Digital Systems Design and Laboratory
[2. Boolean Algebra]

Chung-Wei Lin

cwlin@csie.ntu.edu.tw

CSIE Department

National Taiwan University

We would like to thank Prof. Hui-Ru Jiang (NTU) for contributing the lecture material

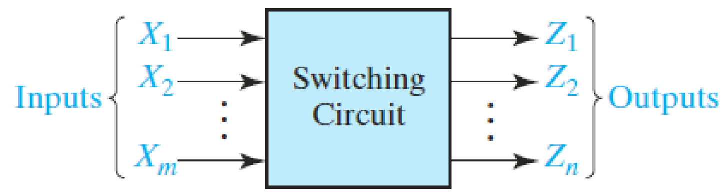
Outline

- **Introduction**
- Basic Operation
- Boolean Expressions and Truth Tables
- Basic Theorems
- Commutative, Associative, Distributive, and DeMorgan's Laws
- Simplification Theorems
- Multiplying Out and Factoring
- Complementing Boolean Expressions

Introduction

- **Boolean algebra**
  - Is the basic mathematics for logic design of digital systems
- **History**
  - George Boole developed Boolean algebra in 1847 and used it to solve problems in mathematical logic
  - Claude Shannon first applied Boolean algebra to the design of switching circuits in 1939
    - Master's thesis (21 years old)
- **Switching devices we will use are essentially two-state devices**
  - Represent an input or output by a Boolean variable
  - 1/0 for High/Low or True/False or Yes/No or Closed/Open
    - Just symbols
    - No numeric value

FIGURE 1-1
Switching Circuit

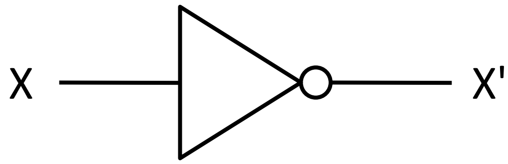


Outline

- ❑ Introduction
- ❑ **Basic Operation**
- ❑ Boolean Expressions and Truth Tables
- ❑ Basic Theorems
- ❑ Commutative, Associative, Distributive, and DeMorgan's Laws
- ❑ Simplification Theorems
- ❑ Multiplying Out and Factoring
- ❑ Complementing Boolean Expressions

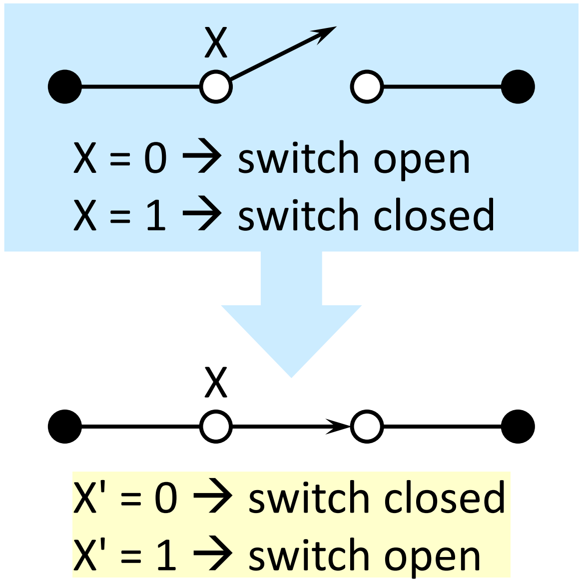
Logic NOT

- Complement = Inverse = Negate = NOT (' ; ¯ ; ~ ; ¬)
  - 0' = 1, 1' = 0
  - Symbol (NOT gate, inverter)



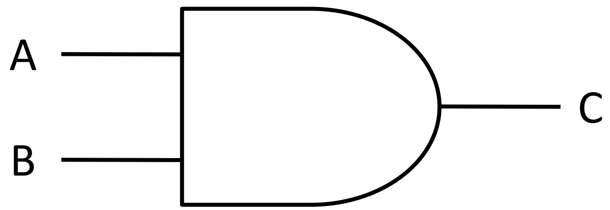
  - Truth table

X (Input)	X' (Output)
0	1
1	0



Logic AND

- AND (• ; ∧ ; sometimes omitted)
  - 0 • 0 = 0, 0 • 1 = 0, 1 • 0 = 0, 1 • 1 = 1
  - Symbol (AND gate)



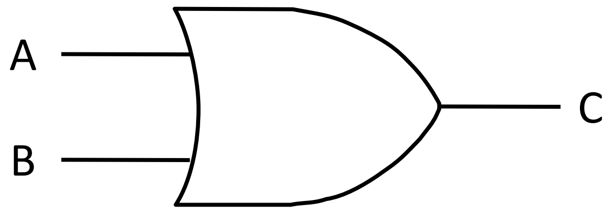
C = 0 → open circuit between 1 and 2
C = 1 → closed circuit between 1 and 2

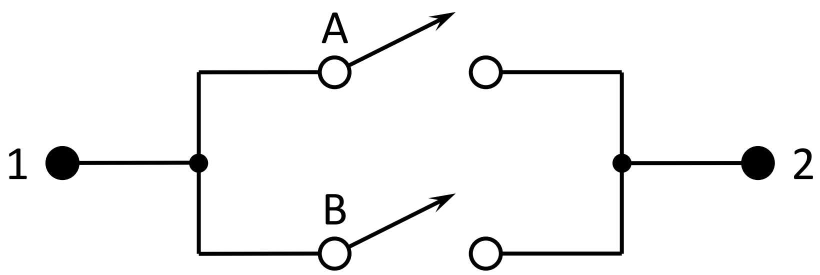
  - Truth table

A	B	C = A • B
0	0	0
0	1	0
1	0	0
1	1	1

Logic OR

- OR (+ ; ∨)
  - 0 + 0 = 0, 0 + 1 = 1, 1 + 0 = 1, 1 + 1 = 1
  - Symbol (OR gate)





C = 0 → open circuit between 1 and 2
C = 1 → closed circuit between 1 and 2

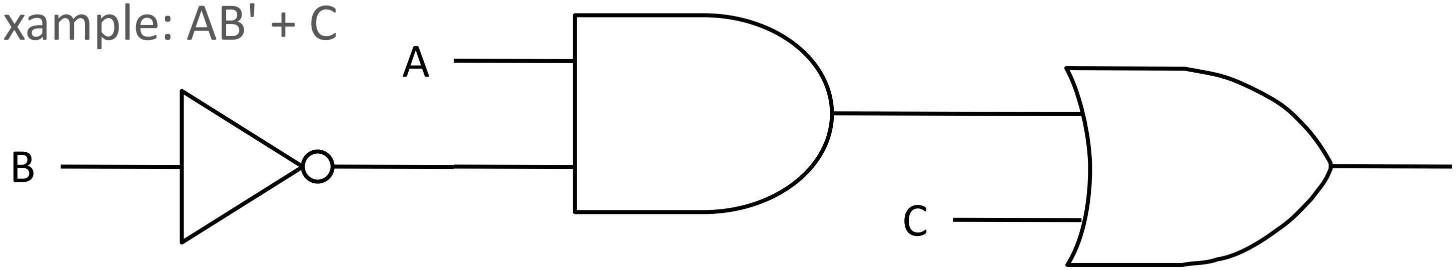
  - Truth table

A	B	C = A + B
0	0	0
0	1	1
1	0	1
1	1	1

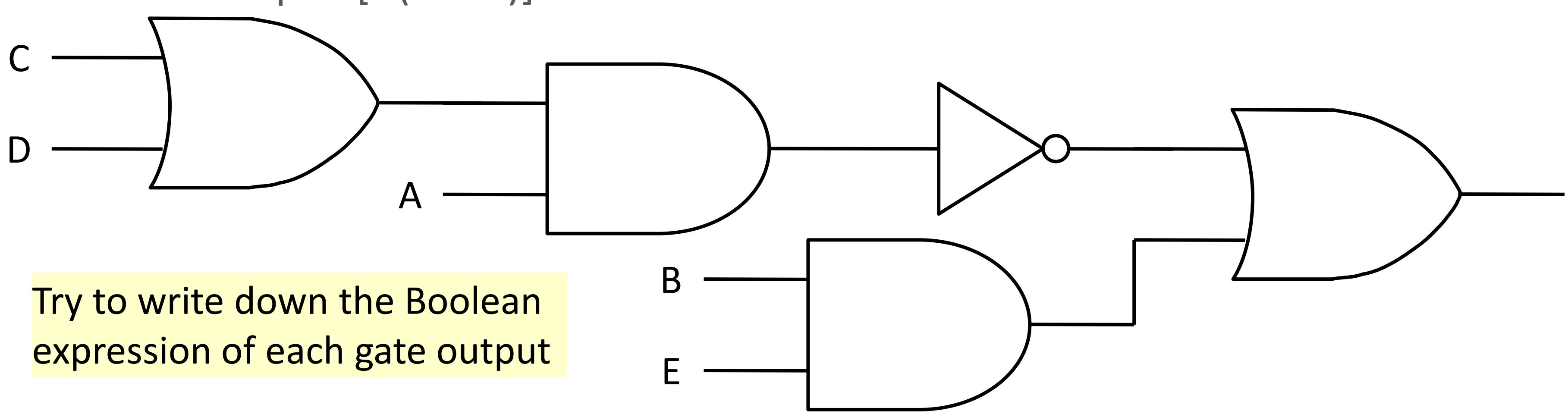
Outline

Boolean Expressions vs. Logic Gates

- A Boolean expression is formed by basic operations on constants or variables, e.g., 0, 1, X, Y'
- Realize a Boolean expression by a circuit of logic gates
  - Perform operations in order: parentheses → NOT → AND → OR
  - Example: AB' + C



  - Example: [A(C + D)]' + BE



Try to write down the Boolean expression of each gate output

Boolean Expressions vs. Truth Tables

- A truth table specifies the output values of a Boolean expression for all possible combinations of input values
  - How to check the equivalence between two expressions?
  - Example: AB' + C = (A + C)(B'+ C)

A	B	C	B'	AB'	LHS	A+C	B'+C	RHS
0	0	0	1	0	0	0	1	0
0	0	1	1	0	1	1	1	1
0	1	0	0	0	0	0	0	0
0	1	1	0	0	1	1	1	1
1	0	0	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1
1	1	0	0	0	0	1	0	0
1	1	1	0	0	1	1	1	1

Outline

- Introduction
- Basic Operation
- Boolean Expressions and Truth Tables
- **Basic Theorems**
- Commutative, Associative, Distributive, and DeMorgan's Laws
- Simplification Theorems
- Multiplying Out and Factoring
- Complementing Boolean Expressions

Basic Theorems

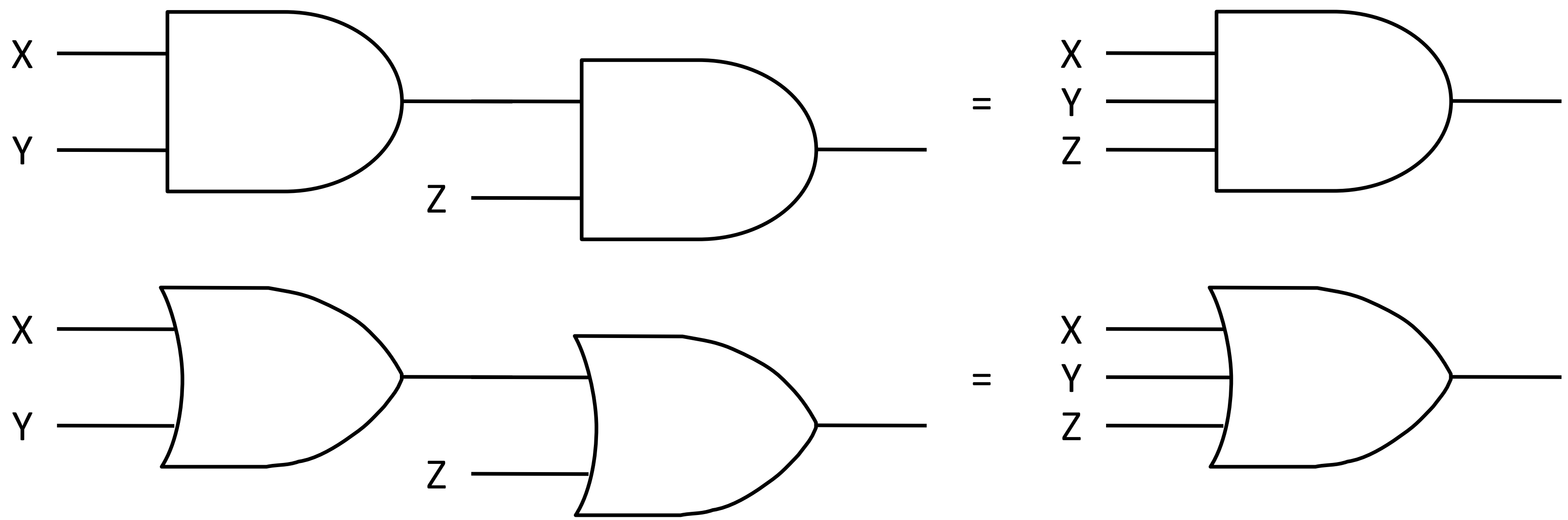
- Operations with 0 and 1
  - $X + 0 = X$
  - $X \bullet 1 = X$
  - $X + 1 = 1$
  - $X \bullet 0 = 0$
- Idempotent laws
  - $X + X = X$
  - $X \bullet X = X$
- Involution law
  - $(X')' = X$
- Laws of complementarity
  - $X + X' = 1$
  - $X \bullet X' = 0$

Outline

- Introduction
- Basic Operation
- Boolean Expressions and Truth Tables
- Basic Theorems
- **Commutative, Associative, Distributive, and DeMorgan's Laws**
- Simplification Theorems
- Multiplying Out and Factoring
- Complementing Boolean Expressions

Commutative and Associative Laws

- Commutative laws for AND and OR
  - XY = YX
  - X + Y = Y + X
- Associative laws for AND and OR
  - (XY)Z = X(YZ) = XYZ
  - (X + Y) + Z = X + (Y + Z) = X + Y + Z



Distributive and DeMorgan's Laws

- Distributive laws
  - Ordinary one : $X(Y + Z) = XY + XZ$
  - Second one: $\mathbf{X + YZ = (X + Y)(X + Z)}$
    - $X + YZ = X \bullet 1 + YZ = X(1 + Y + Z) + YZ = X + XY + XZ + YZ = XX + XY + XZ + YZ$
      $= XX + XZ + YX + YZ = X(X + Z) + Y(X + Z) = (X + Y)(X + Z)$
    - You can also use a truth table to prove it
- DeMorgan's laws
  - $(X + Y)' = X'Y'$
  - $(XY)' = X' + Y'$

Duality (1/2)

- The dual of a Boolean expression is obtained by
  - Interchanging the constants 0 and 1
  - Interchanging the operations of AND and OR
  - Leaving variables and complements unchanged
- Given a Boolean identity, another identity can be obtained by taking the dual of both sides of the identity

Duality (2/2)

Laws of Boolean algebra

Operations with 0 and 1	[1]	$X + 0 = X$	[1D]	$X \bullet 1 = X$
	[2]	$X + 1 = 1$	[2D]	$X \bullet 0 = 0$
Idempotent laws	[3]	$X + X = X$	[3D]	$X \bullet X = X$
Involution law	[4]	$(X')' = X$		
Laws of complementarity	[5]	$X + X' = 1$	[5D]	$X \bullet X' = 0$
Commutative laws	[6]	$X + Y = Y + X$	[6D]	$XY = YX$
Associative laws	[7]	$(X + Y) + Z = X + (Y + Z) = X + Y + Z$		
	[7D]	$(XY)Z = X(YZ) = XYZ$		
Distributive laws	[8]	$X(Y + Z) = XY + XZ$		
	[8D]	$X + YZ = (X + Y)(X + Z)$		
DeMorgan's laws	[9]	$(X + Y)' = X'Y'$		
	[9D]	$(XY)' = X' + Y'$		

Outline

- ❑ Introduction
- ❑ Basic Operation
- ❑ Boolean Expressions and Truth Tables
- ❑ Basic Theorems
- ❑ Commutative, Associative, Distributive, and DeMorgan's Laws
- ❑ **Simplification Theorems**
- ❑ Multiplying Out and Factoring
- ❑ Complementing Boolean Expressions

Simplification Theorems

- Uniting
  - $XY + XY' = X$
  - $(X + Y)(X + Y') = X$
- Absorption
  - $X + XY = X$
  - $X (X + Y) = X$
- Elimination
  - $X + X'Y = X + Y$
  - $X (X' + Y) = XY$
- Consensus
  - $XY + X'Z + YZ = XY + X'Z$
  - $(X + Y)(X' + Z)(Y + Z) = (X + Y)(X' + Z)$

Simplification Practices

- Simplify $Z = A'BC + A'$
- Simplify $Z = [A + B'C + D + EF][A + B'C + (D + EF)']$
- Simplify $Z = (AB + C)(B'D + C'E') + (AB + C)'$

Outline

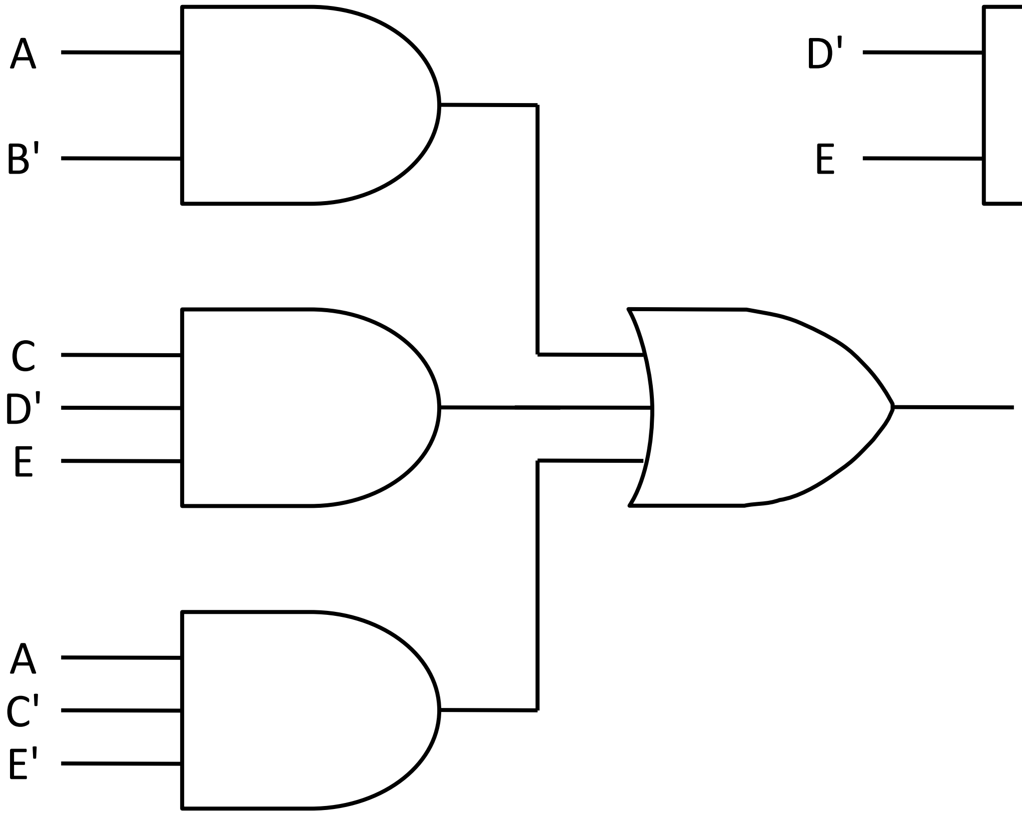
- Introduction
- Basic Operation
- Boolean Expressions and Truth Tables
- Basic Theorems
- Commutative, Associative, Distributive, and DeMorgan's Laws
- Simplification Theorems
- **Multiplying Out and Factoring**
- Complementing Boolean Expressions

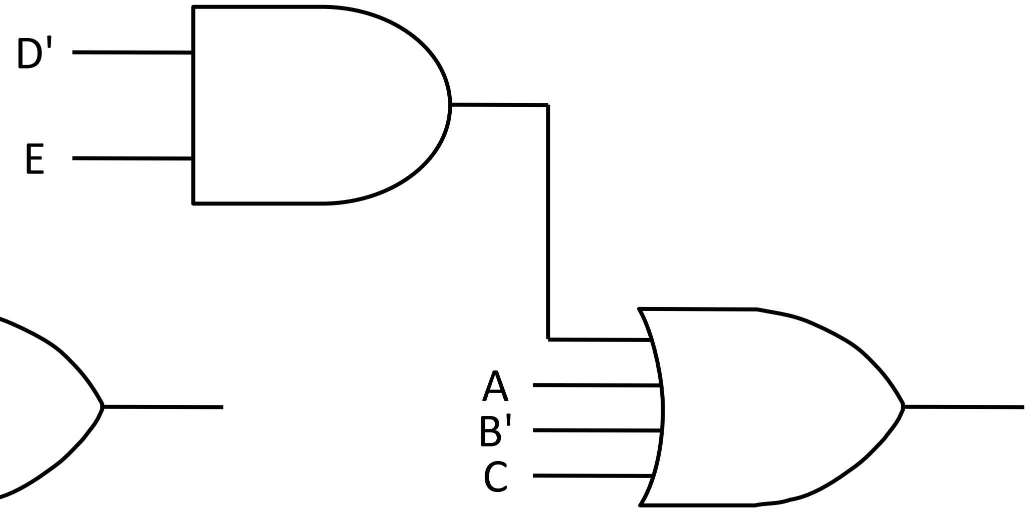
Multiplying Out

- Use the distributive laws to multiply out an expression to obtain a **sum-of-products** (SOP) form
  - Ordinary distributive law: X(Y + Z) = XY + XZ
  - Second distributive law: X + YZ = (X + Y)(X + Z)
- Example: multiply out (A + BC)(A + D + E)
  - Use the ordinary distributive law
    - (A + BC)(A + D + E) = A + AD + AE + ABC + BCD + BCE
      = A(1 + D + E + BC) + BCD + BCE
      = A + BCD + BCE
  - Use the second distributive law
    - (A + BC)(A + D + E) = A + BC(D + E) = A + BCD + BCE

SOP vs. Logic Gates

- Realize SOPs by two-level circuits (AND-OR)
  - AB' + CD'E + AC'E'
  - A + B' + C + D'E



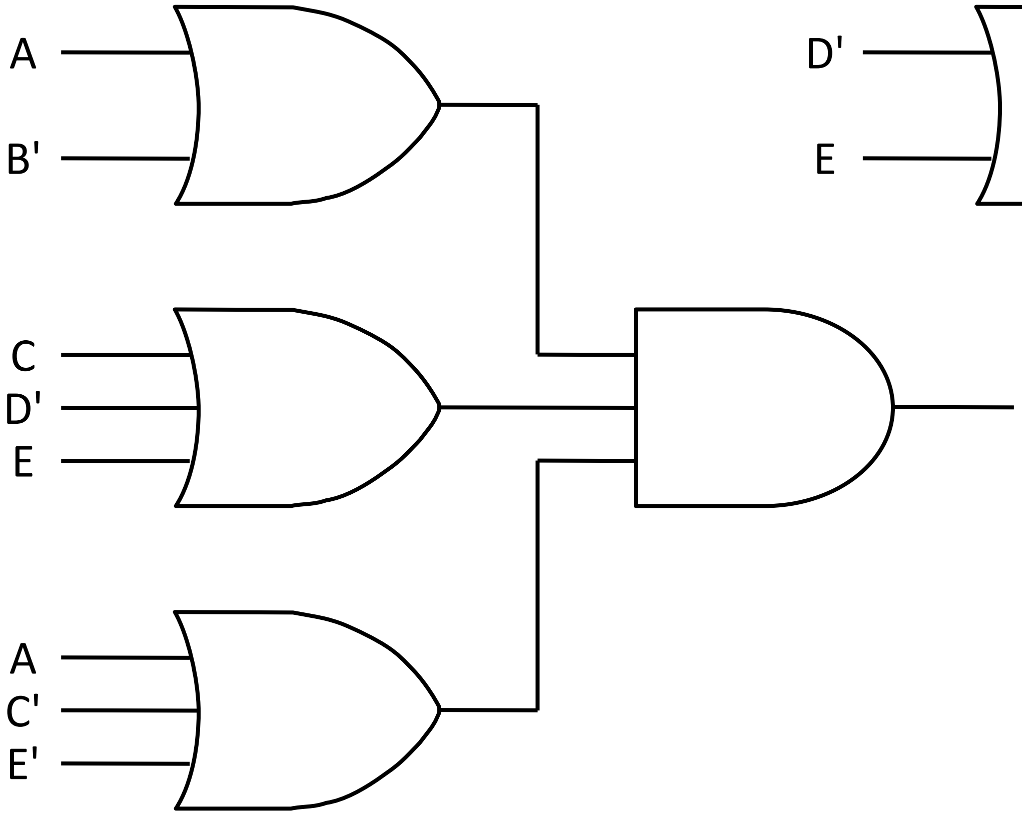


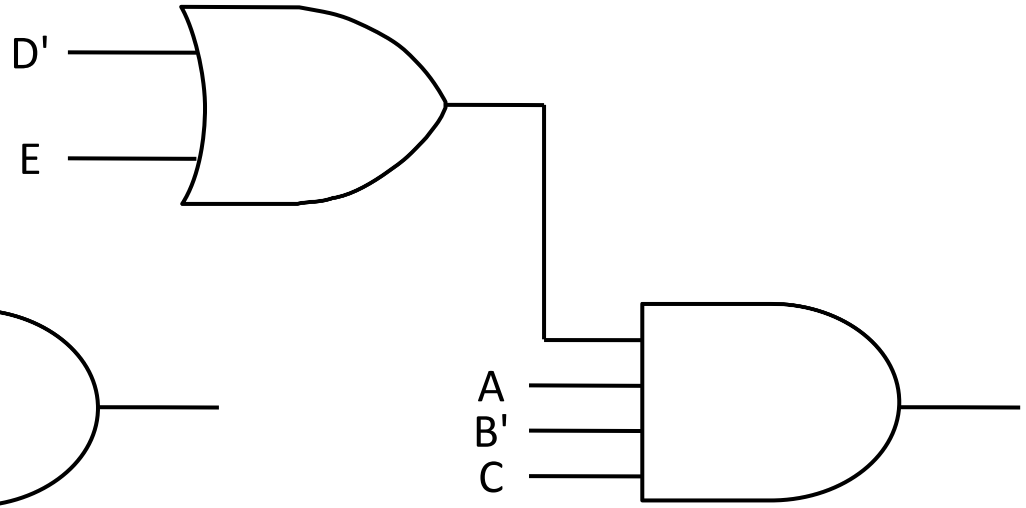
Factoring

- Use the second distributive law to factor an expression to obtain a **product-of-sums** (POS) form
  - X + YZ = (X + Y)(X + Z)
- Example: factor A + B'CD
- Example: factor AB'+ C'D
- Example: factor C'D + C'E' + G'H

POS vs. Logic Gates

- Realize POSs by two-level circuits (OR-AND)
  - (A + B')(C + D' + E)(A + C' + E')
  - AB'C(D' + E)





Outline

- ❑ Introduction
- ❑ Basic Operation
- ❑ Boolean Expressions and Truth Tables
- ❑ Basic Theorems
- ❑ Commutative, Associative, Distributive, and DeMorgan's Laws
- ❑ Simplification Theorems
- ❑ Multiplying Out and Factoring
- ❑ **Complementing Boolean Expressions**

Complementing Boolean Expressions

- DeMorgan's laws with n variables
  - $(X_1 + X_2 + \ldots + X_n)' = X_1'X_2' \ldots X_n'$
  - $(X_1X_2 \ldots X_n)' = X_1' + X_2' + \ldots + X_n'$
- Complement an expression by iteratively applying DeMorgan's laws
  - Example: complement (AB' + C)D' + E so that NOT is applied only to single variables
    - $[(A \bullet B' + C) \bullet D' + E]' = [(A \bullet B' + C) \bullet D']' \bullet E'$
      $= [(A \bullet B' + C)' + D] \bullet E'$
      $= [(A \bullet B')' \bullet C' + D] \bullet E'$
      $= [(A'+B) \bullet C' + D] \bullet E'$

Q&A