

EEL 3701

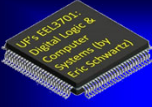
Menu

- K-Maps and Boolean Algebra
 - > K-Maps for MSOP
 - > K-Maps for MPOS
 - > Don't cares
 - > 5 Variable K-maps



University of Florida, EEL 3701 – File 10
© Drs. Schwartz & Arroyo

1



EEL 3701

Karnaugh Maps - Boolean Algebra

- We have discovered that simplification/minimization is an art. If you “see” it, GREAT! Else, work at it, work at it, ..., ...
- A method to exhaustively minimize is due to Quine/McCluskey and is programmable. As the number of variables increase, it gets quite tedious and is not appropriate for our purposes

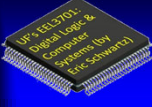
Reminder

▶ $X + X = 1$ and $XY + X/Y = X(Y + /Y) = X$

- Two terms which differ in only one literal can be minimized/ reduced by one literal. A K-Map is a graphical aid (a pattern recognition technique) for humans to implement the above two equations.

University of Florida, EEL 3701 – File 10
© Drs. Schwartz & Arroyo

2

 **EEL 3701**

K-Maps

Consider the following arrangement for f .

A \ BC	00	01	10	11
00	1	0		
01	1	1		
11	1	0		
10	0	1		

Row	A	B	C	f
0	0	0	0	1
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

Positions of Truth Table Rows in K-Map

A \ BC	00	01	10	11
00	R ₀	R ₄		
01	R ₁	R ₅		
11	R ₃	R ₇		
10	R ₂	R ₆		

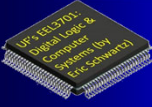
$$f_{SOP} = \overline{A}/\overline{B}/\overline{C} + \overline{A}/\overline{B}C + \overline{A}B\overline{C} + \overline{A}BC + A/\overline{B}\overline{C} + A/\overline{B}C + AB/\overline{C} + ABC$$

But the minterms of function f are now rearranged so they differ in 1 literal.

Look at row 1 (BC=01) of the K-map. The terms of function f differ only in one literal, e.g., f contains $\overline{A}/\overline{B}C + A/\overline{B}C$, the second and fourth terms of the equation, $= (\overline{A} + A)/\overline{B}C = \overline{B}C$. In fact, any two adjacent rows or columns of 1's differ in one literal, and thus, can be simplified.

University of Florida, EEL 3701 – File 10
© Drs. Schwartz & Arroyo

3

 **EEL 3701 K-Maps**

$$f_{SOP} = \overline{A}/\overline{B}/\overline{C} + \overline{A}/\overline{B}C + \overline{A}B\overline{C} + \overline{A}BC + A/\overline{B}\overline{C} + A/\overline{B}C + AB/\overline{C} + ABC$$

- Now, consider the following arrangement for f .

A \ BC	00	01	10	11
00	1	0		
01	1	1		
11	1	0		
10	0	1		

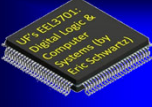
But the minterms of f are now arranged so they differ in 1 literal.

ABC	f
000	1
001	1
010	0
011	1
100	0
101	1
110	1
111	0

Look at the second row of the K-Map. The terms of f differ only in one literal, e.g., f contains $A'B'C + AB'C = (\overline{A} + A)B'C = B'C$. In fact, if any of two rows or columns of 1's differ in one literal, they can be simplified.

University of Florida, EEL 3701 – File 10
© Drs. Schwartz & Arroyo

4



EEL 3701

K-Maps

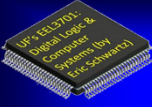
- **Definitions**

Implicants: terms in a K-Map that are 1 iff the function is 1.

Prime implicant: a term in a K-Map that is as big as it can be.
- If we circle in the K-Map those implicants that are logically and spatially adjacent and if we make sure that all the 1's of the function are "covered", we can synthesize the function directly.
- When a 1 is covered by one and only one (prime) implicant, it is called an **essential (prime) implicant**.
- **Goal:**
 - > Step 1. Circle all the (prime) implicants
 - > Step 2. Select a subset that covers the function starting with the prime implicants

University of Florida, EEL 3701 – File 10
© Drs. Schwartz & Arroyo

5



EEL 3701

K-Map: MSOP Example

- <Example 1> Simplify a given function g in SOP form.

$$g = /a /b /c + /a /b c + /a b /c + a b /c + a /b c$$

a	b	c=0	c=1
0	0	1	1
0	1	1	0
1	1	1	0
1	0	0	1

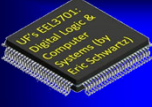
Prime Implicant (because of m_5) $(a + /a) /b c = /b c$

Prime Implicant (because of m_6) $(a + /a) b /c = b /c$

$$g = b /c + /b c + \dots$$

University of Florida, EEL 3701 – File 10
© Drs. Schwartz & Arroyo

6

 **EEL 3701**
K-Map: MSOP Example

- <Example 1> Simplify a given function g in SOP form.

$$g = /a /b /c + /a /b c + /a b /c + a b /c + a /b c$$

$/a /b(c + /c) = /a /b$ $/a /c(b + /b) = /a /c$

a	b	c=0	c=1
0	0	1	1
0	1	1	0
1	1	1	0
1	0	0	1

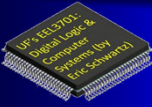
$g_{MSOP1} = b/c + /b c + /a /b$
 $= /a /b + /b c + b/c$

a	b	c=0	c=1
0	0	1	1
0	1	1	0
1	1	1	0
1	0	0	1

$g_{MSOP2} = b/c + /b c + /a /c$
 $= /a /c + /b c + b/c$

University of Florida, EEL 3701 – File 10
© Drs. Schwartz & Arroyo

7

 **EEL 3701**
K-Map: Another MSOP Viewpoint

- Another way to look at it

$$g = /a /b /c + /a /b c + /a b /c + a b /c + a /b c$$

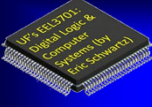
$g_{MSOP1} = b/c + /b c + /a /b$
 $= /a /b + /b c + b/c$

$g_{MSOP2} = b/c + /b c + /a /c$
 $= /a /c + /b c + b/c$

ab		c	
		0	1
a	00	1	1
	01	1	0
	11	1	0
	10	0	1

University of Florida, EEL 3701 – File 10
© Drs. Schwartz & Arroyo

8

 **EEL 3701**

K-Map: MPOS Example

<Example 2> Will it work for POS if we use 0's & maxterms? YES

NOTE: Reversed labels from SOP

ab \ c	0	1
00	1	1
01	1	0
11	1	0
10	0	1

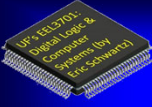
$/b + /c$

$/a + b + c$

$\mathbf{g_{MPOS}} = (/a + b + c)(/b + /c)$

University of Florida, EEL 3701 – File 10
© Drs. Schwartz & Arroyo

9

 **EEL 3701**

K-Map: MSOP vs MPOS

- Let's compare MSOP with MPOS.

$\mathbf{g_{MSOP}} = /a /c + /b c + b /c$ $\mathbf{g_{MPOS}} = (/a + b + c)(/b + /c)$

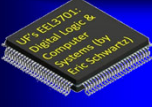
- Q: Which is cheaper? Ans: Depends on cost function; probably MPOS
- Q: Why? Ans: Because MPOS requires less # of gates and inputs.

$\mathbf{g_{MSOP}}$ requires 4 gates and 9 inputs

$\mathbf{g_{MPOS}}$ requires 3 gates and 7 inputs

University of Florida, EEL 3701 – File 10
© Drs. Schwartz & Arroyo

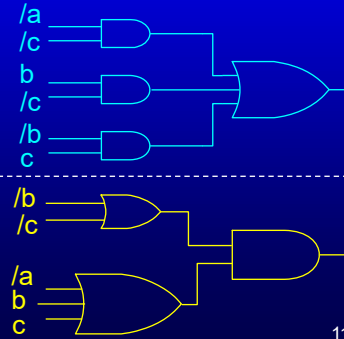
10

 **EEL 3701**
K-Map Heuristic

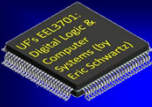
- **Heuristic:** (not a guarantee) We note that the POS was minimum (compared to the SOP) perhaps because we had only three 0's vs five 1's. In general, we synthesize the function based on the least number of either 1's or 0's. But to be sure we **must do both and check!**
- **Observation:** We got 2-level logic with these K-maps, i.e., the maximum delay is the delay of two gates. It is sometimes possible to obtain a cheaper (in gates) cost function using more than 2-level logic. If your application can tolerate larger delays, this is OK.

 **NOTE:** 2-Level logic is the fastest logic possible and K-Maps give it to you.

University of Florida, EEL 3701 – File 10
© Drs. Schwartz & Arroyo



11

 **EEL 3701**
**4-input K-Map Example:
Groups of 2**

- Examples of K-maps

	cd			
ab	00	01	11	10
00	1			
01				
11	1			
10	1	1		1

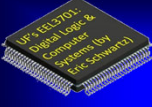
$\Rightarrow g_{1MSOP} = a/b/c + a/b/d + a/c/d + /b/c/d$

Legend:

- /b /c /d prime
- a /b /d prime
- - - a /c /d prime
- a /b /c prime

University of Florida, EEL 3701 – File 10
© Drs. Schwartz & Arroyo

12

 **EEL 3701**
**4-input K-Map Example:
 Groups of 2 and Redundancy**

- Examples of K-maps (continued)

Redundant!
 What is eqn?

cd \ ab	00	01	11	10
00		1		
01		1	1	1
11	1	1	1	
10			1	1

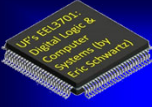
g_2

— a b /c prime
 - - /a /c d prime
 /a b c prime
 - - a c d prime

$\Rightarrow g_{2MSOP} = /a b c + /a /c d + a b /c + a c d$

University of Florida, EEL 3701 – File 10
 © Drs. Schwartz & Arroyo

13

 **EEL 3701**
**4-input K-Map Example:
 Groups of 4**

- Examples of K-maps (continued)

cd \ ab	00	01	11	10
00				
01	1	1	1	1
11		1	1	
10				

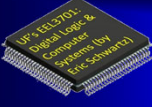
g_3

— /a b prime
 - - b d prime

$\Rightarrow g_{3MSOP} = /a b + b d$

University of Florida, EEL 3701 – File 10
 © Drs. Schwartz & Arroyo

14

 **EEL 3701** 4-input K-Map Example: More Groups of 4

- Examples of K-maps (continued)

	cd			
ab	00	01	11	10
00		1	1	
01	1			1
11	1			1
10		1	1	

g_4

— /b d prime
 - - b /d prime
 $\Rightarrow g_{4MSOP} = /b d + b /d = b \text{ xor } d$

	cd			
ab	00	01	11	10
00	0			0
01		0	0	
11		0	0	
10	0			0

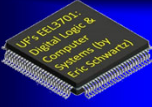
g_4

— /b + /d prime
 - - b + d prime
 $\Rightarrow g_{4MPOS} = (/b + /d) (b + d) = b \text{ xor } d$

Same function!

University of Florida, EEL 3701 – File 10
© Drs. Schwartz & Arroyo

15

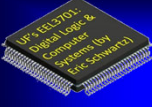
 **EEL 3701** Don't Cares

- Don't Cares:** Suppose we have the following system

- What if some combinations of the inputs {a,b,c,d} are never produced by C_i ?
 - >Q: What should we write in the Truth Table, 0 or 1?
 - >A: We write a don't care (X). We are free to choose X to be 0 or 1 at our convenience. However, once you pick a particular value for X, it must be the same throughout the design!

University of Florida, EEL 3701 – File 10
© Drs. Schwartz & Arroyo

16

 **EEL 3701**

Don't Care Examples

- Examples of K-maps with don't cares

	cd	00	01	11	10
ab	00	0	0	1	1
01	1	X	0	1	
11	1	X	0	0	
10	0	0	X	0	

g_1

— $b\bar{c}$ prime

-- $\bar{a}\bar{b}c$ prime

..... $\bar{a}c\bar{d}$ prime

$\therefore g_{1MSOP} = \bar{a}\bar{b}c + \bar{a}c\bar{d} + b\bar{c}$

	cd	00	01	11	10
ab	00	0	0	1	1
01	1	X	0	1	
11	1	X	0	0	
10	0	0	X	0	

g_2

— $b+c$ prime

-- $\bar{b}+\bar{d}$ prime

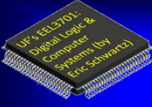
..... $\bar{a}+\bar{c}$ prime

$\therefore g_{2MPOS} = (\bar{a}+\bar{c})(\bar{b}+\bar{d})(b+c)$

Is $g_{1MSOP} = g_{2MPOS}$? Why or why not?

University of Florida, EEL 3701 – File 10
© Drs. Schwartz & Arroyo

17

 **EEL 3701**

5-variable K-Maps

- Five Variable K-maps:** Break the function into two 4-variable K-maps, e.g., for a function $f(a,b,c,d,e)$ of 5 variables, pick one of the variables, say e , and do a K-map when $e=0$ and when $e=1$. Then $f = e' \cdot f_{e=0} + e \cdot f_{e=1}$.

	cd	00	01	11	10
ab	00		1	1	
01					1
11					1
10		1	1		

$f_{e=0}$

	cd	00	01	11	10
ab	00				
01	1	1			1
11					1
10		1			

$f_{e=1}$

Can simplify to $ab'c'd$

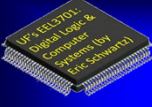
$f = e' \cdot f_{e=0} + e \cdot f_{e=1} = e'(b'd + bcd') + e(a'bc' + ab'c'd + bcd')$

$= ab'c'd e + bcd' + e'(b'd) + e(a'bc')$

Not in lexical order

University of Florida, EEL 3701 – File 10
© Drs. Schwartz & Arroyo

18

 **EEL 3701**

5-variable K-Maps

- Or use:
 - $f = f_{e \& e'} + e' \cdot f_{e=0} \cdot (\text{not } f_{e=1}) + e \cdot f_{e=1} \cdot (\text{not } f_{e=0})$, i.e.,
 - $f = (\text{shared covers}) + e' \cdot (\text{covers in } f_{e=0} \text{ only}) + e \cdot (\text{covers in } f_{e=1} \text{ only})$

	cd			
ab	00	01	11	10
00		1	1	
01				1
11				1
10		1	1	

	cd			
ab	00	01	11	10
00				
01	1	1		1
11				1
10		1		

$f_{e=0}$ $f_{e=1}$

$$f = ab'c'd + bcd' + e'(b'd) + e(a'bc')$$

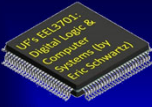
$$= f_{e \& e'} + e' \cdot /f_{e=1} \cdot f_{e=0} + e \cdot /f_{e=0} \cdot f_{e=1}$$

Not in lexical order

University of Florida, EEL 3701 – File 10
© Drs. Schwartz & Arroyo

19

19

 **EEL 3701**

The End!

University of Florida, EEL 3701 – File 10
© Drs. Schwartz & Arroyo

20

20