THE UNIVERSITY OF TEXAS AT DALLAS

# CS4341 Digital Logic & Computer Design

## Lecture Notes 4

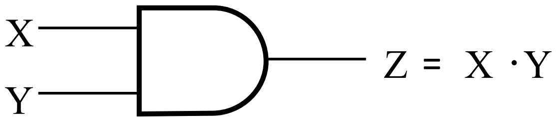
**Omar Hamdy**
Assistant Professor
Department of Computer Science

# Review: Truth Table

- *Truth table is a tabular listing of the values of a function for <u>ALL</u> possible combinations of values on its arguments*
- *Truth table is a useful tool to study the behavior of any Boolean function*

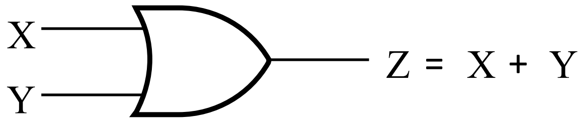
# Review: Basic Logic Gates and Truth Table

- *Logic gates are simple digital circuit that implements the logical operators*



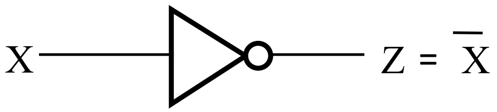

AND gate

| X | Y | Z |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |




OR gate

| X | Y | Z |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |




NOT gate or inverter

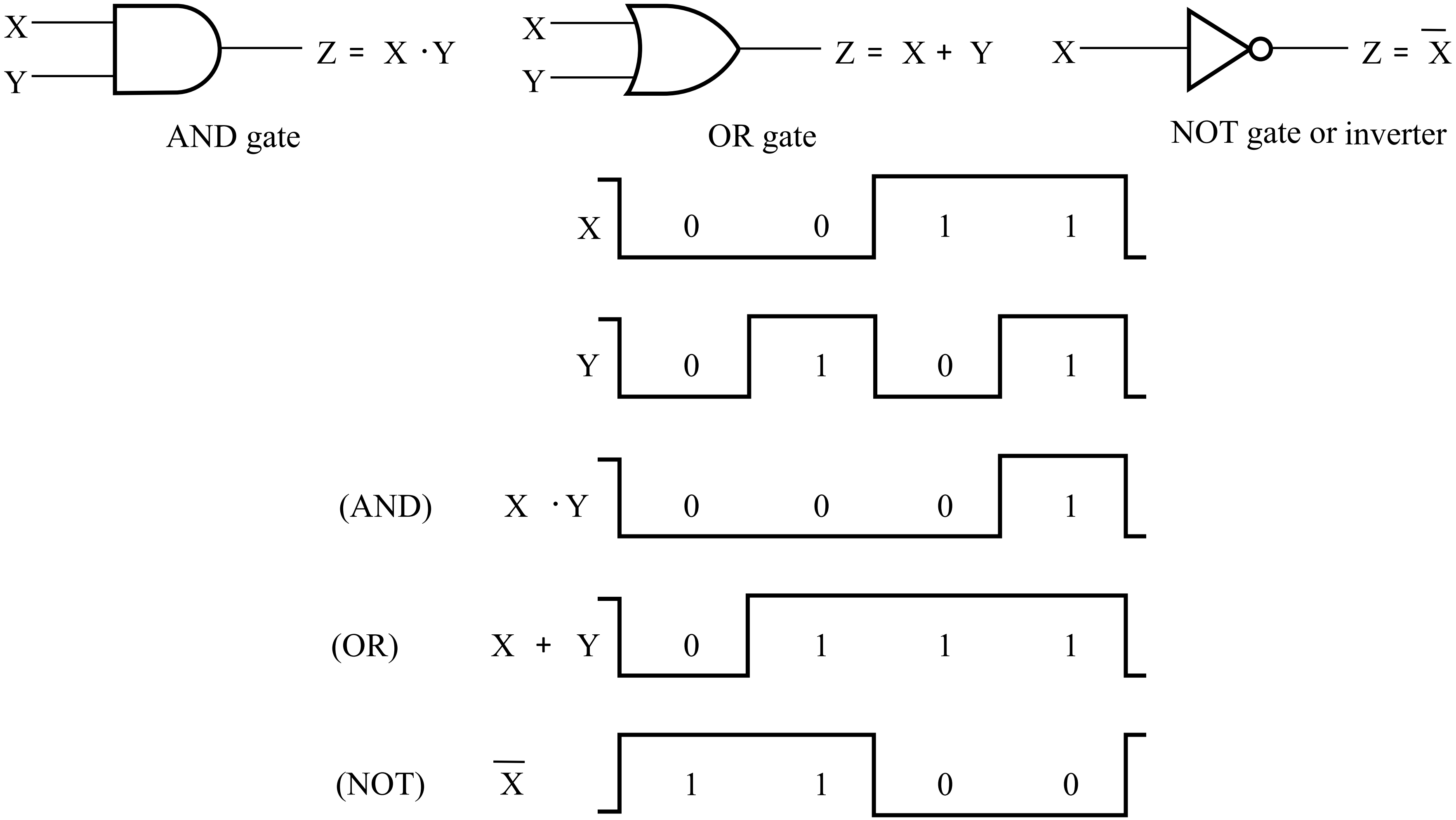
| X | Z |
|---|---|
| 0 | 1 |
| 1 | 0 |

# Review: Example

➢ *Evaluate the following logic function:* $F(X, Y, Z) = X\,Y + \overline{Y}\,Z$

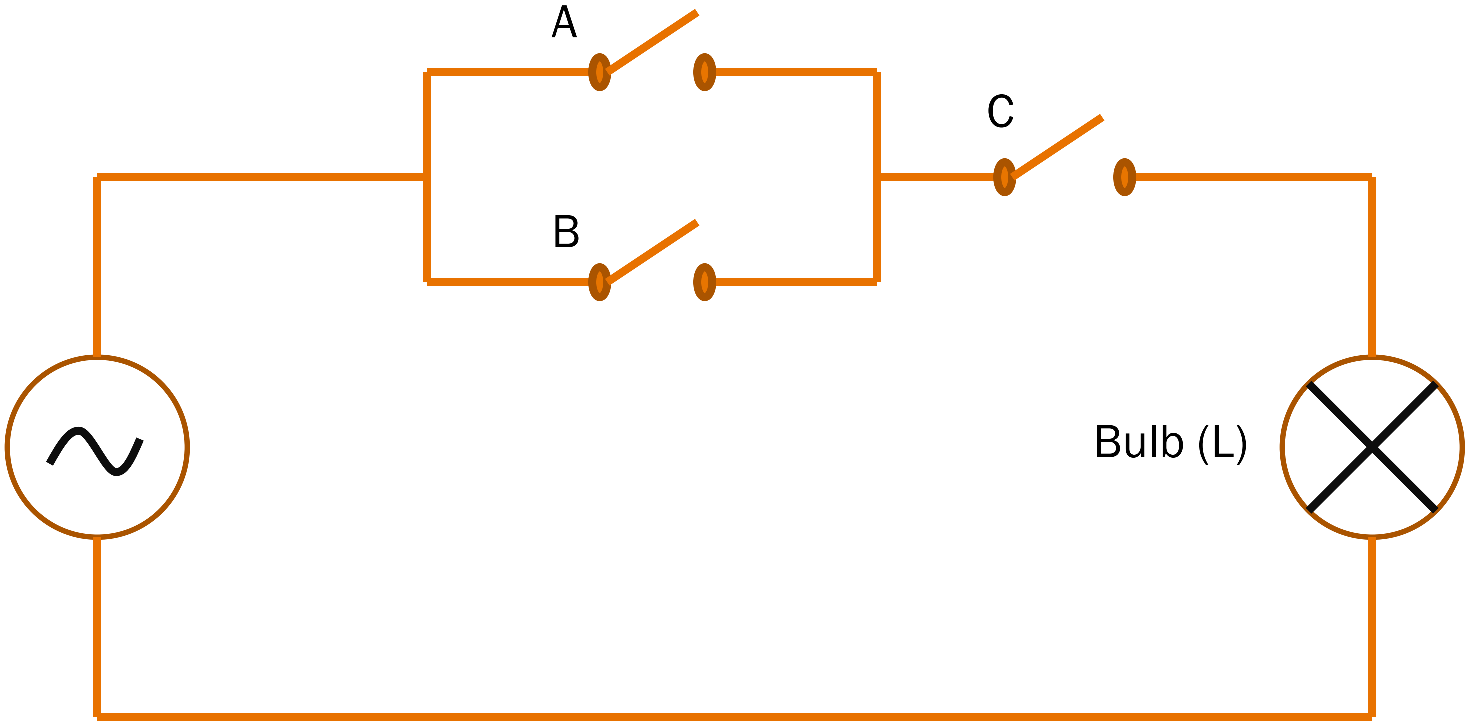
| X | Y | Z | XY | $\overline{Y}$ | $Z\overline{Y}$ | $XY + Z\overline{Y}$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 |

# Waveform Behavior View



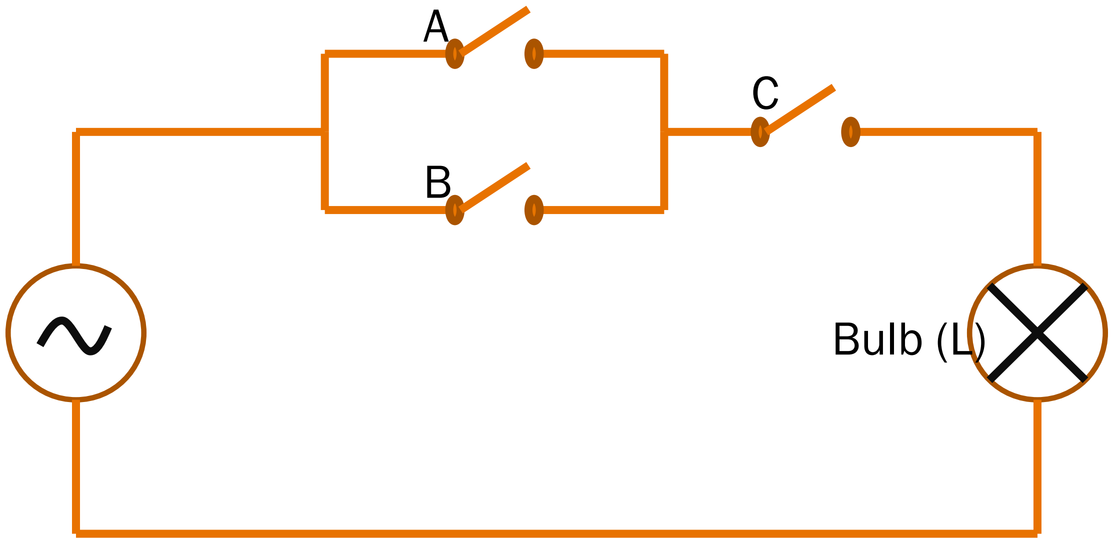
# Logic Function Example

- *Analyze the operation of the following circuit using the truth table. Write the logic function relating L to A, B, and C*



# Solution Using Truth Table

| A | B | C | L |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

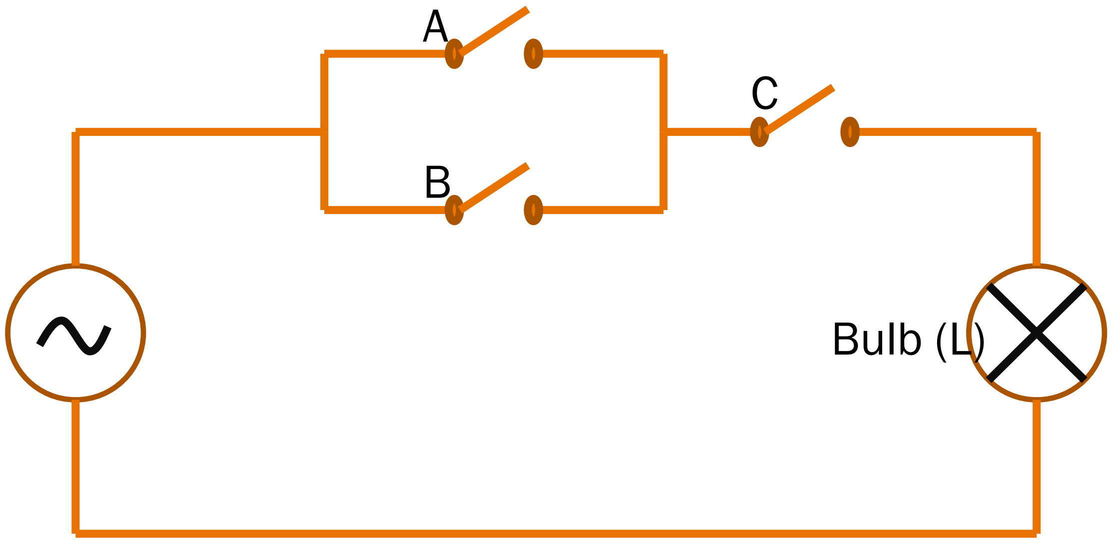


# Solution Using Algebraic Function

- The Light is on if:
- A AND C are closed, OR
- B AND C are closed, OR
- A AND B AND C are closed

L = A.C + B.C + A.B.C

Later we will see that L = C.(A+B) (OPTIMIZED)



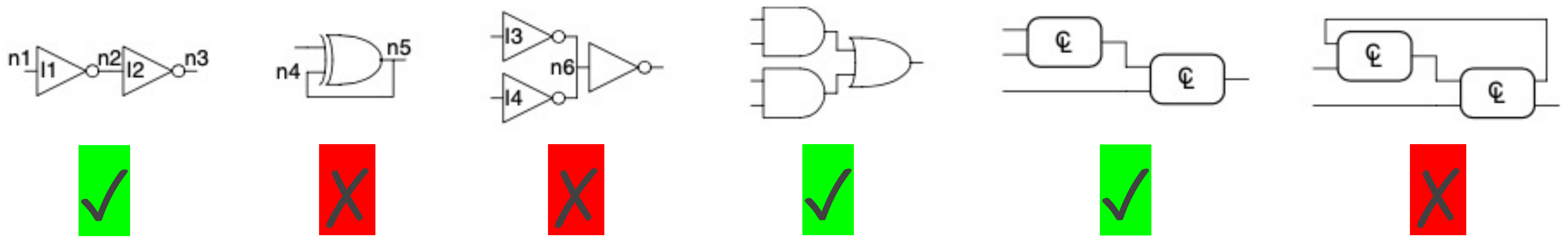
# Combinational Circuits

- Logic functions are implemented using digital logic circuits.
- A digital circuit is composed of:
  - Items: An item is a circuit which implements some logical function
  - Nodes: A node is a connection (wire) that connects to external input, output or between inner items.
- There are two classes of digital logic circuits:
  - Combinational circuits: the output depends only on the current values of the input (memoryless)
  - Sequential circuits: the output depends on both current and previous input value (has memory)

# Combinational Circuits

- Combinational circuits must meet three conditions:
  - Every circuit element is itself combinational.
  - Every node of the circuit is either designated as an input to the circuit or connects to exactly one output terminal of a circuit element
  - The circuit contains no cyclic paths: every path through the circuit visits each circuit node at most once.

Example: Determine the combinational circuits in the following



# Boolean Algebra

## ➢ *In Binary Boolean Algebra:*

- We have a set of elements 0 and 1
- We have a set of operators that operate on those elements such as AND (.), OR (+), and negation (’).
- We have 1’= 0 and 0’=1
- Logic value TRUE has value 1
- Logic value FALSE has value 0

# Operator Precedence

- Parenthesis ()
- NOT (Negation)
- AND
- OR
- Example: Show the order of execution of `NOT X + Y . Z`
  - Answer: `X′ + (Y.Z)`

# Boolean Expression

- Boolean expressions are formed by applying the operators to Boolean variables
- Boolean expressions (functions) could be represented by:
  - Equations
  - Logic gate diagram
  - Truth table

# Duality of Boolean Expressions

- The <u>dual</u> of an algebraic expression is obtained by interchanging + and · and interchanging 0's and 1's.
- Example: Assume: $F = (A + B) . C$, *find Dual(F)*
  - Dual(F) = $A . B + C$
- Dual of a function generates a totally new function
- Dual of a function is not equal to the function
- Dual of a function is not equal to its complement

# Duality Importance

- Every Boolean expression that is proven as true has a dual expression that is also true.
- Examples:

| *Expression* | *Duality* |
|---|---|
| $x + 0 = x$ | $x.1 = x$ |
| $x + x = x$ | $x.x = x$ |

# Postulates of Boolean Algebra

| | Expression | | Duality | Postulate |
|---|---|---|---|---|
| 1 | X + 0 = X | 2 | X . 1 = X | Identity Element |
| 3 | X + 1 = 1 | 4 | X . 0 = 0 | Domination |
| 5 | X + X = X | 6 | X . X = X | Idempotence |
| 7 | X + X’ = 1 | 8 | X . X’ = 0 | Complement |
| 9 | X’’ = X | | | Involution |
| 10 | X + Y = Y + X | 11 | X . Y = Y . X | Commutative |
| 12 | (X + Y) + Z = X + (Y + Z) | 13 | (X . Y) . Z = X . (Y . Z) | Associative |
| 14 | X. (Y + Z) = (X . Y) + (X . Z) | 15 | X + (Y . Z) = (X + Y) . (X + Z) | Distributive |
| 16 | (X + Y)’ = X’ . Y’ | 17 | (X . Y)’ = X’ + Y’ | DeMorgan’s |

# Additional Rules

| | Expression | | Duality | Postulate |
|---|---|---|---|---|
| 1 | $X + (X \cdot Y) = X$ | 2 | $X \cdot (X + Y) = X$ | Absorption |
| 3 | $X + (X' \cdot Y) = X + Y$ | 4 | $X \cdot (X' + Y) = X \cdot Y$ | |
| 5 | $(X + Y) \cdot (X + Y') = X$ | 6 | $(X \cdot Y) + (X \cdot Y') = X$ | Logical Adjacency |

# Algebraic Manipulation

- Example: $F = AB + AC + AB' + AC'$
  - $= AB + AB' + AC + AC'$
  - $= A(B + B') + A(C + C')$
  - $= A + A$
  - $= A$
- Example: $F = x(x' + y)$
- $= xx' + xy$
- $= 0 + xy$
- $= xy$

# Function Complement

- Complement of *F is F′*
- Complement of a function is obtained by interchange of 0's for 1's and 1's for 0's in the values of *F in the truth table.*
- We can also use DeMorgan's theorem to get *F′.*
- The following identities are very useful:
  - $(A + B)' = A' B'$ $\qquad$ $A'+B' = (AB)'$
  - $(A+B+C)' = A'B'C'$ $\qquad$ $A'+B'+C' = (ABC)'$

# Function Complement Example

- Find the complement of $F = AB + C$
- Answer:
  - $F' = (AB + C)'$
  - Applying DeMorgan's theorem:
  - $= (A' + B') . C'$
  - $= A'C' + B'C'$

# Function Complement From Dual

- We can find the complement of a function by finding its dual then complementing each literal
- Example: $F = x'yz' + x'y'z$
  - $F(dual) = (x' + y + z').(x' + y' + z)$
  - $F(complement) = F' = (x + y' + z)(x + y + z')$

# To Do List

- Review lecture notes
- Study 2.1 to 2.3.3