Digital Design and Computer Organization

Module1 : Introduction to Digital Design

Topics

- Binary Logic,
- Basic Theorems And Properties Of Boolean Algebra,
- Boolean Functions,
- Digital Logic Gates, Introduction,
- The Map Method,
- Four-Variable Map,
- Don’t-Care Conditions,
- NAND and NOR Implementation,
- Other Hardware Description Language – Verilog Model of a simple circuit.

Text book 1: 1.9,2.4, 2.5, 2.8, 3.1, 3.2, 3.3, 3.5, 3.6, 3.9

Binary Logic

Definition of Binary Logic

- Binary logic consists of binary variables and a set of logical operations.
- The variables are designated by letters of the alphabet, such as A, B, C, x, y, z, etc., with each variable having two and only two distinct possible values: 1 and 0.
- There are three basic logical operations: AND, OR, and NOT. Each operation produces a binary result, denoted by z.

AND Operation

- This operation is represented by a dot or by the absence of an operator.
- For example, $x \cdot y = z$ or $xy = z$ is read “x AND y is equal to z.”
- The logical operation AND is interpreted to mean that $z=1$ if and only if $x=1$ and $y=1$; otherwise $z=0$. (Remember that x, y, and z are binary variables and can be equal either to 1 or 0, and nothing else.)
- The result of the operation $x \cdot y$ is z.

Binary Logic

OR Operation

- This operation is represented by a plus sign.
- For example, x+y=z is read “x OR y is equal to z,” meaning that z=1 if x=1 or if y=1 or if both x=1 and y=1. If both x=0 and y=0, then z=0.

NOT Operation

- This operation is represented by a prime (sometimes by an overbar).
- For example, x′=z (or $\bar{x}$=z) is read “not x is equal to z,” meaning that z is what x is not.
- In other words, if x=1, then z=0, but if x=0, then z=1.
- The NOT operation is also referred to as the complement operation, since it changes a 1 to 0 and a 0 to 1, that is, the result of complementing 1 is 0, and vice versa.

Binary Logic

Truth Table

Definitions of logical operations may be listed in a compact form called truth tables.

A truth table is a table of all possible combinations of the variables, showing the relation between the values that the variables may take and the result of the operation.

The truth tables for the operations AND and OR with variables x and y are obtained by listing all possible values that the variables may have when combined in pairs.

For each combination, the result of the operation is then listed in a separate row. The truth tables for AND, OR, and NOT are given in below Table. These tables clearly demonstrate the definition of the operations.

AND

X	Y	Z= X.Y
0	0	0
0	1	0
1	0	0
1	1	1

OR

X	Y	Z= X+Y
0	0	0
0	1	1
1	0	1
1	1	1

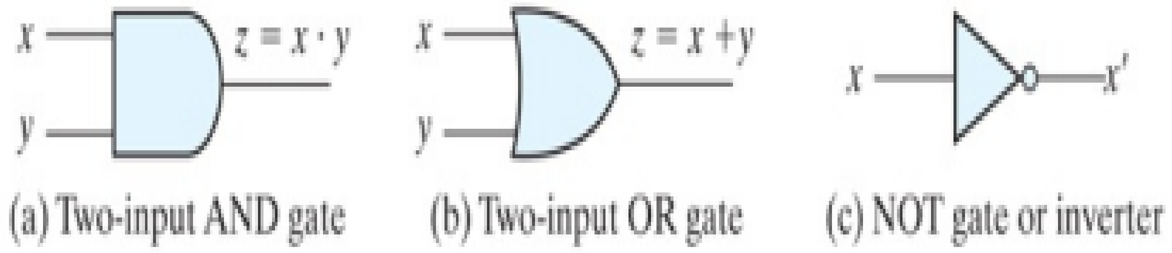
NOT

X	Z=x`
0	1
1	0

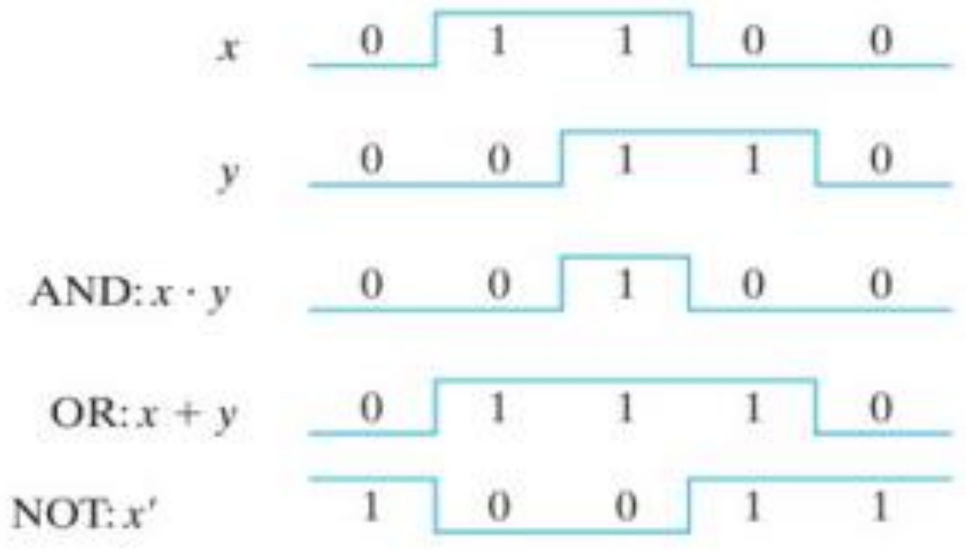
Binary Logic

Logic Gates

Logic gates are electronic circuits that operate on one or more physical input signals to produce an output signal. The graphic symbols used to designate the three types of gates and their input –output signals are shown in Fig.



Symbols for digital logic circuits

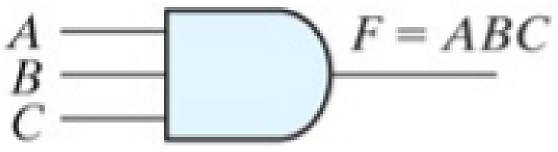


Input–output signals for gates

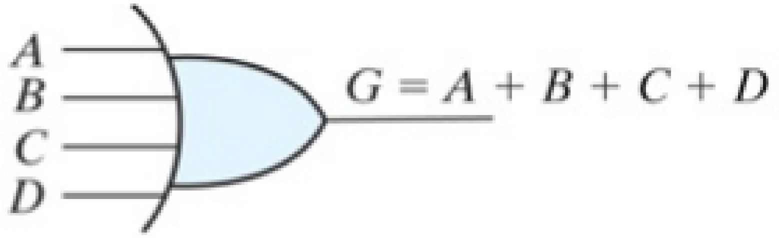
Binary Logic

Gates with multiple inputs

- AND and OR gates may have more than two inputs.
- An AND gate with three inputs and an OR gate with four inputs are shown in Fig..
- The three-input AND gate responds with logic 1 output if all three inputs are logic 1. The output produces logic 0 if any input is logic 0.
- The four-input OR gate responds with logic 1 if any input is logic 1; its output becomes logic 0 only when all inputs are logic 0.



Three-input AND gate



Four-input OR gate

Basic Theorems And Properties Of Boolean Algebra

Table: 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)	$x(x + y) = x$

Basic Theorems And Properties Of Boolean Algebra

THEOREM 1(a): x + x = x .

Statement	Justification
$x + x = (x + x) \cdot 1$	postulate 2(b)
$= (x + x)(x + x')$	5(a)
$= x + x x'$	4(b)
$= x + 0$	5(b)
$= x$	2(a)

THEOREM 1(b): x . x = x

Statement	Justification
$x \cdot x = x x + 0$	postulate 2(a)
$= x x + x x'$	5(b)
$= x(x + x')$	4(a)
$= x \cdot 1$	5(a)
$= x$	2(b)

Note that theorem 1(b) is the dual of theorem 1(a) and that each step of the proof in part (b) is the dual of its counterpart in part (a).

Basic Theorems And Properties Of Boolean Algebra

THEOREM 2(a): $x + 1 = 1$

Statement	Justification
$x + 1 = 1 \cdot (x + 1)$	postulate 2(b)
$= (x + x')(x + 1)$	5(a)
$= x + x' \cdot 1$	4(b)
$= x + x'$	2(b)
$= 1$	5(a)

THEOREM 2(b): $x . 0 = 0$ by duality

THEOREM 6(a): $x + x y = x$

Statement	Justification
$x + x y = x \cdot 1 + x y$	postulate 2(b)
$= x (1 + y)$	4(a)
$= x (y + 1)$	3(a)
$= x \cdot 1$	theorem 2(a)
$= x$	2(b)

THEOREM 6(b): $x (x + y) = x$ by duality

Basic Theorems And Properties of Boolean Algebra

The theorems of Boolean algebra can be proven by means of **truth tables**. In truth tables, both sides of the relation are checked to see whether they yield identical results for all possible combinations of the variables involved. The following truth table verifies the first absorption theorem:

THEOREM 6(a): x + x y = x

X	Y	XY	X+XY
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

THEOREM 6(b): x (x + y) = x

X	Y	X+Y	X(X+Y)
0	0	0	0
0	1	1	0
1	0	1	1
1	1	1	1

Demorgan’s Theorem (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

Basic Theorems And Properties of Boolean Algebra

Operator Precedence

The operator precedence for evaluating Boolean expressions is

(1) parentheses, (2) NOT, (3) AND, and (4) OR.

In other words, expressions inside parentheses must be evaluated before all other operations.

The next operation that holds precedence is the complement, and then follows the AND and, finally, the OR.

As an example, consider the truth table for one of DeMorgan’s theorems.

The left side of the expression is $(x + y)'$. Therefore, the expression inside the parentheses is evaluated first and the result then complemented.

The right side of the expression is $x'y'$, so the complement of x and the complement of y are both evaluated first and the result is then ANDed.

Basic Theorems And Properties of Boolean Algebra

Practice Exercises

1. Using the basic theorems and postulates of Boolean algebra, simplify the following Boolean expression: $F = x'y'z + xyz + x'yz + xy'z$.

$F = (x' + x)\, y'z + (x' + x)\, yz$

$F = y'z + yz$

$F = (y' + y)\, z$

$F = z$

Answer: $F = z$

Basic Theorems And Properties of Boolean Algebra

Practice Exercises

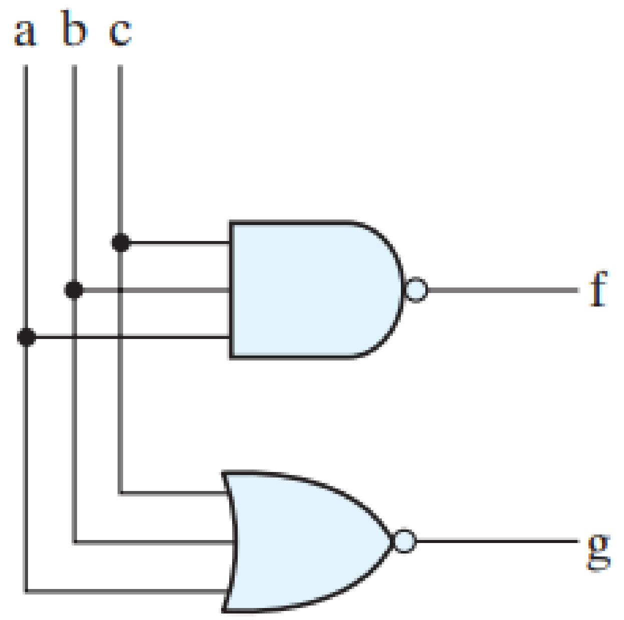
2. Develop a truth table for the Boolean expression F = x ′ y ′ z .

X	Y	Z	X ′	Y ′	x ′ y ′	x ′ y ′ z
0	0	0	1	1	1	0
0	0	1	1	1	1	1
0	1	0	1	0	0	0
0	1	1	1	0	0	0
1	0	0	0	1	0	0
1	0	1	0	1	0	0
1	1	0	0	0	0	0
1	1	1	0	0	0	0

Basic Theorems And Properties of Boolean Algebra

Practice Exercises

By means of a timing diagram, show the signals of the outputs f and g in Fig as functions of the three inputs a, b, and c. Use all eight possible combinations of a, b, and c.

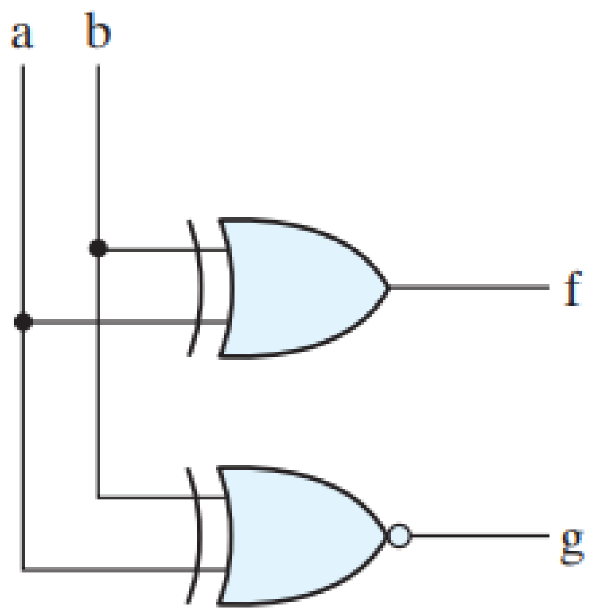


a	b	c	f=(abc)'	g=(a+b+c)'
0	0	0	1	1
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	0	0

Basic Theorems And Properties of Boolean Algebra

Practice Exercises

By means of a timing diagram, show the signals of the outputs f and g in Fig as functions of the two inputs a and b. Use all four possible combinations of a and b.



a	b	f=a xor b	G=a xnor b
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

B O O L E A N F U N C T I O N S

- A Boolean function described by an **algebraic expression** consists of
  - ✓ binary variables,
  - ✓ the constants 0 and 1, and
  - ✓ the logic operation symbols.
- For a given value of the binary variables, the function can be equal to either 1 or 0.
- As an example, consider the Boolean function

  $F1 = x + yz$
- The function F1 is equal to 1 if x is equal to 1 or if both y and z are equal to 1. F1 is equal to 0 otherwise.
- The complement operation dictates that when $y = 1$, $y = 0$. Therefore, $F1 = 1$ if $x = 1$ or if $y = 0$ and $z = 1$.
- A Boolean function expresses the logical relationship between binary variables and is evaluated by determining the binary value of the expression for all possible values of the variables.

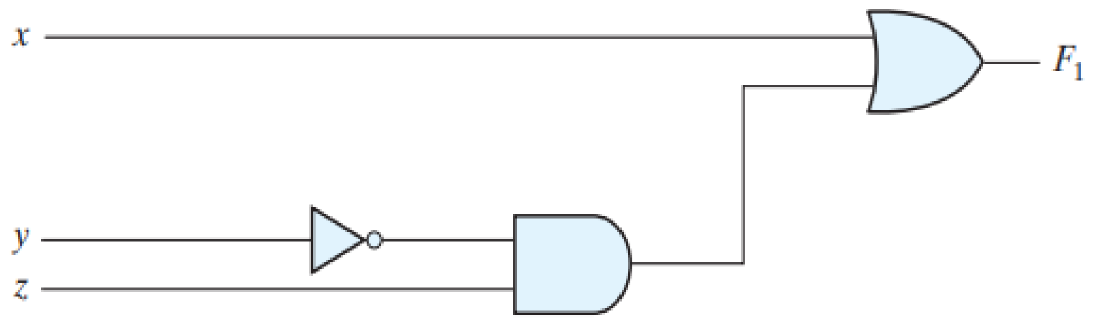
BOOLEANFUNCTIONS

A Boolean function can be represented in a **truth table**.

- The number of rows in the truth table is 2^n , where n is the number of variables in the function.
- The binary combinations for the truth table are obtained from the binary numbers by counting from 0 to $2^n - 1$.
- Table shows the truth table for the function F1.
- There are eight possible binary combinations for assigning bits to the three variables x, y, and z.
- The column labeled F1 contains either 0 or 1 for each of these combinations.
- The table shows that the function is equal to 1 when x = 1 or when yz = 01 and is equal to 0 otherwise.
- A Boolean function can be transformed from an algebraic expression into a circuit diagram composed of logic gates connected in a particular structure.

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

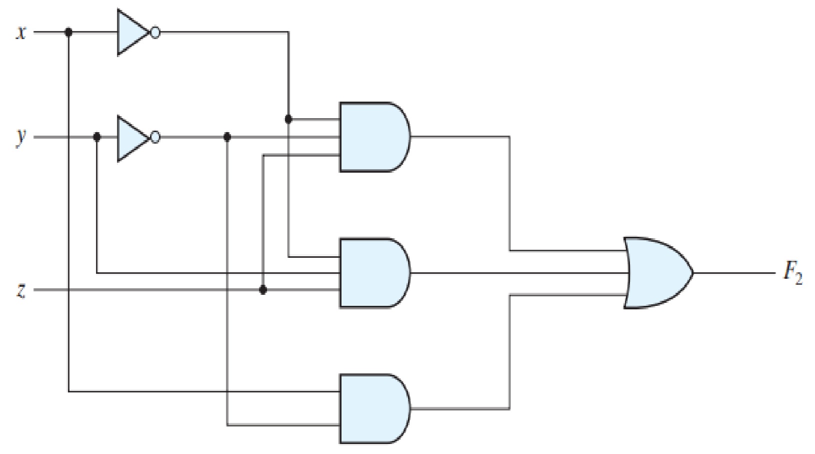


B O O L E A N F U N C T I O N S

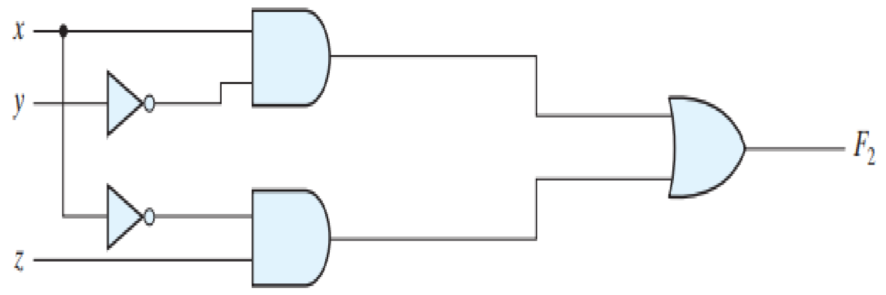
Simplify the function.

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

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



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



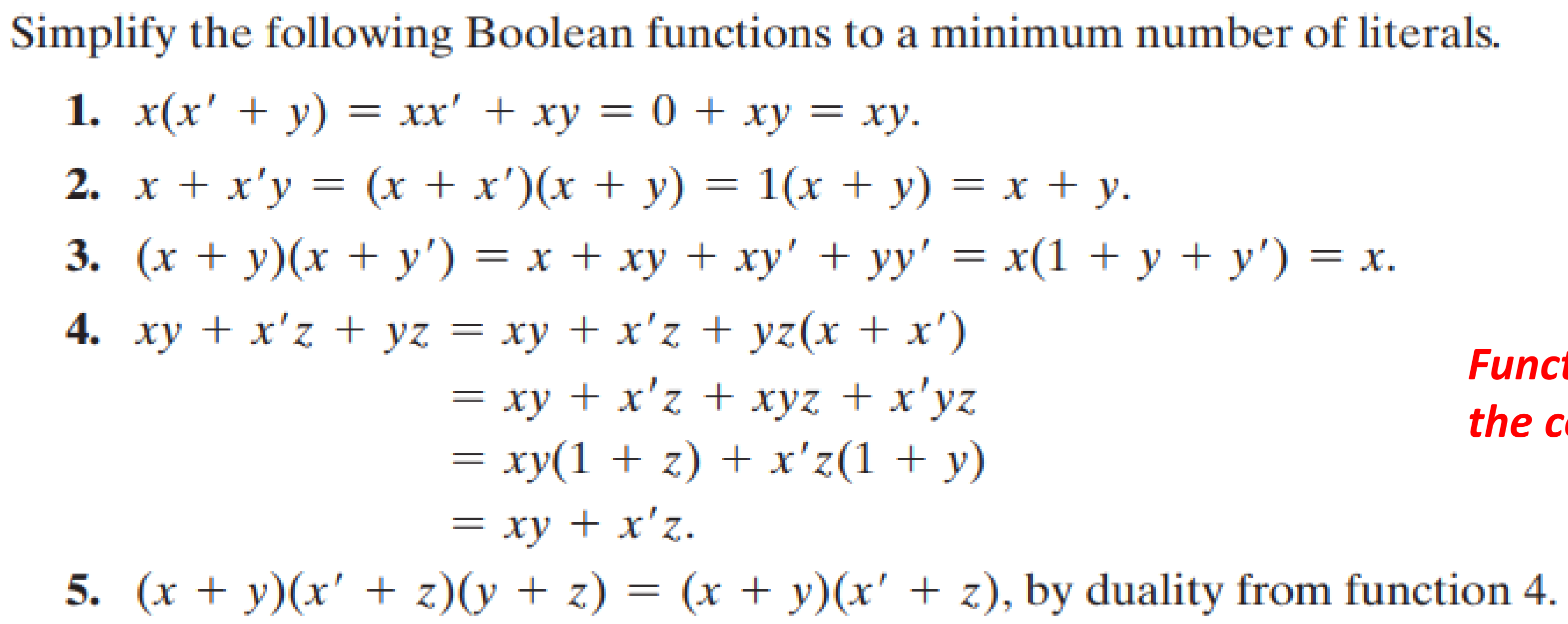
(b) $F_2 = xy' + x'z$

B O O L E A N F U N C T I O N S

Algebraic Manipulation

By reducing the number of terms, the number of literals, or both in a Boolean expression, it is often possible to obtain a simpler circuit.

The manipulation of Boolean algebra consists of reducing an expression for the purpose of obtaining a simpler circuit.



Simplify the following Boolean functions to a minimum number of literals.

1. $x(x' + y) = xx' + xy = 0 + xy = xy.$
2. $x + x'y = (x + x')(x + y) = 1(x + y) = x + y.$
3. $(x + y)(x + y') = x + xy + xy' + yy' = x(1 + y + y') = x.$
4. $$\begin{aligned} xy + x'z + yz &= xy + x'z + yz(x + x') \\ &= xy + x'z + xyz + x'yz \\ &= xy(1 + z) + x'z(1 + y) \\ &= xy + x'z. \end{aligned}$$
5. $(x + y)(x' + z)(y + z) = (x + y)(x' + z)$, by duality from function 4.

Functions 4 and 5 are together known as the consensus theorem.

BOOLEANFUNCTIONS

Complement of a Function

The complement of a function F is F’ and is obtained from an interchange of 0’s for 1’s and 1’s for 0’s in the value of F.

The complement of a function may be derived algebraically through DeMorgan’s theorems.

The three-variable form of the first DeMorgan’s theorem is derived as follows, from the relevant postulates and theorems:

$$
\begin{aligned}
(A + B + C)' &= (A + x)' && \text{let } B + C = x \\
&= A'x' && \text{by theorem 5(a) (DeMorgan)} \\
&= A'(B + C)' && \text{substitute } B + C = x \\
&= A'(B'C') && \text{by theorem 5(a) (DeMorgan)} \\
&= A'B'C' && \text{by theorem 4(b) (associative)}
\end{aligned}
$$

$$(A + B + C + D + \cdots + F)' = A'B'C'D' \ldots F'$$

$$(ABCD \ldots F)' = A' + B' + C' + D' + \cdots + F'$$

The generalized form of DeMorgan’s theorems states that the complement of a function is obtained by interchanging AND and OR operators and complementing each literal.

B O O L E A N F U N C T I O N S

Find the complement of the following functions by applying DeMorgan’s theorems as many times as necessary.

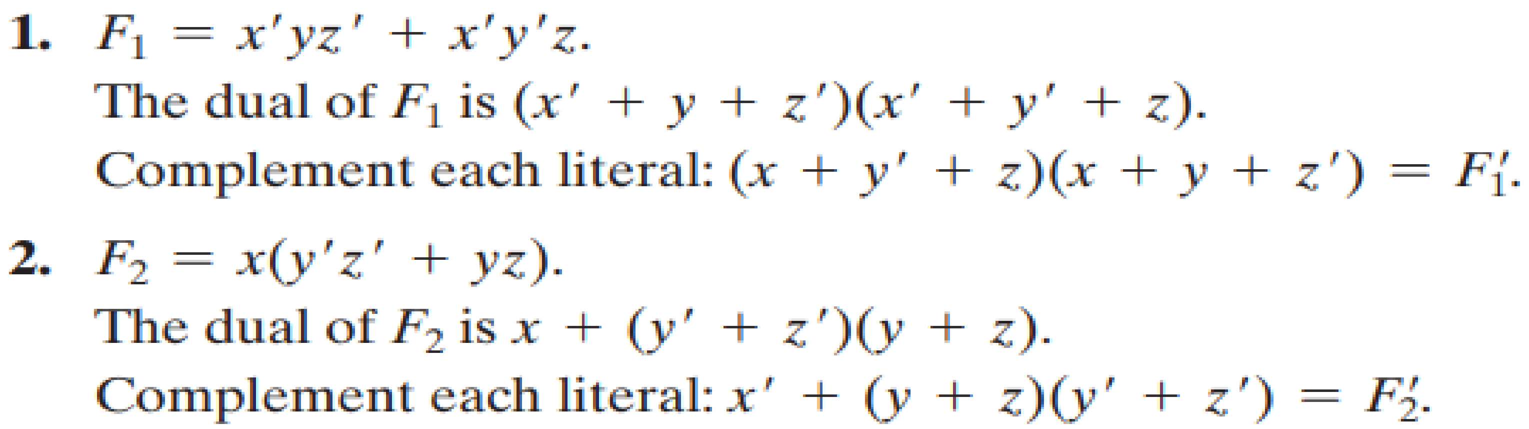
$F_1 = x'yz' + x'y'z$ and $F_2 = x(y'z' + yz)$.

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

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

BOOLEANFUNCTIONS

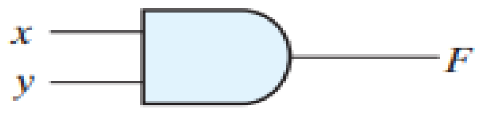
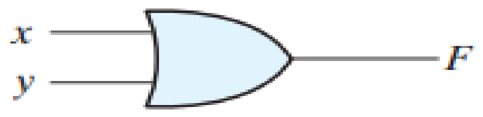
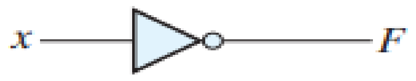
Find the complement of the functions F1 and F2 of previous by taking their duals and complementing each literal.



1. $F_1 = x'yz' + x'y'z$.
   The dual of F_1 is $(x' + y + z')(x' + y' + z)$.
   Complement each literal: $(x + y' + z)(x + y + z') = F_1'$.
2. $F_2 = x(y'z' + yz)$.
   The dual of F_2 is $x + (y' + z')(y + z)$.
   Complement each literal: $x' + (y + z)(y' + z') = F_2'$.

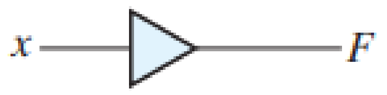
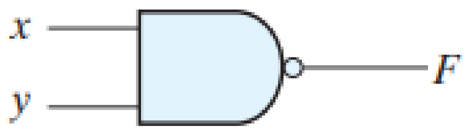
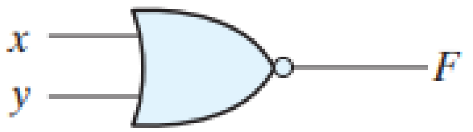
DIGITAL LOGIC GATES

Digital Logic Gates

Name	Graphic symbol	Algebraic function	Truth table
AND		$F = x \cdot y$	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

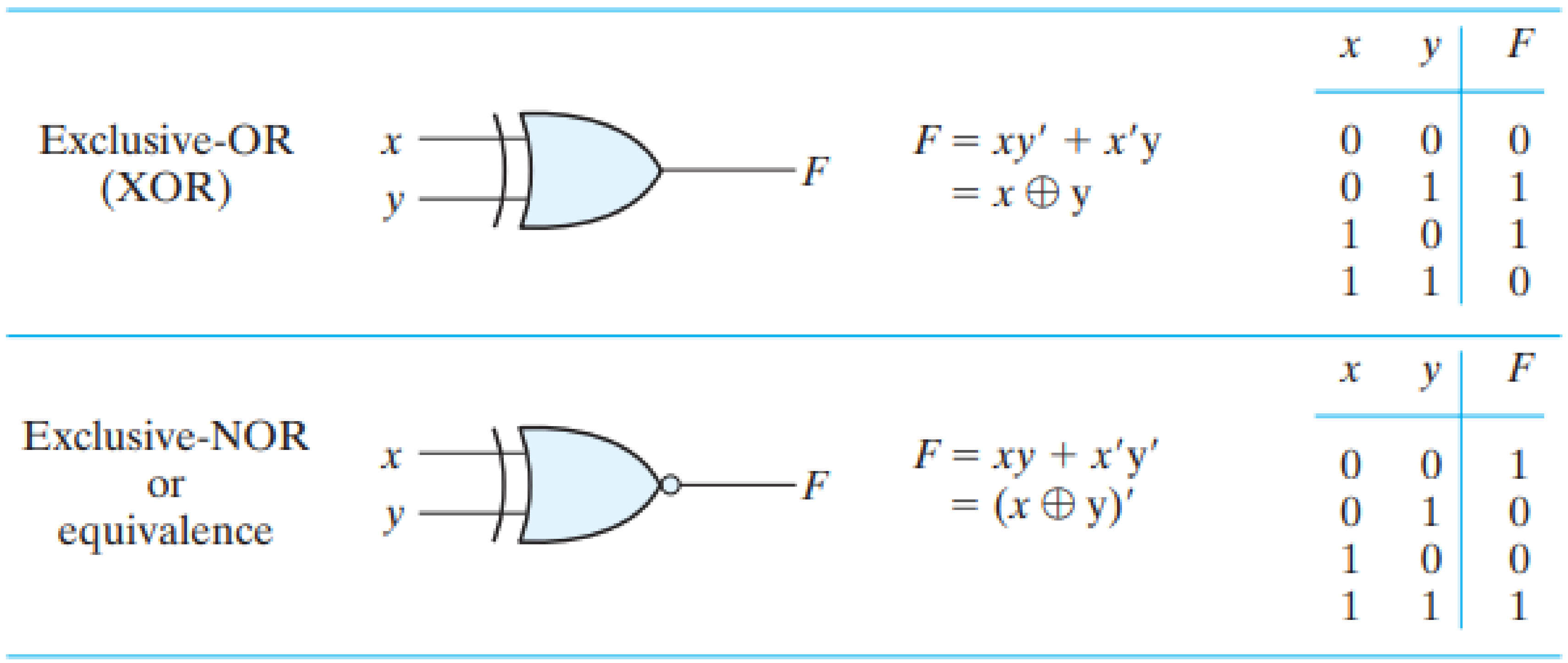
D I G I TAL L O G I C G AT E S

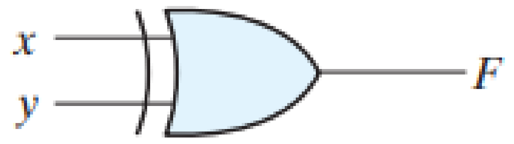
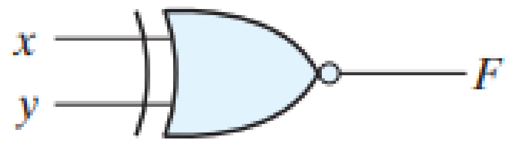
Digital Logic Gates

Buffer		$F = x$	x | F 0 | 0 1 | 1
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

DIGITAL LOGIC GATES

Digital Logic Gates



			x	y	F
Exclusive-OR (XOR)		$F = xy' + x'y$ $= x \oplus y$	0	0	0
			0	1	1
			1	0	1
			1	1	0
Exclusive-NOR or equivalence		$F = xy + x'y'$ $= (x \oplus y)'$	0	0	1
			0	1	0
			1	0	0
			1	1	1

DIGITAL LOGIC GATES

Extension to Multiple Inputs

The gates **except for the inverter and buffer** can be extended to have more than two inputs.

A gate can be extended to have multiple inputs if the binary operation it represents is **commutative and associative.**

The AND and OR operations, defined in Boolean algebra, possess these two properties.

For the OR function, we have $x + y = y + x$ (commutative) and $(x + y) + z = x + (y + z) = x + y + z$ (associative) which indicates that the gate inputs can be interchanged and that the OR function can be extended to three or more variables.

D I G I TA L L O G I C G AT E S