Digital Logic Design

Lecture No. 4

Overview

- Logic Gates
- Timing Diagram
- Gate Delay
- Boolean Equation, Logic Circuit, Truth Table
- Boolean Identities and Laws (Commutative etc.)
- Dual and Duality Principle
- Proof of Distributive Law and Consensus Theorem
- Complementing Function using DE Morgan’s Law and Dual

Binary Logic and Gates

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

Binary Variables

- **Recall that the two binary values have different names:**
  - **True/False**
  - **On/Off**
  - **Yes/No**
  - **1/0**
- **We use 1 and 0 to denote the two values.**

Logical Operations

- **The three basic logical operations are:**
  - **AND**
  - **OR**
  - **NOT**
- **AND is denoted by a dot (·).**
- **OR is denoted by a plus (+).**
- **NOT is denoted by an overbar (¯), a single quote mark (') after, or (~) before the variable.**

Notation Examples

- **Examples:**
  - **Y=A.B is read “Y is equal to A AND B.”**
  - $z = x + y$ **is read “z is equal to x OR y.”**
  - $X = \overline{A}$ **is read “X is equal to NOT A.”**

- **Note: The statement:**

  1 + 1 = 2 (read “one <u>plus</u> one equals two”)

  is not the same as

  1 + 1 = 1 (read “1 <u>or</u> 1 equals 1”).

Operator Definitions

- **Operations are defined on the values "0" and "1" for each operator:**

AND	OR	NOT
$0 \cdot 0 = 0$	$0 + 0 = 0$	$\overline{0} = 1$
$0 \cdot 1 = 0$	$0 + 1 = 1$	$\overline{1} = 0$
$1 \cdot 0 = 0$	$1 + 0 = 1$	
$1 \cdot 1 = 1$	$1 + 1 = 1$	

Truth Tables

- *Truth table* – a tabular listing of the values of a function for all possible combinations of values on its arguments
- Example: Truth tables for the basic logic operations:

AND		
X	Y	$Z = X \cdot Y$
0	0	0
0	1	0
1	0	0
1	1	1

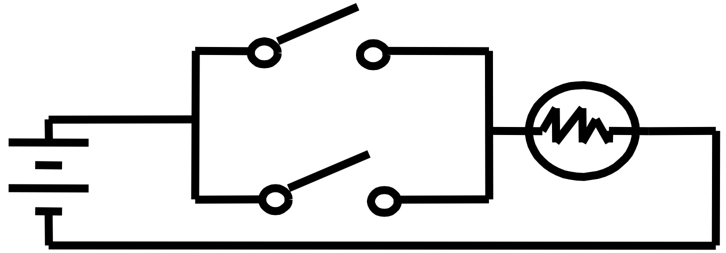
OR		
X	Y	$Z = X + Y$
0	0	0
0	1	1
1	0	1
1	1	1

NOT	
X	$Z = \overline{X}$
0	1
1	0

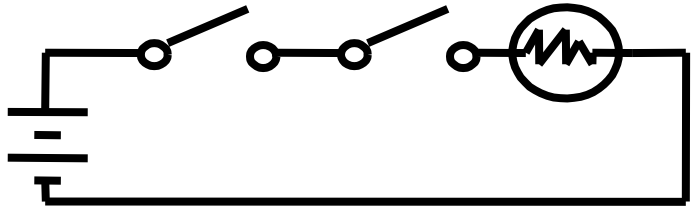
Logic Function Implementation

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

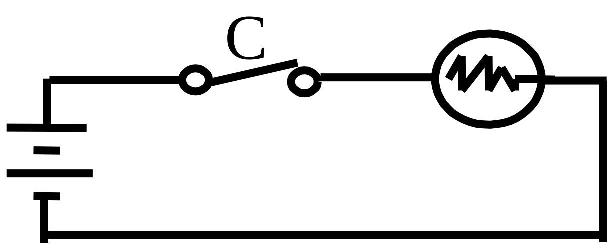
Switches in parallel => OR



Switches in series => AND

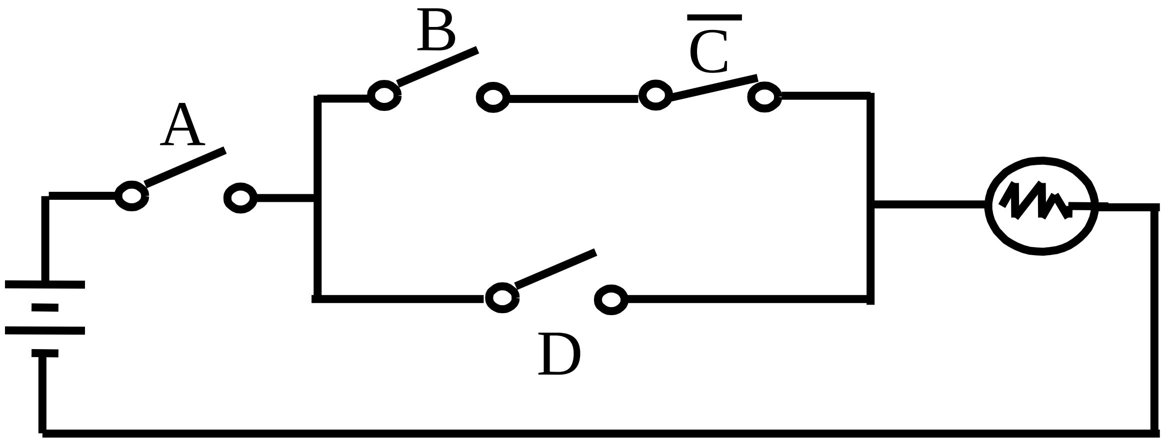


Normally-closed switch => NOT



Logic Function Implementation (Continued)

- **Example: Logic Using Switches**



- **Light is on (L = 1) for**

  L(A, B, C, D) =

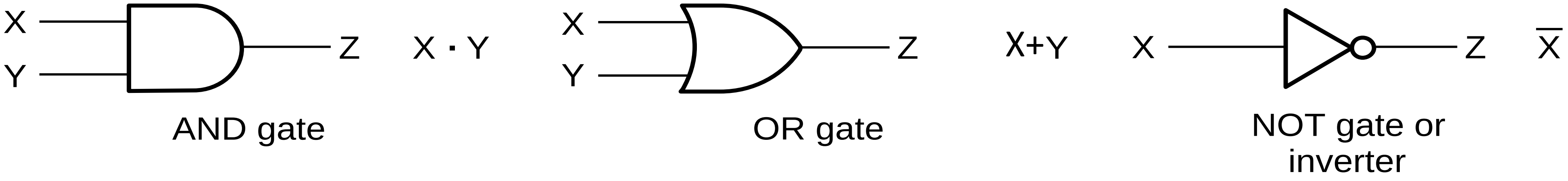
  and off (L = 0), otherwise.

Logic Gates

- **In the earliest computers, switches were opened and closed by magnetic fields produced by energizing coils in *relays*. The switches in turn opened and closed the current paths.**
- **Later, *vacuum tubes* that open and close current paths electronically replaced relays.**
- **Today, *transistors* are used as electronic switches that open and close current paths.**

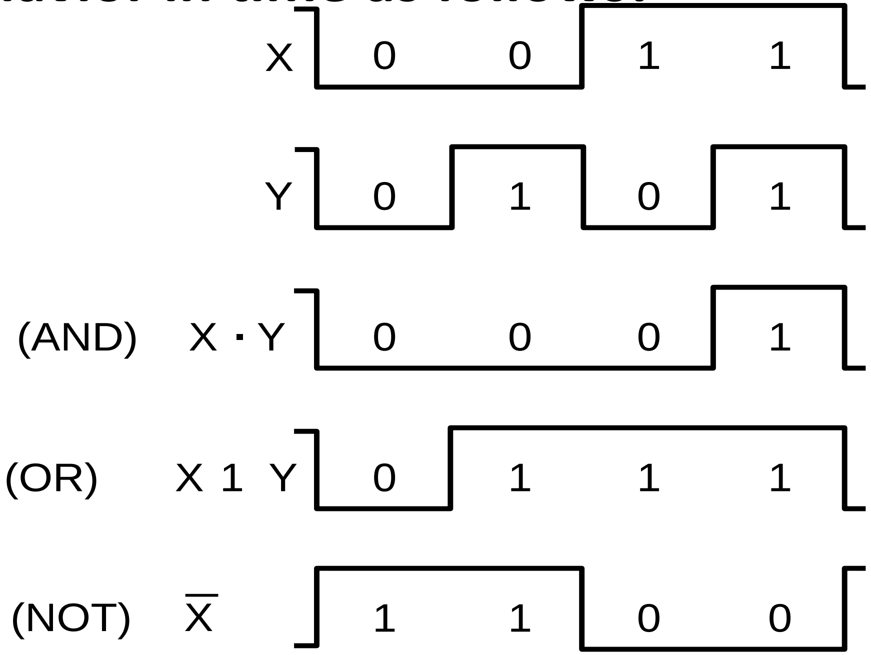
Logic Gate Symbols and Behavior

- **Logic gates have special symbols:**



(a) Graphic symbols

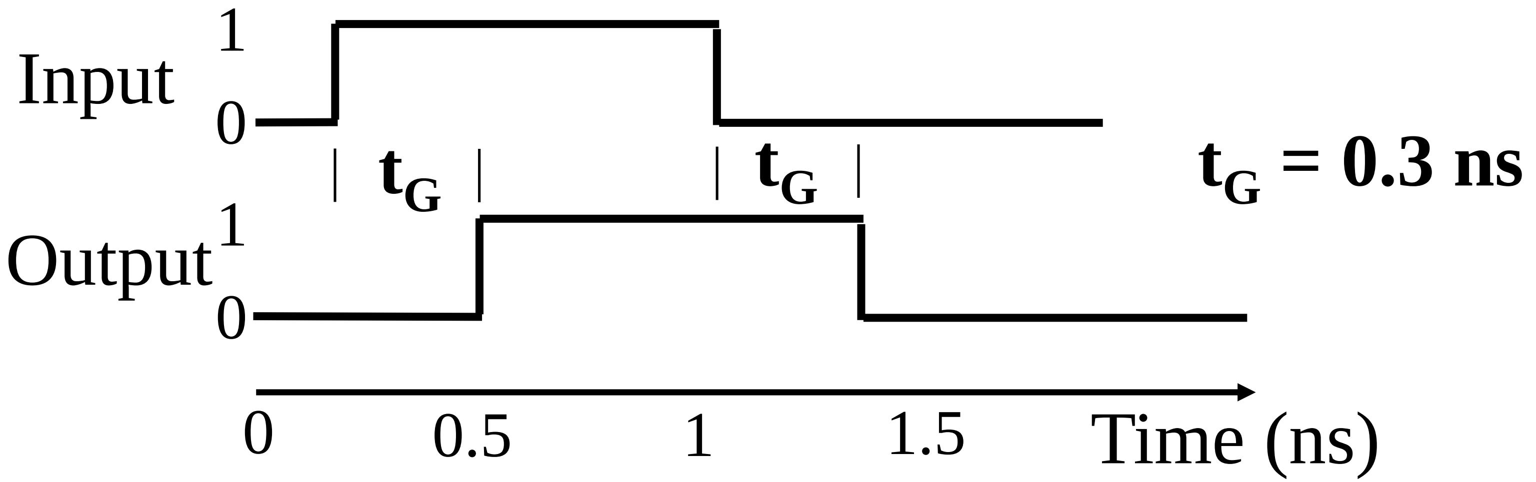
- **And waveform behavior in time as follows:**



(b) Timing diagram

Gate Delay

- **In actual physical gates, if one or more input changes causes the output to change, the output change does not occur instantaneously.**
- **The delay between an input change(s) and the resulting output change is the *gate delay* denoted by t_G:**



Logic Diagrams and Expressions

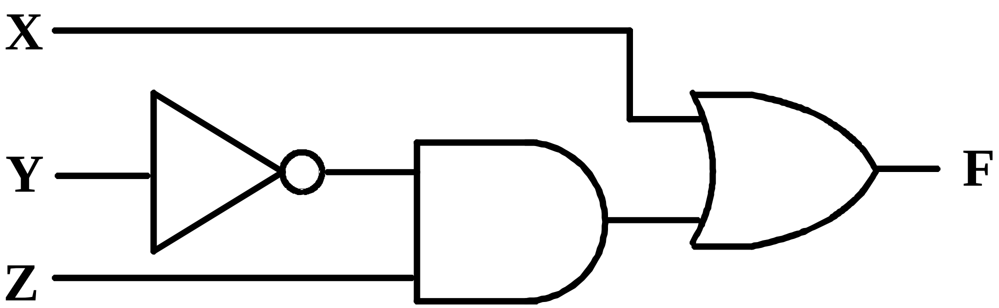
Truth Table

X Y Z	$F = X + \overline{Y} \cdot Z$
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.**

Boolean Algebra

- **An algebraic structure defined on a set of at least two elements, B, together with three binary operators (denoted +, · and $\overline{}$) that satisfies the following basic identities:**

1.	$X + 0 = X$	2.	$X \cdot 1 = X$	
3.	$X + 1 = 1$	4.	$X \cdot 0 = 0$	
5.	$X + X = X$	6.	$X \cdot X = X$	
7.	$X + \overline{X} = 1$	8.	$X \cdot \overline{X} = 0$	
9.	$\overline{\overline{X}} = X$			
10.	$X + Y = Y + X$	11.	$XY = YX$	Commutative
12.	$(X + Y) + Z = X + (Y + Z)$	13.	$(XY)Z = X(YZ)$	Associative
14.	$X(Y + Z) = XY + XZ$	15.	$X + YZ = (X + Y)(X + Z)$	Distributive
16.	$\overline{X + Y} = \overline{X} \cdot \overline{Y}$	17.	$\overline{X \cdot Y} = \overline{X} + \overline{Y}$	DeMorgan's

Boolean Operator Precedence

- **The order of evaluation in a Boolean expression is:**
  1. Parentheses
  2. NOT
  3. AND
  4. OR
- **Consequence: Parentheses appear around OR expressions**
- **Example:** $F = A(B + C)(C + \overline{D})$

Dual and Duality Principle

- The dual of an expression is obtained by changing AND to OR and OR to AND throughout (and 1s to 0s and 0s to 1s if they appear in the expression).
- The *duality principle* of Boolean algebra states that a Boolean equation remains valid if we take the dual of the expressions on both sides of the equals sign.
- Example:

$$XY + X\overline{Y} = X(Y + \overline{Y}) = X \cdot 1 = X$$

This can be written as below equation using duality principle.

$$(X + Y)(X + \overline{Y}) = X + Y\overline{Y} = X + 0 = X$$

Example 1: Boolean Algebraic Proof

- **$A + A \cdot B = A$ (Absorption Theorem)**

Proof Steps	**Justification (identity or theorem)**
$A + A \cdot B$	
$= A \cdot 1 + A \cdot B$	$X = X \cdot 1$
$= A \cdot (1 + B)$	$X \cdot Y + X \cdot Z = X \cdot (Y + Z)$ (Distributive Law)
$= A \cdot 1$	$1 + X = 1$
$= A$	$X \cdot 1 = X$

- **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.**

Distributive law

- The multiplication of two variables and adding the result with a variable will result in same value as multiplication of addition of the variable with individual variables.
- In other words, ANDing two variables and ORing the result with another variable is equal to AND of ORing of the variable with the two individual variables.
- Distributive law can be written as
  - $A + BC = (A + B)(A + C)$
- This is called OR distributes over AND.

Example 2: Proof of Distributive law

- Theorem: A + BC = (A+B) (A+C)
- Taking R.H.S: (A + B) (A + C)

  =A.A + AC +AB + BC

  A + AC +AB +BC

  A (1+ C + B) + BC

  A (1+ B) +BC 1+C=1

  A (1) + BC 1+B=1

  A + BC

  = L.H.S

Example 3: Boolean Algebraic Proofs (*consensus theorem*)

The following *consensus theorem* is useful when simplifying Boolean expressions:

$$XY + \overline{X}Z + YZ = XY + \overline{X}Z$$

The theorem shows that the third term, *YZ*, is redundant and can be eliminated. Note that *Y* and *Z* are associated with *X* and $\overline{X}$ in the first two terms and appear together in the term that is eliminated. The proof of the consensus theorem is obtained by first ANDing *YZ* with ($X + \overline{X}$) = 1 and proceeds as follows:

$$\begin{aligned} XY + \overline{X}Z + YZ &= XY + \overline{X}Z + YZ(X + \overline{X}) \\ &= XY + \overline{X}Z + XYZ + \overline{X}YZ \\ &= XY + XYZ + \overline{X}Z + \overline{X}YZ \\ &= XY(1 + Z) + \overline{X}Z(1 + Y) \\ &= XY + \overline{X}Z \end{aligned}$$

The dual of the consensus theorem is

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

Useful Theorems

- $x \cdot y + \bar{x} \cdot y = y \quad (x + y)(\bar{x} + y) = y$ **Minimization**
- $x + x \cdot y = x \quad x \cdot (x + y) = x$ **Absorption**
- $x \cdot y + \bar{x} \cdot z + y \cdot z = x \cdot y + \bar{x} \cdot z$ **Consensus**

$\overline{x + y} = \bar{x} \cdot \bar{y} \quad \overline{x \cdot y} = \bar{x} + \bar{y}$ **DeMorgan's Laws**

DeMorgan's Laws

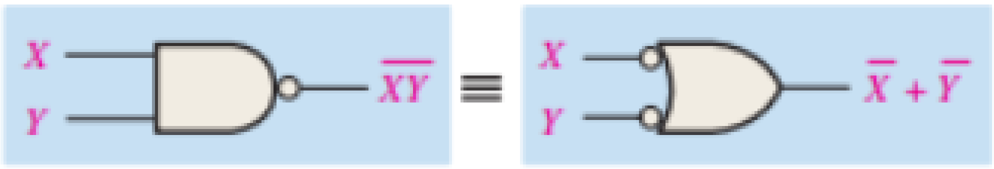
- The first law states that the complement of two or more ANDed variables is equivalent to the OR of the complements of the individual variables.

$$\overline{x \cdot y} = \bar{x} + \bar{y}$$

- This can also be stated as the complement of a product of variables is equal to the sum of the complements of the variables.
- The second law states that the complement of two or more ORed variables is equivalent to the AND of he complements of the individual variables.

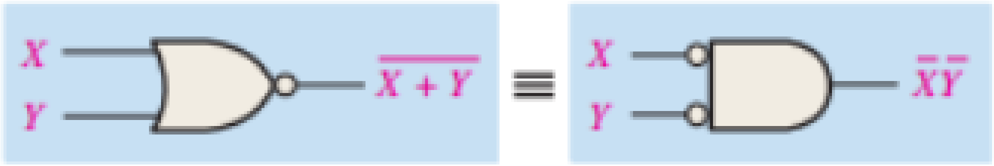
$$\overline{x + y} = \bar{x} \cdot \bar{y}$$

DeMorgan's Laws



NAND Negative-OR

Inputs		Output	
X	Y	$\overline{XY}$	$\overline{X} + \overline{Y}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0



NOR Negative-AND

Inputs		Output	
X	Y	$\overline{X + Y}$	$\overline{X}\,\overline{Y}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

Proof of DeMorgan's Laws

$$\overline{x + y} = \overline{x} \cdot \overline{y} \qquad \overline{x \cdot y} = \overline{x} + \overline{y}$$

Boolean Function Evaluation

$$F1 = xy\overline{z}$$
$$F2 = x + \overline{y}z$$
$$F3 = \overline{x}\,\overline{y}\,\overline{z} + \overline{x}\,y\,z + x\overline{y}$$
$$F4 = x\overline{y} + \overline{x}\,z$$

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

Complement of a Function

- **A simpler method for deriving the complement of a function is to take the dual of the function equation and complement each literal.**
- **The dual of an expression is obtained by interchanging AND and OR operations and 1s and 0s.**

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 =** $\bar{x}y\bar{z} + x\bar{y}\bar{z}$

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

  $\overline{G} = ?$

Complementing Functions

- **Find the complements of the functions by taking the duals of their equations and complementing each literal.**

1.

$$F_1 = \overline{X}Y\overline{Z} + \overline{X}\,\overline{Y}Z = (\overline{X}Y\overline{Z}) + (\overline{X}\,\overline{Y}Z)$$

The dual of F_1 is

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

Complementing each literal, we have

$$(X + \overline{Y} + Z)(X + Y + \overline{Z}) = \overline{F}_1$$

2.

$$F_2 = X(\overline{Y}\,\overline{Z} + YZ) = X((\overline{Y}\,\overline{Z}) + (YZ))$$

The dual of F_2 is

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

Complementing each literal yields

$$\overline{X} + (Y + Z)(\overline{Y} + \overline{Z}) = \overline{F}_2$$