Karnaugh-Maps

To Simplify Logic Expressions

Another, more algorithmic, way to simplify logic expressions: Karnaugh maps or K-maps.

A “MORE” SYSTEMATIC WAY…

Maurice Karnaugh

M. KARNAUGH. *The map method for synthesis of combinational logic circuits.* ***Transactions of the American Institute of Electrical Engineers,*** vol. 72 part I (1953), pp. 593–598.

A modified form of the Veitch chart (XIX 56(2)) method of simplifying truth-functions.

RAYMOND J. NELSON

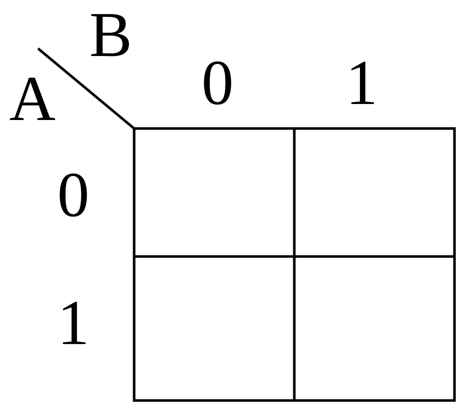
http://en.wikipedia.org/wiki/Maurice_Karnaugh

Karnaugh maps (K-maps)

K-map is a symbolic representation of a truth table that enables us to simplify a logic expression.

- 2-variable K-map
- 3-variable K-map
- 4-variable K-map
- …

2-variable K-map setup



4-cells having values: 0 or 1

3-variable K-map setup

A \ BC	00	01	11	10
0				
1				

Equivalently …

3-variable K-map setup

AB \ C	0	1
00		
01		
11		
10		

The 00, 01, 11, 10 are not in ascending order. This is the **Gray Code ...**

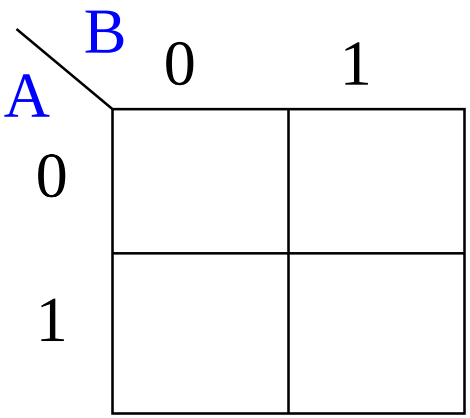
4-variable K-map setup

AB \ CD	00	01	11	10
00				
01				
11				
10				

K-map: 2-variable mapping

EXAMPLE

K-map: 2-variables



given:

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

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

$A\,B$

K-map: 2-variables

A \ B	0	1
0	1	0
1	0	1

given:

A	B	X	
0	0	1	$\rightarrow \overline{A}\,\overline{B}$
0	1	0	
1	0	0	
1	1	1	$\rightarrow A\,B$

K-map: 3-variable mapping

EXAMPLE

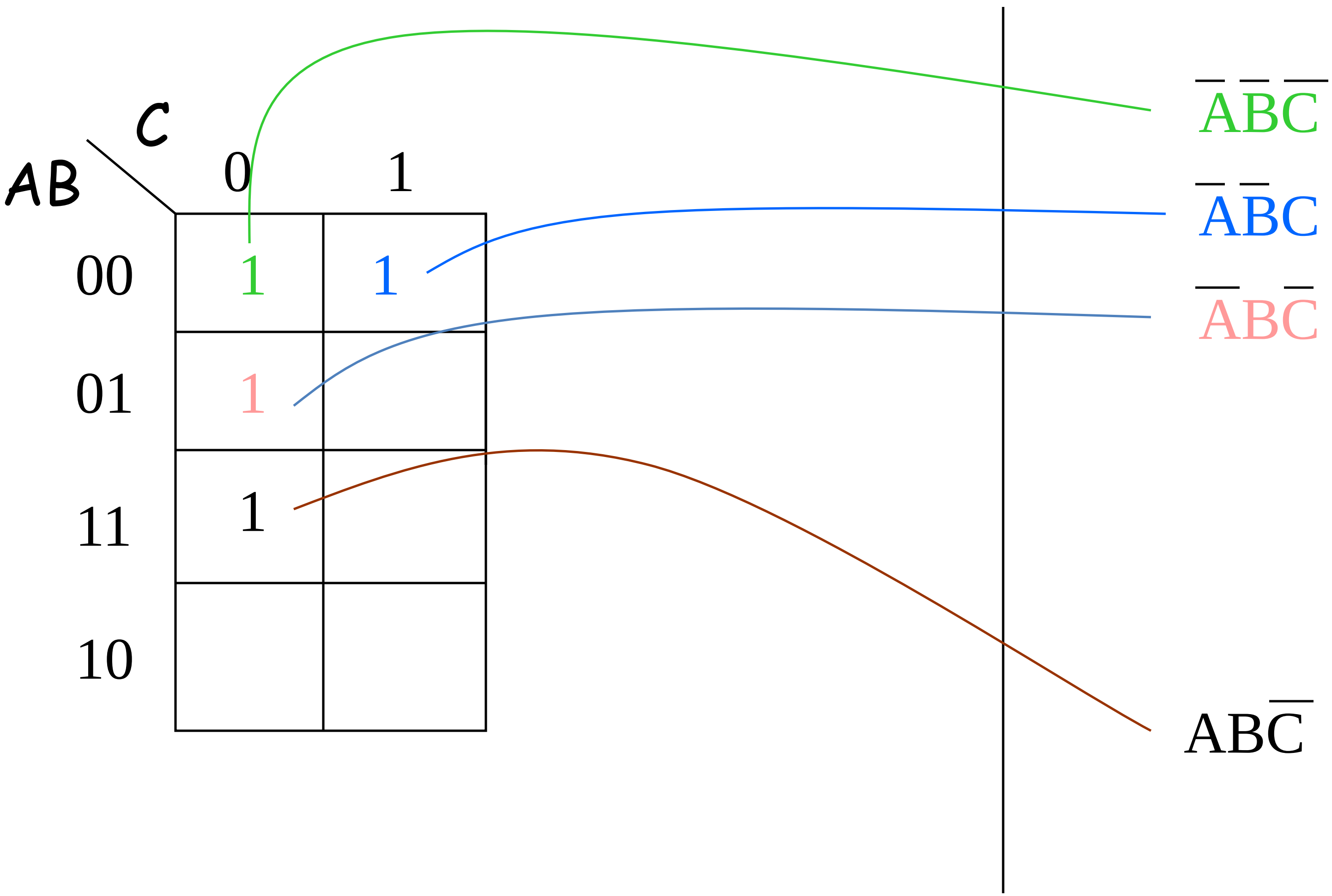
K-map: 3-variables

AB \ C	0	1
00		
01		
11		
10		

given:

A	B	C	X	
0	0	0	1	→ $\overline{A}\,\overline{B}\,\overline{C}$
0	0	1	1	→ $\overline{A}\,\overline{B}C$
0	1	0	1	→ $\overline{A}B\overline{C}$
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	1	→ $AB\overline{C}$
1	1	1	0	

K-map: 3-variables



K-map: 4-variable mapping

EXAMPLE

Four variable K-map

AB \ CD	00	01	11	10
00				
01				
11				
10				

$$X = \bar{A}\,\bar{B}\,\bar{C}\,D + \bar{A}\,B\,\bar{C}\,D + A\,B\,\bar{C}\,D + A\,B\,C\,D$$

Four variable K-map: Example

AB \ CD	00	01	11	10
00		1		
01		1		
11		1	1	
10				

$$X = \bar{A}\,\bar{B}\,\bar{C}\,D + \bar{A}\,B\,\bar{C}\,D + A\,B\,\bar{C}\,D + A\,B\,C\,D$$

How can we simplify using K-maps?

Use looping

looping is a process of combining 1's

Looping: Process of combining 1's

The looping is done in groups of …

2 (pair)

4 (quad)

8 (octel)

1) Looping: Pair (2 … 1’s)

- Looping a pair of adjacent 1’s in a K-map table eliminates one variable that appears in *complemented* (A’) and *uncomplemented* (A) form.

Uniting Theorem

- Looping a pair of adjacent 1's in a K-map table eliminates one variable that appears in *complemented* (A') and *uncomplemented* (A) form.

$$(A' + A)$$

Uniting Theorem

- Looping a pair of adjacent 1's in a K-map table eliminates one variable that appears in *complemented* (A') and *uncomplemented* (A) form.

$$B(A' + A) = ?$$

Uniting Theorem

- Looping a pair of adjacent 1's in a K-map table eliminates one variable that appears in *complemented* (A') and *uncomplemented* (A) form.

$$B(A' + A) = B$$

- The A variable is eliminated …

K-map: 2-variable mapping

EXAMPLE-1

Example-1: Map the X on the K-map

AB \ C	0	1
00		
01		
11		
10		

$$Y = \bar{A}B\bar{C} + AB\bar{C}$$

Y = ?

Example-1: Map the X on the K-map

AB \ C	0	1
00		
01	1	
11		
10		

Y = ?

$$Y = \boxed{\bar{A}B\bar{C}} + AB\bar{C}$$

Example-1: Map the X on the K-map

AB \ C	0	1
00		
01	1	
11	1	
10		

$$Y = ?$$

$$Y = \boxed{\bar{A}B\bar{C}} + \boxed{AB\bar{C}}$$

Example-1: Use the Logic Theorems

AB \ C	0	1
00		
01	1	
11	1	
10		

$Y = ?$

$$Y = \overline{A}B\overline{C} + AB\overline{C}$$

Example-1: Use the Logic Theorems

AB \ C	0	1
00		
01	1	
11	1	
10		

Y = ?

$$Y = \overline{A}B\overline{C} + AB\overline{C}$$

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

$$= B\overline{C}$$

Example-1: 2 logically adjacent 1’s

AB \ C	0	1
00		
01	1	
11	1	
10		

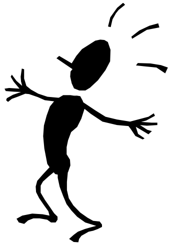
Y = ?

$$Y = \bar{A}B\bar{C} + AB\bar{C}$$

$$= B\bar{C}(\bar{A} + A)$$

$$= B\bar{C}$$

Logically adjacent:
- **Top-Bottom**
- Left-Right



Example-1: Looping of the 1's

AB \ C	0	1
00		
01	1	
11	1	
10		

$Y = ?$

$$Y = \bar{A}B\bar{C} + AB\bar{C}$$

$$= B\bar{C}(\bar{A} + A)$$

$$= B\bar{C}$$

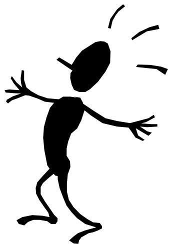
Example-1: Set up the looping Table

AB \ C	0	1
00		
01	1	
11	1	
10		

Y = ?

In the table below the variable **A** is eliminated (X) since it appears to be in complemented (**0**) and un-complemented (**1**) form

A	B	C
0	1	0
1	1	0
X		



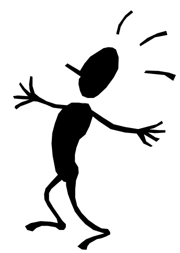
Example-1: The result is: $B\overline{C}$

AB \ C	0	1
00		
01	1	
11	1	
10		

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

In the table below the variable **A** is eliminated (X) since it appears to be in complemented (**0**) and un-complemented (**1**) form

A	B	C
0	1	0
1	1	0
	B	$\overline{C}$



http://www.32x8.com/

K-MAP SOLUTION

K-map: 2-variables

EXAMPLE-2

Example-2

AB \ C	0	1
00		
01	1	1
11		
10		

Y = ?

Logically adjacent:

- Top-Bottom
- **Left-Right**

Example-2; Looping Table

AB \ C	0	1
00		
01	1	1
11		
10		

Y = ?

A	B	C
0	1	0
0	1	1
		X

Example-2; Result = $\bar{A}\,B$

AB \ C	0	1
00		
01	1	1
11		
10		

$Y = \bar{A}\,B$

A B C
0 1 0
0 1 1
$\bar{A}$ B

NEW EXAMPLE-3

Example-3

AB \ C	0	1
00		
01		
11	1	1
10		

Y = ?

Example-3; Looping Table

AB \ C	0	1
00		
01		
11	1	1
10		

Y = ?

A	B	C
1	1	0
1	1	1
		X

Example-3; Solution = AB

AB \ C	0	1
00		
01		
11	1	1
10		

Y = AB

A B C
1 1 0 1 1 1
A B

ANOTHER EXAMPLE-4

Example-4

AB \ C	0	1
00		
01	1	
11	1	
10		

Y = ?

Example-4

AB \ C	0	1
00		
01	1	
11	1	
10		

Y = ?

Example-4; Looping Table

AB \ C	0	1
00		
01	1	
11	1	
10		

$$Y = B\overline{C}$$

A	B	C
0	1	0
1	1	0
	B	$\overline{C}$

K-map: 2-variables

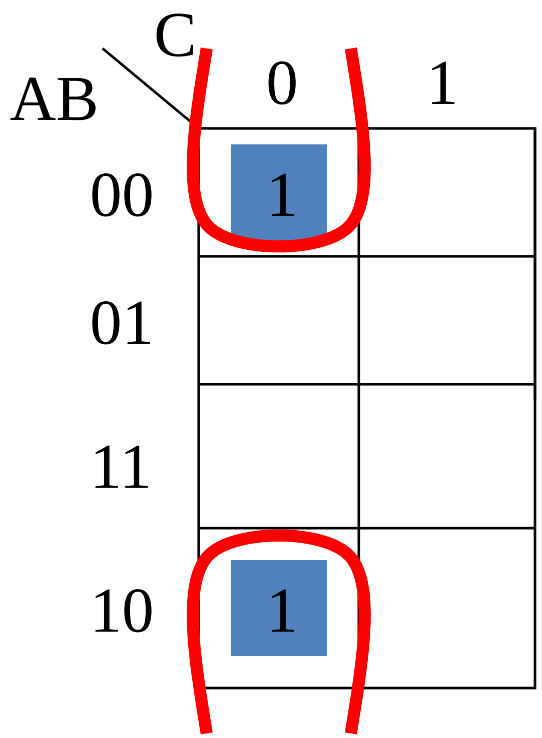
EXAMPLE-5

Example-5

AB \ C	0	1
00	1	
01		
11		
10	1	

Y =

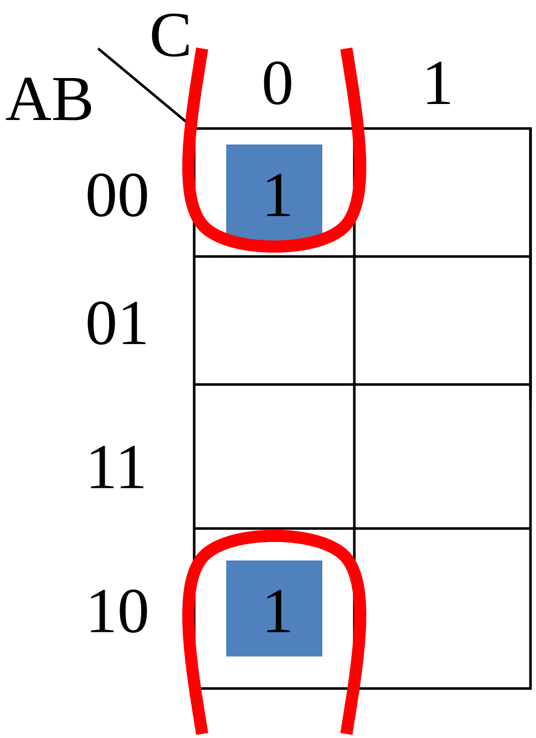
Cyclic property…



Top and bottom rows are considered to be logical adjacent

Y =

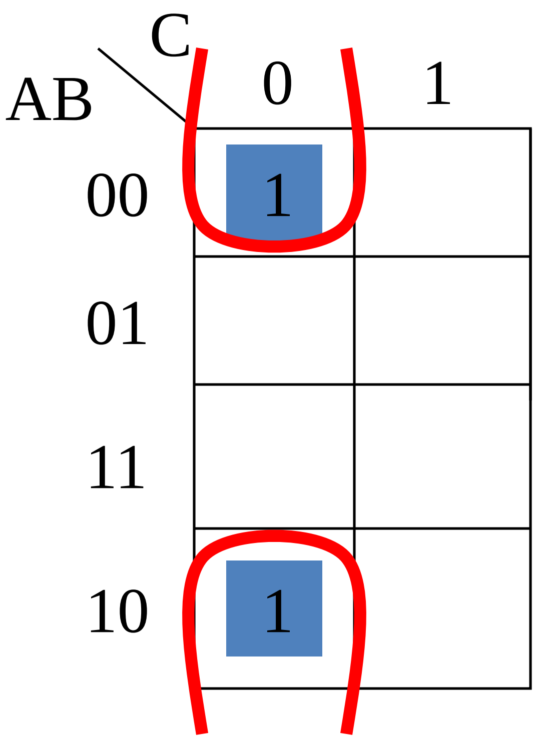
Cyclic property... (Looping Table)



Y =

A	B	C
0	0	0
1	0	0
X		

Result



$$Y = \overline{B}\,\overline{C}$$

A B C
0 0 0
1 0 0
$\overline{B}\,\overline{C}$

K-map: 4-variables

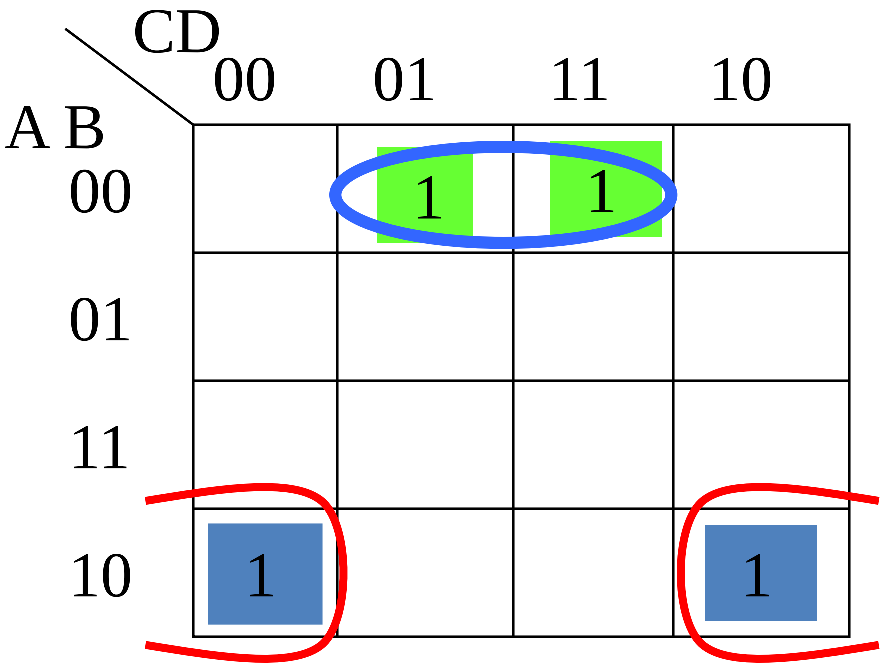
EXAMPLE-6

Example-6

AB \ CD	00	01	11	10
00		1	1	
01				
11				
10	1			1

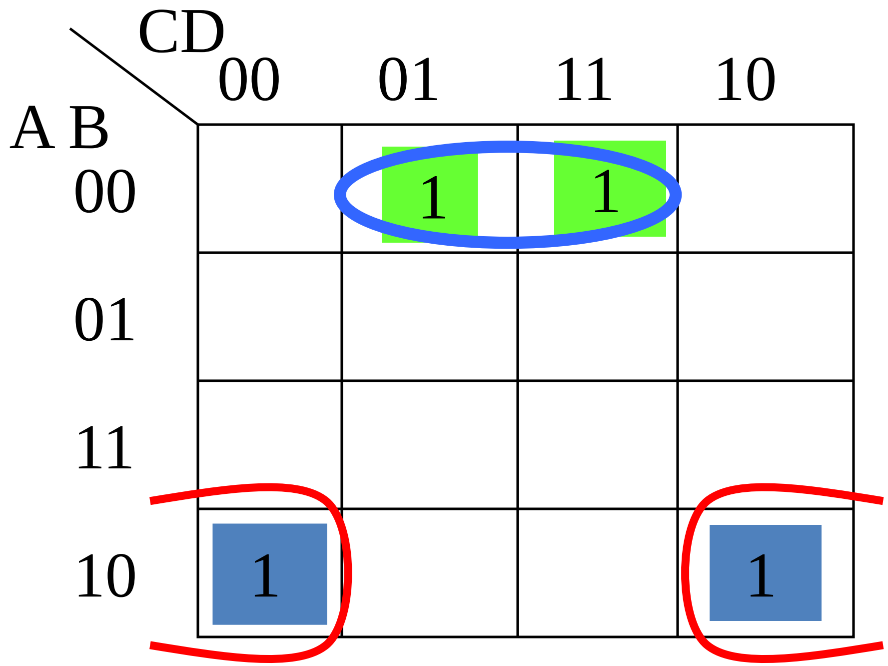
$Y = ?$

Cyclic property … again



Left and right columns are considered to be logical adjacent…

Cyclic property … again



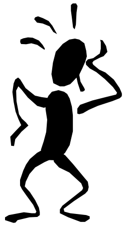
$Y = A\bar{B}\bar{D} + \bar{A}\bar{B}D$

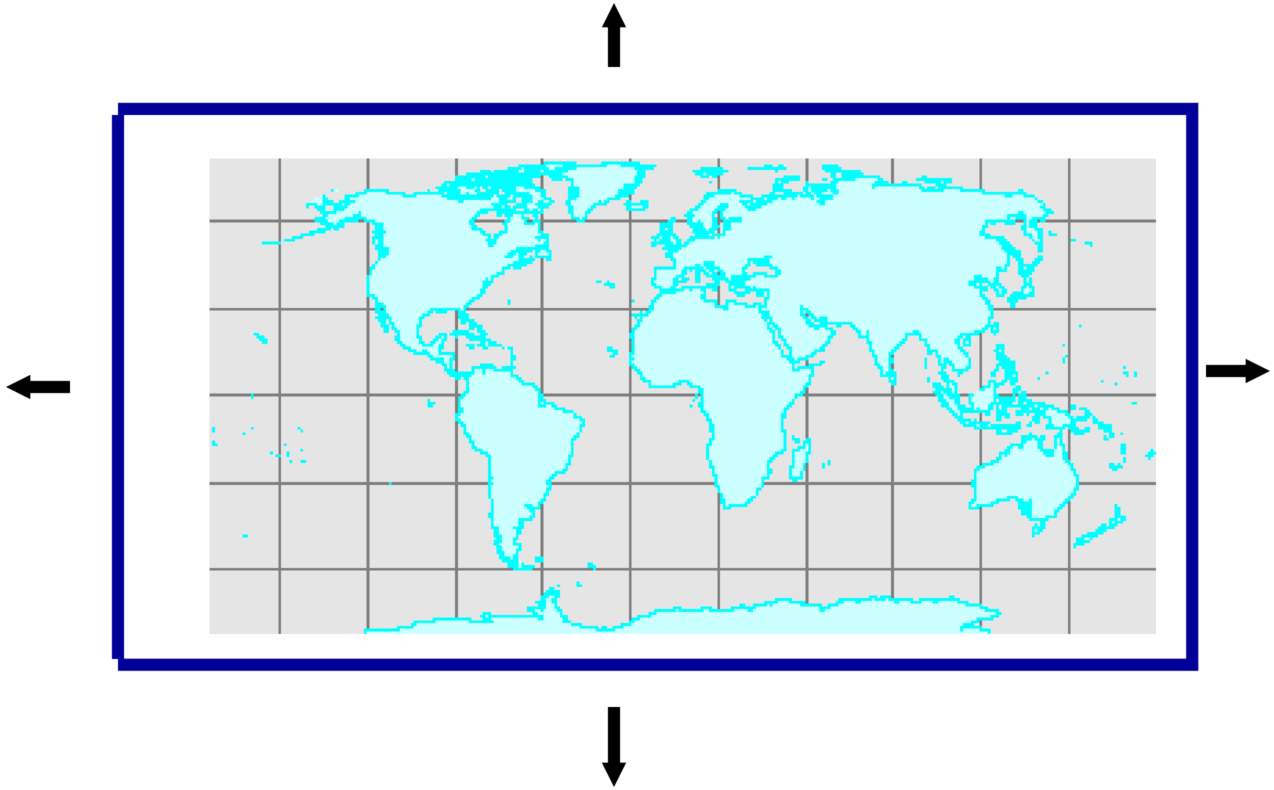
Left and right columns are considered to be logical adjacent…

Adjacent left-right and top-bottom

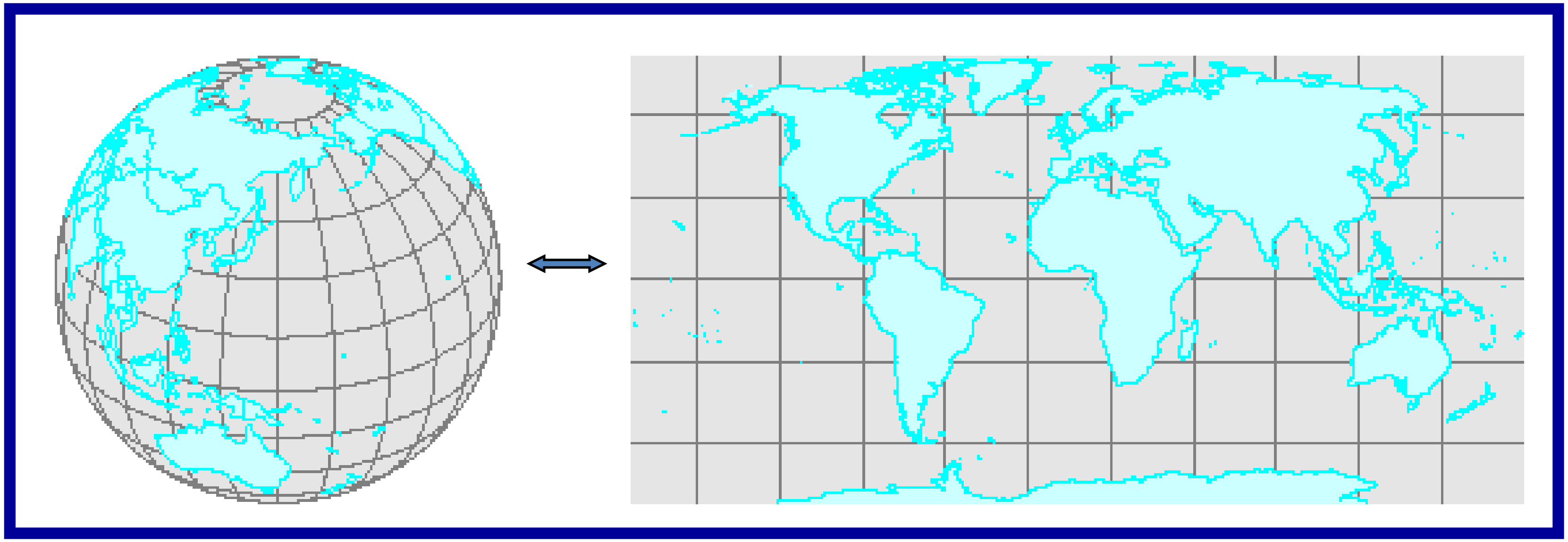
•

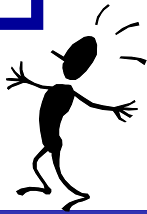
•





Earth





2) Looping: Quad (4 … 1’s)

- Looping (Combining) a quad, of logically adjacent 1’s in a K-map, eliminates two variables that appear in complemented and uncomplemented form.

EXAMPLE-1

3-variable K-Map: Example-1

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

Y = ?

Example-1: Table

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

Y =

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

3-variable K-Map: Example-1

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

$$Y = C$$

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

EXAMPLE-2

Four variable K-map: Example-2

AB \ CD	00	01	11	10
00				
01				
11	1	1	1	1
10				

Y = ?

Example-2; Table

AB \ CD	00	01	11	10
00				
01				
11	1	1	1	1
10				

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

Y = ?

Example-2; Result

AB \ CD	00	01	11	10
00				
01				
11	1	1	1	1
10				

A	B	C	D
1	1	0	0
1	1	0	1
1	1	1	1
1	1	1	0
A	B		

$Y = AB$

EXAMPLE-3

Four variable K-map: Example-3

AB \ CD	00	01	11	10
00				
01		1	1	
11		1	1	
10				

Y = ?

Example-3; Table

AB \ CD	00	01	11	10
00				
01		1	1	
11		1	1	
10				

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

Y = ?

Four variable K-map: Example-3

AB \ CD	00	01	11	10
00				
01		1	1	
11		1	1	
10				

A	B	C	D
0	1	0	1
0	1	1	1
1	1	0	1
1	1	1	1
	B		D

$Y = BD$

EXAMPLE-4

Four variable K-map: Example-4

AB \ CD	00	01	11	10
00				
01				
11	1			1
10	1			1

Y = ?

Left and Right pairs are adjacent

AB \ CD	00	01	11	10
00				
01				
11	1			1
10	1			1

Y = ?

Left and Right pairs are adjacent

AB \ CD	00	01	11	10
00				
01				
11	1			1
10	1			1

$$Y = A\overline{D}$$

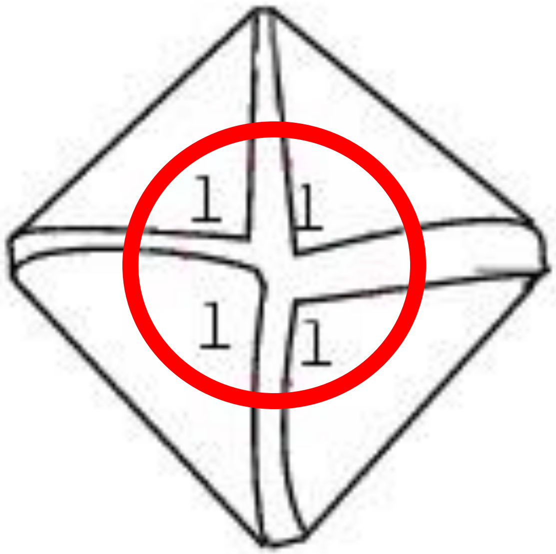
EXAMPLE-5

Four variable K-map: Example-5

AB \ CD	00	01	11	10
00	1			1
01				
11				
10	1			1

Y = ?

The four corner 1's are adjacent



Cyclic property: All 1’s are adjacent

AB \ CD	00	01	11	10
00	1			1
01				
11				
10	1			1

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

$$Y = \overline{B}\,\overline{D}$$

Cyclic property: All 1's are adjacent

AB \ CD	00	01	11	10
00	1			1
01				
11				
10	1			1

A	B	C	D
0	0	0	0
0	0	1	0
1	0	0	0
1	0	1	0
	$\bar{B}$		$\bar{D}$

$$X = \bar{B}\bar{D}$$

3) Looping: Octel (8 … 1’s)

- Looping (combining) an octel, of logically adjacent 1’s, in a K-map eliminates three variables that appear in complemented and uncomplemented form

- In general, looping 2^m terms…eliminates m variables.

EXAMPLE-1

Four variable K-map: Example-1

AB \ CD	00	01	11	10
00				
01	1	1	1	1
11	1	1	1	1
10				

$Y = \ ?$

Looping

AB \ CD	00	01	11	10
00				
01	1	1	1	1
11	1	1	1	1
10				

Y = ?

Table

AB \ CD	00	01	11	10
00				
01	1	1	1	1
11	1	1	1	1
10				

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

Y =

Table

AB \ CD	00	01	11	10
00				
01	1	1	1	1
11	1	1	1	1
10				

A	B	C	D
0	1	0	0
0	1	0	1
0	1	1	1
0	1	1	0
1	1	0	0
1	1	0	1
1	1	1	1
1	1	1	0
	B		

$$Y = B$$

EXAMPLE-2

Four variable K-map: Example-2

AB \ CD	00	01	11	10
00	1	1		
01	1	1		
11	1	1		
10	1	1		

X = ?

Looping Table

AB	C'D' 00	C'D 01	CD 11	CD' 10
00	1	1		
01	1	1		
11	1	1		
10	1	1		

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

$X =$

Looping Table

AB \ CD	C'D' 00	C'D 01	CD 11	CD' 10
00	1	1		
01	1	1		
11	1	1		
10	1	1		

A	B	C	D
0	0	0	0
0	0	0	1
0	1	0	0
0	1	0	1
1	1	0	0
1	1	0	1
1	0	0	0
1	0	0	1

$\overline{C}$

$$X = \overline{C}$$

EXAMPLE-3

Four variable K-map: Example-3

AB \ CD	00	01	11	10
00	1	1	1	1
01				
11				
10	1	1	1	1

X = ?

Solution

AB \ CD	00	01	11	10
00	1	1	1	1
01				
11				
10	1	1	1	1

$$X = \overline{B}$$

EXAMPLE-4

Four variable K-map: Example-4

AB \ CD	00	01	11	10
00	1			1
01	1			1
11	1			1
10	1			1

X = ?

Solution

AB \ CD	00	01	11	10
00	1			1
01	1			1
11	1			1
10	1			1

$$X = \overline{D}$$

MORE EXAMPLES

More Examples-1

AB \ CD	00	01	11	10
00		1		
01		1	1	1
11	1	1	1	
10			1	

X = ?

Looping…

AB \ CD	00	01	11	10
00		1		
01		1	1	1
11	1	1	1	
10			1	

X = ?

Looping...

AB \ CD	00	01	11	10
00		1		
01		1	1	1
11	1	1	1	
10			1	

X =

Looping…

Correct but not optimal

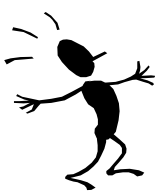
AB \ CD	00	01	11	10
00		1		
01		1	1	1
11	1	1	1	
10			1	

$$X = A'C'D + A'BC + ABC' + ACD + BD$$

BD is not needed

Correct and optimal

AB \ CD	00	01	11	10
00		1		
01		1	1	1
11	1	1	1	
10			1	

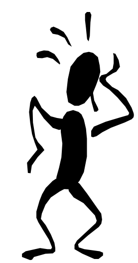


$$X = A'C'D + A'BC + ABC' + ACD + \cancel{BD}$$

More Examples-(2)

AB \ CD	00	01	11	10
00				1
01		1	1	
11		1	1	
10			1	

X = ?



Minimal simplification

AB \ CD	00	01	11	10
00				1
01		1	1	
11		1	1	
10			1	

X = ?

Minimal simplification

Correct and optimal

AB \ CD	00	01	11	10
00				1
01		1	1	
11		1	1	
10			1	

$$X = ACD + BD + A'B'CD'$$

More Examples-(3)

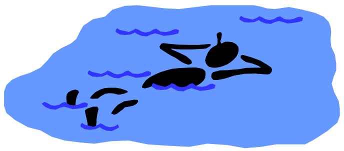
AB \ CD	00	01	11	10
00			1	
01	1	1	1	1
11	1	1		
10				

X = ?

Minimal simplification

AB \ CD	00	01	11	10
00			1	
01	1	1	1	1
11	1	1		
10				

X = ?

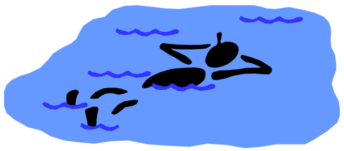


Minimal simplification

Correct and optimal

AB \ CD	00	01	11	10
00			1	
01	1	1	1	1
11	1	1		
10				

$$X = A'CD + A'B + BC'$$



More Examples-(4)

AB \ CD	00	01	11	10
00			1	
01	1	1	1	1
11	1	1	1	
10				

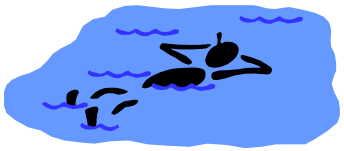
X = ?

Minimal simplification

Correct and optimal

AB \ CD	00	01	11	10
00			1	
01	1	1	1	1
11	1	1	1	
10				

X = A’CD + A’B + BC’ + BD



More Examples-(5)

AB \ CD	00	01	11	10
00			1	1
01	1	1	1	1
11	1	1	1	
10				

X = ?

Simplification: Example-(5)

Correct and optimal

AB \ CD	00	01	11	10
00			1	1
01	1	1	1	1
11	1	1	1	
10				

X = BC' + BD + A'C

More Examples-(6)

AB \ CD	00	01	11	10
00			1	1
01	1	1	1	1
11	1	1	1	
10			1	

X = ?

Simplification: Example-(6)

AB \ CD	00	01	11	10
00			1	1
01	1	1	1	1
11	1	1	1	
10			1	

X = ?

Simplification: Example-(6)

Correct and optimal

AB \ CD	00	01	11	10
00			1	1
01	1	1	1	1
11	1	1	1	
10			1	

X = BC’ + A’C + CD

Simplification: Example-(7)

AB \ CD	00	01	11	10
00			1	1
01	1	1	1	1
11	1	1	1	1
10				

X = ?

Simplification: Example-(7)

Correct and optimal

AB \ CD	00	01	11	10
00			1	1
01	1	1	1	1
11	1	1	1	1
10				

X = B + A'C

Simplification: Example-(8)

AB \ CD	00	01	11	10
00		1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	

X =

Simplification: Example-(8)

Correct and optimal

AB \ CD	00	01	11	10
00		1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	

X = B + D + AC’ + A’C

Summary: Looping (K-Map)

- Loop the isolated 1's (those not logically adjacent to any other 1's). Look for the 1's that are adjacent to any loops and loop any pair containing such 1's. Each 1 must be looped at least once. However, it may be covered more than once (optimal).
  - Loop any octels [8] (optimal)
  - Loop any quads [4] (optimal)
  - Loop any pairs [2] (optimal)
  - Form the OR sum of all terms in the loops.

YouTube and Wikipedia

https://youtu.be/3vkMgTmieZI

https://youtu.be/-2JClp-erHY

https://youtu.be/FOf00W8WSBg

https://en.wikipedia.org/wiki/Karnaugh_map