Dr.SNS RAJALAKSHMI COLLEGE OF ARTS AND SCIENCE

(AUTONOMOUS)
COIMBATORE-2241049
Accredited by NAAC(Cycle III) with "A+" Grade
Recognised by UGC, Approved by AICTE, New Delhi and
Affiliated to Bharathiar University, Coimbatore.

DEPARTMENT OF COMPUTER SCIENCE

Computer System Architecture

I YEAR - I SEM

Unit II – Digital Logic Circuit

Product of Sums and Sum of Products

The *Product of Sum*(PoS) expression comes from the fact that two or more Sum (OR) are product (AND) together. Ex: (x+y+z)(x'+y+z')(x+y'+z)

The *Sum of Product* (SOP) expression comes from the fact that two or more products (AND) are summed (OR) together. Ex: (xyz) +(x'y'z')+(x'y'z)

Min Terms	Max Terms
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	x+ y'+z
x y z'	x +y+ z'
x y z	x +y+ z

Karnaugh Map

The **K-map** is a systematic way of simplifying Boolean expressions. With the help of the K-map method, we can find the simplest POS and SOP expression, which is known as the minimum expression.

2 Variable K-map

Y \ Z	0	1
0	m_0	m_1
1	m_2	m_3

or

YZ	00	01	11	10
	m_0	m_1	m_3	m_2

3 Variable K-map

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

4 Variable K-map

AB \ CD	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}

Karnaugh Map

Pair

A pair can be formed by grouping two horizontal or two vertical '1'.
A pair of '1' reduces 1 variable.

Quad

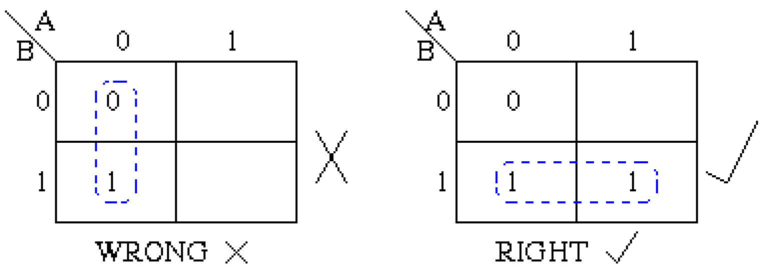
A quad is formed with four adjacent 1's either horizontally, vertically or two 1's horizontal and two 1's vertically adjacent.
A quad reduces two variable.

Octet

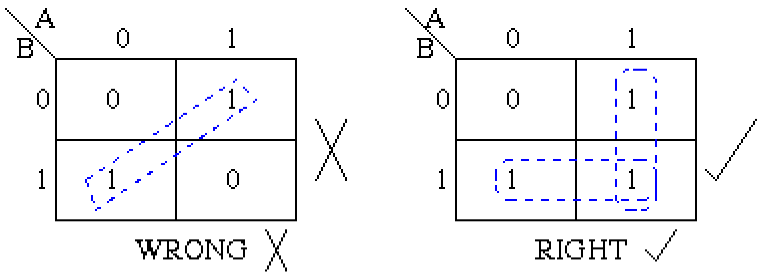
An octet is a group of eight adjacent 1's.
An octet reduces three variable from a Boolean equation.

Karnaugh Maps - Rules of Simplification

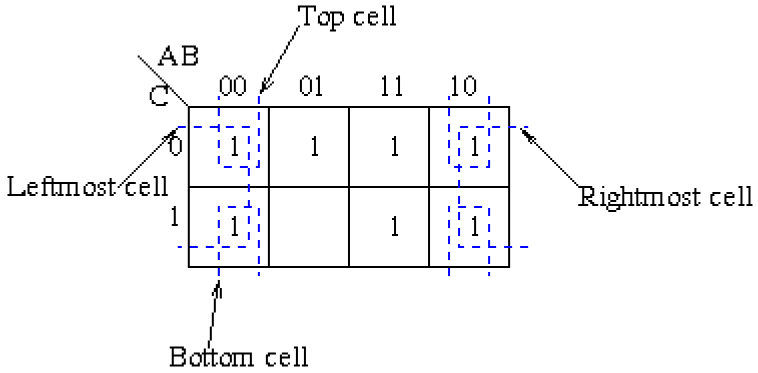
Groups may not include any cell containing a zero



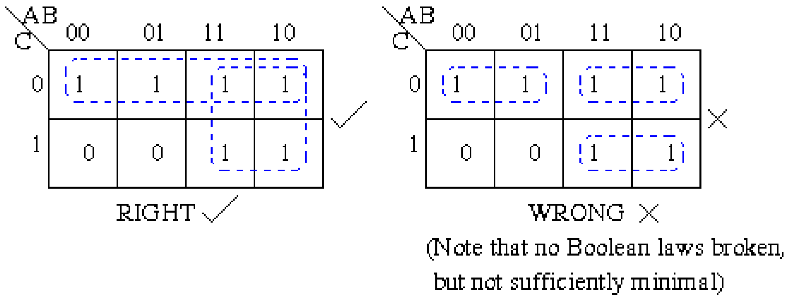
Groups may be horizontal or vertical, but not diagonal.



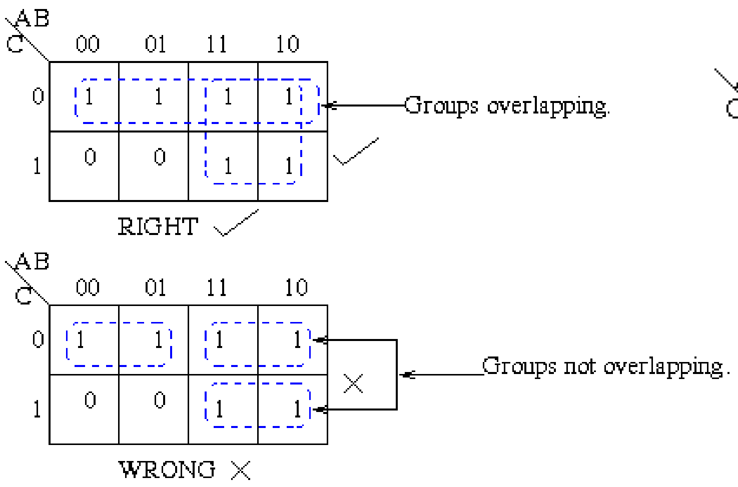
Groups may wrap around the table.



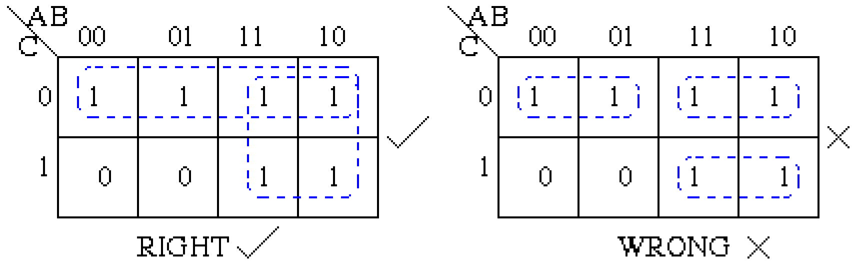
Each group should be as large as possible.



Groups may overlap

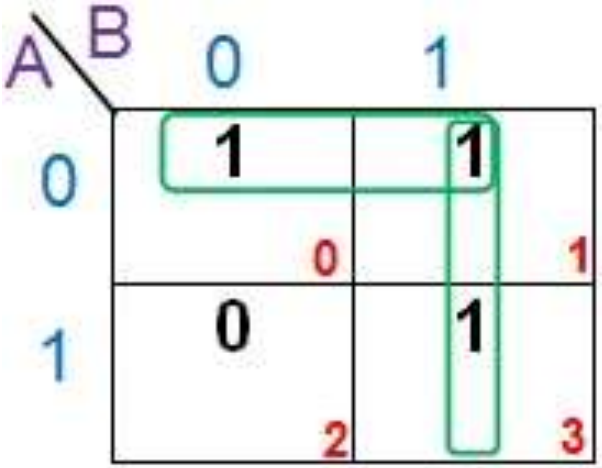


There should be as few groups as possible



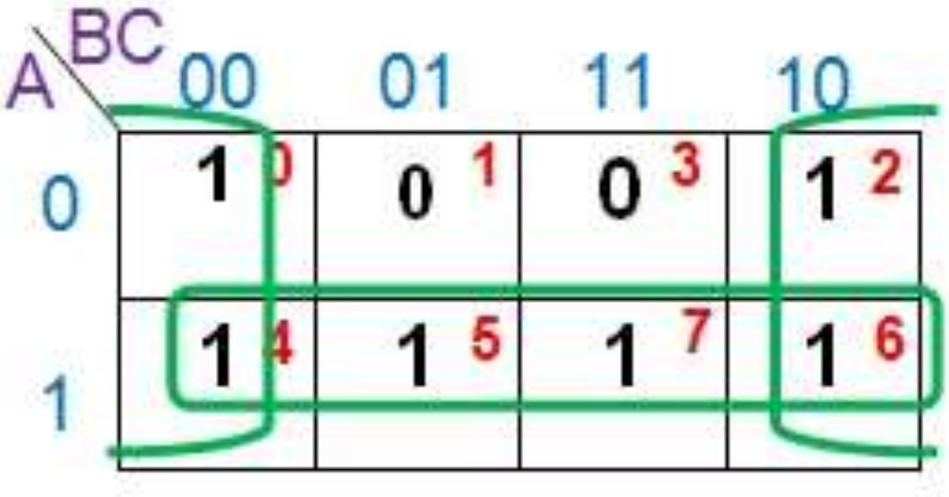
Boolean Expression Simplification using K-Map

$Example\ 1 : Y = \overline{A}\,\overline{B} + \overline{A}B + AB$



$Simplified\ Expression : Y = \overline{A} + B$

$Example\ 2 : Y = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}B\overline{C} + A\overline{B}\,\overline{C} + A\overline{B}C + AB\overline{C} + ABC$



$Simplified\ Expression : Y = A + \overline{C}$

Boolean Expression Simplification using K-Map

$$Example\ 3: Y = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}C\,\overline{D} + \overline{A}BC\,\overline{D} + \overline{A}BCD + A\overline{B}\,\overline{C}\,\overline{D} + A\overline{B}C\overline{D} + ABC\overline{D} + ABCD$$

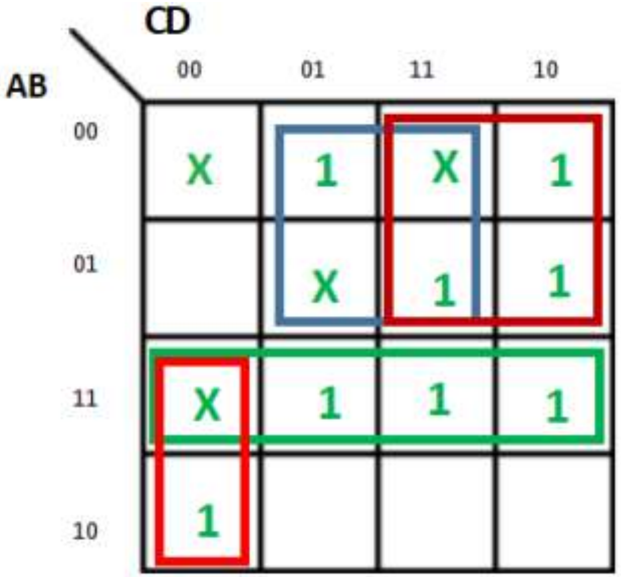
AB \ CD	00	01	11	10
00	1_{0}	0_{1}	0_{3}	1_{2}
01	0_{4}	1_{5}	1_{7}	0_{6}
11	0_{12}	1_{13}	1_{15}	0_{14}
10	1_{8}	0_{9}	0_{11}	1_{10}

$$Simplified\ Expression: Y = BD + \overline{B}\,\overline{D}$$

K-Map with "Don't Care" conditions

The "Don't Care" conditions allow us to replace the empty cell of a K-map to form a grouping of the variables which is larger than that of forming groups without don't care. While forming groups of cells, we can consider a "Don't Care" cell as 1 or 0 or we can also ignore that cell. Therefore, the "Don't Care" condition can help us to form a larger group of cells.

Example: F(A, B, C, D) = m(1, 2, 6, 7, 8, 13, 14, 15) + d(0, 3, 5, 12)



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

Thank You