean expression in AOI form.
- Step 2: Add a bubble at output of each AND gate and input of each OR gate.
- Step 3: Connect a NOT gate on each wire where a bubble is added in Step 2.
- Step 4: Cancel out 2 NOT gates that are connected in series.
- Step 5: Replace NOT gate with NAND gate equivalent.
- Step 6: Replace bubbled OR with NAND gate.

Implementation using NAND Gate

- Implement the following function using NAND gates only
- $Y = AB' + BC$

Implementation using NAND Gate

- Implement the following function using NAND gates only
- $F = AB\ (C+D) + E$

Standard Form (SOP & POS)

- Standard SOP form
  - $Y = ABC + AB'C' + A'BC$
  - Each product term consists of all the literals in the complemented or uncomplemented form

- Standard POS form
  - $Y = (A+B+C')(A+B'+C)(A'+B'+C')$
  - Each sum term consists of all the literals in the complemented or uncomplemented form

Canonical Form (SOP & POS)

				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		$m_7 = X\cdot Y\cdot Z = \min(X,Y,Z)$		$M_7 = \overline{X} + \overline{Y} + \overline{Z} = \max(\overline{X},\overline{Y},\overline{Z})$

- SOP Form $Y = \sum m(0,3,4,5)$
- POS Form $Y = \prod M(1,2,5,6)$

Canonical Form (SOP & POS)

- Convert the following expressions into their standard SOP form.
  - $Y = AB + AC + BC$

Examples

- Convert the following expressions into their standard SOP form.
  - $Y = AB + AC + BC$

Examples

- Convert the following expressions into their standard POS form.
  - $Y = (A+B)(B'+C)$

Examples

- Convert the following expressions into their standard SOP form.
  - $Y = A + BC + ABC$

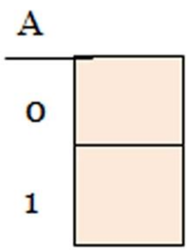
Karnaugh Maps (K-Maps)

- Simplification of logic expression using Boolean algebra is awkward because:
  - it lacks specific rules to predict the most suitable next step in the simplification process
  - it is difficult to determine whether the simplest form has been achieved.
- K-MAPS
  - A visual way to simplify logic expressions
  - It gives the most simplified form of the expression

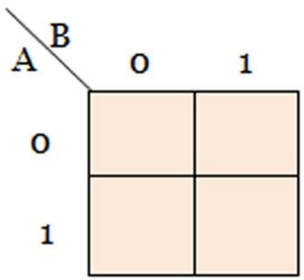
Rules to obtain the most simplified expression

- A Karnaugh map is a graphical method used to obtained the most simplified form of an expression in a standard form (Sum-of-Products or Product-of-Sums).
- The simplest form of an expression is the one that has the minimum number of terms with the least number of literals (variables) in each term.
- By simplifying an expression to the one that uses the minimum number of terms, we ensure that the function will be implemented with the minimum number of gates.
- By simplifying an expression to the one that uses the least number of literals for each terms, we ensure that the function will be implemented with gates that have the minimum number of inputs.

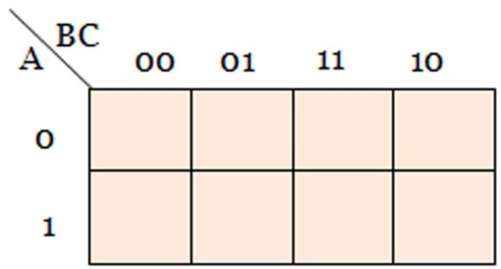
K-maps



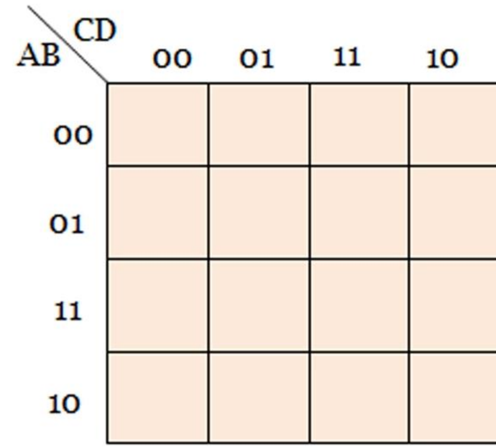
1-variable K-map
$2^1 = 2$ cells



2-variable K-map
$2^2 = 4$ cells



3-variable K-map
$2^3 = 8$ cells



4-variable K-map
$2^4 = 16$ cells

2 Variable K-map

		B'	B
	A \ B	0	1
A'	0	$A'B'$	$A'B$
A	1	AB'	AB

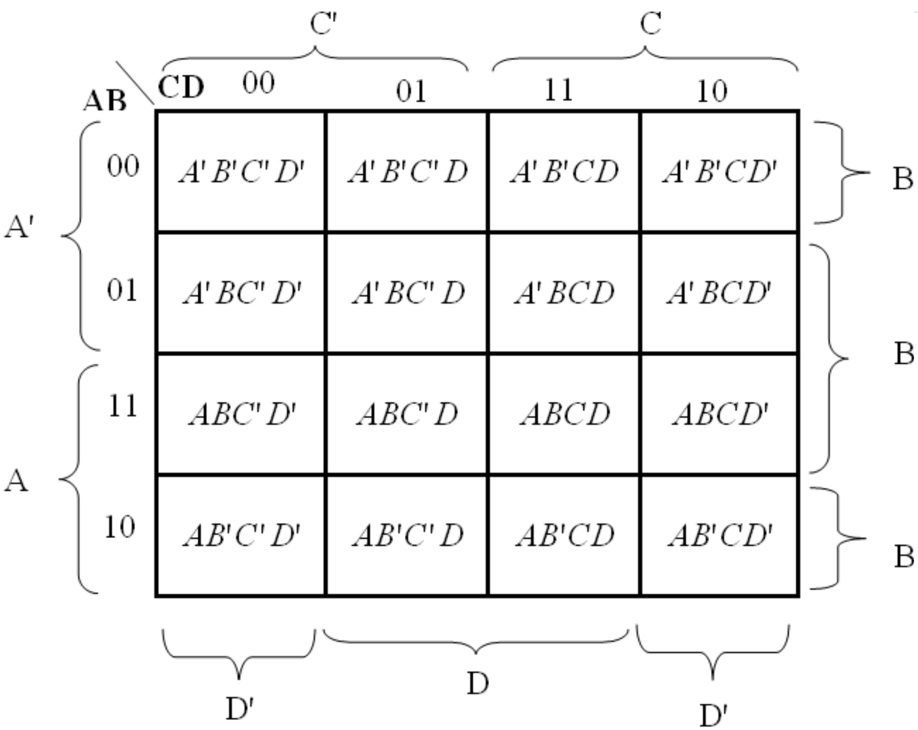
		B'	B
	A \ B	0	1
A'	0	m_0	m_1
A	1	m_2	m_3

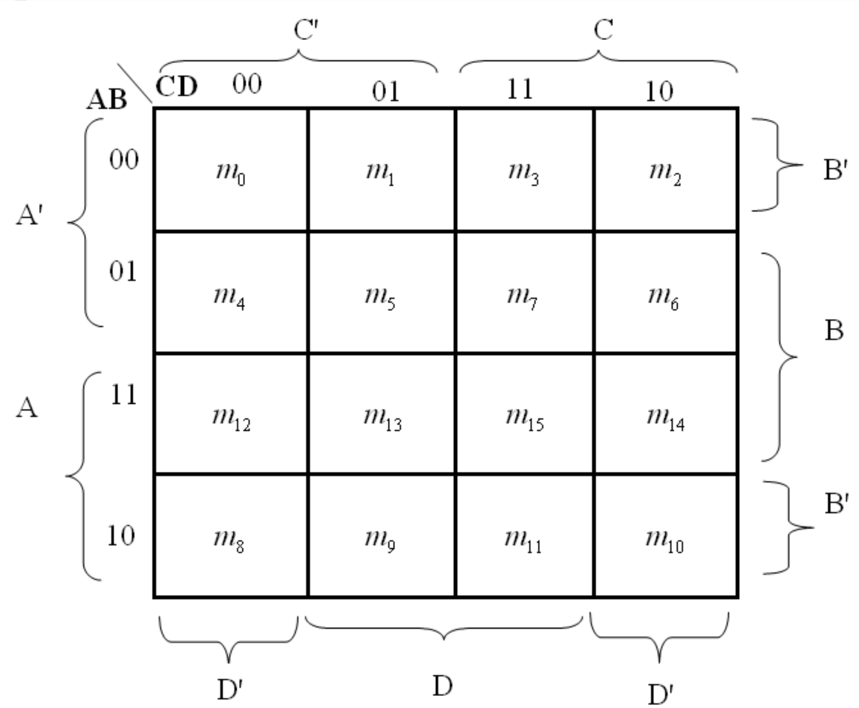
3 Variable K-map

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

		B'		B	
	A \ BC	00	01	11	10
A'	0	m_0	m_1	m_3	m_2
A	1	m_4	m_5	m_7	m_6
		C'	C	C	C'

4 Variable K-map





Three-Variable K-Maps Examples

$f = \sum(0,4) = \overline{B}\,\overline{C}$

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

$f = \sum(4,5) = A\,\overline{B}$

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

$f = \sum(0,1,4,5) = \overline{B}$

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

$f = \sum(0,1,2,3) = \overline{A}$

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

$f = \sum(0,4) = \overline{A}\,C$

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

$f = \sum(4,6) = A\,\overline{C}$

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

$f = \sum(0,2) = \overline{A}\,\overline{C}$

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

$f = \sum(0,2,4,6) = \overline{C}$

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

Three-Variable K-Map Examples

- Minimize the expression
  - Y = AB’C + A’B’C + A’BC + AB’C’ + A’B’C

Four-Variable K-Maps

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

$$f = \sum(0,8) = \overline{B} \bullet \overline{C} \bullet \overline{D}$$

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

$$f = \sum(5,13) = B \bullet \overline{C} \bullet D$$

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

$$f = \sum(13,15) = A \bullet B \bullet D$$

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

$$f = \sum(4,6) = \overline{A} \bullet B \bullet \overline{D}$$

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

$$f = \sum(2,3,6,7) = \overline{A} \bullet C$$

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

$$f = \sum(4,6,12,14) = B \bullet \overline{D}$$

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

$$f = \sum(2,3,10,11) = \overline{B} \bullet C$$

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

$$f = \sum(0,2,8,10) = \overline{B} \bullet \overline{D}$$

Four-Variable K-Maps

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

$$f = \sum(4,5,6,7) = \overline{A} \bullet B$$

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

$$f = \sum(3,7,11,15) = C \bullet D$$

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

$$f = \sum(0,3,5,6,9,10,12,15)$$

$$f = A \otimes B \otimes C \otimes D$$

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

$$f = \sum(1,2,4,7,8,11,13,14)$$

$$f = A \oplus B \oplus C \oplus D$$

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

$$f = \sum(1,3,5,7,9,11,13,15)$$

$$f = D$$

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

$$f = \sum(0,2,4,6,8,10,12,14)$$

$$f = \overline{D}$$

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

$$f = \sum(4,5,6,7,12,13,14,15)$$

$$f = B$$

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

$$f = \sum(0,1,2,3,8,9,10,11)$$

$$f = \overline{B}$$

Special Case

- Find the Boolean expression.

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

Four-Variable K-Maps Examples

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

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

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

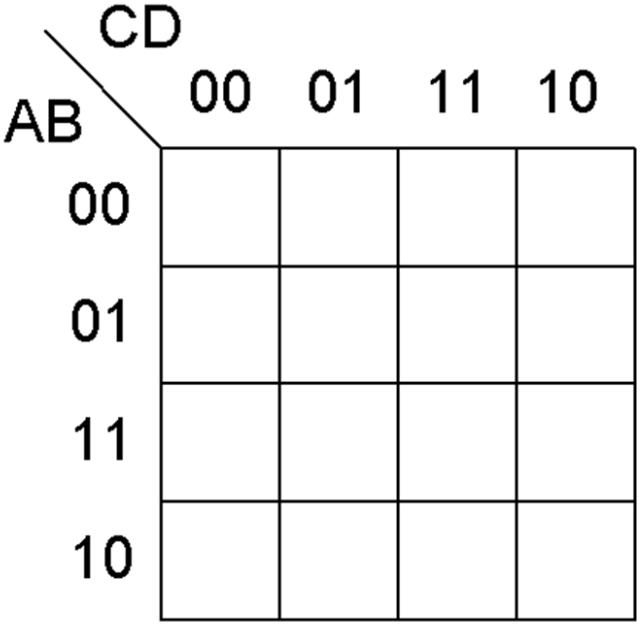
AB \ CD	00	01	11	10
00		1	1	
01	1	1	1	1
11	1		1	1
10			1	

AB \ CD	00	01	11	10
00				
01				
11				
10				

AB \ CD	00	01	11	10
00				
01				
11				
10				

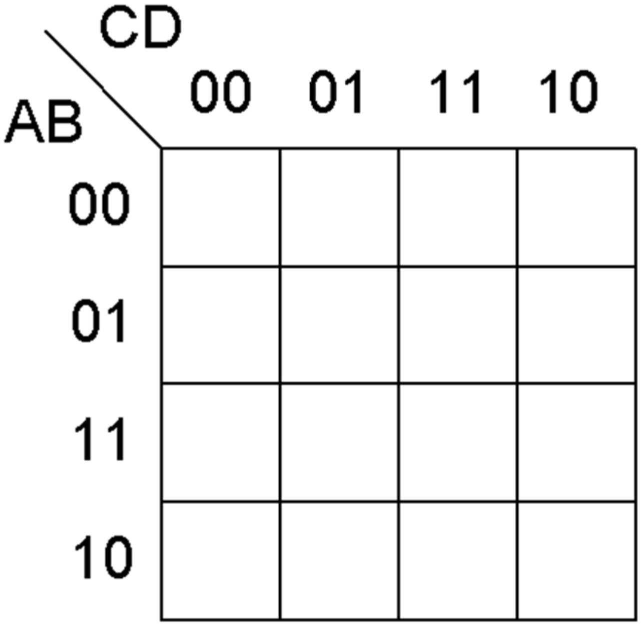
Four-Variable K-Maps don’t care

- $Y = \sum m\,(1,4,8,12,13,15) + d(3,14)$



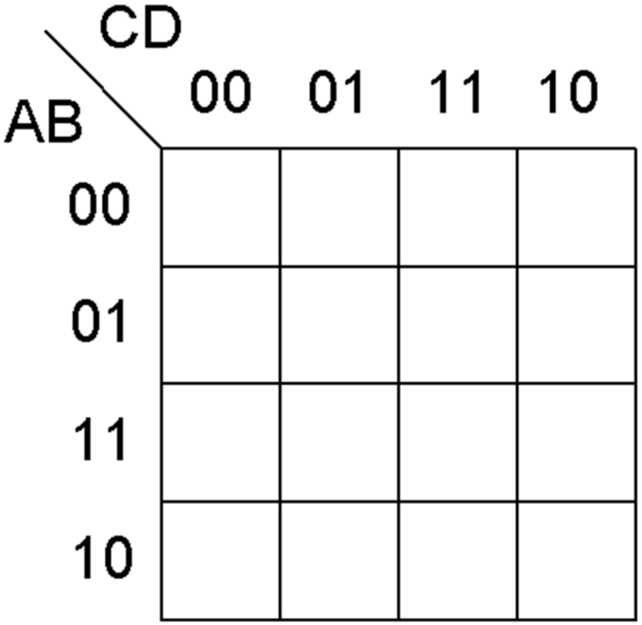
Four-Variable K-Maps don’t care

- $Y = \sum m\,(1,3,7,11,15) + d(0,2,5)$



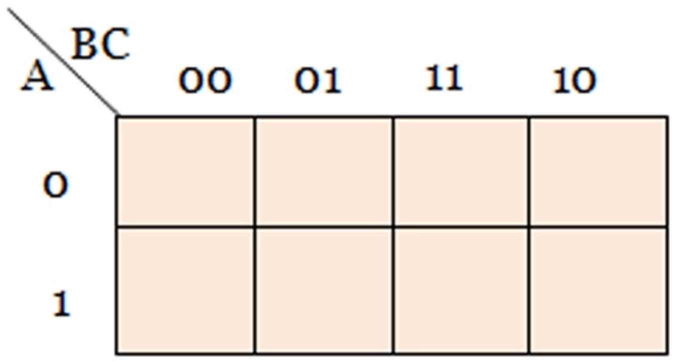
Four-Variable K-Maps don’t care

- $Y = \sum m\,(1,3,7,10,11,15) + d(0,2,5,8,14)$



K-maps

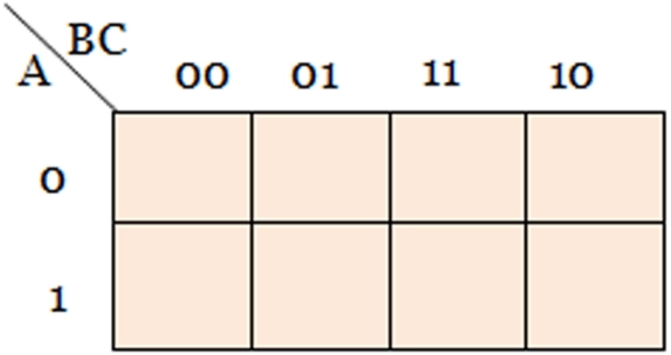
- Obtain the expression in POS form
- F = $\prod M(2,3,7)$



3-variable K-map
$2^3 = 8$ cells

K-maps

- Obtain the expression in POS form
- $F = \prod M(0,2,3,5,7)$



3-variable K-map
$2^3 = 8$ cells

Five Variable K-Map

V=0

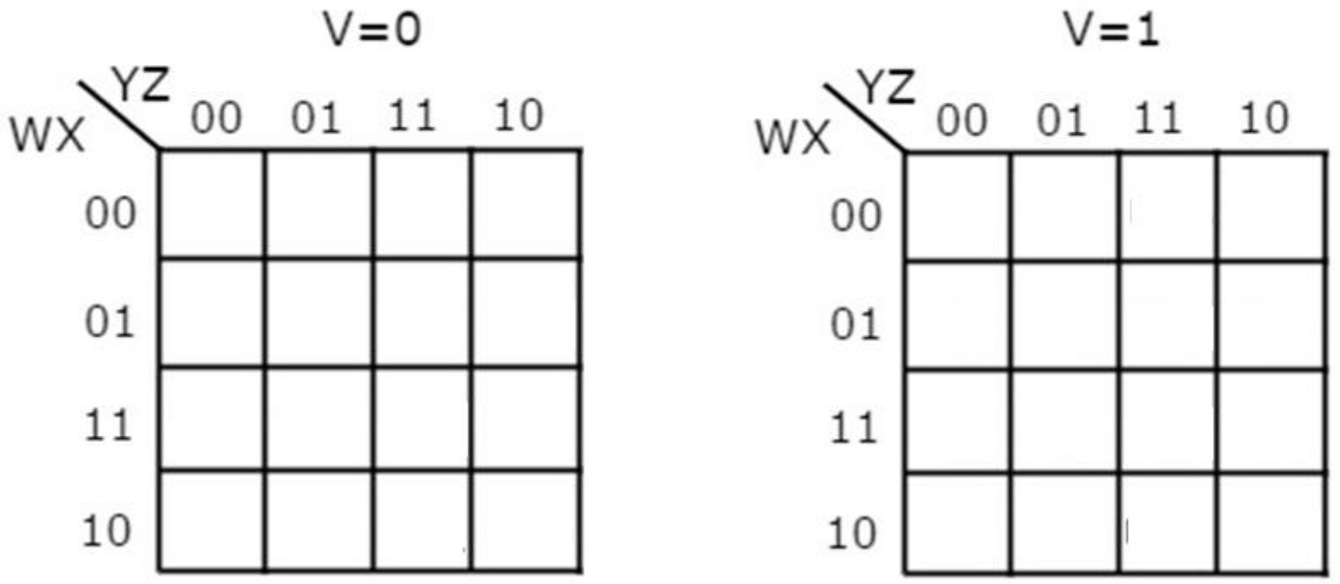
WX \ YZ	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

V=1

WX \ YZ	00	01	11	10
00	m_{16}	m_{17}	m_{19}	m_{18}
01	m_{20}	m_{21}	m_{23}	m_{22}
11	m_{28}	m_{29}	m_{31}	m_{30}
10	m_{24}	m_{25}	m_{27}	m_{26}

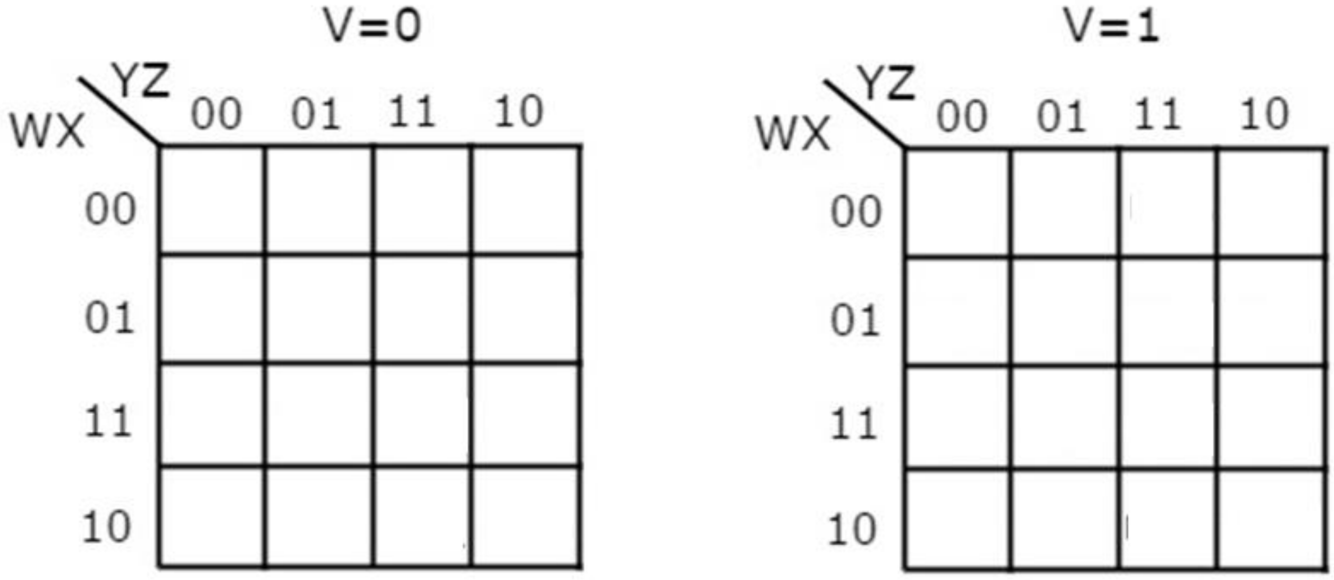
Five Variable K-Map

- F(A,B,C,D,E) = ∑m(0,2,4,6,9,11,13,15,17,21,25,27,29,31)



Five Variable K-Map

- $F(A,B,C,D,E) = \sum m(0,1,7,9,11,13,15,16,17,23,25,27)$



Five Variable K-Map

- F(A,B,C,D,E) =

∑m(0,1,3,4,6,8,9,1,,13,14,16,19,20,21,22,24,25)+ ∑d(5,7,12,15,17,23)

AB \ CDE	000	001	011	010	110	111	101	100
00	0	1	3	2	6	7	5	4
01	8	9	11	10	14	15	13	12
11	24	25	27	26	30	31	29	28
10	16	17	19	18	22	23	21	20

Six variable K-map

$\bar{x}\bar{y}$

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

$\bar{x}y$

zu \ vw	00	01	11	10
00	16 1	17	19	18
01	20	21 1	23 1	22
11	28	29 1	31 1	30
10	24	25	27	26

$x\bar{y}$

zu \ vw	00	01	11	10
00	32 1	33	35	34
01	36 1	37	39	38
11	44 1	45	47	46
10	40 1	41	43	42

xy

zu \ vw	00	01	11	10
00	48 1	49	51	50
01	52 1	53	55	54
11	60 1	61	63	62
10	56 1	57	59	58

Quine-McCluskey Tabular Method

- Quine-McCluskey tabular method is a method based on the concept of prime implicants. We know that prime implicant is a product or sum term, which can't be further reduced by combining with any other product or sum terms of the given Boolean function.

- This tabular method is useful to get the prime implicants by repeatedly using the following Boolean identity.

- $xy + xy' = xy+y' = x.1 = x$

Quine-McCluskey Tabular Method

- **Step 1** − Arrange the given min terms in an **ascending order** and make the groups based on the number of ones present in their binary representations. So, there will be **at most ‘n+1’ groups** if there are ‘n’ Boolean variables in a Boolean function or ‘n’ bits in the binary equivalent of min terms.
- **Step 2** − Compare the min terms present in **successive groups**. If there is a change in only one-bit position, then take the pair of those two min terms. Place this symbol ‘_’ in the differed bit position and keep the remaining bits as it is.
- **Step 3** − Repeat step2 with newly formed terms till we get all **prime implicants**.
- **Step 4** − Formulate the **prime implicant table**. It consists of set of rows and columns. Prime implicants can be placed in row wise and min terms can be placed in column wise. Place ‘1’ in the cells corresponding to the min terms that are covered in each prime implicant.
- **Step 5** − Find the essential prime implicants by observing each column. If the min term is covered only by one prime implicant, then it is **essential prime implicant**. Those essential prime implicants will be part of the simplified Boolean function.
- **Step 6** − Reduce the prime implicant table by removing the row of each essential prime implicant and the columns corresponding to the min terms that are covered in that essential prime implicant. Repeat step 5 for Reduced prime implicant table. Stop this process when all min terms of given Boolean function are over.

Quine Mc'clusky Method

- Simplify- $f(a,b,c,d)=\sum m(0,1,2,5,6,7,8,9,10,14)$

Column I

	Minterm	Binary	
group 0	0	0000	✓
group 1	1	0001	✓
	2	0010	✓
	8	1000	✓
group 2	5	0101	✓
	6	0110	✓
	9	1001	✓
	10	1010	✓
group 3	7	0111	✓
	14	1110	✓

Column II

Minterms	Binary	
0, 1	000–	✓
0, 2	00–0	✓
0, 8	–000	✓
1, 5	0–01	
1, 9	–001	✓
2, 6	0–10	✓
2, 10	–010	✓
8, 9	100–	✓
8, 10	10–0	✓
5, 7	01–1	
6, 7	011–	
6, 14	–110	✓
10, 14	1–10	✓

Column III

Minterms	Binary
0, 1, 8, 9	–00–
0, 2, 8, 10	–0–0
~~0, 8, 1, 9~~	~~–00–~~
~~0, 8, 2, 10~~	~~–0–0~~
2, 6, 10, 14	-- 10
~~2, 10, 6, 14~~	~~-- 10~~

Quine Mc'clusky Method

- f = a'c'd + a'bd + a'bc + b'c' + b'd' + cd'
  (1, 5) (5, 7) (6, 7) (0, 1, 8, 9) (0, 2, 8, 10) (2, 6, 10,14)
- Prime Implicant Table-

PI	Minterms corresponds to PI	Given Minterms									
		0	1	2	5	6	7	8	9	10	14
a'c'd	1, 5										
a'bd	5, 7										
a'bc	6, 7										
b'c'	0, 1, 8, 9										
b'd'	0, 2, 8, 10										
cd'	2, 6, 10,14										

Quine Mc'clusky Method

- Prime Implicant Table-

PI	Minterms corresponds to PI	Given Minterms									
		0	1	2	5	6	7	8	9	10	14
a'c'd	1, 5		X		X						
a'bd	5, 7				X		X				
a'bc	6, 7					X	X				
b'c'	0, 1, 8, 9	X	X					X	X		
b'd'	0, 2, 8, 10	X		X				X		X	
cd'	2, 6, 10,14			X		X				X	X

f = a'bd + b'c' + cd'

Quine Mc'clusky Method

- Find the minimal SOP expression for the given Boolean function- $f(a,b,c,d)=\sum m(1,3,4,5,9,10,11) + \sum d(6,8)$

Group	Column-I		Column-II		Column-III	
	Minterms	Binary (abcd)	Minterm Pairs	Binary (abcd)	Minterm Pairs	Binary (abcd)
Group 0	1	0001	1-3	00-1 *	1-3-9-11	-0-1
	4	0100	1-5	0-01	1-9-3-11	-0-1
	8	1000	1-9	-001 *	8-9-10-11	10--
Group 1	3	0011	4-5	010-	8-10-9-11	10--
	5	0101	4-6	01-0		
	6	0110	8-9	100- *		
	9	1001	8-10	10-0 *		
	10	1010	3-11	-011 *		
Group 2	11	1011	9-11	10-1 *		
			10-11	101- *		

Quine Mc'clusky Method

- f = a'c'd + a'bc' + a'bd' + b'd + ab'

  (1,5) (4,5) (4,6) (1,3,9,11) (8,9,10,11)
- Prime Implicant Table-

Prime Implicants (PI)	Minterms corresponds to PI	Given Minterms						
		1	3	4	5	9	10	11
a'c'd	1, 5							
a'bc'	4, 5							
a'bd'	4, 6							
b'd	1, 3, 9, 11							
ab'	8, 9, 10, 11							

Quine Mc'clusky Method

- Prime Implicant Table-

PI	Minterms corresponds to PI	Given Minterms						
		1	3	4	5	9	10	11
a'c'd	1, 5	X			X			
a'bc'	4, 5			X	X			
a'bd'	4, 6			X				
b'd	1, 3, 9, 11	X	X			X		X
ab'	8, 9, 10, 11					X	X	X

$f = ab' + b'd + a'bc'$

Essential Prime Implicants

Thank You

Any Questions