

BOOLEAN ALGEBRA EXAMPLE 1

Simplification from a truth table

Inputs			Output
A	B	C	D
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

From this truth table we can write a Boolean expression.

Step 1: Look at the values of inputs when the **output is 1**

A '0' input means NOT x and an input of '1' means x where x is the letter from the input.

SO

$\neg A \wedge B \wedge \neg C$	$\neg AB\neg C$	Notice that we can either use the signs \vee (or) or $+$ interchangeably. And multiplying is the same as \wedge (and)
$\neg A \wedge B \wedge C$	$\neg ABC$	
$A \wedge B \wedge \neg C$	$AB\neg C$	
$A \wedge B \wedge C$	ABC	

Step 2: So the sum is $\neg AB\neg C + \neg ABC + AB\neg C + ABC$ (now simplify with Boolean algebra techniques) or we can write this as $(\neg A \wedge B \wedge \neg C) \vee (\neg A \wedge B \wedge C) \vee (A \wedge B \wedge \neg C) \vee (A \wedge B \wedge C)$
 Factor out $\neg AB$ so $\neg AB(\neg C + C)$ + Factor out AB so $AB(\neg C + C)$

Since anything that is not itself or itself is always 1 it eliminates: $\neg AB + AB$

Factor out B so $B(\neg A + A)$

Since anything that is not itself or itself is always 1 it eliminates: B

Hence the entire truth table is equal to the expression: **B**

KARNAUGH MAP EXAMPLE 1

A Karnaugh map is similar to a truth table in the fact it is a box input for every possible input. However, the karnaugh map the column headings only change by 1 bit.

e.g. for the same logic algorithm:

Notice that the column heading goes 00, 01, 11, 10. This pattern is used to ensure that we are only changing 1 bit when looking across a period. So for (c1,r1) A = 0 B = 0 C = 0 but for (C2, r1) A = 0 still & B = 0 still but C = 1 (only 1 bit changes) etc. Similarly, if we move down a row for a particular column, it is only A that changes.

The first variable (A) goes in the left row heading, and both of the inputs are combined in an AND (\wedge) expressions, BC.

A \ BC	00	01	11	10
0				
1				

We place the outputs (from previous truth table) in the corresponding fields.

		B C			
	A	00	01	11	10
0		0	0	1	1
1		0	0	1	1

I have colour coded them so you can see how they match up.

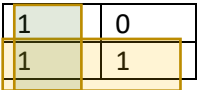

Using a karnaugh map to determine the simplest expression

Step 1: Build Karnaugh map

Step 2: Draw a square or a rectangle to combine outputs of 1 (these can be called “minterms”)

Here are the rules for doing this.....

- 2) Combine minterms (where minterm = 1) into groups
 - a. Group size is a power of 2 (i.e., 1, 2, 4, 8, 16...)
 - b. Group shape is square or rectangle
 - c. Make groups as large as possible
 - d. Groups can overlap
 - e. Groups can stretch around boundaries
 - f. All boxes where minterm = 1 need to be in a group

Rule	Explanation
A	e.g. We can't have a group with 3 minterms
B	Must look like a square or rectangle (encompasses only minterms)
C	Make the groups as large as they can (without crossing other 0's)
D	This means multiple groups may form e.g. an L shape 
E	This means that a column at the side is really adjacent to the column on the other side. E.g. just remember to move the correct column headings 
F	Every box with a minterm (1) must belong to a group.

So in our example

		B C			
	A	00	01	11	10
0		0	0	1	1
1		0	0	1	1

Step 3: Determine the common value of the group. This means what variable has the same value in all of the boxes in the group. In our case B is always = 1 but A and C vary in the group. So the value is B = 1

In our example, there is just one group but it is possible to have multiple groups and so multiple of these expressions.

Step 4: Determine an expression for each group: we look for the variable and if it equals 1 then the expression is just that variable but if the variable is equal to 0 then it is NOT that variable. Say if B was always equal to zero it would be NOT B.

In our case $B = 1$ so expression is B

Step 5: Do the same for all the other groups

Step 6: Add all the expressions together and simplify.

KARNAUGH MAP EXAMPLE 2

From this truth table, determine the simplest possible Karnaugh map and write an expression for the logic operation.

A	B	C	OUT
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

ANSWER

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

Notice that we cannot form a box with 3 minterms (1's in) as we can only have powers of 2 (1,2,4,8...) This means to encompass all whiel using a minimum number of boxes we need 3 boxes.

Group value	Expression (1 = term and 0 = not term)
Group gold: Common value is $B = 1$ AND $C = 1$	$B \wedge C$ (can be written as BC)
Group red: common value is $A = 1$ AND $B = 1$	$A \wedge C$ (can be written as AC)
Green box: common value is $A = 1$ AND $C = 0$	$A \wedge \neg C$ (can be written as A¬C)

Summing the expressions (i.e. add means or)

So.... $(B \wedge C) \vee (A \wedge B) \vee (A \wedge \neg C)$

$B \wedge (C \vee A) \vee (A \wedge \neg C)$

This is already in its simplest form?

Follow the link to a good tutorial: <https://www.youtube.com/watch?v=-gQ6kN7qnmA>

Recapping some simple rules

$$(A \wedge B) \vee \neg C$$

Order of precedence: NOT, AND then OR

$$A \wedge \neg A = 0$$

$$A \vee \neg A = 1$$