DIGITAL ELECTRONICS SYSTEM DESIGN

BROWN
School of Engineering

FALL 2019

PROF. IRIS BAHAR

SEPTEMBER 9, 2019

LECTURE 2: BOOLEAN ALGEBRA

INSTRUCTORS & TAS

- Iris Bahar
  - Office: CIT449
  - Office hours: Mon. 4:30-5:30pm, Tues. 10-11am
- Graduate TAs: Jiwon Choe and Pratistha Shakya
- Undergrad TAs: Monica Alves, Jarod Boone, McKenna Cisler, Eren Derman, Andrew Duncombe

GRADING

- Grade distribution:
  - Laboratory assignments: 55%
  - Midterm exam: 15%
  - Final exam: 25%
  - Class participation: 5%

Updated from syllabus handed out last week

- Labs are graded pass/fail and are worth a variable number of points depending on difficulty
- You need 57 out of a possible 101 points on the lab portion to pass the class
- You must receive a passing grade on both lab and exams portions of the course to get a passing grade

LAB MANUAL, KITS

- Lab manual and kits will be ready tomorrow
  - Pick up both from George Worth (B&H 325)
  - $60 (payable by check to Brown University, cash OK)
  - $50 rebate at end of semester if you return the major parts of the kit
- *Course webpage:*

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

BOOLEAN ALGEBRA AND FUNCTION REPRESENTATIONS

BOOLEAN ALGEBRA

- Set of axioms and theorems to simplify Boolean equations
- Like regular algebra, but in some cases simpler because variables can have only two values (1 or 0)
- Axioms and theorems obey the principles of duality:
  - ANDs and ORs interchanged, 0's and 1's interchanged

BOOLEAN AXIOMS AND THEOREMS OF SINGLE VARIABLES

	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

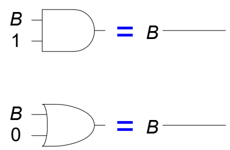
T1: IDENTITY THEOREM

- $B \bullet 1 =$
- $B + 0 =$

1-<8>

T1: IDENTITY THEOREM

- B • 1 = B
- B + 0 = B

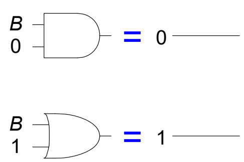


T2: NULL ELEMENT THEOREM

- B • 0 =
- B + 1 =

T2: NULL ELEMENT THEOREM

- B • 0 = 0
- B + 1 = 1



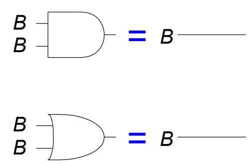
T3: IDEMPOTENCY THEOREM

- B • B =
- B + B =

1-<12>

T3: IDEMPOTENCY THEOREM

- $B \cdot B = B$
- $B + B = B$



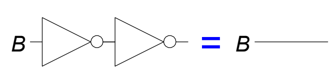
T4: IDENTITY THEOREM

- $\overline{\overline{B}} =$

1-<14>

T4: IDENTITY THEOREM

- $\overline{\overline{B}} = B$

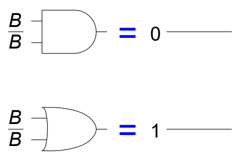


T5: COMPLEMENT THEOREM

- $B \cdot \overline{B} =$
- $B + \overline{B} =$

T5: COMPLEMENT THEOREM

- $B \bullet \overline{B} = 0$
- $B + \overline{B} = 1$



BOOLEAN THEOREMS: SUMMARY

	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

REVIEW OF BOOLEAN ALGEBRA AND SWITCHING FUNCTIONS

AND, OR, NOT

AND

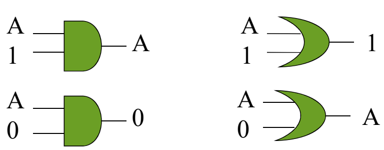
A B	C
0 0	0
0 1	0
1 0	0
1 1	1

OR

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

NOT

A	C
0	1
1	0



0 dominates in AND 1 dominates in OR

BOOLEAN THEOREMS OF SEVERAL VARIABLES

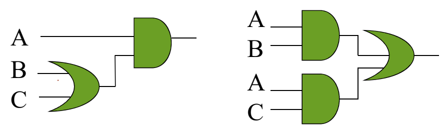
	Theorem		Dual	Name
T6	$B \bullet C = C \bullet B$	T6′	$B + C = C + B$	Commutativity
T7	$(B \bullet C) \bullet D = B \bullet (C \bullet D)$	T7′	$(B + C) + D = B + (C + D)$	Associativity
T8	$(B \bullet C) + B \bullet D = B \bullet (C + D)$	T8′	$(B + C) \bullet (B + D) = B + (C \bullet D)$	Distributivity
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

1. Identity A * 1 = A A + 1 = 1
A * 0 = 0 A + 0 = A

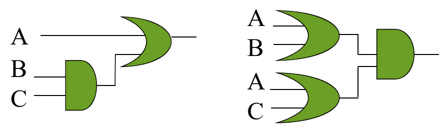
2. Complement A + A’ = 1 A * A’ = 0

T8. Distributivity Law

A(B+C) = AB + AC

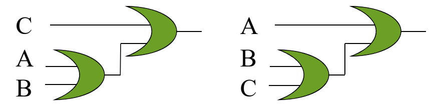


A+BC = (A+B)(A+C)

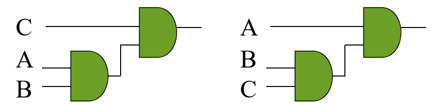


T7. Associativity

(A+B) + C = A + (B+C)



(AB)C = A(BC)



T12. DeMorgan’s Law (A+B)’ = A’B’ (AB)’ = A’ + B’

T11. AC + AB + BC’ = AC + BC’

SIMPLIFYING BOOLEAN EXPRESSIONS: EXAMPLE 1

- $Y = \overline{A}B + AB$

1-<23>

SIMPLIFYING BOOLEAN EXPRESSIONS: EXAMPLE 1

- $Y = \overline{A}B + AB$

  $= B(\overline{A} + A)$ T8

  $= B(1)$ T5’

  $= B$ T1

1-<24>

SIMPLIFYING BOOLEAN EXPRESSIONS: EXAMPLE 2

- $Y = A(AB + ABC)$

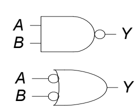
SIMPLIFYING BOOLEAN EXPRESSIONS: EXAMPLE 2

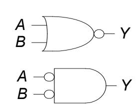
- $Y = A(AB + ABC)$

$= A(AB(1 + C))$	T8
$= A(AB(1))$	T2'
$= A(AB)$	T1
$= (AA)B$	T7
$= AB$	T3

DEMORGAN'S THEOREM

- $Y = \overline{AB} = \overline{A} + \overline{B}$



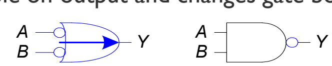


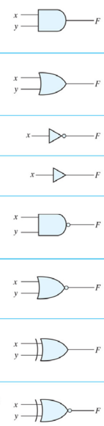
- $Y = \overline{A + B} = \overline{A} \cdot \overline{B}$

BUBBLE PUSHING

- Pushing bubbles backward (from output) or forward (from inputs) changes body of gate from AND to OR or vice versa.
- Pushing bubble from output back to inputs puts bubbles on all gate inputs.

- Pushing bubbles on *all* gate inputs forward toward output puts bubble on output and changes gate body.





Name	Graphic symbol	Algebraic function	Truth table (x y | F)
AND		$F = x \cdot y$	0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR		$F = x + y$	0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
Inverter		$F = x'$	x | F 0 | 1 1 | 0
Buffer		$F = x$	x | F 0 | 0 1 | 1
NAND		$F = (xy)'$	0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0
NOR		$F = (x + y)'$	0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0
Exclusive-OR (XOR)		$F = xy' + x'y = x \oplus y$	0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0
Exclusive-NOR or equivalence		$F = xy + x'y' = (x \oplus y)'$	0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1

OPERATOR PRECEDENCE

- What does $w \bullet x' + y \bullet z$ mean?
- Operator precedence rules:
  1. NOT (highest priority)
  2. AND
  3. OR (lowest priority)

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.

32

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

MINTERMS

$f_1(a,b,c) = a'bc + ab'c + abc' + abc$

$a'bc = 1$ iff $(a,b,c,) = (0,1,1)$
$ab'c = 1$ iff $(a,b,c,) = (1,0,1)$
$abc' = 1$ iff $(a,b,c,) = (1,1,0)$
$abc\ = 1$ iff $(a,b,c,) = (1,1,1)$

$f_1(a,b,c) = 1$ iff $(a,b,c) = (0,1,1), (1,0,1), (1,1,0)$, or $(1,1,1)$

Ex: $f_1(1,0,1) = 1'01 + 10'1 + 101' + 101 = 1$
$f_1(1,0,0) = 1'00 + 10'0 + 100' + 100 = 0$

MAXTERMS

$f_2(a,b,c) = (a+b+c)(a+b+c')(a+b'+c)(a'+b'+c)$

$a + b + c\ = 0$ iff $(a,b,c,) = (0,0,0)$
$a + b + c' = 0$ iff $(a,b,c,) = (0,0,1)$
$a + b' + c = 0$ iff $(a,b,c,) = (0,1,0)$
$a' + b + c = 0$ iff $(a,b,c,) = (1,0,0)$

$f_2(a,b,c) = 0$ iff $(a,b,c) = (0,0,0), (0,0,1), (0,1,0), (1,0,0)$

Ex: $f_2(1,0,1) = (1+0+1)(1+0+1')(1+0'+1)(1'+0+1) = 1$
$f_2(0,1,0) = (0+1+0)(0+1+0')(0+1'+0)(0'+1+0) = 0$

MINTERM VS. MAXTERM

$f_1(a,b,c) = a'bc + ab'c + abc' + abc$
$f_2(a,b,c) = (a+b+c)(a+b+c')(a+b'+c)(a'+b'+c)$

$f_1(a, b, c) = m_3 + m_5 + m_6 + m_7 = \Sigma m(3,5,6,7)$
$f_2(a, b, c) = M_0M_1M_2M_4 = \Pi M(0, 1, 2, 4)$

Question: Does $f_1 = f_2$?
Yes! Prove using Boolean algebra.

36

INCOMPLETELY SPECIFIED FUNCTIONS

Don't care set is important because it allows us to minimize the function

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

1) The input does not happen.
2) The input happens, but the output is ignored.

Examples:

- Decimal number 0… 9 uses 4 bits. Minterm (1,1,1,1) does not happen.
- Final carry out bit (output is ignored).

INCOMPLETELY SPECIFIED FUNCTION

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

$g_1(a,b,c)$ canonical form w/ minterms

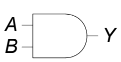
$g_2(a,b,c)$ canonical form w/ maxterms

Does $g_1(a,b,c) = g_2(a,b,c)$?

No! Because there is a group of Don't Care set. (g_1 only covers the onset, g_2 only covers the offset).

TWO-INPUT LOGIC GATES

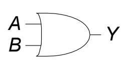
AND



$Y = AB$

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

OR

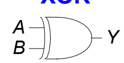


$Y = A + B$

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

MORE TWO-INPUT LOGIC GATES

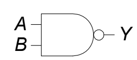
XOR



$Y = A \oplus B$

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

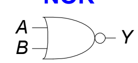
NAND



$Y = \overline{AB}$

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

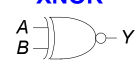
NOR



$Y = \overline{A + B}$

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

XNOR



$Y = \overline{A \oplus B}$

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

Function can be represented by sum of minterms:

f(A,B) = A'B+AB'+AB

This is not optimal however!

We want to *minimize the number of literals and terms*.

We factor out common terms –

A'B+AB'+AB= A'B+AB'+AB+AB

=(A'+A)B+A(B'+B)=B+A

f(A,B) = A+B