

# Digital Logic Design

## ECE300

### Lecture 2

### Boolean Algebra and Logic Gates

Fall 2011

CSE3201

1

## Boolean Algebra (Axiomatic Definition).

- Boolean algebra is an algebraic structure defined by a set of elements  $B$ , together with two binary operators  $+$  and  $\cdot$ , provided that the following postulates are satisfied (Huntington).
- Closure with respect to  $+$  and  $\cdot$ .
- Identity element of  $+$  is  $0$ , to  $\cdot$  is  $1$
- Commutative wrt  $+$  and  $\cdot$ .
- Distributive over  $+$  and  $\cdot$ .  $X \cdot (Y + Z) = (X \cdot Y) + (X \cdot Z)$  and over  $+$ .  $X + (Y \cdot Z) = (X + Y) \cdot (X + Z)$
- For every element in  $B$   $x$ , there is  $x'$  such that  $x + x' = 1$  and  $x \cdot x' = 0$
- There are at least 2 different elements in  $B$

Fall 2011

CSE3201

2

## Two-Valued Boolean Algebra

- The element 1 and 0 operation are OR and AND
- All postulates are satisfied

Fall 2011

CSE3201

3

## Duality

- Duality Principle: In Every algebraic expression deducible from the postulates of Boolean Algebra remains valid if the operators if the operators and identity elements are interchanged. In 2-valued Boolean algebra, exchange AND and OR, and 1 and 0

Fall 2011

CSE3201

4

## Basic Theorems

Postualte 1	$X+0=x$	$X \cdot 1=1$
Postulate 5	$X+X'=1$	$X \cdot X'=0$
Theorem 1	$X+X=X$	$X \cdot X=1$
Theorem 2	$X + 1 = 1$	$X \cdot 0 = 0$
Theorem 3	$(x')'=x$	
Commutative	$X + Y = Y + X$	$X \cdot Y = Y \cdot X$
Associative	$X + (Y + Z)=(X + Y) + Z$	$X(YZ)=(XY)Z$
Distributive	$X(Y+Z)=(X \cdot Y)+(X \cdot Z)$	$X + YZ = (X+Y)(X+Z)$
DeMorgan	$(x+y)'=x'y'$	$(xy)'=x'+y'$
Absorption	$x + xy = x$	$x(x+y)=x$

Fall 2011

CSE3201

5

## Boolean Function

- Boolean functions can be represented in a truth table that shows the value of the function for all different combination of the input variables.
- An algebraic expression
- Circuit diagram that implements the algebraic expression
- Show as an example  $F=x+y'z$  and  $F=x'y'z+xz+y'z'$

Fall 2011

CSE3201

6

## Algebraic manipulation

- We define a *literal* to be a single variable within  $x'y+zxy$  is composed of 2 terms and 5 literals.
- By reducing the number of literals, or terms we can obtain a simpler circuit
- $x(x'+y)=xx'+xy=0+xy=xy$
- $(x+y)(x+y')=x+xy+xy'+yy'=x(1+y+y')=x$

Fall 2011

CSE3201

7

## Algebraic manipulation

- You can find the complement of a function by taking their duals, and complementing each literal.
- $F=x'yz'+x'y'z$
- Dual of F is  $(x'+y+z')(x'+y'+z')$
- Complementing literals  $(x+y'+z)(x+y+z)$
- $F'=(x'yz')' (x'y'z)'$
- $F'=(x+y'+z)(x+y+z)$

Fall 2011

CSE3201

8

# Canonical and Standard Forms

- If you if we have  $n$  variables, we can have  $2^n$  different combination of these variables either in its normal or complemented form.
- Each of these terms is called a *minterm*
- In a similar matter,  $n$  variables added (Ored) can form  $2^n$  *maxterm*
- A boolean function can be expressed algebraically from a given truth table by forming a minterm for each combination of the variables that produces a 1 in the function and taking the OR of all these terms.

Fall 2011

CSE3201

9

## Canonical Form

	minterms		maxterms	
X y z	term		term	
0 0 0	$x'y'z'$	m0	$x+y+z$	M0
0 0 1	$x'y'z$	m1	$x+y+z'$	M1
0 1 0	$x'yz'$	m2	$x+y'+z$	M2
0 1 1	$x'yz$	m3	$x+y'+z'$	M3
1 0 0	$xy'z'$	m4	$x'+y+z$	M4
1 0 1	$xy'z$	m5	$x'+y+z'$	M5
1 1 0	$xyz'$	m6	$x'+y'+z$	M6
1 1 1	$xyz$	m7	$x'+y'+z'$	M7

$$m_i = M_j$$

Fall 2011

CSE3201

10

## Canonical Form

- A Boolean function can be expressed algebraically from a given truth table by forming a minterm for each combination of the variables that produces a 1 in the function, then taking the OR of all these terms.
- It could be also expressed as the product of maxterms, where a maxterm is formed for each combination of the variables that produces a 0 in the function.

Fall 2011

CSE3201

11

## Canonical Form

- Example consider the following table
- $F = x'y'z' + x'yz' + xy'z'$
- $F = m_0 + m_2 + m_4$
- $F' = x'y'z + x'yz + xy'z + xyz' + xyz$
- $F = (x+y+z')(x+y'+z')(x'+y+z)(x'+y'+z)$
- $F = M_1 \cdot M_3 \cdot M_5 \cdot M_6 \cdot M_7$

x	y	z	f
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Fall 2011

CSE3201

12

## Canonical Form

- Example

$$F(A, B, C) = \sum(1, 4, 5, 6, 7)$$

$$F'(A, B, C) = \sum(0, 2, 3) = m_0 + m_2 + m_3$$

$$F = \overline{(m_0 + m_2 + m_3)} = \overline{m_0} \overline{m_2} \overline{m_3} = M_0 + M_2 + M_3$$

$$F = \prod(0, 2, 3)$$

Fall 2011

CSE3201

13

## Canonical Form

- Express the function  $F=A+B'C$  in a sum of minterm
- Method 1 make truth table
- Method 2, note that
  - $A=A(B+B')=AB+AB'$
  - $F=AB+AB'+B'C$
  - $F=AB(C+C')+AB'(C+C')+(A+A')B'C$
  - $F=ABC+ABC'+AB'C+AB'C'+AB'C+A'B'C$
  - $F=m_7+m_6+m_5+m_4+m_3+m_1=\sum(1, 4, 5, 6, 7)$

Fall 2011

CSE3201

14

## Other Logic Functions

x	y	F0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Fall 2011



CSI 20



15

## Other Logic Functions

$F0=0$		Null	Constant 0
$F1=xy$	$x.y$	AND	
$F2=xy'$	$x/y$	Inhibition	X but not y
$F3=x$		transfer	
$F4=x'y$	$y/x$	Inhibition	Y but not x
$F5=y$		Transfer	
$F6=xy'+x'y$	$X \oplus y$	Exclusive OR	X, or y but not both
$F7=x+y$	$X+y$	OR	
$F8=(x+y)'$	$X \downarrow Y$	NOR	Not OR

Fall 2011

CSE3201

16



## Other Functions

$F9=xy+x'y'$	$(x \oplus y)'$	Equivalence	X equals y
$F10=y'$	$Y'$	Complement	NOT y
$F11=x+y'$	$X \subset Y$	Implication	If y, then x
$F12=x'$	$X'$	Complement	NOT x
$F13=x'+y$	$X \supset Y$	Implication	If x, then y
$F14=(xy)'$	$X \uparrow Y$	NAND	NOT AND
$F15=1$		Identity	Constant 1

Fall 2011

CSE3201

17

## Digital Logic Gates

- Explain AND, OR, NOT, Buffer, NAND, NOR, EX-OR, EX-NOR



Negative Logic

Fall 2011

CSE3201

18

## Extension to Multiple Inputs

- The extension of AND, and OR is easy
- Consider NOR
- $(X \downarrow Y) \downarrow Z = ((x+y)' + z)' = xz' + yz'$
- For simplicity we define
- $X \downarrow Y \downarrow Z = (X+Y+Z)'$
- $X \uparrow Y \uparrow Z = (XYZ)'$

Fall 2011

CSE3201

19

## Positive and Negative Logic

- Hardware digital gates are defined in terms of signal values  $H$  and  $L$ , it is up to the user to define what is  $H$  and  $L$
- Consider the following table
- If we define  $H=1, L=0$   
It is AND (+ve logic)
- If we define  $H=0, L=1$   
It is OR (-ve Logic)

X	Y	F
L	L	L
L	H	L
H	L	L
H	H	H

Fall 2011

CSE3201

20

# Digital Logic Families

- TTL: standard
- ECL: high speed
- MOS: high component density
- CMOS: Low power, currently the dominant logic family