# UNIT- 2
# BOOLEAN ALGEBRA & LOGIC GATES

TOPIC 2.3

4 VARIABLE, DON'T CARE CONDITION, QUINE-MCCLUSKY METHOD



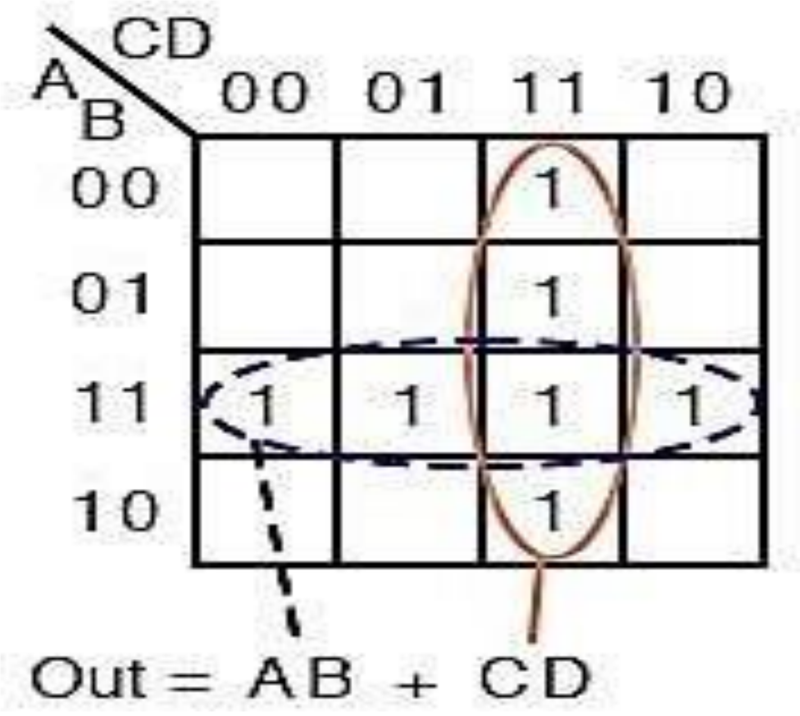
By -Prof. Sonali B. Jadhav

# 4 VARIABLE K-MAP: EXAMPLE

- This Boolean expression has seven product terms. They are mapped top to bottom and left to right on the K- map above.
- For example, the first P-term **A'B'CD** is the first row, 3rd cell, corresponding to map location **A=0, B=0, C=1, D=1**.
- The other product terms are placed in a similar manner. Encircling the largest groups possible, two groups of four are shown above.
- The dashed horizontal group corresponds to the simplified product term **AB**. The vertical group corresponds to Boolean CD. Since there are two groups, there will be two product terms in the Sum-Of- Products result of **Out=AB+CD**.

Example:
f(A,B,C,D)=

$$\bar{A}\bar{B}CD + \bar{A}BCD + ABCD + A\bar{B}CD + AB\bar{C}\bar{D} + AB\bar{C}D + ABC\bar{D}$$

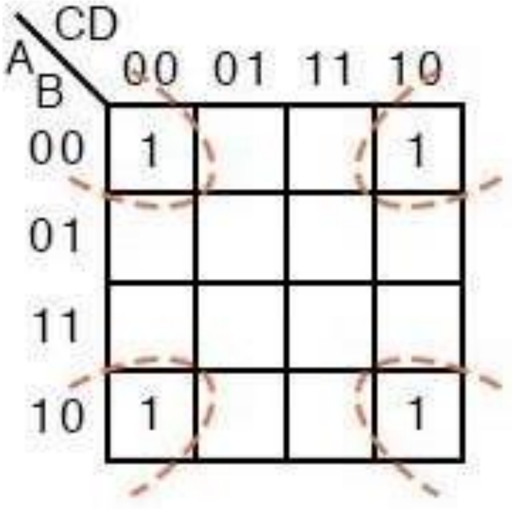


# 4 VARIABLE K-MAP: EXAMPLE

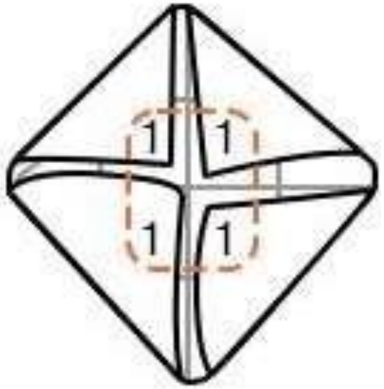
- Fold up the corners of the map here like it is a napkin to make the four cells physically adjacent.
- The four cells here are a group of four because they all have the Boolean variables **B'** and **D'** in common. In other words, **B=0** for the four cells, and **D=0** for the four cells.
- The other variables **(A, C)** are **0** in some cases, **1** in other cases with respect to the four corner cells.
- Thus, these variables **(A, C)** are not involved with this group of four. This single group comes out of the map as one product term for the simplified result: **Out=B'D'**

Example:
f(A,B,C,D)=

$$\bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} + A\bar{B}\bar{C}\bar{D} + A\bar{B}C\bar{D}$$




$\text{Out} = \bar{B}\bar{D}$

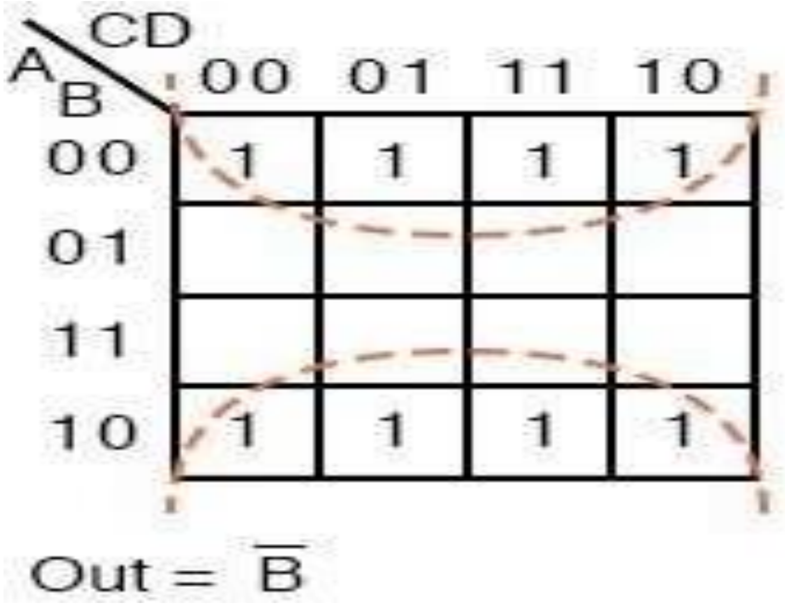


# 4VARIABLE K-MAP: EXAMPLE

- For the K-map here, roll the top and bottom edges  into a cylinder forming eight adjacent cells.
- This group of eight has one Boolean variable in common: **B=0**.
- Therefore, the one group of eight is covered by one  p-term: **B’**.
- The original eight-term Boolean expression  simplifies to **Out=B’**

Example:
f(A,B,C,D)=

$$\overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}\,\overline{C}D + \overline{A}\,\overline{B}CD + \overline{A}\,\overline{B}C\overline{D}$$
$$+ A\overline{B}\,\overline{C}\,\overline{D} + A\overline{B}\,\overline{C}D + A\overline{B}CD + A\overline{B}C\overline{D}$$



# MISSING-TERMS IN 4 VARIABLE K MAPS

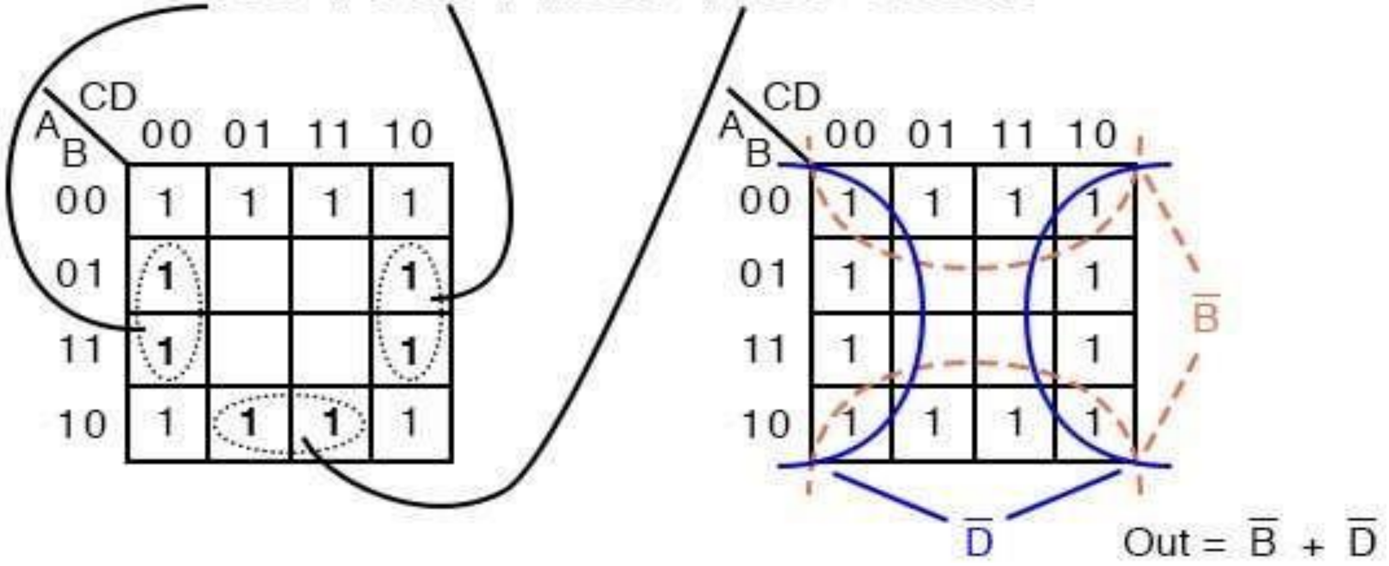
- The Boolean expression here has nine p-terms, three of which have three Booleans instead of four. The difference is that while four Boolean variable product terms cover one cell, the three Boolean p-terms cover a pair of cells each.
- The six product terms of four Boolean variables map in the usual manner above as single cells. The three Boolean variable terms (three each) map as cell pairs, which is shown above.
- Note that we are mapping p-terms into the K-map, not pulling them out at this point.
- For the simplification, we form two groups of eight. Cells in the corners are shared with both groups. This is fine. In fact, this leads to a better solution than forming a group of eight and a group of four without sharing any cells. Final Solution is **Out=B'+D'**

Example:

f(A,B,C,D)=

$$\overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}\,\overline{C}D + \overline{A}\,\overline{B}CD + \overline{A}\,\overline{B}C\overline{D} + B\overline{C}\,\overline{D} + BC\overline{D} + A\overline{B}\,\overline{C}\,\overline{D} + A\overline{B}D + A\overline{B}C\overline{D}$$

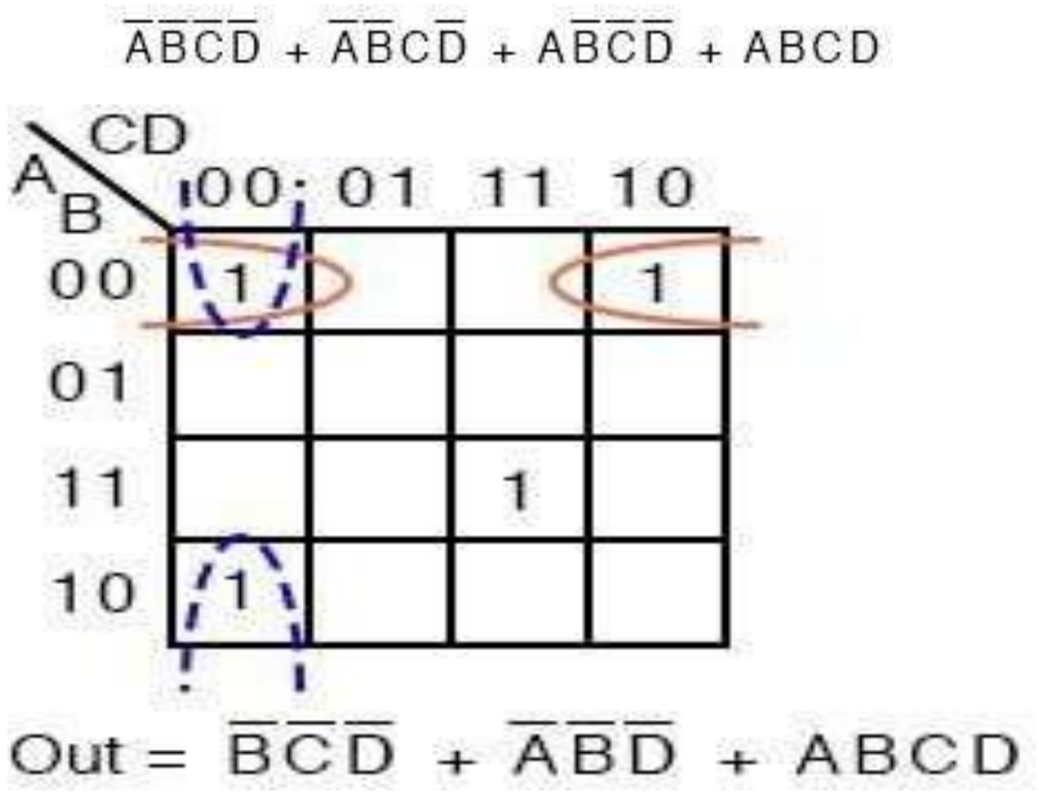
$$Out = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}\,\overline{C}D + \overline{A}\,\overline{B}CD + \overline{A}\,\overline{B}C\overline{D} + B\overline{C}\,\overline{D} + BC\overline{D} + A\overline{B}\,\overline{C}\,\overline{D} + A\overline{B}D + A\overline{B}C\overline{D}$$



# 4VARIABLE K-MAP: EXAMPLE

- Here we map the un-simplified Boolean expression to the Karnaugh map.
- Three of the cells form into groups of two cells.
- A fourth cell cannot be combined with anything, which often happens in "real world" problems. In this case, the Boolean p-term **ABCD** is unchanged in the simplification process.
- Result:
  - **Out= B'C'D'+A'B'D'+ABCD**

Example:
f(A,B,C,D)=

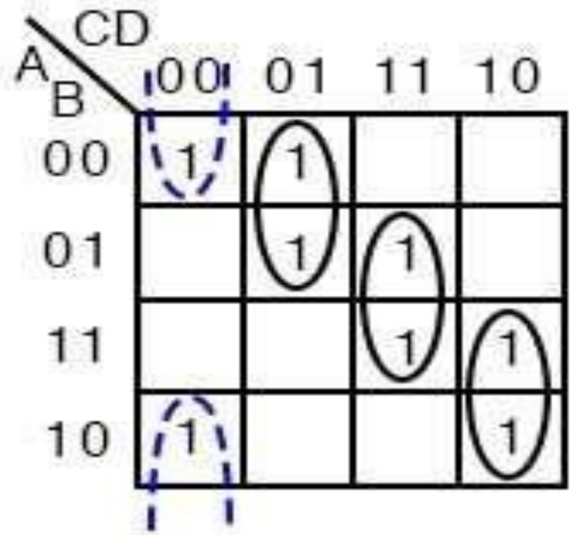


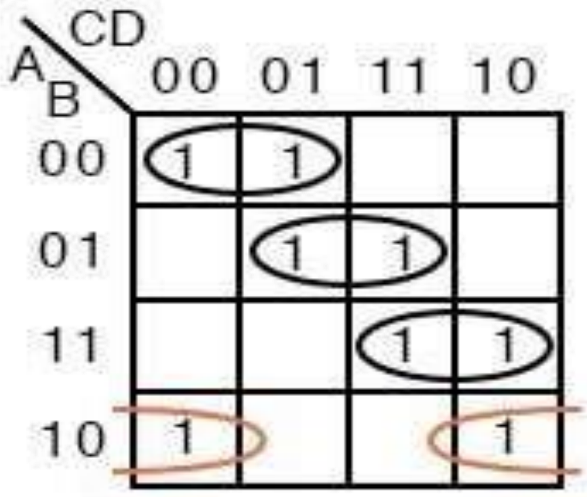
# 4 VARIABLE K-MAP: EXAMPLE

- Often times there is more than one minimum cost solution to a simplification problem. Such is the case illustrated below.
- Both results above have four product terms of three Boolean variable each. Both are equally valid *minimal cost* solutions.
- The difference in the final solution is due to how the cells are grouped as shown here.
- A minimal cost solution is a valid logic design with the minimum number of gates with the minimum number of inputs.

Example:
f(A,B,C,D)=

$$\bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}B\bar{C}D + \bar{A}BCD + ABCD + ABC\bar{D} + A\bar{B}\bar{C}\bar{D} + A\bar{B}C\bar{D}$$




$$Out = \bar{B}\bar{C}\bar{D} + \bar{A}\bar{C}D + BCD + AC\bar{D}$$

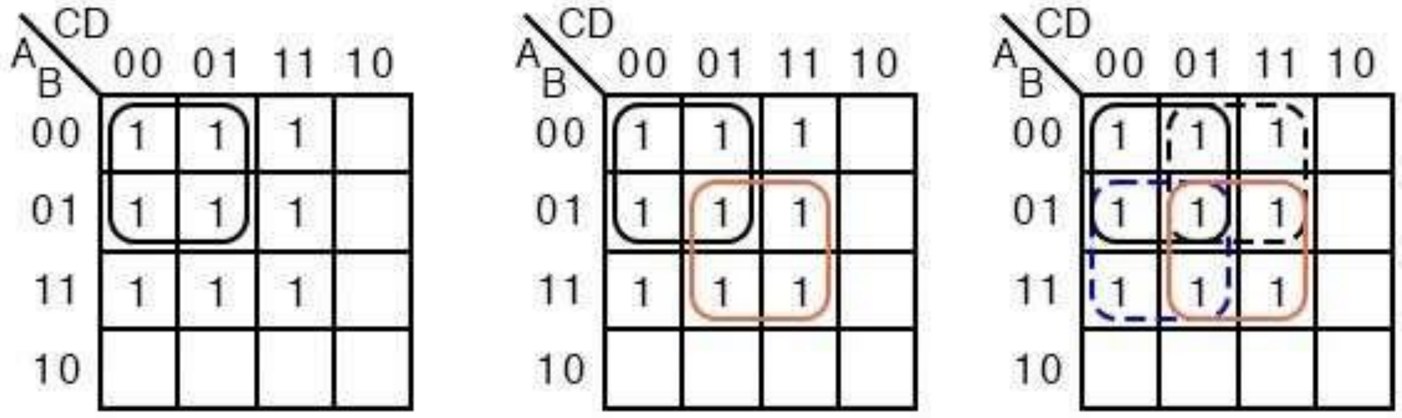
$$Out = \bar{A}\bar{B}\bar{C} + \bar{A}BD + ABC + A\bar{B}\bar{D}$$

# 4VARIABLE K-MAP: EXAMPLE

- Below we map the un-simplified Boolean equation as usual and form a group of four as a first simplification step. It may not be obvious how to pick up the remaining cells.
- Pick up three more cells in a group of four, center above. There are still two cells remaining. the minimal cost method to pick up those is to group them with neighboring cells as groups of four as at above right.
- On a cautionary note, do not attempt to form groups of three.
- Groupings must be powers of 2, that is, 1, 2, 4, 8 ...

Example:
f(A,B,C,D)=

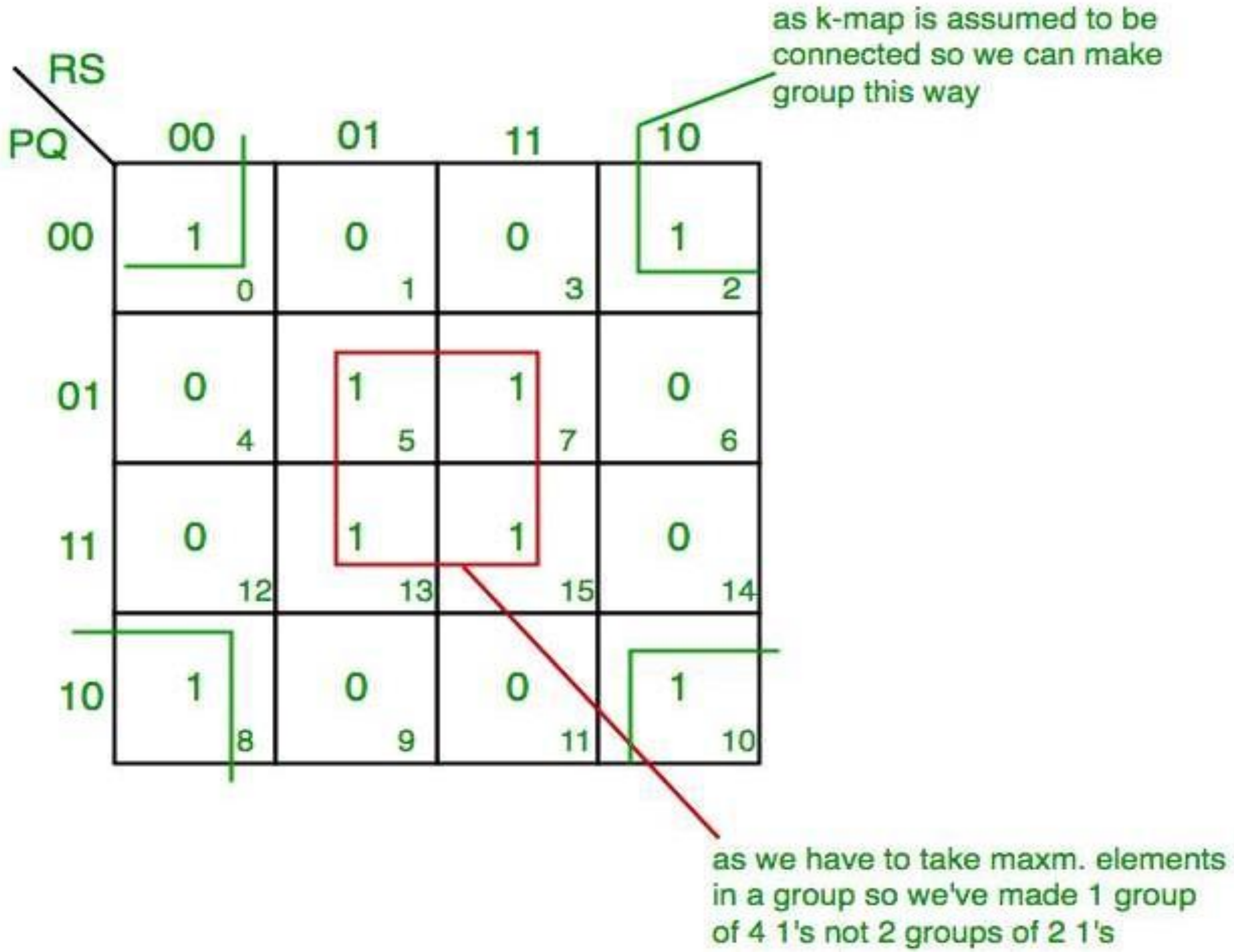
$$\overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}\,\overline{C}D + \overline{A}\,\overline{B}CD + \overline{A}B\overline{C}\,\overline{D} + \overline{A}B\overline{C}D + \overline{A}BCD + AB\overline{C}\,\overline{D} + AB\overline{C}D + ABCD$$




$$Out = \overline{A}\,\overline{C} + \overline{A}D + B\overline{C} + BD$$

# K-MAP FOR 4 VARIABLES (SOP)

- F(P,Q,R,S)=∑m(0,2,5,7,8,10,13,15)
- From **red** group we get product term—
  - QS
- From **green** group we get product term—
  - Q'S'
- Summing these product terms we get-
  - **Final expression (QS+Q'S')**



# K-MAP FOR 4 VARIABLES (POS)

- F(A,B,C,D)=π(3,5,7,8,10,11,12,13)
- We find groping from each color as follows:
- Blue group = (A+C'+D')
- Green group = (B'+C+D')
- Orange group = (A'+C+D)
- Pink group = (A'+B+C')

- Finally we express these as product –
- **= (A+C'+D').(B'+C+D').(A'+C+D).(A'+B+C')**

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | 0 | |
| 01 | | 0 | 0 | |
| 11 | 0 | 0 | | |
| 10 | 0 | | 0 | 0 |

# 4VARIABLE K-MAP: EXAMPLE (Homework)

Eg. 1- Simplify following function using K-map

$F(A,B,C,D) = \sum m\,(1,2,3,4,6,8,10,14,15)$

Eg. 2- Simplify following function using K-map

$Y = A'B'C'D + A'B'CD + A'B'CD' + A'BCD' + ABC'D' + ABC'D + ABCD' + AB'C'D' + AB'C'D + AB'C$

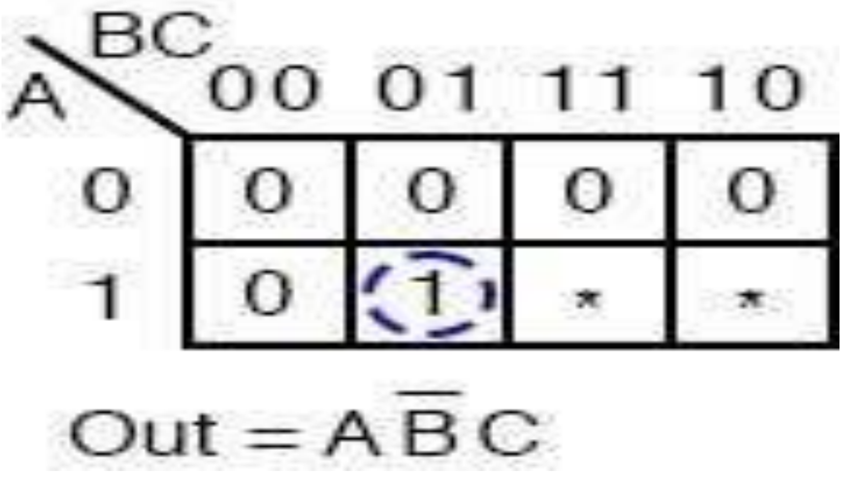
Eg. 3- Simplify following function using K-map. The don't care conditions are indicated by d()

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

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

# K-MAP FOR 4 VARIABLES (DON'T CARES)

- Don't cares in a Karnaugh map, or truth table, may be either **1**s or **0**s, as long as we don't care what the output is for an input condition we never expect to see. We plot these cells with an asterisk, *, among the normal **1**s and **0**s.
- When forming groups of cells, treat the don't care cell as either a **1** or a **0**, or ignore the don't cares.
- This is helpful if it allows us to form a larger group than would otherwise be possible without the don't cares. There is no requirement to group all or any of the don't cares.
- Only use them in a group if it simplifies the logic.



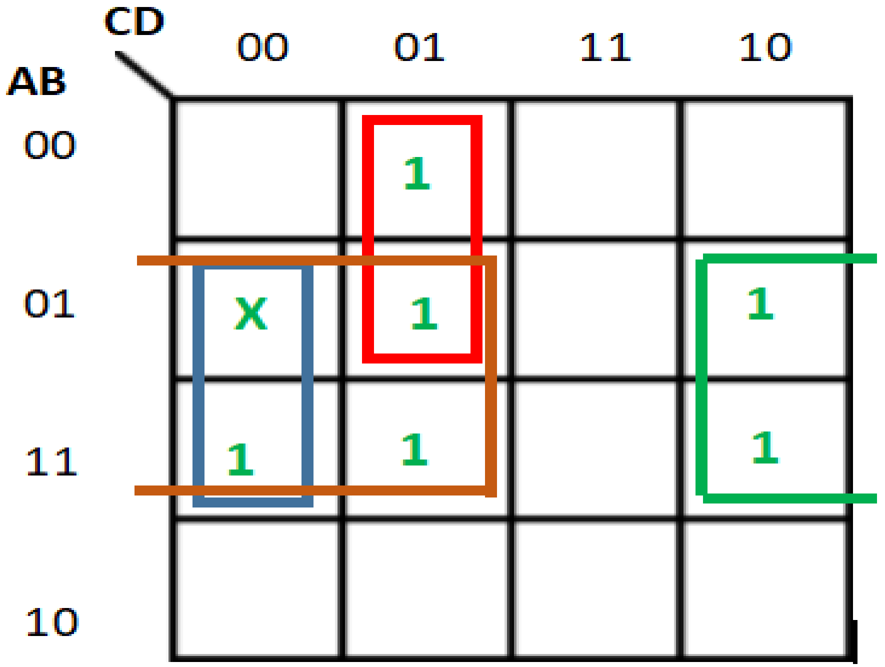
**Example-1:**

Minimize the following function in SOP minimal form using K-Maps:

f = m(1, 5, 6, 12, 13, 14) + d(4)

**Explanation:**

The SOP K-map for the given expression is:



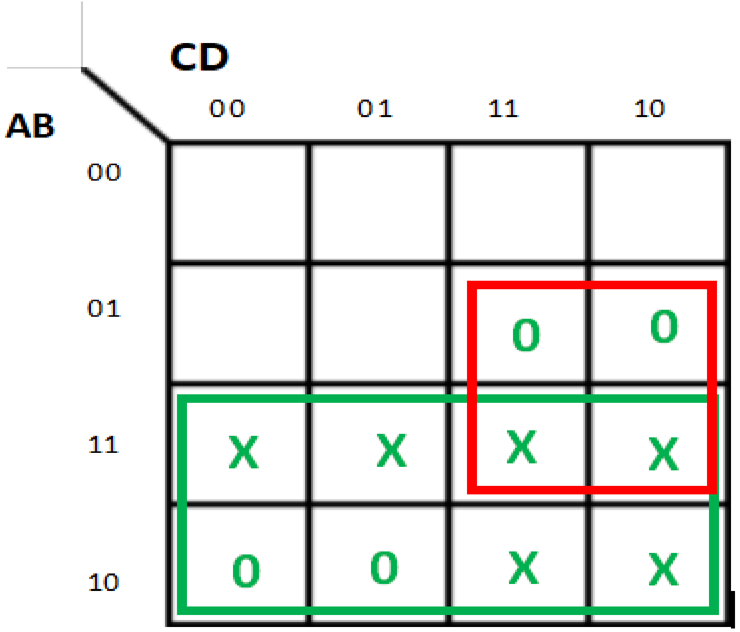

Therefore, SOP minimal is,

f = BC' + BD' + A'C'D

**Example-2:**

## Minimize the following function in POS minimal form using K-Maps:

F(A, B, C, D) = m(0, 1, 2, 3, 4, 5) + d(10, 11, 12, 13, 14, 15)




Explanation:

Writing the given expression in POS form:

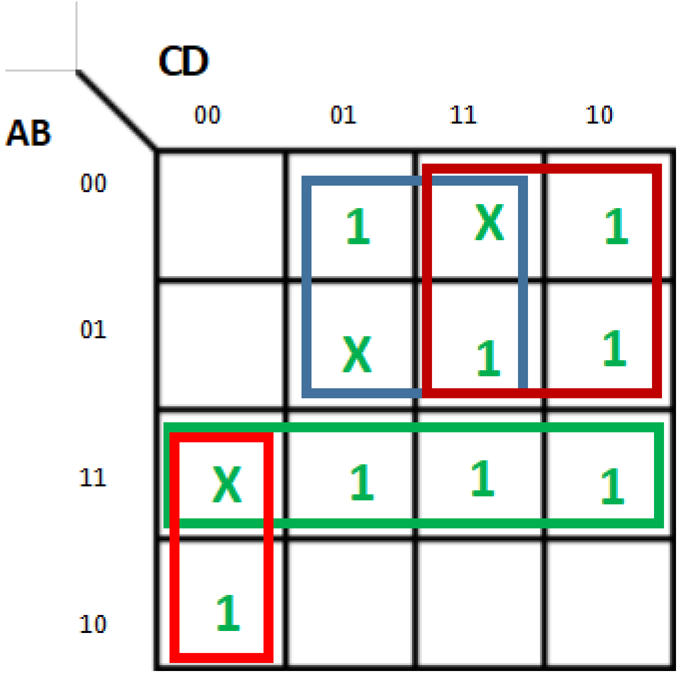
F(A, B, C, D) = M(6, 7, 8, 9) + d(10, 11, 12, 13, 14, 15)

Therefore, POS minimal is,

F = A'(B' + C')

Example-3:
Minimize the following function in SOP minimal form using K-Maps: F(A, B, C, D) = m(1, 2, 6, 7, 8, 13, 14, 15) + d(3, 5, 12)

Explanation:
The SOP K-map for the given expression is:




Therefore,

f = AC'D' + A'D + A'C + AB

# Significance of "Don't Care" Conditions:

Don't Care conditions has the following significance with respect to the digital circuit design:

**Simplification:**

These conditions denotes the set of inputs which never occurs for a given digital circuits. Thus, they are being used to further simplify the boolean output expression.

**Lesser number of gates:**

Simplification reduces the number of gates to be used for implementing the given expression. Therefore, don't cares make the digital circuit design more economical.

**Reduced Power Consumption:**

While grouping the terms long with don't cares reduces switching of the states. This decreases the required memory space which in turn results in less power consumption.

**States in Code Converters:**

These are used in code converters. For example- In design of 4-bit BCD-to-XS-3 code converter, the input combinations 1010, 1011, 1100, 1101, 1110, and 1111 are don't cares.

**Prevention of Hazards:**

Don't cares also prevents hazards in digital systems.

# QUINE-McCLUSKEY TABULAR METHOD

- Quine-McClukey tabular method is a tabular method based on the concept of prime implicants.
- The 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.
- Quine-McClukey tabular method is a tabular 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.

# QUINE-McCLUSKEY TABULAR METHOD: STEPS FOR SIMPLIFYING BOOLEAN FUNCTIONS

- **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**.

# QUINE-McCLUSKEY TABULAR METHOD: STEPS FOR SIMPLIFYING BOOLEAN FUNCTIONS

- **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-McCLUSKEY TABULAR METHOD: EXAMPLE: STEP 1

- simplify the following Boolean function, $f(W,X,Y,Z)=\sum m(2,6,8,9,10,11,14,15)$ using Quine-McClukey tabular method.

- ✓ The given Boolean function is in **sum of min terms** form.
- ✓ It is having 4 variables W,X,Y & Z.
- ✓ The given min terms are 2, 6,8, 9, 10, 11, 14 and 15.
- ✓ The ascending order of these min terms based on the number of ones present in their binary equivalent is 2,8, 6, 9, 10, 11, 14 and 15.
- ✓ The following table shows these **min terms and their equivalent binary** representations.

| Min Term | Binary |
|---|---|
| 2 | 0010 |
| 6 | 0110 |
| 8 | 1000 |
| 9 | 1001 |
| 10 | 1010 |
| 11 | 1011 |
| 14 | 1110 |
| 15 | 1111 |

| Group Name | Min terms | W | X | Y | Z |
|---|---|---|---|---|---|
| GA1 | 2 | 0 | 0 | 1 | 0 |
| | 8 | 1 | 0 | 0 | 0 |
| GA2 | 6 | 0 | 1 | 1 | 0 |
| | 9 | 1 | 0 | 0 | 1 |
| | 10 | 1 | 0 | 1 | 0 |
| GA3 | 11 | 1 | 0 | 1 | 1 |
| | 14 | 1 | 1 | 1 | 0 |
| GA4 | 15 | 1 | 1 | 1 | 1 |

# QUINE-McCLUSKEY TABULAR METHOD: EXAMPLE: STEP 2

- The given min terms are arranged into 4 groups based on the number of ones present in their binary equivalents. The following table shows the possible **merging of min terms** from adjacent groups.

| Group Name | Min terms | W | X | Y | Z |
|---|---|---|---|---|---|
| GB1 | 2,6 | 0 | - | 1 | 0 |
| | 2,10 | - | 0 | 1 | 0 |
| | 8,9 | 1 | 0 | 0 | - |
| | 8,10 | 1 | 0 | - | 0 |
| GB2 | 6,14 | - | 1 | 1 | 0 |
| | 9,11 | 1 | 0 | - | 1 |
| | 10,11 | 1 | 0 | 1 | - |
| | 10,14 | 1 | - | 1 | 0 |
| GB3 | 11,15 | 1 | - | 1 | 1 |
| | 14,15 | 1 | 1 | 1 | - |

# QUINE-McCLUSKEY TABULAR METHOD: EXAMPLE: STEP 3

- The min terms, which are differed in only one-bit position from adjacent groups are merged. That differed bit is represented with this symbol, '-'. In this case, there are three groups and each group contains combinations of two min terms. The following table shows the possible **merging of min term pairs** from adjacent groups.

| Group Name | Min terms | W | X | Y | Z |
|---|---|---|---|---|---|
| GB1 | 2,6,10,14 | - | - | 1 | 0 |
| | 2,10,6,14 | - | - | 1 | 0 |
| | 8,9,10,11 | 1 | 0 | - | - |
| | 8,10,9,11 | 1 | 0 | - | - |
| GB2 | 10,11,14,15 | 1 | - | 1 | - |
| | 10,14,11,15 | 1 | - | 1 | - |

# QUINE-McCLUSKEY TABULAR METHOD: EXAMPLE: STEP 4

- The successive groups of min term pairs, which are differed in only one-bit position are merged. That differed bit is represented with this symbol, '-'. In this case, there are two groups and each group contains combinations of four min terms. Here, these combinations of 4 min terms are available in two rows. So, we can remove the repeated rows. The reduced table after removing the redundant rows is shown below.

| Group Name | Min terms | W | X | Y | Z |
|---|---|---|---|---|---|
| **GC1** | **2,6,10,14** | **-** | **-** | **1** | **0** |
| | **8,9,10,11** | **1** | **0** | **-** | **-** |
| **GC2** | **10,11,14,15** | **1** | **-** | **1** | **-** |

# QUINE-McCLUSKEY TABULAR METHOD: EXAMPLE: STEP 5

- Further merging of the combinations of min terms from adjacent groups is not possible, since they are differed in more than one-bit position. There are three rows in the above table. So, each row will give one prime implicant. Therefore, the **prime implicants** are YZ', WX' & WY.
- The **prime implicant table** is shown below.

| Min terms / Prime Implicants | 2 | 6 | 8 | 9 | 10 | 11 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|
| **YZ'** | 1 | 1 | | | 1 | | 1 | |
| **WX'** | | | 1 | 1 | 1 | 1 | | |
| **WY** | | | | | 1 | 1 | 1 | 1 |

# QUINE-McCLUSKEY TABULAR METHOD: EXAMPLE: STEP 6

- The prime implicants are placed in row wise and min terms are placed in column wise. 1s are placed in the common cells of prime implicant rows and the corresponding min term columns.
- The min terms 2 and 6 are covered only by one prime implicant **YZ'**. So, it is an **essential prime implicant**. This will be part of simplified Boolean function. Now, remove this prime implicant row and the corresponding min term columns. The reduced prime implicant table is shown below.

| Min terms / Prime Implicants | 8 | 9 | 11 | 15 |
|---|---|---|---|---|
| **WX'** | 1 | 1 | 1 | |
| **WY** | | | 1 | 1 |

# QUINE-MCCLUSKEY TABULAR METHOD: EXAMPLE: STEP 7

- The min terms 8 and 9 are covered only by one prime implicant **WX’**. So, it is an **essential prime implicant**. This will be part of simplified Boolean function. Now,remove this prime implicant row and the corresponding min term columns.The reduced prime implicant table is shown below.

| Min terms / Prime Implicants | 15 |
|---|---|
| **WY** | 1 |

The min term 15 is covered only by one prime implicant WY.So,it is an essential prime implicant.This will be part of simplified Boolean function.
In this example problem, we got three prime implicants and all the three are essential.
Therefore, the simplified Boolean function is

**F(W,X,Y,Z) =YZ’ +WX’ +WY.**

# QUINE-McCLUSKEY TABULAR METHOD: EXAMPLE: 2

$$F(W,X,Y,Z) = \Sigma(5,7,9,11,13,15)$$

| | Step 1 | | | Step 2 | | | Step 3 | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | | 5 | | | | | | | |
| | | 9 | | | | | | | |
| | | | | | | | | | |
| | | 7 | | | | | | | |
| 3 | | 11 | | | | | | | |
| | | 13 | | | | | | | |
| | | | | | | | | | |
| 4 | | 15 | | | | | | | |

List minterms by the number of 1s it contains.

F(W,X,Y,Z) = Σ(5,7,9,11,13,15)

| Step 1 | | | Step 2 | | | Step 3 | | |
|---|---|---|---|---|---|---|---|---|
| | 5 | 0101 | | 5,7 | | | | |
| | 9 | 1001 | 2 | 5,13 | | | | |
| | | | | 9,11 | | | | |
| | 7 | 0111 | | 9,13 | | | | |
| | 11 | 1011 | | | | | | |
| | 13 | 1101 | | 7,15 | | | | |
| | | | 3 | 11,15 | | | | |
| | 15 | 1111 | | 13,15 | | | | |

Enter combinations of minterms by the number of 1s it contains.

$F(W,X,Y,Z) = \sum(5,7,9,11,13,15)$

| Step 1 | | | Step 2 | | | Step 3 | | |
|---|---|---|---|---|---|---|---|---|
| ⌧ | 5 | 0101 | | 5,7 | 01-1 | | | |
| ⌧ | 9 | 1001 | | 5,13 | -101 | | | |
| | | | | 9,11 | 10-1 | | | |
| ⌧ | 7 | 0111 | | 9,13 | 1-01 | | | |
| ⌧ | 11 | 1011 | | | | | | |
| ⌧ | 13 | 1101 | | 7,15 | -111 | | | |
| | | | | 11,15 | 1-11 | | | |
| ⌧ | 15 | 1111 | | 13,15 | 11-1 | | | |

Check off elements used from Step 1.

$$F(W,X,Y,Z) = \Sigma(5,7,9,11,13,15)$$

| Step 1 | | | Step 2 | | | Step 3 | | |
|---|---|---|---|---|---|---|---|---|
| ⌧ | 5 | 0101 | | 5,7 | 01-1 | | 5,7,13,15 | -1-1 |
| ⌧ | 9 | 1001 | | 5,13 | -101 | | 5,13,7,15 | -1-1 |
| | | | | 9,11 | 10-1 | | 9,11,13,15 | 1--1 |
| ⌧ | 7 | 0111 | | 9,13 | 1-01 | | 9,13,11,15 | 1--1 |
| ⌧ | 11 | 1011 | | | | | | |
| ⌧ | 13 | 1101 | | 7,15 | -111 | | | |
| | | | | 11,15 | 1-11 | | | |
| ⌧ | 15 | 1111 | | 13,15 | 11-1 | | | |

Enter combinations of minterms by the number of 1s it contains.

$$F(W,X,Y,Z) = \Sigma(5,7,9,11,13,15)$$

| Step 1 | | | Step 2 | | | Step 3 | | |
|---|---|---|---|---|---|---|---|---|
| ⌧ | 5 | 0101 | ⌧ | 5,7 | 01-1 | | 5,7,13,15 | -1-1 |
| ⌧ | 9 | 1001 | ⌧ | 5,13 | -101 | | 5,13,7,15 | -1-1 |
| | | | ⌧ | 9,11 | 10-1 | | 9,11,13,15 | 1- -1 |
| ⌧ | 7 | 0111 | ⌧ | 9,13 | 1-01 | | 9,13,11,15 | 1- -1 |
| ⌧ | 11 | 1011 | | | | | | |
| ⌧ | 13 | 1101 | ⌧ | 7,15 | -111 | | | |
| | | | ⌧ | 11,15 | 1-11 | | | |
| ⌧ | 15 | 1111 | ⌧ | 13,15 | 11-1 | | | |

The entries left unchecked are Prime Implicants.

## Finding Essential Prime Implicants (EPIs)

| | Prime Implicants | Covered Minterms | Minterms 5 | 7 | 9 | 11 | 13 | 15 |
|---|---|---|---|---|---|---|---|---|
| | - 1 - 1 | 5,7,13,15 | X | X | | | X | X |
| | 1 - - 1 | 9,13,11,15 | | | X | X | X | X |
| | | | | | | | | |
| | | | | | | | | |

Enter Xs for the minterms covered.

| | Prime Implicants | Covered Minterms | Minterms 5 | 7 | 9 | 11 | 13 | 15 |
|---|---|---|---|---|---|---|---|---|
| | - 1 - 1 | 5,7,13,15 | X | X | | | X | X |
| | 1 - - 1 | 9,13,11,15 | | | X | X | X | X |
| | | | | | | | | |
| | | | | | | | | |

Circle Xs that are in a column singularly.

XThe circled Xsare the Essential Prime Implicants,so we check them off.

We check off the minterms covered by eachof the EPIs.

| | Prime Implicants | Covered Minterms | Minterms | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 5 | 7 | 9 | 11 | 13 | 15 |
| ⌧ | - 1 - 1 | 5,7,13,15 | X | X | | | X | X |
| ⌧ | 1 - - 1 | 9,13,11,15 | | | X | X | X | X |
| | | | | | | | | |
| | | | ⌧ | ⌧ | ⌧ | ⌧ | ⌧ | ⌧ |

EPIs:

| W | X | Y | Z |
|---|---|---|---|
| - | 1 | - | 1 |
| 1 | - | - | 1 |

$$F = (X . Z) + (W . Z)$$
$$= (X + W) . Z$$