

# CSE 311 Foundations of Computing I

Lecture 4, Boolean Algebra and Circuits  
Spring 2013

1

## Announcements

- Reading assignments
  - Boolean Algebra
    - 12.1 – 12.3 7<sup>th</sup> Edition
    - 11.1 – 11.3 6<sup>th</sup> Edition
  - For next time: Predicates and Quantifiers
    - 1.4 7<sup>th</sup> Edition
    - 1.3 6<sup>th</sup> Edition

3

## Administrative

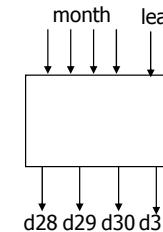
- Course web:  
<http://www.cs.washington.edu/311>
  - Homework, Lecture slides, Office Hours, ...
  - Worksheets from Section, too
- Homework:
  - Due Wednesday at the start of class

2

## Highlights from last lecture

### Calendar: Number of Days in a Month

- Encoding:
  - how many bits for each input/output?
  - binary number for month
  - four wires for 28, 29, 30, and 31



month	leap	d28	d29	d30	d31
0000	–	–	–	–	–
0001	–	0	0	0	1
0010	0	1	0	0	0
0010	1	0	1	0	0
0011	–	0	0	0	1
0100	–	0	0	1	0
0101	–	0	0	0	1
0110	–	0	0	1	0
0111	–	0	0	0	1
1000	–	0	0	0	1
1001	–	0	0	1	0
1010	–	0	0	0	1
1011	–	0	0	1	0
1100	–	0	0	0	1
1101	–	–	–	–	–
1110	–	–	–	–	–
1111	–	–	–	–	–

4

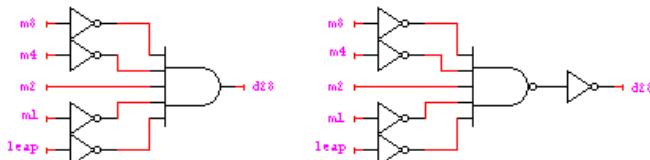
# Highlights from last lecture

## Calendar: Number of Days in a Month

$$d28 = m8' \cdot m4' \cdot m2 \cdot m1' \cdot \text{leap}'$$

$$d29 = m8' \cdot m4' \cdot m2 \cdot m1' \cdot \text{leap}$$

$$\begin{aligned} d30 &= (m8' \cdot m4 \cdot m2' \cdot m1') + (m8' \cdot m4 \cdot m2 \cdot m1') + \\ &\quad (m8 \cdot m4' \cdot m2' \cdot m1) + (m8 \cdot m4' \cdot m2 \cdot m1) \\ &= (m8' \cdot m4 \cdot m1') + (m8 \cdot m4' \cdot m1) \\ d31 &= (m8' \cdot m4 \cdot m2' \cdot m1) + (m8' \cdot m4 \cdot m2 \cdot m1) + \\ &\quad (m8 \cdot m4 \cdot m2' \cdot m1) + (m8 \cdot m4 \cdot m2 \cdot m1) + \\ &\quad (m8 \cdot m4' \cdot m2' \cdot m1') + (m8 \cdot m4' \cdot m2 \cdot m1') + \\ &\quad (m8 \cdot m4 \cdot m2' \cdot m1') \end{aligned}$$



5

# Axioms and theorems of Boolean algebra

### identity

$$1. X + 0 = X$$

$$1D. X \cdot 1 = X$$

### null

$$2. X + 1 = 1$$

$$2D. X \cdot 0 = 0$$

### idempotency:

$$3. X + X = X$$

$$3D. X \cdot X = X$$

### involution:

$$4. (X')' = X$$

### complementarity:

$$5. X + X' = 1$$

$$5D. X \cdot X' = 0$$

### commutativity:

$$6. X \cdot Y = Y \cdot X$$

$$6D. X \cdot Y = Y \cdot X$$

### associativity:

$$7. (X + Y) + Z = X + (Y + Z)$$

$$7D. (X \cdot Y) \cdot Z = X \cdot (Y \cdot Z)$$

### distributivity:

$$8. X \cdot (Y + Z) = (X \cdot Y) + (X \cdot Z)$$

$$8D. X + (Y \cdot Z) = (X + Y) \cdot (X + Z)$$

# Highlights from last lecture

- Boolean algebra to circuit design

- Boolean algebra

  - a set of elements  $B = \{0, 1\}$

  - binary operations  $\{+, \cdot\}$

  - and a unary operation  $\{\prime\}$

  - such that the following axioms hold:

1. the set  $B$  contains at least two elements:  $a, b$

2. closure:  $a + b$  is in  $B$

3. commutativity:  $a + b = b + a$

4. associativity:  $a + (b + c) = (a + b) + c$

5. identity:  $a + 0 = a$

6. distributivity:  $a + (b \cdot c) = (a + b) \cdot (a + c)$

$a \cdot b$  is in  $B$

$a \cdot b = b \cdot a$

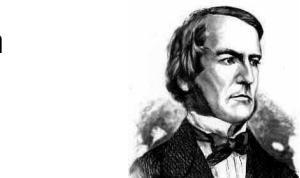
$a \cdot (b \cdot c) = (a \cdot b) \cdot c$

$a \cdot 1 = a$

$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

$a + a' = 1$

$a \cdot a' = 0$



George Boole – 1854

6

# Axioms and theorems of Boolean algebra (cont.)

### uniting:

$$9. X \cdot Y + X \cdot Y' = X$$

$$9D. (X + Y) \cdot (X + Y') = X$$

### absorption:

$$10. X + X \cdot Y = X$$

$$10D. X \cdot (X + Y) = X$$

$$11. (X + Y') \cdot Y = X \cdot Y$$

$$11D. (X \cdot Y') + Y = X + Y$$

### factoring:

$$12. (X + Y) \cdot (X' + Z) = X \cdot Z + X' \cdot Y$$

$$12D. X \cdot Y + X' \cdot Z = (X + Z) \cdot (X' + Y)$$

### consensus:

$$13. (X \cdot Y) + (Y \cdot Z) + (X' \cdot Z) = X \cdot Y + X' \cdot Z$$

$$13D. (X + Y) \cdot (Y + Z) \cdot (X' + Z) = (X + Y) \cdot (X' + Z)$$

### de Morgan's:

$$14. (X + Y + \dots)' = X' \cdot Y' \cdot \dots$$

$$14D. (X \cdot Y \cdot \dots)' = X' + Y' + \dots$$

7

8

# Proving theorems (rewriting)

- Using the laws of Boolean algebra:
  - e.g., prove the theorem:

$$X \cdot Y + X \cdot Y' = X$$

distributivity (8)

$$X \cdot Y + X \cdot Y' = X \cdot (Y + Y')$$

complementarity (5)

$$= X \cdot (1)$$

identity (1D)

$$= X$$

e.g., prove the theorem:

$$X + X \cdot Y = X$$

identity (1D)

$$X + X \cdot Y = X \cdot 1 + X \cdot Y$$

distributivity (8)

$$= X \cdot (1 + Y)$$

uniting (2)

$$= X \cdot (1)$$

identity (1D)

$$= X$$

9

# Proving theorems (truth table)

- Using complete truth table:

– e.g., de Morgan's:

$$(X + Y)' = X' \cdot Y'$$

NOR is equivalent to AND  
with inputs complemented

X	Y	X'	Y'	$(X + Y)'$	$X' \cdot Y'$
0	0	1	1	1	1
0	1	1	0	0	0
1	0	0	1	0	0

$$(X \cdot Y)' = X' + Y'$$

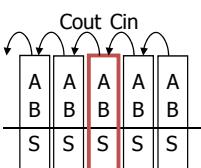
NAND is equivalent to OR  
with inputs complemented

X	Y	X'	Y'	$(X \cdot Y)'$	$X' + Y'$
0	0	1	1	1	1
0	1	1	0	0	1
1	0	0	1	0	0

10

## A simple example: 1-bit binary adder

- Inputs: A, B, Carry-in
- Outputs: Sum, Carry-out



A	B	Cin	Cout	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$S = A' B' \text{Cin} + A' B \text{Cin}' + A B' \text{Cin}' + A B \text{Cin}$$

$$\text{Cout} = A' B \text{Cin} + A B' \text{Cin} + A B \text{Cin}' + A B \text{Cin}$$

$$\text{Cout} = A' B \text{Cin} + A B' \text{Cin} + A B \text{Cin}' + A B \text{Cin}$$

11

12

# Apply the theorems to simplify expressions

- The theorems of Boolean algebra can simplify expressions
  - e.g., full adder's carry-out function

$$\begin{aligned}
 \text{Cout} &= A' B \text{ Cin} + A B' \text{ Cin} + A B \text{ Cin}' + A B \text{ Cin} \\
 &= A' B \text{ Cin} + A B' \text{ Cin} + A B \text{ Cin}' + A B \text{ Cin} + A B \text{ Cin} + A B \text{ Cin} \\
 &= A' B \text{ Cin} + A B \text{ Cin} + A B' \text{ Cin} + A B \text{ Cin}' + A B \text{ Cin} \\
 &= (A' + A) B \text{ Cin} + A B' \text{ Cin} + A B \text{ Cin}' + A B \text{ Cin} \\
 &= (1) B \text{ Cin} + A B' \text{ Cin} + A B \text{ Cin}' + A B \text{ Cin} \\
 &= B \text{ Cin} + A B' \text{ Cin} + A B \text{ Cin}' + A B \text{ Cin} + A B \text{ Cin} \\
 &= B \text{ Cin} + A B' \text{ Cin} + A B \text{ Cin} + A B \text{ Cin}' + A B \text{ Cin} \\
 &= B \text{ Cin} + A (B' + B) \text{ Cin} + A B \text{ Cin}' + A B \text{ Cin} \\
 &= B \text{ Cin} + A (1) \text{ Cin} + A B \text{ Cin}' + A B \text{ Cin} \\
 &= B \text{ Cin} + A \text{ Cin} + A B (\text{Cin}' + \text{Cin}) \\
 &= B \text{ Cin} + A \text{ Cin} + A B (1) \\
 &= B \text{ Cin} + A \text{ Cin} + A B
 \end{aligned}$$

adding extra terms creates new factoring opportunities

13

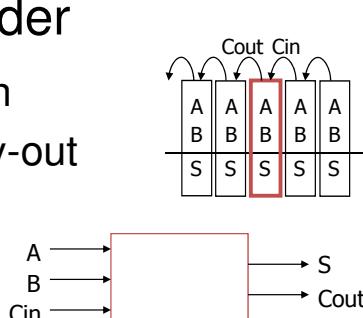
$$S = A' B' \text{ Cin} + A' B \text{ Cin}' + A B' \text{ Cin}' + A B \text{ Cin}$$

14

## A simple example: 1-bit binary adder

- Inputs: A, B, Carry-in
- Outputs: Sum, Carry-out

A	B	Cin	Cout	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



$$\text{Cout} = B \text{ Cin} + A \text{ Cin} + A B$$

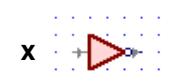
$$S = A \text{ xor } (B \text{ xor Cin})$$

15

## Recall Gates

- NOT

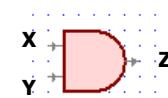
$$x' \quad \bar{x} \quad \neg x$$



X	Y
0	1
1	0

- AND

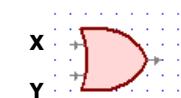
$$X \cdot Y \quad XY \quad X \wedge Y$$



X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

- OR

$$X + Y \quad X \vee Y$$



X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

16

## Recall gates (cont)

- NAND



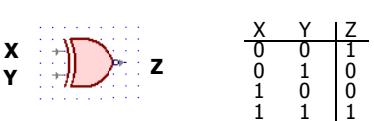
- NOR



- XOR  
 $X \oplus Y$

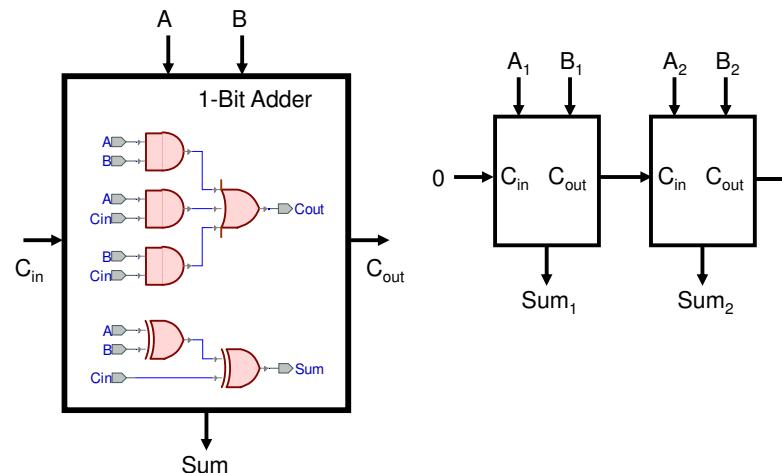


- XNOR  
 $X \leftrightarrow Y, X = Y$



17

## A 2-bit ripple-carry adder



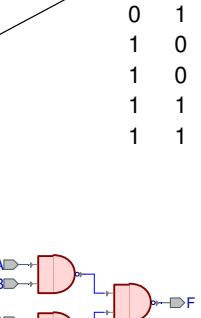
18

## Mapping truth tables to logic gates

- Given a truth table:
  - Write the Boolean expression
  - Minimize the Boolean expression
  - Draw as gates
  - Map to available gates

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

(1)  $F = A'BC' + A'BC + AB'C + ABC$   
       =  $A'B(C' + C) + AC(B' + B)$   
       =  $A'B + AC$



19

## Canonical forms

- Truth table is the unique signature of a Boolean function
- The same truth table can have many gate realizations
  - we've seen this already
  - depends on how good we are at Boolean simplification
- Canonical forms
  - standard forms for a Boolean expression
  - we all come up with the same expression

20

## Sum-of-products canonical form

- Also known as Disjunctive Normal Form (DNF)
- Also known as minterm expansion

			F	F'			
A	B	C	001	011	101	110	111
0	0	0	0	1			
0	0	1	1	0			
0	1	0	0	1			
0	1	1	1	0			
1	0	0	0	1			
1	0	1	1	0			
1	1	0	1	0			
1	1	1	1	0			

$F = A'B'C + A'BC + AB'C + ABC' + ABC$

$F' = A'B'C' + A'BC' + AB'C'$

21

## Sum-of-products canonical form (cont)

- Product term (or minterm)
  - ANDED product of literals – input combination for which output is true
  - each variable appears exactly once, true or inverted (but not both)

A	B	C	minterms
0	0	0	$A'B'C'$ m0
0	0	1	$A'B'C$ m1
0	1	0	$A'BC'$ m2
0	1	1	$A'BC$ m3
1	0	0	$AB'C'$ m4
1	0	1	$AB'C$ m5
1	1	0	$ABC'$ m6
1	1	1	$ABC$ m7

F in canonical form:  
 $F(A, B, C) = \Sigma m(1,3,5,6,7)$   
 $= m_1 + m_3 + m_5 + m_6 + m_7$   
 $= A'B'C + A'BC + AB'C + ABC' + ABC$

canonical form  $\neq$  minimal form  
 $F(A, B, C) = A'B'C + A'BC + AB'C + ABC + ABC'$   
 $= (A'B' + A'B + AB' + AB)C + ABC'$   
 $= ((A' + A)(B' + B))C + ABC'$   
 $= C + ABC'$   
 $= ABC' + C$   
 $= AB + C$

short-hand notation for minterms of 3 variables

22

## Product-of-sums canonical form

- Also known as Conjunctive Normal Form (CNF)
- Also known as maxterm expansion

			F	F'	
A	B	C	000	010	100
0	0	0	0	1	
0	0	1	1	0	
0	1	0	0	1	
0	1	1	1	0	
1	0	0	0	1	
1	0	1	1	0	
1	1	0	1	0	
1	1	1	1	0	

$F = (A + B + C) (A + B' + C) (A' + B + C)$

$F' = (A + B + C') (A + B' + C') (A' + B + C') (A' + B' + C) (A' + B' + C')$

23

## Product-of-sums canonical form (cont)

- Sum term (or maxterm)
  - OREd sum of literals – input combination for which output is false
  - each variable appears exactly once, true or inverted (but not both)

A	B	C	maxterms
0	0	0	$A+B+C$ M0
0	0	1	$A+B+C'$ M1
0	1	0	$A+B'+C$ M2
0	1	1	$A+B'+C'$ M3
1	0	0	$A'+B+C$ M4
1	0	1	$A'+B+C'$ M5
1	1	0	$A'+B'+C$ M6
1	1	1	$A'+B'+C'$ M7

F in canonical form:  
 $F(A, B, C) = \Pi M(0,2,4)$   
 $= M_0 \cdot M_2 \cdot M_4$   
 $= (A + B + C) (A + B' + C) (A' + B + C)$

short-hand notation for maxterms of 3 variables

24