Lecture 4

- Logistics
  - Classroom permanently changed to this one, EEB105
  - Lab2 is assigned today --- don't fall behind
  - HW1 is due on Wednesday in class before lecture
- Last lecture --- Boolean algebra
  - Axioms
  - Useful laws and theorems
  - Simplifying Boolean expressions
- Today's lecture
  - One more example of Boolean logic simplification
  - Logic gates and truth tables
  - Implementing logic functions

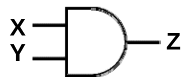
One more example of logic simplification

- Example:

$$Z = A'BC + AB'C' + AB'C + ABC' + ABC$$

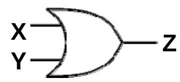
Logic gates and truth tables

- AND   X•Y   XY



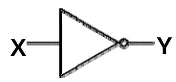
X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

- OR   X+Y



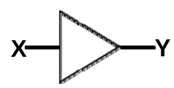
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

- NOT $\overline{X}$ X’



X	Y
0	1
1	0

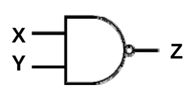
- Buffer X



X	Y
0	
1	

Logic gates and truth tables (con’t)

- NAND $\overline{X \bullet Y}$ $\overline{XY}$



X	Y	Z
0	0	
0	1	
1	0	
1	1	

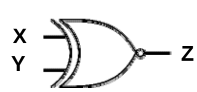
- NOR $\overline{X + Y}$

X	Y	Z
0	0	
0	1	
1	0	
1	1	

- XOR $X \oplus Y$

X	Y	Z
0	0	
0	1	
1	0	
1	1	

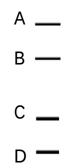
- XNOR $\overline{X \oplus Y}$



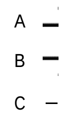
X	Y	Z
0	0	
0	1	
1	0	
1	1	

Boolean expressions ⇨ logic gates

- Example: F = (A•B)' + C•D



- Example: F = C•(A+B)'



Truth tables ⇨ logic gates

- Given a truth table
  - Write the Boolean expression
  - Minimize the Boolean expression
  - Draw as gates
  - Example:

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Example: A binary full adder

- 1-bit binary adder
  - Inputs: A, B, Carry-in
  - Outputs: Sum, Carry-out

A	B	Cin	S	Cout
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

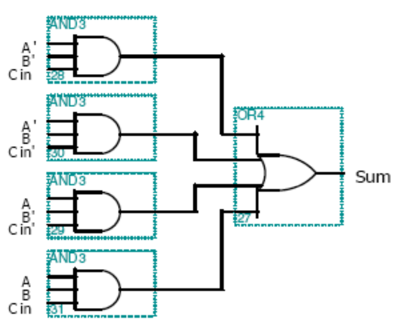
Sum =

Cout =

Full adder: Sum

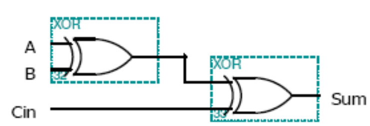
<u>Before Boolean minimization</u>

Sum = A'B'Cin + A'BCin'
+ AB'Cin' + ABCin



<u>After Boolean minimization</u>

Sum = (A⊕B) ⊕ Cin



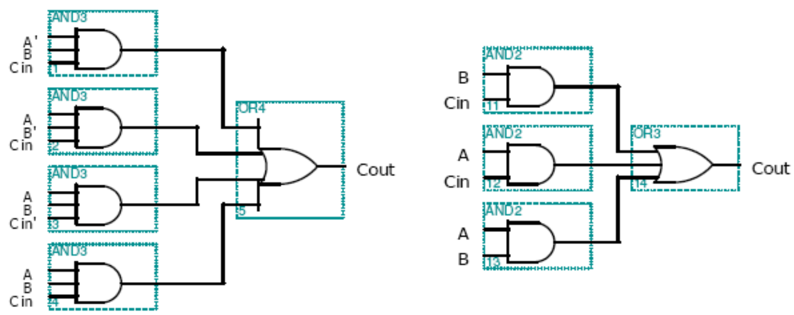
Full adder: Carry-out

Before Boolean minimization

Cout = A'BCin + AB'Cin
+ ABCin' + ABCin

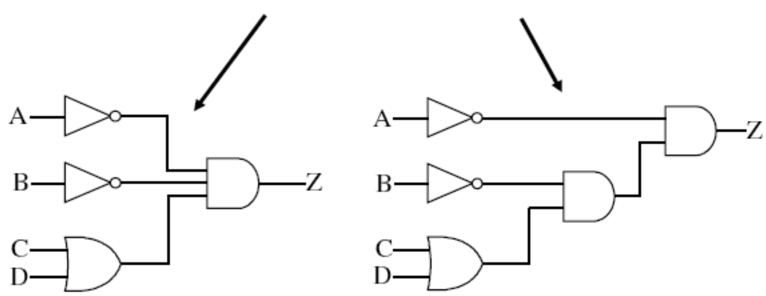
After Boolean minimization

Cout = BCin + ACin + AB



Many possible mappings

- Many ways to map expressions to gates
  - Example: $Z = \overline{A} \bullet \overline{B} \bullet (C + D) = \overline{A} \bullet \ \overline{B} \bullet (C + D)$



What is the optimal gate realization?

- We use the axioms and theorems of Boolean algebra to "optimize" our designs
- Design goals vary
  - Reduce the number of gates?
  - Reduce the number of gate inputs?
  - Reduce number of chips and/or wire?
- How do we explore the tradeoffs?
  - CAD tools
  - Logic minimization: Reduce number of gates and complexity
  - Logic optimization: Maximize speed and/or minimize power

Minimal set

- We can implement any logic function from NOT, NOR, and NAND
  - Example: (X and Y) = not (X nand Y)
- In fact, we can do it with only NOR or only NAND
  - NOT is just NAND or NOR with two identical inputs

X	Y	X nor Y
0	0	1
1	1	0

X	Y	X nand Y
0	0	1
1	1	0

  - NAND and NOR are duals: Can implement one from the other
    - X nand Y = not ((not X) nor (not Y))
    - X nor Y = not ((not X) nand (not Y))