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
    o Lists cells in Gray code order, and
    o Each cell corresponds to a row of the truth table
- Two-variable Boolean function example:
    o 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

- Similarly, we can obtain K-maps for 3- and 4-variable Boolean functions

Problem 1

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

yz

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

f(x,y,z)=y′

yz

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

f(x,y,z)=x

yz

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

f(x,y,z)=x′y′z

yz

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

f(x,y,z)=yz′

yz

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

f(x,y,z)=xz′

yz

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

f(x,y,z)=x′y′z+yz′+xz′

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

yz

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

f(w,x,y,z)=w′xz′+x′y′z+w′xy+xyz′

yz

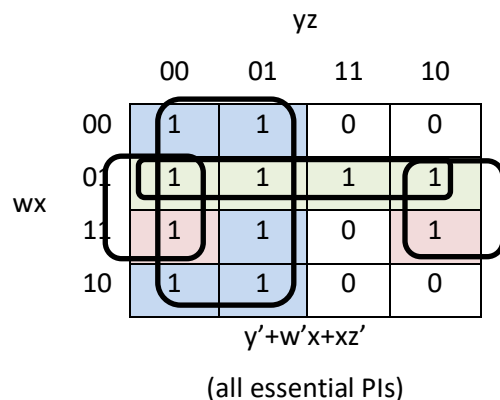| wx \ yz	00	01	11	10
00	1	1	0	1
01	0	0	1	1
11	0	0	1	1
10	1	1	0	1

f(w,x,y,z)=x′y′+yz′+xy
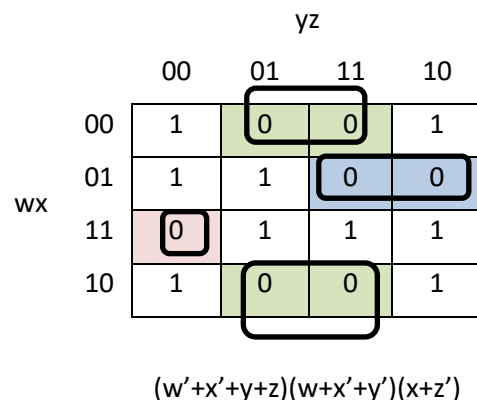
Function simplification

- 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* loops
    - Example:

Prime implicants

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

f(w,x,y,z)

w'y', xy'z, wxz, wyz', wxy
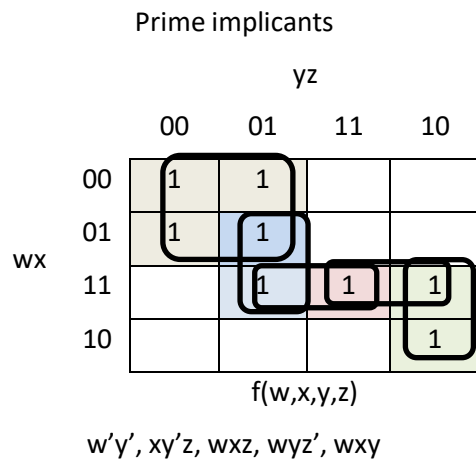
- **An SOP (or POS) expression is *minimal* if**
    - **It has the minimum number of product (sum) terms, and**
    - **Among expressions with minimum number of terms, it has fewest literals**
- A minimal SOP expression is a sum of prime implicants.

- Minimal SOP example: Minimal POS example:

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

y'+w'x+xz'

(all essential PIs)

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

(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!

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

wx

Min SOP:

$x'y+w'z'+wxy'+wxz$

min SOP is not unique, there is another way to cover m_{12}, m_{13}, m_{15}:

$x'y+w'z'+xy'z'+wxz$

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

wx

Min SOP:

$wxy'+w'y'z+wyz+w'xy$

(all essential PIs)

b) Find minimal POS examples from the following K-maps. Circle the corresponding loops on the K-maps and write down the Boolean expressions:

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

wx

Min POS:

$(w'+y'+z')(x+y')$

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

wx

Min POS:

$(w+y+z)(w+x+y') \bullet$
$(w'+y'+z)(w'+x+y)$

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
00	0	0	0	0
01	1	1	0	0
11	0	1	0	0
10	0	0	0	0

CD (rows: 00, 01, 11, 10)

Min SOP: F(A,B,C,D) = **A'BD + A'C'D**

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