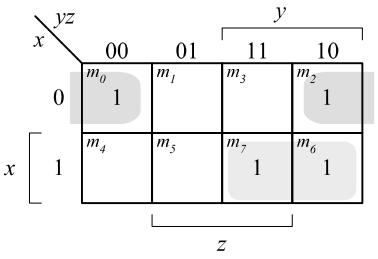
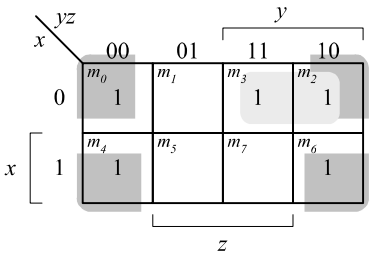
Chapter 3

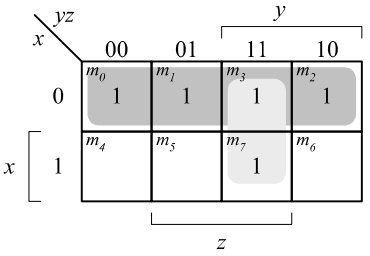
3.1



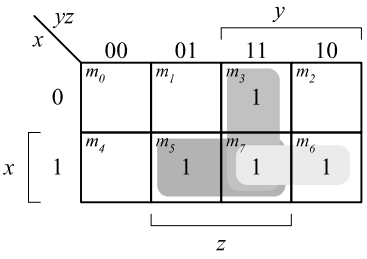
(a) $F = xy + x'z'$



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

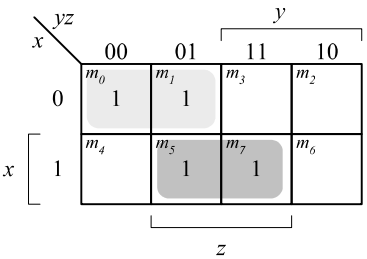


(c) $F = x' + yz$

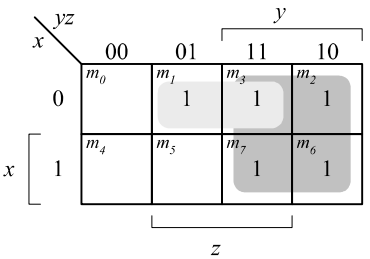


(d) $F = xy + xz + yz$

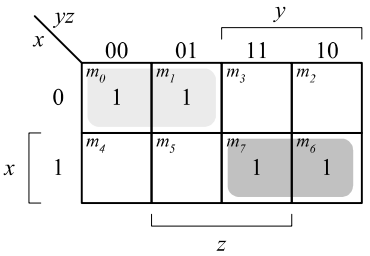
3.2



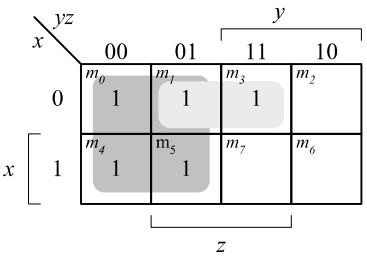
(a) $F = x'y' + xz$



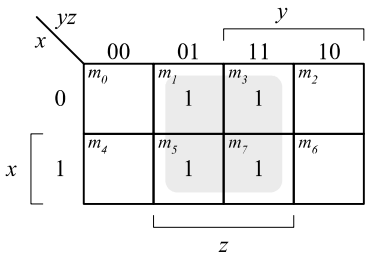
(b) $F = y + x'z$



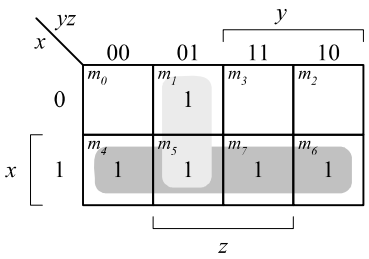
(c) $F = x'y' + xy$



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

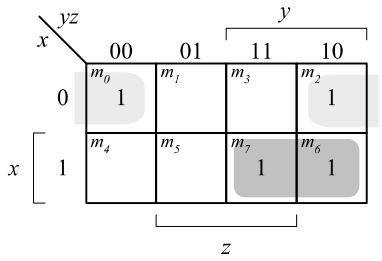


(e) $F = z$

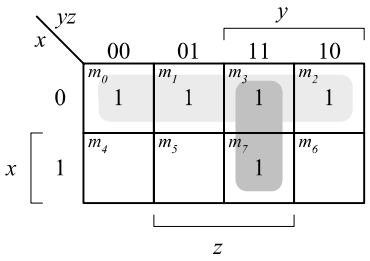


(f) $F = x + y'z$

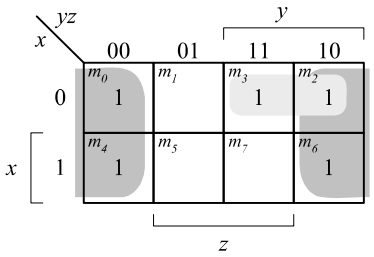
3.3



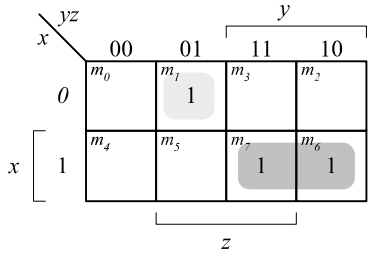
(a) $F = xy + x'y'z' + x'yz'$
$F = xy + x'z'$



(b) $F = x'y' + yz + x'yz'$
$F = x' + yz$

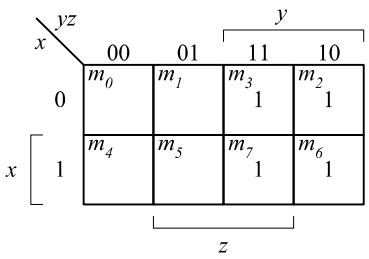


(c) $F = x'y + yz' + y'z'$
$F = = x'y + z'$

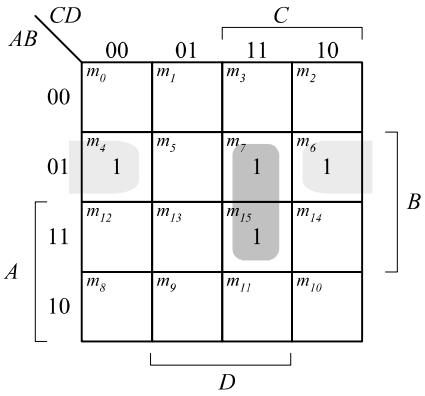


(d) $F = xyz + x'y'z + xyz'$
$F = x'y'z + xy$

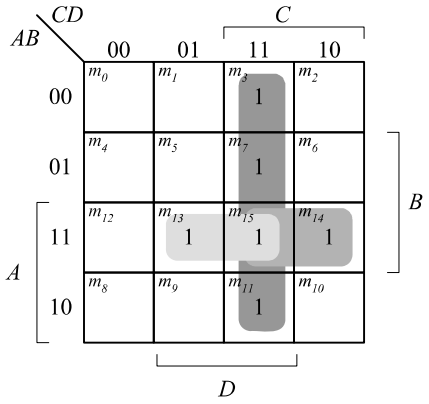
3.4



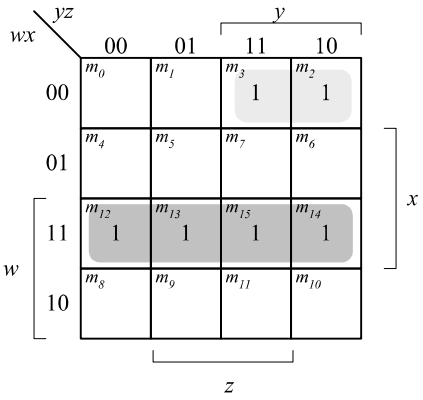
(a) $F = y$



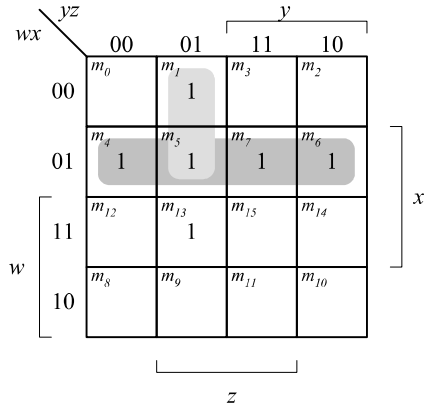
(b) $F = BCD + A'BD'$



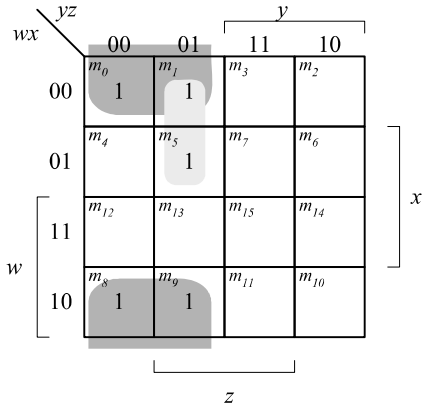
(c) $F = CD + ABD + ABC$



(d) $F = w'x'y + wx$

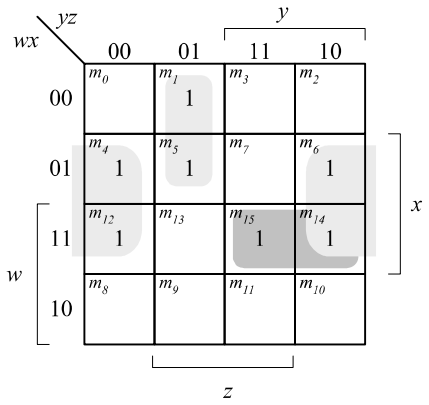


(e) $F = w'x + w'y'z$

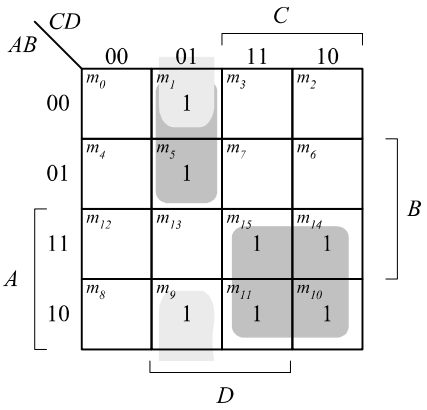


(f) $F = x'y' + w'y'z$

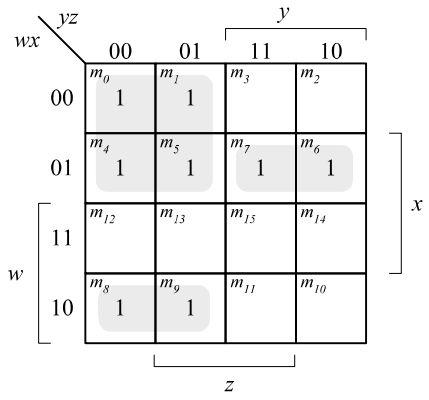
3.5



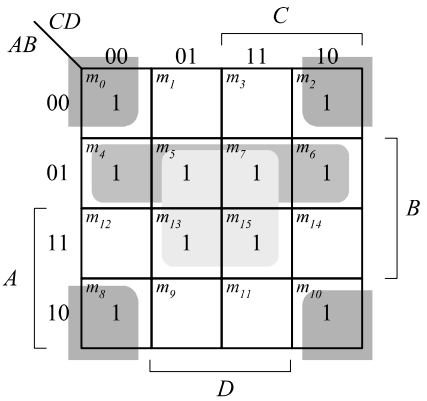
(a) $F = xz' + w'y'z + wxy$



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

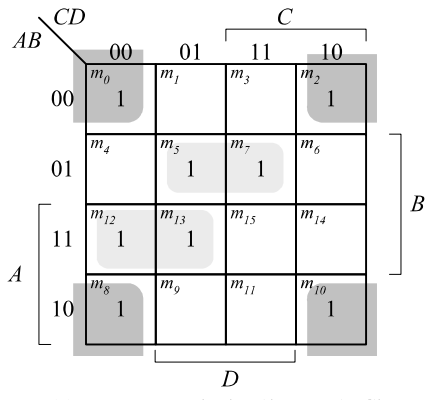


(c) $F = w'y' + wx'y' + w'xy$

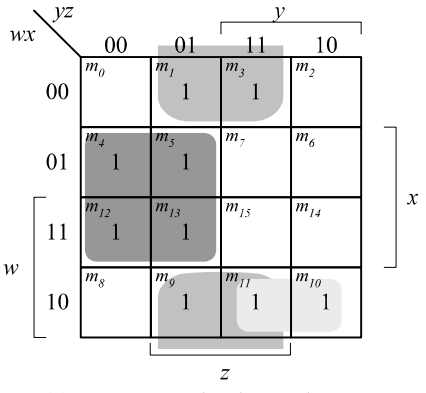


(d) $F = BD + A'B + B'D'$
$or = BD + B'D' + A'D'$

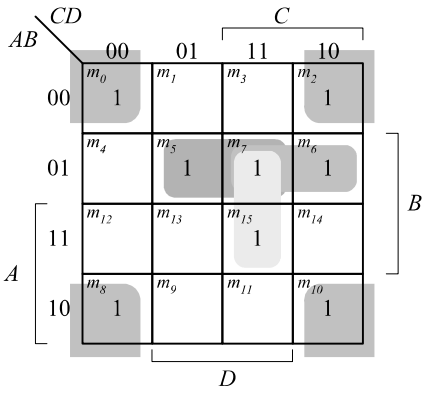
3.6



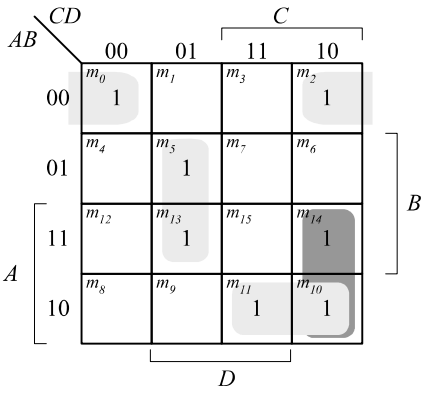
(a) $F = B'D' + A'BD + ABC'$



(b) $F = xy' + x'z + wx'y$

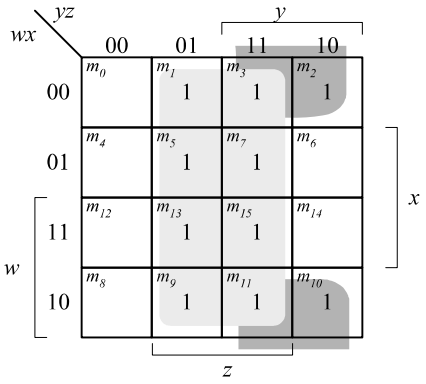


(c) $F = B'D' + BCD + A'BD + A'BC$

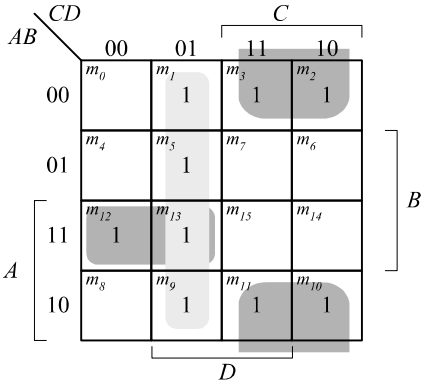


(d) $F = A'B'D' + BC'D + ACD' + AB'C$

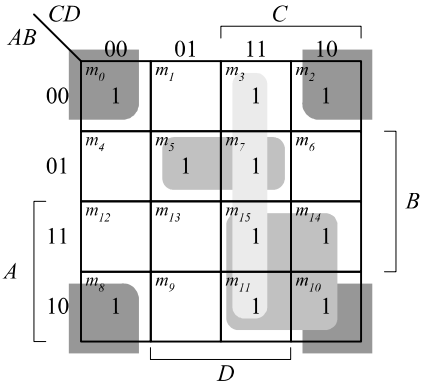
3.7



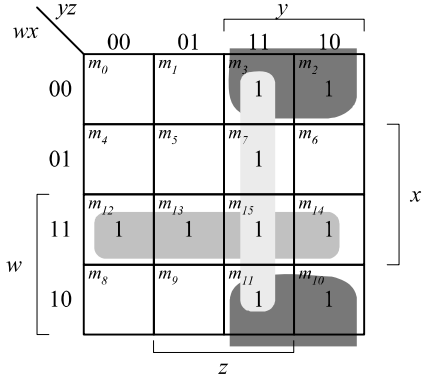
(a) $F = z + x'y$



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



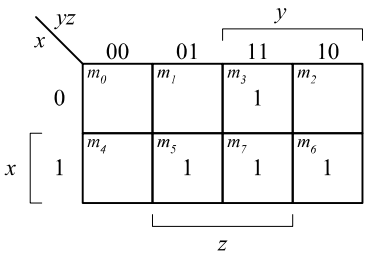
(c) $F = B'D' + AC + A'BD + CD$ (or $B'C$)



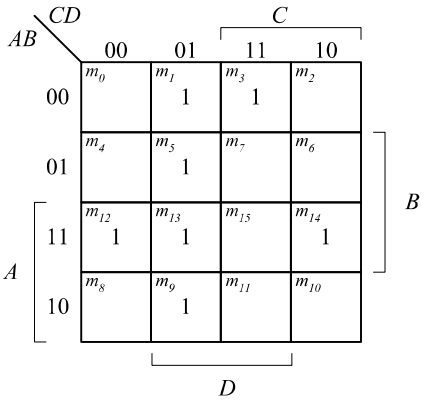
(d) $F = wx + x'y + yz$

3.8

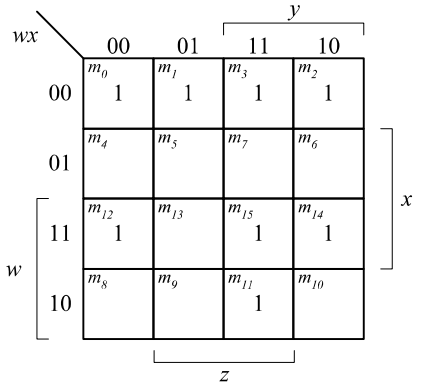
(a) $F(x, y, z) = \Sigma(3, 5, 6, 7)$



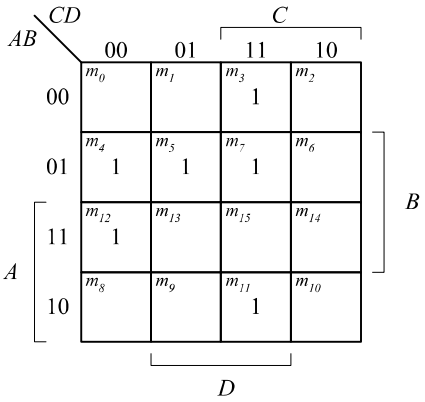
(b) $F = \Sigma(1, 3, 5, 9, 12, 13, 14)$



(c) $F = \Sigma(0, 1, 2, 3, 11, 12, 14, 15)$

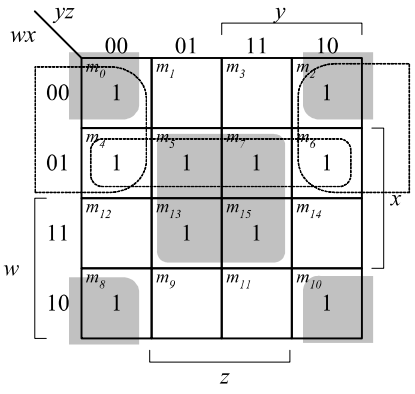


(d) $F = \Sigma(3, 4, 5, 7, 11, 12)$



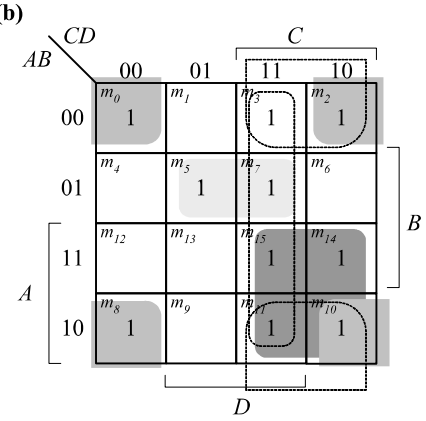
3.9

(a)



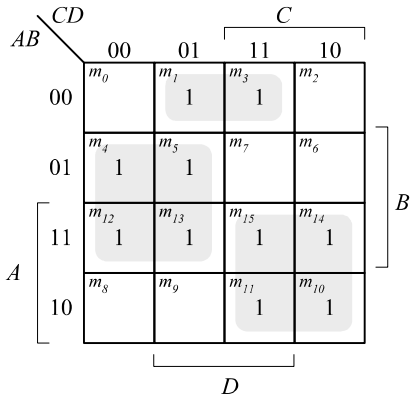
Essential: *xz, x'z'*
Non-essential: *w'x, w'z'*
F = xz + x'z' + (w'x or w'z')

(b)



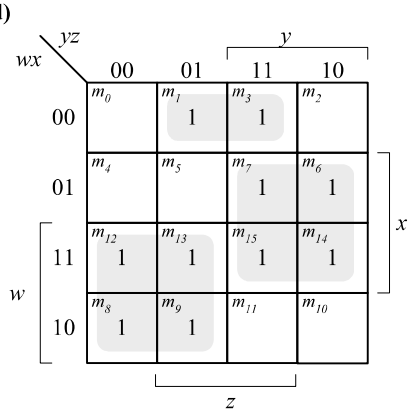
Essential: *B'D', AC, A'BD*
Non-essential: *CD, B'C*
F = B'D' + AC + A'BD + (CD OR B'C)

(c)



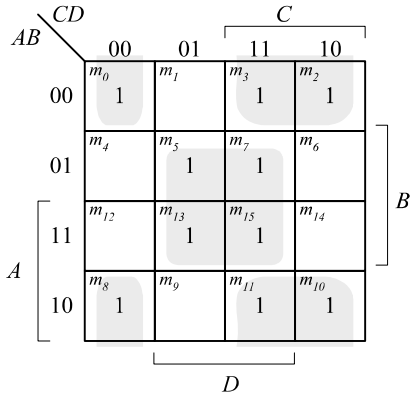
Essential: *BC', AC, A'B'D*
F = BC' + AC + A'B'D

(d)



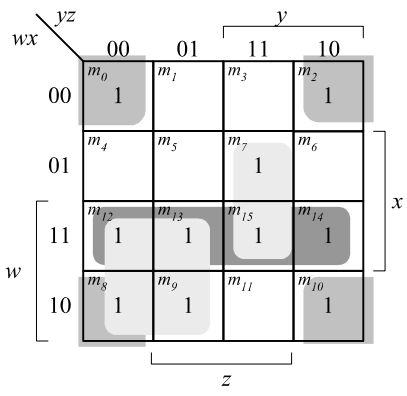
Essential: *wy', xy, w'x'z*
F = wy' + xy + w'x'z

(e)



Essential: *BD, B'C, B'C'D'*
F = BD + B'C + B'C'D'

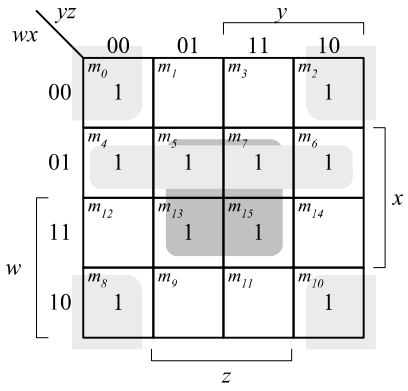
(f)



Essential: *wy', wx, x'z', xyz*
F = wy' + wx + x'z' + xyz

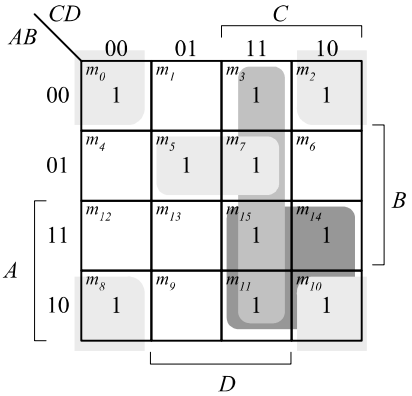
3.10

(a)



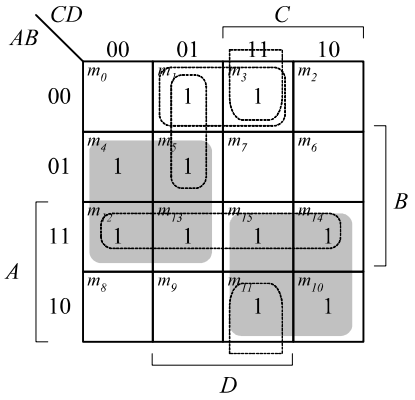
Essential: *xz, w'x, x'z'*
F = xz + w'x + x'z'

(b)



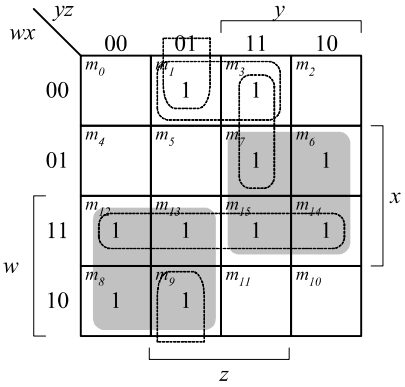
Essential: *AC, B'D', CD, A'BD*
F = AC + B'D' + CD + A'BD

(c)



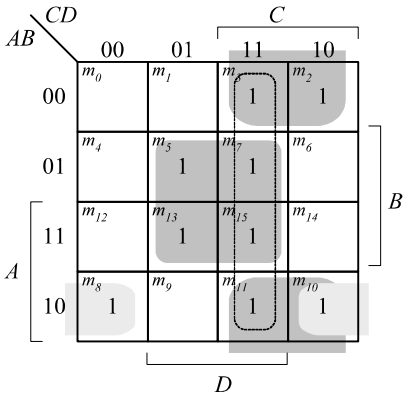
Essential: BC', AC
Non-essential: *AB, A′B′D, B′CD, A′C′D*
F = BC′ + AC + A′B′D

(d)



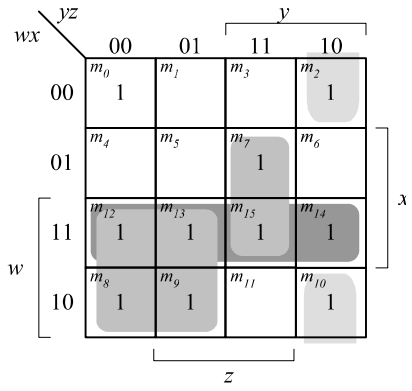
Essential: *wy′, xy*
Non-essential: *wx, x′y′z, w′wz, w′x′z*
F = wy′ + xy + w′x′z

(e)



Essential: *BD, B′C, AB′C*
Non-essential: *CD*
F = BD + B′C + AB′C

(f)

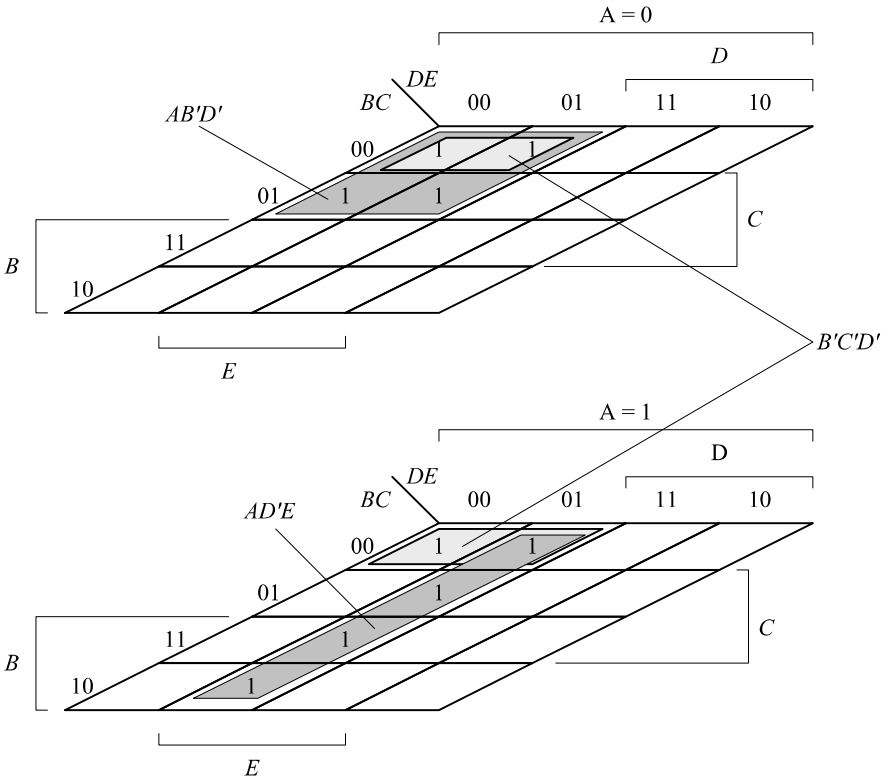


Essential: *wy′, wx, xyz, x′yz′*
F = wy′ + wx + xyz + x=yz′

3.11 **(a)** $F(A, B, C, D, E) = \sum (0, 1, 4, 5, 16, 17, 21, 25, 29$

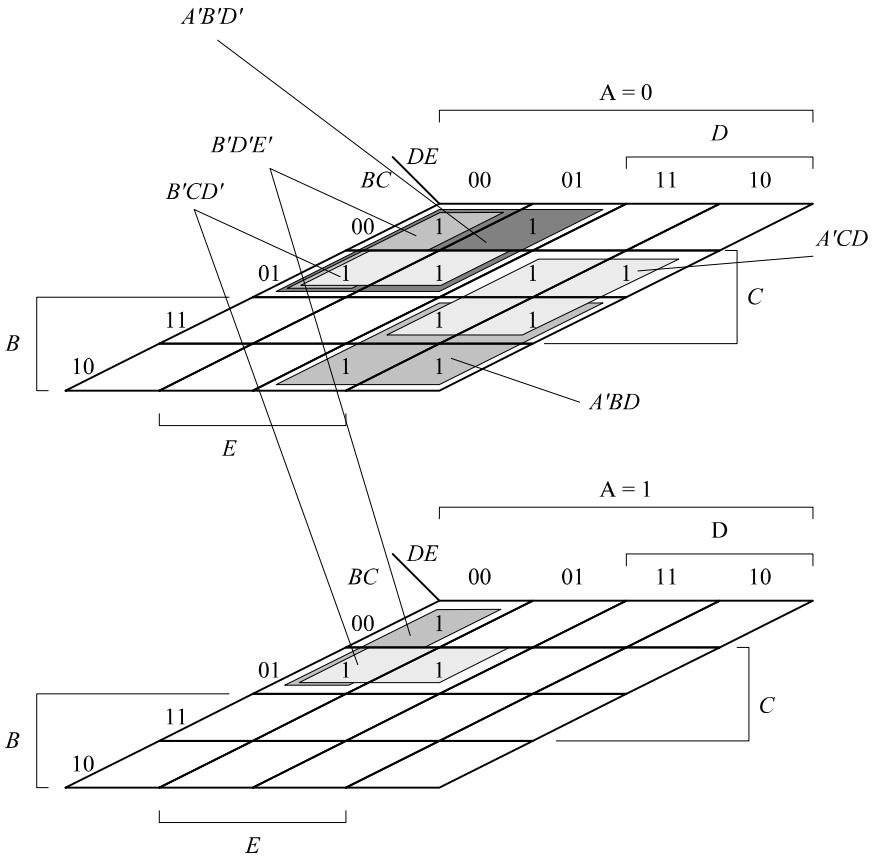
$$F = A'B'D' + AD'E + B'C'D'$$

m_0: $A'B'C'D'E'$ = 00000
m_1: $A'B'C'D'E$ = 00001
m_4: $A'B'CD'E'$ = 00100
m_5: $A'B'CD'E$ = 00101
m_{16}: $AB'C'D'E'$ = 10000
m_{17}: $AB'C'D'E$ = 10001
m_{21}: $AB'CD'E$ = 10101
m_{25}: $ABC'D'E$ = 11001
m_{29}: $ABCD'E$ = 11101



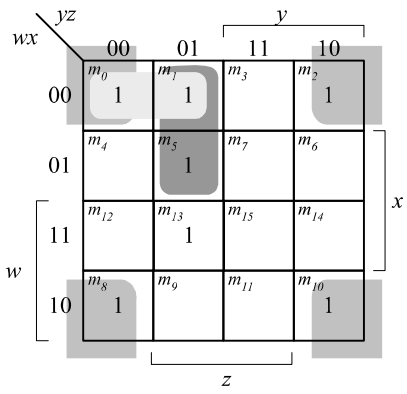
(b) *F(A, B, C, D, E) = A'B'CE' + B'C'D'E' + A'B'D' + B'CD' + A'CD + A'BD*
F(A, B, C, D, E) = A'B'D' + B'D'E' + B'CD' + A'CD + A'BD

A'B'CE': *AB'CDE' + A'B'CD'E'*
B'C'D'E': *AB'C'D'E' + A'B'C'D'E'*
A'B'D': *A'B'CD'E + A'B'CD'E' + A'B'C'D'E + A'B'C'D'E'*
B'CD': *AB'CD'E + AB'CD'E' + A'B'CD'E + A'B'CD'E'*
A'CD: *A'BCDE + A'BCDE' + A'B'CDE + A'B'CDE'*
A'BD: *A'BCDE + A'BCDE' + A'BC'DE + A'BC'DE'*



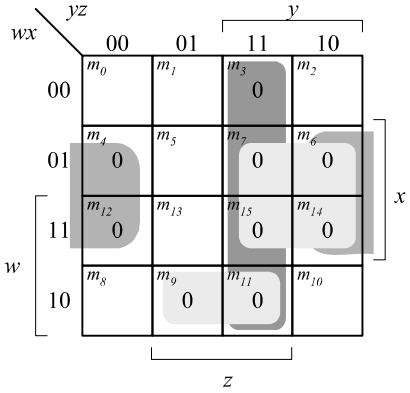
3.12

(a)



$F = \Sigma(0, 1, 2, 5, 8, 10, 13)$

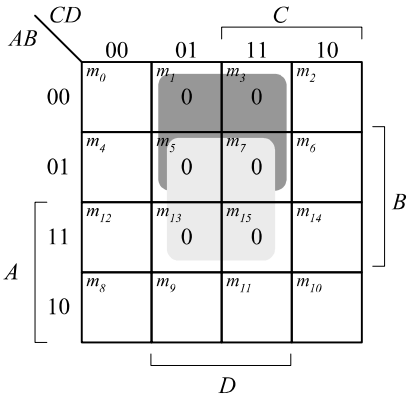
$F = x'z' + w'x'y' + w'y'z$



$F' = yz + xz' + xy + wx'z$

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

(b)



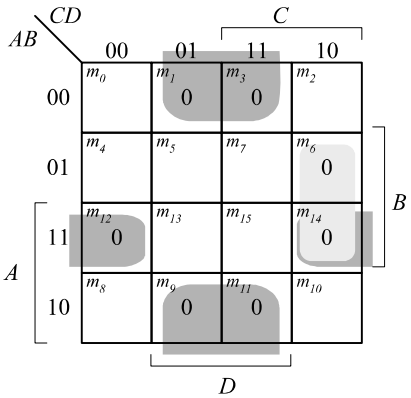
$F = \Pi(1, 3, 5, 7, 13, 15)$

$F' = A'D + B'D$

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

$F = C'D' + AB' + CD'$

(c)



$F = \Pi(1, 3, 6, 9, 11, 12, 14)$

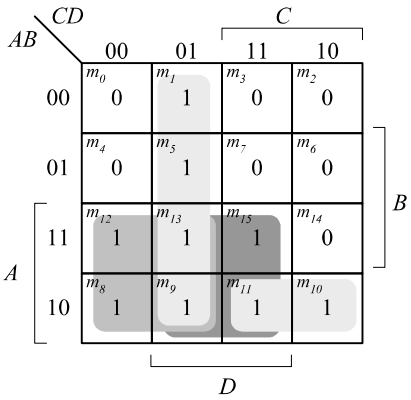
$F' = B'D + BCD' + ABD'$

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

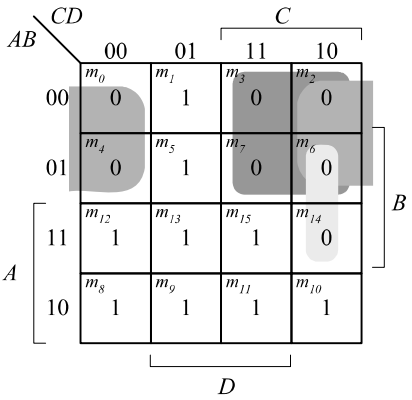
$F = BD + B'D' + A'C'D'$

3.13 **(a)** $F = xy + z' = (x + z')(y + z')$

(b)



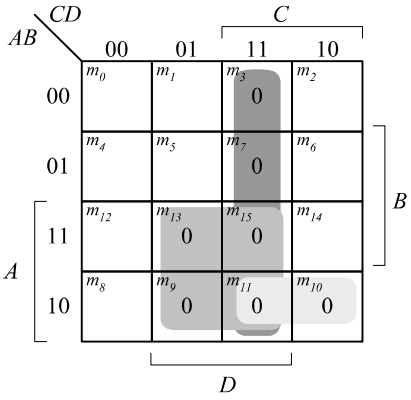
$F = AC' + AD + C'D + AB'C$

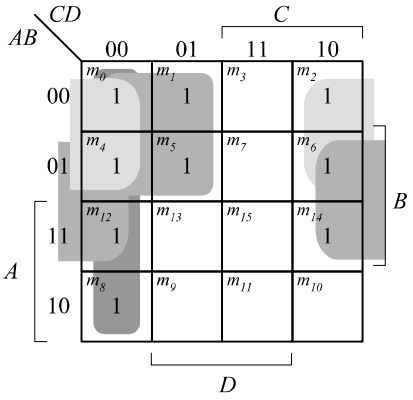


$F' A'D' + A'C + BCD'$

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

(c)





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

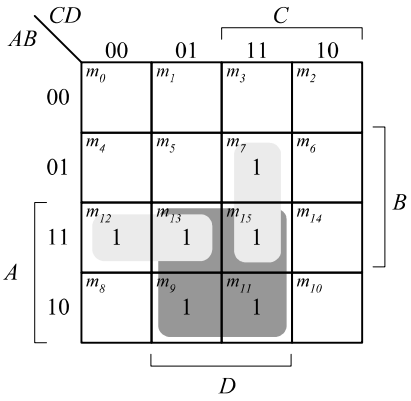
$F' = A'CD + ABD + AB'D + AB'C$

$F = A'C + A'D' + BD' + C'D'$

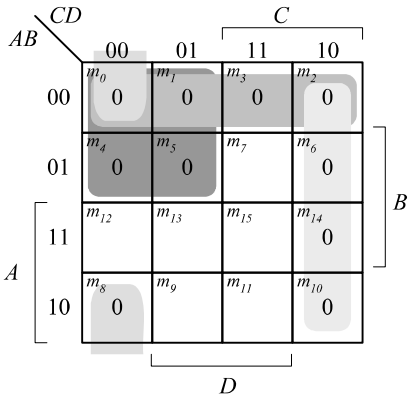
$F' = AD + CD + AB'C$

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

(d)

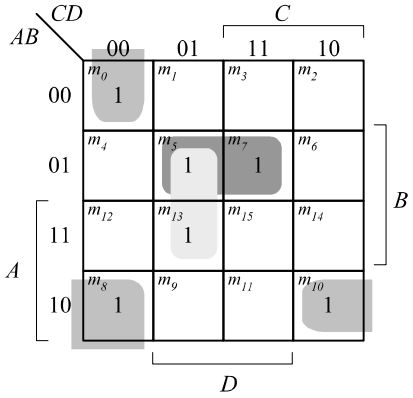


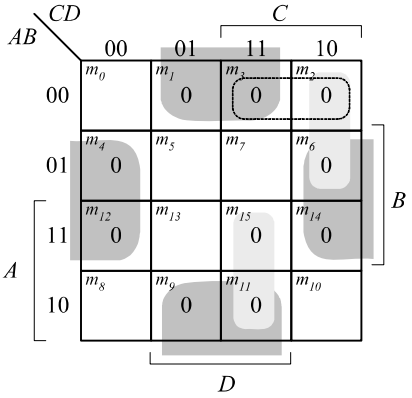
$F = ABC' + AB'D + BCD$
$F = AD + ABC' + BCD$



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

3.14





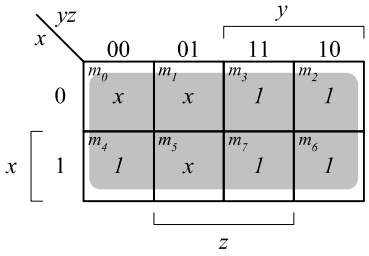
SOP form (using 1s): $F = B'C'D' + AB'D' + BC'D + A'BD$
$F = B'D'(A + C') + BD(A' + C')$

POS form (using 0s): $F' = BD' + B'D + A'CD' + ACD$
$F = [(B' + D)(B + D')][(A + C' + D)(A' + C' + D')]$

Alternative POS: $F' = BD' + B'D + A'CD' + A'B'C$
$F = [(B' + D)(B + D')][(A + C' + D)(A' + B + C)]$

3.15

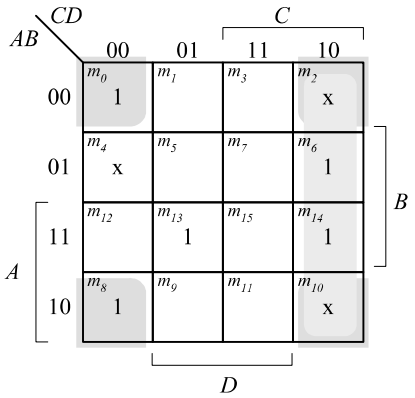
(a)



$F = 1$

$F = \Sigma(0,1,2,3,4,5,6,7)$

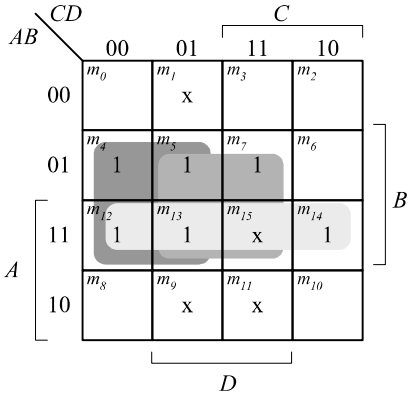
(b)



$F = B'D' + ABC'D$

$F = \Sigma(0, 2, 6, 8, 10, 13, 14)$

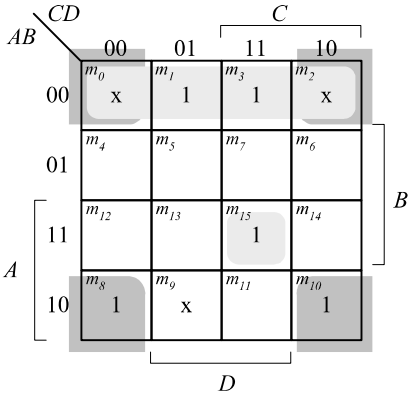
(c)



$F = BC' + BD + AB$

$F = \Sigma(4, 5, 7, 12, 13, 14, 15)$

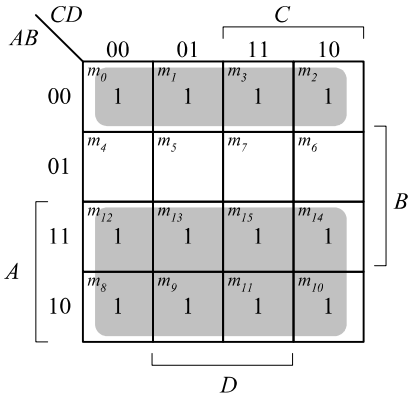
(d)



$F = B'D' + A'B' + ABCD$

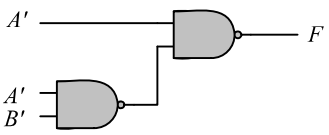
$F =$ $F = \Sigma(0, 1, 2, 3, 8, 10, 15)$

3.16 **(a)**

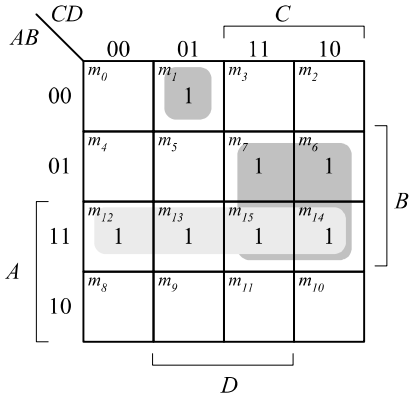


$F = A + A'B'$

$F = (A'(A'B')')'$

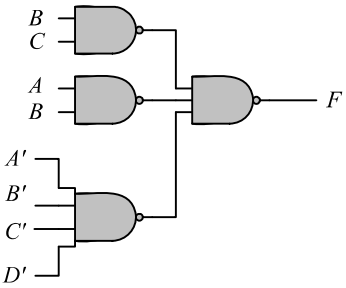


(b)

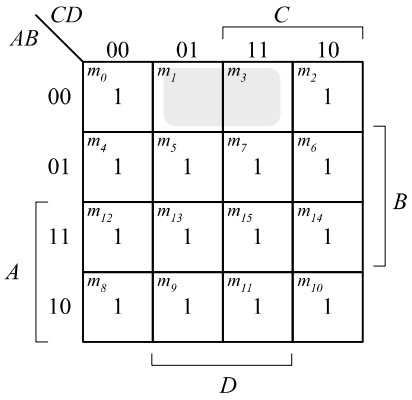


$F = BC + AB + A'B'C'D$

$F = ((BC)'(AB)'(A'B'C'D)')'$

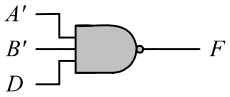


(c)

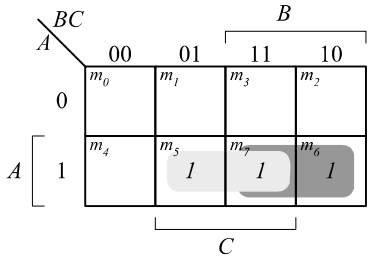


$F' = A'B'D$

$F = (A'B'D)'$

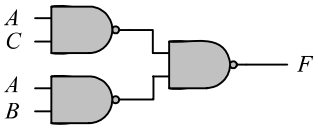


(d)

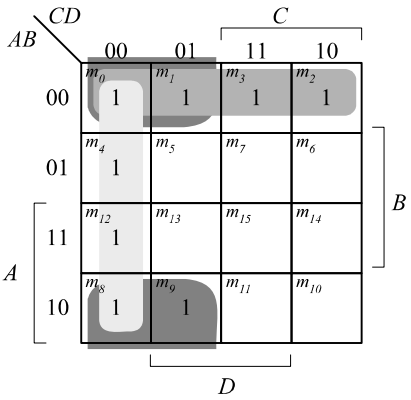


$F = AC + AB$

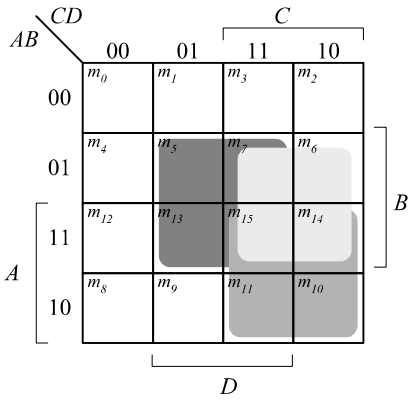
$F = ((AC)'(AB)')'$



3.17

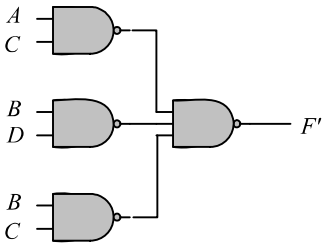


$F = A'B' + C'D' + B'C'$

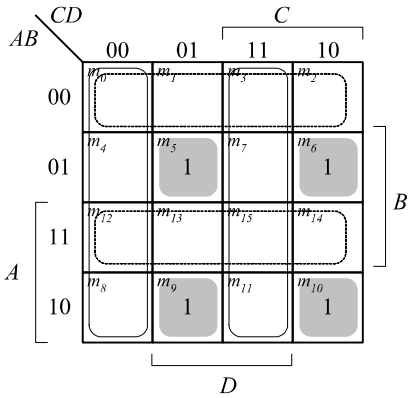


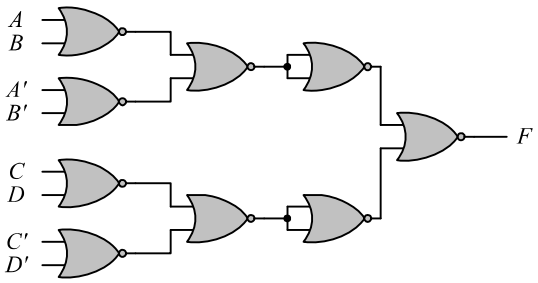
$F' = BC + AC + BD$

$F = (BC)'(AC)'(BD)'$



3.18 $F = (A \oplus)B'(C \oplus D) = (AB' + A'B)(CD' + C'D) = AB'CD' + AB'C'D + A'BCD' + A'BC'D$





$F = AB'CD' + AB'C'D + A'BCD' + A'BC'D$ and $F' = A'B' + AB + C'D' + CD$

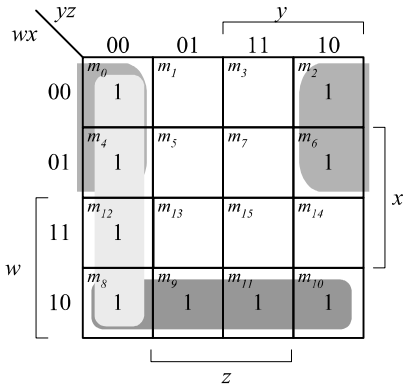
$F = (A'B')'(AB)'(C'D')'(CD)' = (A + B)(A' + B')\ (C' + D')(C + D)$

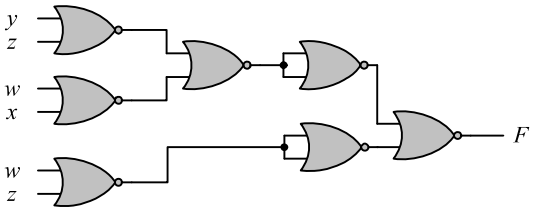
$F' = [(A + B)(A' + B')]' + [(C' + D')(C + D)]'$

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

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

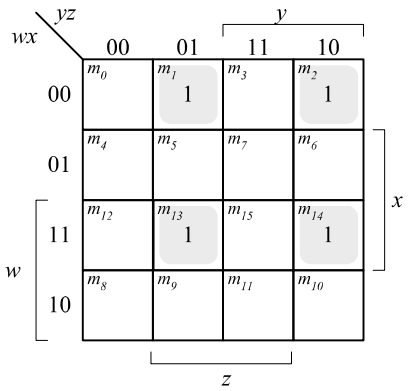
3.19 **(a)** $F = (w + z')(x' + z')(w' + x' + y')$

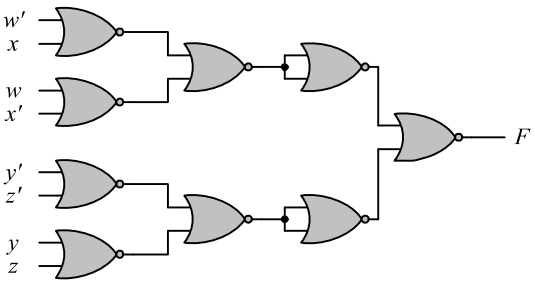




$F = y'z' + wx' + w'z'$
$F = [(y + z)' + (w' + x)' + (w + z)']$
$F' = [(y + z)' + (w' + x)' + (w + z)']'$

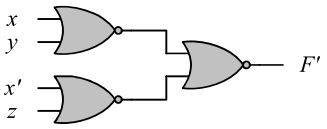
(b)



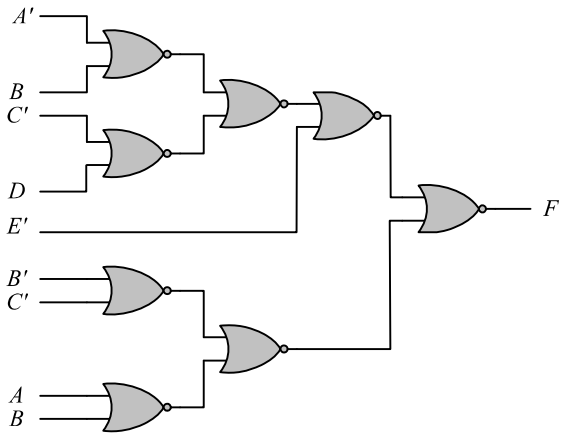


$F = \Sigma(1, 2, 13, 14)$
$F' = w'x + wx' + y'z' + yz = [(w + x')(w' + x)(y + z)(y' + z')]'$
$F = (w + x')' + (w' + x)' + (y + z)' + (y' + z')$

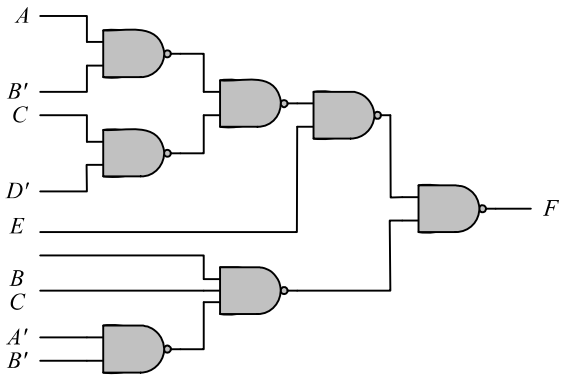
(c) $F = [(x + y)(x' + z)]' = (x + y)' + (x' + z)'$
$F' = [(x + y)' + (x' + z')']'$



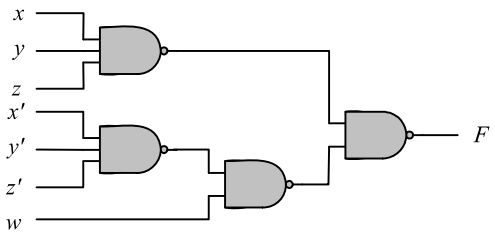
3.20 Multi-level NOR:
$F = (AB' + CD')E + BC(A + B)$
$F' = [(AB' + CD')E + BC(A + B)]'$
$F' = [\ [(AB' + CD')' + E']' + [\ (BC)' + (A + B)']'\ ]'$
$F' = [\ [((A' + B)' + (C' + D)')' + E']' + [\ (B' + C')' + (A + B)']'\ ]'$



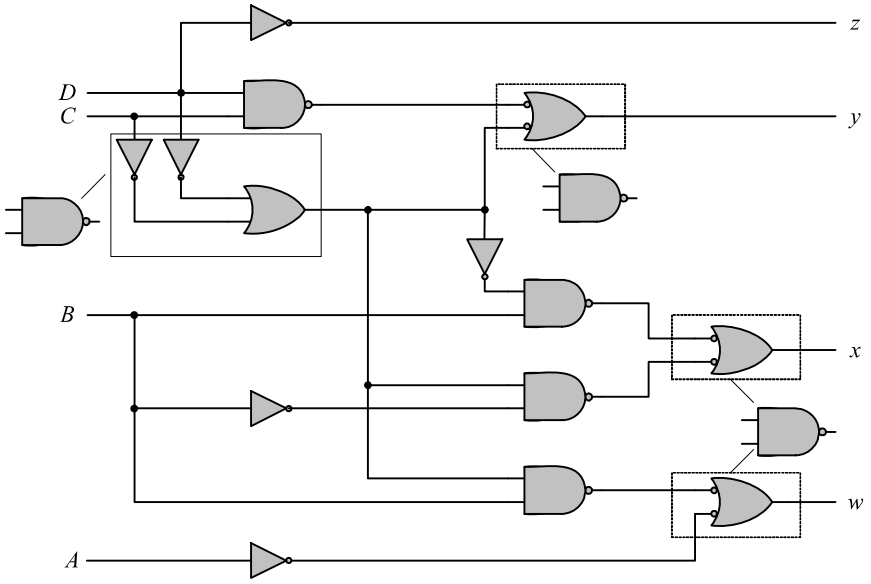
Multi-level NAND:
F = (AB' + CD')E + BC(A + B)
F' = [(AB' + CD')E]' [BC(A + B)]'
F' = [((AB')'(CD')')'E]' [BC(A'B')']'



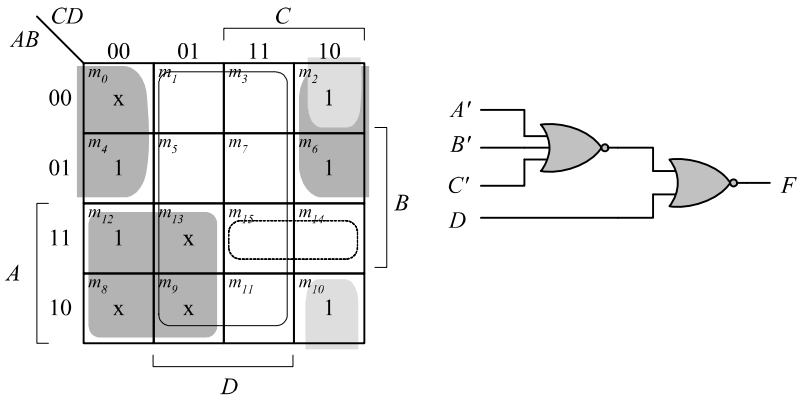
3.21 *F = w(x + y + z) + xyz*
F' = [w(x + y + z)]'[xyz]' = [w(x'y'z')']'(xyz)'



3.22



3.23

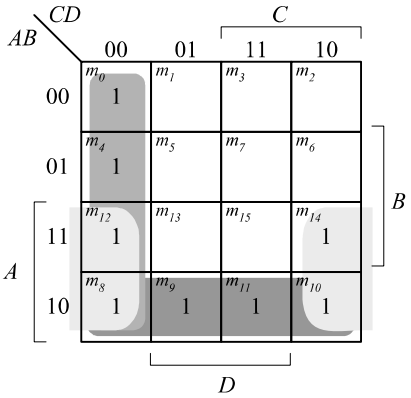


$F = AC' + A'D' + B'CD'$

$F' = D + ABC$

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

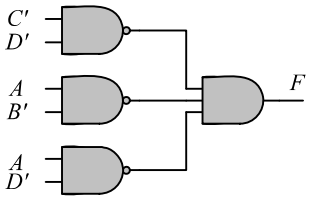
3.24



(a) $F = C'D' + AB' + AD'$

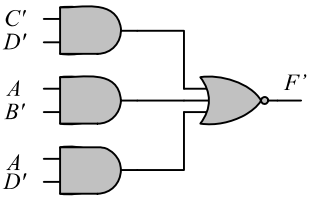
$F' = (C'D')'(AB')'(AD')'$

AND-NAND:



(b) $F' = [C'D' + AB' + AD']'$

AND-NOR:

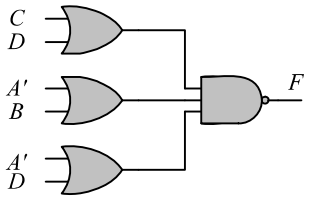


(c) $F = C'D' + AB' + AD' = (C + D)' + (A' + B)' + (A' + D)'$

$F' = (C'D')'(AB')'(AD')' = (C + D)(A' + B)(A' + D)$

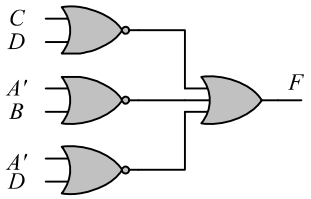
$F = [\,(C + D)(A' + B)(A' + D)\,]'$

OR-NAND:

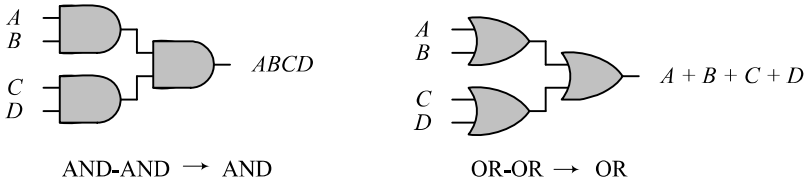


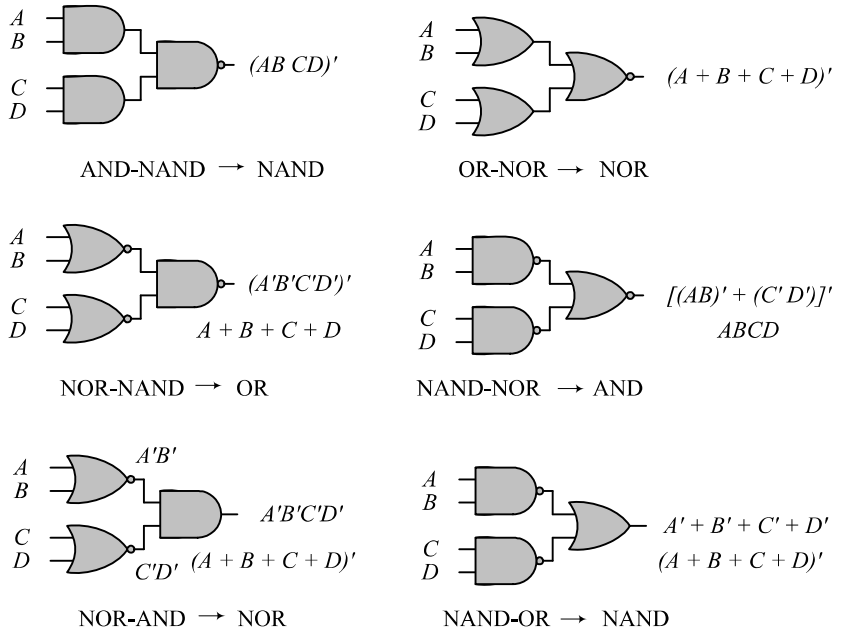
(d) $F = C'D' + AB' + AD' = (C + D)' + (A' + B)' + (A' + D)'$

NOR-OR:



3.25





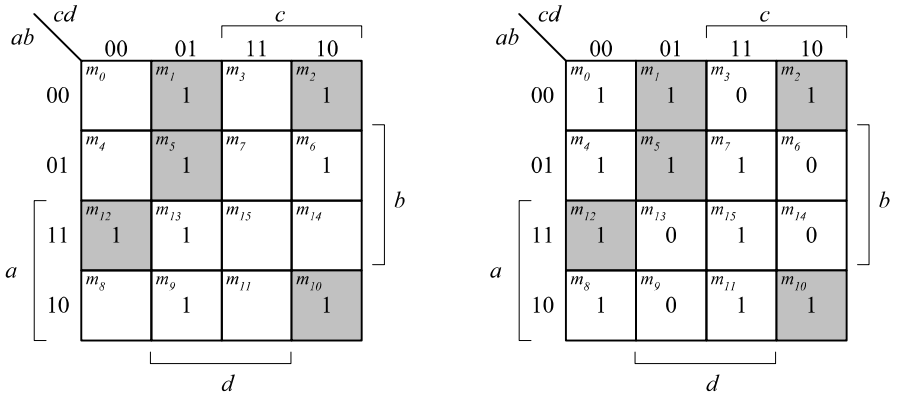
The degenerate forms use 2-input gates to implement the functionality of 4-input gates.

3.26

$f = abc' + c'd + a'cd' + b'cd'$

$g = (a + b + c' + d')(b' + c' + d)(a' + c + d')$

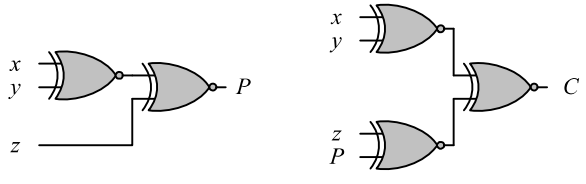
$g' = a'b'cd + bcd' + ac'd$



$fg = ac'd + abc'd + b'cd'$

3.27 $x \oplus y = x'y + xy'$; *Dual* $= (x' + y)(x + y') = (x \oplus y)'$

3.28



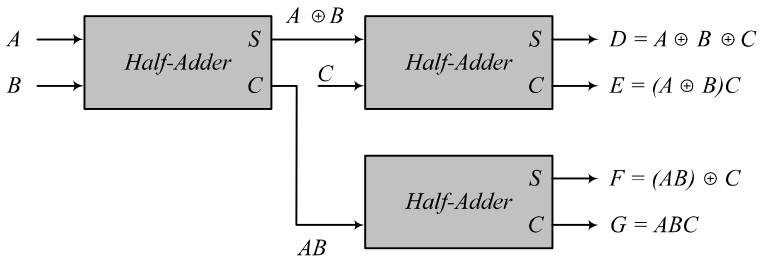
(a) 3-bit odd parity generator *(b) 4-bit odd parity generator*

3.29 $D = A \oplus B \oplus C$

$E = A'BC + AB'C = (A \oplus B)C$

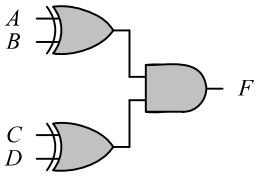
$F = ABC' + (A' + B')C = ABC' + (AB)'C = (AB) \oplus C$

$G = ABC$



3.30 $F = AB'CD' + A'BCD' + AB'C'D + A'BC'D$

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



3.31 Note: It is assumed that a complemented input is generated by another circuit that is not part of the circuit that is to be described.

(a)

```
module Fig_3_22a_gates (F, A, B, C, C_bar, D);
  output   F;
  input    A, B, C, C_bar, D;
  wire     w1, w2, w3, w4;
  and      (w1, C, D);
  or       (w2, w1, B);
  and      (w3, w2, A);
  and      (w4, B, C_bar);
  or       (F, w3, w4);
endmodule
```

(b)

```
module Fig_3_22b_gates (F, A, B, C, C_bar, D);
  output   F;
  input    A, B, C, C_bar, D;
  wire     w1, w2, w3, w4;
  not      (w1_bar, w1);
  not      (B_bar, B);
  not      (w3_bar, w3);
  not      (w4_bar, w4);
  nand     (w1, C, D);
  or       (w2, w1_bar, B_bar);
  nand     (w3, w2, A);
  nand     (w4, B, C_bar);
  or       (F, w3_bar, w4_bar);
endmodule
```