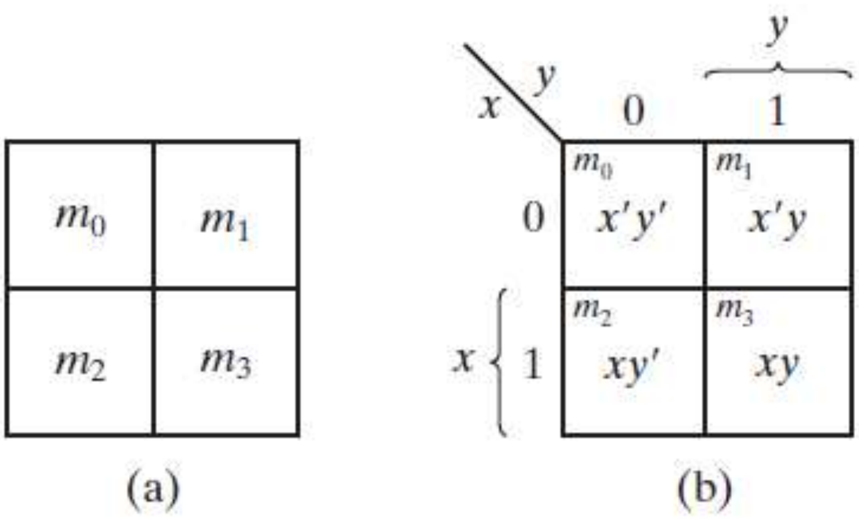
K-maps - Two, Three, and Four Variable K-maps, Don’t-Care Conditions

THE MAP METHOD

- The map method provides a simple, straightforward procedure for minimizing Boolean functions.
- This method may be regarded as a pictorial form of a truth table.
- The map method is also known as the Karnaugh map or K-map.
- K-map is a chart or a graph, composed of an arrangement of adjacent cells, each representing a Minterm or Maxterm.
- The simplified expressions produced by the map are always in one of the two standard forms: Sum of Products (SoP) or Product of Sums (PoS).

Two-Variable K-Map

- There are four Minterms or Maxterms for two variables; hence, the map consists of four squares, one for each minterm.
- The two-variable K-map (minterms) is shown in Fig.



- The map is redrawn in (b) to show the relationship between the squares and the two variables *x* and *y*.
- The 0 and 1 marked in each row and column designate the values of variables.
- Variable *x* appears primed in row 0 and unprimed in row 1.
- Similarly, *y* appears primed in column 0 and unprimed in column 1.

The rules of K-map simplification are:

- Groupings can contain only 1s (for Minterms) and only 0s (for Maxterms).
- Groups can be formed only at right angles (horizontal or vertical); diagonal groups are not allowed.
- The number of 1s or 0s in a group must be a power of 2 – even if it contains a single 1 or 0.
- The groups must be made as large as possible.

- Each cell containing a single 1 or 0 must be in at least one group.
- Groups can overlap.
- Groups can wrap around the sides of the *K*-map.
  - The leftmost cell in a row may be grouped with the rightmost cell and the top cell in a column may be grouped with the bottom cell.

Example: Simplify the Boolean functions

$F(x, y) = \sum(0, 1)$

$x \backslash y$	0	1
0	1	1
1	0	0

$F(x, y) = x'$

$F(x, y) = \sum(2, 3)$

$x \backslash y$	0	1
0	0	0
1	1	1

$F(x, y) = x$

$F(x, y) = \sum(0, 2)$

$x \backslash y$	0	1
0	1	0
1	1	0

$F(x, y) = y'$

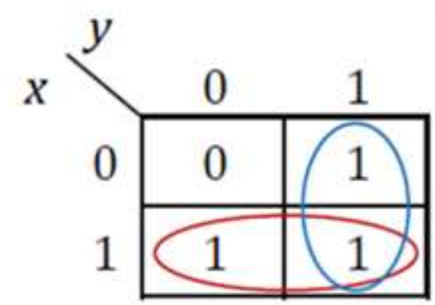
$F(x, y) = \sum(1, 3)$

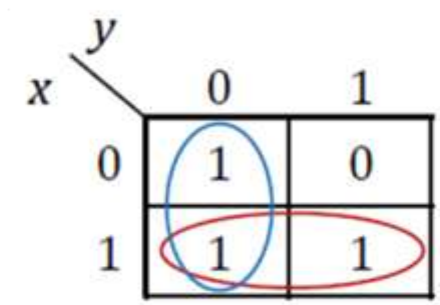
$x \backslash y$	0	1
0	0	1
1	0	1

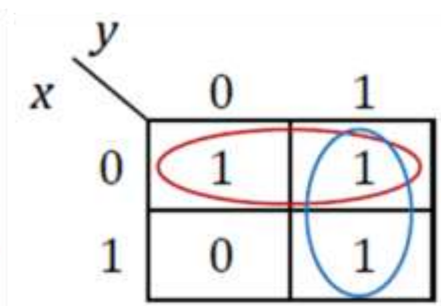
$F(x, y) = y$

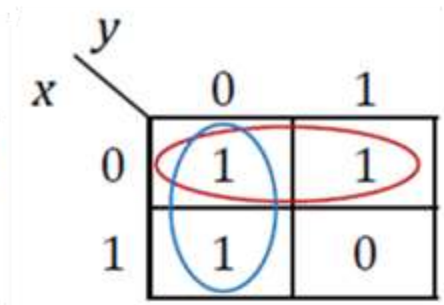
Example: Simplify the Boolean functions

$F(x, y) = \sum(1, 2, 3)$ $F(x, y) = \sum(0, 2, 3)$ $F(x, y) = \sum(0, 1, 3)$ $F(x, y) = \sum(0, 1, 2)$









$F(x, y) = x + y$ $F(x, y) = x + y'$ $F(x, y) = x' + y$ $F(x, y) = x' + y'$

Example: Simplify the Boolean functions

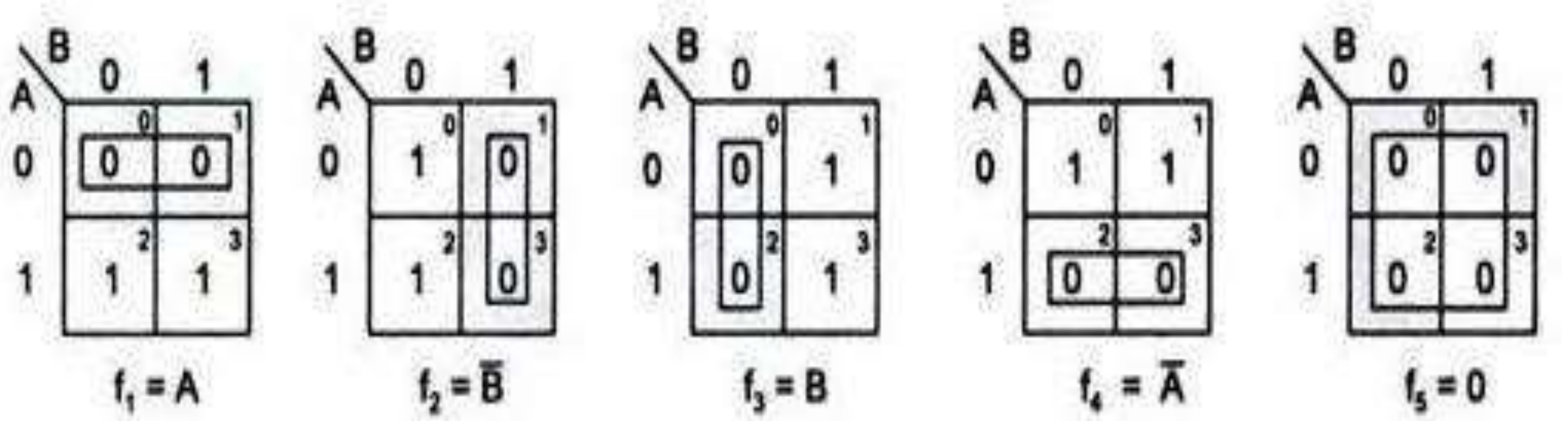
1. $F(A,B) = \prod(0, 1)$

2. $F(A,B) = \prod(1,3)$

3. $F(A,B) = \prod(0, 2)$

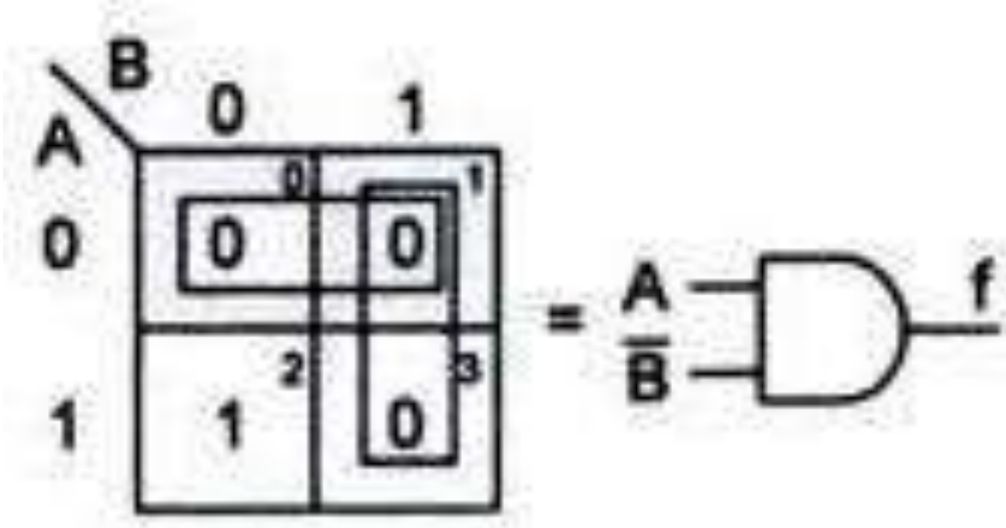
4. $F(A,B) = \prod(2,3)$

5. $F(A,B) = \prod(0,1,2,3)$



Example: Reduce the expression f=(A+B)(A+B')(A'+B') using mapping.

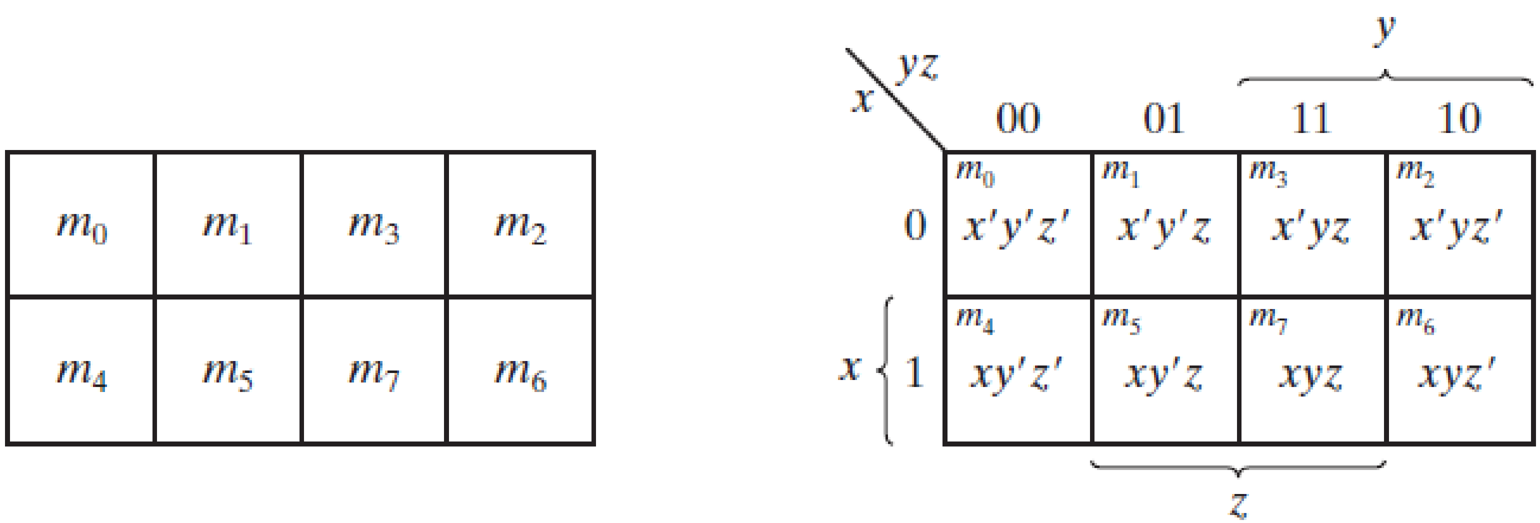
The given expression in terms of maxterms is $f=\pi(0,1,3)$.



F=AB'

Three-Variable K-Map

- There are eight Minterms or Maxterms for three binary variables; therefore, the map consists of eight squares.
- Note that the Minterms or Maxterms are arranged in gray code manner. The characteristic of this sequence is that only one bit changes in value from one adjacent column to the next.
- The map drawn in part (b) is marked with numbers in each row and each column, to show the relationship between the squares and the three variables.
- A three-variable K-map is shown in Fig.



m_0 m_1 m_3 m_2
m_4 m_5 m_7 m_6
y
yz
x
00 01 11 10
0
m_0 $x'y'z'$
m_1 $x'y'z$
m_3 $x'yz$
m_2 $x'yz'$
x 1
m_4 $xy'z'$
m_5 $xy'z$
m_7 xyz
m_6 xyz'
z

(a) (b)

Example: Simplify the Boolean functions

***F*(*A*, *B*, *C*) = ∑(0, 4)**

A \ BC	00	01	11	10
0	1	0	0	0
1	1	0	0	0

$$F(A,B,C) = \overline{B}\,\overline{C}$$

***F*(*A*, *B*, *C*) = ∑(1, 3)**

A \ BC	00	01	11	10
0	0	1	1	0
1	0	0	0	0

$$F(A,B,C) = \overline{A}\,C$$

***F*(*A*, *B*, *C*) = ∑(4, 5)**

A \ BC	00	01	11	10
0	0	0	0	0
1	1	1	0	0

$$F(A,B,C) = A\,\overline{B}$$

Example: Simplify the Boolean functions

$F(A, B, C) = \sum(0, 2)$

A \ BC	00	01	11	10
0	1	0	0	1
1	0	0	0	0

$$F(A,B,C) = \overline{A}\,\overline{C}$$

$F(A, B, C) = \sum(4, 6)$

A \ BC	00	01	11	10
0	0	0	0	0
1	1	0	0	1

$$F(A,B,C) = A\,\overline{C}$$

Example: Simplify the Boolean functions

$F(A, B, C) = \sum(0, 1, 2, 3)$

A \ BC	00	01	11	10
0	1	1	1	1
1	0	0	0	0

$$F(A,B,C) = \overline{A}$$

$F(A, B, C) = \sum(0, 1, 4, 5)$

A \ BC	00	01	11	10
0	1	1	0	0
1	1	1	0	0

$$F(A,B,C) = \overline{B}$$

$F(A, B, C) = \sum(0, 2, 4, 6)$

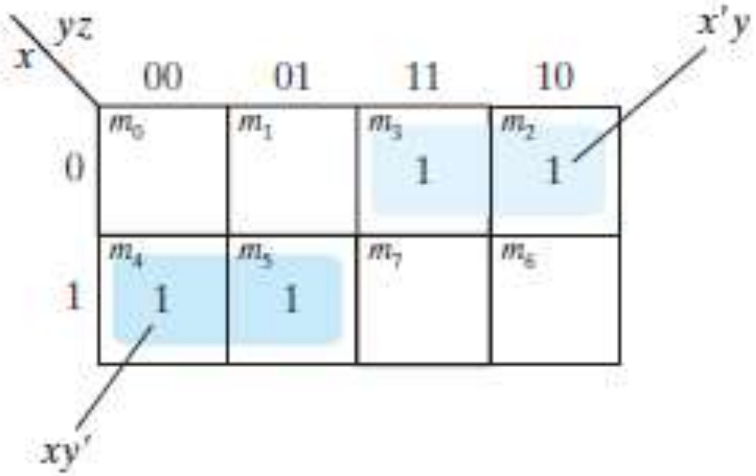
A \ BC	00	01	11	10
0	1	0	0	1
1	1	0	0	1

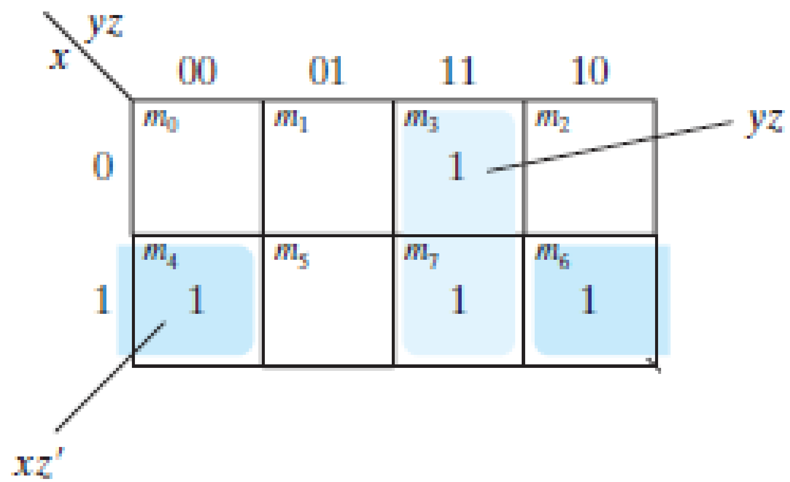
$$F(A,B,C) = \overline{C}$$

Example: Simplify the Boolean functions

$F(x, y, z) = \sum(2, 3, 4, 5)$

$F(x, y, z) = \sum(3, 4, 6, 7)$





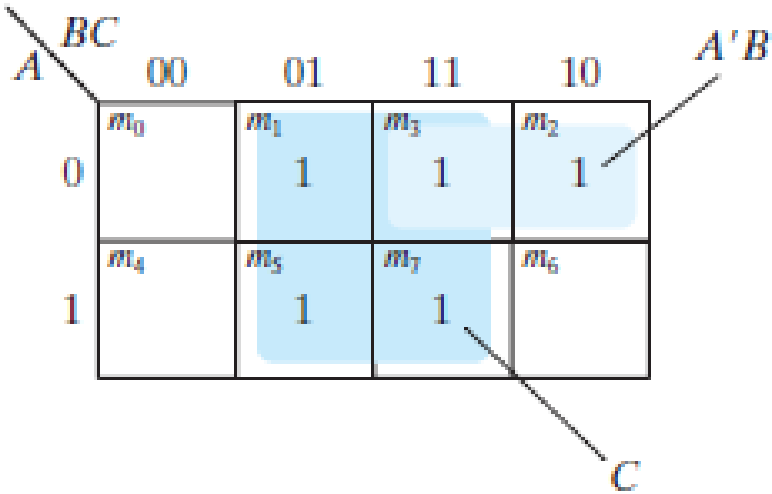
$F(x, y, z) = x'y + xy'$

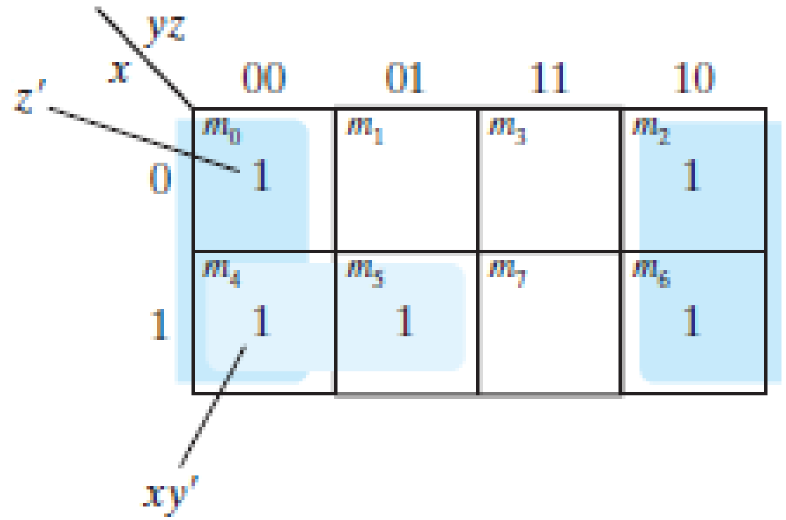
$F(x, y, z) = xz' + yz$

Example: Simplify the Boolean functions

$F(A, B, C) = \sum(1, 2, 3, 5, 7)$

$F(x, y, z) = \sum(0, 2, 4, 5, 6)$





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

$F(x, y, z) = z' + xy'$

Four-Variable K-Map

- There are 16 Minterms or Maxterms for 4 variables; therefore, the map consists of 16 squares.
- The map for Boolean functions of four binary variables (w, x, y, z) is shown in Fig.
- In Fig. (a) are listed the 16 Minterms and the squares assigned to each.
- In Fig. (b), the map is redrawn to show the relationship between the squares and the four variables.
- The rows and columns are numbered in a Gray code sequence, with only one digit changing value between two adjacent rows or columns.
- The minterm corresponding to each square can be obtained from the concatenation of the row number with the column number.

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)

$wx \backslash yz$	00	01	11	10
00	m_0 $w'x'y'z'$	m_1 $w'x'y'z$	m_3 $w'x'yz$	m_2 $w'x'yz'$
01	m_4 $w'xy'z'$	m_5 $w'xy'z$	m_7 $w'xyz$	m_6 $w'xyz'$
11	m_{12} $wxy'z'$	m_{13} $wxy'z$	m_{15} $wxyz$	m_{14} $wxyz'$
10	m_8 $wx'y'z'$	m_9 $wx'y'z$	m_{11} $wx'yz$	m_{10} $wx'yz'$

(b)

The combination of adjacent squares that is useful during the simplification process is easily determined from inspection of the four-variable map:

- One square represents one minterm, giving a term with four literals.
- Two adjacent squares represent a term with three literals.
- Four adjacent squares represent a term with two literals.
- Eight adjacent squares represent a term with one literal.
- Sixteen adjacent squares produce a function that is always equal to 1.

Example: Simplify the Boolean functions

$F(A, B, C, D) = \sum(13, 15)$

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	1	1	0
10	0	0	0	0

$$F(A,B,C,D) = ABD$$

$F(A, B, C, D) = \sum(5, 13)$

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	1	0	0
11	0	1	0	0
10	0	0	0	0

$$F(A,B,C,D) = B\overline{C}D$$

Example: Simplify the Boolean functions

$F(A, B, C, D) = \sum(4, 6)$

AB \ CD	00	01	11	10
00	0	0	0	0
01	1	0	0	1
11	0	0	0	0
10	0	0	0	0

$$F(A,B,C,D) = \overline{A}B\overline{D}$$

$F(A, B, C, D) = \sum(0, 8)$

AB \ CD	00	01	11	10
00	1	0	0	0
01	0	0	0	0
11	0	0	0	0
10	1	0	0	0

$$F(A,B,C,D) = \overline{B}\,\overline{C}\,\overline{D}$$

Example: Simplify the Boolean functions

$F(A, B, C, D) = \sum(4, 5, 6, 7)$

AB \ CD	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	0	0	0	0

$$F(A,B,C,D) = \overline{A}B$$

$F(A, B, C, D) = \sum(3, 7, 11, 15)$

AB \ CD	00	01	11	10
00	0	0	1	0
01	0	0	1	0
11	0	0	1	0
10	0	0	1	0

$$F(A,B,C,D) = CD$$

Example: Simplify the Boolean functions

***F*(*A*, *B*, *C*, *D*) = ∑(2, 3, 6, 7)**

AB \ CD	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	0	0	0	0
10	0	0	0	0

$$F(A, B, C, D) = \overline{A}C$$

***F*(*A*, *B*, *C*, *D*) = ∑(4, 6, 12, 14)**

AB \ CD	00	01	11	10
00	0	0	0	0
01	1	0	0	1
11	1	0	0	1
10	0	0	0	0

$$F(A, B, C, D) = B\overline{D}$$

Example: Simplify the Boolean functions

$F(A, B, C, D) = \sum(2, 3, 10, 11)$

AB \ CD	00	01	11	10
00	0	0	1	1
01	0	0	0	0
11	0	0	0	0
10	0	0	1	1

$$F(A,B,C,D) = \overline{B}\,C$$

$F(A, B, C, D) = \sum(0, 2, 8, 10)$

AB \ CD	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	0	0	0	0
10	1	0	0	1

$$F(A,B,C,D) = \overline{B}\,\overline{D}$$

Example: Simplify the Boolean functions

$F(A, B, C, D) = \sum(4, 5, 6, 7, 12, 13, 14, 15)$

AB \ CD	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	1	1	1	1
10	0	0	0	0

$$F(A,B,C,D) = B$$

$F(A, B, C, D) = \sum(0, 1, 2, 3, 8, 9, 10, 11)$

AB \ CD	00	01	11	10
00	1	1	1	1
01	0	0	0	0
11	0	0	0	0
10	1	1	1	1

$$F(A,B,C,D) = \overline{B}$$

Example: Simplify the Boolean functions

$F(A, B, C, D) = \sum(1, 3, 5, 7, 9, 11, 13, 15)$

AB \ CD	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	0	1	1	0
10	0	1	1	0

$$F(A,B,C,D) = D$$

$F(A, B, C, D) = \sum(0, 2, 4, 6, 8, 10, 12, 14)$

AB \ CD	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	1	0	0	1
10	1	0	0	1

$$F(A,B,C,D) = \overline{D}$$

Example: **Simplify the following Boolean functions, using four-variable maps:**

$F(A, B, C, D) = \sum(0, 1, 2, 3, 4, 6, 8, 9, 10, 11, 12, 14)$

$F(A, B, C, D) = \sum(1, 3, 4, 5, 6, 7, 8, 9, 12, 13, 14, 15)$

AB \ CD	00	01	11	10
00	1	1	1	1
01	1	0	0	1
11	1	0	0	1
10	1	1	1	1

AB \ CD	00	01	11	10
00	0	1	1	0
01	1	1	1	1
11	1	1	1	1
10	1	1	0	0

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

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

Example: Simplify the following Boolean functions, using four-variable maps:

$F(A, B, C, D) = \sum(0, 1, 2, 3, 5, 7, 8, 9, 10, 12, 13)$

$F(P, Q, R, S) = \sum(0, 1, 3, 4, 5, 6, 7, 13, 15)$

AB \ CD	00	01	11	10
00	1	1	1	1
01	0	1	1	0
11	1	1	0	0
10	1	1	0	1

PQ \ RS	00	01	11	10
00	1	1	1	0
01	1	1	1	1
11	0	1	1	0
10	0	0	0	0

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

$F(P, Q, R, S) = P'R' + P'S + P'Q + QS$

Example: **Simplify the following Boolean functions, using four-variable maps:**

$F(w, x, y, z) = \sum(1, 4, 5, 6, 12, 14, 15)$

$wx \backslash yz$	00	01	11	10
00	0	1	0	0
01	1	1	0	1
11	1	0	1	1
10	0	0	0	0

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

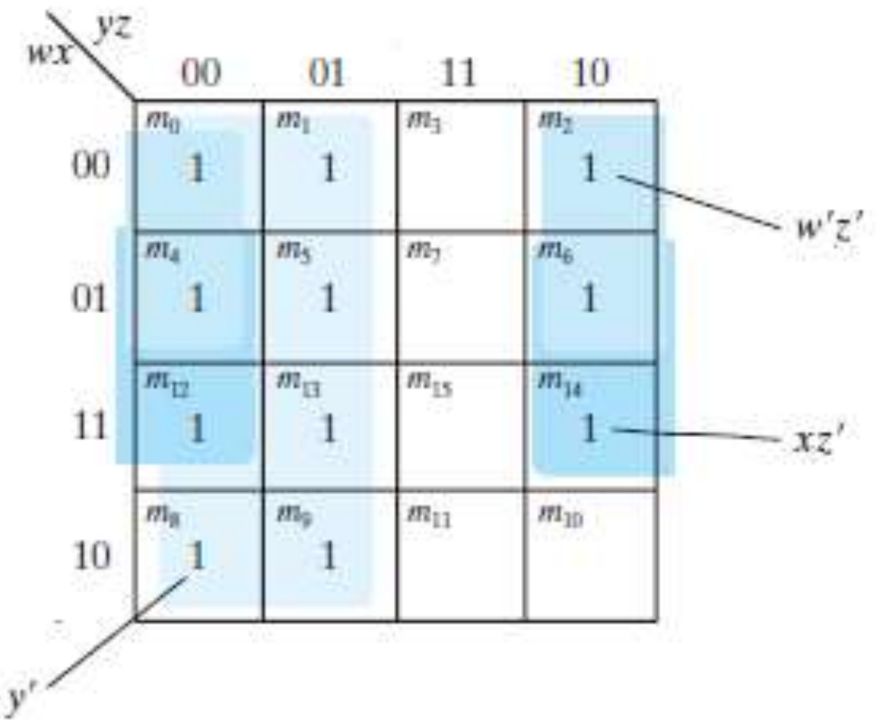
$F(w, x, y, z) = \sum(0, 2, 4, 6, 9, 10, 11, 12, 14)$

$wx \backslash yz$	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	1	0	0	1
10	0	1	1	1

$F(w, x, y, z) = w'z' + xz' + yz' + wx'z$

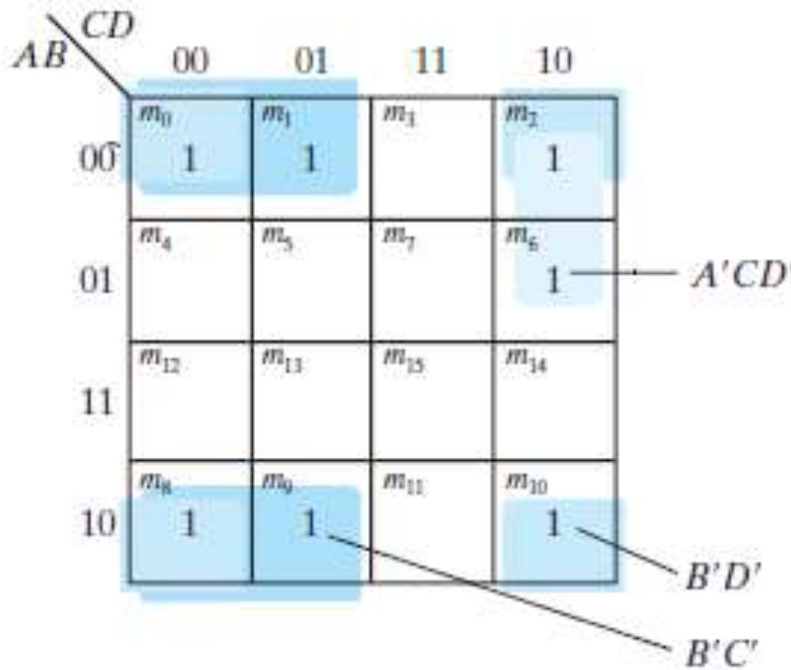
Example: Simplify the Boolean functions

***F*(*w*, *x*, *y*, *z*) = ∑ (0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14)**



F*(*w*, *x*, *y*, *z*) = *y’* + *w’z’* + *xz’

***F*(*A*, *B*, *C*, *D*) = ∑ (0, 1, 2, 6, 8, 9, 10)**



F*(*A*, *B*, *C*, *D*) = *B’D’* + *B’C’* + *A’CD’

DON’T-CARE CONDITIONS

- In some applications the function is not specified for certain combinations of the variables.
- A don’t-care condition is a combination of variables whose logical value is not specified.
- To distinguish the don’t-care condition from 1’s and 0’s, an X is used.
- Thus, an X inside a square in the map indicates that we don’t care whether the value of 0 or 1 is assigned to *Function* for the particular minterm.

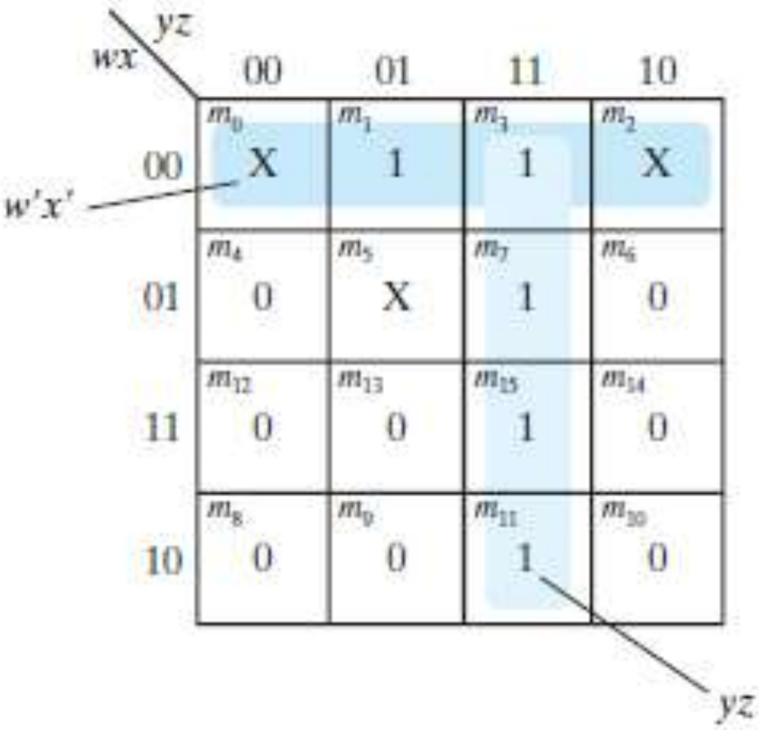
- These don’t-care conditions can be used on a map to provide further simplification of the Boolean expression.
- In choosing adjacent squares to simplify the function in a map, the don’t-care minterms may be assumed to be either 0 or 1.
- When simplifying the function, we can choose to include each don’t-care minterm with either the 1’s or the 0’s, depending on which combination gives the simplest expression.

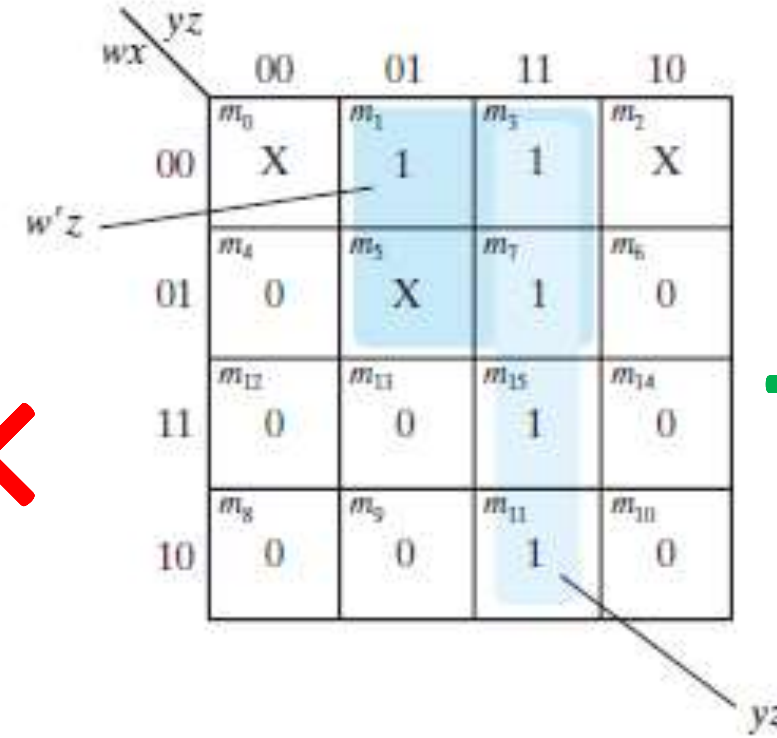
Example: Simplify the Boolean function

$F\,(w, x, y, z) = \sum(1, 3, 7, 11, 15)$

which has the don't-care conditions

$d\,(w, x, y, z) = \sum(0, 2, 5)$





$F = yz + w'x'$

$F = yz + w'z$

Example: Simplify the following Boolean function *F*, together with the don't-care conditions *d*.

$F(A, B, C, D) = \sum(1, 5, 6, 12, 13, 14)$

$d(A, B, C, D) = \sum(2, 4)$

AB \ CD	00	01	11	10
00	0	1	0	X
01	X	1	0	1
11	1	1	0	1
10	0	0	0	0

$F(A, B, C, D) = BC' + BD' + A'C'D$

$F(A, B, C, D) = \sum(4, 5, 7, 12, 14, 15)$

$d(A, B, C, D) = \sum(3, 8, 10)$

AB \ CD	00	01	11	10
00	0	0	X	0
01	1	1	1	0
11	1	0	1	1
10	X	0	0	X

$F(A, B, C, D) = AD' + A'BC' + BCD$

Example: **Simplify the following Boolean function *F*, together with the don't-care conditions *d*.**

$F(A, B, C, D) = \sum(1, 3, 5, 7, 9, 15)$

$d(A, B, C, D) = \sum(4, 6, 12, 13)$

AB \ CD	00	01	11	10
00	0	1	1	0
01	X	1	1	X
11	X	X	1	0
10	0	1	0	0

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

$F(A, B, C, D) = \sum(1, 3, 4, 5, 8, 10, 11, 15)$

$d(A, B, C, D) = \sum(0, 2, 7, 14)$

AB \ CD	00	01	11	10
00	X	1	1	X
01	1	1	X	0
11	0	0	1	X
10	1	0	1	1

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