

# CPE201

# Digital Design

By Benjamin Haas

Class 12: Combinational Logic



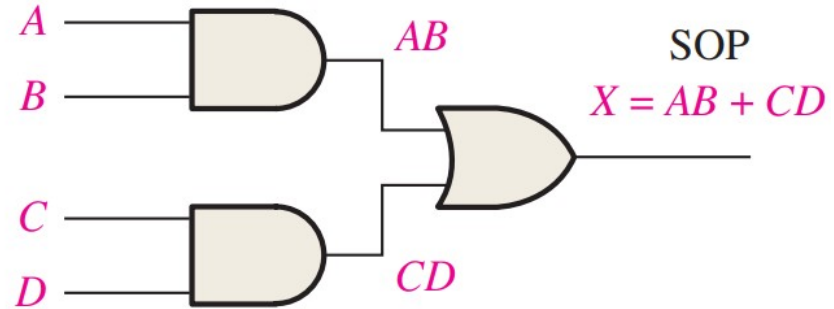
# Focus on Implementation

- Creating circuits from logic in Ch4
  - AND-OR Logic
  - AND-OR-Invert Logic
- Quick Review
  - XOR
  - XNOR



# AND-OR Logic

- What you already know from SOP expressions



- Example:  $AB + CD$

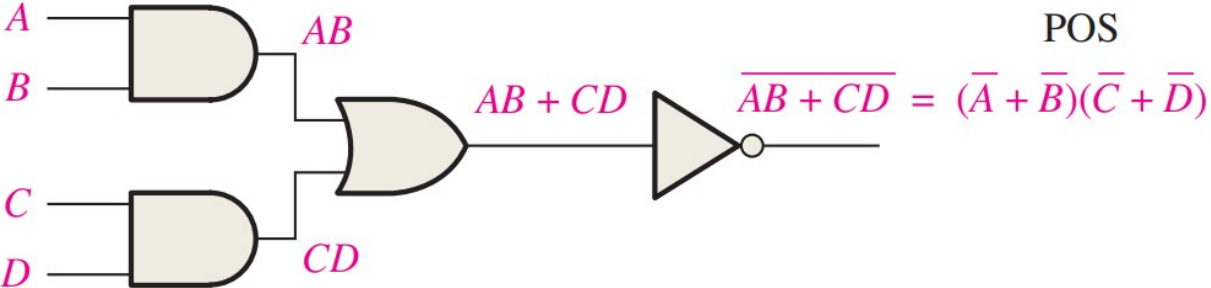


# AND-OR-Invert Logic

- Implements POS expressions
  - Heavy use of DeMorgan's
  - Implementation looks like SOP



# Example

- $(A' + B')(C' + D') = (AB)'(CD)' = ((AB)'(CD)')''$ 
    - Two NOTs would cancel, distribute one
  - $= (AB + CD)'$
- 
- The logic diagram illustrates the implementation of the expression  $(AB + CD)'$ . It consists of three main components: two AND gates and one OR gate followed by a NOT gate. The first AND gate takes inputs A and B, producing the output AB. The second AND gate takes inputs C and D, producing the output CD. These two outputs, AB and CD, are connected to the inputs of a third OR gate, which produces the output AB + CD. Finally, the output of the OR gate is connected to a NOT gate, which produces the final output  $\overline{AB + CD}$ . The expression  $\overline{AB + CD} = (\bar{A} + \bar{B})(\bar{C} + \bar{D})$  is shown next to the NOT gate, indicating that the output is in Product of Sums (POS) form.



# Example

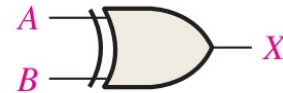
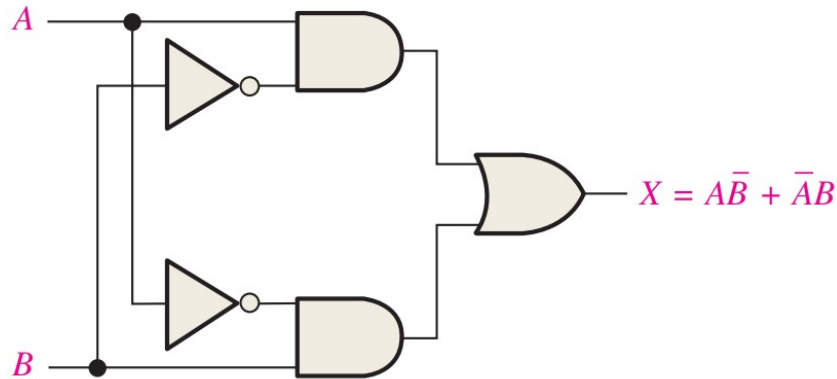
- Convert POS terms to Invert SOP terms
- $(A + B' + C')(A' + B' + C)(A' + B + C) =$
- $(A'BC + ABC' + AB'C')'$



# XOR

- Simplify your logic

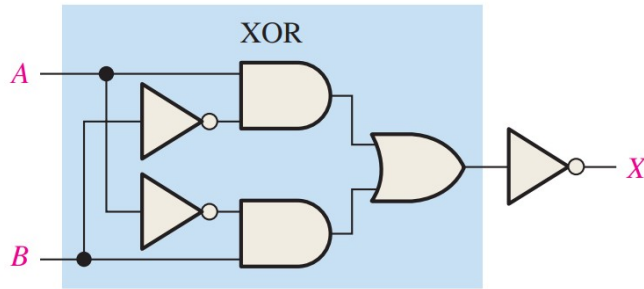
<i>A</i>	<i>B</i>	<i>X</i>
0	0	0
0	1	1
1	0	1
1	1	0



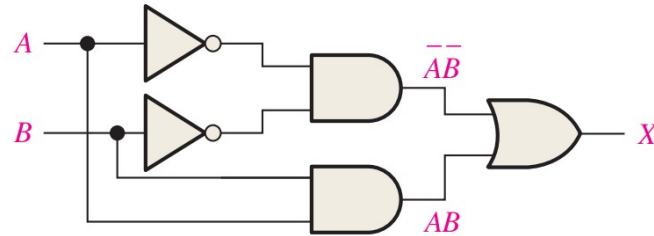
# XNOR

- Simplify your logic

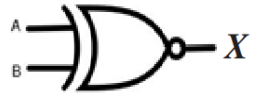
<i>A</i>	<i>B</i>	<i>X</i>
0	0	1
0	1	0
1	0	0
1	1	1



(a)  $X = \overline{A\overline{B}} + \overline{\overline{A}B}$



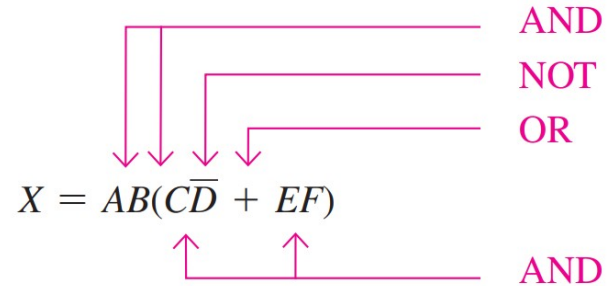
(b)  $X = \overline{\overline{A}B} + A\overline{B}$





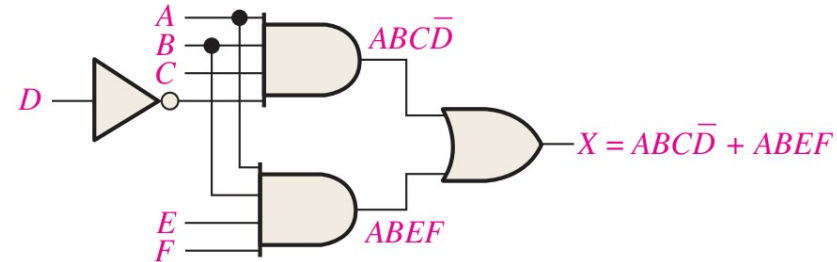
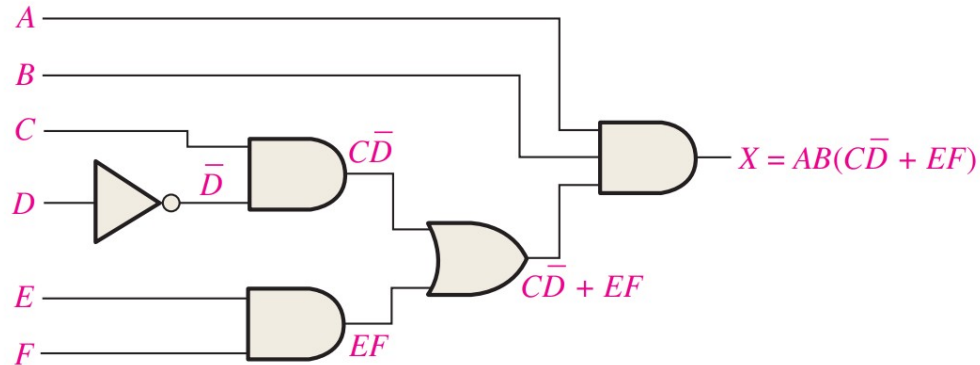
# Implementation

- Order of operations for Boolean expressions
  - Same as regular math expression symbols



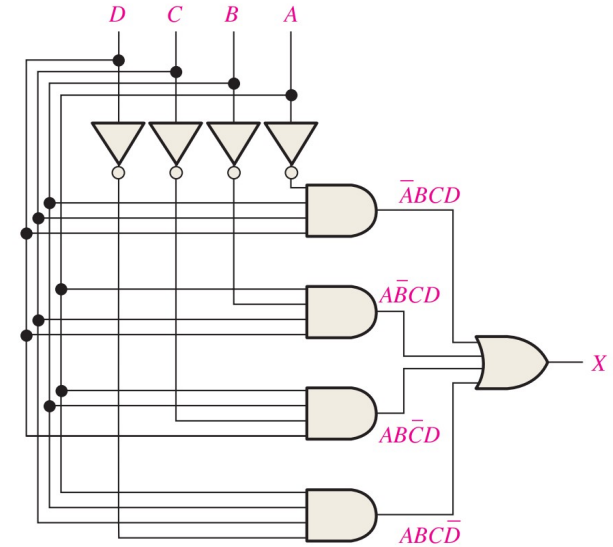
# Example

- $AB(CD' + EF) = ABCD' + ABEF$ 
  - Distributed to SOP

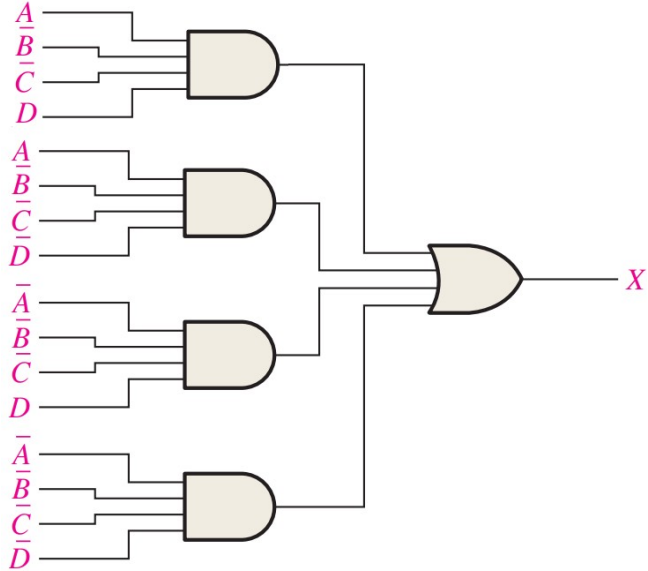


# Ye Olde Grid

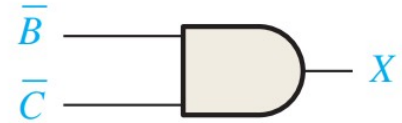
- Makes Standard SOP
  - Add a NOT and get Standard



# Minimize



AB \ CD	CD			
	00	01	11	10
00	1	1		
01				
11				
10	1	1		



# Universal Gates

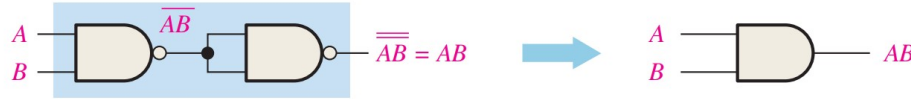
- NAND and NOR
  - You can make every other gate with them
- AND and OR are not universal
  - Can't make NOT



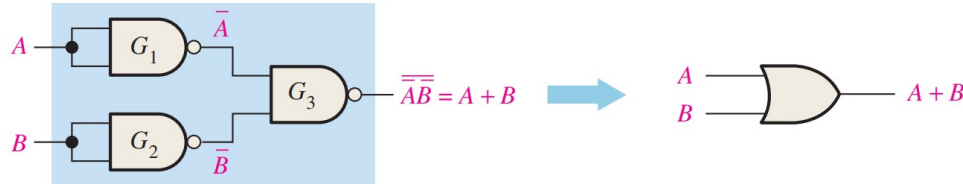
# NAND to Other Gates



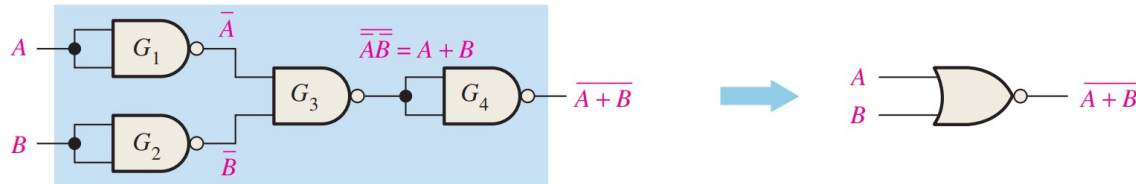
(a) One NAND gate used as an inverter



(b) Two NAND gates used as an AND gate



(c) Three NAND gates used as an OR gate



(d) Four NAND gates used as a NOR gate

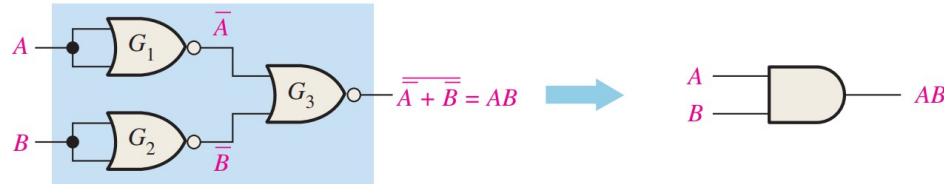
# NOR to Other Gates



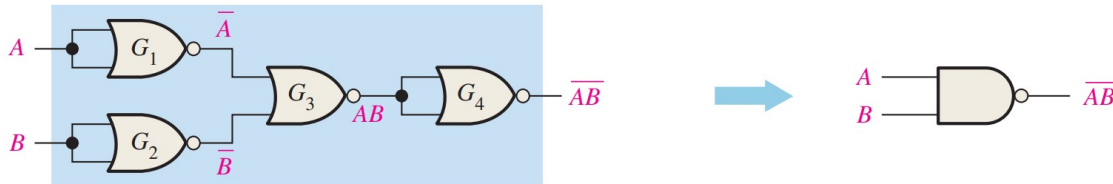
(a) One NOR gate used as an inverter



(b) Two NOR gates used as an OR gate



(c) Three NOR gates used as an AND gate



(d) Four NOR gates used as a NAND gate

# Reading

- This lecture
  - Sections 5.1-5.3
- Next lecture
  - Sections 5.4-5.5, 3.8

