

# Katz Chapter 2 - Two-level Combinational Logic

- ① Basic Logic Functions (already done)**
- ② Gate Logic**
  - Laws & theorems of Boolean Algebra
  - 2-level canonical forms in completely specified functions
- ③ 2-level Simplification**
  - Boolean Cubes
  - Karnaugh Maps
  - Espresso Method

# Boolean Algebra

Set of elements  $B$ , binary operators  $\{+, \bullet\}$ , unary operation  $\{ '\}$ , such that the following axioms hold :

1.  $B$  contains at least two elements  $a, b$  such that  $a \neq b$ .

2. Closure :  $a, b$  in  $B$ ,

$$a + b \text{ in } B, \quad a \bullet b \text{ in } B, \quad a' \text{ in } B.$$

3. Commutative laws :

$$a + b = b + a, \quad a \bullet b = b \bullet a.$$

4. Identities :  $0, 1$  in  $B$

$$a + 0 = a, \quad a \bullet 1 = a.$$

5. Distributive laws :

$$a + (b \bullet c) = (a + b) \bullet (a + c), \quad a \bullet (b + c) = a \bullet b + a \bullet c.$$

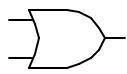
6. Complement :

$$a + a' = 1, \quad a \bullet a' = 0.$$

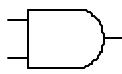
# Logic Functions

$B = \{0,1\}$ ,  $+$  = OR,  $\bullet$  = AND,  $'$  = NOT

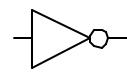
is a valid Boolean Algebra.



00		0
01		1
10		1
11		1



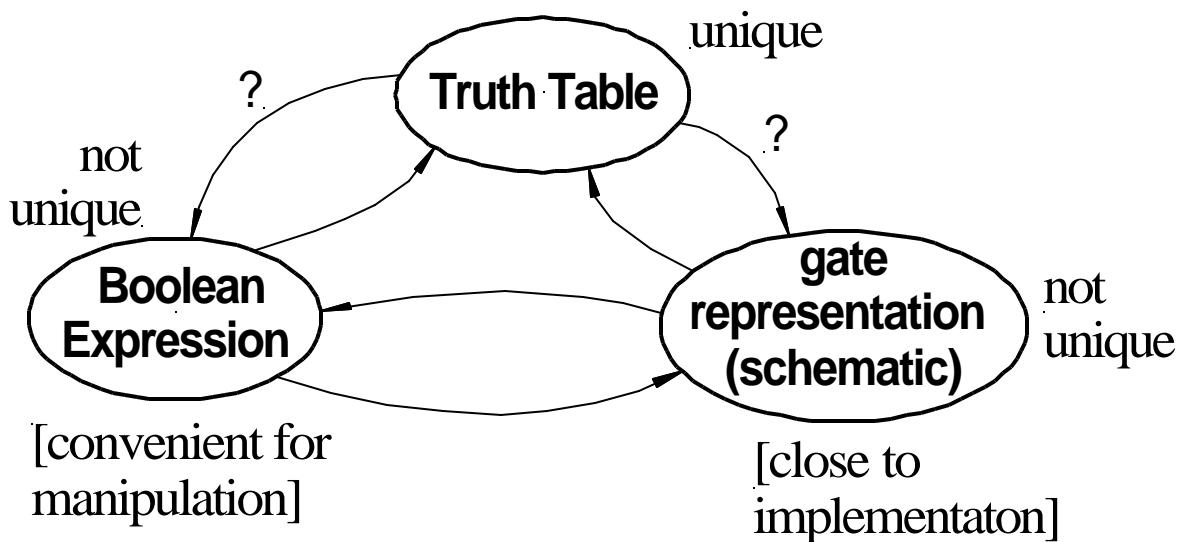
00		0
01		0
10		0
11		1



0		1
1		0

- Do the axioms hold?
  - Ex: communititive law:  $0+1 = 1+0$ ?

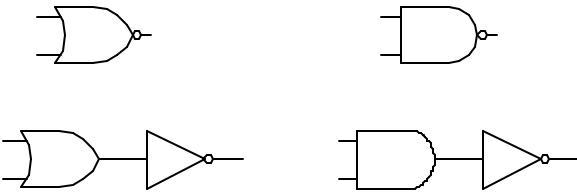
- \* Theorem: Any Boolean function that can be expressed as a truth table can be written as an expression in Boolean Algebra using AND, OR, NOT.

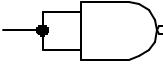
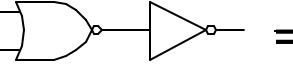
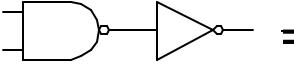


# Other logic functions of 2 variables (x,y)

xy	f0	f1													
00	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
01	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1
10	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
11	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1
	0	AND	X	Y	$\oplus$	OR	NOR	XNOR	NAND						1

Look at NOR and NAND:



- Simple implementation.
- Theorem: Any Boolean function that can be expressed as a truth table can be expressed using **NAND** and **NOR**.
  - Proof sketch:
    -  = NOT
    -  = OR
    -  = AND
  - HW: Show that NAND or NOR is sufficient.

# Laws of Boolean Algebra

Duality: A dual of a Boolean expression is derived by interchanging OR and AND operations, and 0s and 1s (literals are left unchanged).

$$\{F(x_1, x_2, \dots, x_n, 0, 1, +, \bullet)\}^D = \{F(x_1, x_2, \dots, x_n, 1, 0, \bullet, +)\}$$

Any law that is true for an expression is also true for its dual.

Operations with 0 and 1:

$$\begin{array}{ll} 1. x + 0 = x & x * 1 = x \\ 2. x + 1 = 1 & x * 0 = 0 \end{array}$$

Idempotent Law:

$$3. x + x = x \quad x \cdot x = x$$

Involution Law:

$$4. (x')' = x$$

Laws of Complementarity:

$$5. x + x' = 1 \quad x \cdot x' = 0$$

Commutative Law:

$$6. x + y = y + x \quad x \cdot y = y \cdot x$$

# Laws of Boolean Algebra (cont.)

Associative Laws:

$$(x + y) + z = x + (y + z) \quad x \cdot y \cdot z = x \cdot (y \cdot z)$$

Distributive Laws:

$$x \cdot (y + z) = (x \cdot y) + (x \cdot z) \quad x + (y \cdot z) = \\ (x + y)(x + z)$$

“Simplification” Theorems:

$$\begin{array}{ll} x \cdot y + x \cdot y' = x & (x + y) \cdot (x + y') = x \\ x + x \cdot y = x & x \cdot (x + y) = x \\ (x + y') \cdot y = x \cdot y & (x \cdot y') + y = x + y \end{array}$$

DeMorgan’s Law:

$$(x + y + z + \dots) = \quad (x \cdot y \cdot z \cdot \dots)' = \\ x' \cdot y' \cdot z' \cdot \dots \quad x' + y' + z' + \dots$$

Theorems for Multiplying and Factoring:

$$\begin{array}{ll} (x + y) \cdot (x' + z) = & x \cdot y + x' \cdot z = \\ x \cdot z + x' \cdot y & (x + z) \cdot (x' + y) \end{array}$$

Consensus Theorem:

$$\begin{array}{ll} x \cdot y + y \cdot z + x' \cdot z = & (x + y) \cdot (y + z) \cdot (x' + z) \\ x \cdot y + x' \cdot z & = (x + y) \cdot (x' + z) \end{array}$$

# Proving Theorems via axioms of Boolean Algebra

Ex: prove the theorem:  $x y + x y' = x$

$$x y + x y' = x (y + y') \text{ distributive law}$$

$$x (y + y') = x (1) \text{ complementary law}$$

$$x (1) = x \text{ identity}$$

Ex: prove the theorem:  $x + x y = x$

$$x + x y = x 1 + x y \text{ identity}$$

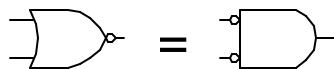
$$x 1 + x y = x (1 + y) \text{ distributive law}$$

$$x (1 + y) = x (1) \text{ identity}$$

$$x (1) = x \text{ identity}$$

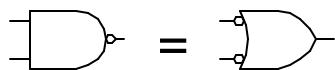
# DeMorgan's Law

$$(x + y)' = x' y'$$



x	y	x'	y'	$(x + y)'$	$x'y'$
0	0	1	1	1	1
0	1	1	0	0	0
1	0	0	1	0	0
1	1	0	0	0	0

$$(xy)' = x' + y'$$



x	y	x'	y'	$(xy)'$	$x' + y'$
0	0	1	1	1	1
0	1	1	0	1	1
1	0	0	1	1	1
1	1	0	0	0	0

- \* DeMorgan's Law can be used to convert AND/OR expressions to OR/AND expressions:

Example:  $z = a'b'c + a'bc + ab'c + abc'$

$$z' =$$

# Algebraic Simplification

Ex: full adder (FA) carry out function

$$Cout = a'b'c + ab'c + abc' + abc$$

# Algebraic Simplification

Ex: full adder (FA) carry out function

$$\text{Cout} = a'b'c + ab'c + abc' + abc$$

$$= a'b'c + ab'c + abc' + \mathbf{abc} + \mathbf{abc}$$

$$= a'b'c + \mathbf{abc} + ab'c + abc' + \mathbf{abc}$$

$$= (\mathbf{a}' + a)\mathbf{bc} + ab'c + abc' + abc$$

$$= (1)\mathbf{bc} + ab'c + abc' + abc$$

$$= bc + ab'c + abc' + \mathbf{abc} + \mathbf{abc}$$

$$= bc + ab'c + \mathbf{abc} + abc' + \mathbf{abc}$$

$$= bc + \mathbf{a(b' + b)c} + abc' + abc$$

$$= bc + \mathbf{a(1)c} + abc' + abc$$

$$= bc + ac + \mathbf{ab(c' + c)}$$

$$= bc + ac + \mathbf{ab(1)}$$

$$= bc + ac + ab$$