K-Maps continue

SIMPLIFY A LOGIC EXPRESSION USING K-MAPS

Simplify an expression using K-maps

Given,

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

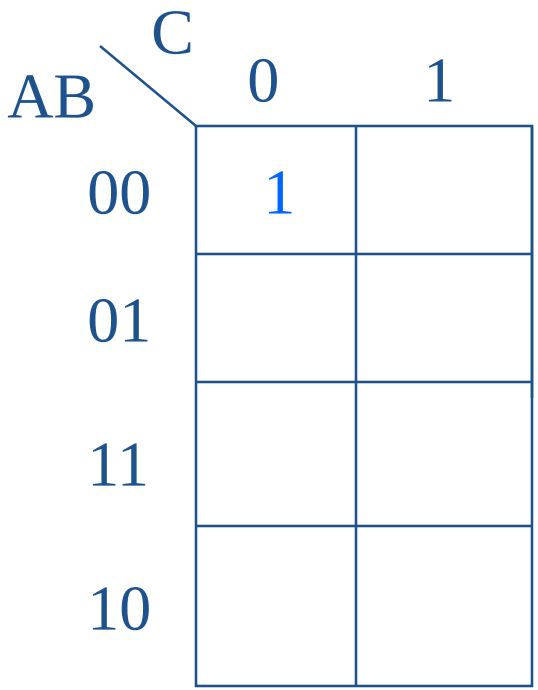
Simplify an expression using K-maps

Given,

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

Simplify the above expression, using K-maps.

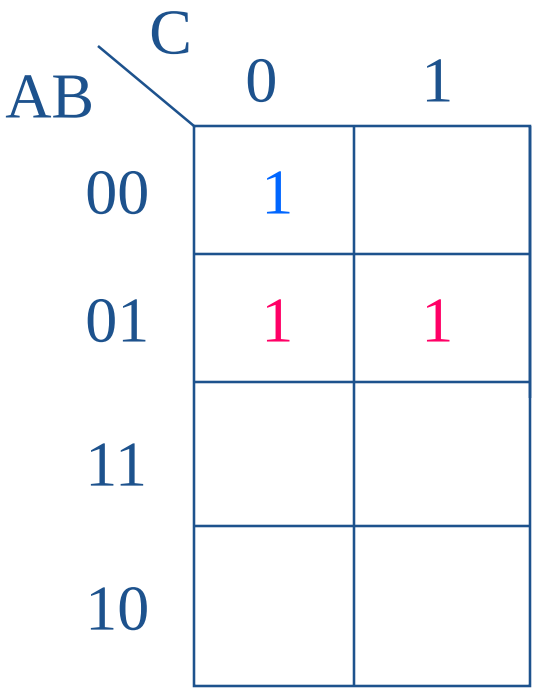
Set-up K-map table



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

Set-up K-map table

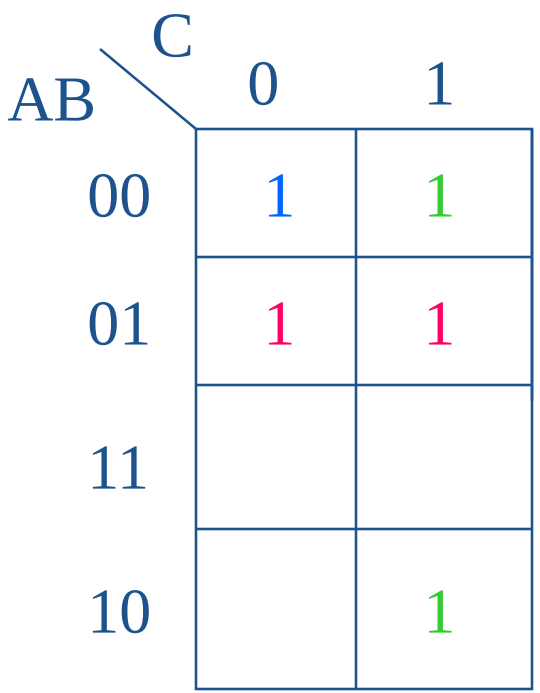
$$X = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}B + B\overline{C}$$



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

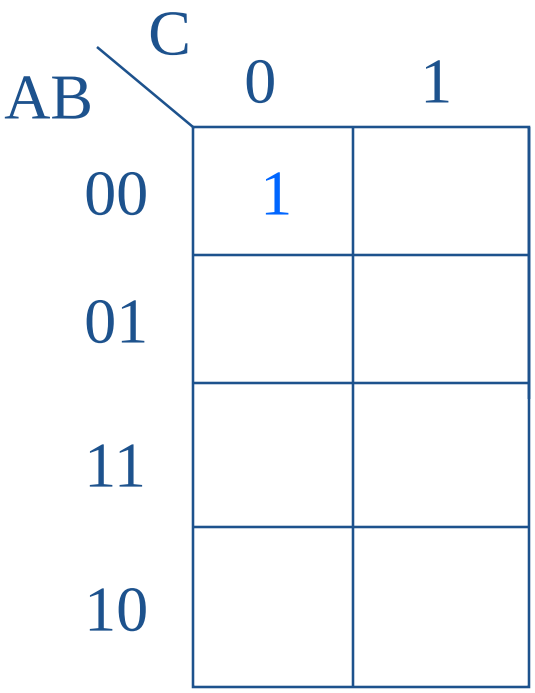
Set-up K-map table

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



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

A more systematic way …



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

A	B	C	A'B'C'	A'B	B'C
0	0	0	1		
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Set-up K-map table

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

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

A	B	C	A'B'C'	A'B	B'C
0	0	0	1		
0	0	1			
0	1	0		1	
0	1	1		1	
1	0	0			
1	0	1			
1	1	0			
1	1	1			

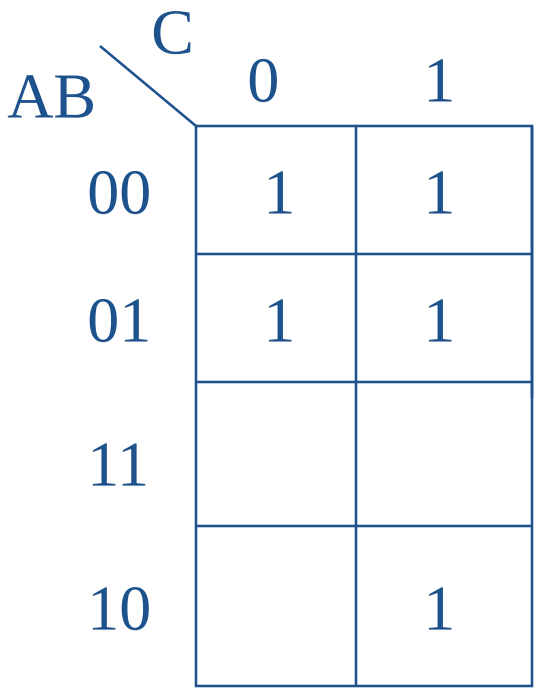
Set-up K-map table

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

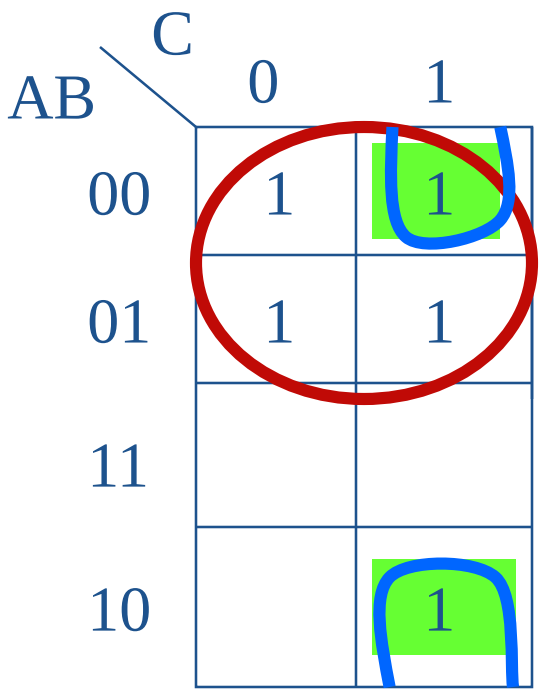
AB \ C	0	1
00	1	1
01	1	1
11		
10		1

A	B	C	A'B'C'	A'B	B'C
0	0	0	1		
0	0	1			1
0	1	0		1	
0	1	1		1	
1	0	0			
1	0	1			1
1	1	0			
1	1	1			

Simplification



Simplification

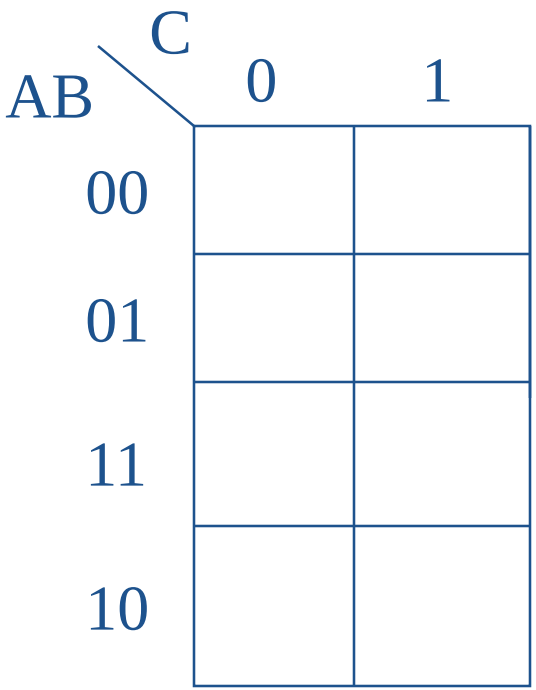


$$X = \overline{A} + \overline{B}C$$

Another useful example…

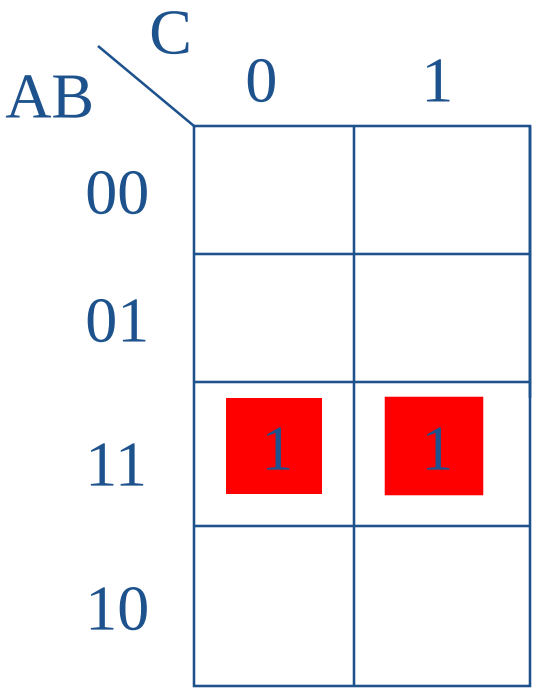
$$X = AB + \overline{A}C + BC$$

Another useful example…



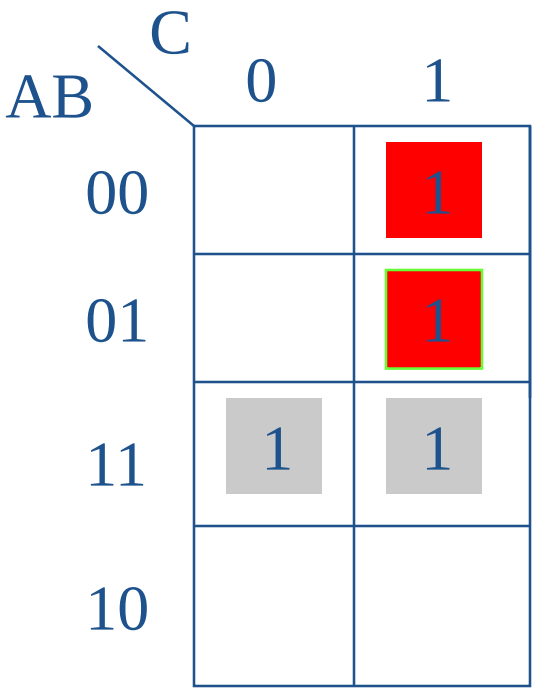
$$X = AB + \overline{A}C + BC$$

AB



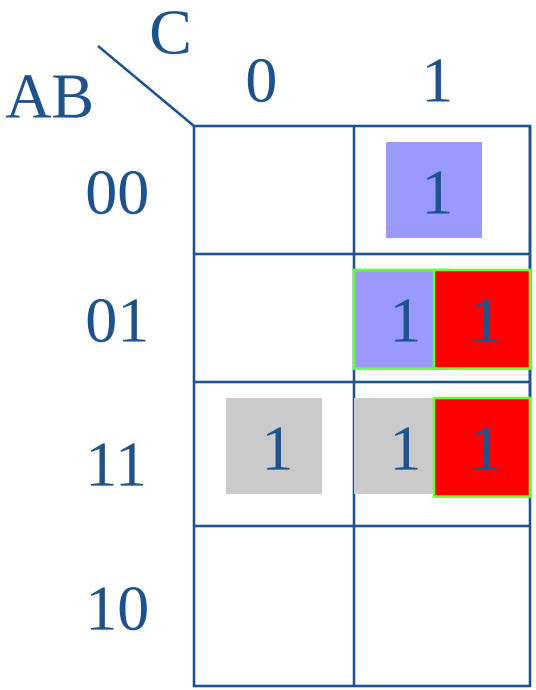
$$X = AB + \overline{A}C + BC$$

A'C



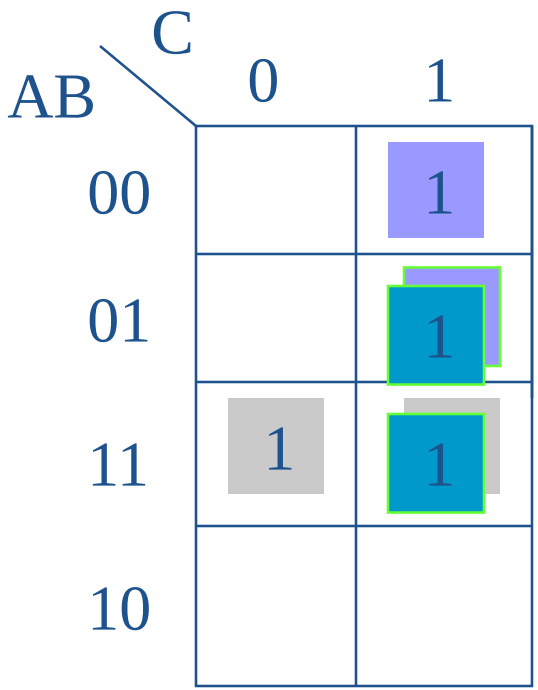
$$X = AB + \overline{A}C + BC$$

BC



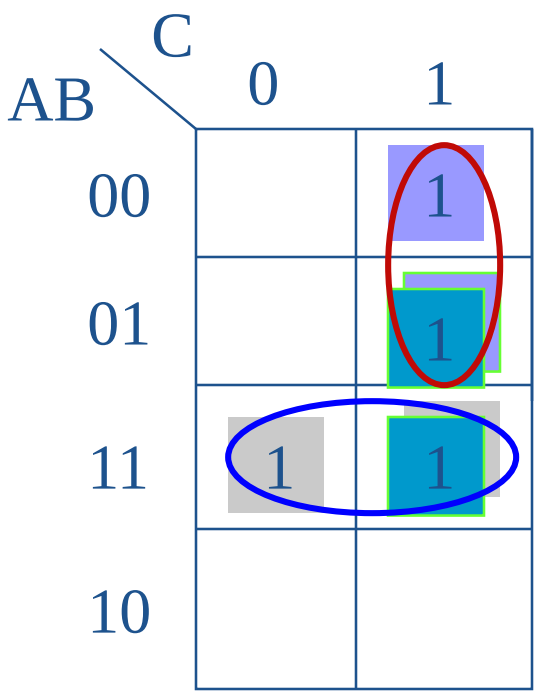
$$X = AB + \overline{A}C + BC$$

Consensus theorem



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

Consensus theorem



…useful algebraic simplification theorem

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

Using the Table

A	B	C	AB	A'C	BC
0	0	0			
0	0	1		1	
0	1	0			
0	1	1		1	1
1	0	0			
1	0	1			
1	1	0	1		
1	1	1	1		1

- How can we simplify the table?
- Why it is: $AB + AC'$
- (project or independent study = 3 credits research)

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 … $AB+A'C+BC = AB + A'C$

K-Map simplification technique

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

WWW-based simplification method

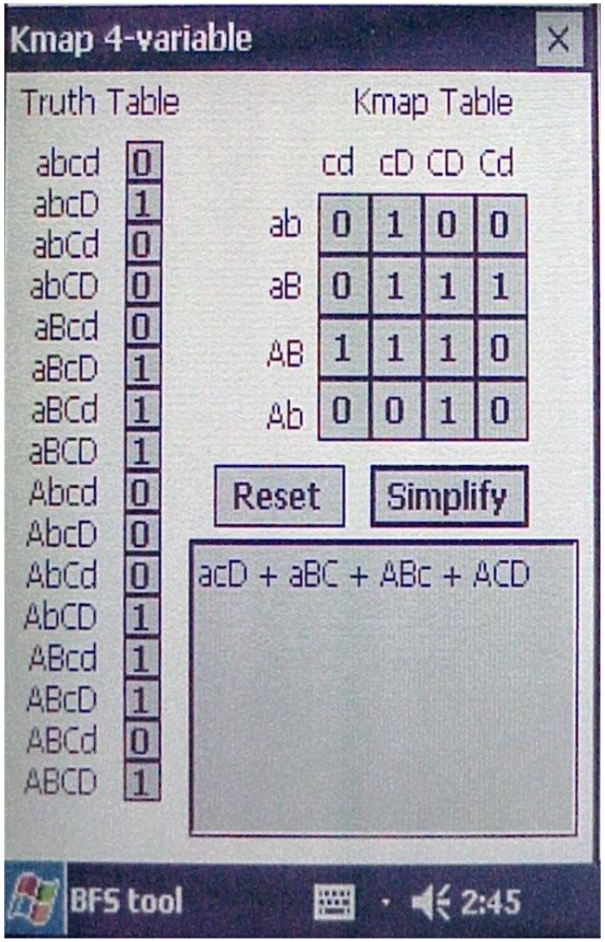
Tomaszewski, S.P, I.U. Ilgaz and Antoniou, G.E. (2003). WWW-Based Boolean function simplification, *International Journal of Applied Mathematics and Computer Science,* 13 (4), 577-583.

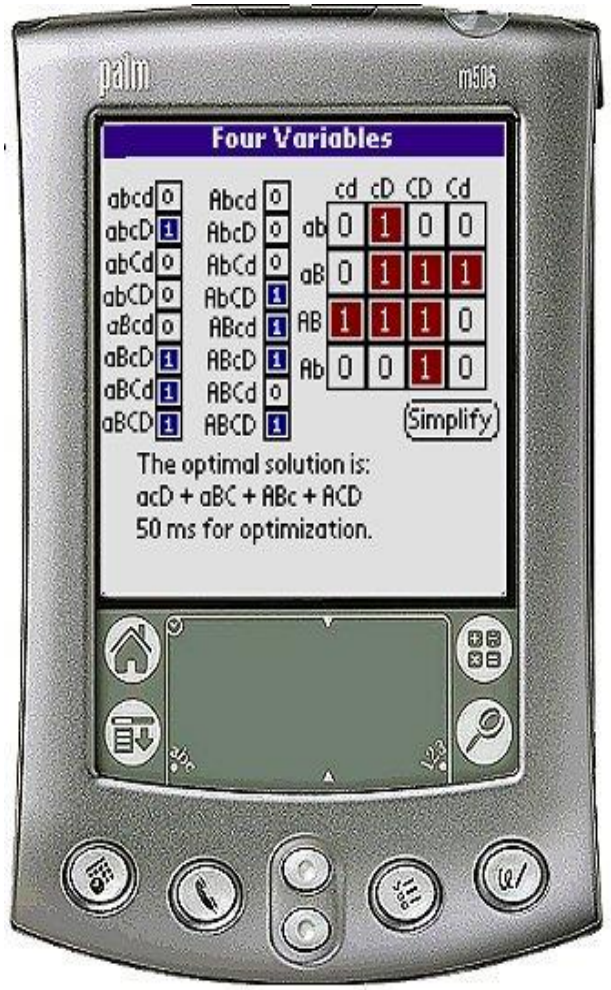
Quine W.V. (1952). The problem of simplifying truth tables, *American. Math. Monthly*, 59(8), 521-531.

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





PDA-based simplification

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.***

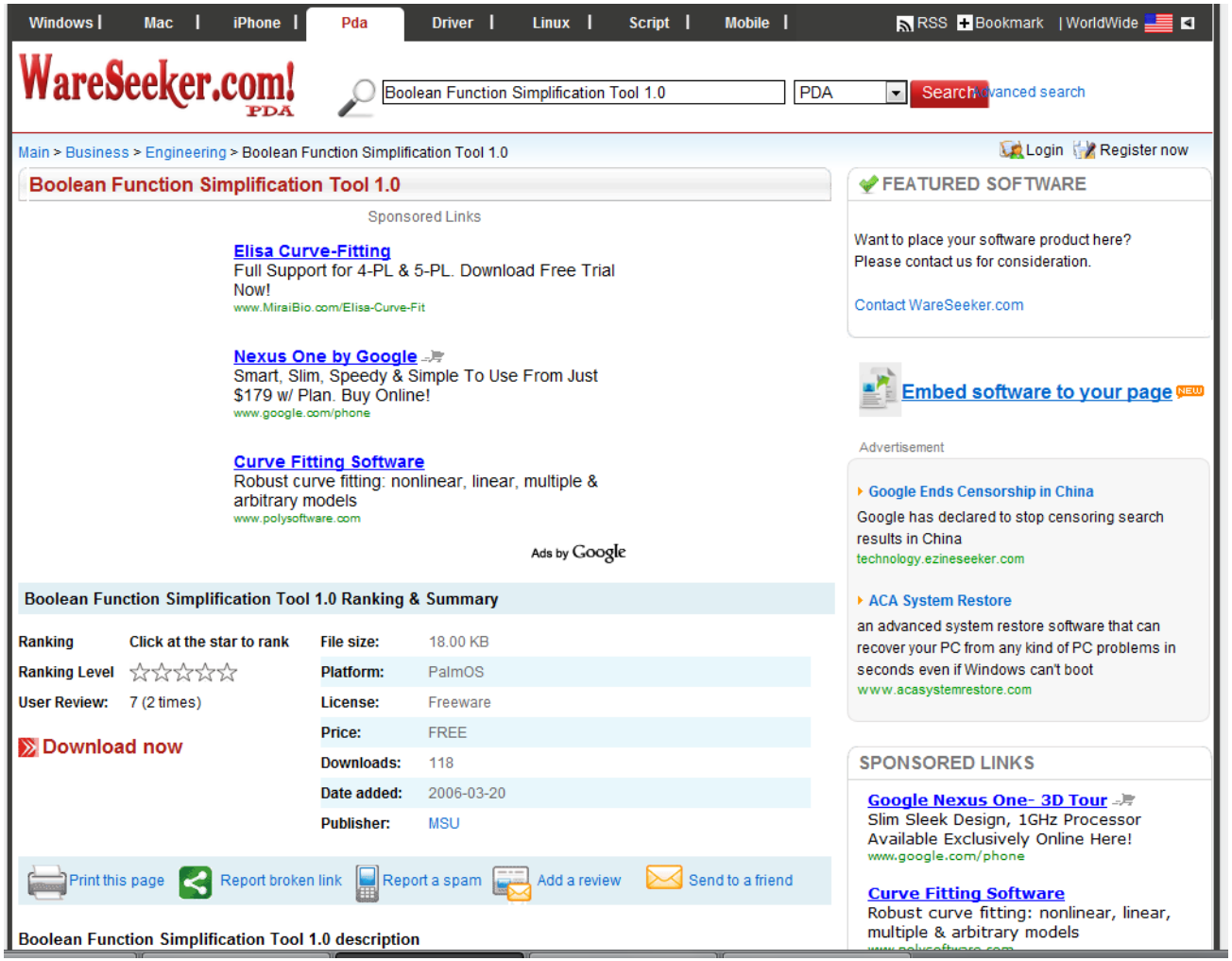
Ledion Bitincka, George E. Antoniou, (2005), Pocket-PC Boolean Function Simplification, ***International Journal in Electrical Engineering, Vol. 56, No. 7-8, pp. 1-4.***

Ledion Bitincka is MSU/CS Alumni (2003)

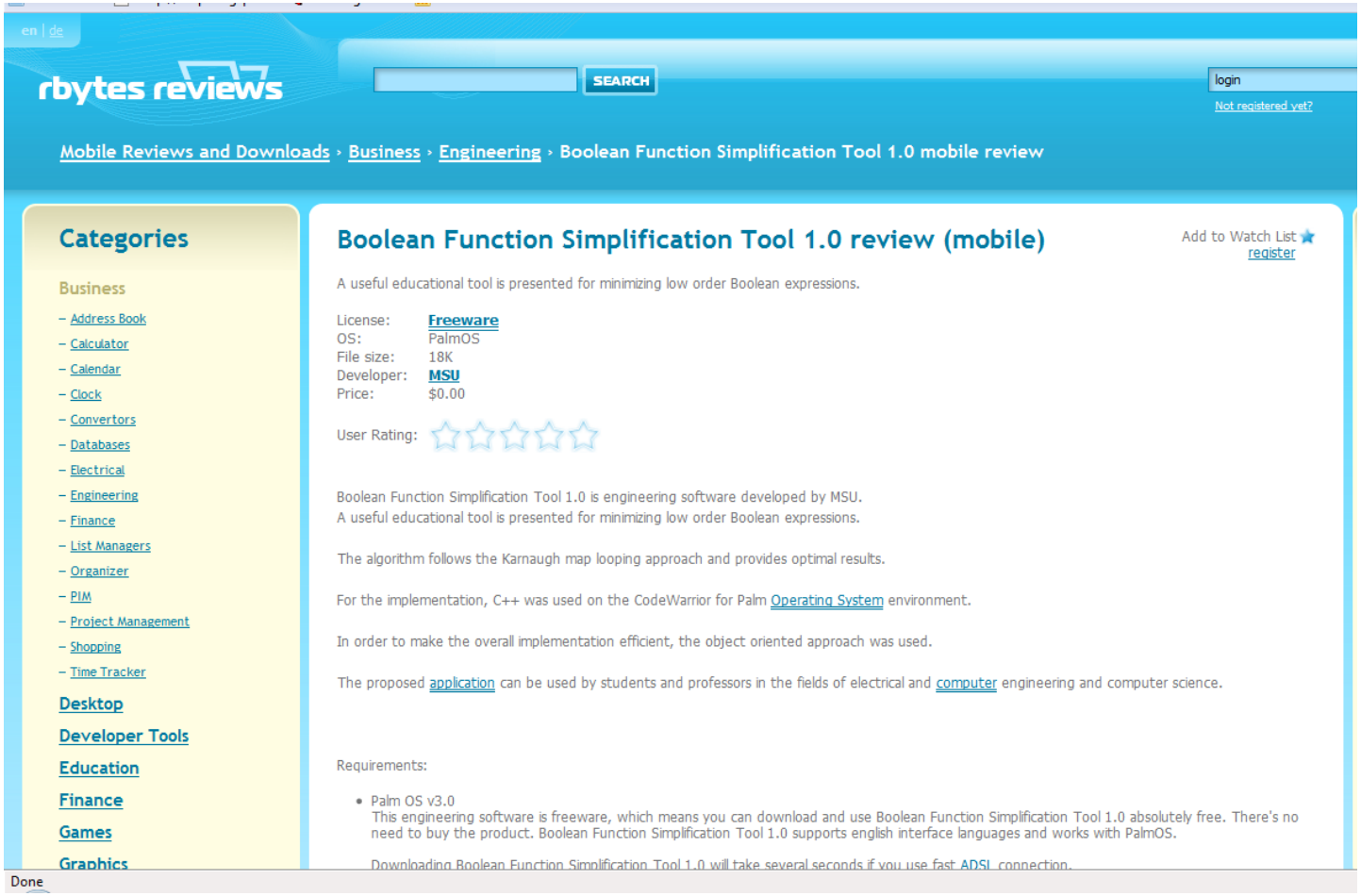
BFSTool 1.0 (Softpedia)

BFSTool 1.0 (Freeware Palm OS)

BFSTool 1.0 (WareSeeker)



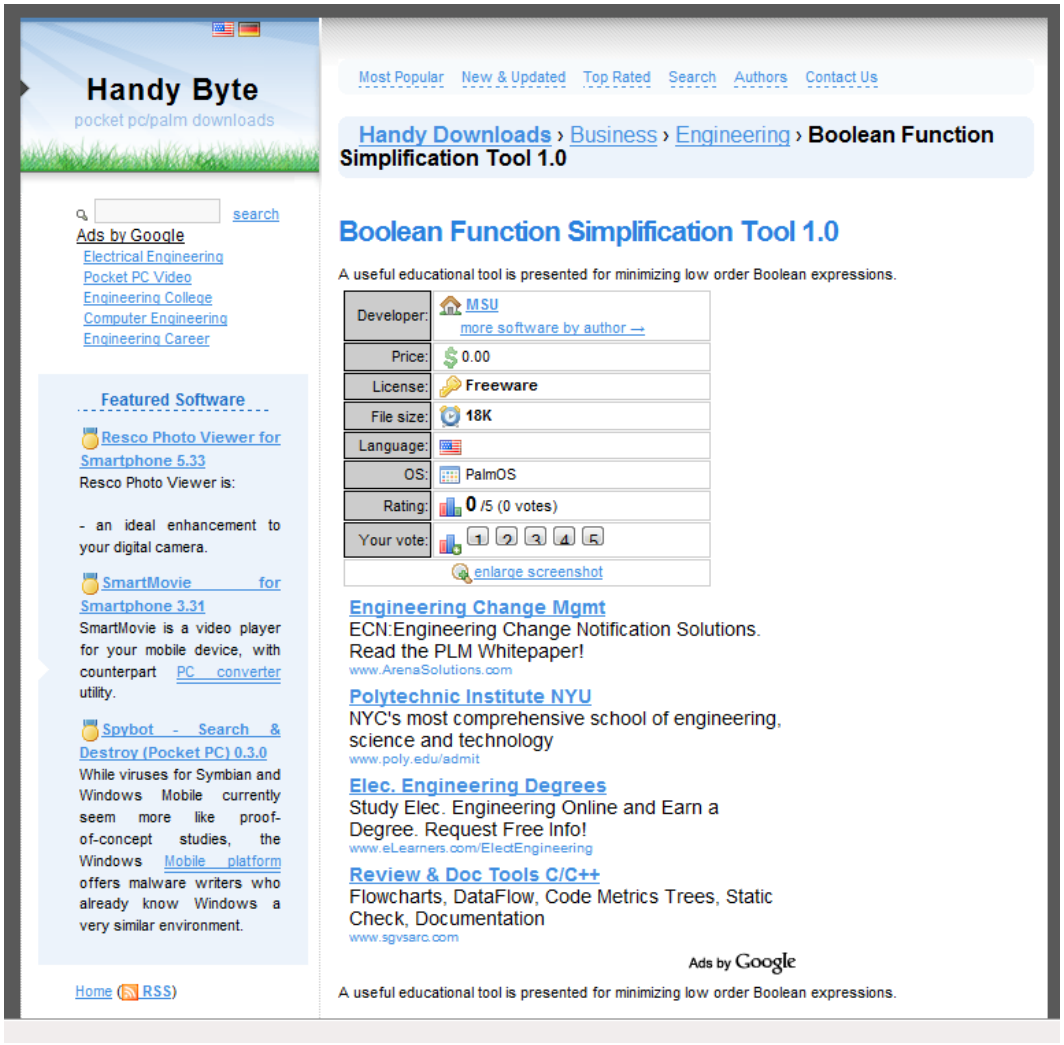
BFSTool 1.0 (rbytes reviews)



BFSTool 1.0 (Shareware)

BFSTool 1.0 (Mobile fun.net)

BFSTool 1.0 (Handy Byte)



Incompletely specified functions*

- 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

ISF-Example

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

- 0 if $F(A,B,C) \leq 2_{10}$
- 1 if $F(A,B,C) \geq 5_{10}$
- × otherwise

Set-up the Truth Table

Truth table

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

- 0 if F(A,B,C) <= 2_{10}
- 1 if F(A,B,C) >= 5_{10}
- x otherwise

	A	B	C	Z
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 F(A,B,C) digital circuit is:

- 0 if F(A,B,C) <= 2_{10}
- 1 if F(A,B,C) >= 5_{10}
- x otherwise

	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

Output and don't care equations

Z =

and

X =

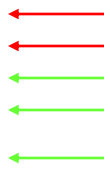
Output and don't care equations

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

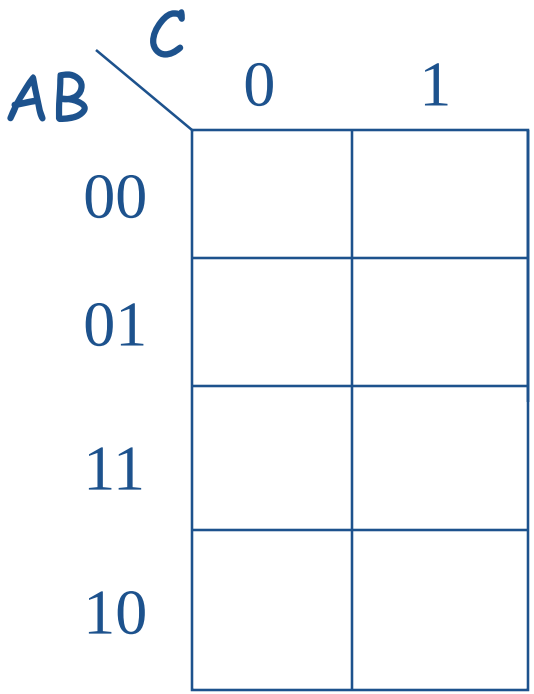
and

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

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

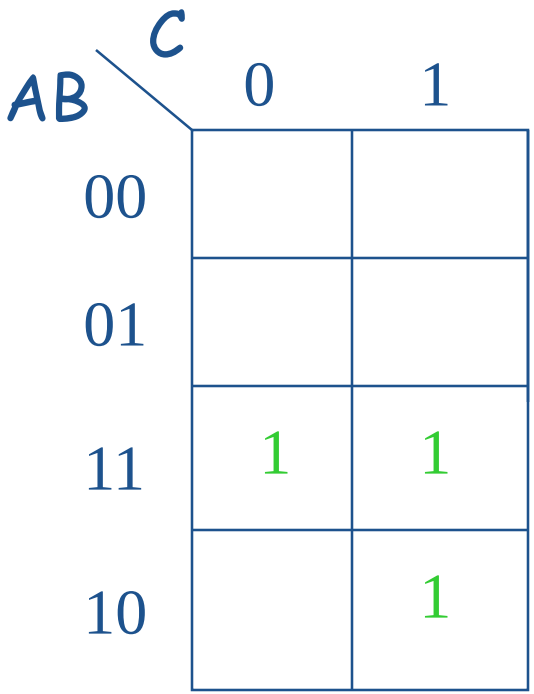


Set-Up K-map table - Z



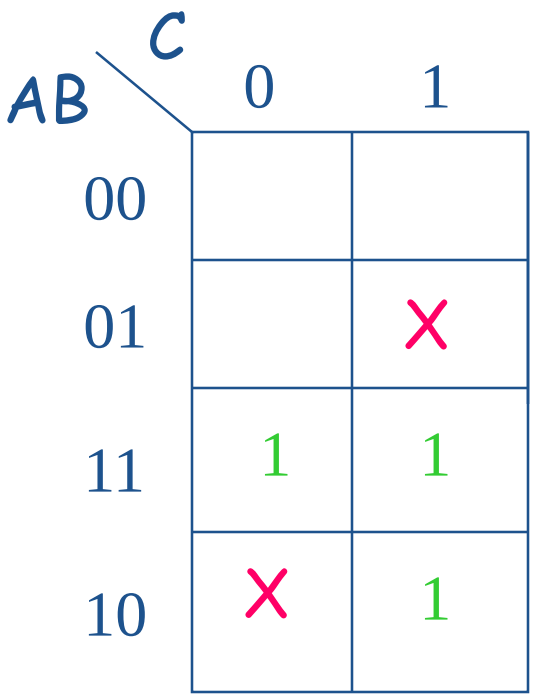
A	B	C	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	X
1	0	0	X
1	0	1	1
1	1	0	1
1	1	1	1

Set-Up K-map table - Z



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

Set-Up K-map table

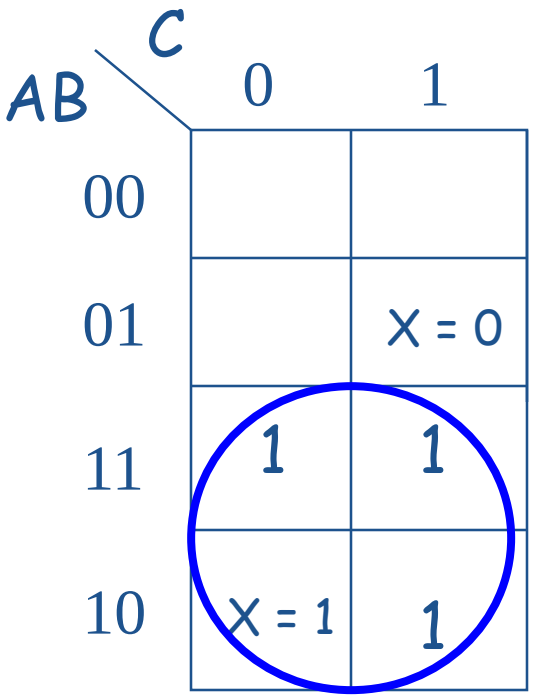


Optimal values

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

Z = ?

Result: Z = A



Z = A