Boolean Algebra

CSIM601251

Instructor: Tim Dosen DDAK

Slide By : Erdefi Rakun

Fasilkom UI

Outline

- **Introduction**
- Laws and basic theorems
- Proofing a theorem

Digital Circuits

Advantages of digital circuits over analog circuits

- More reliable (simpler circuits, less noise-prone)
- Specified accuracy (determinable)
- Binary variables take on one of two values.
- Logical operators operate on binary values and binary variables.
- Basic logical operators are the logic functions AND, OR and NOT.
- Logic gates implement logic functions.

Boolean Algebra

- Boolean Algebra: a useful mathematical system for specifying and transforming logic functions.
- We study Boolean algebra as a foundation for designing and analyzing digital systems!

Boolean Algebra

- Boolean values:
  - True (1)
  - False (0)
- Connectives
  - Conjunction (AND)
    - A • B; A ∧ B
  - Disjunction (OR)
    - A + B; A ∨ B
  - Negation (NOT)
    - Ā ; ¬A; A'

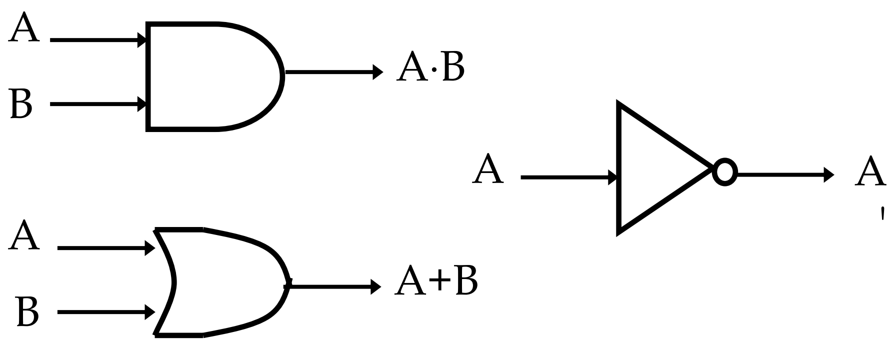
Truth tables

A	B	A • B
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1

A	A'
0	1
1	0

Logic gates



Notation Examples

- Examples:

  $\mathbf{Y = A \times B}$ is read "Y is equal to A AND B."

  $\mathbf{z = x + y}$ is read "z is equal to x OR y."

  $\mathbf{X = \overline{A}}$ is read "X is equal to NOT **A.**"

- Note: The statement:

  1 + 1 = 2 (read "one plus one equals two")

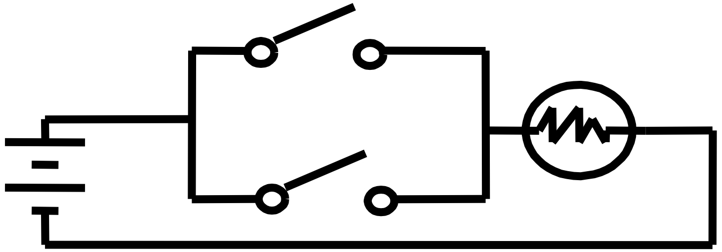
  is not the same as

  1 + 1 = 1 (read "1 or 1 equals 1").

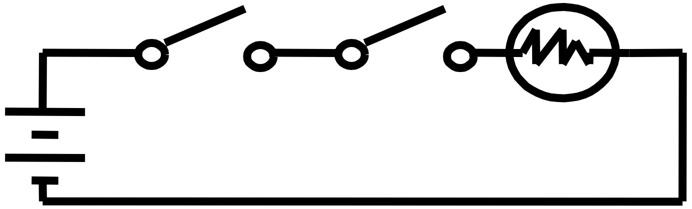
Logic Function Implementation

- Using Switches
  - For inputs:
    - logic 1 is switch closed
    - logic 0 is switch open
  - For outputs:
    - logic 1 is light on
    - logic 0 is light off.
  - NOT uses a switch such that:
    - logic 1 is switch open
    - logic 0 is switch closed

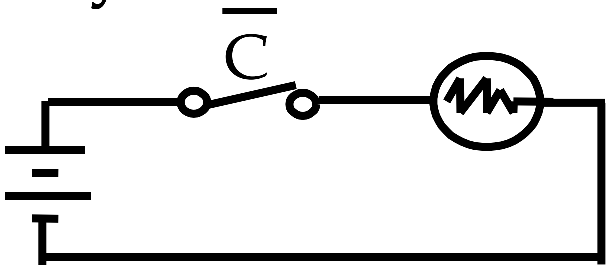
Switches in parallel => OR



Switches in series => AND

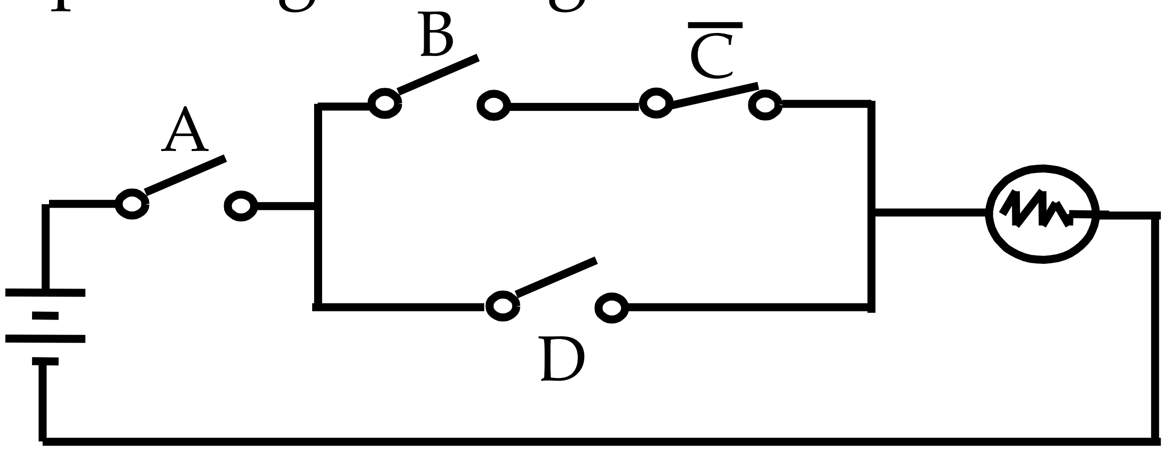


Normally-closed switch => NOT



Logic Function Implementation (Continued)

- Example: Logic Using Switches



- Light is on (L = 1) for

  L(A, B, C, D) = ?

  and off (L = 0), otherwise.
- Useful model for relay circuits and for CMOS gate circuits, the foundation of current digital logic technology

Logic Diagrams and Expressions

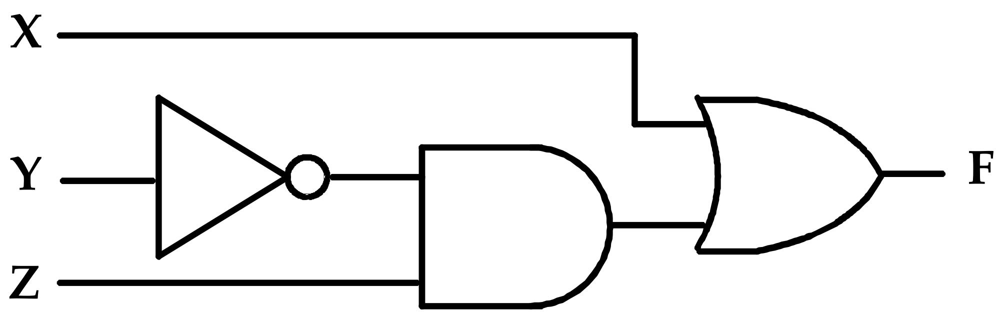
Truth Table

X Y Z	$F = X + \overline{Y} \times Z$
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	0
1 0 0	1
1 0 1	1
1 1 0	1
1 1 1	1

Equation

$$F = X + \overline{Y}\,Z$$

Logic Diagram



- Boolean equations, truth tables and logic diagrams describe the same function!
- Truth tables are unique; expressions and logic diagrams are not. This gives flexibility in implementing functions.

Outline

- Introduction
- **Laws and basic theorems**
- Proofing a theorem

Note: Portion of this material are taken from Aaron Tan's slide and other portions of this material © 2008 by Pearson Education,Inc

Laws of Boolean Algebra

- Identity laws

  $A + 0 = 0 + A = A$; $A \cdot 1 = 1 \cdot A = A$

- Inverse/complement laws

  $A + A' = 1$; $A \cdot A' = 0$

- Commutative laws

  $A + B = B + A$; $A \cdot B = B \cdot A$

- Associative laws

  $A + (B + C) = (A + B) + C$; $A \cdot (B \cdot C) = (A \cdot B) \cdot C$

- Distributive laws

  $A \cdot (B + C) = (A \cdot B) + (A \cdot C)$; $A + (B \cdot C) = (A + B) \cdot (A + C)$

Precedence of Operators

- Precedence from highest to lowest
  - Not
  - And
  - Or
- Examples:
  - $A \cdot B + C = (A \cdot B) + C$
  - $X + Y' = X + (Y')$
  - $P + Q' \cdot R = P + ((Q') \cdot R)$
- Use parenthesis to overwrite precedence. Examples:
  - $A \cdot (B + C)$
  - $(P + Q)' \cdot R$

Duality

- If the AND/OR operators and identity elements 0/1 in a Boolean equation are interchanged, it remains valid

  Example:

  The dual equation of $a+(b \cdot c)=(a+b) \cdot (a+c)$ is $a \cdot (b+c)=(a \cdot b)+(a \cdot c)$

- Duality gives free theorems – "two for the price of one". You prove one theorem and the other comes for free!

  Examples:

  If $(x+y+z)' = x' \cdot y' \cdot z'$ is valid, then its dual is also valid:

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

  If $x+1 = 1$ is valid, then its dual is also valid:

  $x \cdot 0 = 0$

Basic Theorems (1/2)

1. Idempotency

   $X + X = X$;　　$X \cdot X = X$

2. Zero and One elements

   $X + 1 = 1$;　　$X \cdot 0 = 0$

3. Involution

   $(X')' = X$

4. Absorption

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

5. Absorption (variant)

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

Basic Theorems (2/2)

6. DeMorgan's

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

   DeMorgan's Theorem can be generalized to more than two variables, example: $(A + B + \ldots + Z)' = A' \cdot B' \cdot \ldots \cdot Z'$

7. Consensus

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

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

Identity Name	AND Form	OR Form
Identity Law	$1x = x$	$0+x = x$
Null (or Dominance) Law	$0x = 0$	$1+x = 1$
Idempotent Law	$xx = x$	$x+x = x$
Inverse Law	$x\overline{x} = 0$	$x+\overline{x} = 1$
Commutative Law	$xy = yx$	$x+y = y+x$
Associative Law	$(xy)z = x(yz)$	$(x+y)+z = x+(y+z)$
Distributive Law	$x+yz = (x+y)(x+z)$	$x(y+z) = xy+xz$
Absorption Law	$x(x+y) = x$	$x+xy = x$
DeMorgan's Law	$(\overline{xy}) = \overline{x}+\overline{y}$	$(\overline{x+y}) = \overline{x}\overline{y}$
Double Complement Law	$\overline{\overline{x}} = x$	

TABLE 3.5 **Basic Identities of Boolean Algebra**

Outline

- Introduction
- Laws and basic theorems
- **Proofing a theorem**

Note: Portion of this material are taken from Aaron Tan's slide and other portions of this material © 2008 by Pearson Education,Inc

Proving a Theorem

- Theorems can be proved using truth table, or by algebraic manipulation using other theorems/laws.

  Example: Prove absorption theorem $X + X \cdot Y = X$

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

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

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

  $= X \cdot 1$ (by one element)

  $= X$ (by identity)

  By duality, we have also proved $X \cdot (X+Y) = X$
- Our primary reason for doing proofs is to learn:
  - Careful and efficient use of the identities and theorems of Boolean algebra, and
  - How to choose the appropriate identity or theorem to apply to make forward progress, irrespective of the application.

Truth Table

- Provide a listing of every possible combination of inputs and its corresponding outputs.
  - Inputs are usually listed in binary sequence.
- Example
  - Truth table with 3 inputs and 2 outputs

x	y	z	y + z	x • (y + z)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

Proof Using Truth Table

- Prove: **x • (y + z) = (x • y) + (x • z)**
  - Construct truth table for LHS and RHS

x	y	z	y + z	**x • (y + z)**	x • y	x • z	**(x • y) + (x • z)**
0	0	0					
0	0	1					
0	1	0					
0	1	1					
1	0	0					
1	0	1					
1	1	0					
1	1	1					

  - Check that column for LHS = column for RHS

Example 2: Boolean Algebraic Proofs

- $AB + \overline{A}C + BC = AB + \overline{A}C$ (Consensus Theorem)

Proof Steps | Justification (identity or theorem)

$AB + \overline{A}C + BC$

$= AB + \overline{A}C + 1 \cdot BC$?

$= AB + \overline{A}C + (A + \overline{A}) \cdot BC$?

$=$

Example 3: Boolean Algebraic Proofs

- $(\overline{X + Y})Z + X\overline{Y} = \overline{Y}(X + Z)$

Proof Steps Justification (identity or theorem)

$(\overline{X + Y})Z + X\overline{Y}$

$=$

Boolean Functions

- Examples of Boolean functions (logic equations):

  F1(x,y,z) = x·y·z'

  F2(x,y,z) = x + y'·z

  F3(x,y,z) = x'·y'·z + x'·y·z + x·y'

  F4(x,y,z) = x·y' + x'·z

x	y	z	F1	F2	F3	F4
0	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	0	0	0	0	0
0	1	1	0	0	1	1
1	0	0	0	1	1	1
1	0	1	0	1	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	0

Complement

- Given a Boolean function F, the complement of F, denoted as F', is obtained by interchanging 1 with 0 in the function's output values.
- Example: F1 = x·y·z'
- What is F1' ?

x	y	z	F1	F1'
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	1	
1	1	1	0	

Complementing Functions

- Use DeMorgan's Theorem to complement a function:
  1. Interchange AND and OR operators
  2. Complement each constant value and literal
- Example: Complement F = $\bar{x}y\bar{z} + x\bar{y}\bar{z}$

  $\bar{F} = (x + \bar{y} + z)(\bar{x} + y + z)$
- Example: Complement G = $(\bar{a} + bc)\bar{d} + e$

  $\bar{G} =$

Expression Simplification

- An application of Boolean algebra
- Simplify to contain the smallest number of <u>literals</u> (complemented and uncomplemented variables):

$$AB + \bar{A}CD + \bar{A}BD + \bar{A}C\bar{D} + ABCD$$

$$= AB + ABCD + \bar{A}CD + \bar{A}C\bar{D} + \bar{A}BD$$

$$= AB + AB(CD) + \bar{A}C(D + \bar{D}) + \bar{A}BD$$

$$= AB + \bar{A}C + \bar{A}BD = B(A + \bar{A}D) + \bar{A}C$$

$$= B(A + D) + \bar{A}C \quad \text{5 literals}$$