Your Name: _____ netid:_____ Group #:
Name: _____ netid:_____
Name: _____ netid:_____
Name: _____ netid:_____
Name: _____ netid:_____

ECE 120 Worksheet 4: Karnaugh Maps

Before you begin today's discussion, be sure that you are familiar with terminology, such as literals, minterms, maxterms, canonical forms, implicants, prime implicants, and K-maps. Also be sure that you know how to find canonical SOP and POS forms for Boolean functions, and that you know how to use K-maps. To check these skills, you can make up a truth table at random, find a Boolean expression for the function, then check your result by writing a truth table for your expression.

K-maps

- Think about *Karnaugh map*, or *K-map*, as an alternative representation of truth table that
  - Lists cells in Gray code order, and
  - Each cell corresponds to a row of the truth table
- Two-variable Boolean function example:
  - four possible values, m_0 - m_4, which can be arranged into a Karnaugh map as follows:

Truth table for a 2-variable function

x	y	f(x,y)
0	0	m_0
0	1	m_1
1	0	m_2
1	1	m_3

Corresponding K-map representation

x \ y	0	1
0	m_0	m_1
1	m_2	m_3

f(x,y)

- Similarly, we can obtain K-maps for 3- and 4-variable Boolean functions
  - *Note the order in which variables x, y, and z appear on the sides of the K-map.*

x \ yz	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

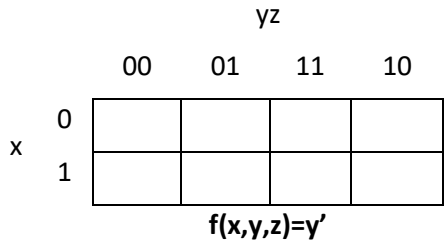
f(x,y,z)

wx \ yz	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

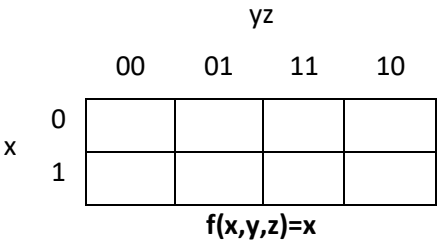
f(w,x,y,z)

Problem 1

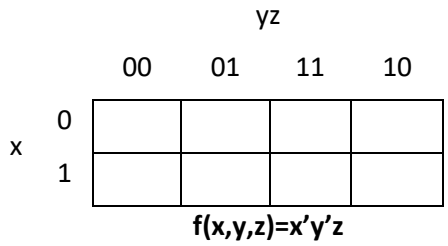
a) Fill in 3-variable K-maps for the functions shown below:



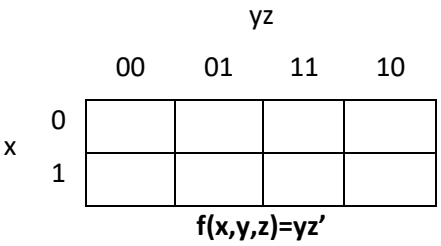
f(x,y,z)=y'



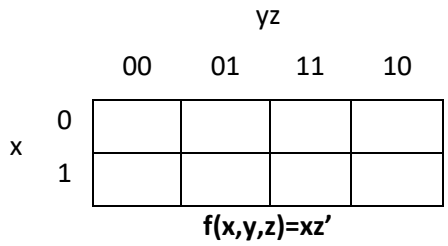
f(x,y,z)=x



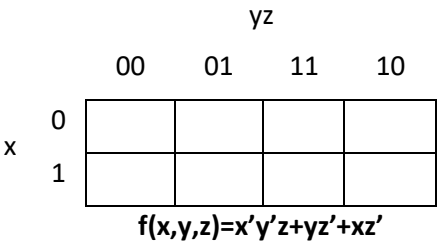
f(x,y,z)=x'y'z



f(x,y,z)=yz'

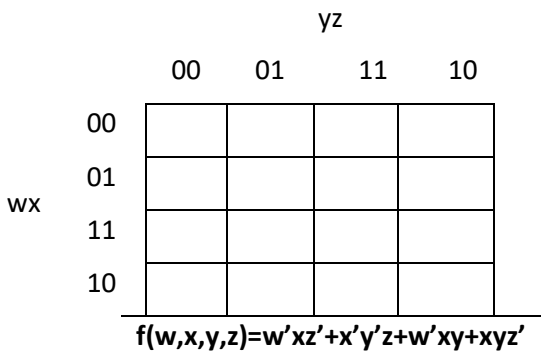


f(x,y,z)=xz'

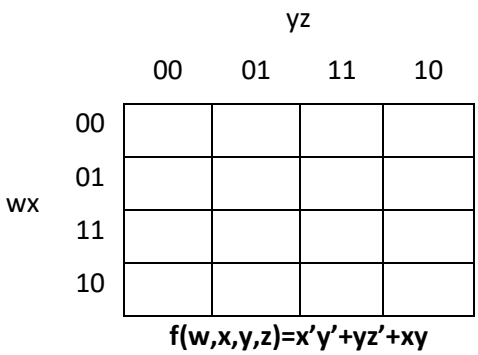


f(x,y,z)=x'y'z+yz'+xz'

b) Fill in 4-variable K-maps for the functions shown below:



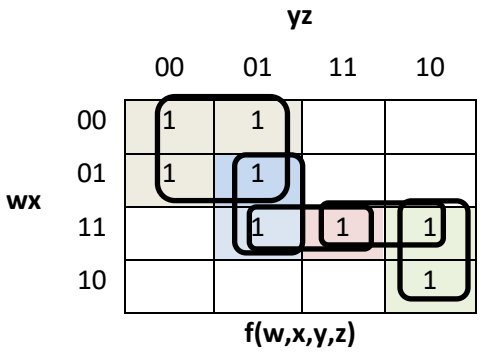
f(w,x,y,z)=w'xz'+x'y'z+w'xy+xyz'



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

Function simplification

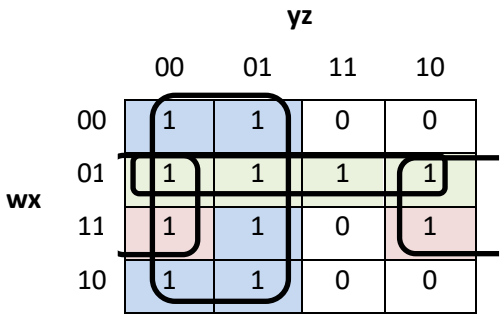
- K-maps is a great tool for simplifying Boolean expressions
- A product term is an *implicant* of a function if the function has the value 1 for all minterms of the product term
    - In terms of K-map, implicants correspond to *all* legal loops
- An implicant is a *prime implicant* if it is not contained within a larger implicant
    - In terms of K-map, prime implicants correspond to *all biggest* possible loops
    - Example of prime imlicants:



Prime implicants: w'y', xy'z, wxz, wyz', wxy

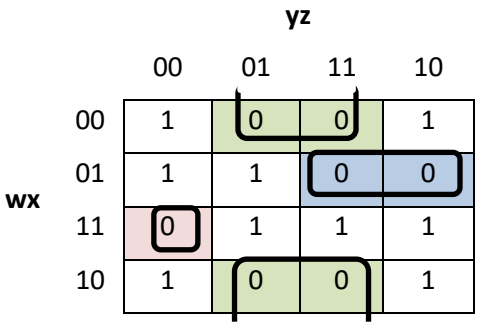
- An SOP (or POS) expression is *minimal* if
    - It has the minimum number of product (sum) terms, and
    - Among expressions with minimum number of terms, it has fewest number of literals
- **A minimal SOP expression is a sum of (a subset of) prime implicants.**

- Minimal SOP example:



Min SOP: y'+w'x+xz'

Minimal POS example:



Min POS:
(w'+x'+y+z)(w+x'+y')(x+z')

Use the area heuristic—number of literals plus the number of operators, not including complemented literals—to find minimal expressions in today's problems. Choosing a minimal number of prime implicants from a K-map will give you minimal solutions in this sense.

Problem 2

a) Find all minimal SOP expressions from the following K-maps. Circle the corresponding loops on the K-maps and write down the Boolean expressions. Note that min SOP may not be unique!

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

Min SOP:

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

Min SOP:

b) Find minimal POS examples from the following K-maps. Circle the corresponding loops on the K-maps and write down the Boolean expressions. Note that min POS may not be unique!

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

Min POS:

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

Min POS:

c) Use the K-map below to calculate both a minimal SOP expression and a minimal POS expression for *F*.

F		***AB***			
		00	**01**	**11**	**10**
CD	**00**	0	0	0	0
	01	1	1	0	0
	11	0	1	0	0
	10	0	0	0	0

Min SOP: F(A,B,C,D) =

Min POS: F(A,B,C,D) =