Karnaugh Mapping

Digital Electronics

Prof. Rasmita Lenka

Karnaugh Mapping or K-Mapping

This presentation will demonstrate how to

- Create and label two, three, & four variable K-Maps.
- Use the K-Mapping technique to simplify logic designs with two, three, and four variables.
- Use the K-Mapping technique to simplify logic design containing *don't care* conditions.

$$F = \bar{A}B\bar{C} + \bar{A}BC + \bar{A}\bar{B}C + A\bar{B}C$$

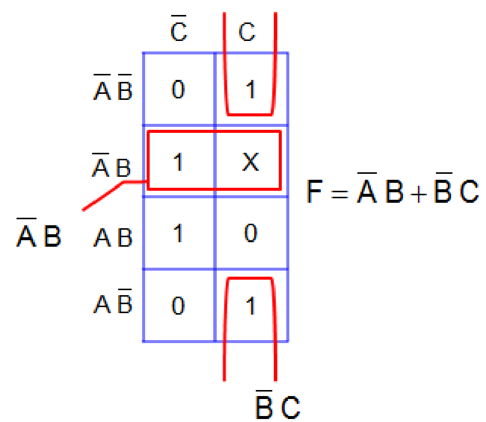
$$F = \bar{A}B\left(\bar{C} + C\right) + \bar{A}\bar{B}C + A\bar{B}C$$

$$F = \bar{A}B + \bar{A}\bar{B}C + A\bar{B}C$$

$$F = \bar{A}B + \bar{B}C\left(\bar{A} + A\right)$$

$$F = \bar{A}B + \bar{B}C$$

$$\equiv$$



Boolean Algebra Simplification

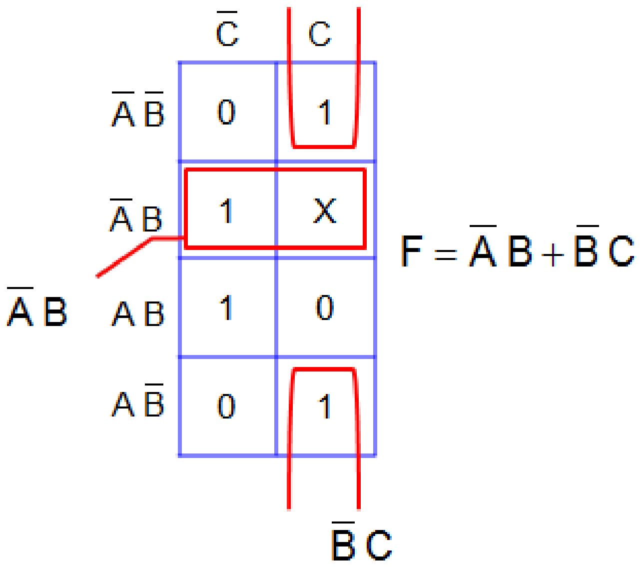
K-Mapping Simplification

Karnaugh Map Technique

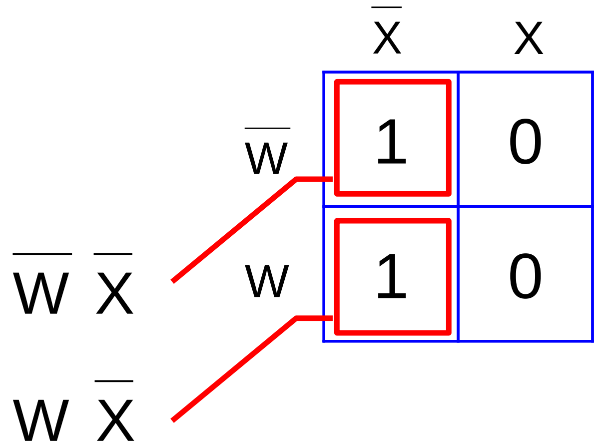
- K-Maps are a graphical technique used to simplify a logic equation.
- K-Maps are procedural and much *cleaner* than Boolean simplification.
- K-Maps can be used for any number of input variables, BUT are only practical for two, three, and four variables.

K-Map Format

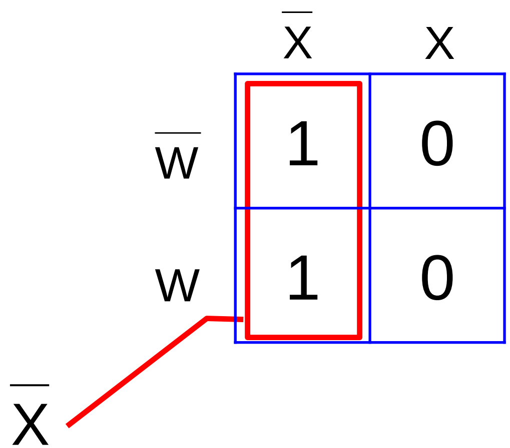
- Each minterm in a truth table corresponds to a cell in the K-Map.
- K-Map cells are labeled such that both horizontal and vertical movement diffe only by one variable.
- Since the adjacent cells differ by only one variable, they can be grouped to create simpler terms in the sum-of-products expression.
- The sum-of-products expression for the logic function can be obtained by OR-ing together the cells or group of cells that contain 1s.



Adjacent Cells = Simplification



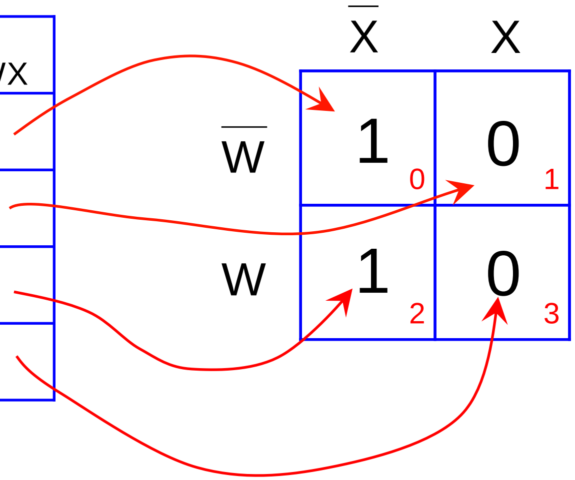
$$\overline{w}\ \overline{x} + w\ \overline{x} = \overline{x}\left(\overline{w} + w\right) = \overline{x}$$



Truth Table to K-Map Mapping

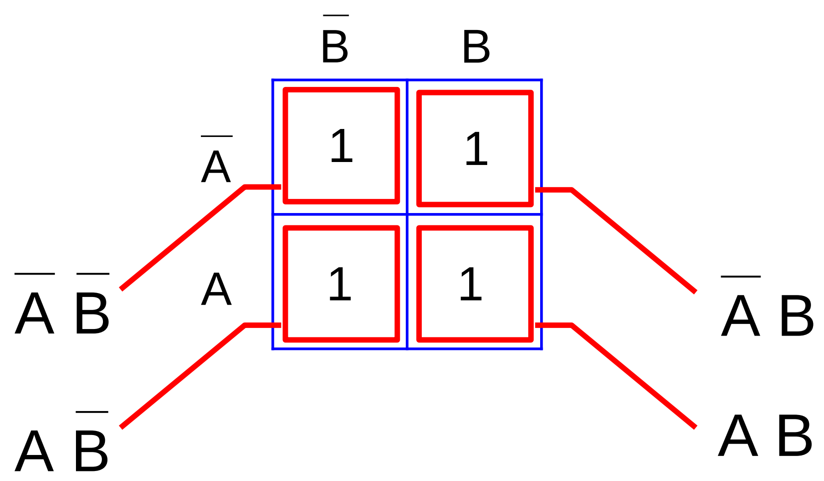
Two Variable K-Map

	W	X	F_{WX}
Minterm – 0	0	0	1
Minterm – 1	0	1	0
Minterm – 2	1	0	1
Minterm – 3	1	1	0



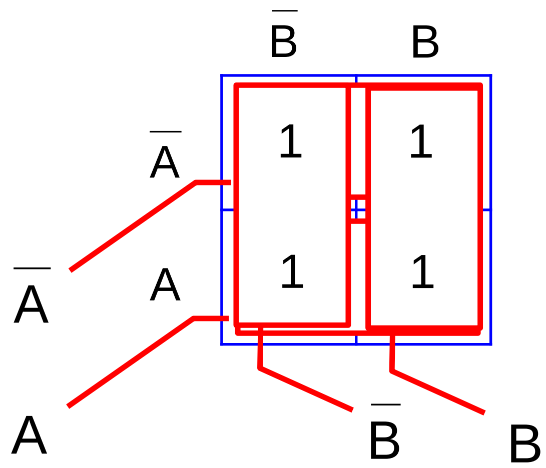
Two Variable K-Map Groupings

Groups of One – 4



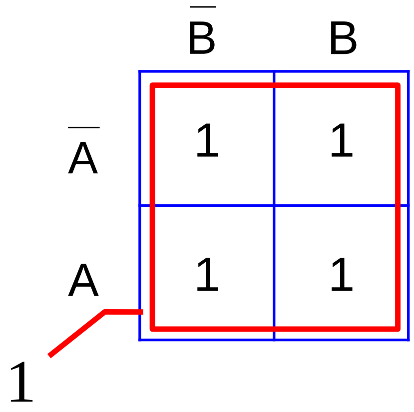
Two Variable K-Map Groupings

Groups of Two – 4



Two Variable K-Map Groupings

Group of Four – 1



K-Map Simplification Process

1. Construct a label for the K-Map. Place 1s in cells corresponding to the 1s in the truth table. Place 0s in the other cells.
2. Identify and group all <u>isolated 1's</u>. Isolated 1's are ones that cannot be grouped with any other one, or can only be grouped with one other adjacent one.
3. Group any hex.
4. Group any octet, even if it contains <u>some</u> 1s already grouped but not enclosed in a hex.
5. Group any quad, even if it contains <u>some</u> 1s already grouped but not enclosed in a hex or octet.
6. Group any pair, even if it contains <u>some</u> 1s already grouped but not enclosed in a hex, octet, or quad.
7. OR together all terms to generate the SOP equation.

Example #1: 2 Variable K-Map

Example:

After labeling and transferring the truth table data into the K-Map, write the simplified sum-of-products (SOP) logic expression for the logic function F_1.

J	K	F_1
0	0	1
0	1	1
1	0	0
1	1	0

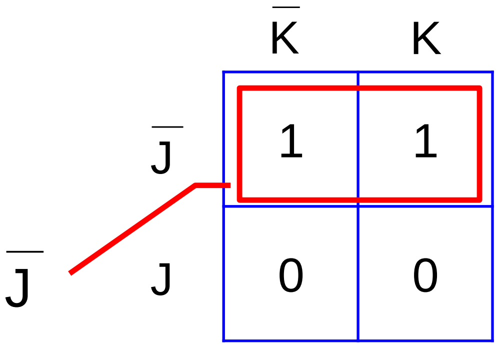
Example #1: 2 Variable K-Map

Example:

After labeling and transferring the truth table data into the K-Map, write the simplified sum-of-products (SOP) logic expression for the logic function F_1.

Solution:

J	K	F_1
0	0	1
0	1	1
1	0	0
1	1	0

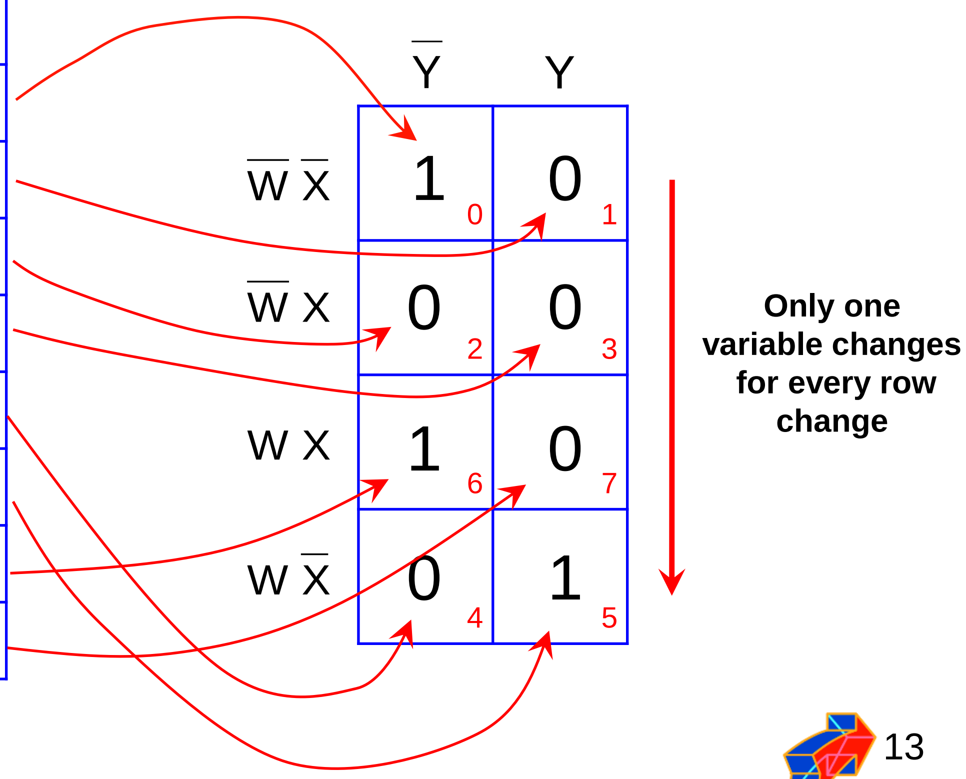


$$F_1 = \overline{J}$$

Truth Table to K-Map Mapping

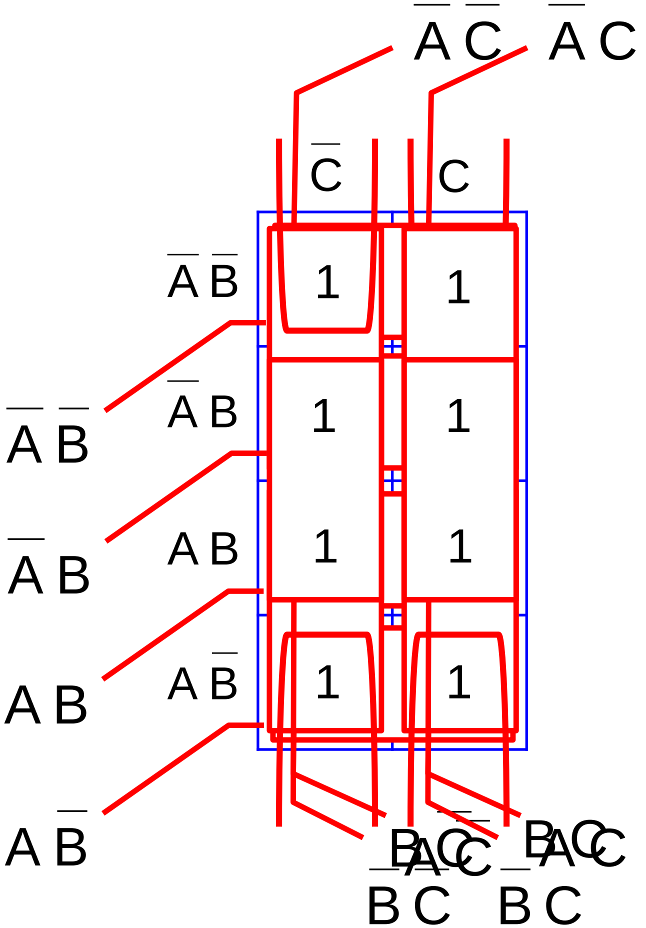
Three Variable K-Map

	W	X	Y	F_{WXY}
Minterm – 0	0	0	0	1
Minterm – 1	0	0	1	0
Minterm – 2	0	1	0	0
Minterm – 3	0	1	1	0
Minterm – 4	1	0	0	0
Minterm – 5	1	0	1	1
Minterm – 6	1	1	0	1
Minterm – 7	1	1	1	0



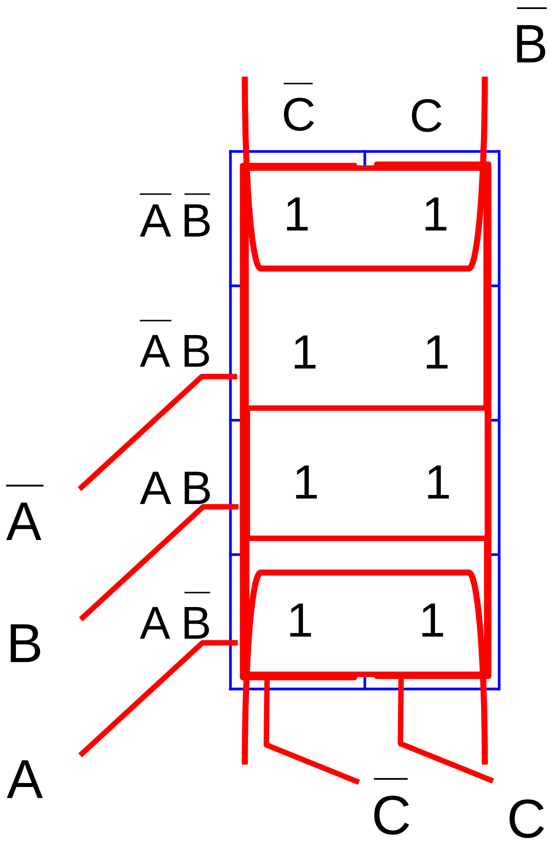
Three Variable K-Map Groupings

Groups of One – 8 (not shown)
Groups of Two – 12



Three Variable K-Map Groupings

Groups of Four – 6



Three Variable K-Map Groupings

Group of Eight - 1

	$\bar{C}$	C
$\bar{A}\bar{B}$	1	1
$\bar{A}B$	1	1
AB	1	1
$A\bar{B}$	1	1

1

Example #2: 3 Variable K-Map

Example:

After labeling and transferring the truth table data into the K-Map, write the simplified sum-of-products (SOP) logic expression for the logic function F_2.

E	F	G	F_2
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

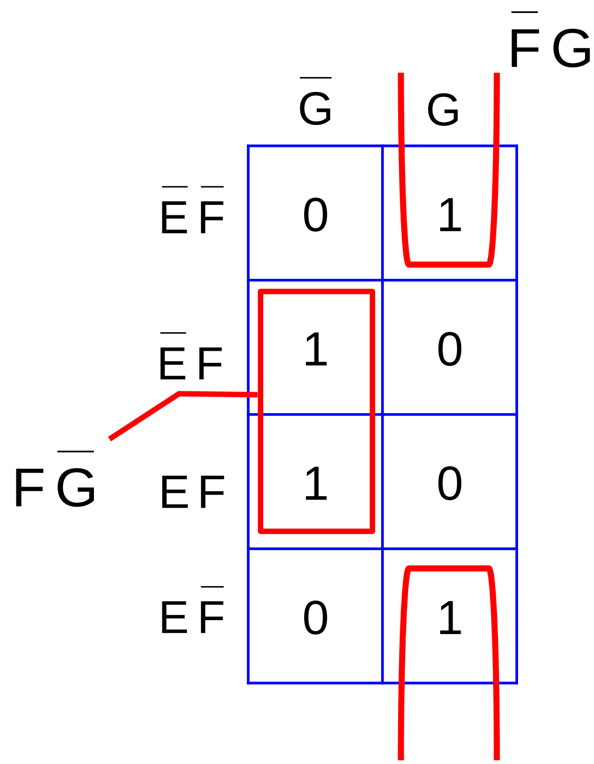
Example #2: 3 Variable K-Map

Example:

After labeling and transferring the truth table data into the K-Map, write the simplified sum-of-products (SOP) logic expression for the logic function F_2.

Solution:

E	F	G	F_2
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0



$$F_2 = F\overline{G} + \overline{F}G$$

Truth Table to K-Map Mapping

Four Variable K-Map

	W	X	Y	Z	F_{WXYZ}
Minterm – 0	0	0	0	0	0
Minterm – 1	0	0	0	1	1
Minterm – 2	0	0	1	0	1
Minterm – 3	0	0	1	1	0
Minterm – 4	0	1	0	0	1
Minterm – 5	0	1	0	1	1
Minterm – 6	0	1	1	0	0
Minterm – 7	0	1	1	1	1
Minterm – 8	1	0	0	0	0
Minterm – 9	1	0	0	1	0
Minterm – 10	1	0	1	0	1
Minterm – 11	1	0	1	1	0
Minterm – 12	1	1	0	0	1
Minterm – 13	1	1	0	1	0
Minterm – 14	1	1	1	0	1
Minterm – 15	1	1	1	1	1

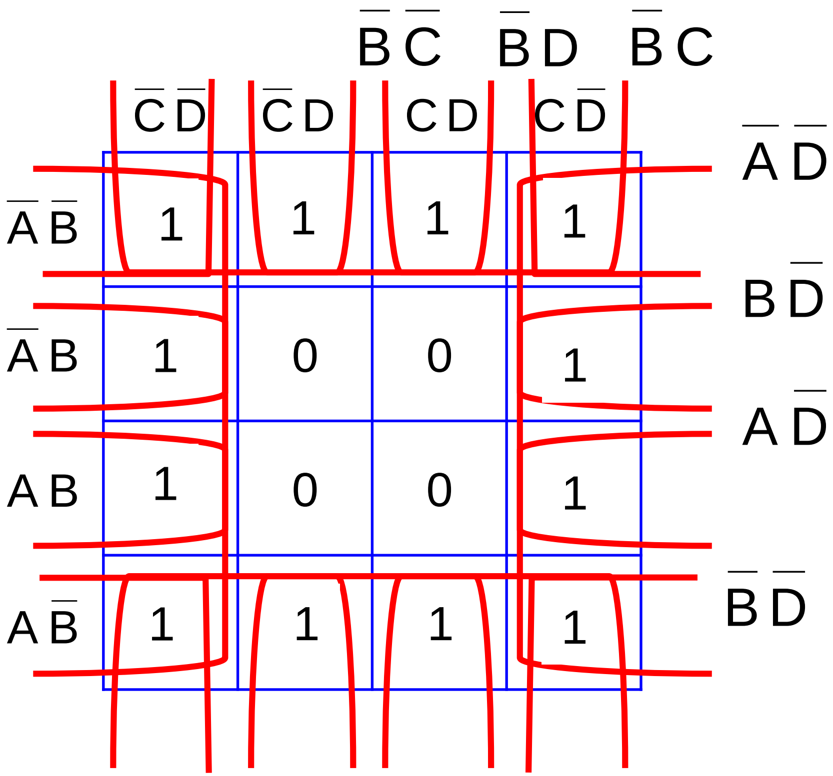
Only one variable changes for every column change

	$\overline{Y}\,\overline{Z}$	$\overline{Y}\,Z$	$Y\,Z$	$Y\,\overline{Z}$
$\overline{W}\,\overline{X}$	0_{0}	1_{1}	0_{3}	1_{2}
$\overline{W}\,X$	1_{4}	1_{5}	1_{7}	0_{6}
$W\,X$	1_{12}	0_{13}	1_{15}	1_{14}
$W\,\overline{X}$	0_{8}	0_{9}	0_{11}	1_{10}

Only one variable changes for every row change

Four Variable K-Map Groupings

Groups of One – 16 (not shown)
Groups of Two – 32 (not shown)
Groups of Four – 24 (seven shown)



Four Variable K-Map Groupings

Groups of Eight – 8 (two shown)

$\overline{B}$

$\overline{D}$

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

Four Variable K-Map Groupings

Group of Sixteen – 1

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	1	1	1
$\bar{A}B$	1	1	1	1
AB	1	1	1	1
$A\bar{B}$	1	1	1	1

1

Example #3: 4 Variable K-Map

Example:

After labeling and transferring the truth table data into the K-Map, write the simplified sum-of-products (SOP) logic expression for the logic function F_3.

R	S	T	U	F_3
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	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

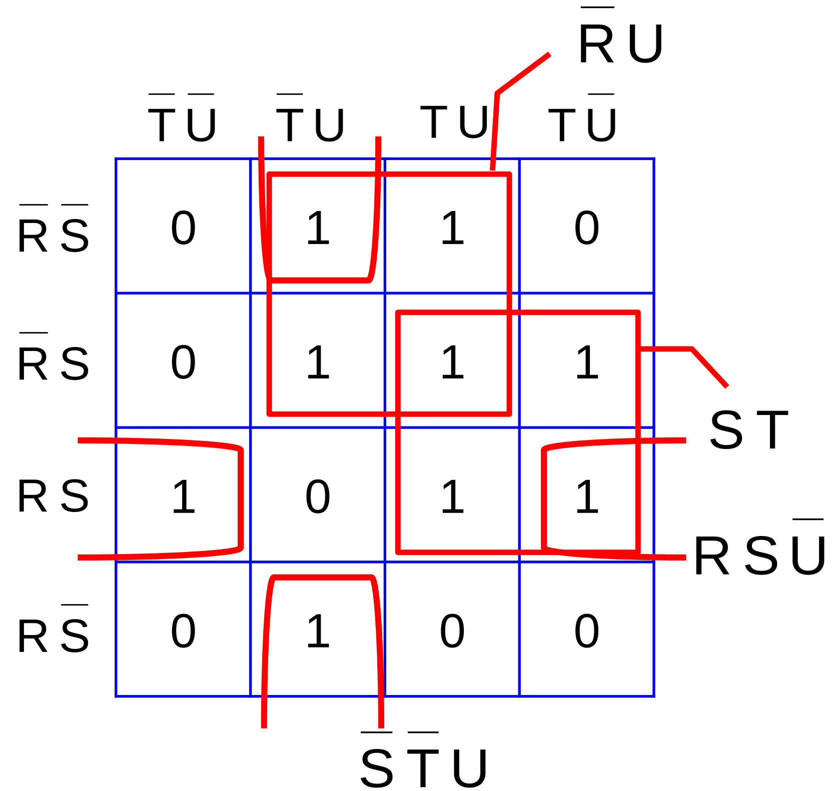
Example #3 : 4 Variable K-Map

Example:

After labeling and transferring the truth-table data into the K-Map, write the simplified sum-of-products (SOP) logic expression for the logic function F_3.

Solution:

R	S	T	U	F_3
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	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

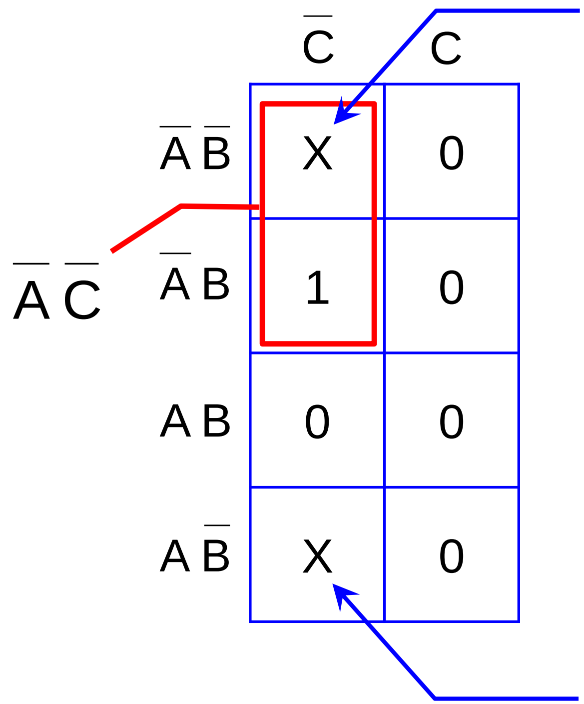


$$F_3 = RS\bar{U} + \bar{S}\bar{T}U + \bar{R}U + ST$$

Don't Care Conditions

- A *don't care* condition, marked by (X) in the truth table, indicates a condition where the design doesn't care if the output is a (0) or a (1).
- A *don't care* condition can be treated as a (0) or a (1) in a K-Map.
- Treating a *don't care* as a (0) means that you do not need to group it.
- Treating a *don't care* as a (1) allows you to make a grouping larger, resulting in a simpler term in the SOP equation.

Some You Group, Some You Don’t



This *don’t care* condition was treated as a (1). This allowed the grouping of a single one to become a grouping of two, resulting in a simpler term.

There was no advantage in treating this *don’t care* condition as a (1), thus it was treated as a (0) and not grouped.

Example #4: *Don't Care* Conditions

Example:

After labeling and transferring the truth table data into the K-Map, write the simplified sum-of-products (SOP) logic expression for the logic function F_4. Be sure to take advantage of the *don't care* conditions.

R	S	T	U	F_4
0	0	0	0	X
0	0	0	1	0
0	0	1	0	1
0	0	1	1	X
0	1	0	0	0
0	1	0	1	X
0	1	1	0	X
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	X
1	1	0	0	X
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Example #4: *Don't Care* Conditions

Example:

After labeling and transferring the truth table data into the K-Map, write the simplified sum-of-products (SOP) logic expression for the logic function F_4. Be sure to take advantage of the *don't care* conditions.

Solution:

R	S	T	U	F_4
0	0	0	0	X
0	0	0	1	0
0	0	1	0	1
0	0	1	1	X
0	1	0	0	0
0	1	0	1	X
0	1	1	0	X
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	X
1	1	0	0	X
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

	$\overline{T}\,\overline{U}$	$\overline{T}\,U$	$T\,U$	$T\,\overline{U}$
$\overline{R}\,\overline{S}$	X	0	X	1
$\overline{R}\,S$	0	X	1	X
$R\,S$	X	0	0	0
$R\,\overline{S}$	1	1	X	1

$\overline{R}\,T$

$R\,\overline{S}$

$$F_4 = \overline{R}\,T + R\,\overline{S}$$