數位電路設計

Digital Circuit Design

莊仁輝

交通大學資訊工程系

Chapter 2

Boolean Algebra and Logic Gates

2.3 Axiomatic Definitions of Boolean Algebra

- An algebraic structure defined by a set of elements B and two binary operators + and ·, providing the following postulates are satisfied (Boolean Algebra, E. V. Huntington, 1904):

  1. Closure w.r.t. the operator + (·)
  2. An identity element w.r.t. + (·)
  3. Commutative w.r.t. + (·)
  4. · is distributive over +

     + is distributive over ·
  5. $\forall\, x \in$ B, $\exists\, x' \in$ B (complement of x)
  6. $\exists$ at least two elements $x, y \in$ B such that $x \neq y$

2.3 Axiomatic Definitions of Boolean Algebra

- Two-valued Boolean Algebra
  - B = {0,1} $\rightarrow$ Postulate 6
  - The rules of operations

AND

x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

OR

x	y	$x+y$
0	0	0
0	1	1
1	0	1
1	1	1

NOT

x	x'
0	1
1	0

Postulate 1

2.4 Basic Theorems and Properties

x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

x	y	$x+y$
0	0	0
0	1	1
1	0	1
1	1	1

x	x'
0	1
1	0

Table 2.1
Postulates and Theorems of Boolean Algebra

Postulate 2	(a)	$x + 0 = x$	(b)	$x \cdot 1 = x$
Postulate 5	(a)	$x + x' = 1$	(b)	$x \cdot x' = 0$
Theorem 1	(a)	$x + x = x$	(b)	$x \cdot x = x$
Theorem 2	(a)	$x + 1 = 1$	(b)	$x \cdot 0 = 0$
Theorem 3, involution		$(x')' = x$		
Postulate 3, commutative	(a)	$x + y = y + x$	(b)	$xy = yx$
Theorem 4, associative	(a)	$x + (y + z) = (x + y) + z$	(b)	$x(yz) = (xy)z$
Postulate 4, distributive	(a)	$x(y + z) = xy + xz$	(b)	$x + yz = (x + y)(x + z)$
Theorem 5, DeMorgan	(a)	$(x + y)' = x'y'$	(b)	$(xy)' = x' + y'$
Theorem 6, absorption	(a)	$x + xy = x$	(b)	$x(x + y) = x$

- Duality between (a) and (b): AND OR, 1 0

2.4 Basic Theorems and Properties

- Theorem 1(a): x+x = x
  - $x+x = (x+x)\,1$ by postulate: 2(b)
    $= (x+x)(x+x')$ 5(a)
    $= x+xx'$ 4(b)
    $= x+0$ 5(b)
    $= x$ 2(a)
- Theorem 1(b): x x = x
  - $xx = x\,x + 0$
    $= xx + xx'$
    $= x\,(x + x')$
    $= x\,1$
    $= x$

2.4 Basic Theorems and Properties

- Theorem 2
  - $x + 1 = 1$ $(x + 1)$
    $= (x + x')(x + 1)$
    $= x + x'\,1$
    $= x + x'$
    $= 1$
  - $x\,0 = 0$ by duality
- Theorem 3: (x')' = x
  - Postulate 5 defines the complement of x, $x + x' = 1$ and $x\,x' = 0$
  - The complement of x' is x and is also $(x')'$
    (the complement is unique)

2.4 Basic Theorems and Properties

- Theorem 6
  - $x + xy = x\,1 + xy$
    $= x\,(1 + y)$
    $= x\,1$
    $= x$
  - $x\,(x + y) = x$ by duality
- By means of truth table

x	y	xy	$x + xy$
0	0	0	
0	1	0	
1	0	0	
1	1	1	

2.4 Basic Theorems and Properties

- DeMorgan's Theorems
  - $(x+y)' = x'\,y'$
  - $(x\,y)' = x' + y'$

x	y	$x+y$	$(x+y)'$	x'	y'	$x'y'$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

- The distributive laws …

2.4 Basic Theorems and Properties

- Operator Precedence
  - parentheses
  - NOT
  - AND
  - OR

- Examples
  - $x\,y' + z$
  - $(x\,y + z)'$

2.5 Boolean Functions

- A Boolean function
  - binary variables
  - binary operators OR and AND
  - unary operator NOT
  - Parentheses

- Examples
  - $F_1 = x + y'z$
  - $F_2 = x'\,y'\,z + x'\,y\,z + x\,y'$
  - $F_3 = x\,y' + x'\,z$

2.5 Boolean Functions

- $F_1 = x + y'z$
- $F_2 = x' y' z + x' y z + x y'$
- $F_3 = x y' + x' z$

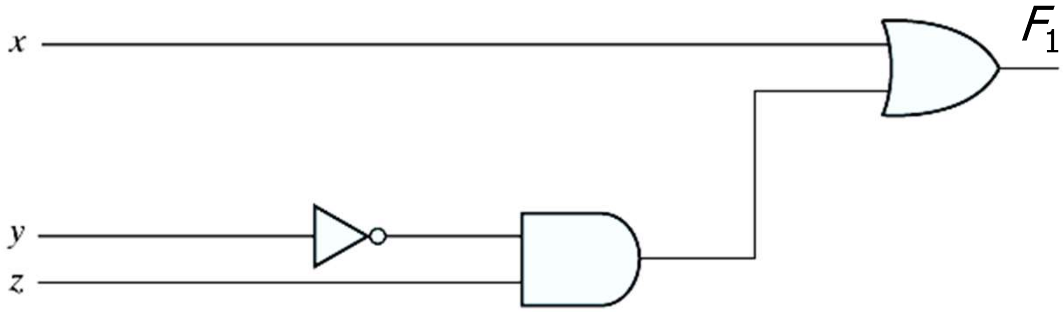
Table 2.2
Truth Tables for F_1 and F_2

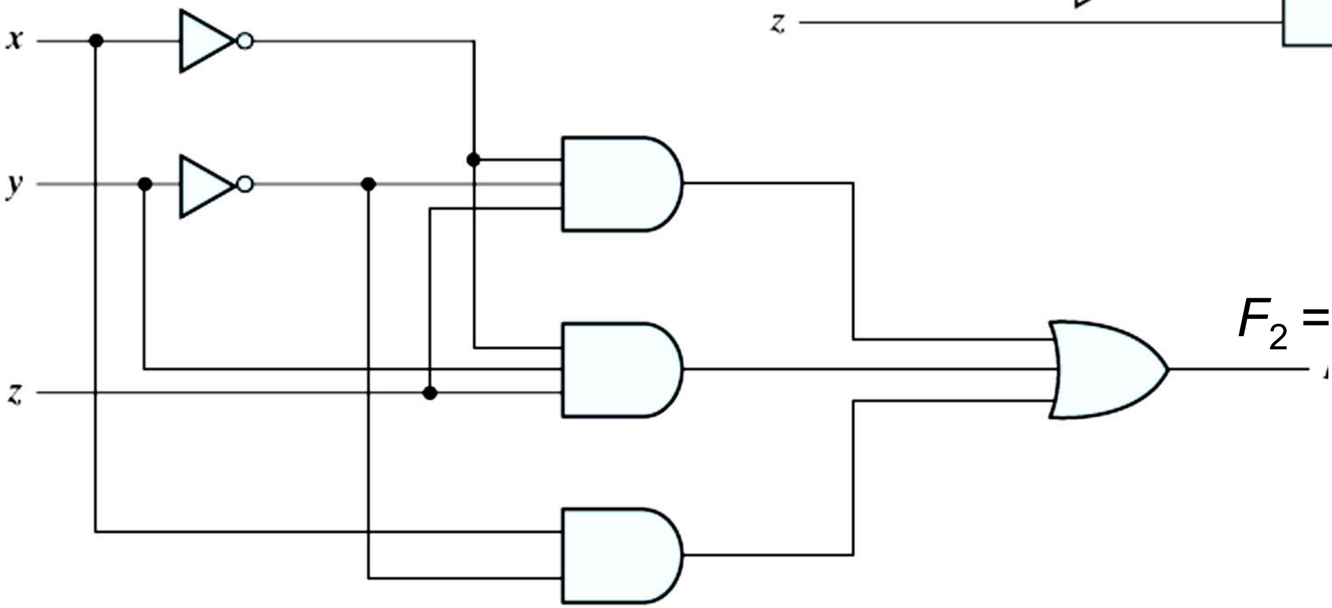
x	y	z	F_1	F_2
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	0
1	1	1	1	0

- Two Boolean expressions may specify the same function: $F_2 = F_3$

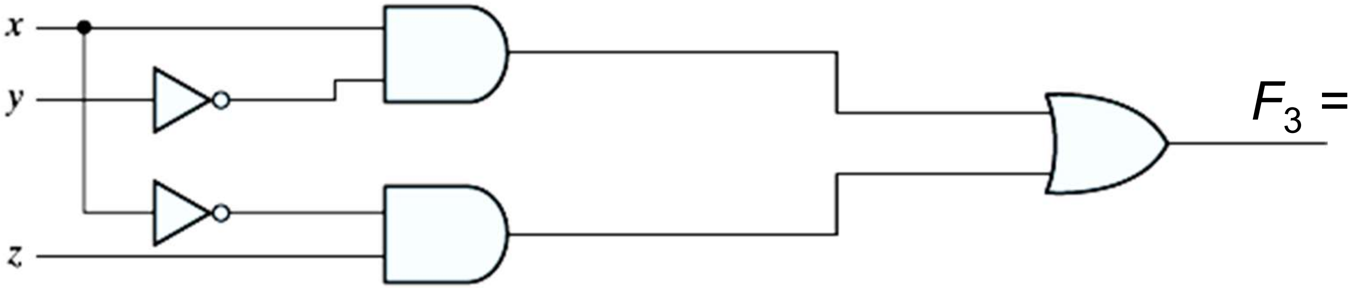
2.5 Boolean Functions

- Implementation with logic gates





$F_2 = x'\,y'\,z + x'\,y\,z + x\,y'$



$F_3 = x\,y' + x'\,z \leftarrow$ *Better!*

Algebraic Manipulation

- To minimize Boolean expressions
  - literal: a primed or unprimed variable (an input to a gate)
  - term: an implementation with a gate
  - The minimization of the number of literals and the number of terms → a circuit with less equipment
  - It is a hard problem (no specific rules to follow)

Algebraic Manipulation

- **Ex. 2.1**
  1. $x(x'+y) = xx' + xy = \ldots = xy$
  2. $x + x'y =$
  3. $(x+y)(x+y') =$
  4. $xy + x'z + yz = xy + x'z + yz(x+x')$
     $= xy + x'z + yzx + yzx'$
     $= xy(1+z) + x'z(1+y)$
     $= xy + x'z$
  5. $(x+y)(x'+z)(y+z) =$
  - Duality: 1 $\leftrightarrow$ 2, 4 $\leftrightarrow$ 5 (consensus theorem)

Complement of a Function: A or B

A. $F' \leftarrow$ an interchange of 0's for 1's and 1's for 0's in the value of F (Table)

B. $F' \leftarrow$ Applying DeMorgan's theorem to F

$(A+B)' = A'B'$, $(AB)' = A' + B'$ (Algebra)

- 3-variable DeMorgan's theorem
  - $(A+B+C)' = (A+X)'$ let $B+C = X$
    $= A'X'$ by DeMorgan's
    $= A'(B+C)'$
    $= A'(B'C')$ by DeMorgan's
    $= A'B'C'$ associative

Complement of a Function

- Generalizations of DeMorgan's theorem
  - $(A+B+C+ \ldots +F)' = A'B'C' \ldots F'$
  - $(ABC \ldots F)' = A'+B'+C'+ \ldots +F'$
- **Ex. 2.2** (use DeMorgan's theorem)
  - $(x'yz' + x'y'z)' = (x'yz')' (x'y'z)'$
    $= (x+y'+z) (x+y+z')$
  - $[x(y'z'+yz)]' = x' + (y'z'+yz)'$
    $= x' + (y'z')' (yz)'$
    $= x' + (y+z) (y'+z') = \ldots$
- **Ex. 2.3** (take the dual + complement each literal)
  - $x'yz' + x'y'z \rightarrow (x'+y+z') (x'+y'+z)$ (the dual)
    $\rightarrow (x+y'+z)(x+y+z')$
  - $[x(y'z'+yz)]' \rightarrow \ldots$

2.6 Canonical and Standard Forms

- Minterms and Maxterms
  - A minterm: an AND term consists of all literals (in their normal form or in their complement form)
  - For example, two binary variables x and y,
    - $xy, xy', x'y, x'y'$
  - It is also called a standard product
  - n variables can be combined to form 2^n minterms

  - A maxterm: an OR term
  - It is also call a standard sum
  - 2^n maxterms

2.6 Canonical and Standard Forms

- each maxterm is the complement of its corresponding minterm, and vice versa

Table 2.3
Minterms and Maxterms for Three Binary Variables

			Minterms		Maxterms	
x	**y**	**z**	**Term**	**Designation**	**Term**	**Designation**
0	0	0	$x'y'z'$	m_0	$x + y + z$	M_0
0	0	1	$x'y'z$	m_1	$x + y + z'$	M_1
0	1	0	$x'yz'$	m_2	$x + y' + z$	M_2
0	1	1	$x'yz$	m_3	$x + y' + z'$	M_3
1	0	0	$xy'z'$	m_4	$x' + y + z$	M_4
1	0	1	$xy'z$	m_5	$x' + y + z'$	M_5
1	1	0	xyz'	m_6	$x' + y' + z$	M_6
1	1	1	xyz	m_7	$x' + y' + z'$	M_7

2.6 Canonical and Standard Forms

- A Boolean function can be expressed by
  - a truth table or sum of minterms
- **Ex.**
  - $f_1 = x'y'z + xy'z' + xyz = m_1 + m_4 + m_7$
  - $f_2 = x'yz + xy'z + xyz' + xyz = m_3 + m_5 + m_6 + m_7$

Table 2.4
Functions of Three Variables

x	y	z	Function f_1	Function f_2
0	0	0	0	0
0	0	1	1	0
0	1	0	0	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

2.6 Canonical and Standard Forms

- The complement of a Boolean function
  - the minterms that produce a 0
  - $f_1' = m_0 + m_2 + m_3 + m_5 + m_6$
    $= x'y'z' + x'yz' + x'yz + xy'z + xyz'$
  - $f_1 = (f_1')'$
    $= (x+y+z)(x+y'+z)(x+y'+z')(x'+y+z')(x'+y'+z)$
    $= M_0 M_2 M_3 M_5 M_6$
  - Any Boolean function can be expressed as
    - a sum of minterms
    - a product of maxterms

    →canonical forms

2.6 Canonical and Standard Forms

- **Ex. 2.4** Sum of minterms
  - $F = A+B'C$
    $= A(B+B') + B'C$
    $= AB + AB' + B'C$
    $= AB(C+C') + AB'(C+C') + (A+A')B'C$
    $= ABC + ABC' + AB'C + AB'C' + A'B'C$
    $= m_1 + m_4 + m_5 + m_6 + m_7$
  - $F(A,B,C) = \Sigma(1, 4, 5, 6, 7)$
  - or, built the truth table first →

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

2.6 Canonical and Standard Forms

- **Ex. 2.5** Product of maxterms
  - $F = xy + x'z$
    $= (xy + x')(xy + z)$
    $= (x+x')(y+x')(x+z)(y+z)$
    $= (x'+y)(x+z)(y+z)$
  - $x'+y = x' + y + zz'$
    $= (x'+y+z)(x'+y+z')$
  - $F = (x+y+z)(x+y'+z)(x'+y+z)(x'+y+z')$
    $= M_0 M_2 M_4 M_5$
  - $F(x,y,z) = \Pi(0,2,4,5)$

Conversion between Canonical Forms

- Example

  $F(A,B,C) = \Sigma(1,4,5,6,7)$

  $F'(A,B,C) = \Sigma(0,2,3)$

  $F(A,B,C) = \Pi(0,2,3)$

  (By DeMorgan's theorem: $m_j' = M_j$)

- sum of minterms $\leftrightarrow$ product of maxterms
  - interchange the symbols Σ and Π and list those numbers missing from the original form
  - Σ of 1's
  - Π of 0's

Conversion between Canonical Forms

- Example
  - $F(x, y, z) = \Sigma(1, 3, 6, 7)$
  - $F(x, y, z) = \Pi\,(0, 2, 4, 5)$

Table 2.6
Truth Table for $F = xy + x'z$

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

- Consensus theorem: $xy + x'z + yz = xy + x'z$
  - redundant term → can eliminate race hazards

Standard Forms

- Canonical forms are seldom used
- Sum of products (SOP)

  $$F_1 = y' + zy + x'yz'$$

- Product of sums (POS)

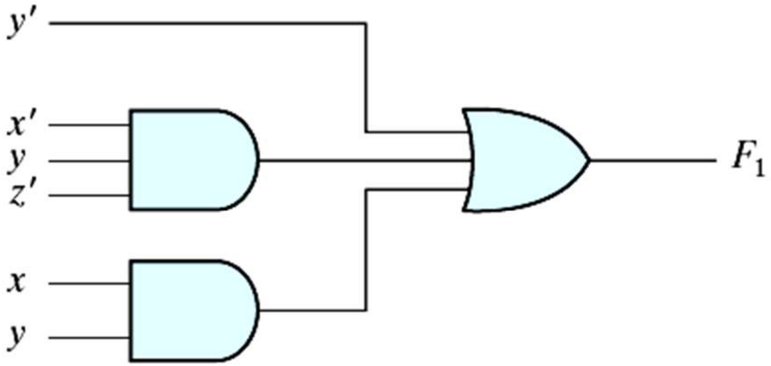
  $$F_2 = x(y'+z)(x'+y+z'+w)$$

- Nonstandard form → standard form

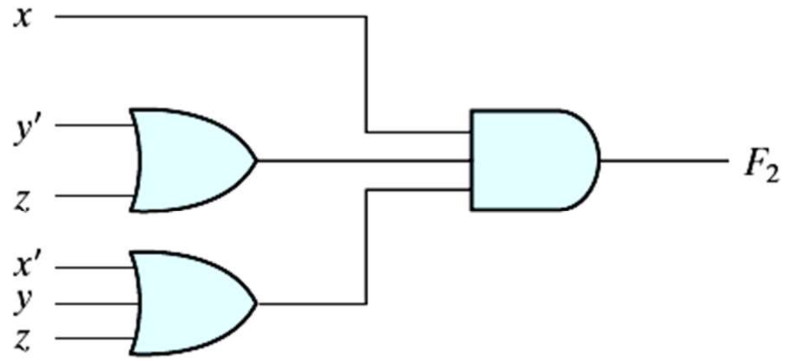
  $$F_3 = AB + C(D+E)$$
  $$= AB + CD + CE$$

Standard Forms

- Two-level implementation

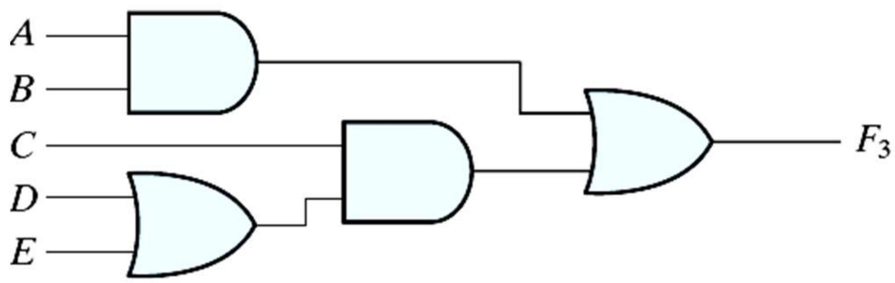


(a) Sum of Products

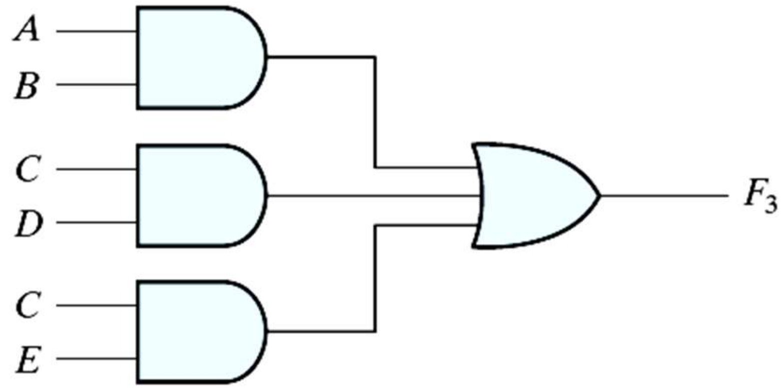


(b) Product of Sums

- Three- and two-level implementation



(a) $AB + C(D + E)$



(b) $AB + CD + CE$

2.7 Other Logic Operations

- 2^n rows in the truth table of n binary variables
- 2^{2^n} functions for n binary variables
- 16 functions of two binary variables

Table 2.7
Truth Tables for the 16 Functions of Two Binary Variables

x	y	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

- All the new symbols except for the exclusive-OR symbol are not in common use by digital designers

Table 2.8

Boolean Expressions for the 16 Functions of Two Variables

	Boolean Functions	Operator Symbol	Name	Comments
0000	$F_0 = 0$		Null	Binary constant 0
0001	$F_1 = xy$	$x \cdot y$	AND	x and y
0010	$F_2 = xy'$	x/y	Inhibition	x, but not y
0011	$F_3 = x$		Transfer	x
0100	$F_4 = x'y$	y/x	Inhibition	y, but not x
0101	$F_5 = y$		Transfer	y
0110	$F_6 = xy' + x'y$	$x \oplus y$	Exclusive-OR	x or y, but not both
0111	$F_7 = x + y$	$x + y$	OR	x or y
1000	$F_8 = (x + y)'$	$x \downarrow y$	NOR	Not-OR
1001	$F_9 = xy + x'y'$	$(x \oplus y)'$	Equivalence	x equals y
1010	$F_{10} = y'$	y'	Complement	Not y
1011	$F_{11} = x + y'$	$x \subset y$	Implication	If y, then x
1100	$F_{12} = x'$	x'	Complement	Not x
1101	$F_{13} = x' + y$	$x \supset y$	Implication	If x, then y
1110	$F_{14} = (xy)'$	$x \uparrow y$	NAND	Not-AND
1111	$F_{15} = 1$		Identity	Binary constant 1

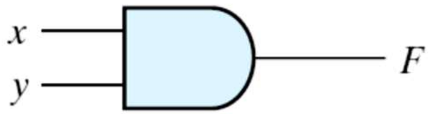
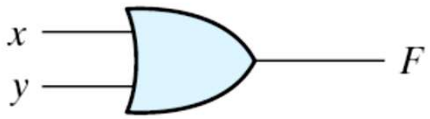
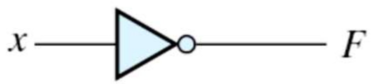
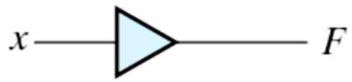
2.8 Digital Logic Gates

- Boolean expressions: AND, OR and NOT operations
- Constructing gates of other logic operations
  - the feasibility and economy
  - the possibility of extending gate's inputs
  - the basic properties of the binary operations
  - the ability of the gate to implement Boolean functions alone or …

2.8 Digital Logic Gates

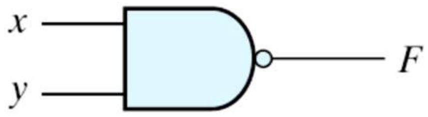
- Consider the 16 functions
  - two are equal to a constant
  - four are repeated
  - inhibition and implication are not commutative or associative
  - the other eight: complement, transfer, AND, OR, NAND, NOR, XOR, and equivalence are used as standard gates

Standard Gates

Name	Graphic symbol	Algebraic function	Truth table
AND		$F = xy$	x y | F 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR		$F = x + y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
Inverter		$F = x'$	x | F 0 | 1 1 | 0
Buffer		$F = x$	x | F 0 | 0 1 | 1

Standard Gates

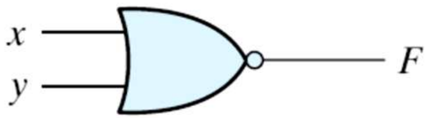
NAND



$F = (xy)'$

x	y	F
0	0	1
0	1	1
1	0	1
1	1	0

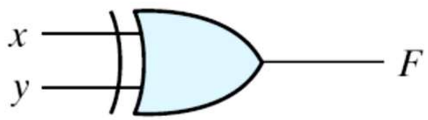
NOR



$F = (x + y)'$

x	y	F
0	0	1
0	1	0
1	0	0
1	1	0

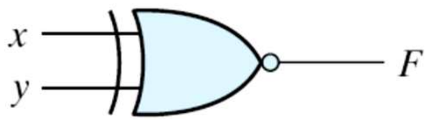
Exclusive-OR (XOR)



$F = xy' + x'y$
$= x \oplus y$

x	y	F
0	0	0
0	1	1
1	0	1
1	1	0

Exclusive-NOR or equivalence (XNOR)

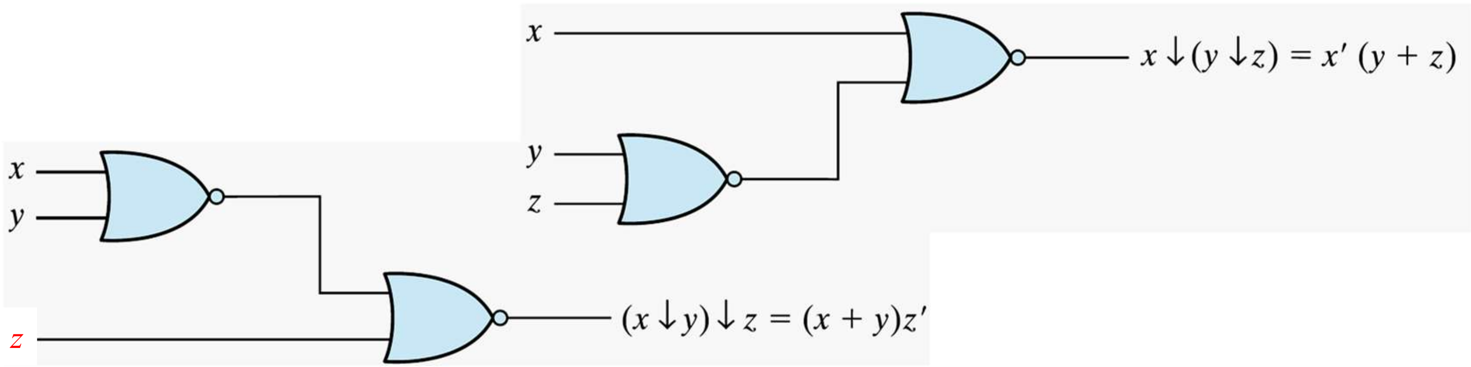


$F = xy + x'y'$
$= (x \oplus y)'$

x	y	F
0	0	1
0	1	0
1	0	0
1	1	1

Extension to Multiple Inputs

- AND and OR are commutative and associative
  - $(x+y)+z = x+(y+z) = x+y+z$
  - $(x\ y)z = x(y\ z) = x\ y\ z$
- NAND and NOR are commutative but not associative $\rightarrow$ they are not extendable



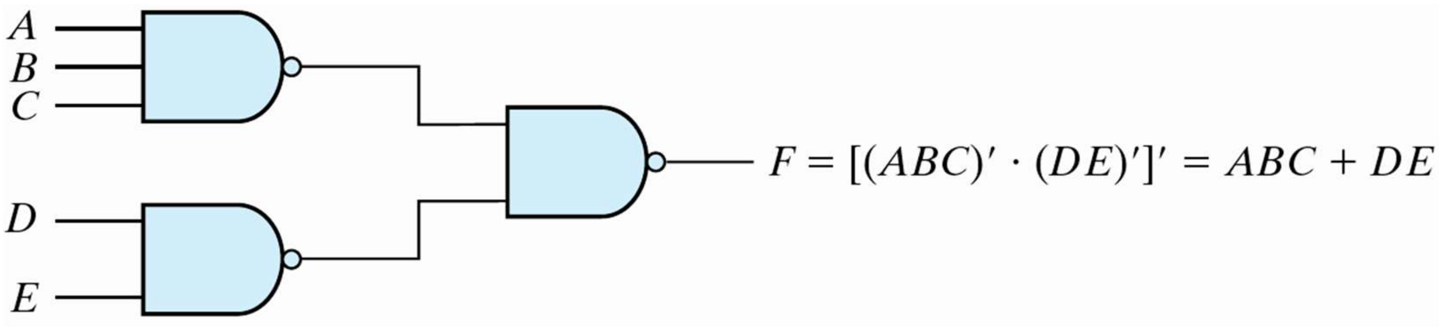
$$\boxed{(x \downarrow y) \downarrow z \neq x \downarrow (y \downarrow z)}$$

Extension to Multiple Inputs

- Multiple NOR ≜ a complemented OR gate
- Multiple NAND ≜ a complemented AND gate

(a) 3-input NOR gate (b) 3-input NAND gate

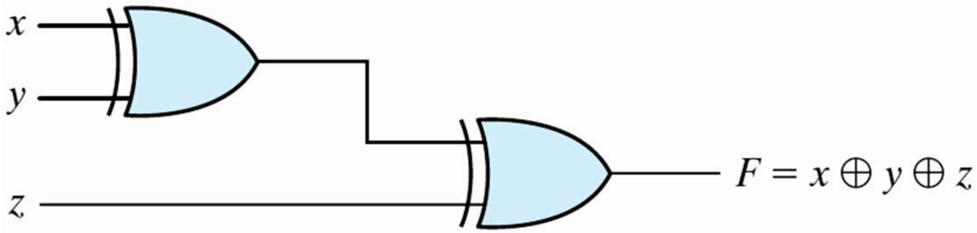
- The cascaded NAND operations = sum of products
- The cascaded NOR operations = product of sums



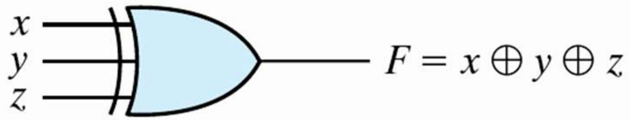
(c) Cascaded NAND gates

Extension to Multiple Inputs

- The XOR and XNOR gates are commutative and associative
  - Example: 3-input XOR gate



(a) Using 2-input gates



(b) 3-input gate

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

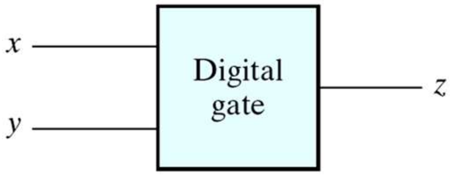
(c) Truth table

  - XOR is an odd function: it is equal to 1 if the inputs variables have an odd number of 1's

Positive and Negative Logic

x	y	z
L	L	L
L	H	L
H	L	L
H	H	H

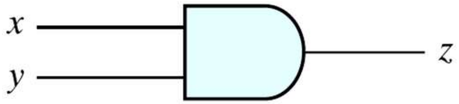
(a) Truth table with H and L



(b) Gate block diagram

x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

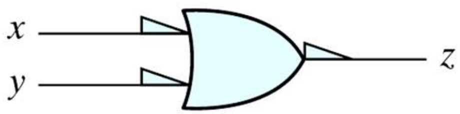
(c) Truth table for positive logic



(d) Positive logic AND gate

x	y	z
1	1	1
1	0	1
0	1	1
0	0	0

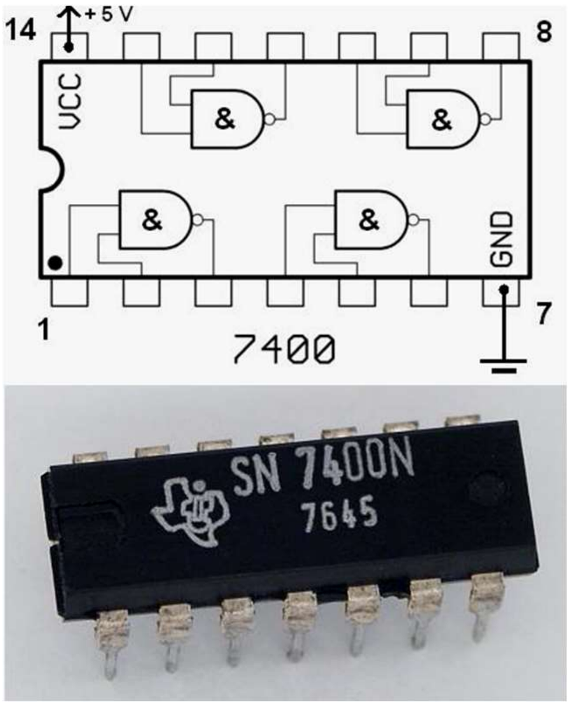
(e) Truth table for negative logic



(f) Negative logic OR gate

2.9 Integrated Circuits (IC, chips)

- Levels of integration
  - SSI: < 10 gates
  - MSI: 10 ~ 1000 gates (Chapter 4)
  - LSI: 1000 ~ 100k gates (Chapter 7)
  - VLSI: > 100k gates
    - small size (compact size)
    - low cost
    - low power consumption
    - high reliability
    - high speed



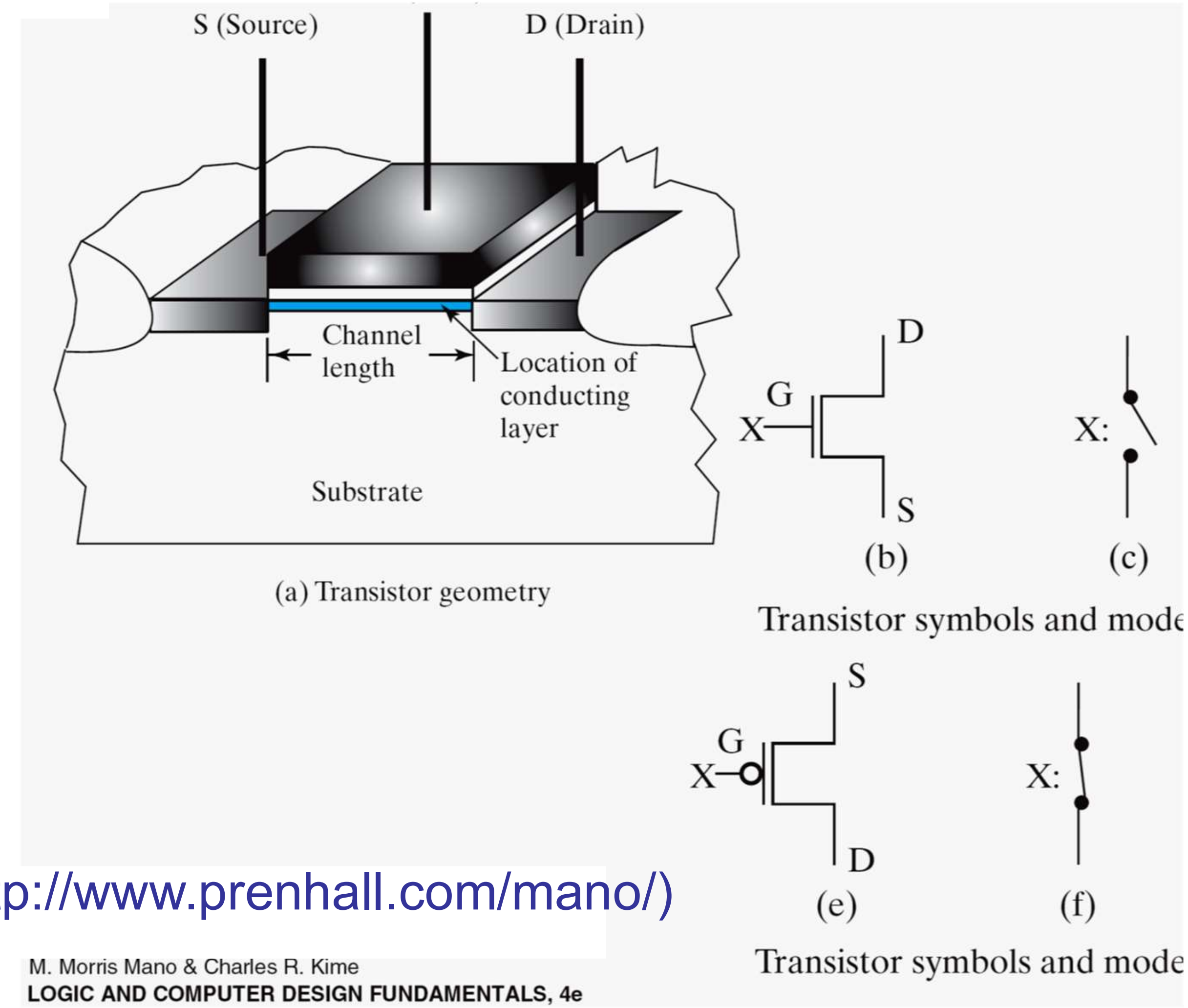
(from wikipedia)

2.9 Integrated Circuits

- Digital logic families: circuit technology
  - TTL: transistor-transistor logic
  - ECL: emitter-coupled logic (high speed, high power consumption)
  - MOS: metal-oxide semiconductor (NMOS, high density)
  - CMOS: complementary MOS (low power)
  - BiCMOS: high speed, high density

2.9 Integrated Circuits

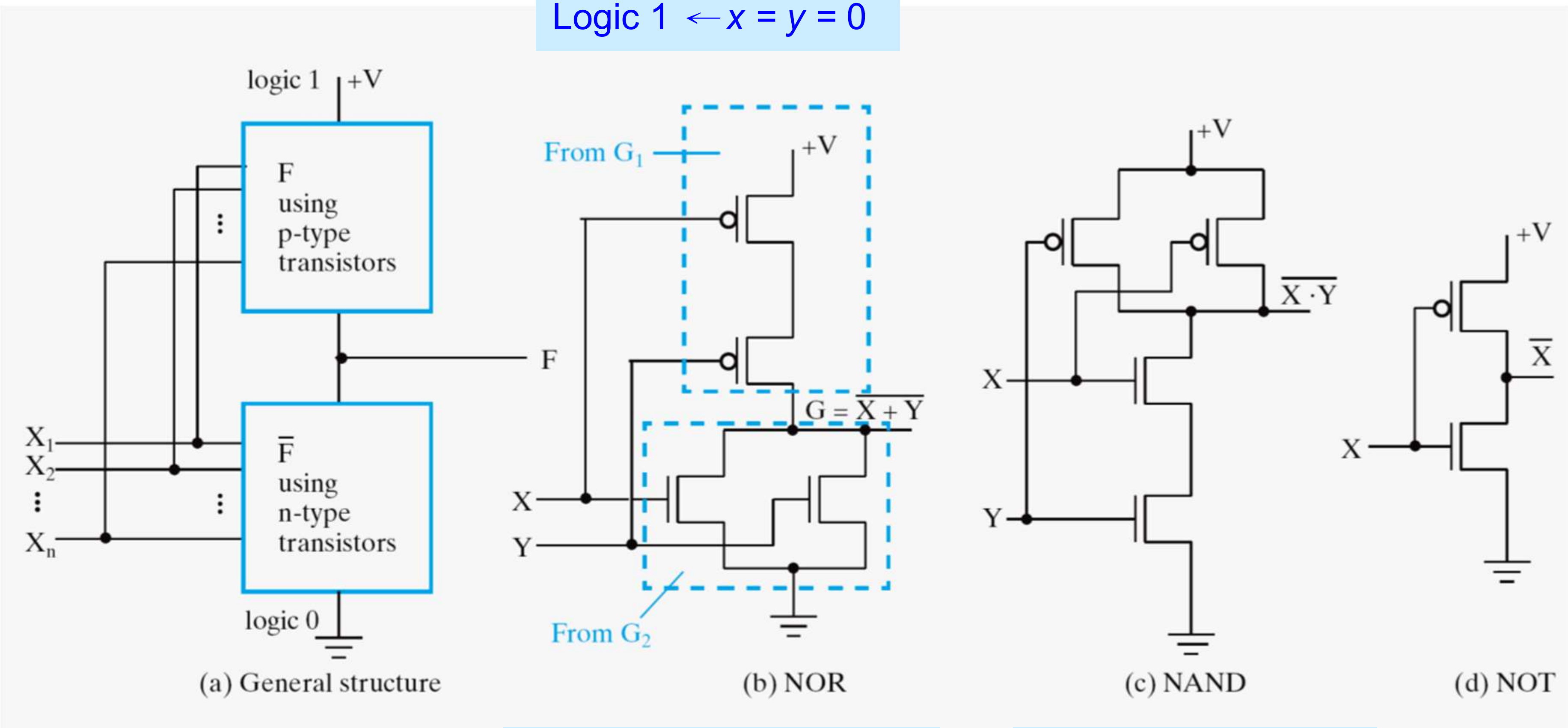
- CMOS circuit



(from: http://www.prenhall.com/mano/)

2.9 Integrated Circuits

Logic 1 $\leftarrow x = y = 0$

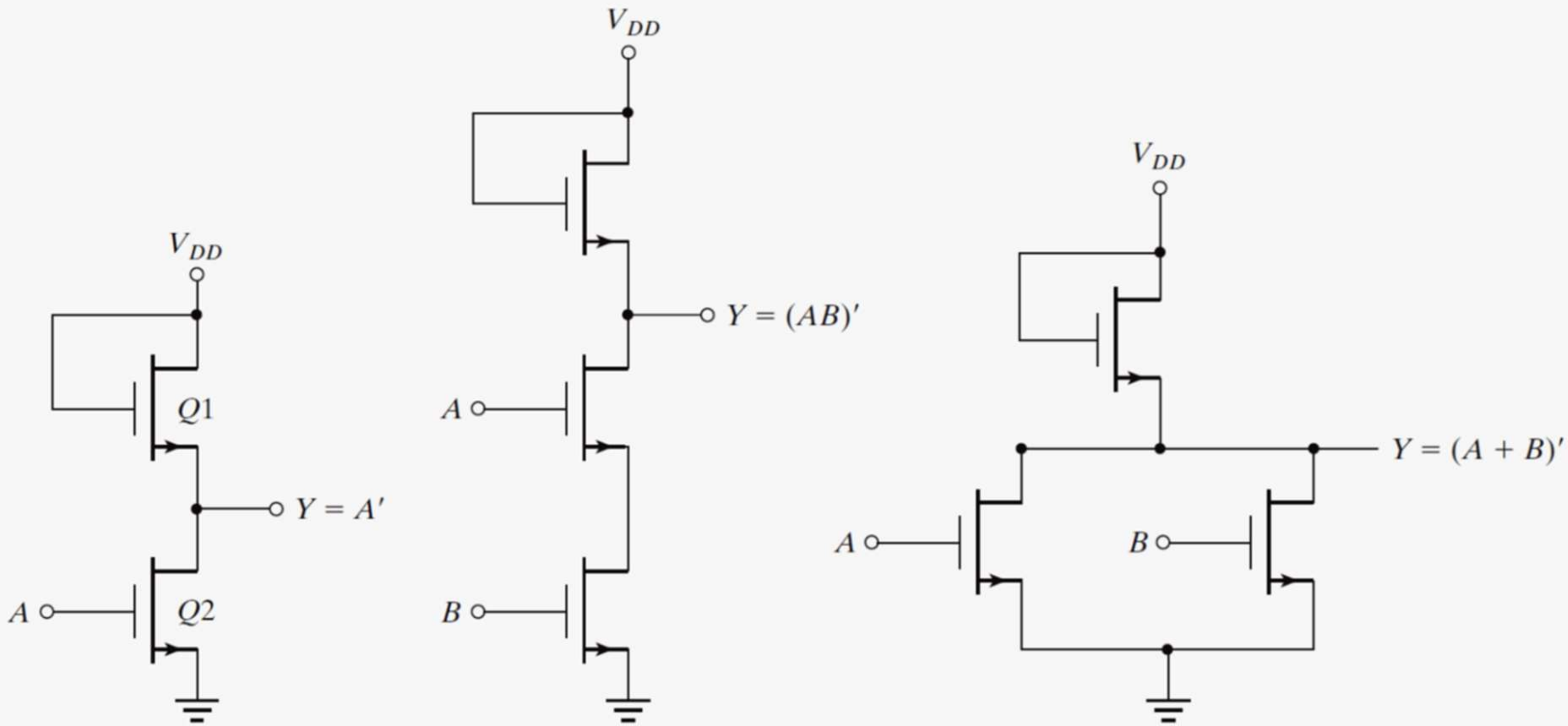


Logic 0 $\leftarrow x = 1$ or $y = 1$

(better than NOR)

2.9 Integrated Circuits

- NMOS circuit



(a) Inverter (b) NAND gate (c) NOR gate

2.9 Integrated Circuits

- The most important parameters
  - Fan-out: the number of standard loads that the output of a typical gate can drive
  - Fan-in: …
  - Power dissipation
  - Propagation delay: the average transition delay time for the signal to propagate from input to output
  - Noise margin: the maximum external noise voltage added to an input signal …

2.9 Integrated Circuits

- Computer-Aided Design (CAD) tools
  - Millions of transistors → software programs
  - Support computer-based representation
  - Automate the design process
- A typical design flow
  - Design entry
    - Schematic capture
    - HDL – Hardware Description Languages (Verilog, VHDL)
  - Simulation
  - Physical realization (ASIC, FPGA, PLD)

2.9 Integrated Circuits

- Ex. A digital version of oscilloscope
  - an analog one: