Simplify a logic expression

… using K-Maps

Simplify an expression using K-Maps

Given,

$$X = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,B + \overline{B}\,C$$

Simplify the above expression, using K-Maps.

Set-up K-Map table

$$X = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}B + \overline{B}C$$

AB \ C	0	1
00		
01		
11		
10		

Set-up K-Map table

$$X = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}B + \overline{B}C$$

AB \ C	0	1
00	1	
01		
11		
10		

$\overline{A}\,\overline{B}\,\overline{C}$

Set-up K-Map table

$$X = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}B + \overline{B}C$$

AB \ C	0	1
00	1	
01	1	1
11		
10		

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

Set-up K-Map table

$$X = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,B + \overline{B}\,C$$

AB \ C	0	1
00	1	1
01	1	1
11		
10		1

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

Using Tables

A MORE SYSTEMATIC WAY

A more systematic way …

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

AB \ C	0	1
00	1	
01		
11		
10		

Second term ($\bar{A}B$)

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

AB \ C	0	1
00	1	
01		
11		
10		

A	B	C	A	B	C
0	1	X			
0	1	X			

Second term ($\overline{A}B$)

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

AB \ C	0	1
00	1	
01	1	1
11		
10		

A	B	C	A	B	C
0	1	X	0	1	1
0	1	X	0	1	0

Third term ($\bar{B}C$)

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

AB \ C	0	1
00	1	
01	1	1
11		
10		

A	B	C	A	B	C
X	0	1			
X	0	1			

Third term ($\overline{B}C$)

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

AB \ C	0	1
00	1	1
01	1	1
11		
10		1

A	B	C	A	B	C
X	0	1	0	0	1
X	0	1	1	0	1

Simplification

AB \ C	0	1
00	1	1
01	1	1
11		
10		1

Simplification

AB \ C	0	1
00	1	1
01	1	1
11		
10		1

X = ?

Simplification

AB \ C	0	1
00	1	1
01	1	1
11		
10		1

$$X = \overline{A} + \overline{B}C$$

Another useful example...

$$X = AB + \overline{A}C + BC$$

Another useful example…

AB \ C	0	1
00		
01		
11		
10		

$$X = AB + \overline{A}C + BC$$

AB

AB \ C	0	1
00		
01		
11	1	1
10		

A	B	C	A	B	C
1	1	X	1	1	1
1	1	X	1	1	0

AB

$$X = AB + \overline{A}C + BC$$

A'C

AB \ C	0	1
00		1
01		1
11	1	1
10		

A	B	C	A	B	C
0	X	1	0	0	1
0	X	1	0	1	1

$\bar{A}C$

$$X = AB + \overline{A}C + BC$$

BC

AB \ C	0	1
00		1
01		1 1
11	1	1 1
10		

A	B	C	A	B	C
X	1	1	0	1	1
X	1	1	1	1	1

BC

$$X = AB + \overline{A}C + BC$$

Consensus theorem

AB \ C	0	1
00		1
01		1
11	1	1
10		

$$X = AB + \overline{A}C + BC$$

Consensus theorem

AB \ C	0	1
00		1
01		1
11	1	1
10		

…useful algebraic simplification theorem

$$X = AB + \overline{A}C + BC = AB + \overline{A}C$$

Consensus Theorem

- AB + A'C + BC
- AB + A'C + 1 BC
- AB + A'C + (A + A') BC
- AB + A'C + ABC + A'BC
- AB(1+C) + A'C(1 + B)
- AB + A'C

- Which is > AB+A'C+BC = AB + A'C

Using the Boolean Theorems

K-Map simplification technique

- Good only for small circuits
- Excellent academic method
- There are better computer-based techniques

Our work on the subject

WWW-based simplification method

Tomaszewski, S.P, I.U. Ilgaz and Antoniou, G.E. (2003). "WWW-Based Boolean function simplification", *International Journal of Applied Mathematics and Computer Science,* 13 (4), 577-583.

(To download the paper: http://www.researchgate.net/profile/George_Antoniou)

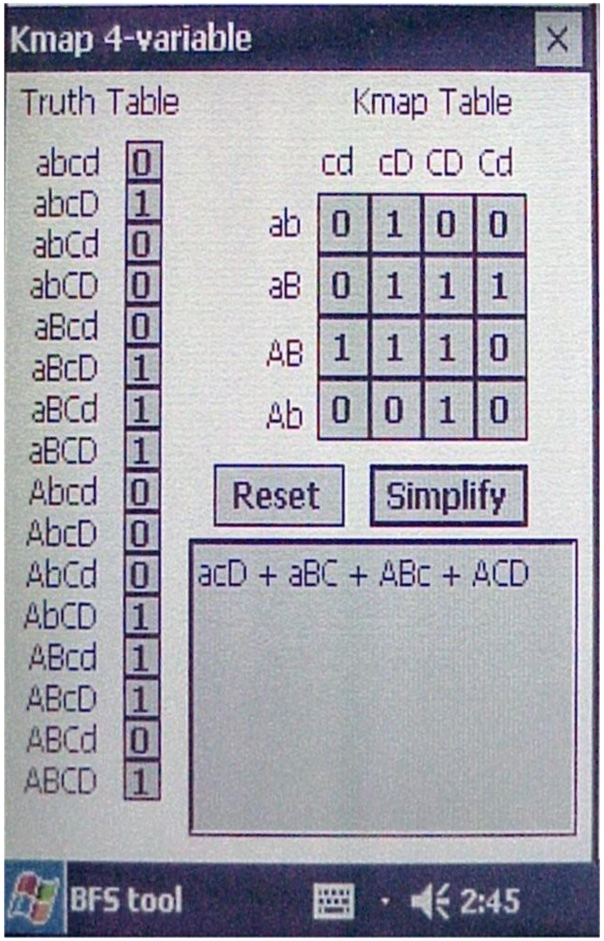
References:

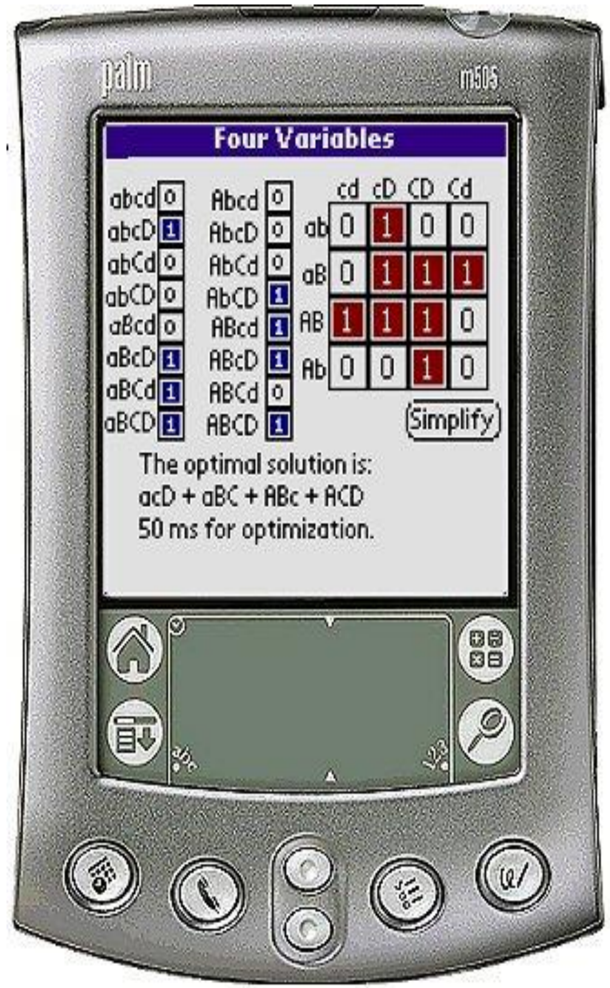
Quine W.V. (1952). The problem of simplifying truth tables, *American. Math. Monthly*, 59(8), 521-531.

McCluskey E.J. *(1956*). Minimization of Boolean functions, *Bell System Technical Journal*, 35(5), 1417-1444.

Sebastian Tomaszewski and I.U. Ilgaz are MSU/CS Alumni (2002)

PDA-based simplification





Journal Publications

Ledion Bitincka, George E. Antoniou, (2004), "PDA-Based Boolean Function Simplification: A Useful Educational Tool", ***International Journal INFORMATICA, Vol. 15, No. 3, pp. 329-336.***

Ledion Bitincka, George E. Antoniou, (2005), "Pocket-PC Boolean Function Simplification", ***International Journal in Electrical Engineering, Vol. 56, No. 7-8, pp. 1-4.***

(To download the papers: http://www.researchgate.net/profile/George_Antoniou)

Ledion Bitincka is MSU/CS Alumni (2003)

BFSTool 1.0 (Softpedia)

Incompletely **S**pecified **F**unctions

ISF

ISF

- In some design problems a number of the inputs never occur, so there is no specified output. Such an output is denoted by (X) and is called Don’t Care Condition.

ISF-Example

The output (Z) of a three-input (A,B,C) digital circuit is:

- 0 if The input is: $\leq 2_{10}$
- 1 if The input is: $\geq 5_{10}$
- × otherwise

Set-up the Truth Table

Truth table

The output (Z) of a three-input (A,B,C) digital circuit is:

- 0 if The input is: <= 2_{10}
- 1 if The input is: >= 5_{10}
- x otherwise

	A	B	C	Z (output)
0	0	0	0	
1	0	0	1	
2	0	1	0	
3	0	1	1	
4	1	0	0	
5	1	0	1	
6	1	1	0	
7	1	1	1	

In some design problems a number of the inputs never occur, so there is no specified output. Such an output is denoted by (X) and is called don't care condition.

Truth table

The output (Z) of a three-input (A,B,C) digital circuit is:

- 0 if The input is: <= 2_{10}
- 1 if The input is: >= 5_{10}
- x otherwise

output

	A	B	C	Z
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	x
4	1	0	0	x
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

In some design problems a number of the inputs never occur, so there is no specified output. Such an output is denoted by (X) and is called don't care condition.

Output and don't care equations

$Z =$

and

$X =$

Output and don't care equations

$$Z = A\overline{B}C + AB\overline{C} + ABC$$

and

$$X = \overline{A}BC + A\overline{B}\,\overline{C}$$

	A	B	C	Z
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	x
4	1	0	0	x
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Set-Up the K-Map

Set-up K-Map table - Z

AB \ C	0	1
00		
01		
11		
10		

	A	B	C	Z
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	x
4	1	0	0	x
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Set-up K-Map table - Z

AB \ C	0	1
00		
01		
11	1	1
10		1

	A	B	C	Z
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	x
4	1	0	0	x
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Set-up K-Map table

The x's can be either 1 or 0 as long as the final result has **absolutely-minimum** number of terms

AB \ C	0	1
00		
01		X
11	1	1
10	X	1

	A	B	C	Z
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	x
4	1	0	0	x
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Optimal values

Setting the top x=0 and the bottom x=1,…

AB \ C	0	1
00		
01		X = 0
11	1	1
10	X = 1	1

Z = ?

Result: Z = A

The final result has **absolutely-minimum (optimal)** number of terms.

AB \ C	0	1
00		
01		X = 0
11	1	1
10	X = 1	1

$$Z = A$$