Gate-level minimization:
The map method up to four variable
don’t care conditions
POS simplification

Lecture 3

Dronacharya Group of Institutions

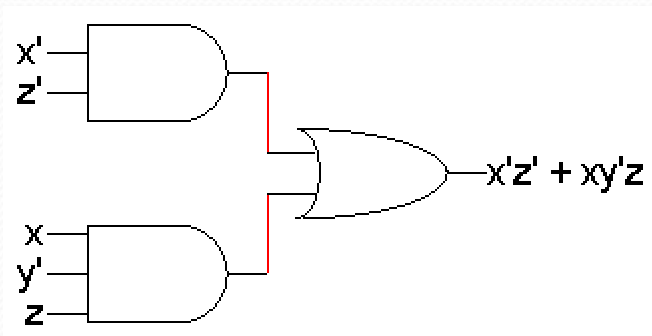
Karnaugh Maps

- Boolean algebra helps us simplify expressions and circuits

- Karnaugh Map: A graphical technique for simplifying a Boolean expression into either form:
  - minimal sum of products (MSP)
  - minimal product of sums (MPS)

- Goal of the simplification.
  - There are a minimal number of product/sum terms
  - Each term has a minimal number of literals

- Circuit-wise, this leads to a *minimal* two-level implementation



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:

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

X \ YZ	00	01	11	10
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		

Why the funny ordering?

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

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

K-maps From Truth Tables

- We can 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"

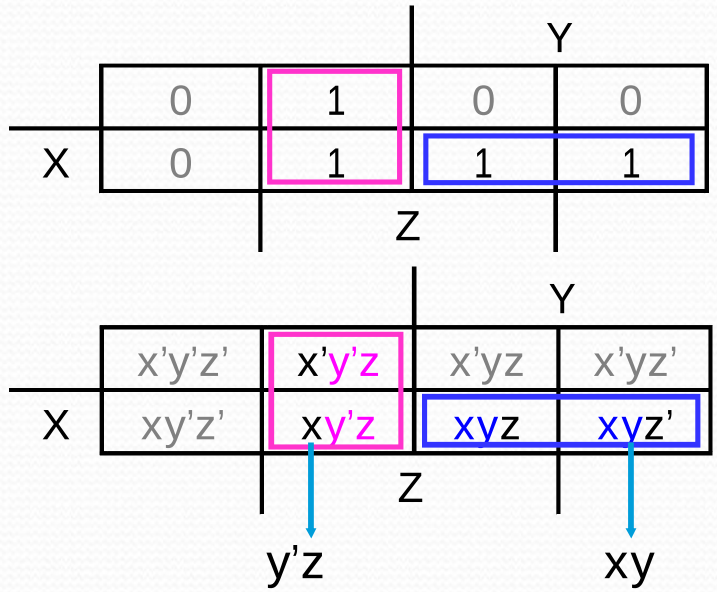
			Y	Y
	m_0	m_1	m_3	m_2
X	m_4	m_5	m_7	m_6
		Z	Z	

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	

Reading the MSP from the K-map

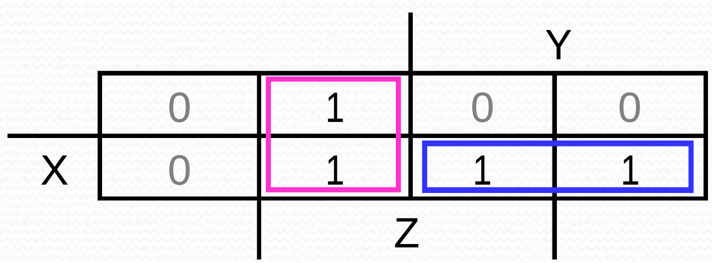
- You can find the minimal SoP expression
  - Each rectangle corresponds to one product term
  - The product is determined by finding the common literals in that rectangle



F(x,y,z)= y'z + xy

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.
- Rectangles can be overlapped, if that makes them larger.

K-map Simplification of SoP Expressions

- Let's consider simplifying $f(x,y,z) = xy + y'z + xz$
- You should convert the expression into a sum of minterms form,
  - 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

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

$$= m_1 + m_5 + m_6 + m_7$$

Unsimplifying Expressions

- You can convert the expression to a sum of minterms with Boolean algebra
  - Apply the distributive law in reverse to add in missing variables.
  - Very few people actually do this, but it's occasionally useful.

$$
\begin{aligned}
xy + y'z + xz &= (xy \bullet 1) + (y'z \bullet 1) + (xz \bullet 1) \\
&= (xy \bullet (z' + z)) + (y'z \bullet (x' + x)) + (xz \bullet (y' + y)) \\
&= (xyz' + xyz) + (x'y'z + xy'z) + (xy'z + xyz) \\
&= xyz' + xyz + x'y'z + xy'z \\
&= m_1 + m_5 + m_6 + m_7
\end{aligned}
$$

- In both cases, we're actually "unsimplifying" our example expression
  - The resulting expression is larger than the original one!
  - But having all the individual minterms makes it easy to combine them together with the K-map

Making the Example K-map

- In our example, we can write f(x,y,z) in two equivalent ways

  $f(x,y,z) = x'y'z + xy'z + xyz' + xyz$ $\qquad$ $f(x,y,z) = m_1 + m_5 + m_6 + m_7$

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

			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	

Practice K-map 1

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

			Y	Y
X				
		Z	Z	

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

Solutions for Practice K-map 1

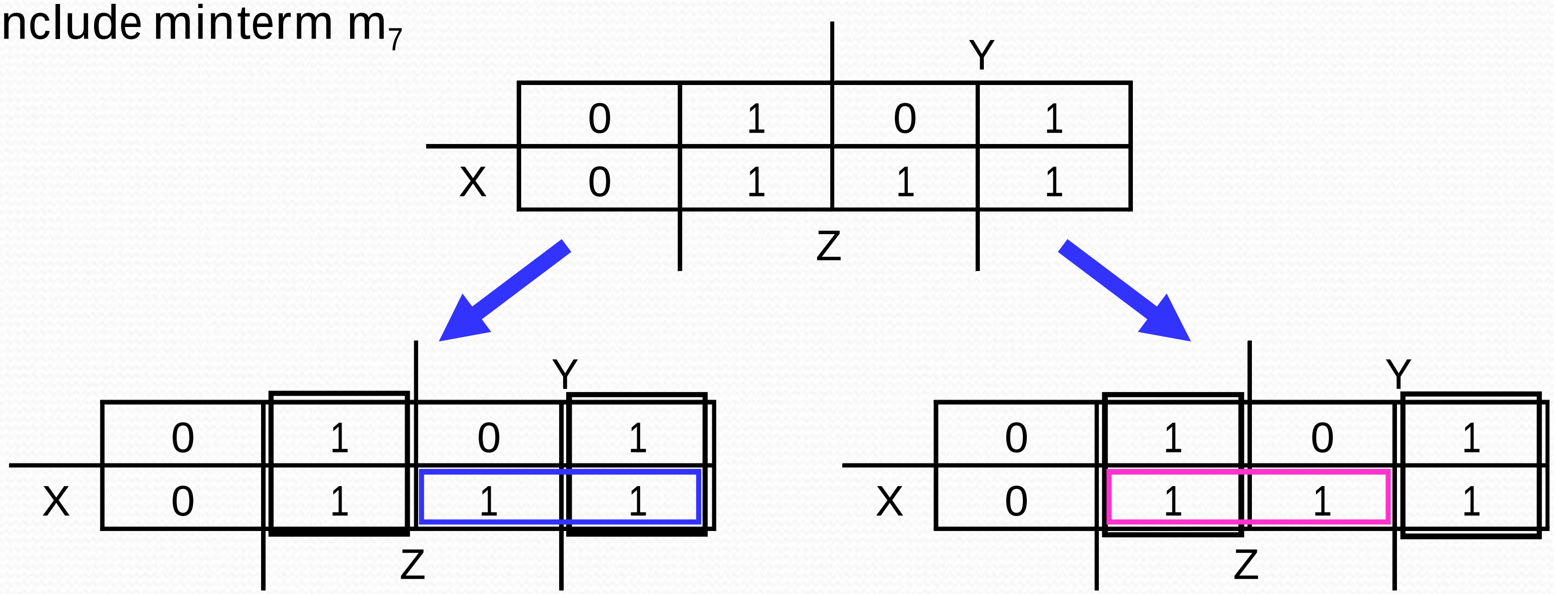
- Here is the filled in K-map, with all groups shown
  - The magenta and green groups overlap, which makes each of them as large as possible
  - Minterm m_6 is in a group all by its lonesome

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

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

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

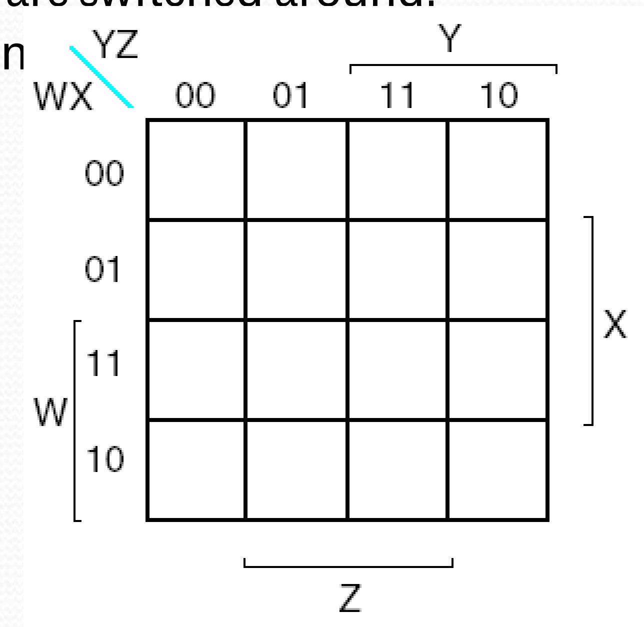


y'z + yz' + xy

y'z + yz' + xz

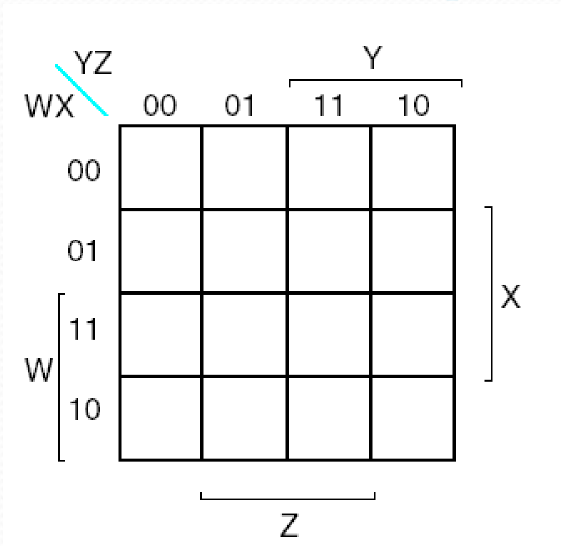
- Remember that overlapping groups is possible, as shown above

Four-variable K-maps – f(W,X,Y,Z)

- 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 enve common literals

- 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

Four-variable K-maps



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’
wxy’z’	wxy’z	wxyz	wxyz’
wx’y’z’	wx’y’z	wx’yz	wx’yz’

X

W

Z

Y

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

X

W

Z

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		

PoS Optimization

- Maxterms are grouped to find minimal PoS expression

x \ yz	00	01	11	10
0	x +y+z	x+y+z'	x+y'+z'	x+y'+z
1	x' +y+z	x'+y+z'	x'+y'+z'	x'+y'+z

PoS Optimization

- F(W,X,Y,Z)= ∏ M(0,1,2,4,5)

x \ yz	00	01	11	10
0	**x +y+z**	**x+y+z'**	x+y'+z'	**x+y'+z**
1	**x' +y+z**	**x'+y+z'**	x'+y'+z'	x'+y'+z

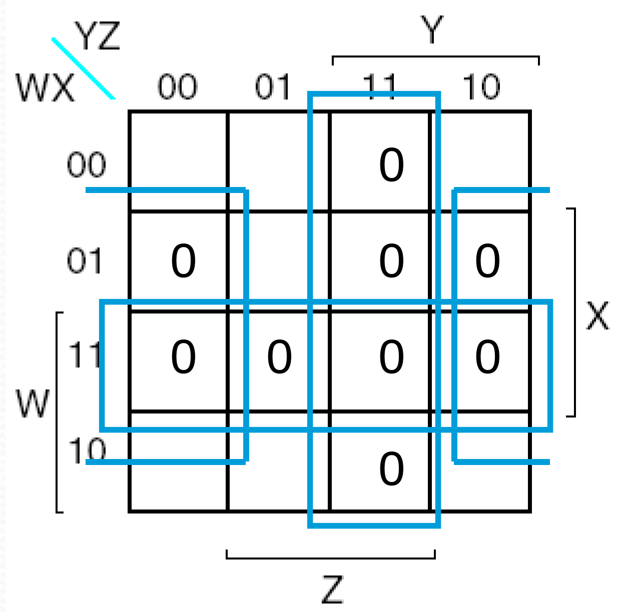
x \ yz	00	01	11	10
0	0	0	1	0
1	0	0	1	1

F(W,X,Y,Z)= Y . (X + Z)

PoS Optimization from SoP

F(W,X,Y,Z)= $\Sigma m(0,1,2,5,8,9,10)$

$= \Pi M(3,4,6,7,11,12,13,14,15)$



F(W,X,Y,Z)= (W' + X')(Y' + Z')(X' + Z)

Or,

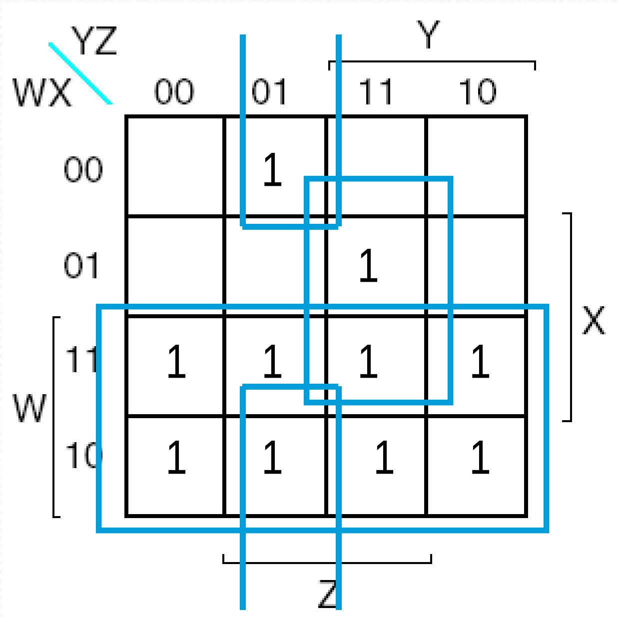
F(W,X,Y,Z)= X'Y' + X'Z' + W'Y'Z

Which one is the minimal one?

SoP Optimization from PoS

F(W,X,Y,Z)= ∏ M(0,2,3,4,5,6)

= Σm(1,7,8,9,10,11,12,13,14,15)



F(W,X,Y,Z)= W + XYZ + X’Y’Z

I don't care!

- You don't always need all 2^n input combinations in an n-variable function
  - If you can guarantee that certain input combinations never occur
  - If some outputs aren't used in the rest of the circuit
- We mark don't-care outputs in truth tables and K-maps with Xs.

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

- Within a K-map, each X can be considered as either 0 or 1. You should pick the interpretation that allows for the most simplification.

Practice K-map

- Find a MSP for

$$f(w,x,y,z) = \Sigma m(0,2,4,5,8,14,15),\ d(w,x,y,z) = \Sigma m(7,10,13)$$

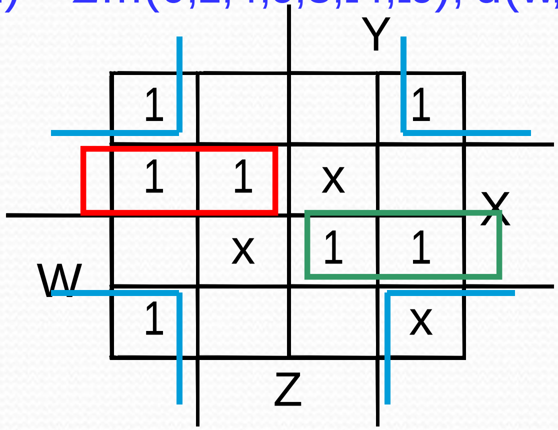
This notation means that input combinations wxyz = 0111, 1010 and 1101 (corresponding to minterms m_7, m_{10} and m_{13}) are unused.

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

Solutions for Practice K-map

- Find a MSP for:

$f(w,x,y,z) = \Sigma m(0,2,4,5,8,14,15)$, $d(w,x,y,z) = \Sigma m(7,10,13)$



f(w,x,y,z)= x'z' + w'xy' + wxy

K-map Summary

- K-maps are an alternative to algebra for simplifying expressions
  - The result is a MSP/MPS, which leads to a minimal two-level circuit
  - It's easy to handle don't-care conditions
  - K-maps are really only good for manual simplification of small expressions...
- Things to keep in mind:
  - Remember the correct order of minterms/maxterms on the K-map
  - When grouping, you can wrap around all sides of the K-map, and your groups can overlap
  - Make as few rectangles as possible, but make each of them as large as possible. This leads to fewer, but simpler, product terms
  - There may be more than one valid solution