TABLE 2-1
Truth Tables for the Three Basic Logic Operations

AND

X	Y	$Z = X \cdot Y$
0	0	0
0	1	0
1	0	0
1	1	1

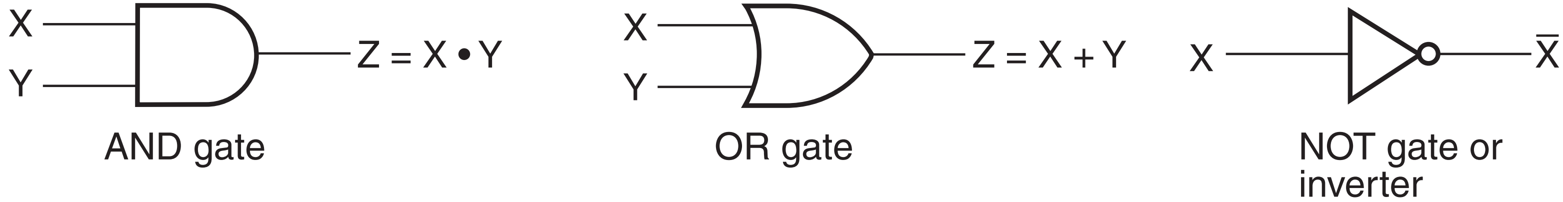
OR

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

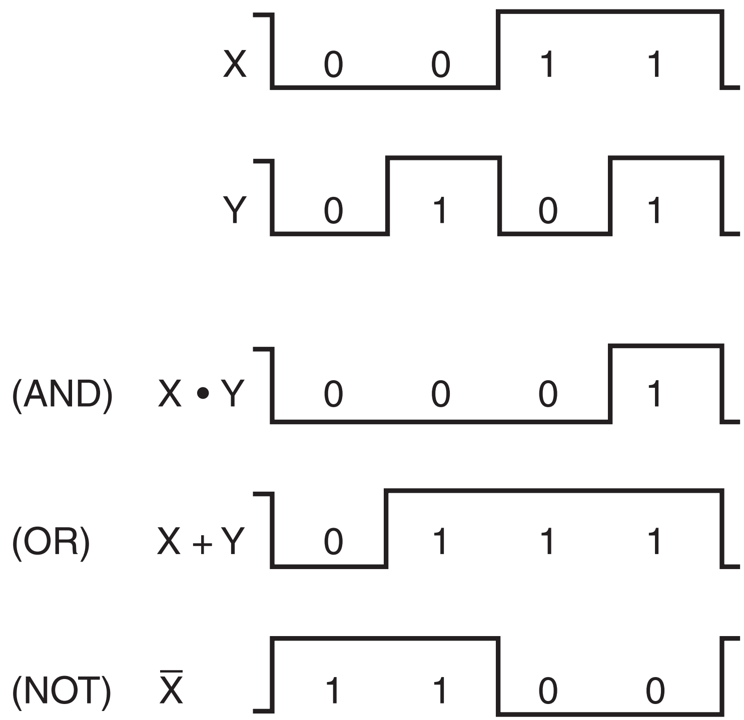
NOT

X	$Z = \overline{X}$
0	1
1	0

Table 2-1 Truth Tables for the Three Basic Logical Operations



(a) Graphic symbols



(b) Timing diagram

Fig. 2-1 Digital Logic Gates

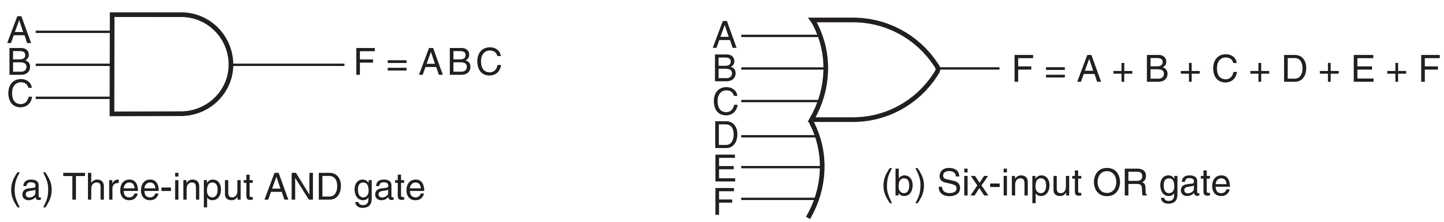


Fig. 2-2 Gates with More than Two Inputs

TABLE 2-2
Truth Table
for the Function $F = X + \overline{Y}Z$

X	Y	Z	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 2-2 Truth Table for the Function $F = X + \overline{Y}Z$

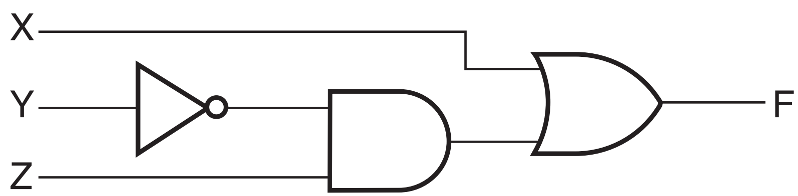


Fig. 2-3 Logic Circuit Diagram for $F = X + \overline{Y}Z$

TABLE 2-3
Basic Identities of Boolean Algebra

1.	$X+0=X$	2.	$X\cdot 1=X$	
3.	$X+1=1$	4.	$X\cdot 0=0$	
5.	$X+X=X$	6.	$X\cdot X=X$	
7.	$X+\overline{X}=1$	8.	$X\cdot\overline{X}=0$	
9.	$\overline{\overline{X}}=X$			
10.	$X+Y=Y+X$	11.	$XY=YX$	Commutative
12.	$X+(Y+Z)=(X+Y)+Z$	13.	$X(YZ)=(XY)Z$	Associative
14.	$X(Y+Z)=XY+XZ$	15.	$X+YZ=(X+Y)(X+Z)$	Distributive
16.	$\overline{X+Y}=\overline{X}\cdot\overline{Y}$	17.	$\overline{X\cdot Y}=\overline{X}+\overline{Y}$	DeMorgan's

Table 2-3 Basic Identities of Boolean Algebra

TABLE 2-4
Truth Tables to Verify DeMorgan's Theorem

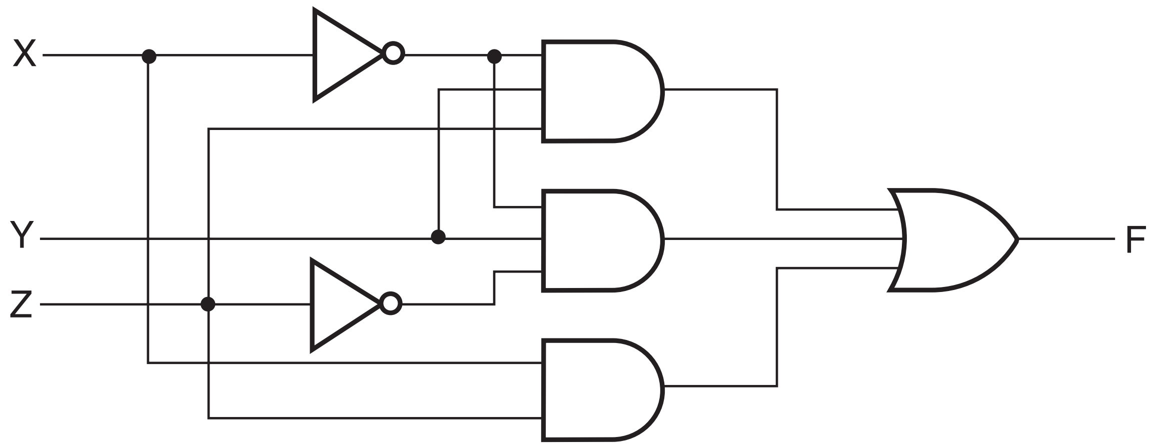
A) X	Y	$X + Y$	$\overline{X + Y}$	B) X	Y	$\overline{X}$	$\overline{Y}$	$\overline{X} \cdot \overline{Y}$
0	0	0	1	0	0	1	1	1
0	1	1	0	0	1	1	0	0
1	0	1	0	1	0	0	1	0
1	1	1	0	1	1	0	0	0

Table 2-4 Truth Tables to Verify DeMorgan's Theorem

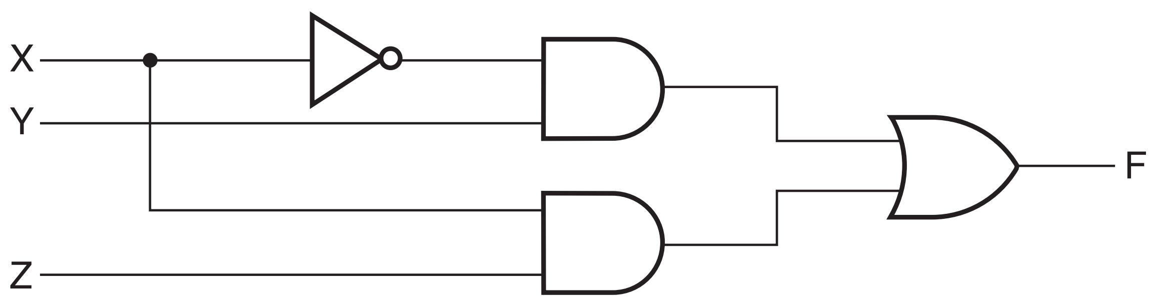
TABLE 2-5
Truth Table for Boolean Function

X	Y	Z	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Table 2-5 Truth Table for Boolean Function



(a) $F = \overline{X}YZ + \overline{X}Y\overline{Z} + XZ$



(b) $F = \overline{X}Y + XZ$

Fig. 2-4 Implementation of Boolean Function with Gates

M. Morris Mano & Charles R. Kime
LOGIC AND COMPUTER DESIGN FUNDAMENTALS, 2e, Updated.

TABLE 2-6
Minterms for Three Variables

X	Y	Z	Product Term	Symbol	m_0	m_1	m_2	m_3	m_4	m_5	m_6	m_7
0	0	0	$\overline{X}\,\overline{Y}\,\overline{Z}$	m_0	1	0	0	0	0	0	0	0
0	0	1	$\overline{X}\,\overline{Y}Z$	m_1	0	1	0	0	0	0	0	0
0	1	0	$\overline{X}Y\overline{Z}$	m_2	0	0	1	0	0	0	0	0
0	1	1	$\overline{X}YZ$	m_3	0	0	0	1	0	0	0	0
1	0	0	$X\overline{Y}\,\overline{Z}$	m_4	0	0	0	0	1	0	0	0
1	0	1	$X\overline{Y}Z$	m_5	0	0	0	0	0	1	0	0
1	1	0	$XY\overline{Z}$	m_6	0	0	0	0	0	0	1	0
1	1	1	XYZ	m_7	0	0	0	0	0	0	0	1

Table 2-6 Minterms for Three Variables

TABLE 2-7
Maxterms for Three Variables

X	Y	Z	Sum Term	Symbol	M_0	M_1	M_2	M_3	M_4	M_5	M_6	M_7
0	0	0	$X+Y+Z$	M_0	0	1	1	1	1	1	1	1
0	0	1	$X+Y+\overline{Z}$	M_1	1	0	1	1	1	1	1	1
0	1	0	$X+\overline{Y}+Z$	M_2	1	1	0	1	1	1	1	1
0	1	1	$X+\overline{Y}+\overline{Z}$	M_3	1	1	1	0	1	1	1	1
1	0	0	$\overline{X}+Y+Z$	M_4	1	1	1	1	0	1	1	1
1	0	1	$\overline{X}+Y+\overline{Z}$	M_5	1	1	1	1	1	0	1	1
1	1	0	$\overline{X}+\overline{Y}+Z$	M_6	1	1	1	1	1	1	0	1
1	1	1	$\overline{X}+\overline{Y}+\overline{Z}$	M_7	1	1	1	1	1	1	1	0

Table 2-7 Maxterms for Three Variables

TABLE 2-8
Boolean Functions of Three Variables

(a) X	Y	Z	F	$\overline{F}$	(b) X	Y	Z	E
0	0	0	1	0	0	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	1	0	0	1	0	1
0	1	1	0	1	0	1	1	0
1	0	0	0	1	1	0	0	1
1	0	1	1	0	1	0	1	1
1	1	0	0	1	1	1	0	0
1	1	1	1	0	1	1	1	0

Table 2-8 Boolean Functions of Three Variables

M. Morris Mano & Charles R. Kime
LOGIC AND COMPUTER DESIGN FUNDAMENTALS, 2e, Updated.

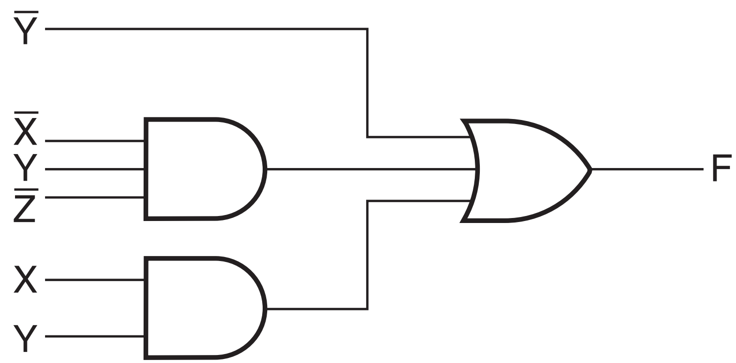
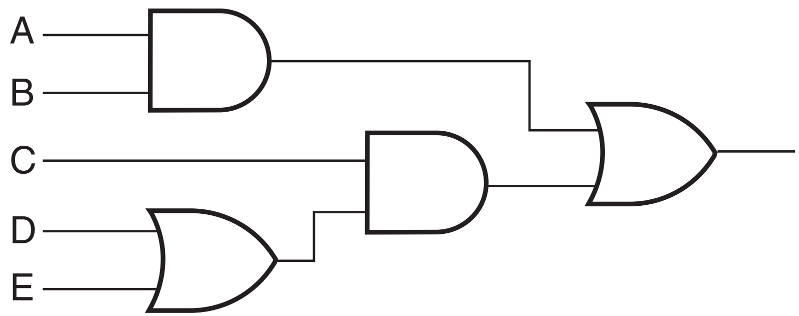
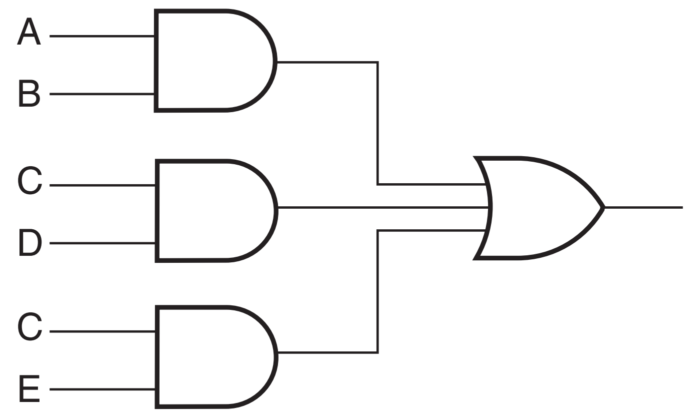


Fig. 2-5 Sum-of-Products Implementation

M. Morris Mano & Charles R. Kime
LOGIC AND COMPUTER DESIGN FUNDAMENTALS, 2e, Updated.



(a) AB + C (D + E)



(b) AB + CD + CE

Fig. 2-6 Three-Level and Two-Level Implementation

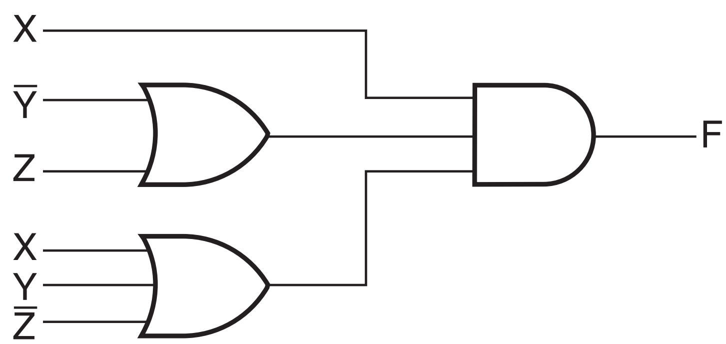
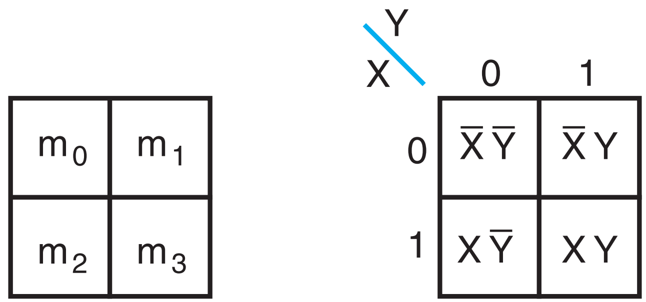
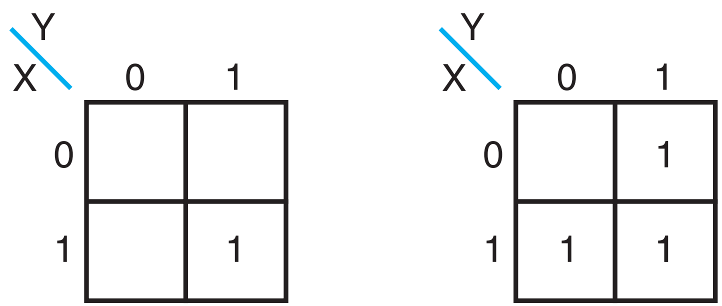


Fig. 2-7 Product-of-Sums Implementation



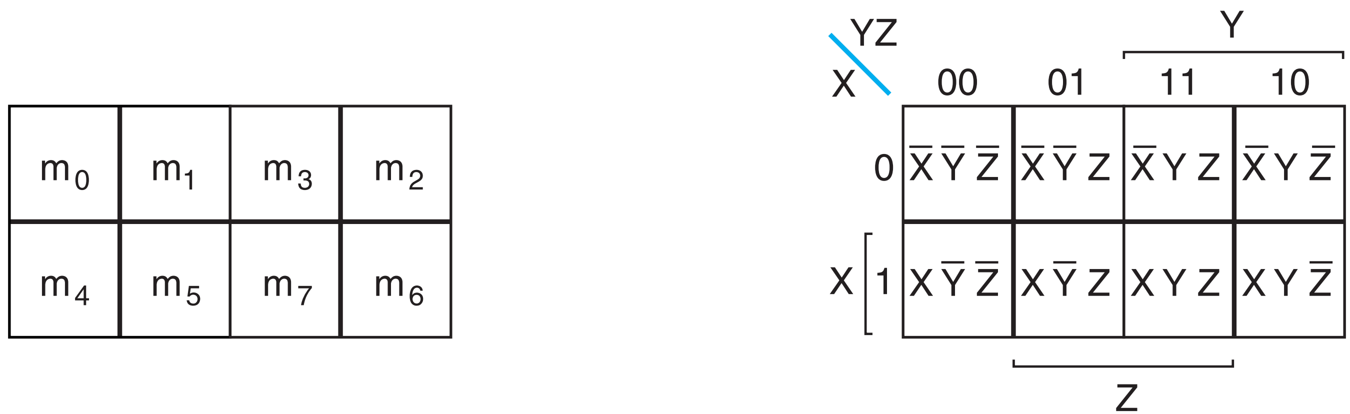
(a) (b)

Fig. 2-8 Two-Variable Map



(a) XY (b) X + Y

Fig. 2-9 Representation of Functions in the Map



(a) (b)

Fig. 2-10 Three-Variable Map

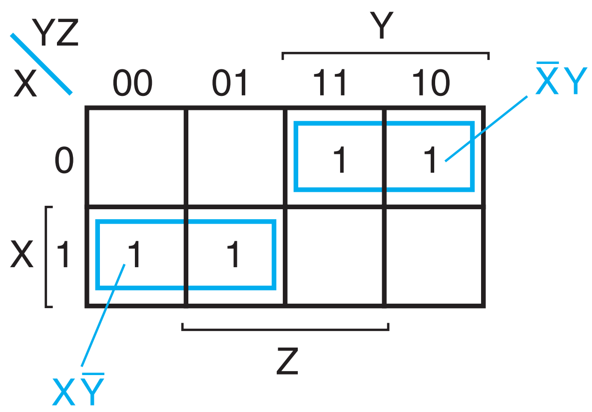


Fig. 2-11 Map for Example 2-3

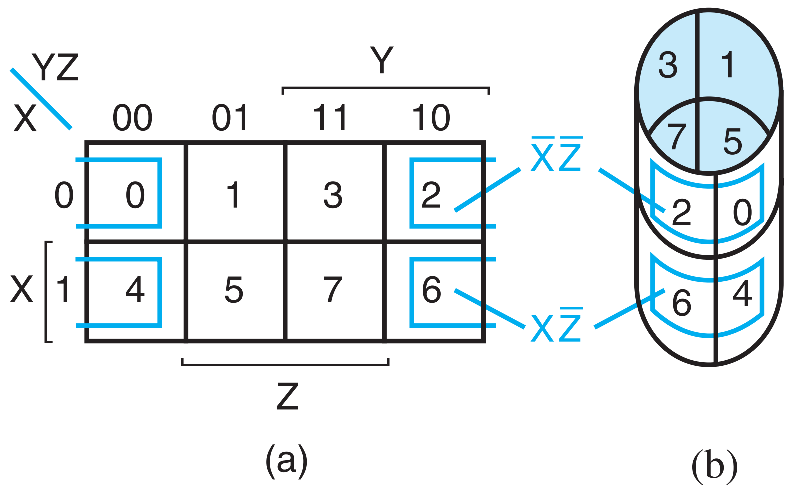


Fig. 2-12 Three-Variable Map: Flat and on a Cylinder to Show Adjacent Squares

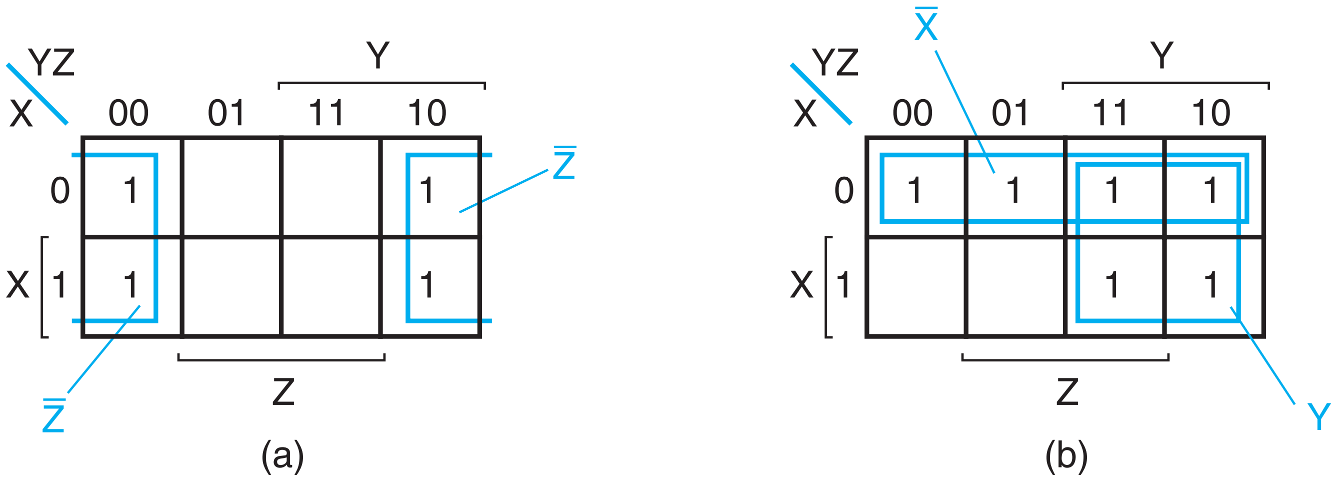
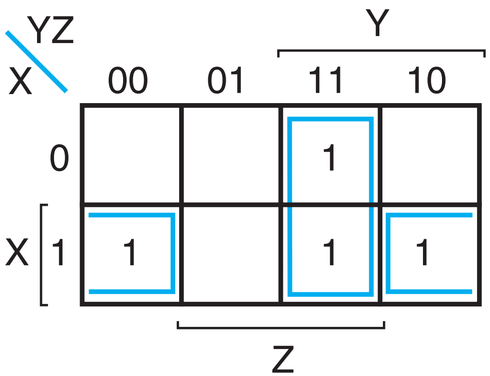
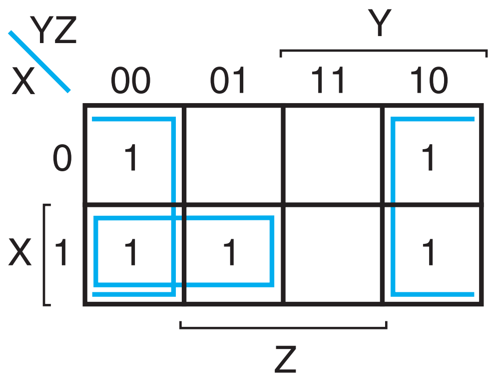


Fig. 2-13 Product Terms Using Four Minterms



(a) $F_1(X, Y, Z) = \Sigma m(3, 4, 6, 7)$
$= YZ + X\overline{Z}$



(b) $F_2(X, Y, Z) = \Sigma m(0, 2, 4, 5, 6)$
$= \overline{Z} + X\overline{Y}$

Fig. 2-14 Maps for Example 2-4

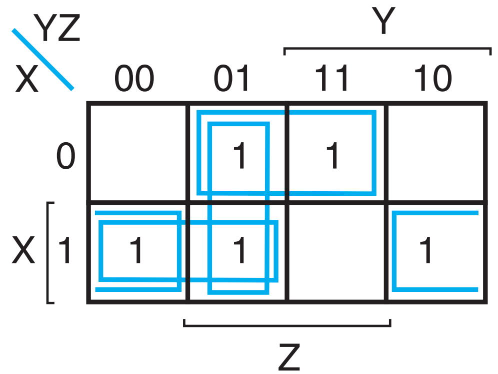


Fig. 2-15 $F(X, Y, Z) = \Sigma m(1, 3, 4, 5, 6)$

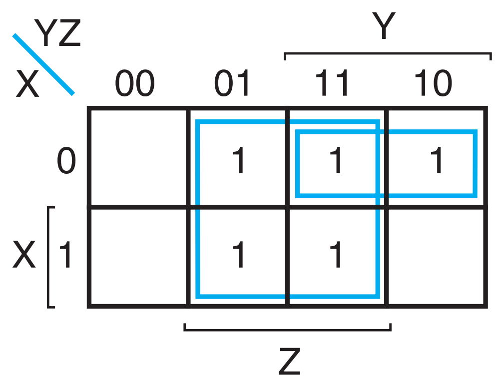
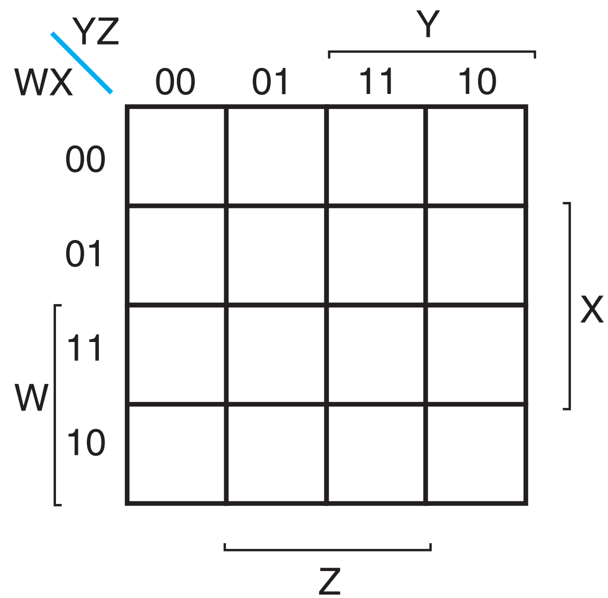


Fig. 2-16 $F(X,Y,Z) = \Sigma m(1,2,3,5,7)$

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6
m_{12}	m_{13}	m_{15}	m_{14}
m_8	m_9	m_{11}	m_{10}

(a)



(b)

Fig. 2-17 Four-Variable Map

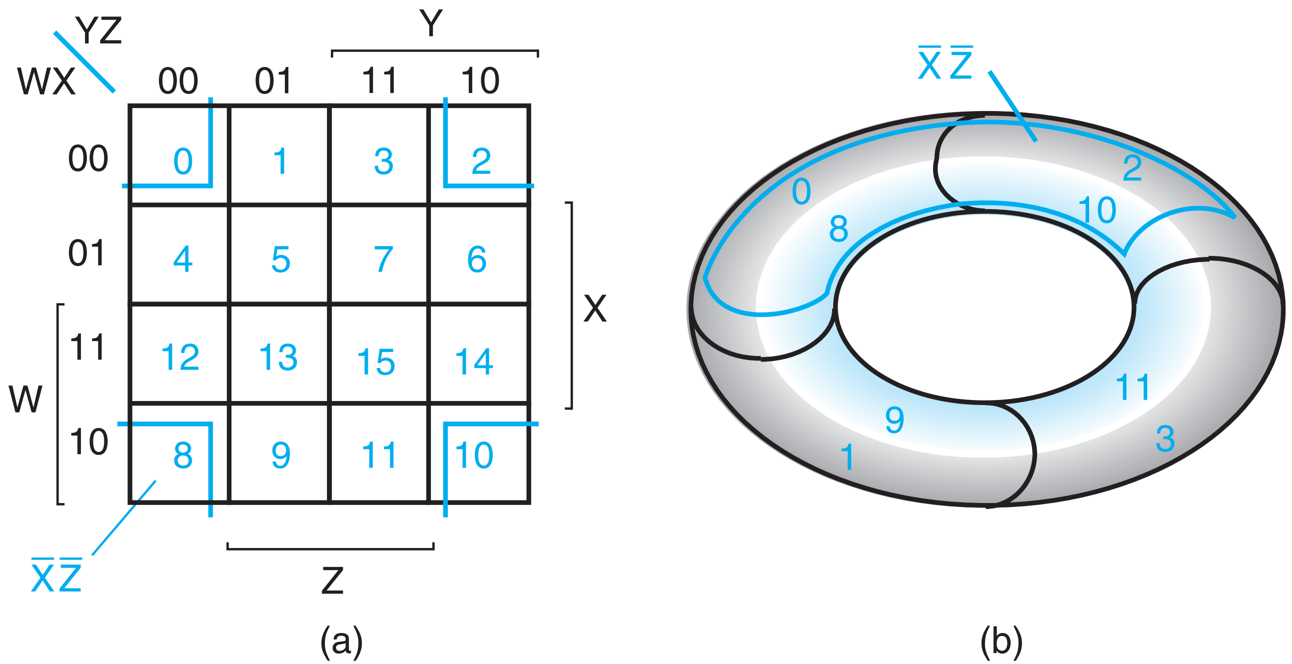


Fig. 2-18 Four-Variable Map: Flat and on a Torus to Show Adjacencies

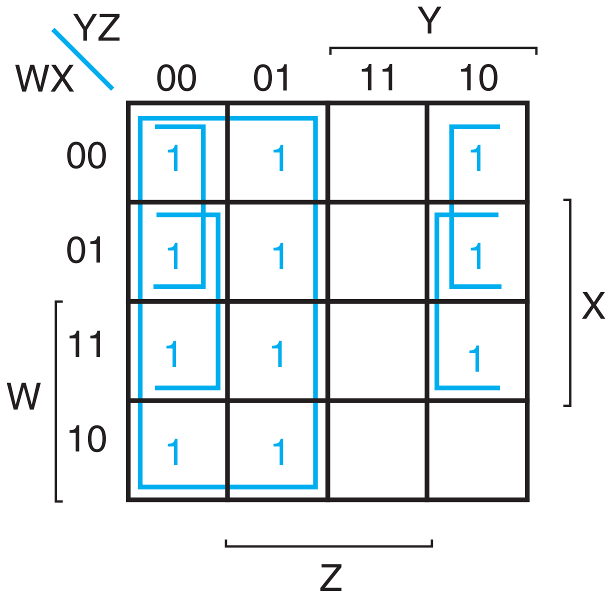


Fig. 2-19 Map for Example 2-5: $F = \overline{Y} + \overline{W}\overline{Z} + X\overline{Z}$

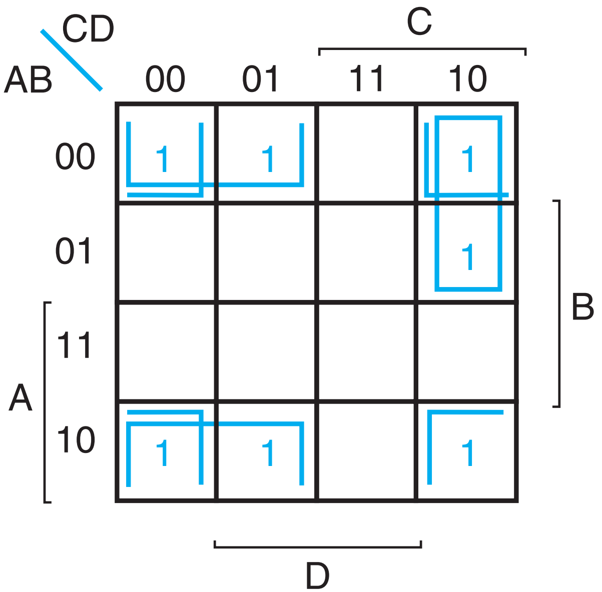


Fig. 2-20 Map for Example 2-6: $F = \overline{B}\,\overline{D} + \overline{B}\,\overline{C} + \overline{A}C\overline{D}$

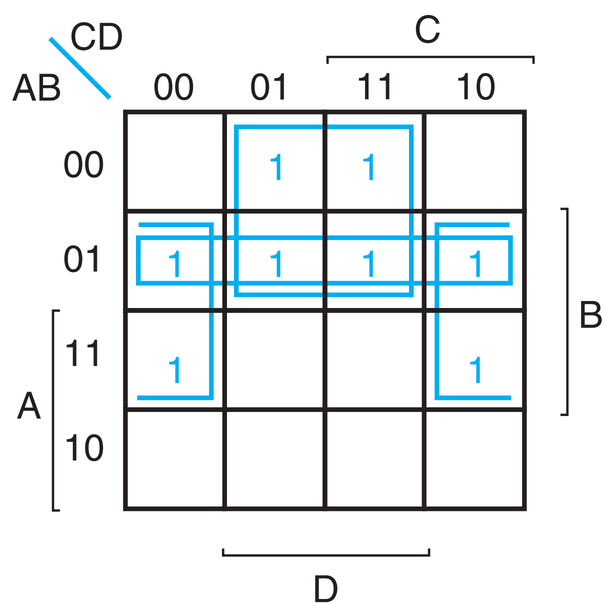
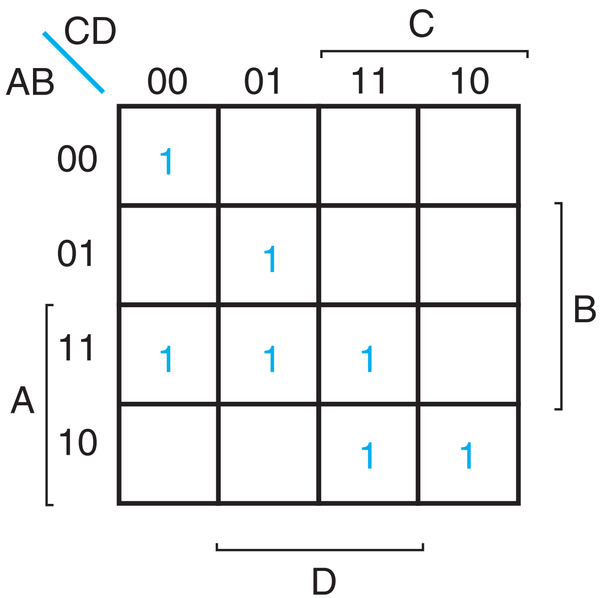
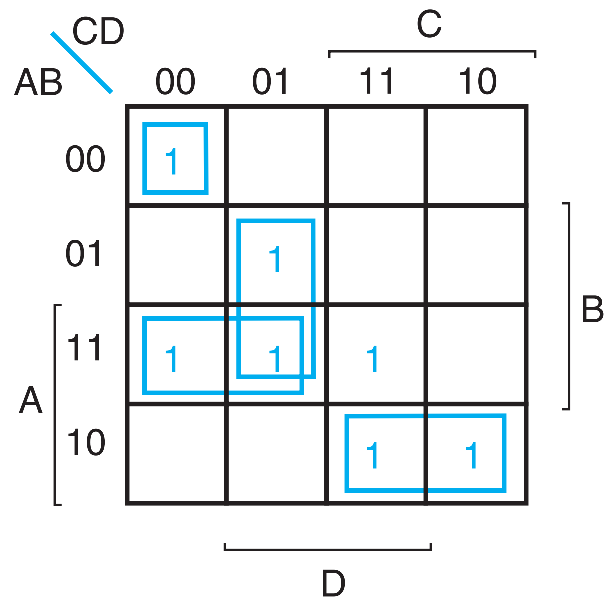


Fig. 2-21 Prime Implicants for Example 2-7: $\bar{A}D$, $B\bar{D}$, and $\bar{A}B$



(a) Plotting the minterms



(b) Essential prime implicants

Fig. 2-22 Simplification with Prime Implicants in Example 2-8

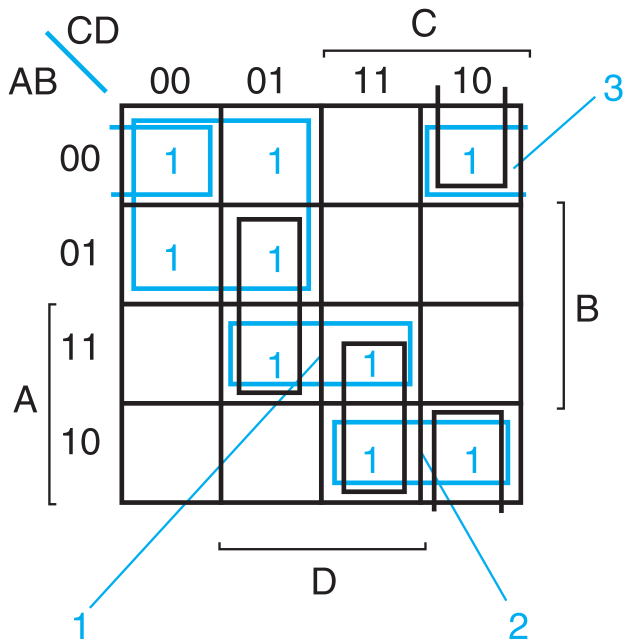


Fig. 2-23 Map for Example 2-9

M. Morris Mano & Charles R. Kime
LOGIC AND COMPUTER DESIGN FUNDAMENTALS, 2e, Updated.

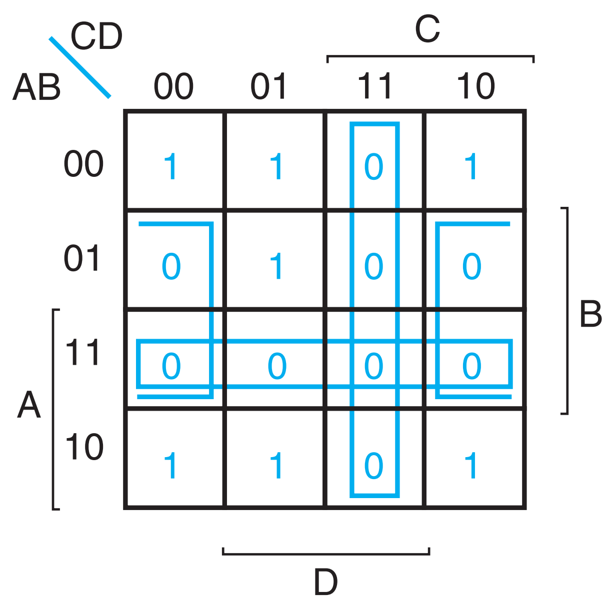
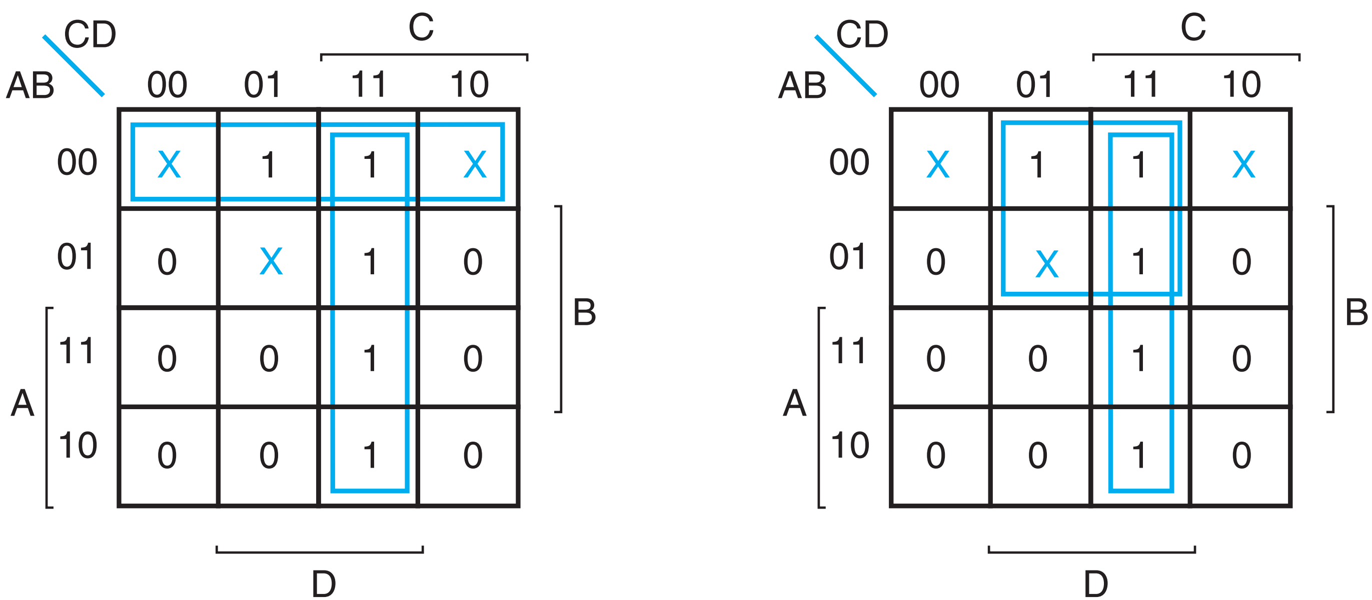


Fig. 2-24 Map for Example 2-10: $F = (\overline{A} + \overline{B})\,(\overline{C} + \overline{D})\,(\overline{B} + D)$



(a) $F = CD + \overline{A}\,\overline{B}$

(b) $F = CD + \overline{A}D$

Fig. 2-25 Example with Don't-Care Conditions

Graphics Symbols

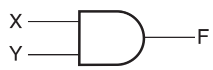
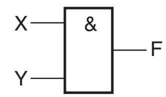
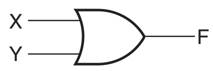
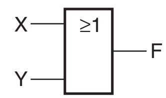
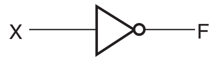
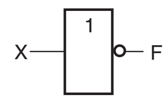
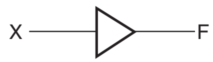
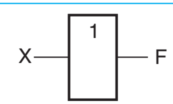
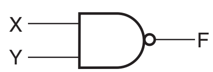
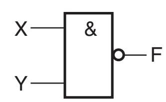
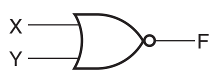
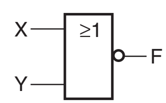
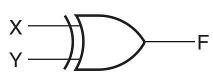
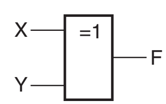
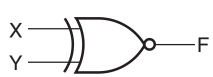
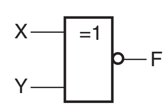
Name	Distinctive shape	Rectangular shape	Algebraic equation	Truth table
AND			$F = XY$	X Y F 0 0 0 0 1 0 1 0 0 1 1 1
OR			$F = X + Y$	X Y F 0 0 0 0 1 1 1 0 1 1 1 1
NOT (inverter)			$F = \overline{X}$	X F 0 1 1 0
Buffer			$F = X$	X F 0 0 1 1
NAND			$F = \overline{X \cdot Y}$	X Y F 0 0 1 0 1 1 1 0 1 1 1 0
NOR			$F = \overline{X + Y}$	X Y F 0 0 1 0 1 0 1 0 0 1 1 0
Exclusive–OR (XOR)			$F = X\overline{Y} + \overline{X}Y$ $= X \oplus Y$	X Y F 0 0 0 0 1 1 1 0 1 1 1 0
Exclusive–NOR (XNOR)			$F = XY + \overline{X}\overline{Y}$ $= \overline{X \oplus Y}$	X Y F 0 0 1 0 1 0 1 0 0 1 1 1

Fig. 2-26 Digital Logic Gates

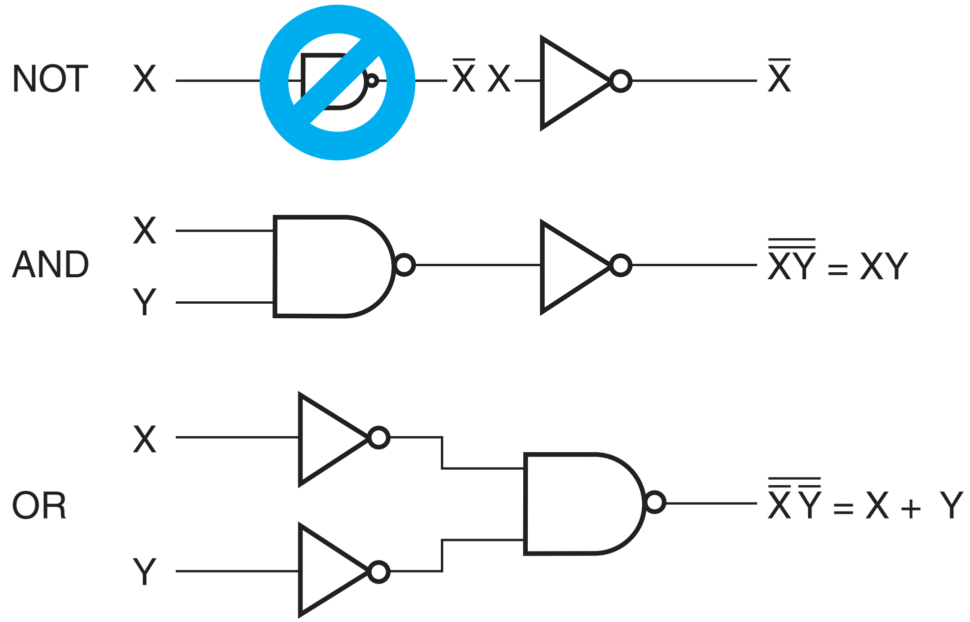


Fig. 2-27 Logical Operations with NAND Gates

M. Morris Mano & Charles R. Kime
LOGIC AND COMPUTER DESIGN FUNDAMENTALS, 2e, Updated.

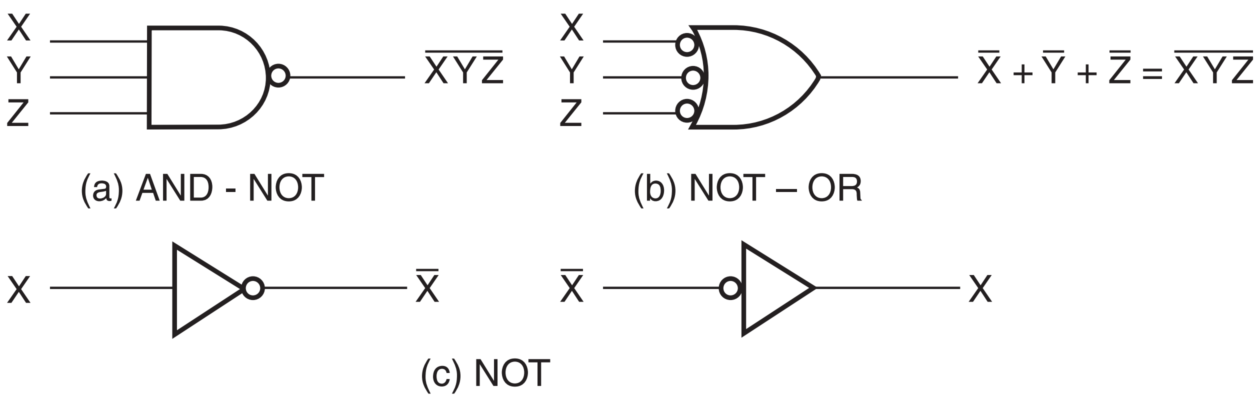


Fig. 2-28 Alternative Graphics Symbols for NAND and NOT Gates

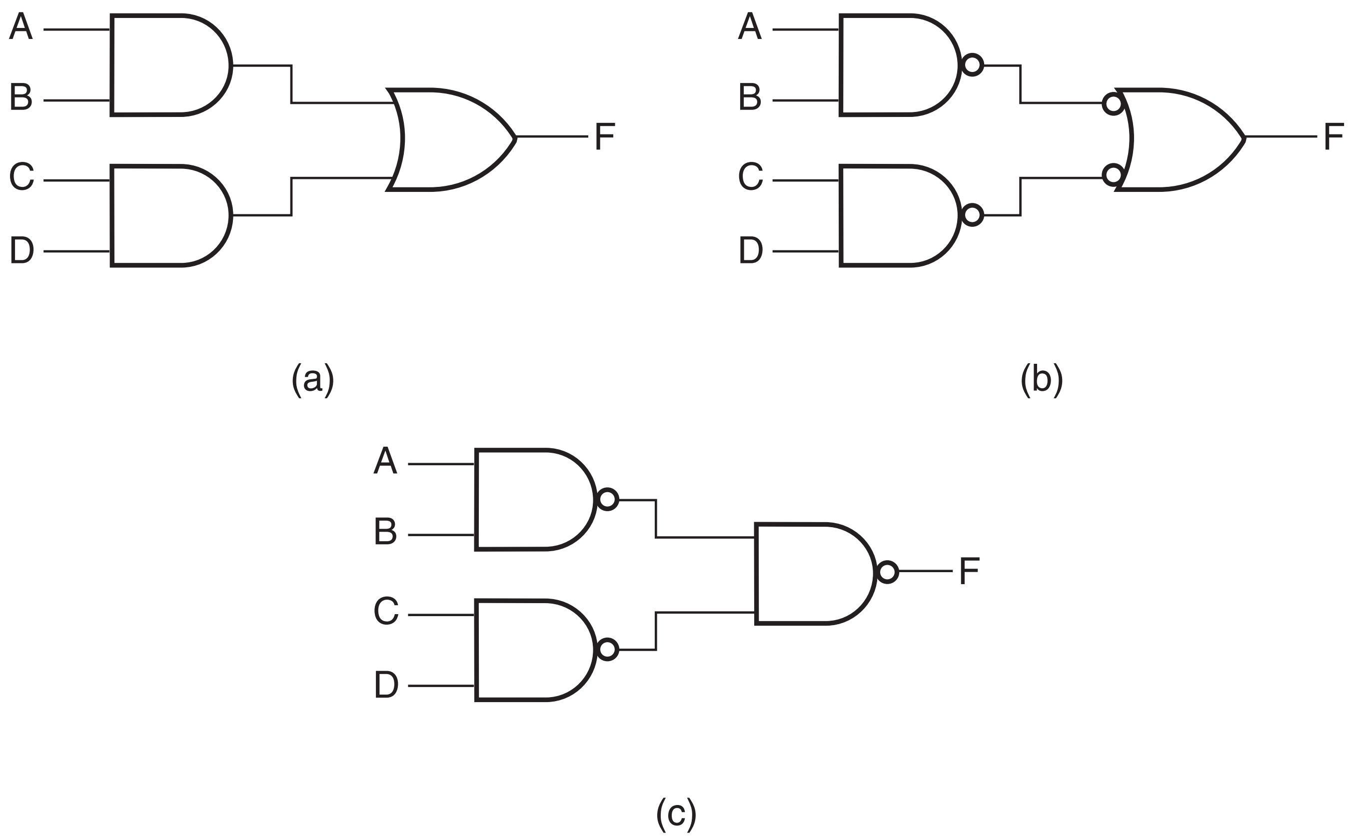


Fig. 2-29 Three Ways to Implement F = AB + CD

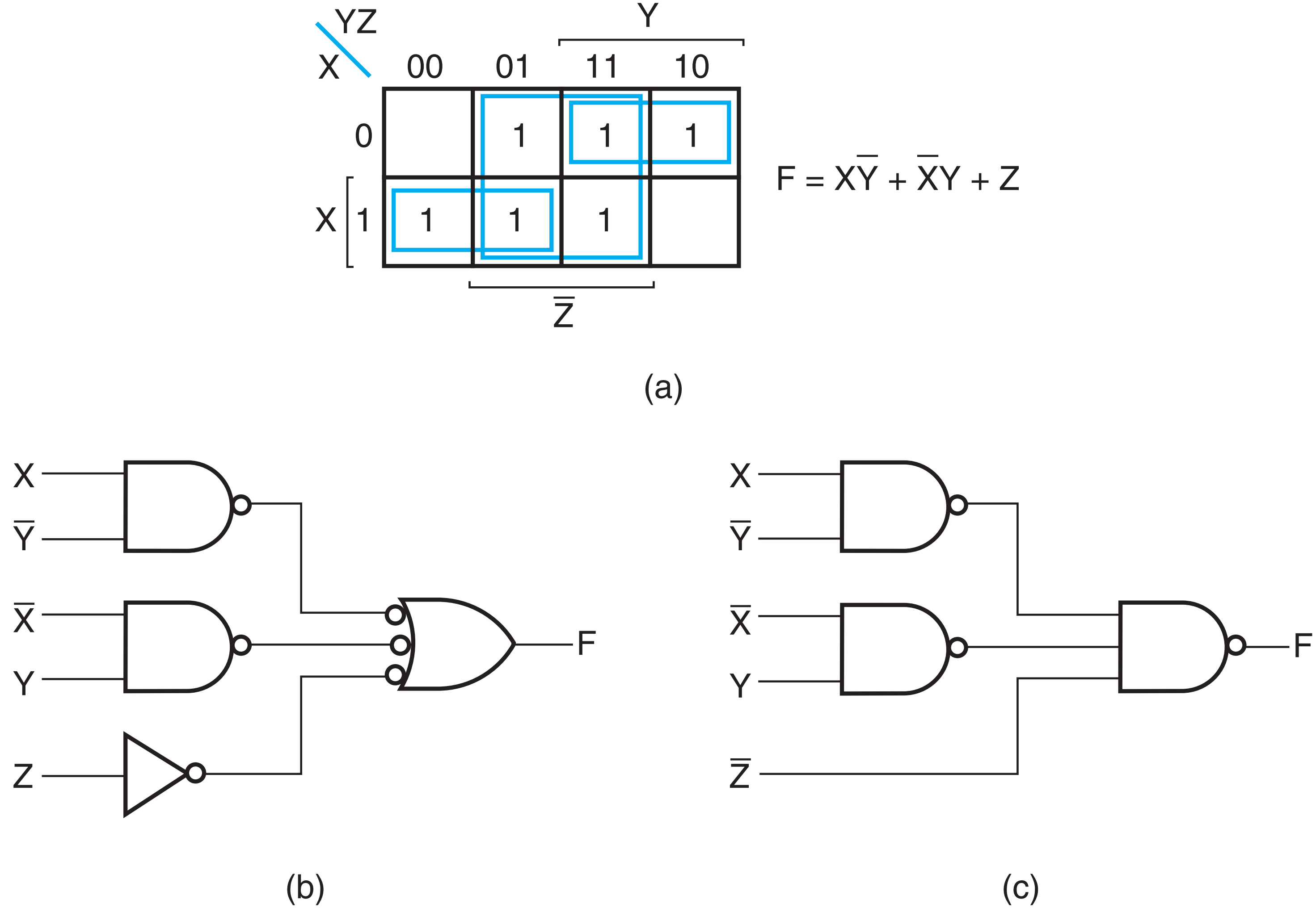
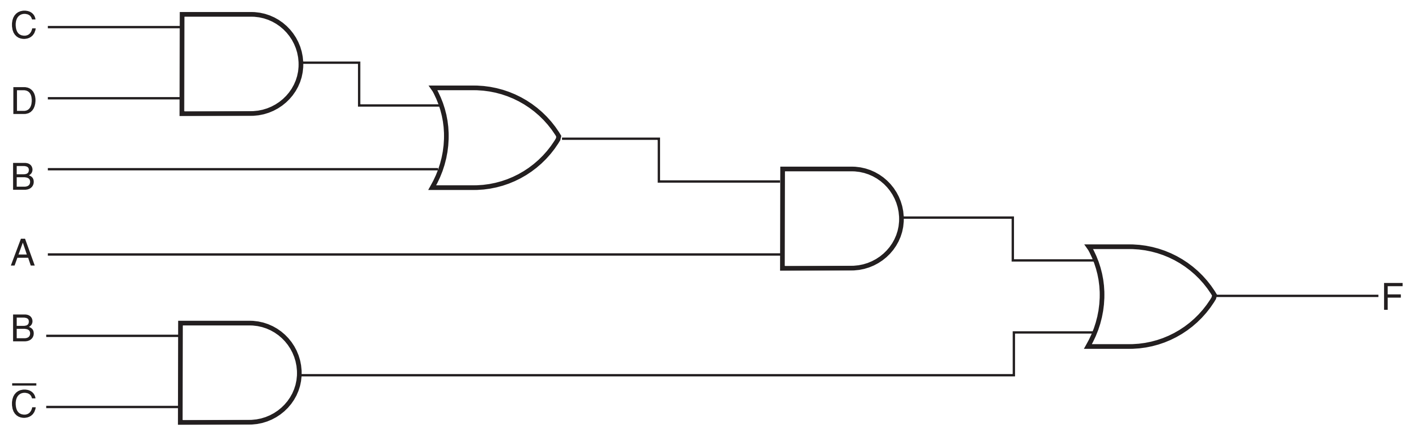
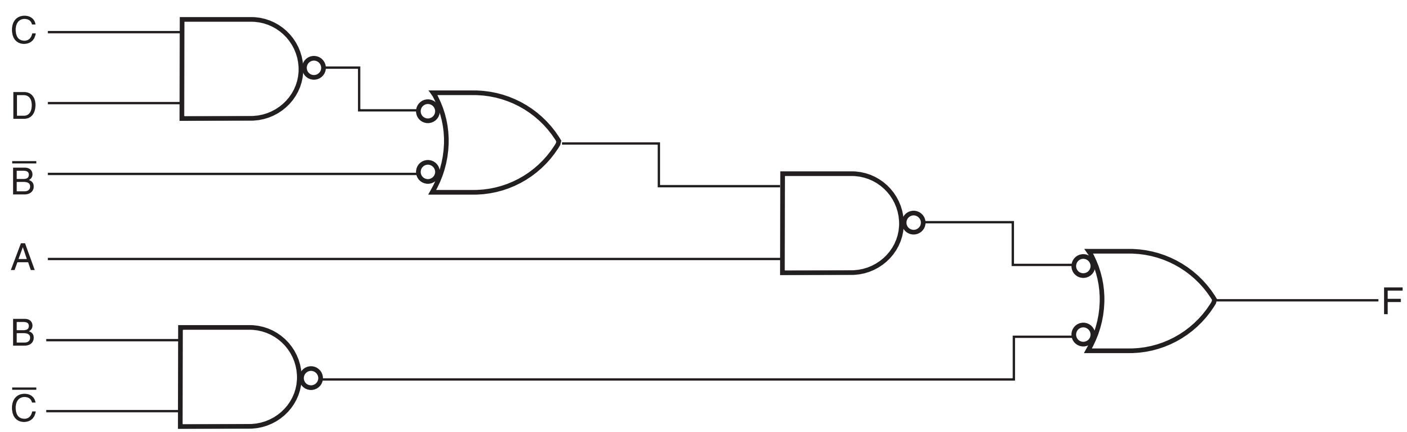


Fig. 2-30 Solution to Example 2-12



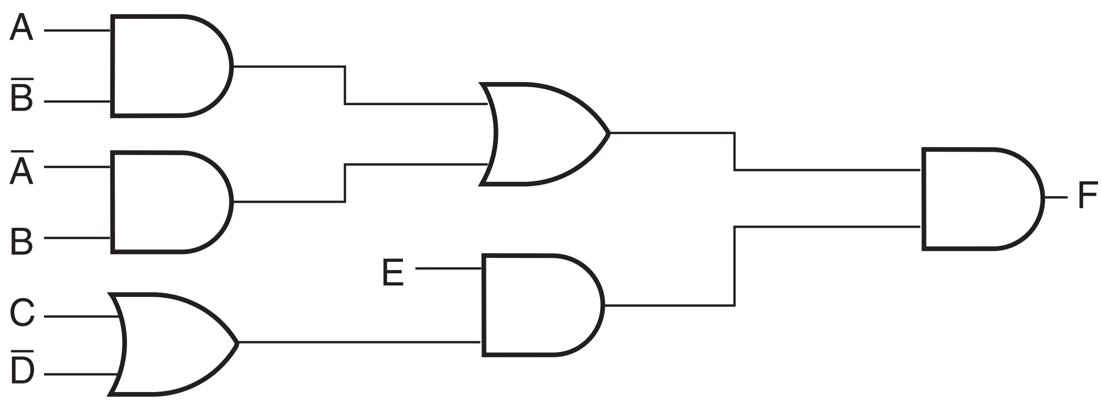
(a) AND – OR gates



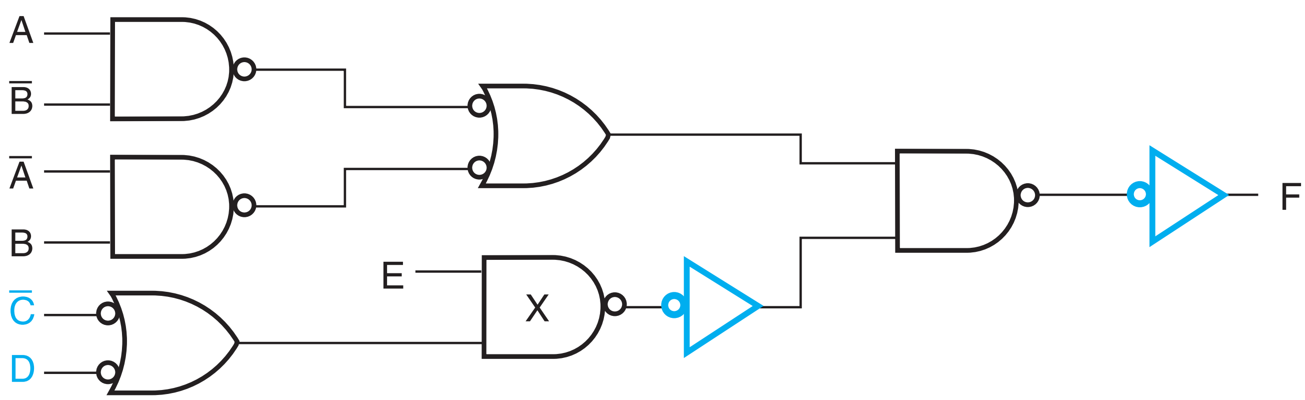
(b) NAND gates

Fig. 2-31 Implementing $F = A\,(CD + B) + B\overline{C}$

M. Morris Mano & Charles R. Kime
LOGIC AND COMPUTER DESIGN FUNDAMENTALS, 2e, Updated.



(a) AND – OR gates



(b) NAND gates

Fig. 2-32 Implementing $F = (A\overline{B} + \overline{A}B)E(C + \overline{D})$

M. Morris Mano & Charles R. Kime
LOGIC AND COMPUTER DESIGN FUNDAMENTALS, 2e, Updated.

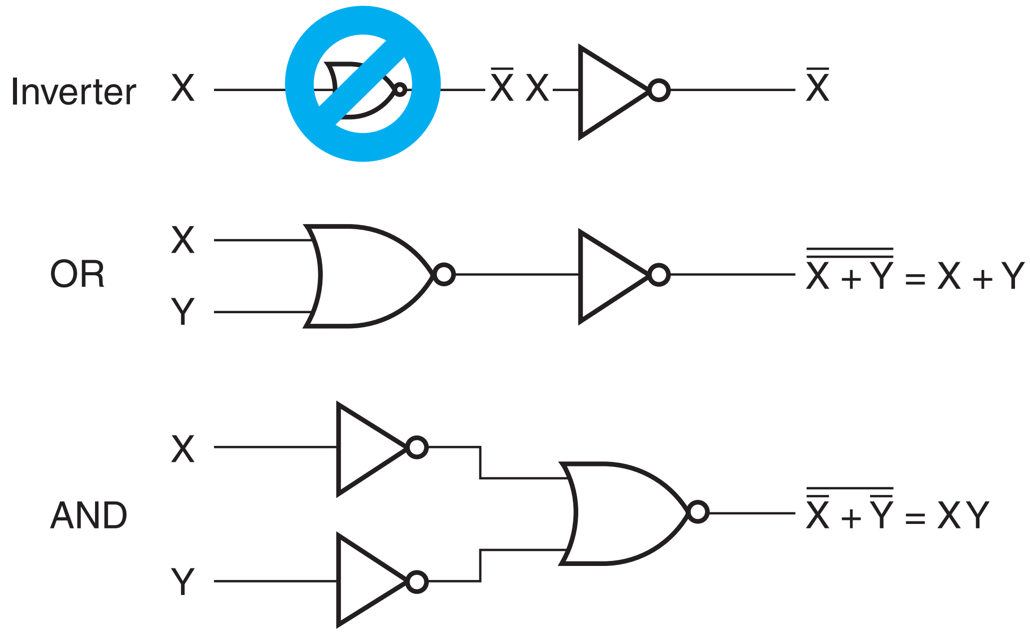
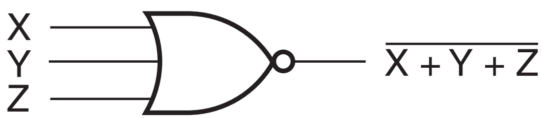
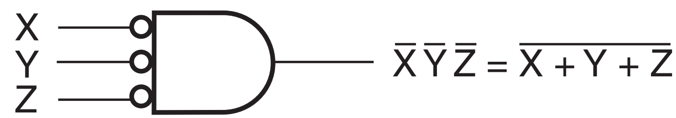


Fig. 2-33 Logic Operations with NOR Gates



(a) OR – NOT



(b) NOT – AND

Fig. 2-34 Two Graphic Symbols for NOR Gate

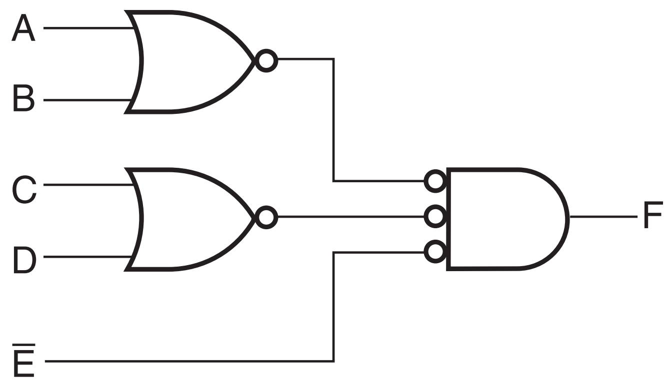


Fig. 2-35 Implementing $F = (A + B)\ (C + D)\ E$ with NOR Gates

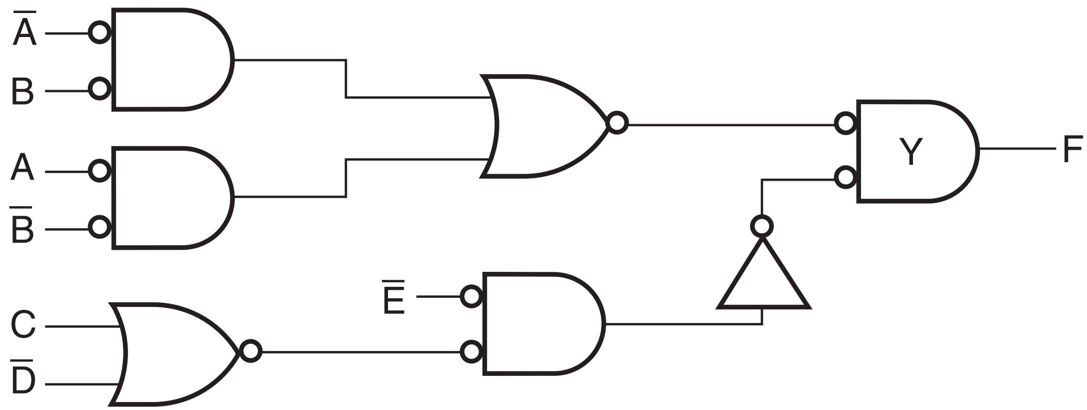


Fig. 2-36 Implementing $F = (A\overline{B} + \overline{A}B)\ E\ (C + \overline{D})$ with NOR Gates

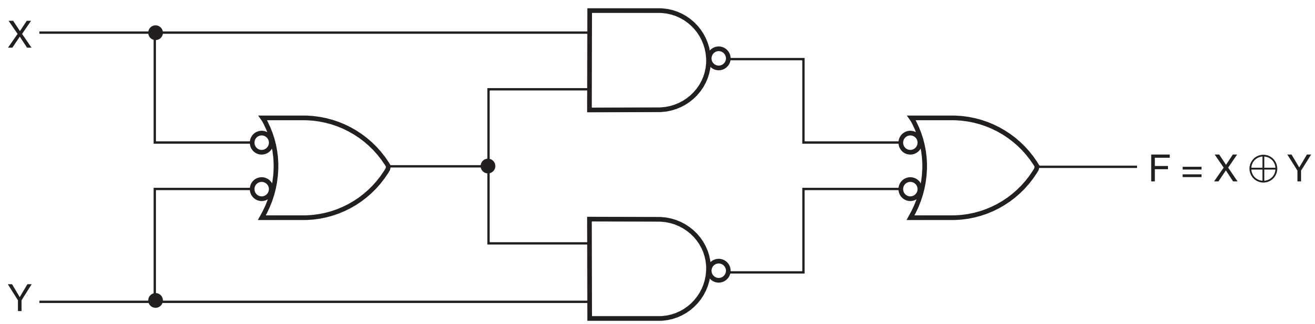
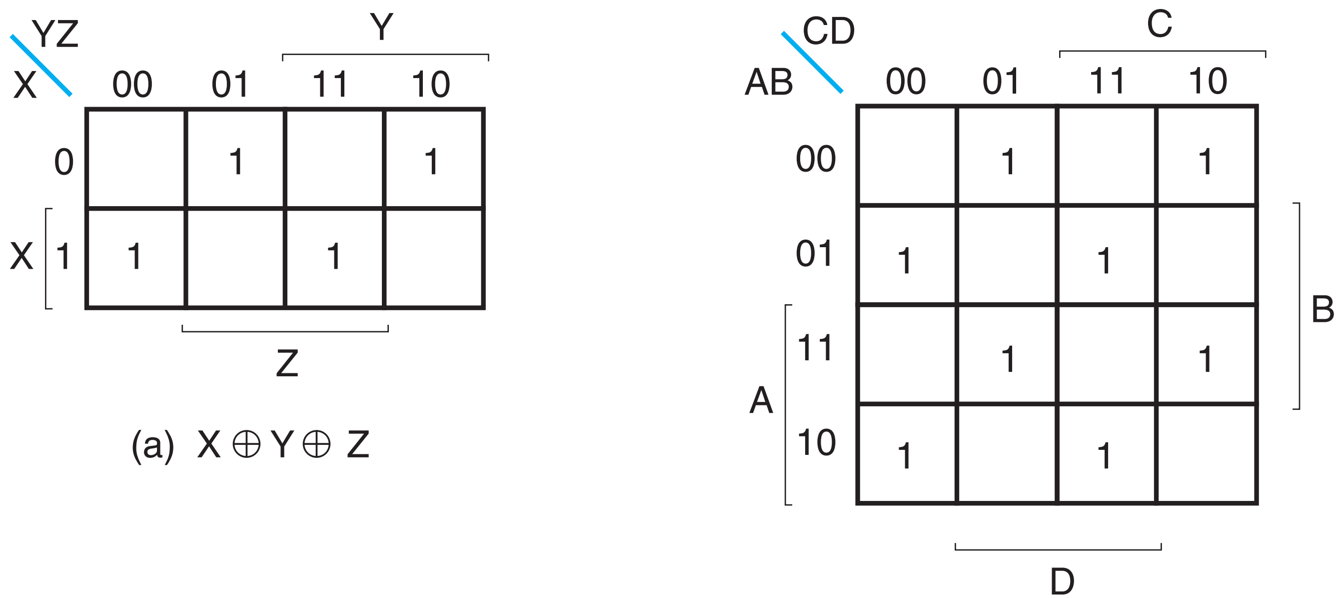


Fig. 2-37 Exclusive-OR Constructed with NAND Gates

M. Morris Mano & Charles R. Kime
LOGIC AND COMPUTER DESIGN FUNDAMENTALS, 2e, Updated.



(a) $X \oplus Y \oplus Z$

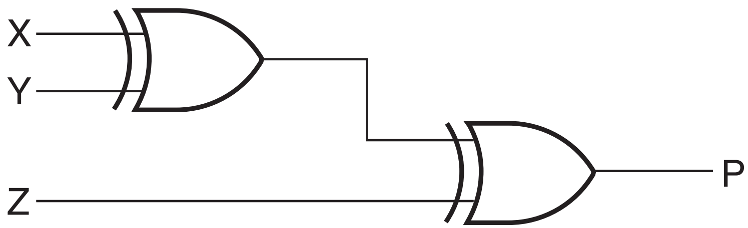
(b) $A \oplus B \oplus C \oplus D$

Fig. 2-38 Maps for Multiple-Variable Odd Functions

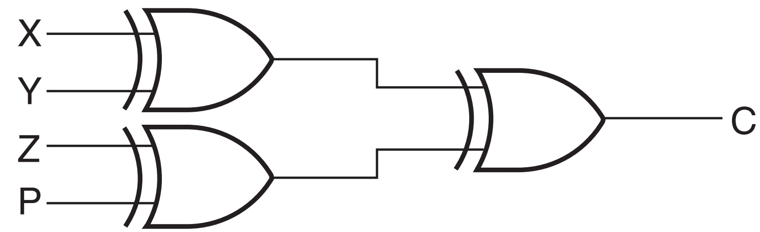
TABLE 2-9
Truth Table for an Even Parity Generator

Three-Bit Message			Parity Bit
X	**Y**	**Z**	**P**
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

Table 2-9 Truth Table for an Even Parity Generator



(a) $P = X \oplus Y \oplus Z$



(b) $C = X \oplus Y \oplus Z \oplus P$

Fig. 2-39 Multiple-Input Odd Functions

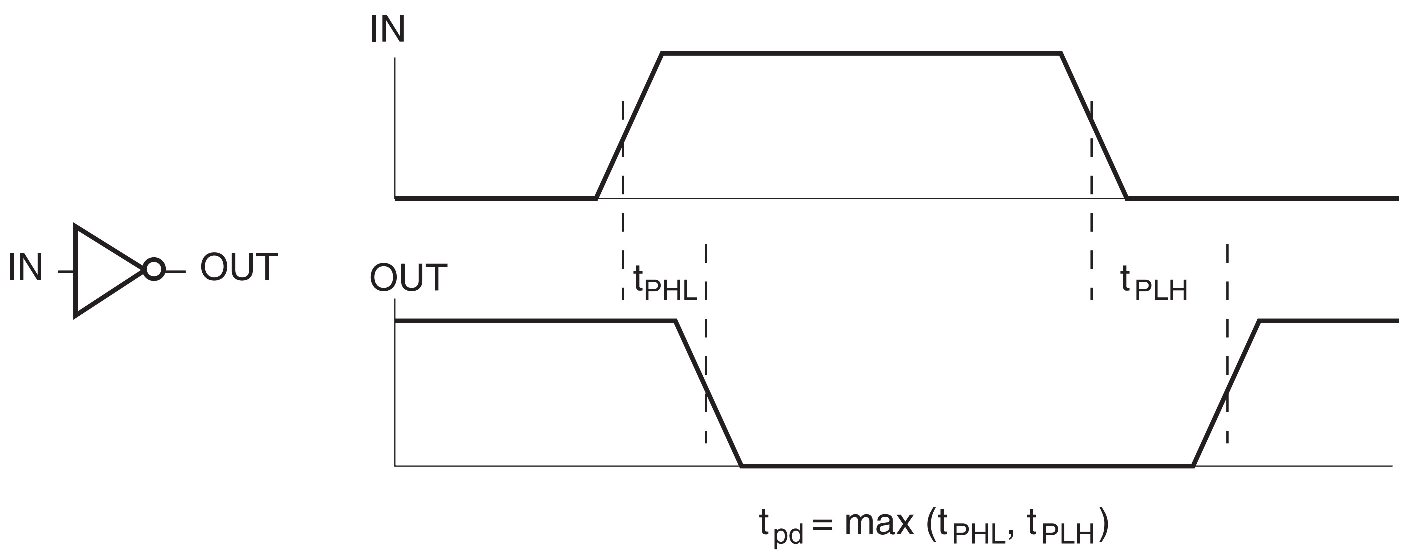


Fig. 2-40 Propagation Delay for an Inverter

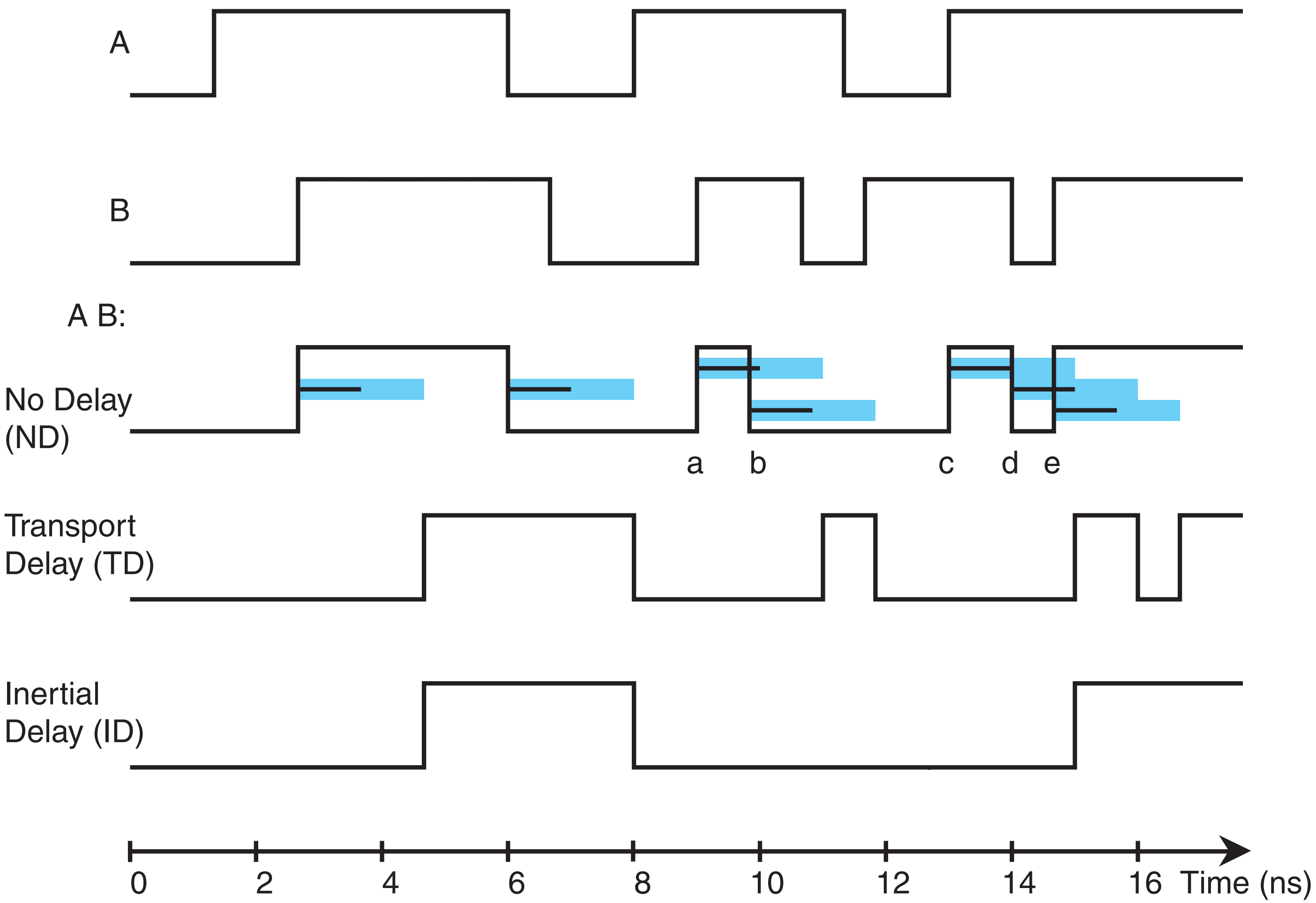


Fig. 2-41 Examples of Behavior of Transport and Inertial Delays

Signal value	Logic value
H	1
L	0

(a) Positive logic

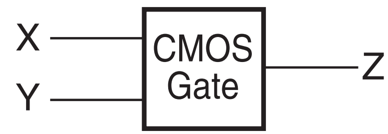
Signal value	Logic value
H	0
L	1

(b) Negative logic

Fig. 2-42 Signal Assignment and Logic Polarity

X	Y	Z
L	L	L
L	H	L
H	L	L
H	H	H

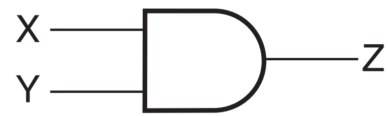
(a) Truth table with H and L



(b) Gate block diagram

X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

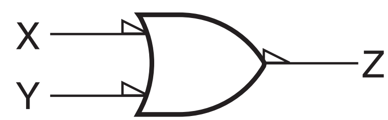
(c) Truth table for positive logic



(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

Fig. 2-43 Demonstration of Positive and Negative Logic

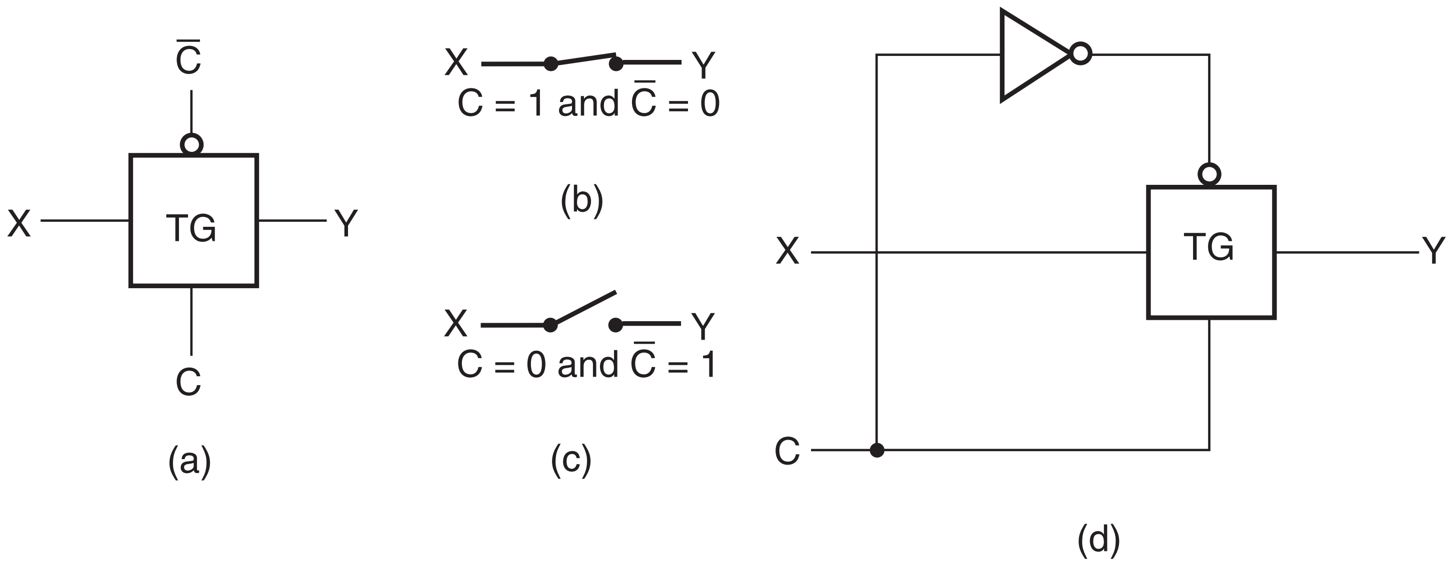
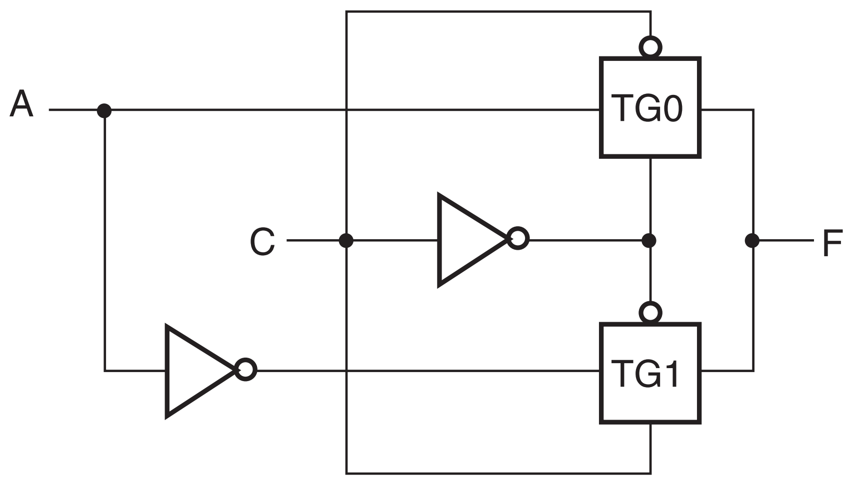


Fig. 2-44 Transmission Gate (TG)

M. Morris Mano & Charles R. Kime
LOGIC AND COMPUTER DESIGN FUNDAMENTALS, 2e, Updated.



(a)

A	C	TG1	TG0	F
0	0	No path	Path	0
0	1	Path	No path	1
1	0	No path	Path	1
1	1	Path	No path	0

(b)

Fig. 2-45 Transmission Gate Exclusive-OR