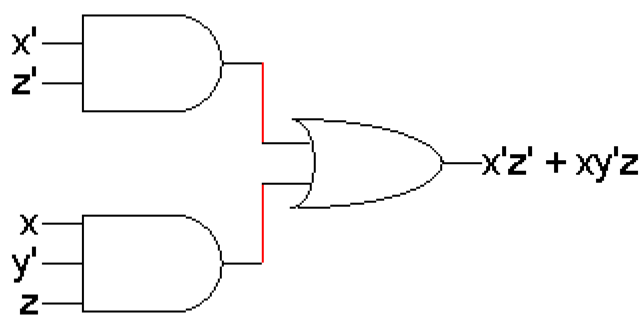
Karnaugh maps

- The basic Boolean operations are AND, OR and NOT.
- These operations can be combined to form complex expressions, which can also be directly translated into a hardware circuit.
- Boolean algebra helps us simplify expressions and circuits.

- we'll look at a graphical technique for simplifying an expression into a minimal sum of products (MSP) form:
  - There are a minimal number of product terms in the expression.
  - Each term has a minimal number of literals.
- Circuit-wise, this leads to a *minimal* two-level implementation.

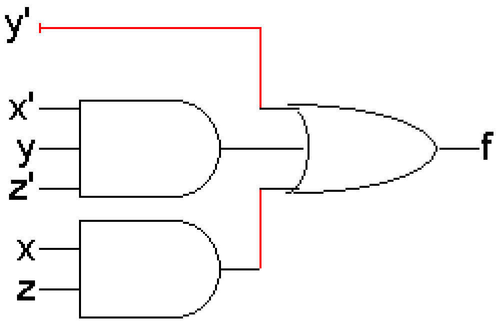


Standard forms of expressions

- A sum of products (SOP) expression contains:
  - Only OR (sum) operations at the "outermost" level
  - Each term that is summed must be a product of literals

$$f(x,y,z) = y' + x'yz' + xz$$

- The advantage is that any sum of products expression can be implemented using a two-level circuit
  - literals and their complements at the "0th" level
  - AND gates at the first level
  - a single OR gate at the second level



Terminology: Minterms

- A minterm is a special product of literals, in which each input variable appears exactly once.
- A function with n variables has 2^n minterms (since each variable can appear complemented or not)
- A three-variable function, such as f(x,y,z), has $2^3 = 8$ minterms:

x'y'z'	x'y'z	x'yz'	x'yz
xy'z'	xy'z	xyz'	xyz

- Each minterm is true for exactly one combination of inputs:

Minterm	Is true when…	Shorthand
x'y'z'	x=0, y=0, z=0	m_0
x'y'z	x=0, y=0, z=1	m_1
x'yz'	x=0, y=1, z=0	m_2
x'yz	x=0, y=1, z=1	m_3
xy'z'	x=1, y=0, z=0	m_4
xy'z	x=1, y=0, z=1	m_5
xyz'	x=1, y=1, z=0	m_6
xyz	x=1, y=1, z=1	m_7

Terminology: Sum of minterms form

- Every function can be written as a sum of minterms, which is a special kind of sum of products form
- If you have a truth table for a function, you can write a sum of minterms expression just by picking out the rows of the table where the function output is 1.

x	y	z	f(x,y,z)	f'(x,y,z)
0	0	0	1	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	1	0
1	1	1	0	1

$$f = x'y'z' + x'y'z + x'yz' + x'yz + xyz'$$
$$= m_0 + m_1 + m_2 + m_3 + m_6$$
$$= \Sigma m(0,1,2,3,6)$$

$$f' = xy'z' + xy'z + xyz$$
$$= m_4 + m_5 + m_7$$
$$= \Sigma m(4,5,7)$$

Re-arranging the truth table

- A two-variable function has four possible minterms. We can re-arrange these minterms into a Karnaugh map.

x	y	minterm
0	0	x'y'
0	1	x'y
1	0	xy'
1	1	xy

		Y	
		0	1
X	0	x'y'	x'y
	1	xy'	xy

- Now we can easily see which minterms contain common literals.
  - Minterms on the left and right sides contain y' and y respectively.
  - Minterms in the top and bottom rows contain x' and x respectively.

		Y	
		0	1
X	0	x'y'	x'y
	1	xy'	xy

	Y'	Y
X'	x'y'	x'y
X	xy'	xy

Karnaugh map simplifications

- Imagine a two-variable sum of minterms:

$$x'y' + x'y$$

- Both of these minterms appear in the top row of a Karnaugh map, which means that they both contain the literal x'.

		Y
	$x'y'$	$x'y$
X	xy'	xy

- What happens if you simplify this expression using Boolean algebra?

$$
\begin{aligned}
x'y' + x'y &= x'(y' + y) && [\text{ Distributive }] \\
&= x' \bullet 1 && [\ y + y' = 1\ ] \\
&= x' && [\ x \bullet 1 = x\ ]
\end{aligned}
$$

More two-variable examples

- Another example expression is x’y + xy.
  - Both minterms appear in the right side, where y is uncomplemented.
  - Thus, we can reduce x’y + xy to just y.

		Y
	x’y’	x’y
X	xy’	xy

- How about x’y’ + x’y + xy?
  - We have x’y’ + x’y in the top row, corresponding to x’.
  - There’s also x’y + xy in the right side, corresponding to y.
  - This whole expression can be reduced to x’ + y.

		Y
	x’y’	x’y
X	xy’	xy

A three-variable Karnaugh map

- For a three-variable expression with inputs x, y, z, the arrangement of minterms is more tricky:

		YZ			
		00	01	11	10
X	0	x’y’z’	x’y’z	x’yz	x’yz’
	1	xy’z’	xy’z	xyz	xyz’

		YZ			
		00	01	11	10
X	0	m_0	m_1	m_3	m_2
	1	m_4	m_5	m_7	m_6

- Another way to label the K-map (use whichever you like):

			Y	
	x’y’z’	x’y’z	x’yz	x’yz’
X	xy’z’	xy’z	xyz	xyz’
		Z		

			Y	
	m_0	m_1	m_3	m_2
X	m_4	m_5	m_7	m_6
		Z		

- With this ordering, any group of 2, 4 or 8 adjacent squares on the map contains common literals that can be factored out.

			Y	Y
	x'y'z'	x'y'z	x'yz	x'yz'
X	xy'z'	xy'z	xyz	xyz'
		Z	Z	

$$
\begin{aligned}
& x'y'z + x'yz \\
=\ & x'z(y' + y) \\
=\ & x'z \bullet 1 \\
=\ & x'z
\end{aligned}
$$

- "Adjacency" includes wrapping around the left and right sides:

			Y	Y
	x'y'z'	x'y'z	x'yz	x'yz'
X	xy'z'	xy'z	xyz	xyz'
		Z	Z	

$$
\begin{aligned}
& x'y'z' + xy'z' + x'yz' + xyz' \\
=\ & z'(x'y' + xy' + x'y + xy) \\
=\ & z'(y'(x' + x) + y(x' + x)) \\
=\ & z'(y'+y) \\
=\ & z'
\end{aligned}
$$

- We'll use this property of adjacent squares to do our simplifications.

Example K-map simplification

- Let's consider simplifying $f(x,y,z) = xy + y'z + xz$.
- First, you should convert the expression into a sum of minterms form, if it's not already.
  - The easiest way to do this is to make a truth table for the function, and then read off the minterms.
  - You can either write out the literals or use the minterm shorthand.
- Here is the truth table and sum of minterms for our example:

x	y	z	f(x,y,z)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$$\begin{aligned} f(x,y,z) &= x'y'z + xy'z + xyz' + xyz \\ &= m_1 + m_5 + m_6 + m_7 \end{aligned}$$

Making the example K-map

- Next up is drawing and filling in the K-map.
  - Put 1s in the map for each minterm, and 0s in the other squares.
  - You can use either the minterm products or the shorthand to show you where the 1s and 0s belong.
- In our example, we can write f(x,y,z) in two equivalent ways.

$$f(x,y,z) = x'y'z + xy'z + xyz' + xyz$$

			Y	Y
	x'y'z'	x'y'z	x'yz	x'yz'
X	xy'z'	xy'z	xyz	xyz'
		Z	Z	

$$f(x,y,z) = m_1 + m_5 + m_6 + m_7$$

			Y	Y
	m_0	m_1	m_3	m_2
X	m_4	m_5	m_7	m_6
		Z	Z	

- In either case, the resulting K-map is shown below.

			Y	Y
	0	1	0	0
X	0	1	1	1
		Z	Z	

K-maps from truth tables

- You can also fill in the K-map directly from a truth table.
  - The output in row *i* of the table goes into square m_i of the K-map.
  - Remember that the rightmost columns of the K-map are “switched.”

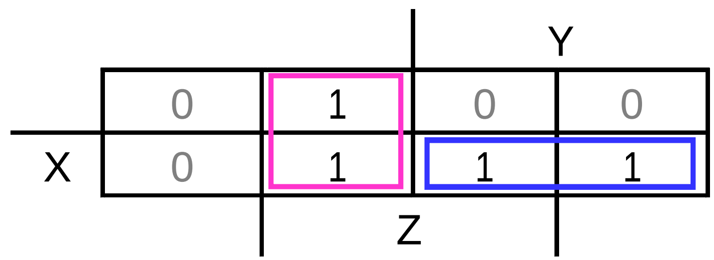
			Y	Y
	m_0	m_1	m_3	m_2
X	m_4	m_5	m_7	m_6
		Z	Z	

x	y	z	f(x,y,z)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

			Y	Y
	0	1	0	0
X	0	1	1	1
		Z	Z	

Grouping the minterms together

- The most difficult step is grouping together all the 1s in the K-map.
  - Make rectangles around groups of one, two, four or eight 1s.
  - All of the 1s in the map should be included in at least one rectangle.
  - Do *not* include any of the 0s.



- Each group corresponds to one product term. For the simplest result:
  - Make as few rectangles as possible, to minimize the number of products in the final expression.
  - Make each rectangle as large as possible, to minimize the number of literals in each term.
  - It's all right for rectangles to overlap, if that makes them larger.

Reading the MSP from the K-map

– Each rectangle corresponds to one product term.

– The product is determined by finding the common literals in that rectangle.

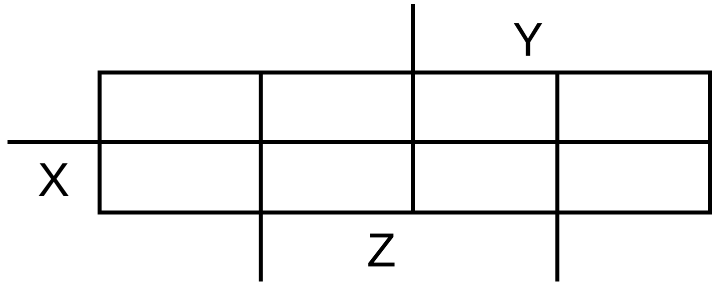
			Y	
	0	1	0	0
X	0	1	1	1
		Z		

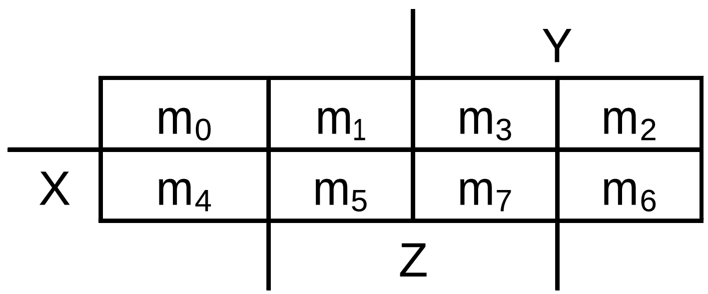
			Y	
	x'y'z'	x'y'z	x'yz	x'yz'
X	xy'z'	xy'z	xyz	xyz'
		Z		

- For our example, we find that xy + y'z + xz = y'z + xy.

Practice K-map 1

- Simplify the sum of minterms $m_1 + m_3 + m_5 + m_6$.





Solutions for practice K-map 1

- Here is the filled in K-map, with all groups shown.
  - The magenta and green groups overlap, which makes each of them as large as possible.
  - Minterm m_6 is in a group all by its lonesome.

			Y	
	0	1	1	0
X	0	1	0	1
		Z		

- The final MSP here is $x'z + y'z + xyz'$.

Four-variable K-maps

- We can do four-variable expressions too!
  - The minterms in the third and fourth columns, *and* in the third and fourth rows, are switched around.
  - Again, this ensures that adjacent squares have common literals.

		00	01	11	10	
				Y	Y	
00		w’x’y’z’	w’x’y’z	w’x’yz	w’x’yz’	
01		w’xy’z’	w’xy’z	w’xyz	w’xyz’	X
11	W	wxy’z’	wxy’z	wxyz	wxyz’	X
10	W	wx’y’z’	wx’y’z	wx’yz	wx’yz’	
			Z	Z		

			Y	Y	
	m_0	m_1	m_3	m_2	
	m_4	m_5	m_7	m_6	X
W	m_{12}	m_{13}	m_{15}	m_{14}	X
W	m_8	m_9	m_{11}	m_{10}	
		Z	Z		

- Grouping minterms is similar to the three-variable case, but:
  - You can have rectangular groups of 1, 2, 4, 8 or 16 minterms.
  - You can wrap around *all four* sides.

Example: Simplify $m_0+m_2+m_5+m_8+m_{10}+m_{13}$

- The expression is already a sum of minterms, so here's the K-map:

			Y	Y	
	1	0	0	1	
	0	1	0	0	X
W	0	1	0	0	X
W	1	0	0	1	
		Z	Z		

			Y	Y	
	m_0	m_1	m_3	m_2	
	m_4	m_5	m_7	m_6	X
W	m_{12}	m_{13}	m_{15}	m_{14}	X
W	m_8	m_9	m_{11}	m_{10}	
		Z	Z		

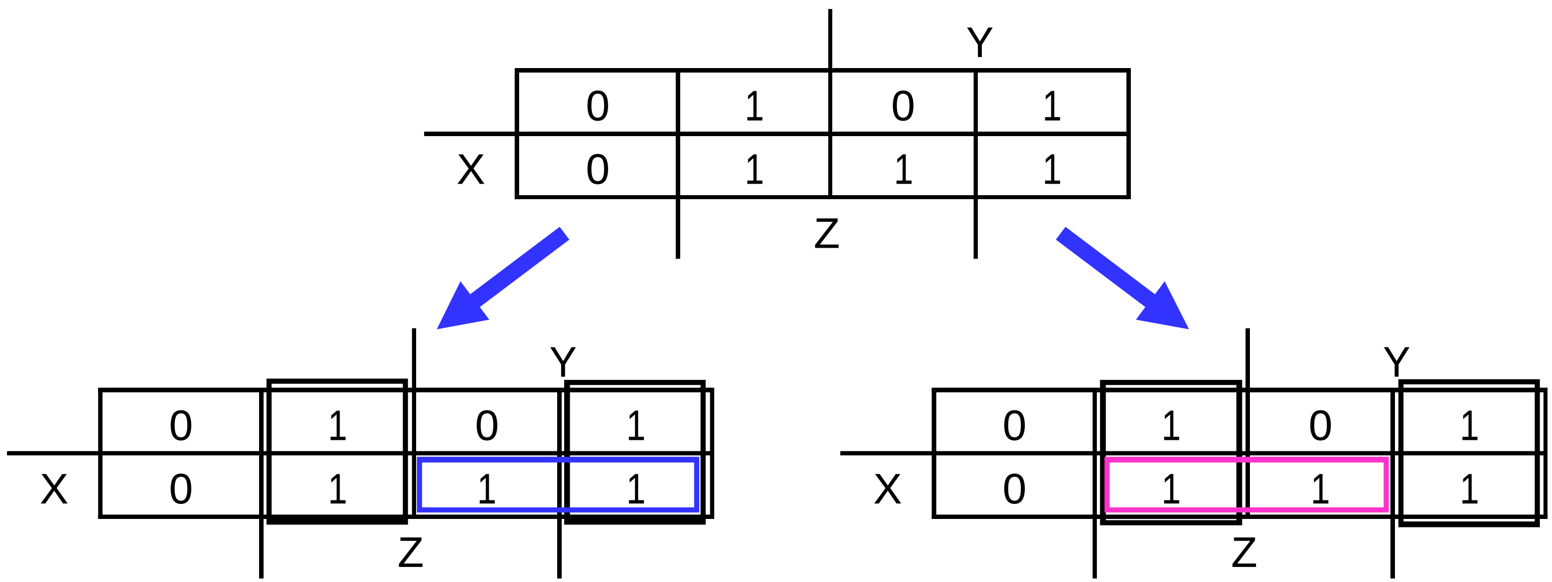
- We can make the following groups, resulting in the MSP x'z' + xy'z.

			Y	Y	
	1	0	0	1	
	0	1	0	0	X
W	0	1	0	0	X
W	1	0	0	1	
		Z	Z		

			Y	Y	
	w'x'y'z'	w'x'y'z	w'x'yz	w'x'yz'	
	w'xy'z'	w'xy'z	w'xyz	w'xyz'	X
W	wxy'z'	wxy'z	wxyz	wxyz'	X
W	wx'y'z'	wx'y'z	wx'yz	wx'yz'	
		Z	Z		

K-maps can be tricky!

- There may not necessarily be a *unique* MSP. The K-map below yields two valid and equivalent MSPs, because there are two possible ways to include minterm m_7.



y'z + yz' + xy

y'z + yz' + xz

- Remember that overlapping groups is possible, as shown above.

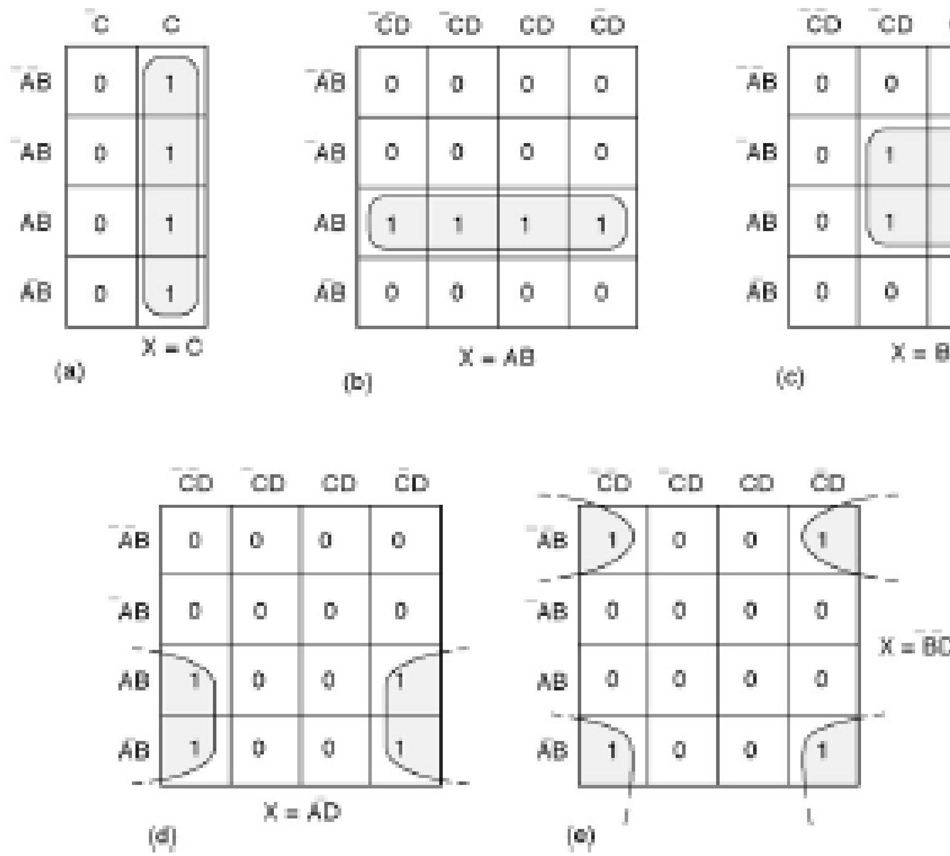
More Examples of grouping

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

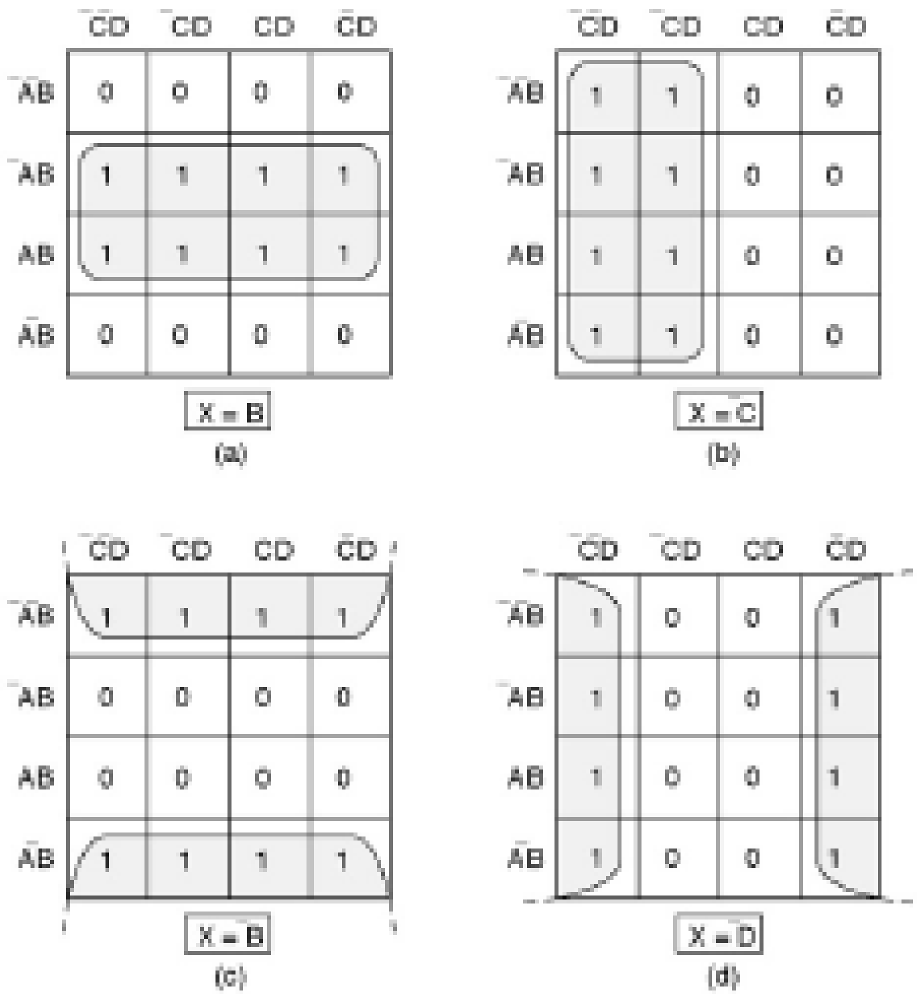
AB\CD	00	01	11	10
00	0	0	0	0
01	1	0	0	1
11	1	0	1	1
10	0	0	0	0

$$AB + \overline{C}D$$

$$B\overline{D} + ABC$$



X = C
(a)
X = AB
(b)
X = BD
(c)
X = AD
(d)
X = BD
(e)



(a) (b) (c) (d)