ECE 2300
Digital Logic & Computer Organization

Fall 2016

Boolean Algebra

Cornell University

Boolean Algebra

- Mathematical tool for analyzing and simplifying logic circuits

- Boolean algebra (George Boole, 1854)
  - Two-valued algebraic system
  - Used to formulate true or false postulations

- Switching algebra (Claude Shannon, 1938)
  - Adopted Boolean algebra for digital circuits
  - Terms “Boolean algebra” and “switching algebra” are used interchangeably

Boolean Equations

- Describe digital functions
- `variable = expression`
- Variables are either 1 or 0
  - True or False
  - On or Off
  - Set or Reset (or Not Set)
  - Asserted or Deasserted
- Basic operators are AND, OR, and NOT

Operator Precedence

- What does W•X’+Y•Z mean?

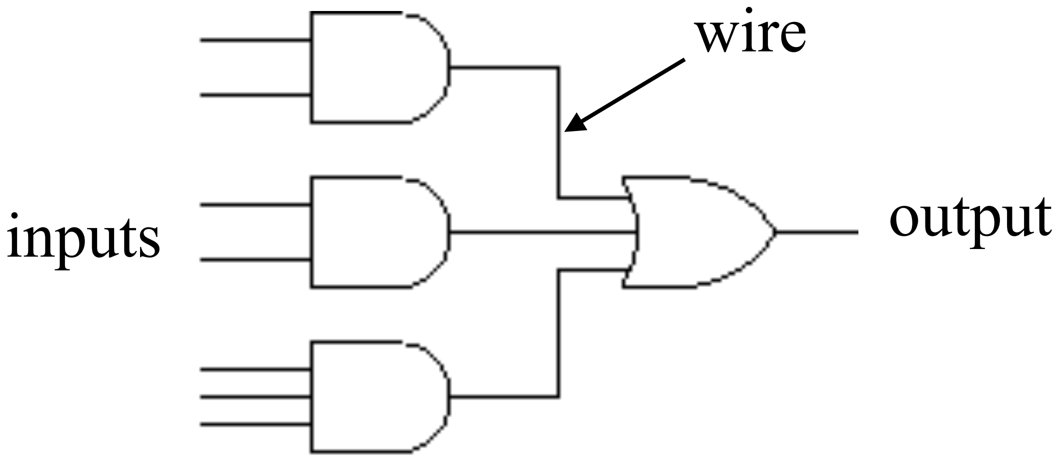
- Operator precedence rules
  1. NOT
  2. AND
  3. OR

Definitions

- Literal: variable or its complement
- Product term: AND of literals
  - X'•Y
- Sum term: OR of literals
  - X+Y+Z'

Sum-of-products

- OR of product terms
  - $X \bullet Y \bullet Z + Y' \bullet Z'$
- Circuits look something like this

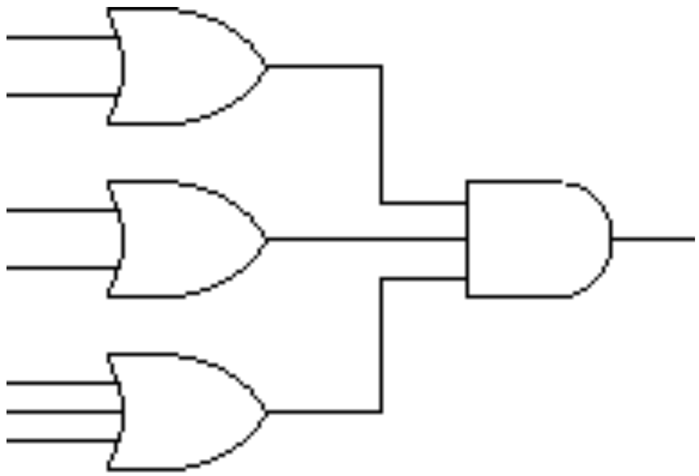


AND-OR

Product-of-sums

- **AND of sum terms**
  - (X+Y'+Z) • (Y+Z') • (X+Z') • (X+Y+Z')

- **Circuits look something like this**



OR-AND

Digital logic functions can be expressed as SOP or POS

More Definitions

- Normal term: Product or sum term in which every variable appears, and exactly once
- Minterm: Normal product
  - $X \cdot Y' \cdot Z$
- Maxterm: Normal sum
  - $X' + Y + Z'$

Minterms & Maxterms

XYZ	minterm	minterm name	maxterm	maxterm name
000	X'Y'Z'	m_0	X+Y+Z	M_0
001	X'Y'Z	m_1	X+Y+Z'	M_1
010	X'YZ'	m_2	X+Y'+Z	M_2
011	X'YZ	m_3	X+Y'+Z'	M_3
100	XY'Z'	m_4	X'+Y+Z	M_4
101	XY'Z	m_5	X'+Y+Z'	M_5
110	XYZ'	m_6	X'+Y'+Z	M_6
111	XYZ	m_7	X'+Y'+Z'	M_7

Two Ways to Express a Logic Function

- Canonical sum: The sum (OR) of minterms for which F=1
  - $F = X'\bullet Y'\bullet Z' + X'\bullet Y\bullet Z + X\bullet Y'\bullet Z' + X\bullet Y\bullet Z$
    $= \Sigma_{X,Y,Z}(0,3,4,7)$
- Canonical product: The product (AND) of maxterms for which F=0
  - $F = (X+Y+Z')\bullet(X+Y'+Z)\bullet(X'+Y+Z')\bullet(X'+Y'+Z)$
    $= \Pi_{X,Y,Z}(1,2,5,6)$
- $F = \Sigma_{X,Y,Z}(0,3,4,7) = \Pi_{X,Y,Z}(1,2,5,6)$

XYZ	F
000	1
001	0
010	0
011	1
100	1
101	0
110	0
111	1

Axioms of Boolean Algebra

- Definitions that are assumed true
- Obey the principle of duality
  - Interchange 1 and 0, AND and OR, still correct
  - Many axioms come in pairs

Axioms of Boolean Algebra

- Binary

(A1) X = 0 if X ≠ 1 (A1') X = 1 if X ≠ 0

- Complement

(A2) If X = 0, then X' = 1 (A2') If X = 1, then X' = 0

Other complement symbols: ~X, /X, $\overline{X}$

Axioms of Boolean Algebra

- AND and OR

(A3)	$0 \bullet 0 = 0$	(A3')	$1+1=1$
(A4)	$1 \bullet 1 = 1$	(A4')	$0+0=0$
(A5)	$0 \bullet 1 = 1 \bullet 0 = 0$	(A5')	$1+0=0+1=1$

- A1-A5 completely define Boolean algebra
  - Everything else derived from these axioms

Single Variable Theorems

- Identity: (T1) $X \cdot 1 = X$ (T1') $X + 0 = X$
- Null Element: (T2) $X \cdot 0 = 0$ (T2') $X + 1 = 1$
- Idempotency: (T3) $X \cdot X = X$ (T3') $X + X = X$
- Involution: (T4) $(X')' = X$
- Complements: (T5) $X \cdot X' = 0$ (T5') $X + X' = 1$

- Can prove by ***perfect induction***
  - **Show that all possible inputs meet the theorem**

Proof by Perfect Induction

(T3) X•X=X

X=0 → 0•0=0

X=1 → 1•1=1

(T3') X+X=X

X=0 → 0+0=0

X=1 → 1+1=1

(T5) X•X'=0

(T5') X+X'=1

Two and Three Variable Theorems

- **Commutativity**

  (T6) $X \cdot Y = Y \cdot X$ (T6') $X+Y = Y+X$

- **Associativity**

  (T7) $(X \cdot Y) \cdot Z = X \cdot (Y \cdot Z)$ (T7') $(X+Y)+Z = X+(Y+Z)$

- **Distributivity**

  (T8) $X \cdot Y + X \cdot Z = X \cdot (Y+Z)$ (T8') $(X+Y) \cdot (X+Z) = X+(Y \cdot Z)$

  AND distributes over OR OR distributes over AND

Two and Three Variable Theorems

- Covering

  (T9) $X \bullet (X+Y) = X$ (T9') $X + X \bullet Y = X$

- Combining

  (T10) $X \bullet Y + X \bullet Y' = X$ (T10') $(X+Y) \bullet (X+Y') = X$

- Consensus

  (T11) $X \bullet Y + X' \bullet Z + Y \bullet Z = X \bullet Y + X' \bullet Z$

  (T11') $(X+Y) \bullet (X'+Z) \bullet (Y+Z) = (X+Y) \bullet (X'+Z)$

Duality

- An' is the dual of An, Tn' is the dual of Tn

- Deriving the dual
  - Use parentheses to denote operator precedence
  - Swap 0's and 1's, AND's and OR's

Duality Derivation Examples

- Derive T9' from T9

$$X+X \bullet Y = X \quad \text{(T9)}$$

$$X+(X \bullet Y) = X \quad \text{(precedence)}$$

$$X \bullet (X+Y) = X \quad \text{(T9')}$$

- Derive T11' from T11

$$X \bullet Y+X' \bullet Z+Y \bullet Z = X \bullet Y+X' \bullet Z \quad \text{(T11)}$$

$$(X \bullet Y)+(X' \bullet Z)+(Y \bullet Z) = (X \bullet Y)+(X' \bullet Z) \quad \text{(precedence)}$$

$$(X+Y) \bullet (X'+Z) \bullet (Y+Z) = (X+Y) \bullet (X'+Z) \quad \text{(T11')}$$

De Morgan's Theorem

- So important, also known as De Morgan's *Law*

(T12) $(X1 \cdot X2 \cdot \ldots \cdot Xn)' = X1' + X2' + \ldots + Xn'$

(T12') $(X1 + X2 + \ldots + Xn)' = X1' \cdot X2' \cdot \ldots \cdot Xn'$

De Morgan Example

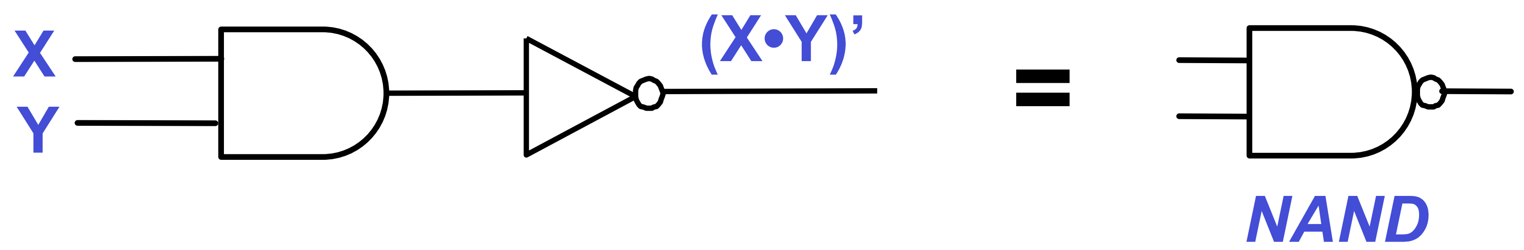
- By DeMorgan's Law

  (X•Y•Z)' = X'+Y'+Z'

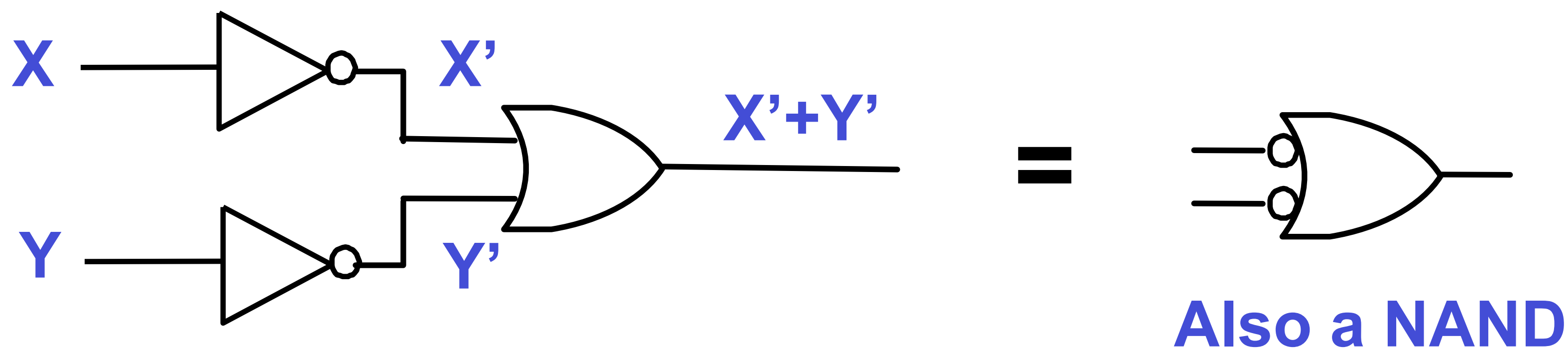
- Proof by perfect induction

XYZ	(X•Y•Z)'	X'+Y'+Z'
000	1	1
001	1	1
010	1	1
011	1	1
100	1	1
101	1	1
110	1	1
111	0	0

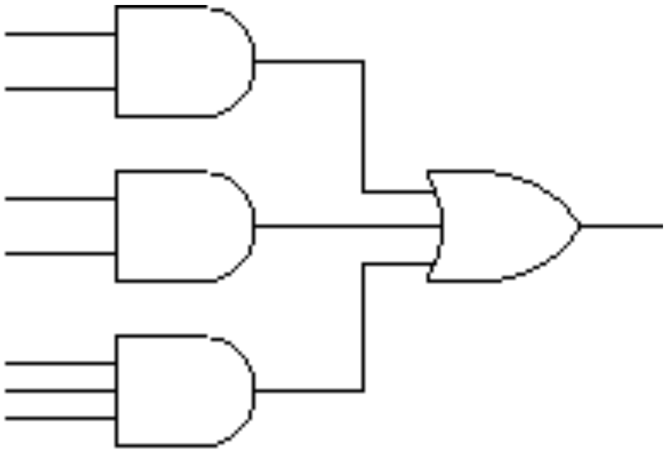
NAND Logic Gate



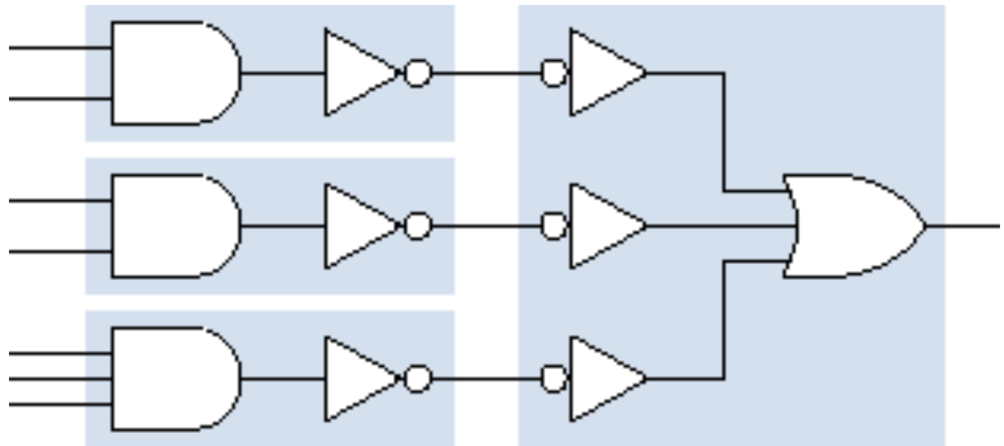
Using De Morgan's Law: **(X•Y)' = X'+Y'**

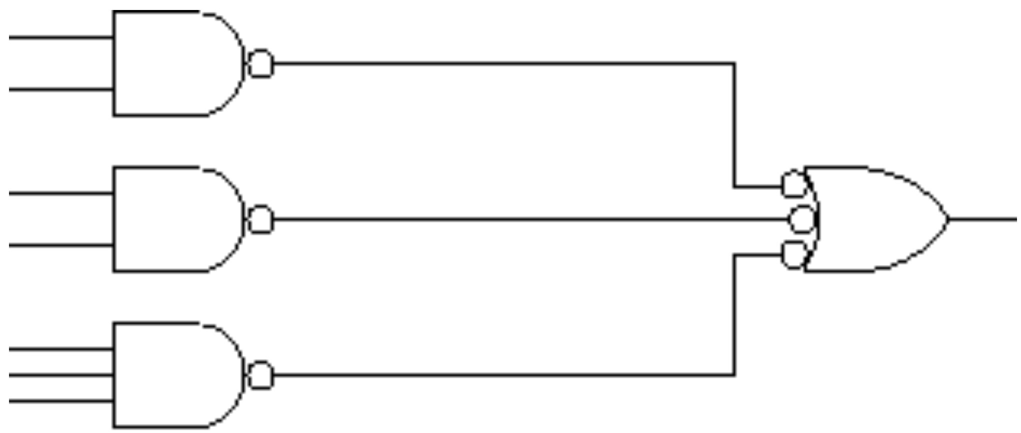


Sum-of-products Revisited



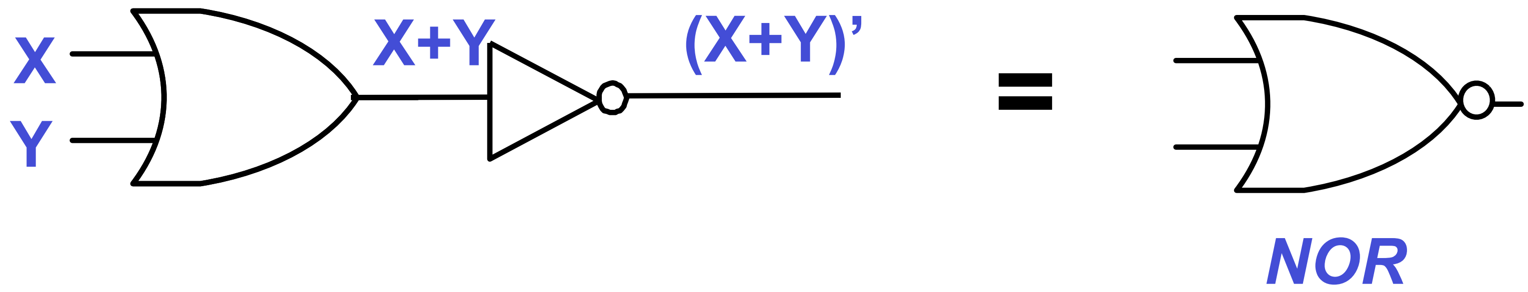
AND-OR



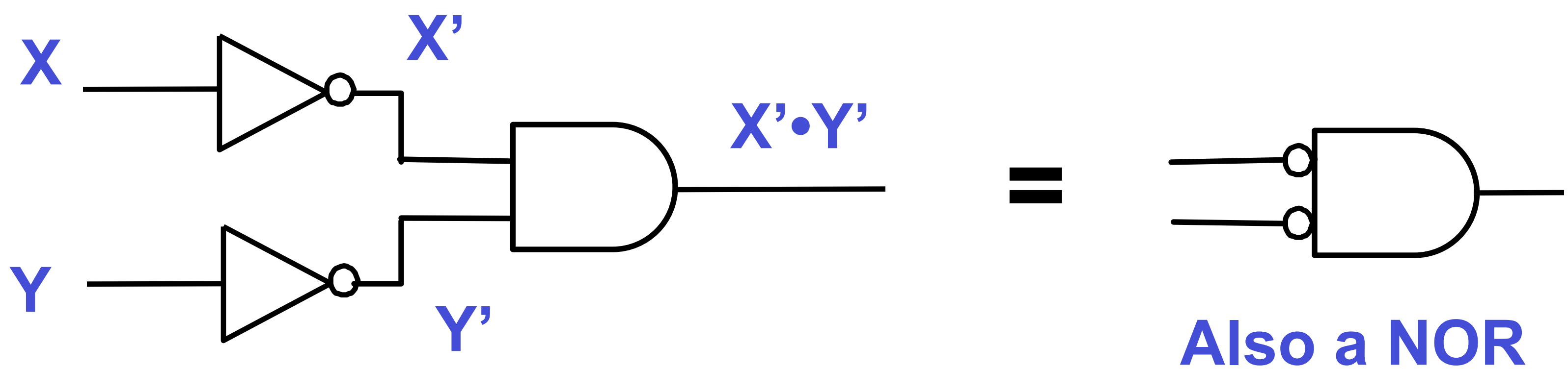


NAND-NAND

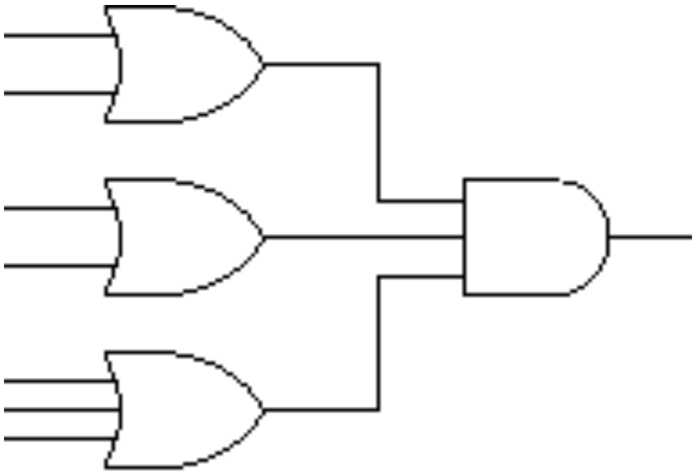
NOR Logic Gate



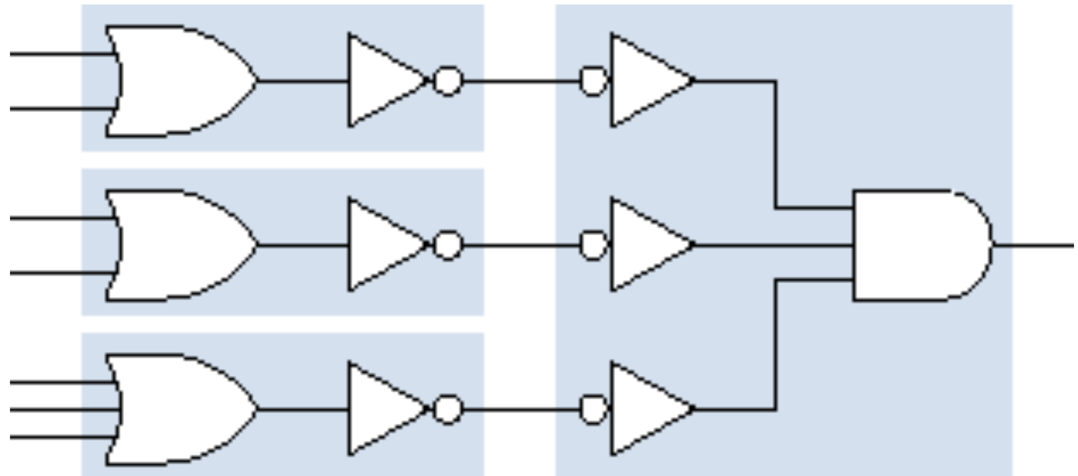
Using De Morgan's Law: **(X+Y)' = X'•Y'**

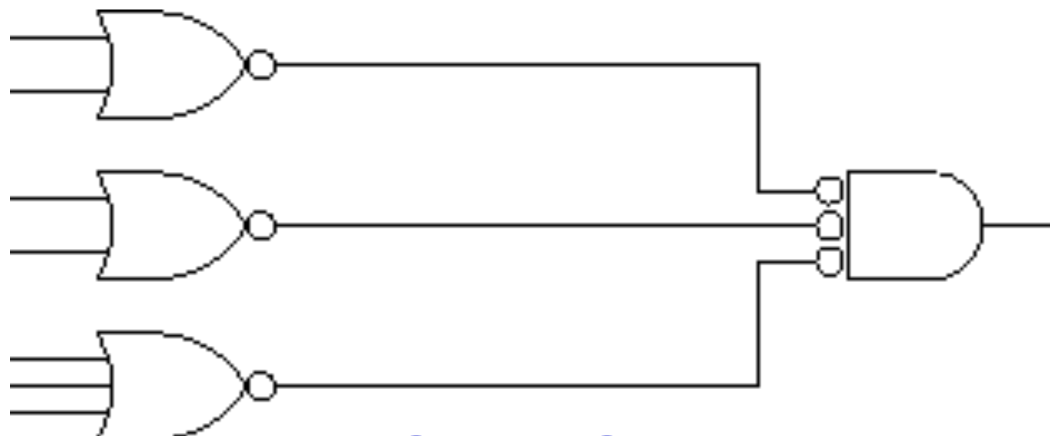


Product-of-sums Revisited



OR-AND





NOR-NOR

Two Ways to Express a Logic Function

- **Canonical sum: The sum of minterms for which F=1**
  - $F = X'\cdot Y'\cdot Z' + X'\cdot Y\cdot Z + X\cdot Y'\cdot Z' + X\cdot Y\cdot Z$

    $= \Sigma_{X,Y,Z}(0,3,4,7)$
- **Canonical product: The product of maxterms for which F=0**
  - $F = (X+Y+Z')\cdot(X+Y'+Z)\cdot(X'+Y+Z')\cdot(X'+Y'+Z)$

    $= \Pi_{X,Y,Z}(1,2,5,6)$
- $F = \Sigma_{X,Y,Z}(0,3,4,7) = \Pi_{X,Y,Z}(1,2,5,6)$

XYZ	F
000	1
001	0
010	0
011	1
100	1
101	0
110	0
111	1

Algebraic Simplification

- Apply theorems to canonical sum (product) to reduce (1) the number of terms, and (2) the number of literals in each term
- Results in a more compact expression and lower cost digital logic implementation

Algebraic Simplification Example

- 1-bit binary adder
  - inputs: A, B, Carry-in
  - outputs: Sum, Carry-out

- Truth Table → Canonical sum

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

S = A'•B'•Cin + A'•B•Cin' + A•B'•Cin' + A•B•Cin

Cout = A'•B•Cin + A•B'•Cin + A•B•Cin' + A•B•Cin

Idempotency and Combining Theorems

- Idempotency: $X+X = X$
- Combining: $X \cdot Y+X \cdot Y' = X$

Algebraic Simplification Example

Cout = A'•B•Cin + A•B'•Cin + A•B•Cin' + A•B•Cin

= A'•B•Cin + A•B'•Cin + A•B•Cin' + A•B•Cin + A•B•Cin (idempotency)

= B•Cin + A•B'•Cin + A•B•Cin' + A•B•Cin (combining)

= B•Cin + A•B'•Cin + A•B•Cin' + A•B•Cin + A•B•Cin (idempotency)

= B•Cin + A•Cin + A•B•Cin' + A•B•Cin (combining)

= B•Cin + A•Cin + A•B (combining)

Reduction in Hardware Cost

Cout = A'•B•Cin + A•B'•Cin + A•B•Cin' + A•B•Cin

= B•Cin + A•Cin + A•B

3 inverters
4 three-input ANDs
1 four-input OR

→

3 two-input ANDs
1 three-input OR

Example: Prime Number Detector

- F = 1 if number xyz is prime

- Step 1: Lay out truth table

- Step 2: Derive canonical form
  - $F = \sum_{x,y,z}(1,2,3,5,7) = x'y'z + x'yz' + x'yz + xy'z + xyz$
  - $F = \prod_{x,y,z}(0,4,6) = (x+y+z)(x'+y+z)(x'+y'+z)$

- Step 3: Simplify expression
  - Algebraic simplification
  - Systematic minimization (next time)

xyz	F
000	0
001	1
010	1
011	1
100	0
101	1
110	0
111	1

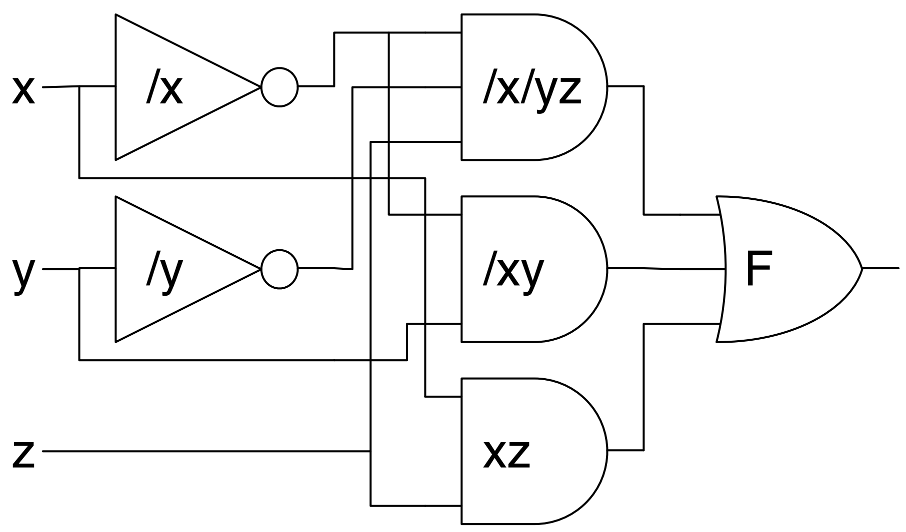
Algebraic Simplification

- $F = \sum_{x,y,z}(1,2,3,5,7)$

  $= x'y'z + x'yz' + x'yz + xy'z + xyz$

  $= x'y'z + x'y + xy'z + xyz$ [combining]

  $= x'y'z + x'y + xz$ [combining]



Before Next Class

- H&H 2.4-2.7

Next Time

Combinational Logic Minimization