Fig. 3-1 Two-variable Map
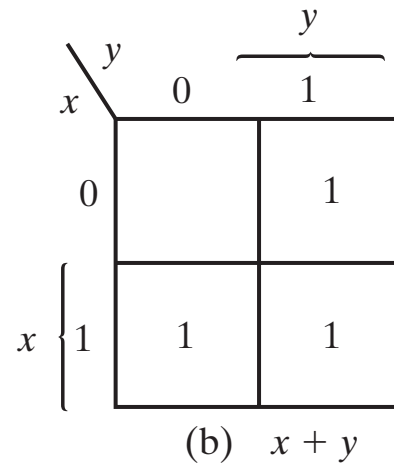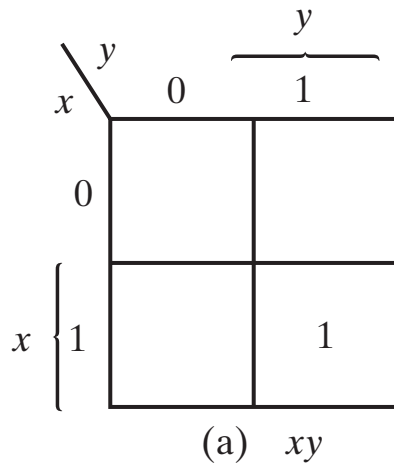
M. Morris Mano
DIGITAL DESIGN, 3e.

Fig. 3-2 Representation of Functions in the Map

Fig. 3-3 Three-variable Map

	m_0	m_1	m_3	m_2
	m_4	m_5	m_7	m_6

(a)

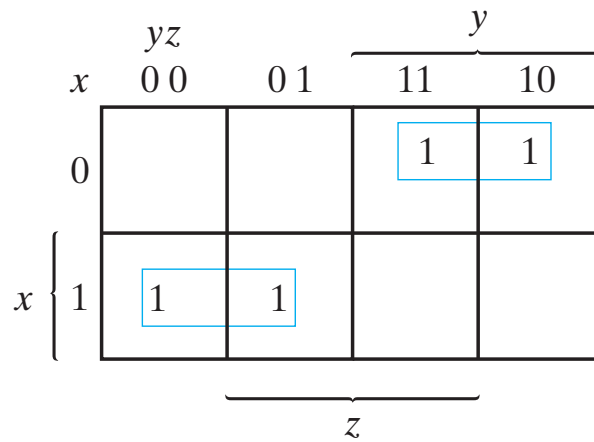|	0 0	0 1	11	10
0	$x'y'z'$	$x'y'z$	$x'yz$	$x'yz'$
1	$xy'z'$	$xy'z$	xyz	xyz'

(b)

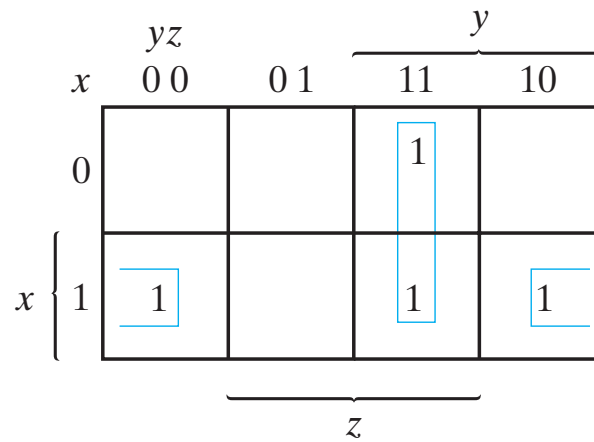Fig. 3-4 Map for Example 3-1; $F(x, y, z) = \Sigma(2, 3, 4, 5) = x'y + xy'$

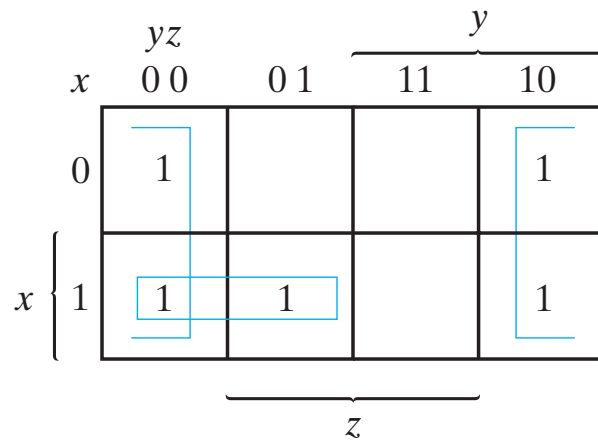Fig. 3-5 Map for Example 3-2; $F(x, y, z) = \Sigma(3, 4, 6, 7) = yz + xz'$

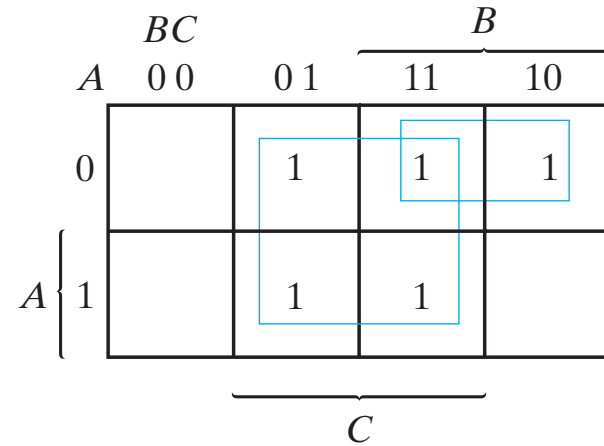Fig. 3-6 Map for Example 3-3; $F(x, y, z) = \Sigma(0, 2, 4, 5, 6) = z' + xy'$

Fig. 3-7 Map for Example 3-4; $A'C + A'B + AB'C + BC = C + A'B$

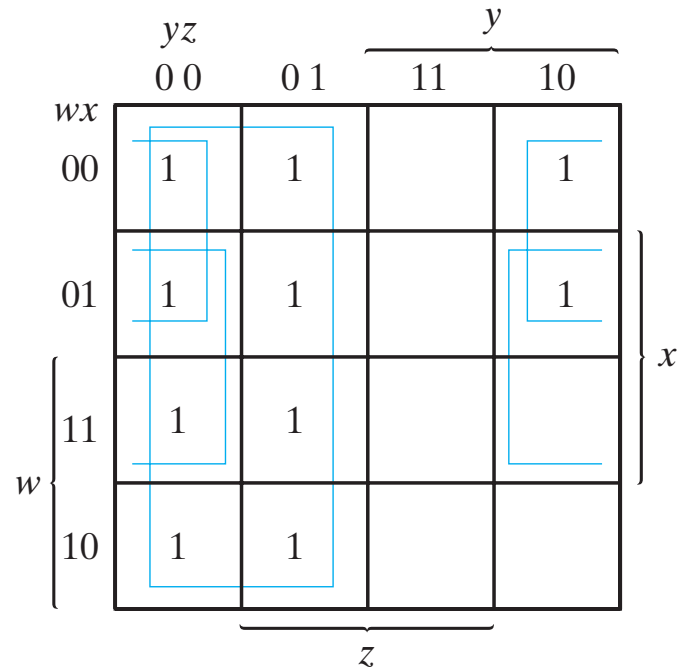|	yz		y	
wx	0 0	0 1	11	10
00	$w'x'y'z'$	$w'x'y'z$	$w'x'yz$	$w'x'yz'$
01	$w'xy'z'$	$w'xy'z$	$w'xyz$	$w'xyz'$
11	$wxy'z'$	$wxy'z$	$wxyz$	$wxyz'$
10	$wx'y'z'$	$wx'y'z$	$wx'yz$	$wx'yz'$

(a)

(b)

Fig. 3-8 Four-variable Map

Fig. 3-9 Map for Example 3-5; $F(w, x, y, z)$
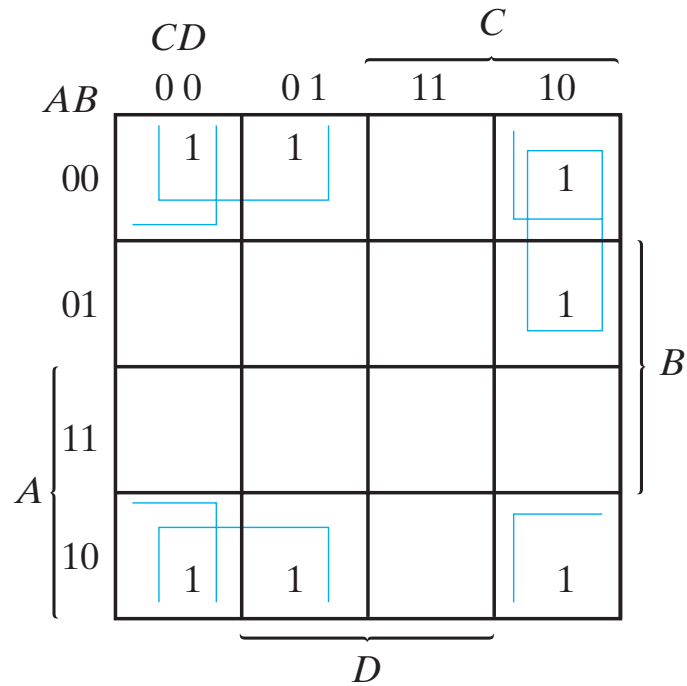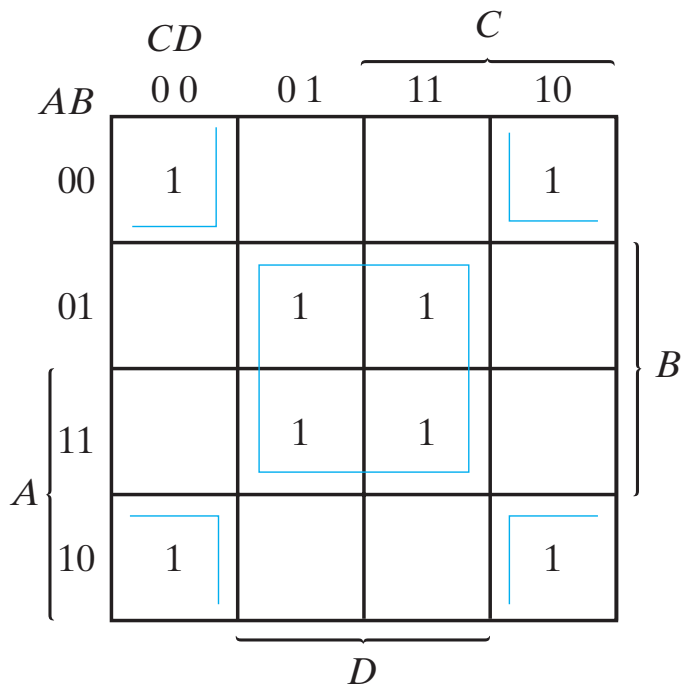$= \Sigma (0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14) = y' + w'z' + xz'$
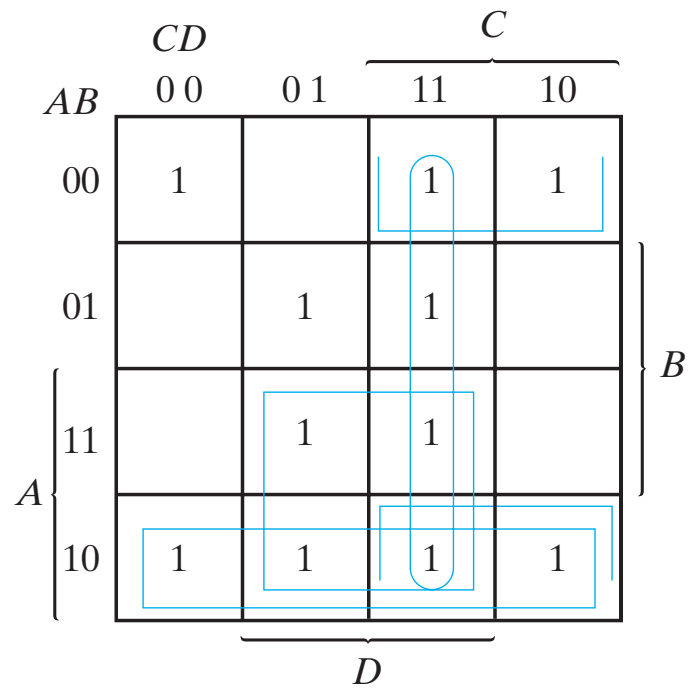
Fig.3-10 Map for Example 3-6; $A'B'C + B'CD' + A'BCD'$
$+ AB'C' = B'D' + B'C' + A'CD'$

(a) Essential prime implicants
BD and B′D′

(b) Prime implicants CD, B′C
AD, and AB′

Fig. 3-11 Simplification Using Prime Implicants

A = 0

BC \ DE	0 0	0 1	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

A = 1

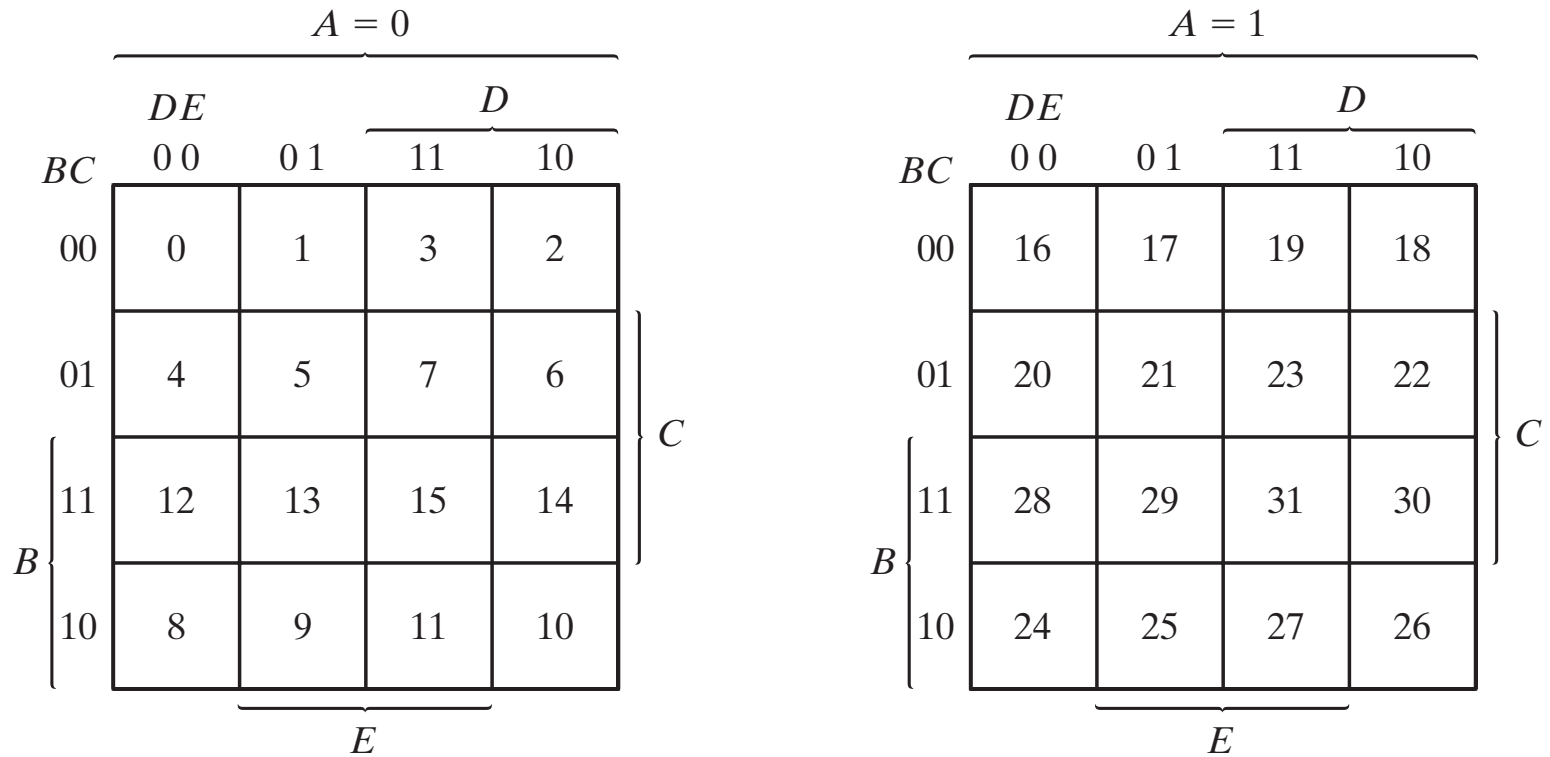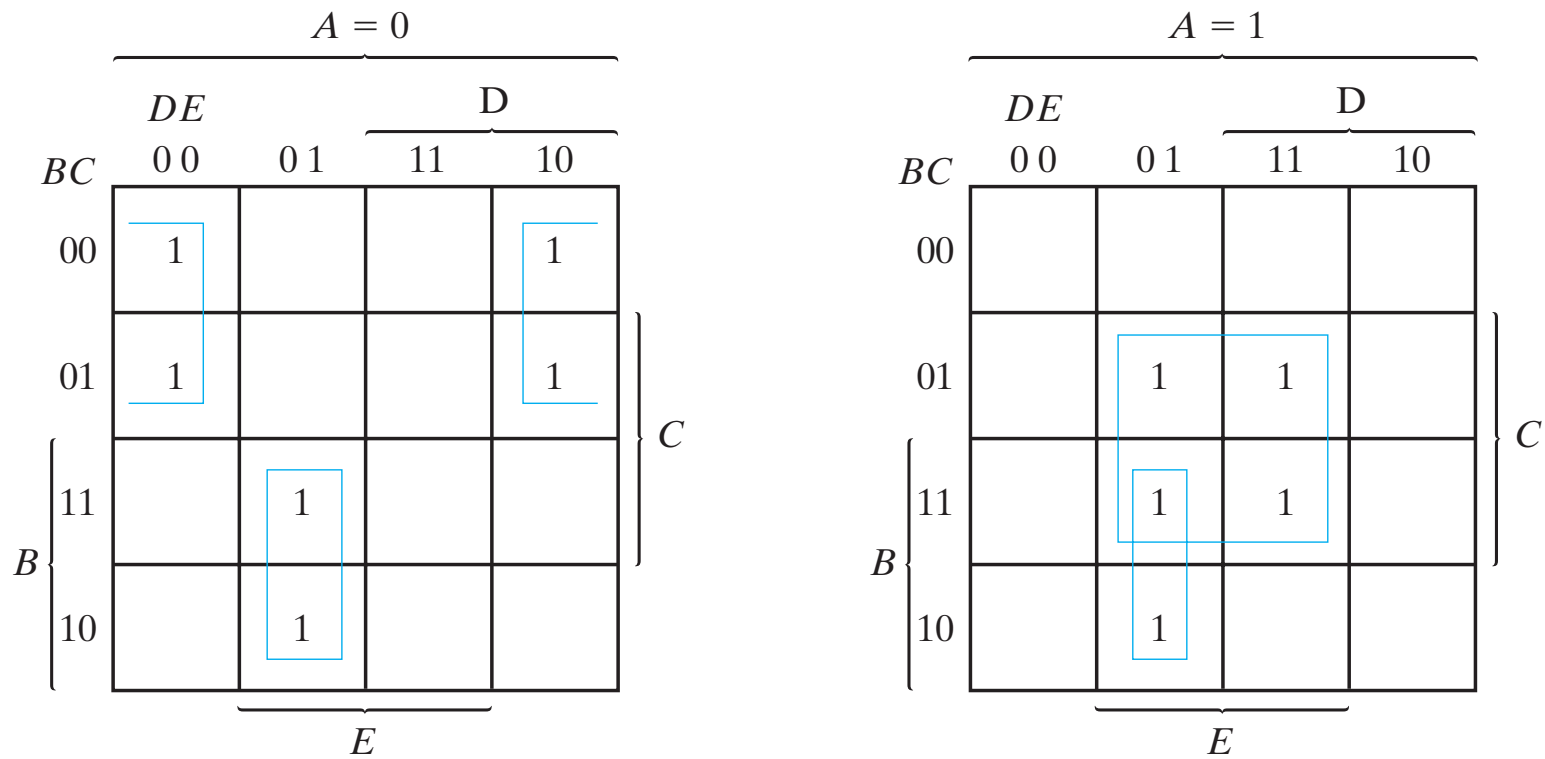| BC \ DE	0 0	0 1	11	10
00	16	17	19	18
01	20	21	23	22
11	28	29	31	30
10	24	25	27	26

Fig. 3-12 Five-variable Map

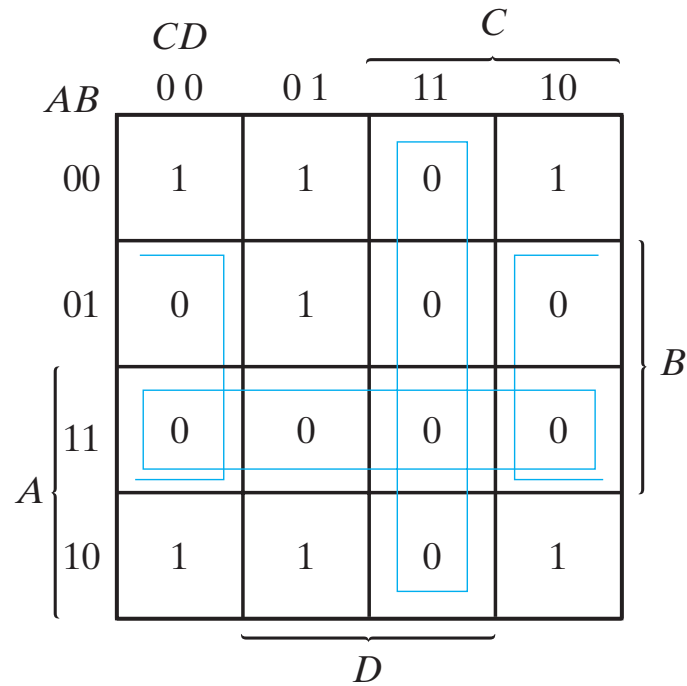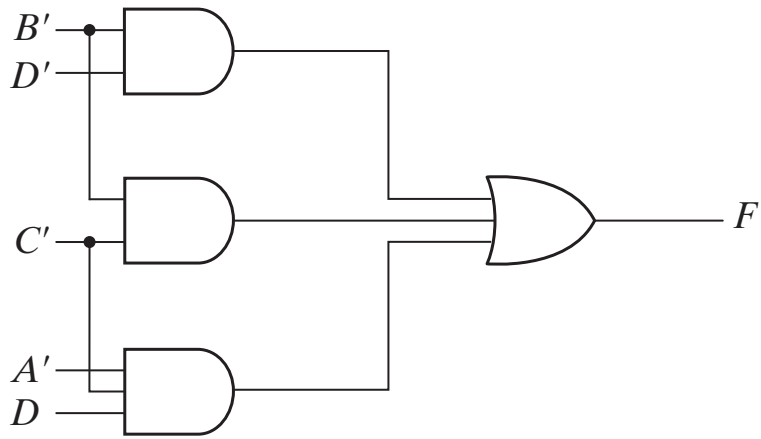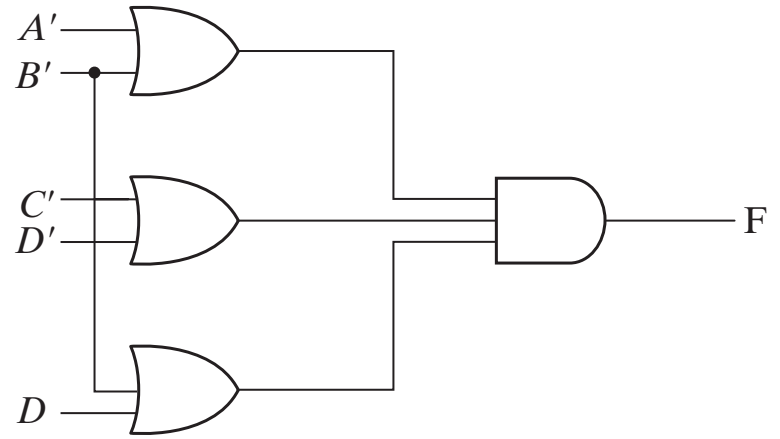Fig. 3-13 Map for Example 3-7; $F = A'B'E' + BD'E + ACE$

Fig. 3-14 Map for Example 3-8; $F(A, B, C, D) = \Sigma(0, 1, 2, 5, 8, 9, 10)$
$= B'D' + B'C' + A'C'D = (A' + B')(C' + D')(B' + D)$

(a) $F = B'D' + B'C' + A'C'D$

(b) $F = (A' + B')(C' + D')(B' + D)$

Fig. 3-15 Gate Implementation of the Function of Example 3-8

Fig. 3-16 Map for the Function of Table 3-2

Left map (a)

wx \ yz	00	01	11	10
00	X	1	1	X
01	0	X	1	0
11	0	0	1	0
10	0	0	1	0

(a) $F = yz + w'x'$

Right map (a)

wx \ yz	00	01	11	10
00	X	1	1	X
01	0	X	1	0
11	0	0	1	0
10	0	0	1	0

(a) $F = yz + w'z$

Fig. 3-17 Example with don't-care Conditions

Inverter x ———▷○———————————— x'

AND x y ———▷○————▷○— xy
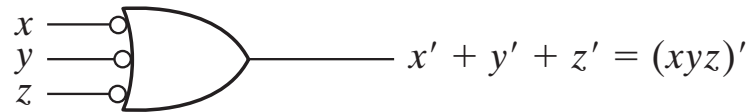
OR x ———▷○— y ———▷○— $(x'y')' = x + y$

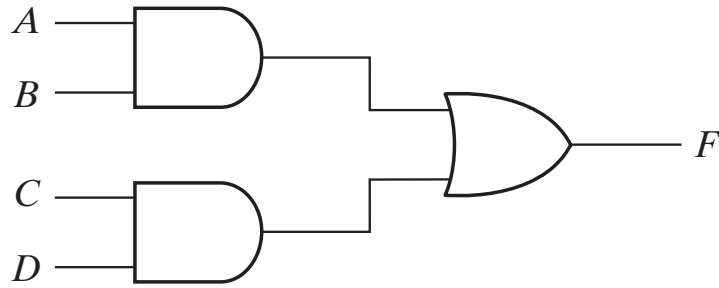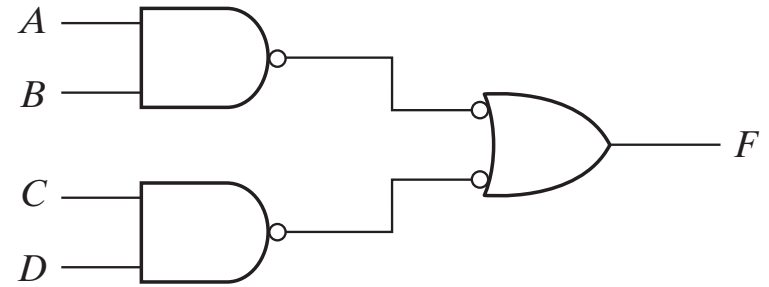Fig. 3-18 Logic Operations with NAND Gates

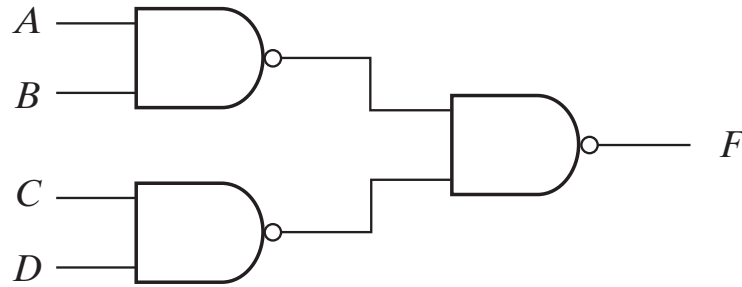(a) AND–invert   (b) Invert–OR

Fig. 3-19 Two Graphic Symbols for NAND Gate

(a)
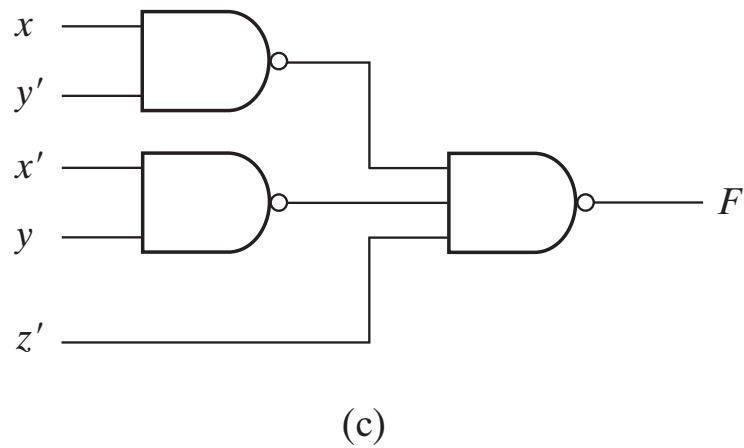
(b)

(c)

Fig. 3-20 Three Ways to Implement $F = AB + CD$

M. Morris Mano
DIGITAL DESIGN, 3e.
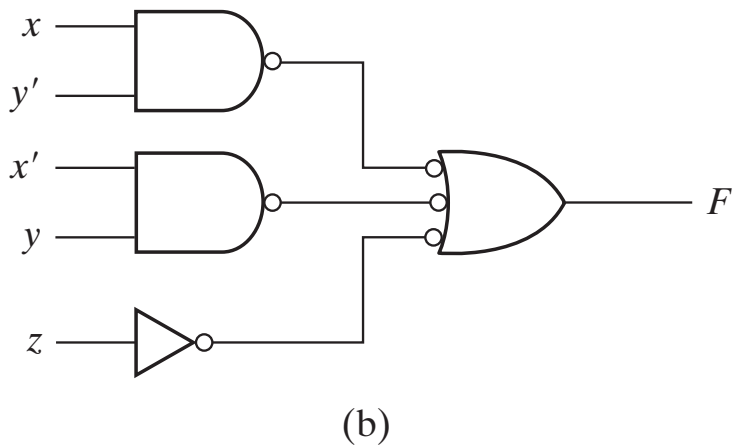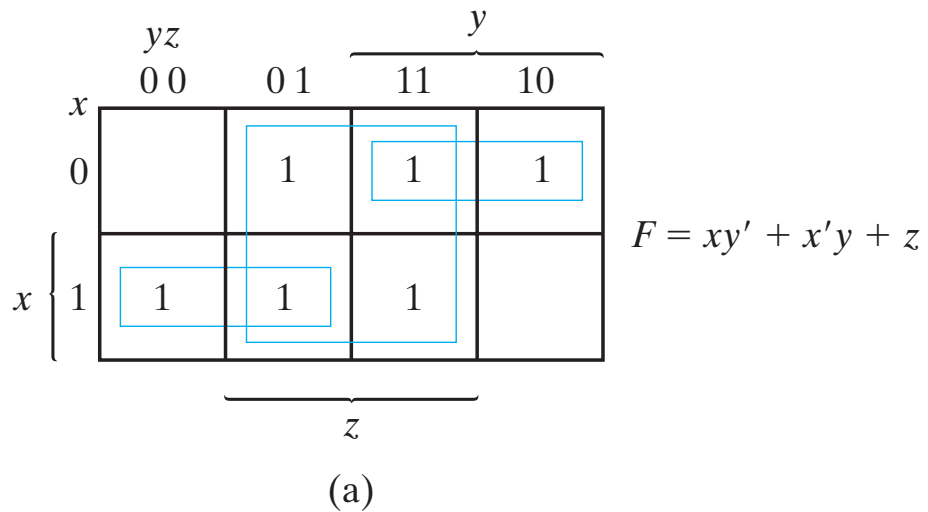
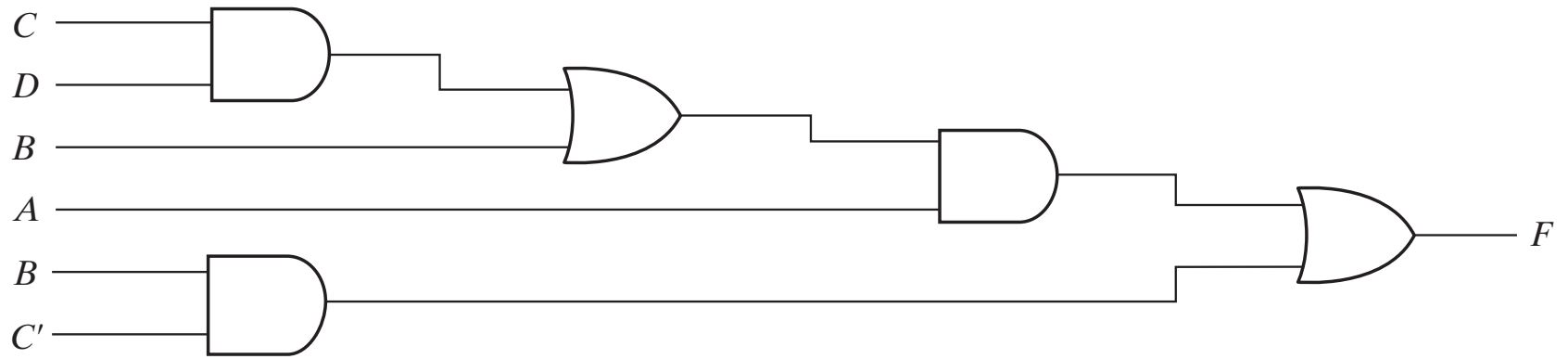$$F = xy' + x'y + z$$

(a)

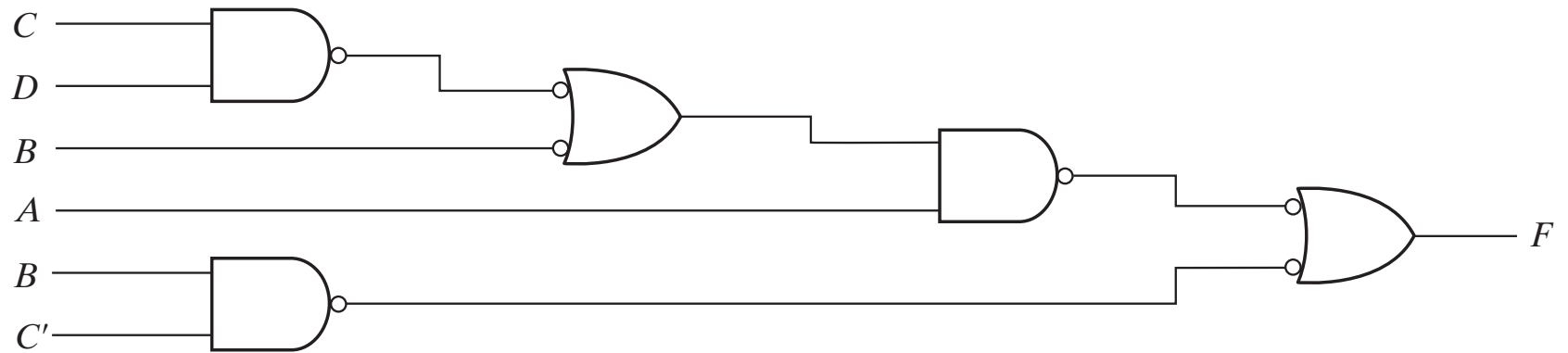(b)   (c)

Fig. 3-21 Solution to Example 3-10
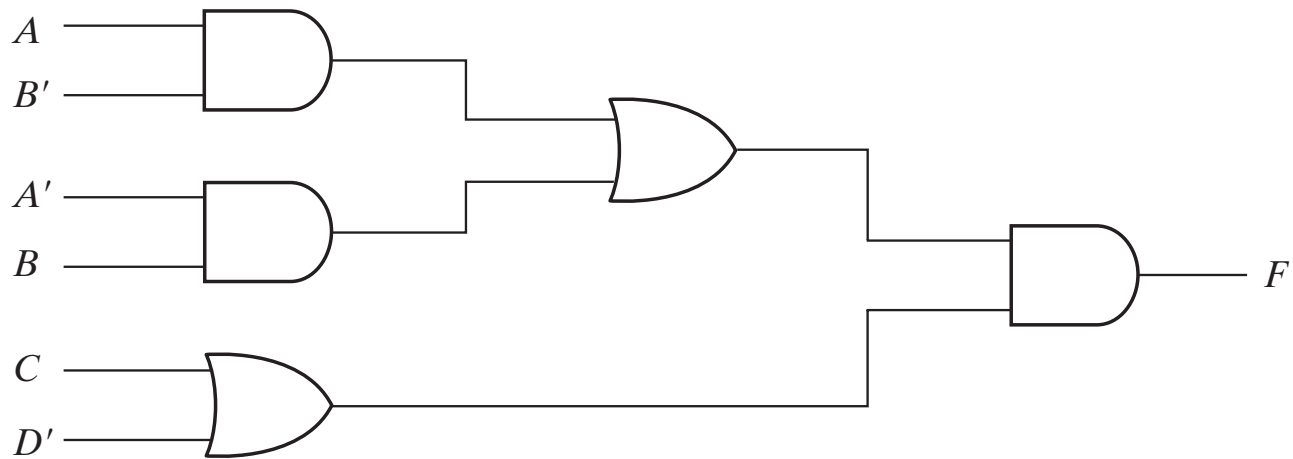
(a) AND-OR gates

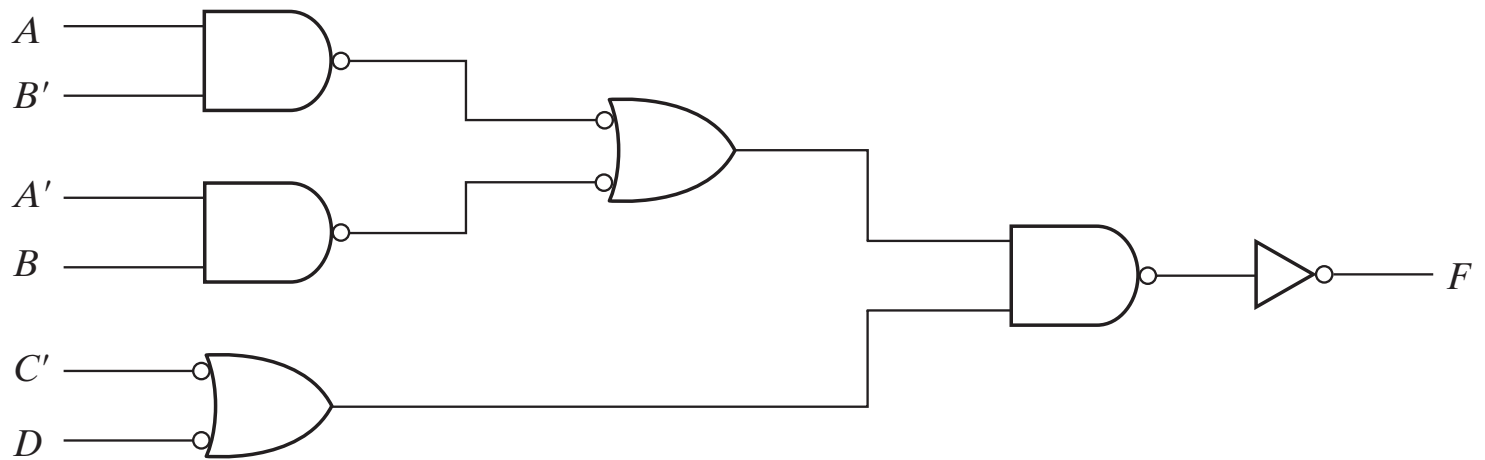(a) NAND gates

Fig. 3-22 Implementing $F = A(CD + B) + BC$

(a) AND-OR gates

(b) NAND gates

Fig. 3-23 Implementing $F = (AB'+A'B)(C+D')$
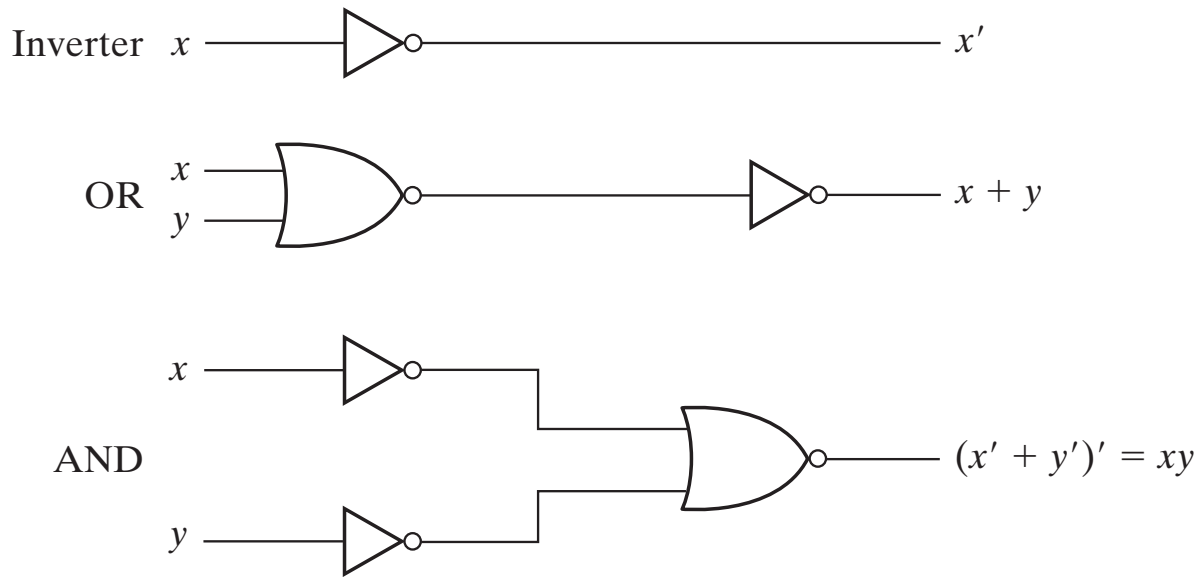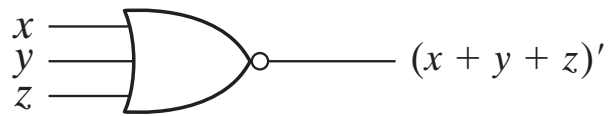
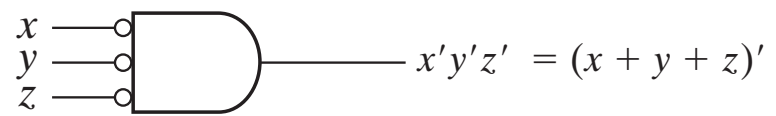Inverter x —▷o— x'

OR $\begin{matrix} x \\ y \end{matrix}$ —▷o— ▷o— $x + y$

AND $\begin{matrix} x \\ y \end{matrix}$ ▷o ▷o ⊃o— $(x' + y')' = xy$

Fig. 3-24 Logic Operations with NOR Gates

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

(a) OR–invert

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

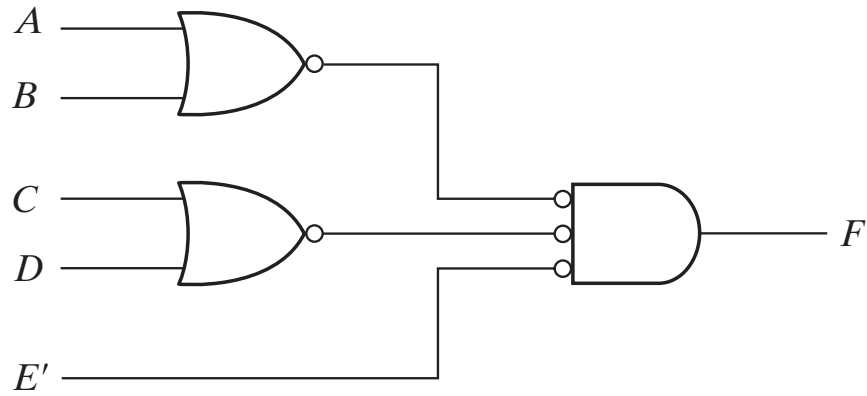(a) Invert–AND

Fig. 3-25 Two Graphic Symbols for NOR Gate

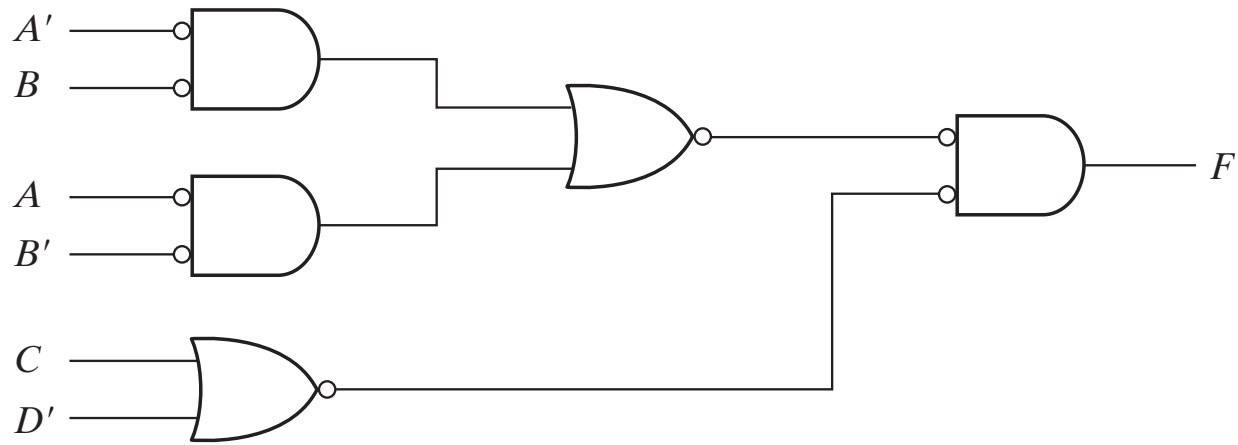Fig. 3-26 Implementing $F = (A + B)(C + D)E$
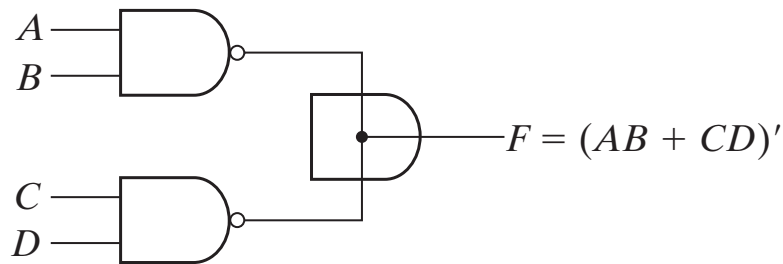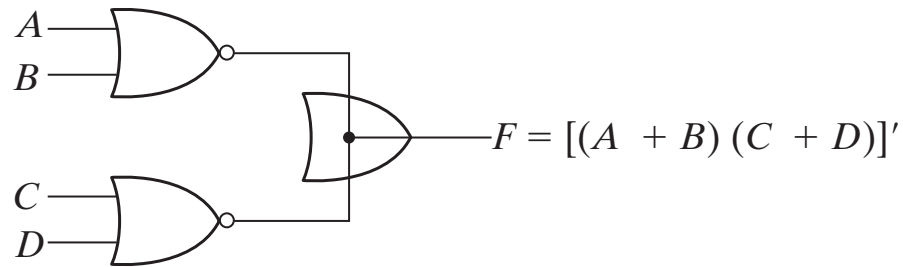
Fig. 3-27 Implementing $F = (AB' + A'B)(C + D')$ with NOR Gates
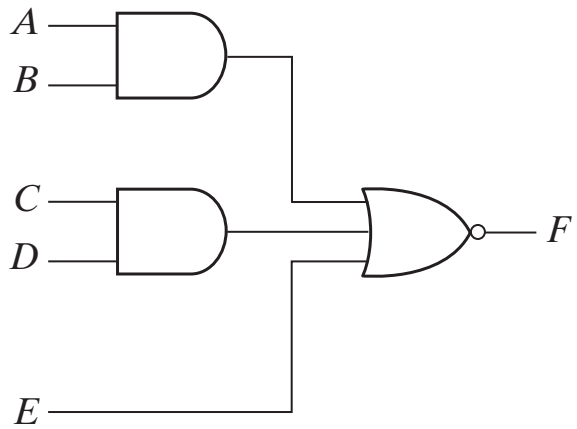
(a) Wired-AND in open-collector
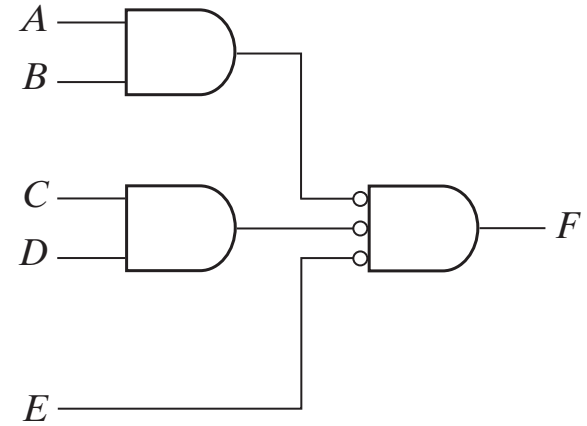   TTL NAND gates.

$F = (AB + CD)'$

(AND-OR-INVERT)

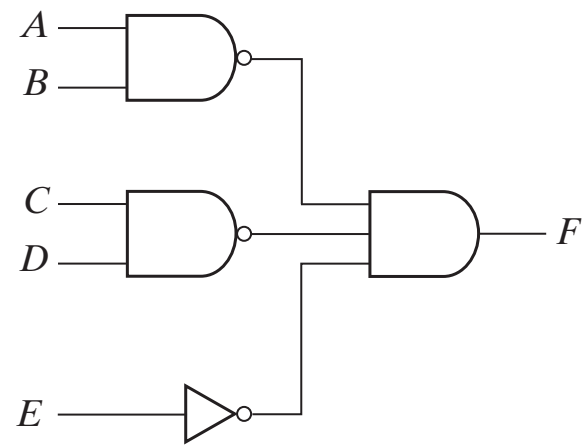(b) Wired-OR in ECL gates

$F = [(A + B)(C + D)]'$

(OR-AND-INVERT)

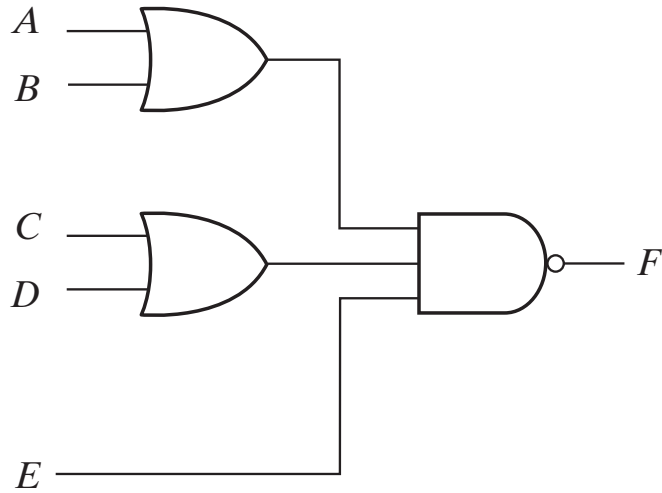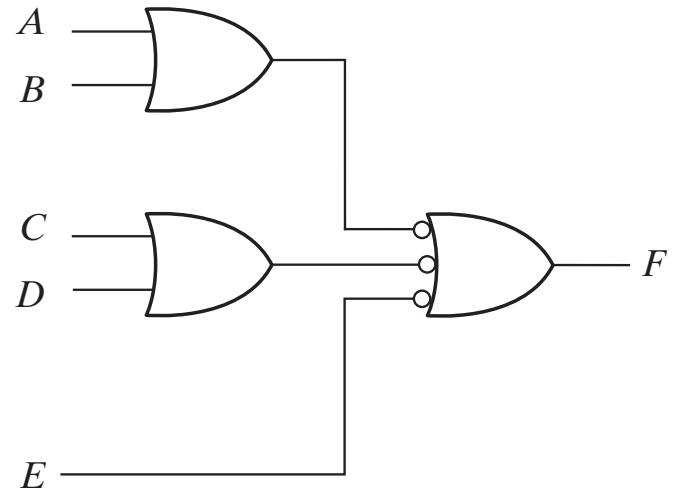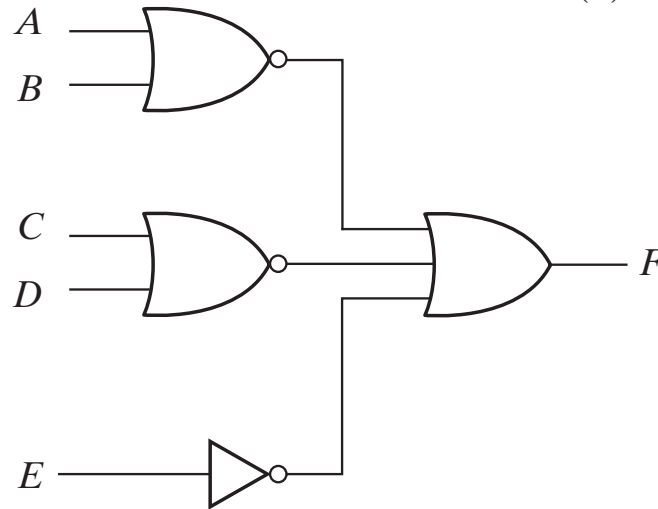Fig. 3-28 Wired Logic

(a) AND-NOR

(b) AND-NOR

(c) NAND-AND

Fig. 3-29 AND-OR-INVERT Circuits; $F = (AB + CD + E)'$

(a) OR-NAND

(b) OR-NAND

(c) NOR-OR

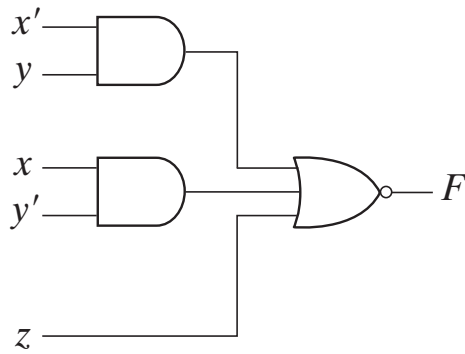Fig. 3-30 OR-AND-INVERT Circuits; $F = [(A + B)(C + D)E]'$

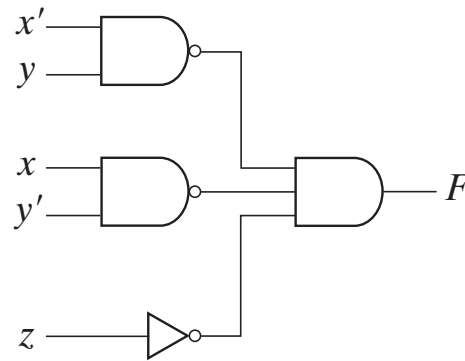|	yz		$\overline{\quad y \quad}$	
x	0 0	0 1	11	10
0	1	0	0	0
1	0	0	0	1

$F = x'y'z' + xyz'$

$F' = x'y + xy' + z$

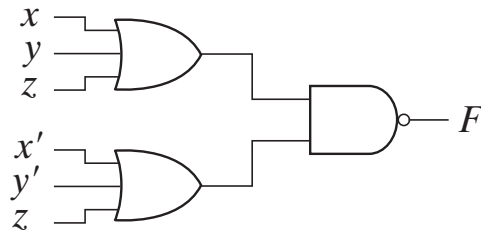(a) Map simplification in sum of products.
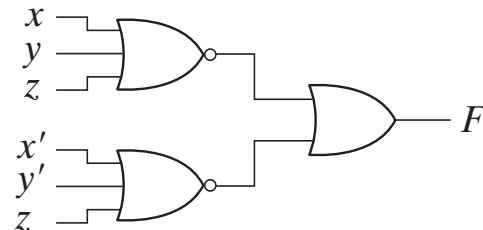
AND-NOR   NAND-AND

(b) $F = (x'y + xy' + z)'$
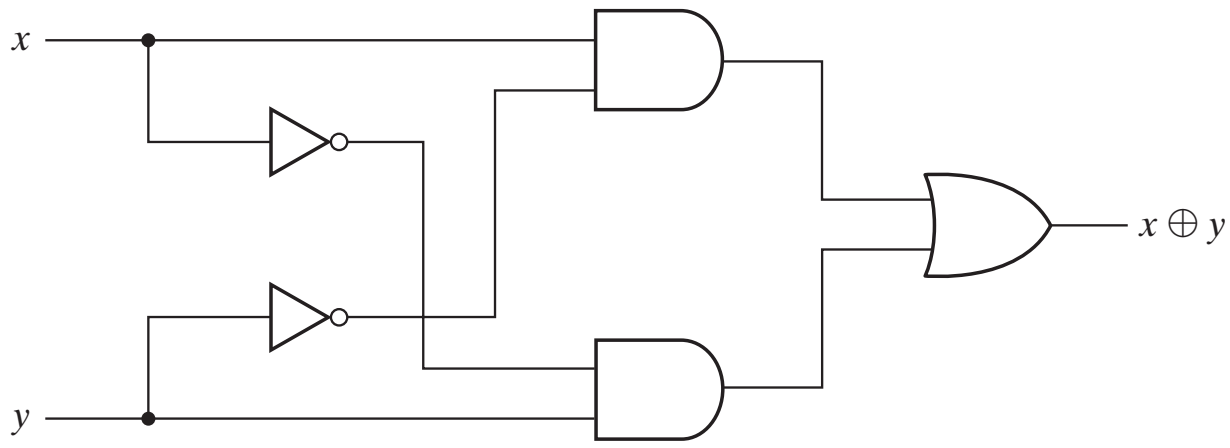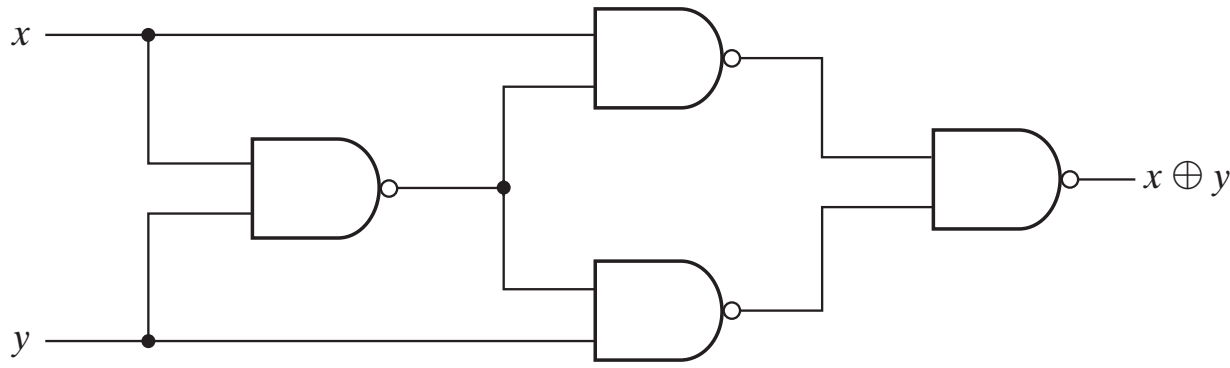
OR-NAND   NOR-OR

(c) $F = [(x + y + z)(x' + y' + z)]'$

Fig. 3-31 Other Two-level Implementations

(a) With AND-OR-NOT gates

(b) With NAND gates

Fig. 3-32 Exclusive-OR Implementations

Left map:

	BC 0 0	0 1	11	10
A 0		1		1
A 1	1		1	

B spans 11, 10. C spans 0 1, 11, 10.

(a) Odd function

$F = A \oplus B \oplus C$

Right map:

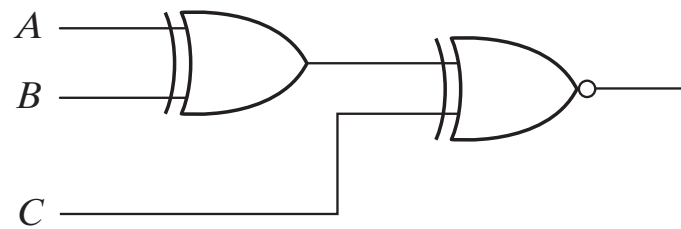|	BC 0 0	0 1	11	10
A 0	1		1	
A 1		1		1

(a) Even function

$F = (A \oplus B \oplus C)'$

Fig. 3-33 Map for a Three-variable Exclusive-OR Function

(a) 3-input odd function   (b) 3-input even function

Fig. 3-34 Logic Diagram of Odd and Even Functions

CD   **C**

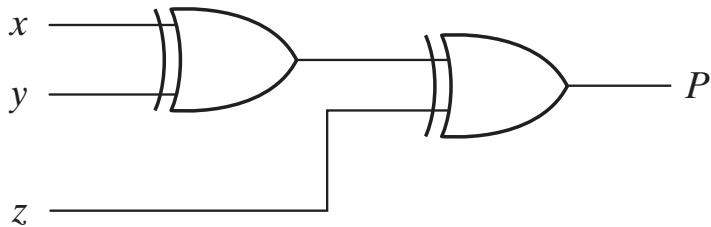| | 0 0 | 0 1 | 11 | 10 |
AB				
00		1		1
01	1		1	
11		1		1
10	1		1	

A { 11, 10 }   *B* { 01, 11 }   *D*

(a) Odd function

$$F = A \oplus B \oplus C \oplus D$$

CD   **C**

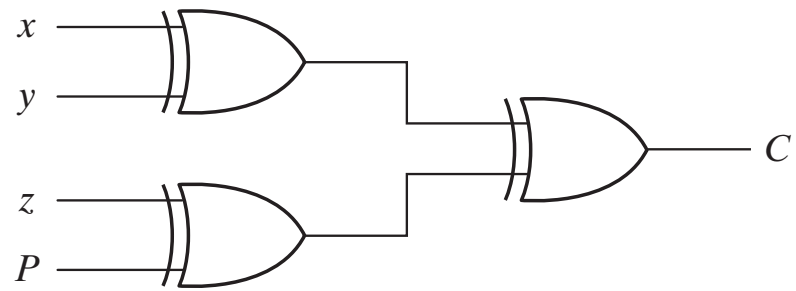| | 0 0 | 0 1 | 11 | 10 |
AB				
00	1		1	
01		1		1
11	1		1	
10		1		1

A { 11, 10 }   *B* { 01, 11 }   *D*

(b) Even function

$$F = (A \oplus B \oplus C \oplus D)'$$

Fig. 3-35 Map for a Four-variable Exclusive-OR Function

(a) 3-bit even parity generator   (a) 4-bit even parity checker

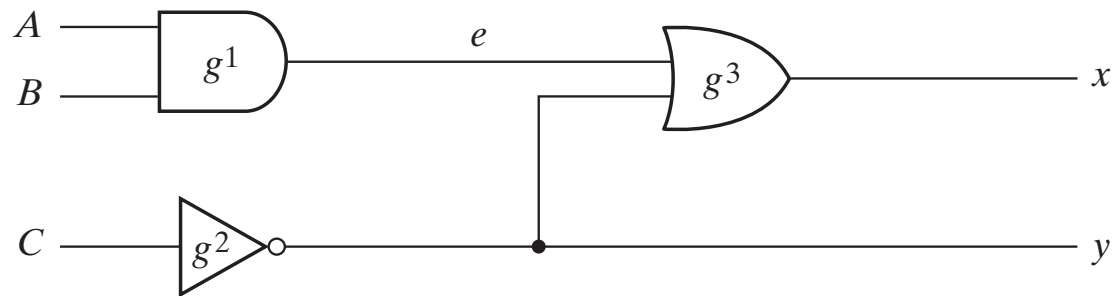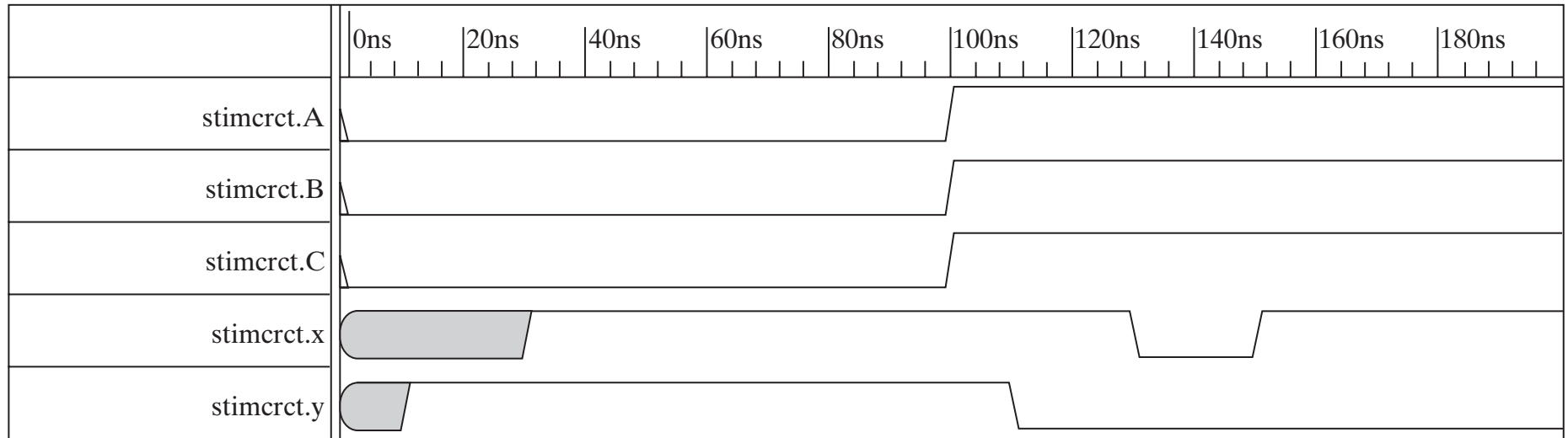Fig. 3-36 Logic Diagram of a Parity Generator and Checker

Fig. 3-37 Circuit to Demonstrate HDL

Fig. 3-38 Simulation Output of HDL Example 3-3