

Boolean Algebra and Logic Gates

Chapter 2

Basic Definitions

Boolean Algebra defined with a set of elements, a set of operators and a number of unproved axioms or postulates.

A set is a collection of objects having a common property

x, y	Elements
$x \in S$	x is an element of set S
$y \notin S$	y is not an element of set S
$a * b = c$	* Binary operator and result c of operation on a and b is an element of set S
$a, b, c \in S$	

Basic Definitions

1. Closure: A set S is closed with respect to a binary operator if, for every pair of elements of S , the binary operator specifies a rule for obtaining a unique element of S .
2. Associative law: $(x*y)*z=x*(y*z)$ for all $x, y, z \in S$.
3. Commutative law: $x*y=y*x$
4. Identity Element: $e*x=x*e=x$
5. Inverse: A set S having the identity element e with respect to binary operator $*$ is said to have an inverse whenever, for every $x \in S$, there exists an element y such that $x*y=e$
6. Distributive law: If $*$ and $.$ are binary operators on S , $*$ is said to be distributive over $.$ whenever $x*(y.z)=(x*y).(x*z)$

Basic Definitions

Some examples. With the set of integers

$$I = \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$$

$$x + 0 = 0 + x = x$$

Some examples. With the set of natural numbers

$$N = \{1, 2, 3, \dots\}$$

There is no inverse since 0 does not belong to the set of Natural numbers.

With the set of integers

$$e = 0$$

$$a + (-a) = 0$$

There is 0 and therefore there is inverse for addition.

Basic Definitions

- A field is an example of an algebraic structure.
- A field is a set of elements, together with two binary operators, each having properties 1-5 and both combining to give property 6.

Basic Definitions

- Is the set of natural numbers a field? Why?
- Is the set of integers a field? Why?
- Is the set of real numbers a field? Why?

Basic Definitions

- On the real numbers, let $+$ define addition and \cdot define multiplication
- What is the additive identity?
- The additive inverse defines subtraction
- What is the multiplicative identity?
- What is the multiplicative inverse of a ?
- The only distributive law is:

$$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

Axiomatic Definition of Boolean Algebra

Boolean algebra is an algebraic structure defined by a set of elements, B , together with binary operators $+$ and \cdot .

- The structure is closed with respect to $+$
 - The structure is closed with respect to \cdot .
- The element 0 is an identity element with respect to $+$
 - The element 1 is an identity element with respect to \cdot .
- The structure is commutative with respect to $+$
 - The structure is commutative with respect to \cdot .
- The operator \cdot is distributive over $+$
 - The operator $+$ is distributive over \cdot .
- For every element $x \in B$, there exists an element $x' \in B$ called the complement of x such that (a) $x+x'=1$ and (b) $x \cdot x'=0$
- There exist at least two elements $x, y \in B$ such that $x \neq y$

Axiomatic Definition of Boolean Algebra

- Compare Boolean Algebra with arithmetic and ordinary algebra
- What about associative law (not included in Huntington postulates)?
- What about distributed law?
- What about additive or multiplicative inverses in Boolean Algebra?

Two-valued Boolean Algebra

Defined on a set of two elements $B = \{0,1\}$

x	y	x.y	x	y	x+y
0	0	0	0	0	0
0	1	0	0	1	1
1	0	0	1	0	1
1	1	1	1	1	1

x	x'
0	1
1	0

Two-valued Boolean Algebra

Let's check if the Huntington postulates are satisfied by B together with the two operators $+$ and \cdot .

1. Closure?
2. Identity elements for the two operators $+$ and \cdot ?
3. Commutative laws?
4. Distributed laws? Do the truth tables
5. Complements for 0 and 1?
6. Postulate 6?

Two-valued Boolean Algebra

<u>x</u>	<u>y</u>	<u>z</u>	<u>y+z</u>	<u>x.(y+z)</u>	<u>x.y</u>	<u>x.z</u>	<u>(x.y)+(x.z)</u>
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

Basic Theorems and Property of Boolean Algebra

- Duality:
Exchange parts (a) and (b) of Boolean Algebra (operators and identity element) and postulates remain valid
- In two valued Boolean Algebra identity elements of set B are the same: 1 and 0; interchange 1's by 0's and 0's by 1's and interchange OR and AND operators.

Basic Theorems

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) $x(x + y) = x$

Theorem 1(a)

$$x + x = x$$

$$x + x = (x + x) \cdot 1$$

but

$$1 = x + x'$$

thus

$$x + x = (x + x)(x + x')$$

Distributing + over . gives in general

$$x + (xy) = x + xy = (x + x)(x + y)$$

Replace x by x'

$$x + x = (x + x)(x + x') = x + xx'$$

$$= x + 0$$

$$= x$$

Theorem 1(b)

$$x \cdot x = x$$

$$\begin{aligned}x \cdot x &= xx + 0 \\ &= xx + xx' \\ &= x(x + x') \\ &= x \cdot 1 \\ &= x\end{aligned}$$

It can be proved by duality of Theorem 1(a)

Theorem 2(a)

$$x + 1 = 1$$

$$\begin{aligned}x + 1 &= 1 \cdot (x + 1) \\ &= (x + x')(x + 1) \\ &= x + x' \cdot 1 \\ &= x + x' \\ &= 1\end{aligned}$$

Theorem 2(b)

$$x \cdot 0 = 0$$

By duality of Theorem 2(a)

Theorem 3

$$(x')' = x$$

$$x + x' = 1 \quad \text{and} \quad x \cdot x' = 0$$

Both equations define the complement

The complement of x' is x and is also $(x')'$

Since the complement is unique $(x')' = x$

Theorem 6(a) Absorption

$$x + xy = x$$

$$x + xy = x \cdot 1 + xy$$

$$= x(1 + y)$$

$$= x(y + 1)$$

$$= x \cdot 1$$

$$= x$$

Theorem 6(b) Absorption

$$x(x + y) = x$$

By duality of Theorem 6(a)

Theorem 6(a) Absortion

Proof by truth table

x	y
0	0
0	1
1	0
1	1

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

DeMorgan's Theorem

Proof by truth table

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

Operator precedence

1. Parenthesis
2. NOT
3. AND
4. OR

Boolean Functions

A Boolean functions described by an algebraic expression consists of binary variables, the binary constants 0 and 1, and the logic operation symbols

$$F_1 = x + y'z$$

$$F_2 = x'y'z + x'yz + xy'$$

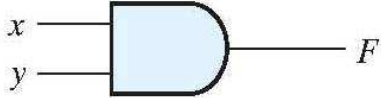
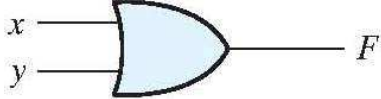
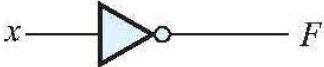
Do the truth tables for these functions

Boolean functions

Table 2.2
Truth Tables for F_1 and F_2

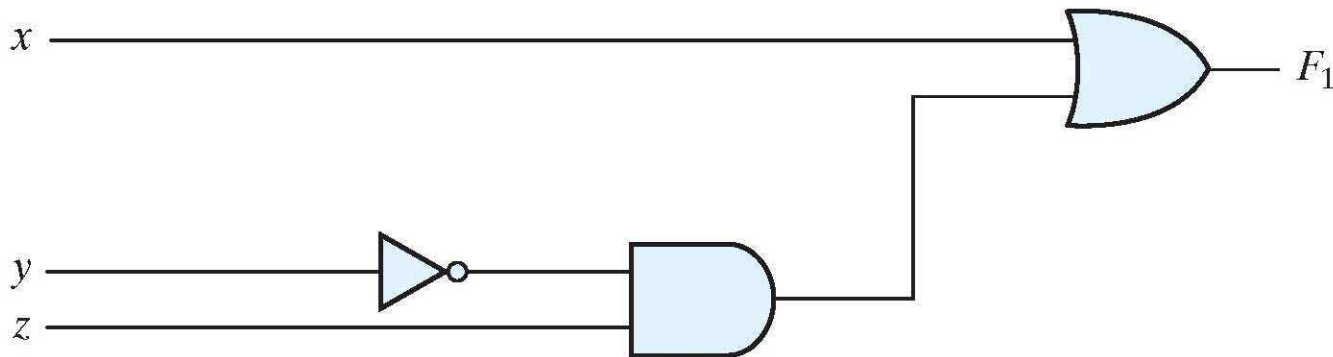
x	y	z	F_1	F_2
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	0
1	1	1	1	0

Boolean Functions

Name	Graphic symbol	Algebraic function	Truth table															
AND		$F = xy$	<table border="1"> <thead> <tr> <th>x</th> <th>y</th> <th>F</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	x	y	F	0	0	0	0	1	0	1	0	0	1	1	1
x	y	F																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR		$F = x + y$	<table border="1"> <thead> <tr> <th>x</th> <th>y</th> <th>F</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	x	y	F	0	0	0	0	1	1	1	0	1	1	1	1
x	y	F																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
Inverter		$F = x'$	<table border="1"> <thead> <tr> <th>x</th> <th>F</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	x	F	0	1	1	0									
x	F																	
0	1																	
1	0																	

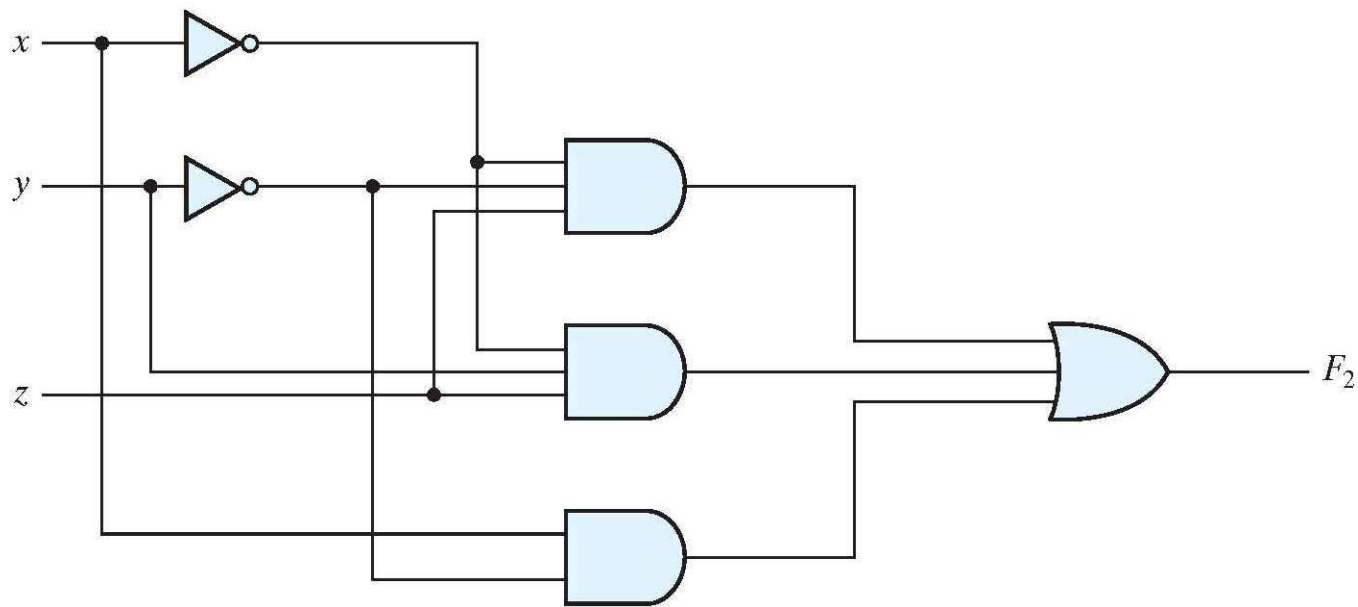
Boolean functions

Transform the algebraic equation of F_1 to a circuit diagram using logic gates



Boolean functions

Transform the algebraic equation of F_2 to a circuit diagram using logic gates



(a) $F_2 = x'y'z + x'yz + xy'$

Boolean functions

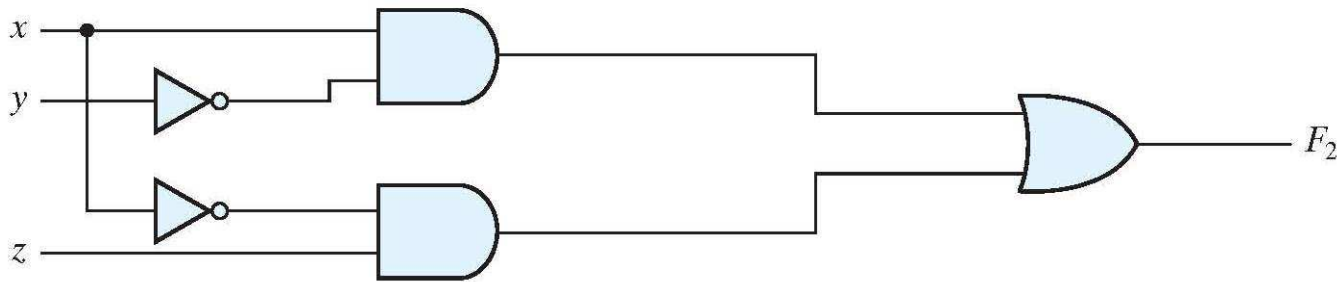
Algebraic manipulation can yield smaller (simpler) expressions, and a smaller implementation (lower cost)

$$F_2 = x'y'z + x'yz + xy'$$

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

$$F_2 = x'z + xy'$$

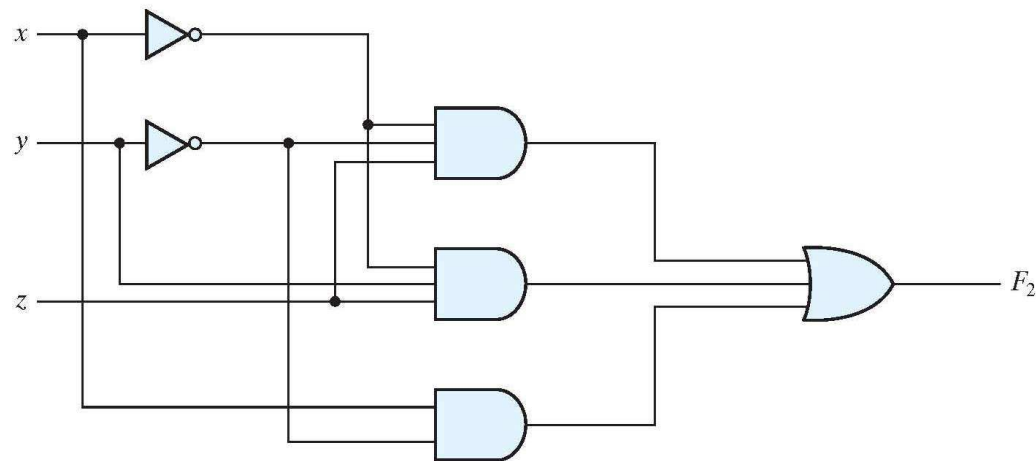
Boolean functions



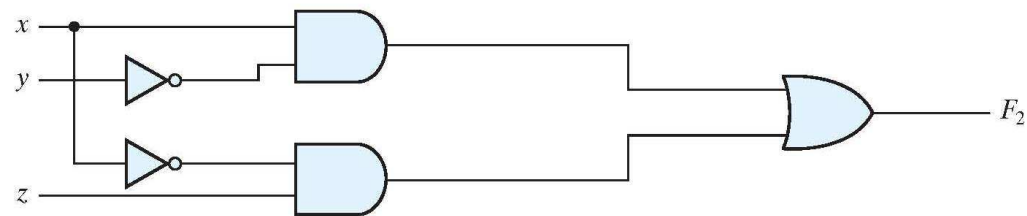
(b) $F_2 = xy' + x'z$

Boolean functions

Compare the two implementations



(a) $F_2 = x'y'z + x'yz + xy'$



(b) $F_2 = xy' + x'z$

Algebraic manipulation

Example 1 $x(x' + y) = xx' + xy = 0 + xy = xy$

Example 2 $x + x'y = (x + x')(x + y) = 1(x + y)$
 $= x + y$

Example 3 $(x + y)(x + y') = x + xy + xy' + yy'$
 $= x(1 + y + y') + 0 = x$

Algebraic manipulation

Example 4

$$\begin{aligned}xy + x'z + yz &= xy + x'z + yz(x + x') \\ &= xy + x'z + xyz + x'yz \\ &= xy(1 + z) + x'z(1 + y) \\ &= xy + x'z\end{aligned}$$

Example 5

$$\begin{aligned}(x + y)(x' + z)(y + z) \\ = (x + y)(x' + z)\end{aligned}$$

By duality of Example 4

Complement of a function

Generalization of DeMorgan's theorems

$$(A + B + C + \dots + F)' = A' B' C' \dots F'$$

$$(ABC \dots F)' = A' + B' + C' + \dots F'$$

Complement of a function

Find the complement of the following functions

$$\begin{aligned}F_1' &= (x'yz' + x'y'z)'
&= (x'yz')'(x'y'z)'
&= (x + y' + z)(x + y + z')$$

$$\begin{aligned}F_2' &= [x(y'z' + yz)]'
&= x' + (y'z' + yz)'
&= x' + (y'z')'(yz)'
&= x' + (y + z)(y' + z')
&= x' + yz' + yz'$$

Complement of a function

Find the complement of the following functions by taking their duals and complementing each literal

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

The dual $(x' + y + z')(x' + y' + z)$

Complement each literal $(x + y' + z)(x + y + z') = F_1'$

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

The dual $x + (y' + z')(y + z)$

Complement each literal $x' + (y + z)(y' + z') = F_2'$

Canonical and Standard Form

Table 2.3
Minterms and Maxterms for Three Binary Variables

x	y	z	Minterms		Maxterms	
			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

Minterms and maxterms

- Minterm or standard product: product of variables or their complement (all) in Boolean function.
- If n variables 2^n possible combinations.
- Obtained from truth table of function

Minterms and maxterms

Example of two Boolean functions on truth table

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

Minterms

From truth table

$$\begin{aligned}f_1 &= x'y'z + xy'z' + xyz \\ &= m_1 + m_4 + m_7\end{aligned}$$

$$\begin{aligned}f_2 &= x'yz + xy'z + xyz' + xyz \\ &= m_3 + m_5 + m_6 + m_7\end{aligned}$$

Maxterms

Complementing function and applying DeMorgan's Theorem

$$f_1' = x'y'z' + x'yz' + x'yz + xy'z + xyz'$$

Complement applying DeMorgan's

$$\begin{aligned} f_1 &= (x + y + z)(x + y' + z)(x + y' + z' + z')(x' + y + z')(x' + y' + z) \\ &= M_0 M_2 M_3 M_5 M_6 \end{aligned}$$

Maxterms

Complementing function and applying DeMorgan's Theorem

$$f_2' = x'y'z' + x'y'z + x'yz' + xy'z'$$

Complement applying DeMorgan's

$$\begin{aligned} f_2 &= (x + y + z)(x + y + z')(x + y' \\ &\quad + z)(x' + y + z) \\ &= M_0 M_1 M_2 M_4 \end{aligned}$$

Example 2.4

Table 2.5
Truth Table for $F = A + B'C$

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

Figure Number: Table02_05
Mano/Ciotti
Digital Design, 4e

$$F = A + B'C$$

$$= m_1 + m_4 + m_5 + m_6 + m_7$$

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

Conversion between canonical forms

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

Complement function

$$\begin{aligned} F'(A, B, C) &= \sum (0,2,3) \\ &= m_0 + m_2 + m_3 \end{aligned}$$

Complement using DeMorgan's Theorem

$$\begin{aligned} F &= (m_0 + m_2 + m_3)' \\ &= m'_0 m'_2 m'_3 \\ &= M_0 M_2 M_3 \\ &= \prod (0,2,3) \end{aligned}$$

Conversion between canonical forms

1. Interchange the symbols Σ and Π
2. List the numbers missing from the original form (remember there are 2^n minterms of maxterms , where n is the number of binary variables in the function)

Conversion between canonical forms

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

$$F(x, y, z) = xy + x'z$$

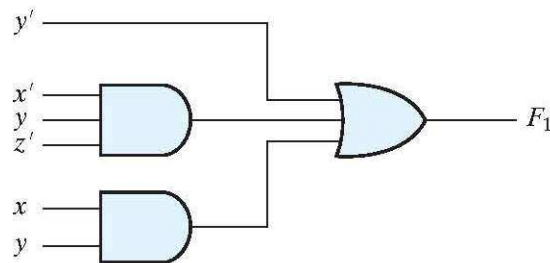
$$F(x, y, z) = \sum (1, 3, 6, 7)$$

$$F(x, y, z) = \prod (0, 2, 4, 5)$$

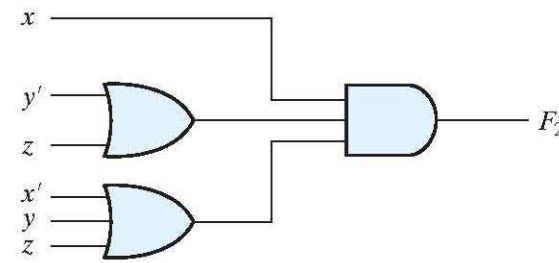
Standard Forms

- Sum of products (not canonical necessarily)
- Product of sums (not canonical necessarily)

Standard forms



(a) Sum of Products



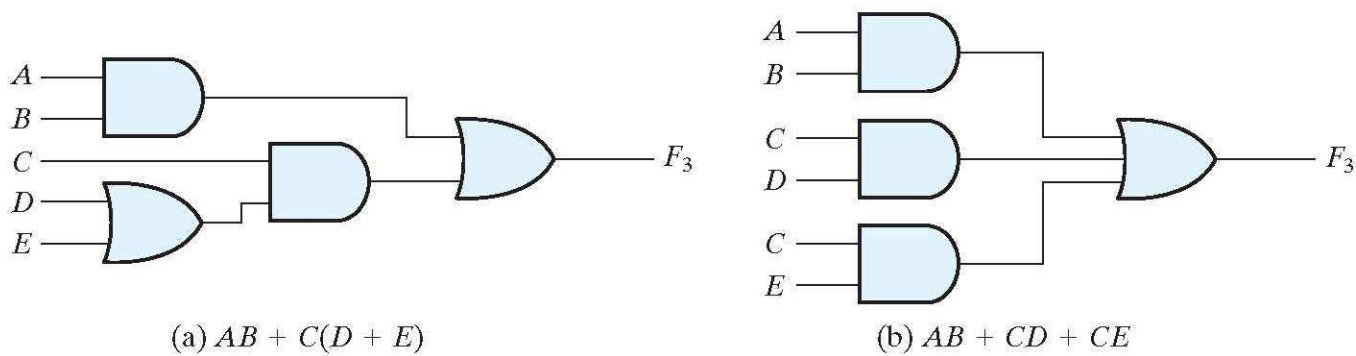
(b) Product of Sums

Obtain from diagram the standard forms

$$F_1 = y' + xy + x'yz' \quad F_2 = x(y' + z)(x' + y + z')$$

Standard forms

Three- and two-level implementations



Use distributive law to remove parenthesis

Other logic operations

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



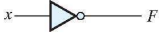

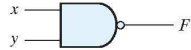


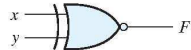
x	y	F₀	F₁	F₂	F₃	F₄	F₅	F₆	F₇	F₈	F₉	F₁₀	F₁₁	F₁₂	F₁₃	F₁₄	F₁₅
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Other Logic Operations

Table 2.8
Boolean Expressions for the 16 Functions of Two Variables

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

Digital Logic Gates

Name	Graphic symbol	Algebraic function	Truth table															
AND		$F = xy$	<table border="1"> <thead> <tr> <th>x</th> <th>y</th> <th>F</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	x	y	F	0	0	0	0	1	0	1	0	0	1	1	1
x	y	F																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR		$F = x + y$	<table border="1"> <thead> <tr> <th>x</th> <th>y</th> <th>F</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	x	y	F	0	0	0	0	1	1	1	0	1	1	1	1
x	y	F																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
Inverter		$F = x'$	<table border="1"> <thead> <tr> <th>x</th> <th>F</th> </tr> </thead> <tbody> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </tbody> </table>	x	F	0	1	1	0									
x	F																	
0	1																	
1	0																	
Buffer		$F = x$	<table border="1"> <thead> <tr> <th>x</th> <th>F</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td></tr> </tbody> </table>	x	F	0	0	1	1									
x	F																	
0	0																	
1	1																	
NAND		$F = (xy)'$	<table border="1"> <thead> <tr> <th>x</th> <th>y</th> <th>F</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	x	y	F	0	0	1	0	1	1	1	0	1	1	1	0
x	y	F																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR		$F = (x + y)'$	<table border="1"> <thead> <tr> <th>x</th> <th>y</th> <th>F</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	x	y	F	0	0	1	0	1	0	1	0	0	1	1	0
x	y	F																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
Exclusive-OR (XOR)		$F = xy' + x'y$ $= x \oplus y$	<table border="1"> <thead> <tr> <th>x</th> <th>y</th> <th>F</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	x	y	F	0	0	0	0	1	1	1	0	1	1	1	0
x	y	F																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
Exclusive-NOR or equivalence		$F = xy + x'y'$ $= (x \oplus y)'$	<table border="1"> <thead> <tr> <th>x</th> <th>y</th> <th>F</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	x	y	F	0	0	1	0	1	0	1	0	0	1	1	1
x	y	F																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

Extension to multiple inputs

- NOT: only one input
- OR: commutative and associative, thus no problem
- AND: commutative and associative, thus no problem

- NAND: commutative but not associative!
- NOR: commutative but not associative!

Extension to multiple inputs

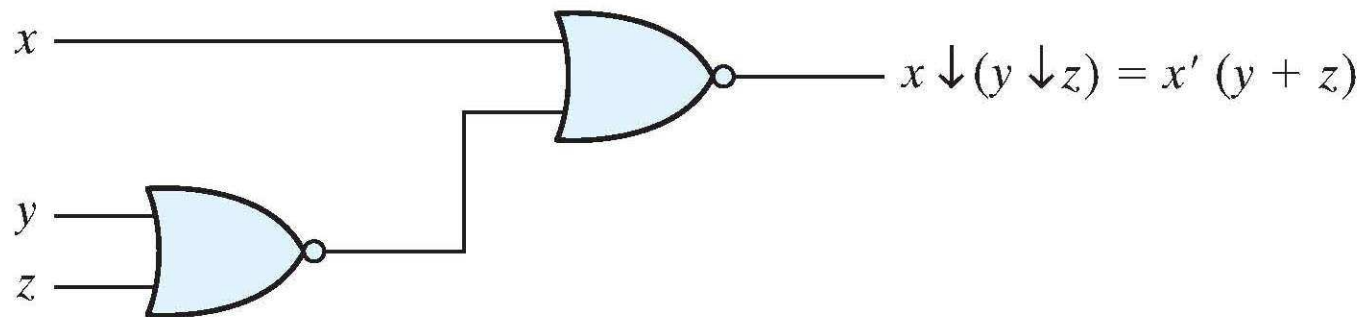
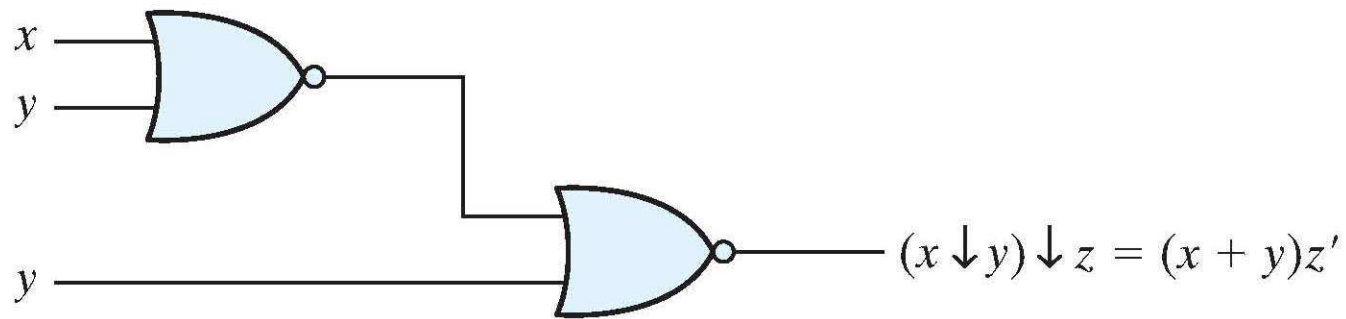
NOR gate

$$x \downarrow y \downarrow z = (x + y + z)'$$

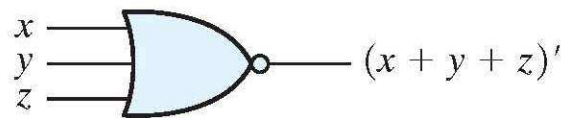
NAND gate

$$x \uparrow y \uparrow z = (xyz)'$$

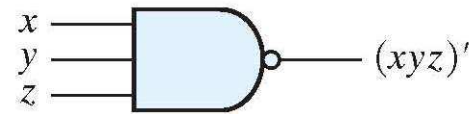
Extension to multiple inputs



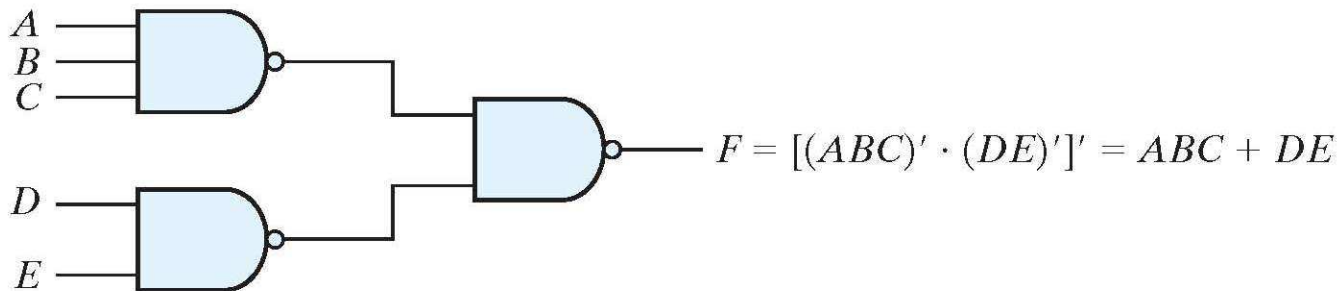
Extension to multiple inputs



(a) 3-input NOR gate

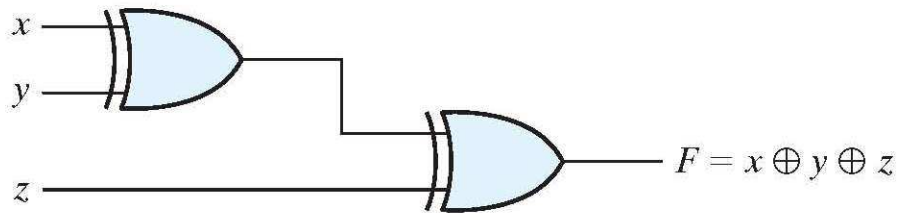


(b) 3-input NAND gate

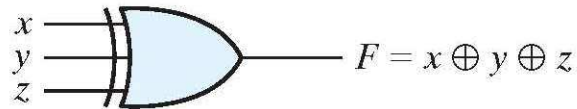


(c) Cascaded NAND gates

Extension to multiple inputs



(a) Using 2-input gates

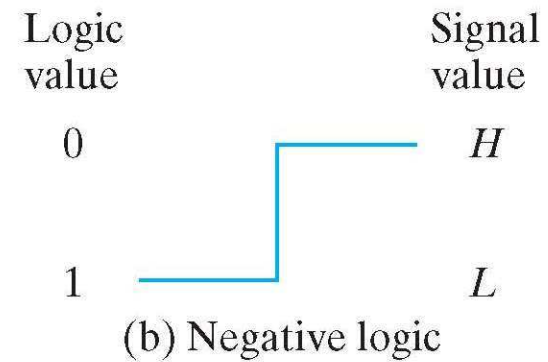
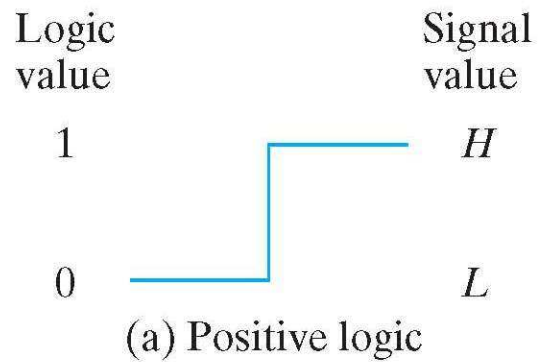


(b) 3-input gate

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

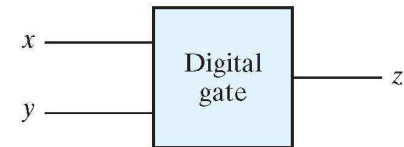
Positive and Negative Logic



Positive and Negative Logic

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

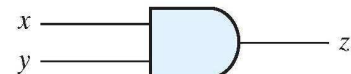
(a) Truth table with H and L



(b) Gate block diagram

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

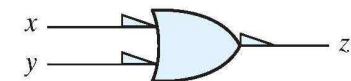
(c) Truth table for positive logic



(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

Integrated Circuits

- Small-Scale Integration (SSI): fewer than 10 gates per chip
- Medium-Scale Integration (MSI): 10 to 1000 gates per chip
- Large-Scale Integration (LSI): Several thousands of gates per chip
- Very Large-Scale Integrations (VLSI): hundreds of thousands of gates per chip

Digital logic families

- TTL: Transistor-Transistor Logic
- ECL: Emitter-Coupled Logic
- MOS: Metal-Oxide Semiconductor
- CMOS: Complementary Metal-Oxide Semiconductor

Computer-Aided Design

- Hardware Description Language (HDL)
 - Tools to simulate, verify functionality, and automatically and optimally synthesize the logic described in HDL
- IEEE approved HDL standards:
 - Verilog HDL
 - VHDL

Homework Assignment

- 2.1
- 2.2
- 2.5
- 2.9
- 2.10
- 2.12
- 2.14
- 2.18
- 2.20
- 2.28

Homework Assignment (Bonus)

- Obtain the datasheet for the CD4001 CMOS Quad two-input NOR Gates and explain in your own words the data in the Static Electrical Characteristics, Recommended Operating Conditions and the Dynamic Electrical Characteristics
- You can work in groups and turn your bonus assignment in one per group, but to obtain the bonus (5 points for your first test) you must answer one question about the datasheet in your test.