Karnaugh Maps

Simplifying Boolean expressions systematically

Simplifying Boolean Expressions

- We've seen why it's advantageous to simplify Boolean expressions
- But doing that manually (using the Boolean laws) is not a very systematic process
- There are **algorithms** that are guaranteed to produce the simplest possible expression
- We will look at a particular one here

Karnaugh Maps (K-Maps)

- Systematic manual method for simplification

- Best for at most 6 variables
- Will produce the simplest Sums-Of-Products
- Similar to a truth table – maps all combinations

Karnaugh Maps (K-Maps)

3-Variable K-Map:

XY \ Z	0	1
00	0	0
01	1	1
11	1	1
10	1	0

Note: 11 before 10!

transfer

X	Y	Z	F(X,Y,Z)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

How do K-Maps help us simplify?

A common form of simplification in Boolean algebra is

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

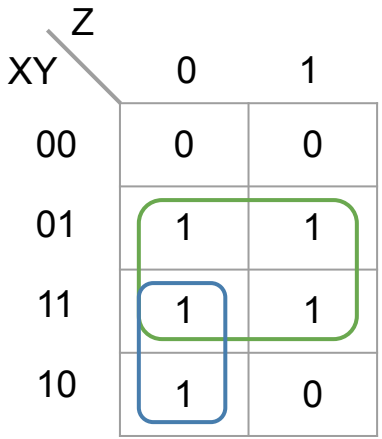
Also works for 3 variables:

$$A\overline{B}C + ABC = AC$$

Both are independent of B!

K-Maps are a visual way for finding independent terms.

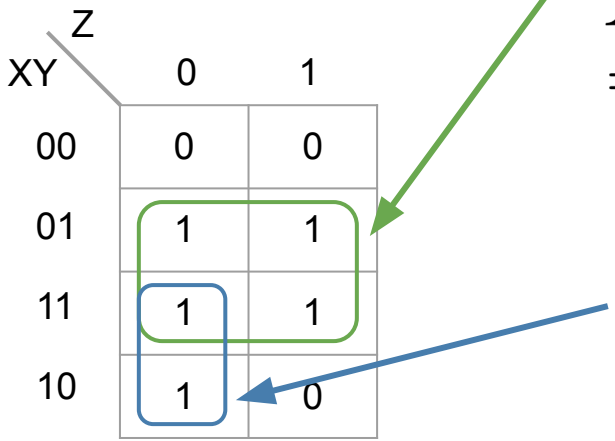
3-variable K-Map



Goal: Find **groups of ones**!

- Groups must be as large as possible
- Group size must be power of 2
- Contain only ones, no zeros
- All ones must be in at least one group
- Groups can overlap

3-variable K-Map



X and Z are irrelevant (group covers both 0 and 1 for these variables).
Y needs to be 1 for this group.

$$\overline{X}\,Y\,\overline{Z} + \overline{X}\,Y\,Z + X\,Y\,\overline{Z} + X\,Y\,Z = Y$$

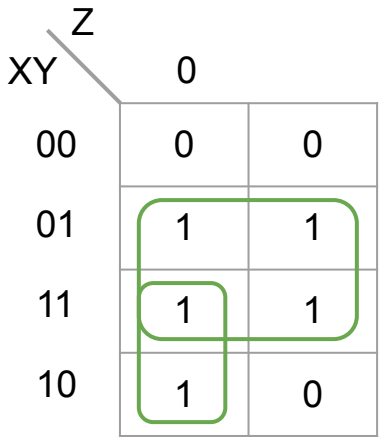
Y is irrelevant (group covers both 0 and 1 for Y).
X needs to be 1, Z needs to be 0 for this group.

$$X\,Y\,\overline{Z} + X\,\overline{Y}\,\overline{Z} = X\,\overline{Z}$$

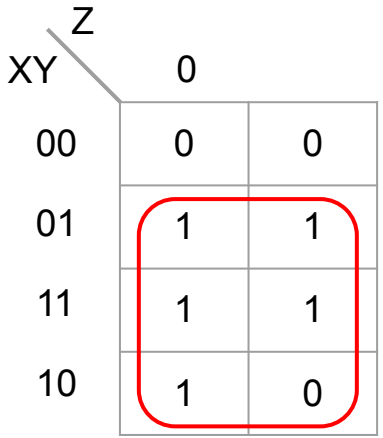
Result: $F(X, Y, Z) = X\overline{Z} + Y$

K-map rules

- Groups can’t contain 0



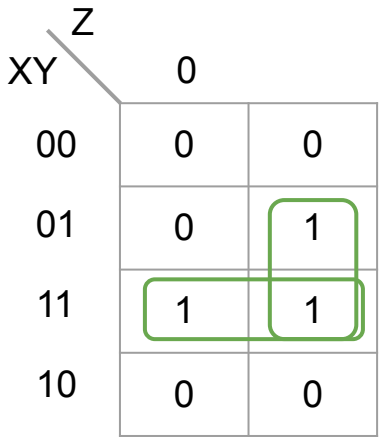
Correct



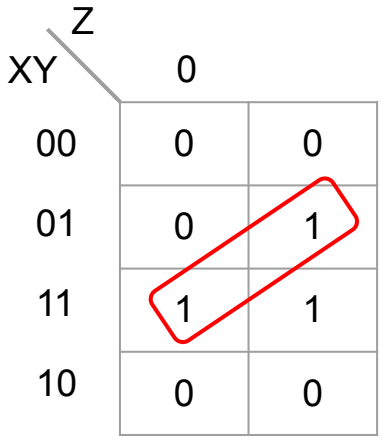
Wrong

K-map rules

- Groups must be rectangular (can't be e.g. diagonal)



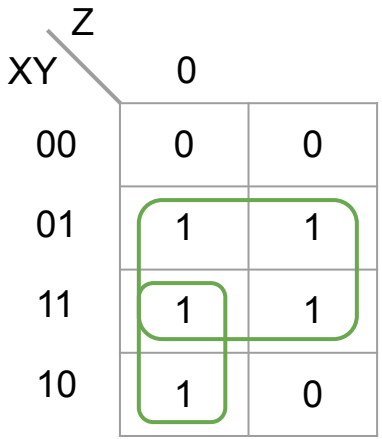
Correct



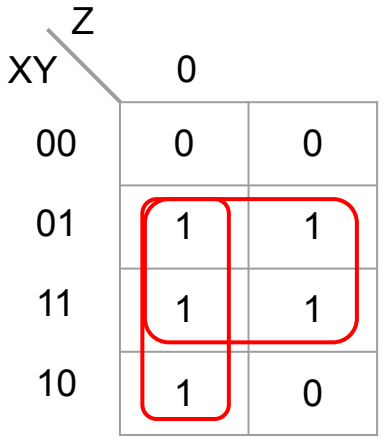
Wrong

K-map rules

- Group size is power of 2



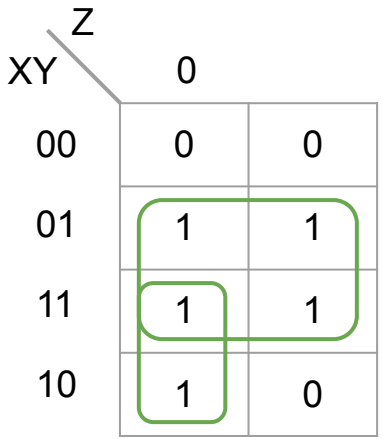
Correct



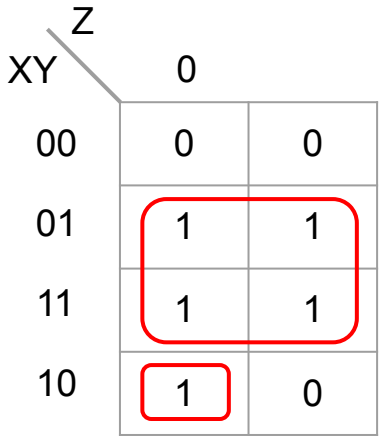
Wrong

K-map rules

- Each group is as large as possible
- Groups may overlap



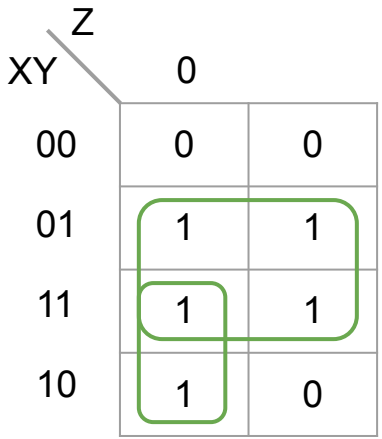
Correct



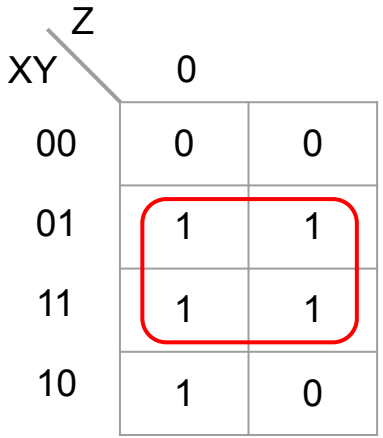
Wrong

K-map rules

- Each 1 must be in at least one group



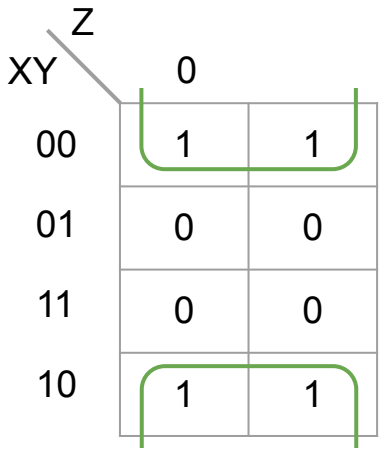
Correct



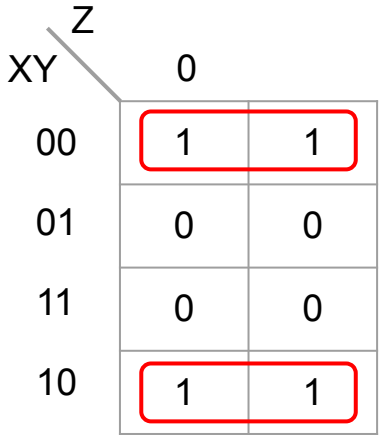
Wrong

K-map rules

- Groups may wrap around



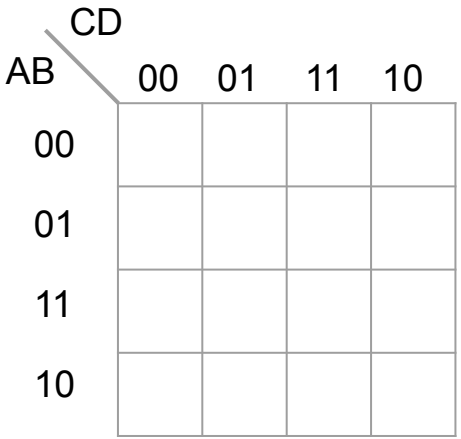
Correct

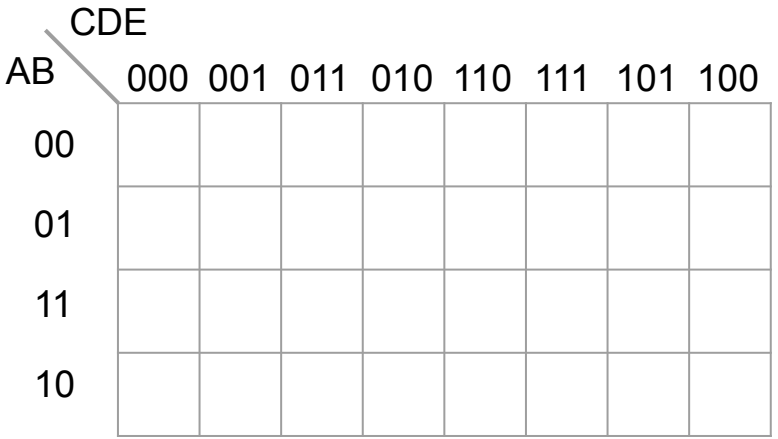


Wrong

K-Maps with more variables

Same idea, but more variables on the axes:





Summary

- K-Maps are a visual method for simplifying Boolean functions
- They produce *minimal* sum-of-product representations
- Work well for small functions (up to 6 inputs)
- Automatic approaches exist for optimising large functions

EOF