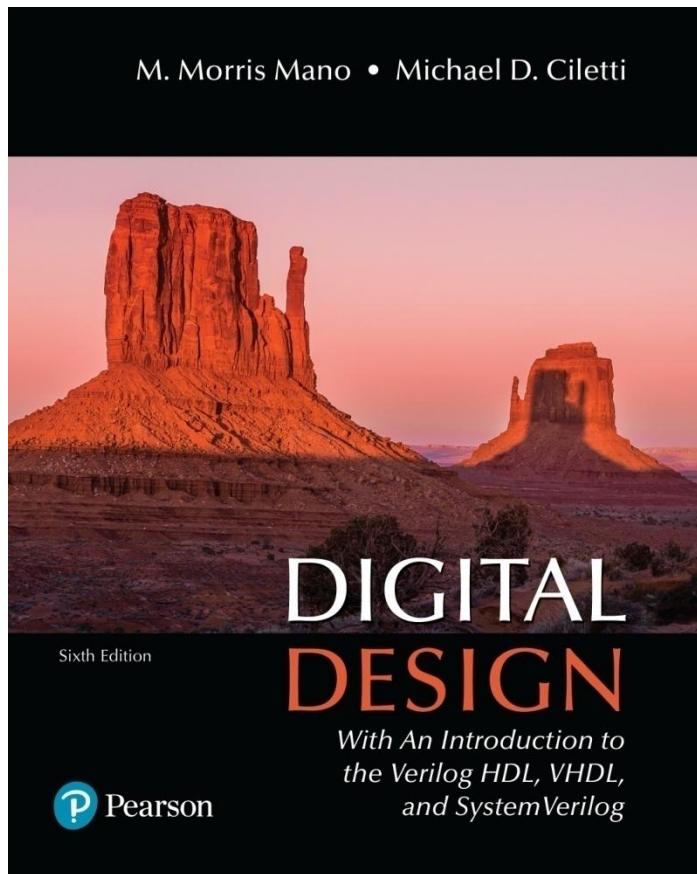


Digital Design

With an Introduction to the Verilog HDL, VHDL, and SystemVerilog

6th Edition



Chapter 02

Boolean Algebra and Logic Gates

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)	$x(x + y) = x$

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

Figure 2.1
Logic diagram for the Boolean function $F_1 = x + y'z$.

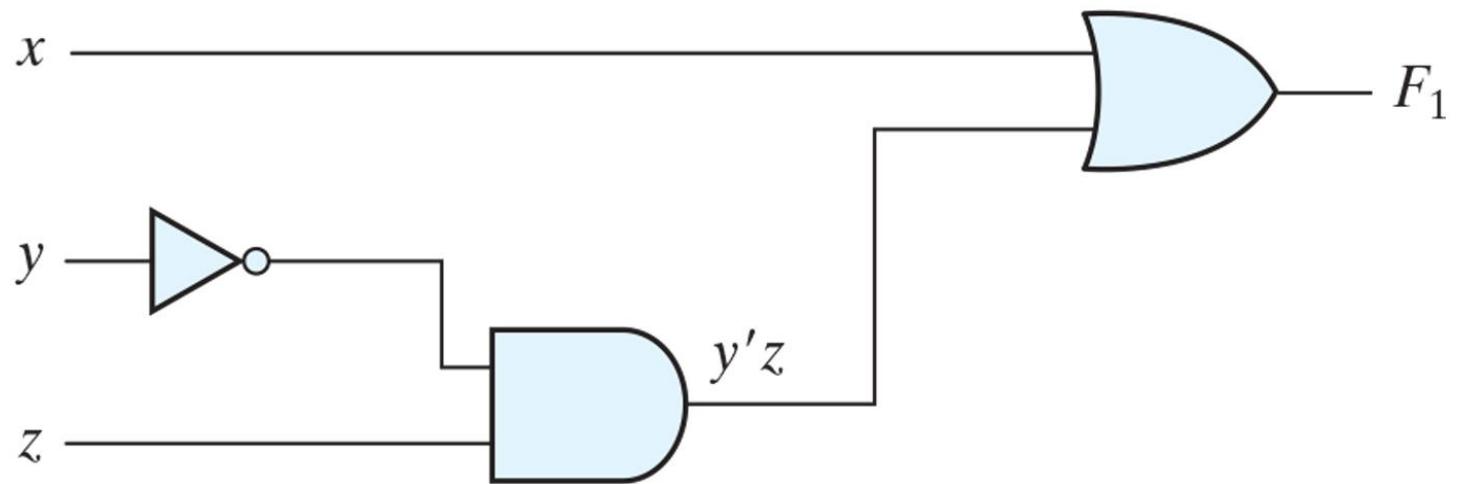
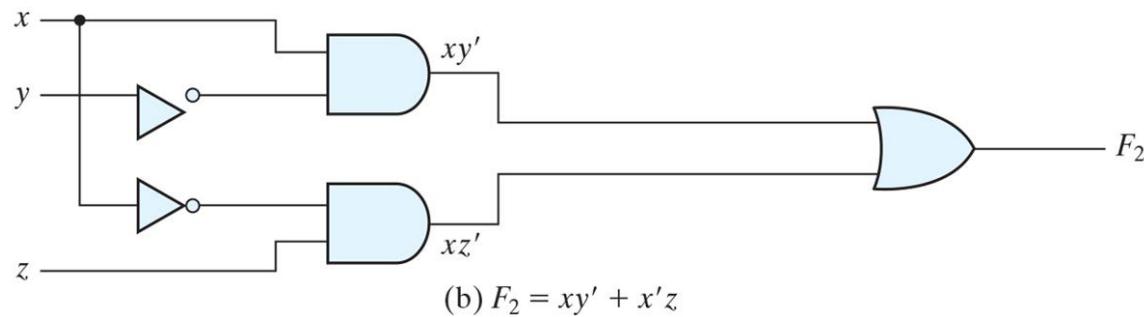
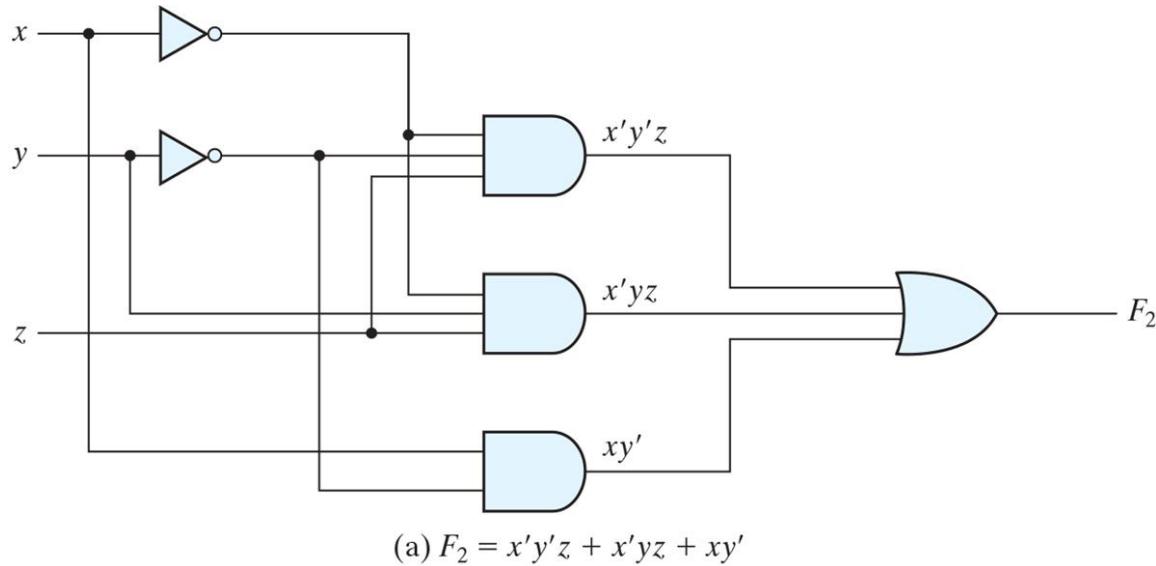


Figure 2.2
Implementation of Boolean function F_2 with gates.



Practice Exercise 2.3

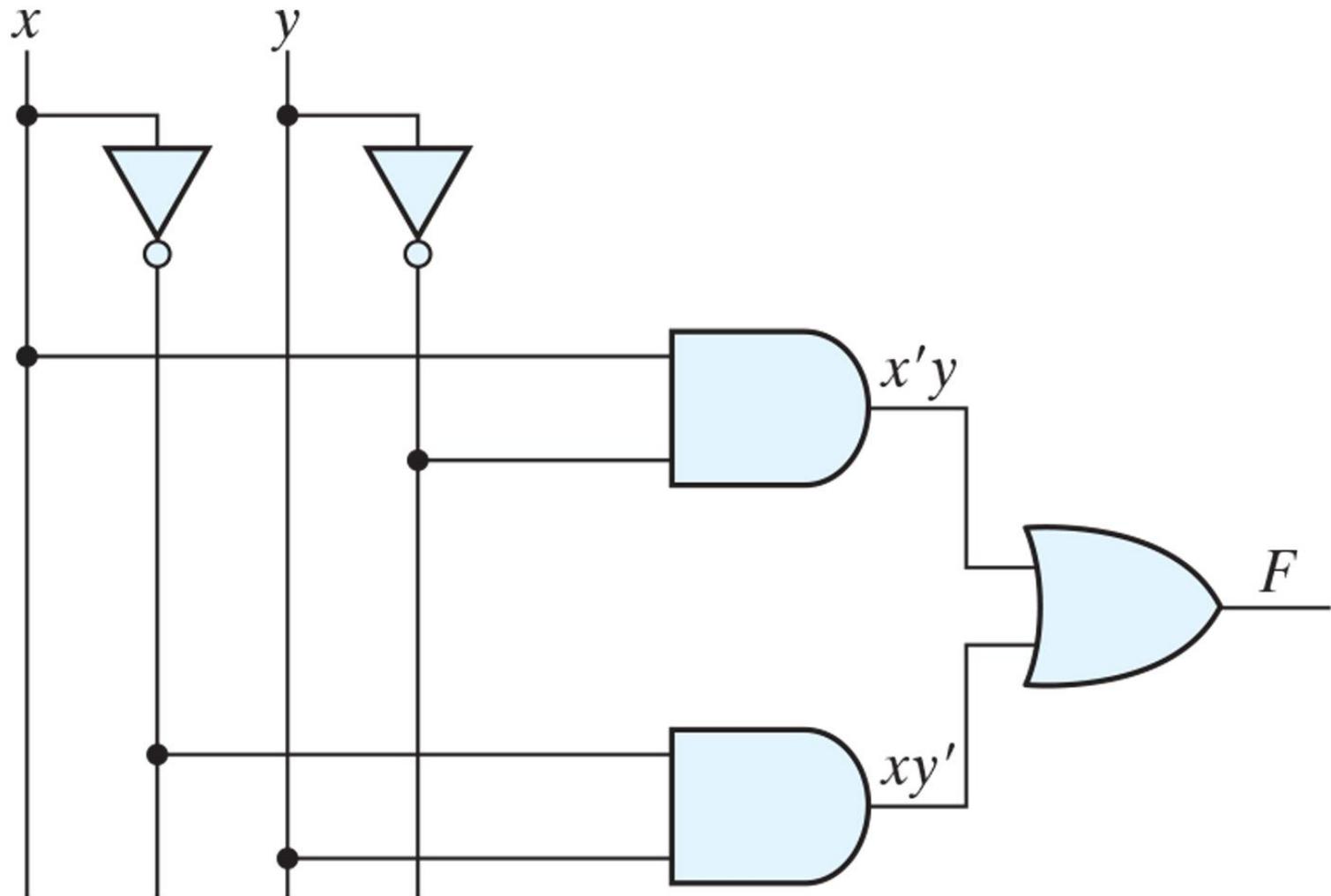


Figure PE2.4

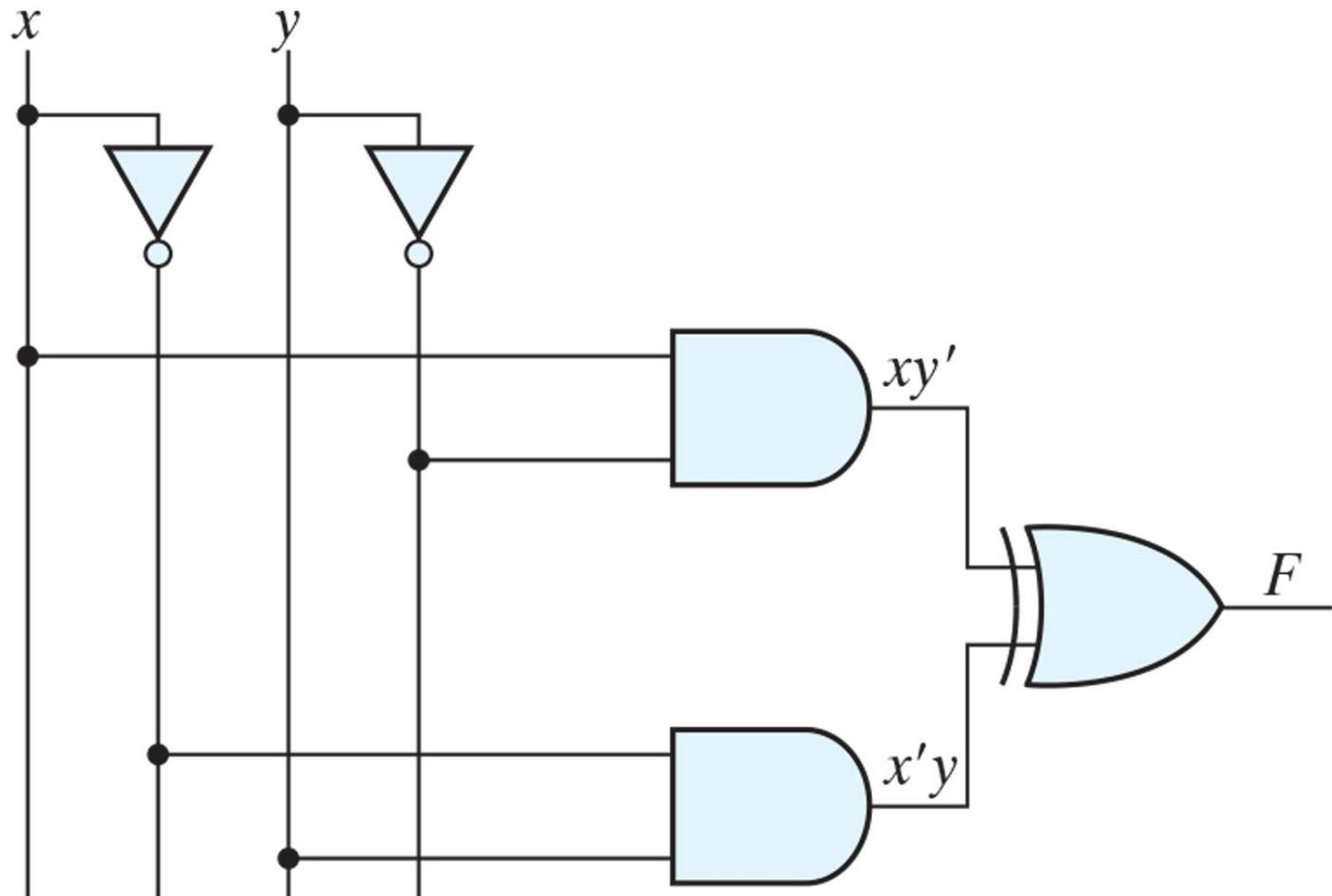


Table 2.3
Minterms and Maxterms for Three Binary Variables.

			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		

Table 2.4
Functions of Three Variables.

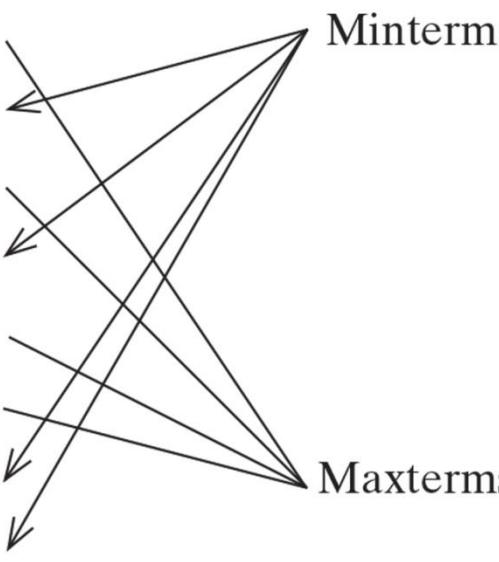
x	y	z	Function f_1	Function f_2
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

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

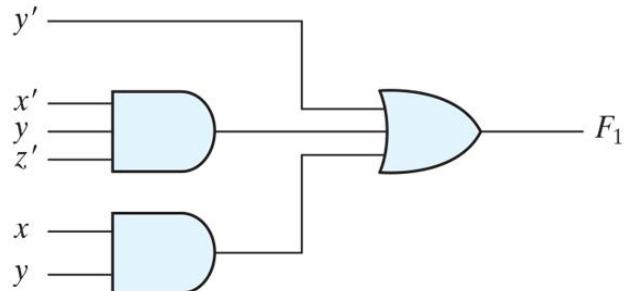
Table 2.6
Truth Table for $F = xy + x'z$.

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

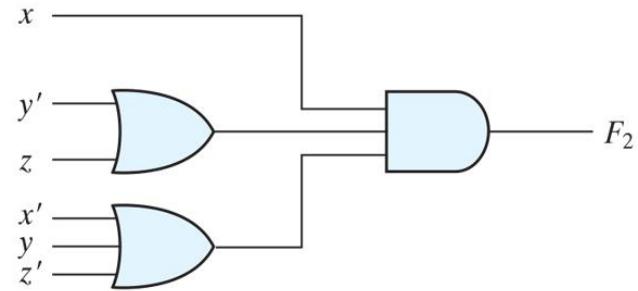


The diagram illustrates the relationship between the Minterms and Maxterms representations of the function F . It shows two sets of arrows originating from the rows of the truth table and pointing to the 'F' column. One set of arrows originates from the rows where $F=1$ (Minterms: m_1, m_3, m_7) and points to the value 1 in the 'F' column. Another set of arrows originates from the rows where $F=0$ (Maxterms: M_0, M_2, M_4, M_5, M_6) and points to the value 0 in the 'F' column.

Figure 2.3
Two-level implementation.

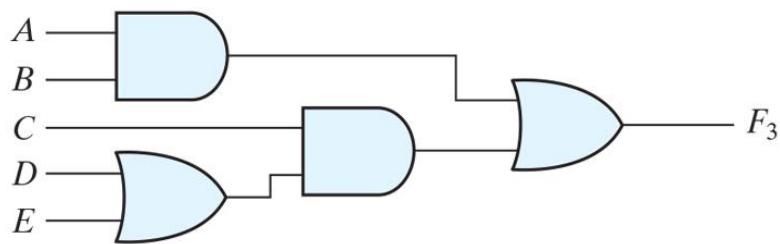


(a) Sum of Products

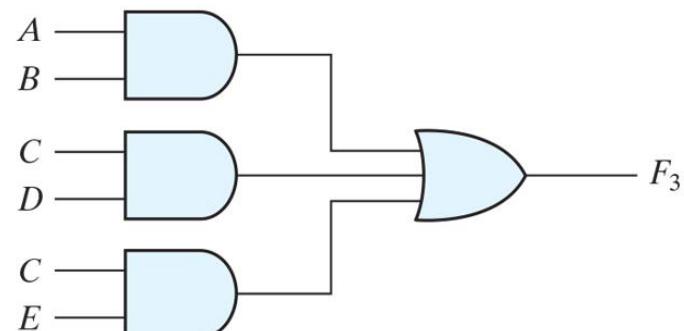


(b) Product of Sums

Figure 2.4
Three- and two-level implementation.



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



$$(b) AB + CD + CE$$

Practice Exercise 2.12

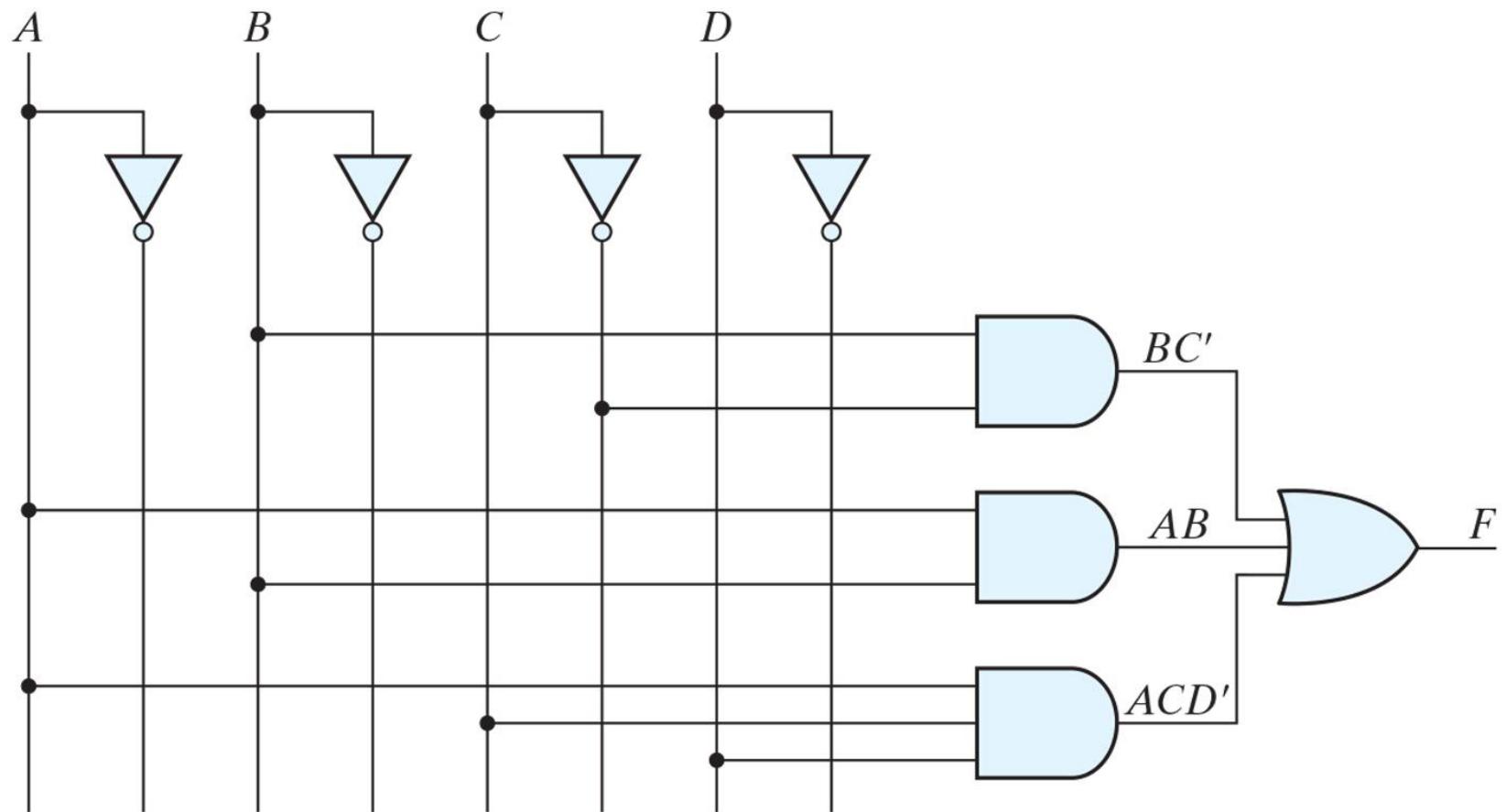


Table 2.7
Truth Tables for the 16 Functions of Two Binary Variables.

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	1	1	1	1	1	1	1	1	1
0	1	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1
1	0	0	0	1	1	0	0	1	0	0	1	1	0	0	1	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

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

Figure 2.5

Digital logic gates.

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

Figure 2.6
Demonstrating the nonassociativity of the NOR operator: $(x \downarrow y) \downarrow z \neq x \downarrow(y \downarrow z)$.

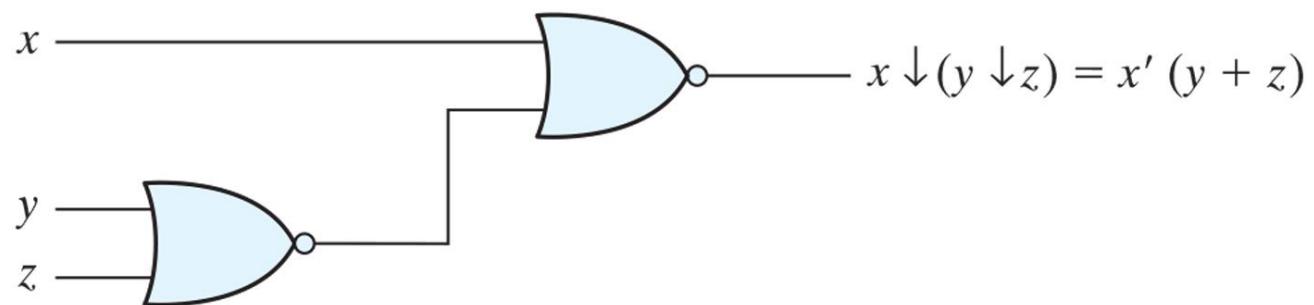
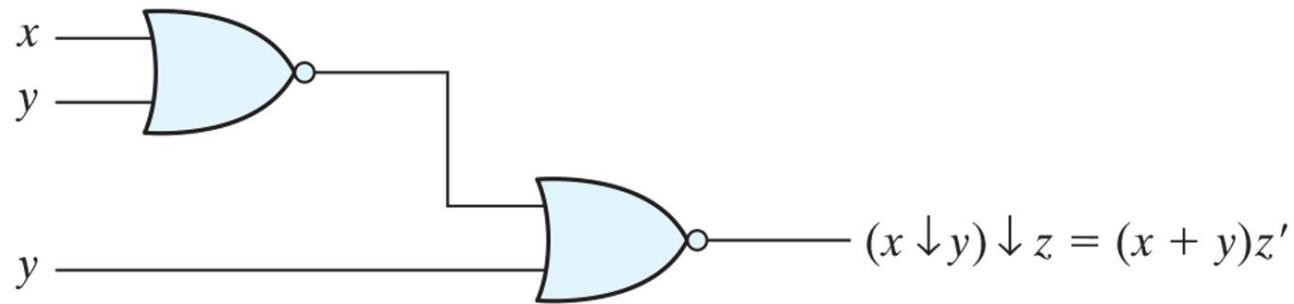
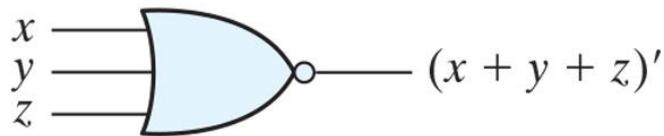
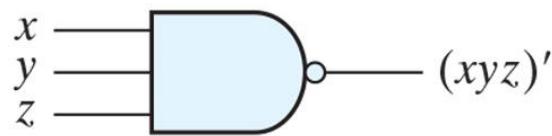


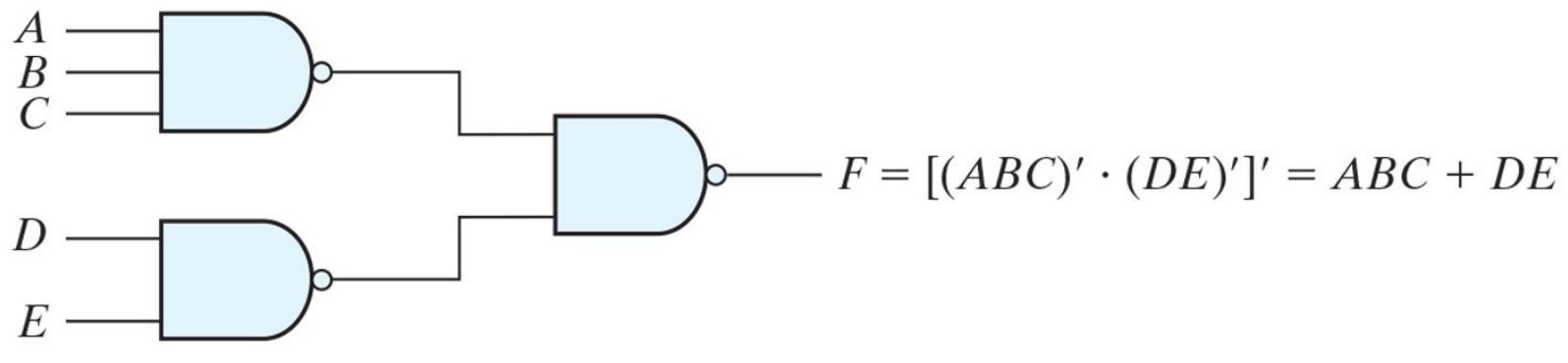
Figure 2.7
Multiple-input and cascaded NOR and NAND gates.



(a) 3-input NOR gate

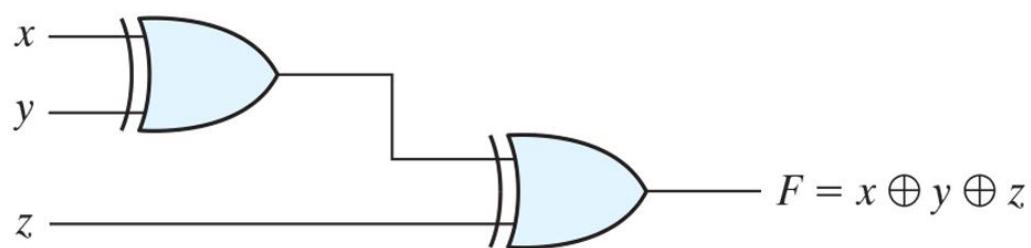


(b) 3-input NAND gate

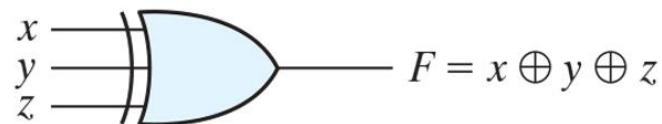


(c) Cascaded NAND gates

Figure 2.8
Three-input exclusive-OR 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

Figure 2.9
Signal assignment and logic polarity.

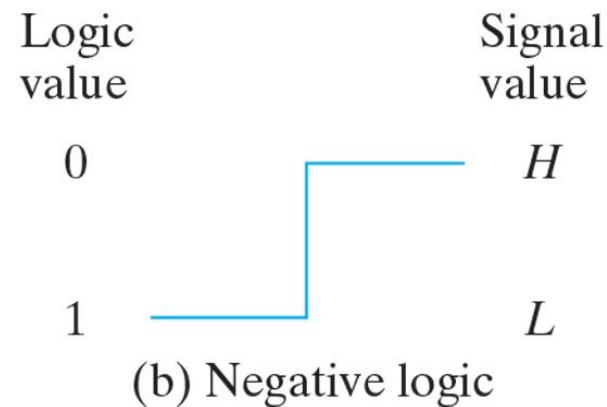
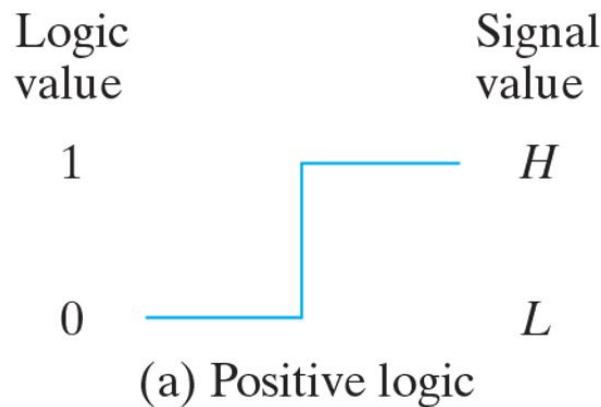
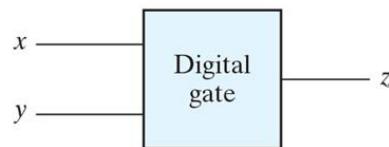


Figure 2.10
Demonstration of positive and negative logic.

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

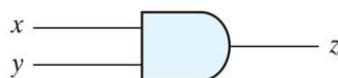
(a) Truth table
with H and L



(b) Gate block diagram

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

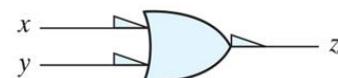
(c) Truth table for
positive logic



(d) Positive logic AND gate

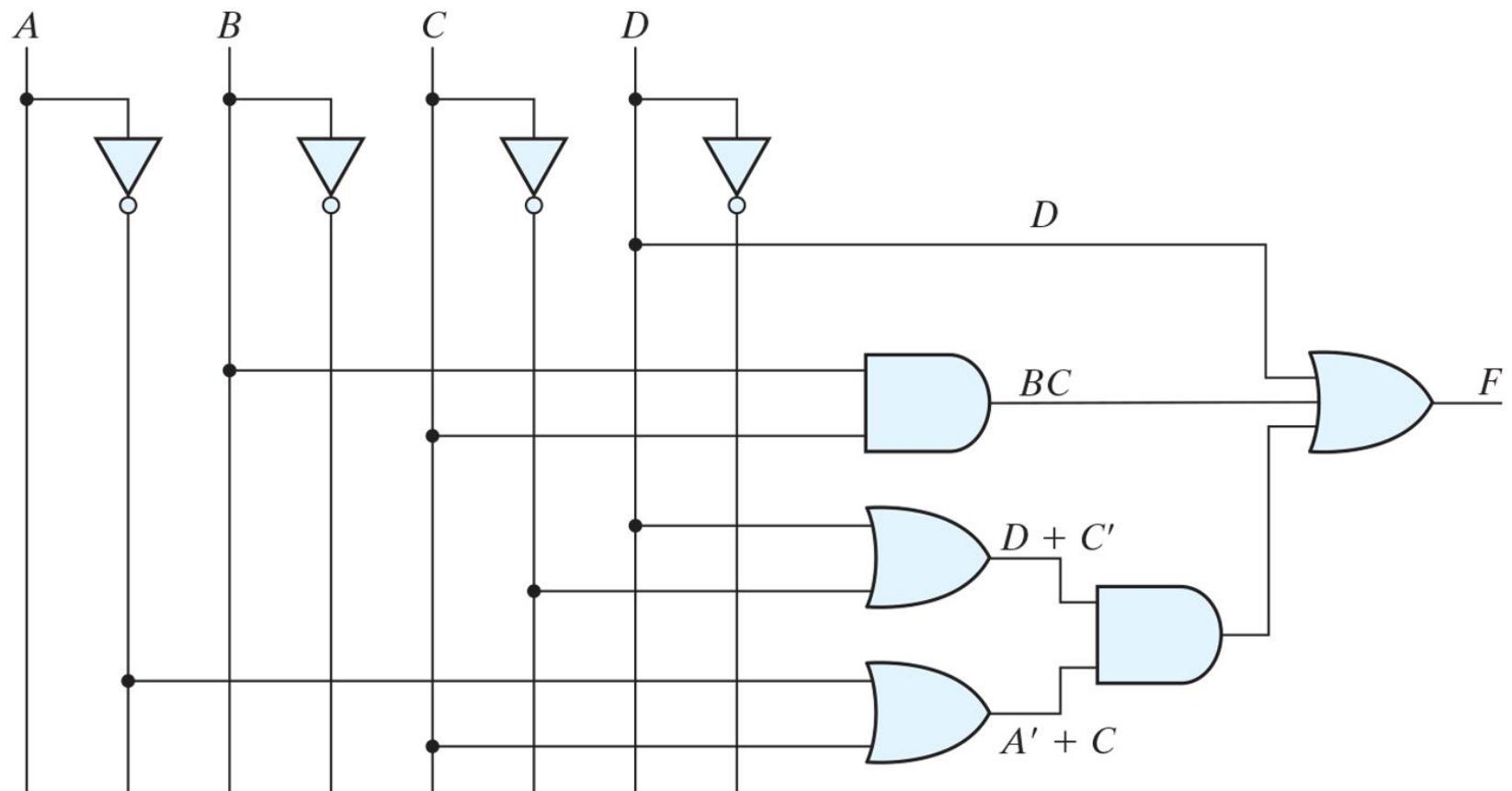
x	y	z
1	1	1
1	0	1
0	1	1
0	0	0

(e) Truth table for
negative logic



(f) Negative logic OR gate

Practice Exercise 2.13



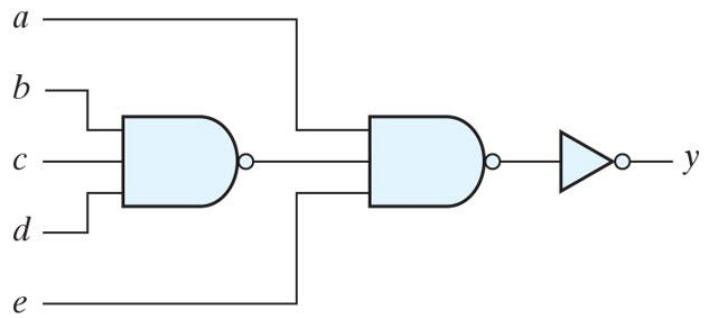
Problem 2.15

A	B	C	T₁	T₂
0	0	0	1	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	0	1

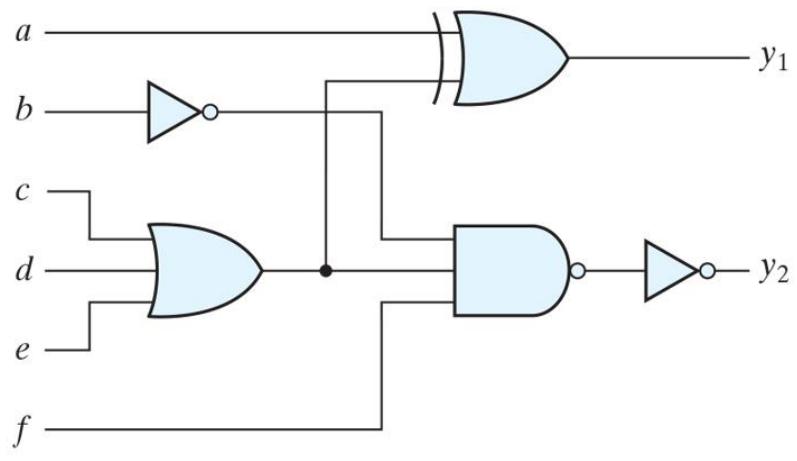
Table P2.27

f_1	f_2	a	b	c
1	1	0	0	0
0	1	0	0	1
1	0	0	1	0
1	1	0	1	1
1	0	1	0	0
0	1	1	0	1
1	0	1	1	1

Figure P2.28



(a)



(b)

Figure P2.32

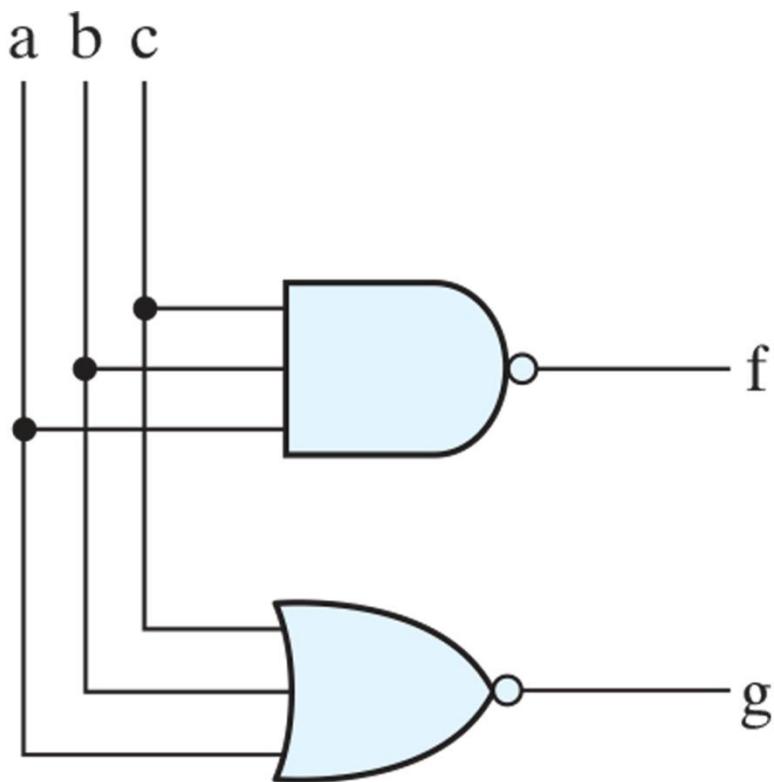


Figure P2.33

