BROWN
School of Engineering

DIGITAL ELECTRONICS SYSTEM DESIGN

FALL 2019

PROFS. IRIS BAHAR

SEPTEMBER 11, 2019

LECTURE 3: KARNAUGH MAPS & LOGIC MINIMIZATION

LECTURE SLIDES

- I will post lecture slides on the course webpage within 2 days of class
- Access the lecture slides from:

www.brown.edu/Departments/Engineering/Courses/En163/home.html

- Updated syllabus and other course materials are also available online

LABORATORY ASSIGNMENTS

- Lab kits and lab manuals are now ready for pickup
  - See George Worth in B&H325 for pickup (remember your $60 --- check or cash)
- Lab manuals can also be found on the course webpage
- TA will start holding lab hours on Thursday
  - Schedule can be found on the course webpage
- We created a Piazza page as a discussion forum for issues/questions on lab assignments.
  - Join by going to: piazza.com/brown/fall2019/engn1630

LAB TUTORIAL

- Andrew Duncombe (one of our TAs) will be holding a tutorial session tomorrow (Thursday, Sept. 12) @ 5pm in the lab to go over the basics of building digital electronics
  - How to use a breadboard
  - How to wire up a breadboard
  - How to connect to a power source
  - Etc.
- Lab: B&H196

BOOLEAN AXIOMS

	Axiom		Dual	Name
A1	$B = 0$ if $B \neq 1$	A1′	$B = 1$ if $B \neq 0$	Binary field
A2	$\overline{0} = 1$	A2′	$\overline{1} = 0$	NOT
A3	$0 \bullet 0 = 0$	A3′	$1 + 1 = 1$	AND/OR
A4	$1 \bullet 1 = 1$	A4′	$0 + 0 = 0$	AND/OR
A5	$0 \bullet 1 = 1 \bullet 0 = 0$	A5′	$1 + 0 = 0 + 1 = 1$	AND/OR

	Theorem		Dual	Name
T1	$B \bullet 1 = B$	T1′	$B + 0 = B$	Identity
T2	$B \bullet 0 = 0$	T2′	$B + 1 = 1$	Null Element
T3	$B \bullet B = B$	T3′	$B + B = B$	Idempotency
T4		$\overline{\overline{B}} = B$		Involution
T5	$B \bullet \overline{B} = 0$	T5′	$B + \overline{B} = 1$	Complements

BOOLEAN THEOREMS OF SEVERAL VARIABLES

	Theorem		Dual	Name
T6	$B \bullet C = C \bullet B$	T6′	$B + C = C + B$	Commutativity
T7	$(B \bullet C) \bullet D = B \bullet (C \bullet D)$	T7′	$(B + C) + D = B + (C + D)$	Associativity
T8	$(B \bullet C) + B \bullet D = B \bullet (C + D)$	T8′	$(B + C) \bullet (B + D) = B + (C \bullet D)$	Distributivity
T9	$B \bullet (B + C) = B$	T9′	$B + (B \bullet C) = B$	Covering
T10	$(B \bullet C) + (B \bullet \overline{C}) = B$	T10′	$(B + C) \bullet (B + \overline{C}) = B$	Combining
T11	$(B \bullet C) + (\overline{B} \bullet D) + (C \bullet D)$ $= B \bullet C + \overline{B} \bullet D$	T11′	$(B + C) \bullet (\overline{B} + D) \bullet (C + D)$ $= (B + C) \bullet (\overline{B} + D)$	Consensus
T12	$\overline{B_0 \bullet B_1 \bullet B_2 \ldots}$ $= (\overline{B_0} + \overline{B_1} + \overline{B_2} \ldots)$	T12′	$\overline{B_0 + B_1 + B_2 \ldots}$ $= (\overline{B_0} \bullet \overline{B_1} \bullet \overline{B_2})$	De Morgan's Theorem

CANONICAL FORM

- Canonical form: a unique representation for a Boolean function
  - Given a fixed ordering of the input variables
  - Two equivalent functions have the same canonical form
  - Examples: truth table, canonical sum, canonical product

DEFINITIONS

- Literals x_i or x_i'
- Product Term x_2x_1'
- Sum Term $x_2 + x_1' + x_0$
- Minterm of *n* variables: A product of *n* variables in which every variable appears exactly once.
- Maxterm of *n* variables: A sum of *n* variables in which every variable appears exactly once.

8

MINTERM AND MAXTERM

Id	a	b	c_{in}	carryout	minterm	maxterm
0	0	0	0	0		a+b+c
1	0	0	1	0		a+b+c'
2	0	1	0	0		a+b'+c
3	0	1	1	1	a' b c	
4	1	0	0	0		a'+b+c
5	1	0	1	1	a b'c	
6	1	1	0	1	a b c'	
7	1	1	1	1	a b c	

Note that the literals are complemented for MAXTERMS

MINIMIZING A FUNCTION

- Note that a canonical representation of a function is not always the most compact way to represent a function
  - To implement a function in hardware we want to minimize the number of literals and number of terms
  - We can apply Boolean Theorems to minimize a function, but this can be cumbersome and lacks a visual understanding
  - Instead, represent the function using Karnaugh maps…

REPRESENTING FUNCTIONS WITH KARNAUGH MAPS

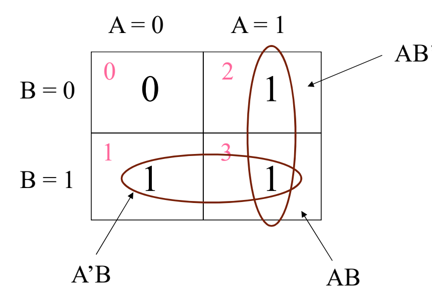
Id	A	B	f (A, B)
0	0	0	m_0
1	0	1	m_1
2	1	0	m_2
3	1	1	m_3

	A = 0	A = 1
B = 0	0 $\overline{A}\,\overline{B}$	2 $A\overline{B}$
B = 1	1 $\overline{A}B$	3 AB

- How do we represent f(A,B) = A+B on the K-map

On the K-map:

Id	A	B	f (A, B)
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	1

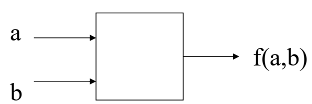


f(A,B) = A + B

TWO VARIABLE K-MAPS

Id	a	b	f (a, b)
0	0	0	f (0, 0)
1	0	1	f (0, 1)
2	1	0	f (1, 0)
3	1	1	f (1, 1)

2 variables means we have 2^2 entries and thus we have 2 to the 2^2, or 2^4 possible functions for 2 bits, which is 16.



THREE VARIABLES K-MAPS

Id	a	b	c	f (a,b,c)
0	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	0

- What does the k-map look like?

CORRESPONDING K-MAP

Gray code

b = 1

	(0,0)	(0,1)	(1,1)	(1,0)
c = 0	0: 1	2: 1	6: 1	4: 1
c = 1	1: 0	3: 0	7: 0	5: 0

a = 1

f(a,b,c) = c'

This is a minimum cover for this function

KARNAUGH MAPS (K-MAPS)

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Y: C \ AB	00	01	11	10
0	1	0	0	0
1	1	0	0	0

Y: C \ AB	00	01	11	10
0	$\bar{A}\bar{B}\bar{C}$	$\bar{A}B\bar{C}$	$AB\bar{C}$	$A\bar{B}\bar{C}$
1	$\bar{A}\bar{B}C$	$\bar{A}BC$	ABC	$A\bar{B}C$

- Boolean expressions can be minimized by combining terms
- Karnaugh maps (K-maps) minimize equations graphically
- $PA + P\bar{A} = P$

K-MAP

- Circle 1's in adjacent squares
- In the Boolean expression, include only the literals that are true for each square

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Y: C \ AB	00	01	11	10
0	1	0	0	0
1	1	0	0	0

y(A,B)=A'B'

MINTERM AND MAXTERM

Id	a	b	c_{in}	carryout	minterm	maxterm
0	0	0	0	0		a+b+c
1	0	0	1	0		a+b+c'
2	0	1	0	0		a+b'+c
3	0	1	1	1	a' b c	
4	1	0	0	0		a'+b+c
5	1	0	1	1	a b'c	
6	1	1	0	1	a b c'	
7	1	1	1	1	a b c	

maxterm

minterm

- What does the K-map look like?

ANOTHER 3-INPUT EXAMPLE

Id	a	b	c	f (a,b,c)
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	-
7	1	1	1	1

22

CORRESPONDING K-MAP

b = 1

	(0,0)	(0,1)	(1,1)	(1,0)
c = 0	[0] 0	[2] 1	[6] -	[4] 1
c = 1	[1] 0	[3] 0	[7] 1	[5] 1

a = 1

f(a,b,c) = a + bc'

Adding the don't care term to the ON set allows for greater reduction in the functional representation (fewer literals and product terms)

YET ANOTHER EXAMPLE

Id	a	b	c	f (a,b,c,d)
0	0	0	0	1
1	0	0	1	1
2	0	1	0	-
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	0
7	1	1	1	0

24

CORRESPONDING K-MAP

b = 1

	(0,0)	(0,1)	(1,1)	(1,0)
c = 0	0: 1	2: -	6: 0	4: 1
c = 1	1: 1	3: 0	7: 0	5: 1

a = 1

f(a,b,c) = b'

4-INPUT K-MAP

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Y

CD \ AB	00	01	11	10
00				
01				
11				
10				

4-INPUT K-MAP

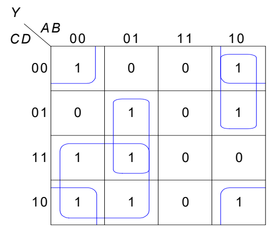
A	B	C	D	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Y

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

4-INPUT K-MAP

A	*B*	*C*	*D*	*Y*
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0



$Y = \overline{A}C + \overline{A}BD + A\overline{B}\,\overline{C} + \overline{B}\,\overline{D}$