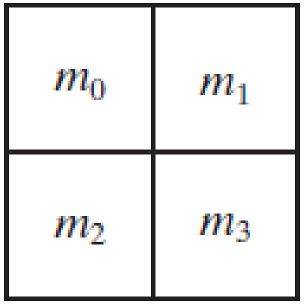
K-maps - Two, Three, and Four Variable K-maps, Don’t-Care Conditions

THE MAP METHOD

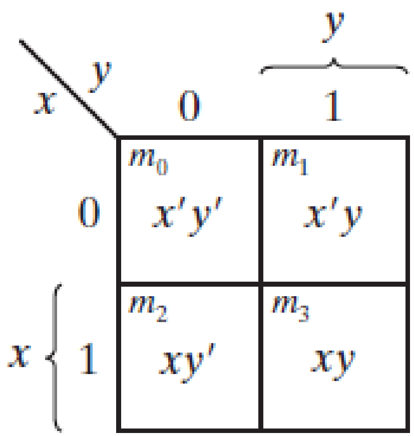
- The map method provides a simple, straightforward procedure for minimizing Boolean functions.
- This method may be regarded as a pictorial form of a truth table.
- The map method is also known as the *Karnaugh map* or *K-map*.
- The simplified expressions produced by the map are always in one of the two standard forms: sum of products or product of sums.

Two-Variable K-Map

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



(a)



(b)

- 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; no 0s.
- Groups can be formed only at right angles (horizontal or vertical); diagonal groups are not allowed.
- The number of 1s in a group must be a power of 2 – even if it contains a single 1.
- The groups must be made as large as possible.

- Each cell containing a one 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*) = ∑(0, 1)**

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

F*(*x*, *y*) = *x'

***F*(*x*, *y*) = ∑(2, 3)**

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

F*(*x*, *y*) = *x

***F*(*x*, *y*) = ∑(0, 2)**

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

F*(*x*, *y*) = *y'

***F*(*x*, *y*) = ∑(1, 3)**

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

F*(*x*, *y*) = *y

Example: Simplify the Boolean functions

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

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

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

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

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

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

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

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

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

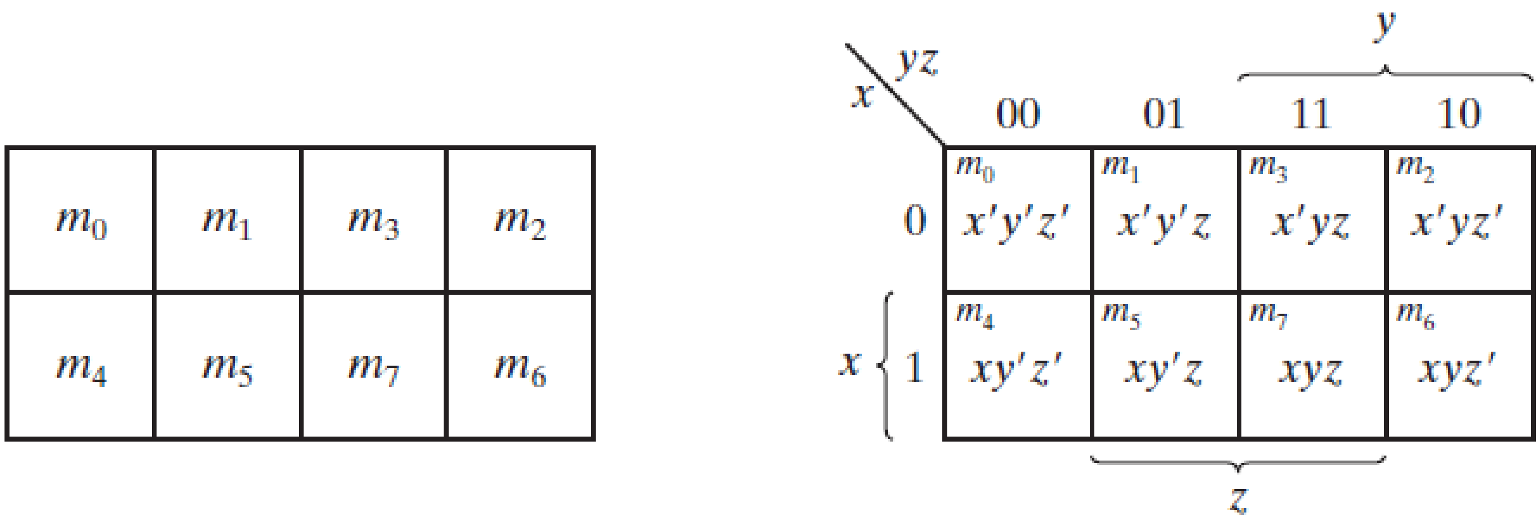
$F(x, y) = \sum(0, 1, 2)$

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

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

Three-Variable K-Map

- A three-variable K-map is shown in Fig.
- There are eight minterms for three binary variables; therefore, the map consists of eight squares.
- Note that the minterms are arranged, not in a binary sequence, but in a sequence similar to the Gray code.
- 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.


m_0 m_1 m_3 m_2
m_4 m_5 m_7 m_6
yz
x
y
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

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

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

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

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

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

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

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

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*) = ∑(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*) = ∑(0, 1, 4, 5)**

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

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

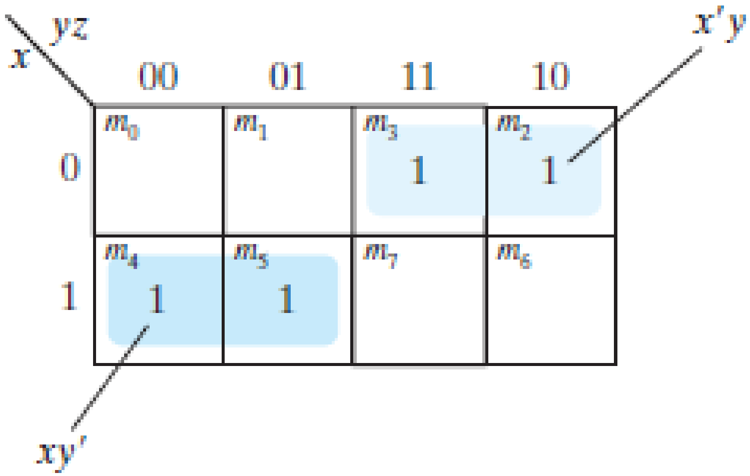
***F*(*A*, *B*, *C*) = ∑(0, 2, 4, 6)**

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

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

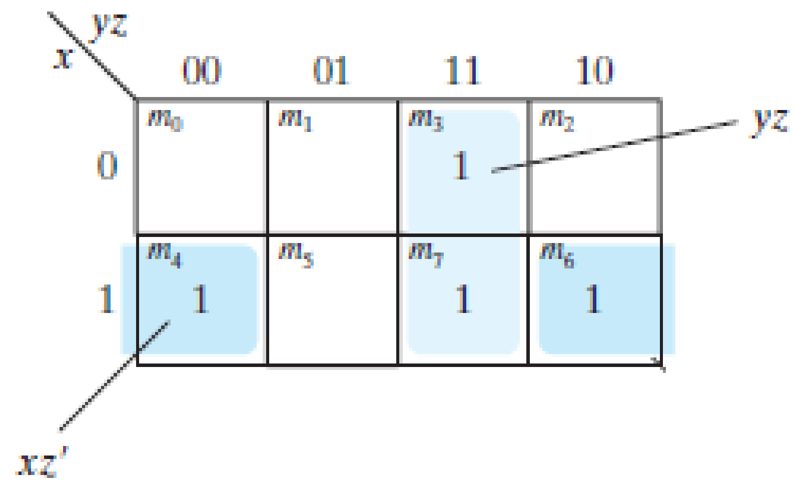
Example: Simplify the Boolean functions

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



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

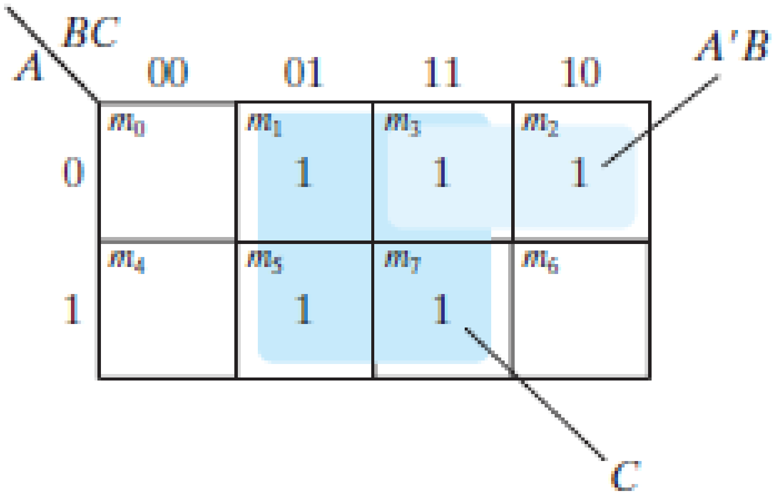
$F(x, y, z) = \sum(3, 4, 6, 7)$



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

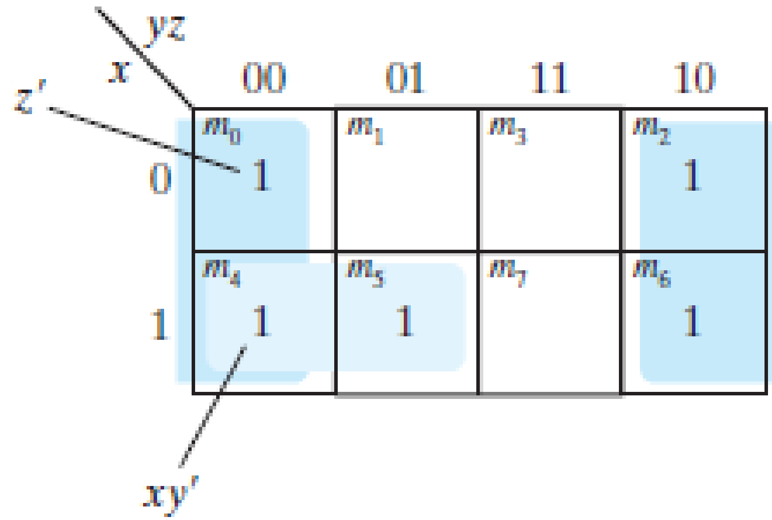
Example: Simplify the Boolean functions

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



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

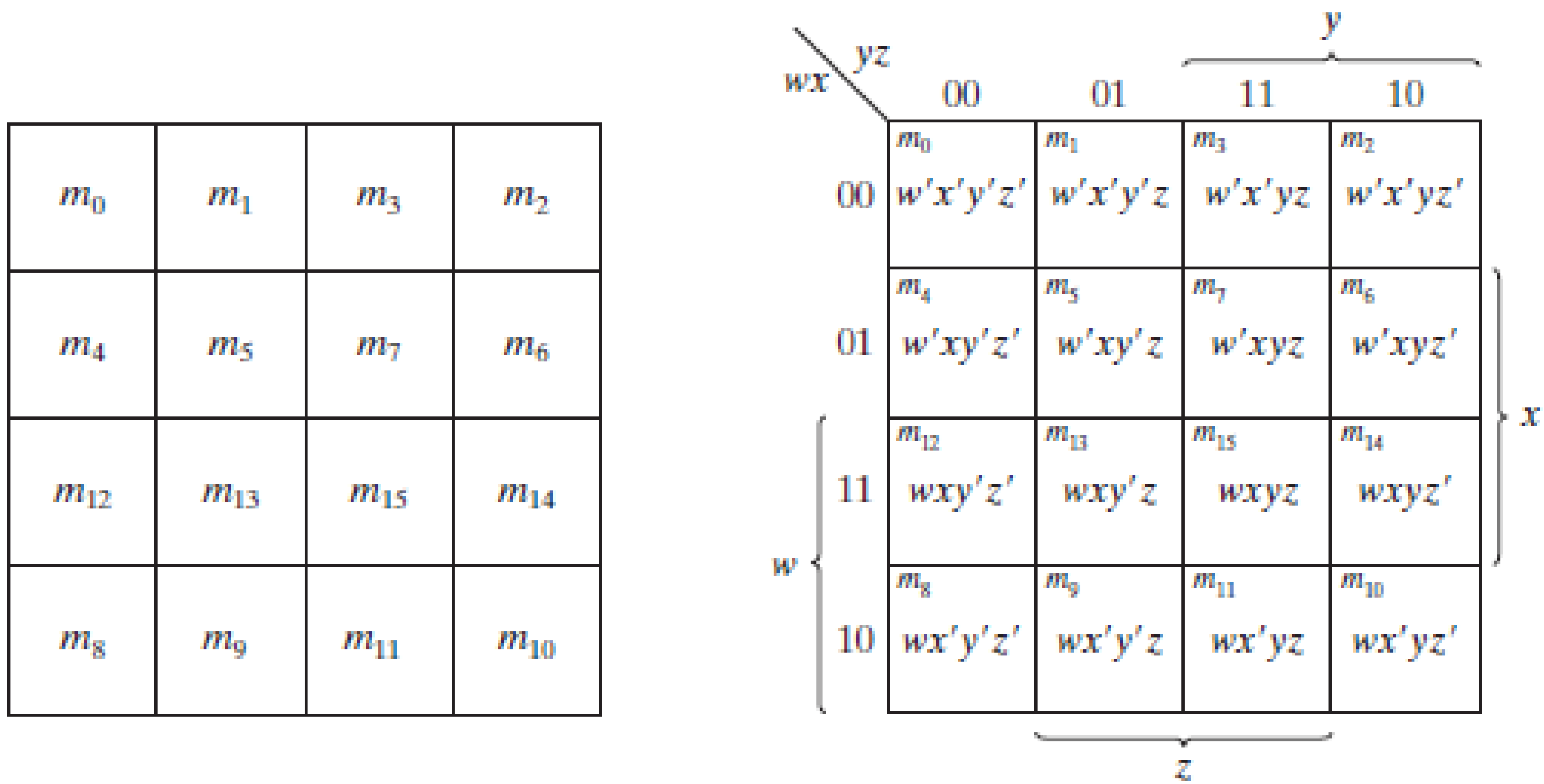
$F(x, y, z) = \sum(0, 2, 4, 5, 6)$



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

Four-Variable K-Map

- 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.



m0 m1 m3 m2
m4 m5 m7 m6
m12 m13 m15 m14
m8 m9 m11 m10
yz
wx
y
00 01 11 10
m0 w'x'y'z'
m1 w'x'y'z
m3 w'x'yz
m2 w'x'yz'
m4 w'xy'z'
m5 w'xy'z
m7 w'xyz
m6 w'xyz'
m12 wxy'z'
m13 wxy'z
m15 wxyz
m14 wxyz'
m8 wx'y'z'
m9 wx'y'z
m11 wx'yz
m10 wx'yz'
x
w
z

(a) (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

$F(A, B, C, D) = \sum(2, 3, 6, 7)$

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

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

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

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

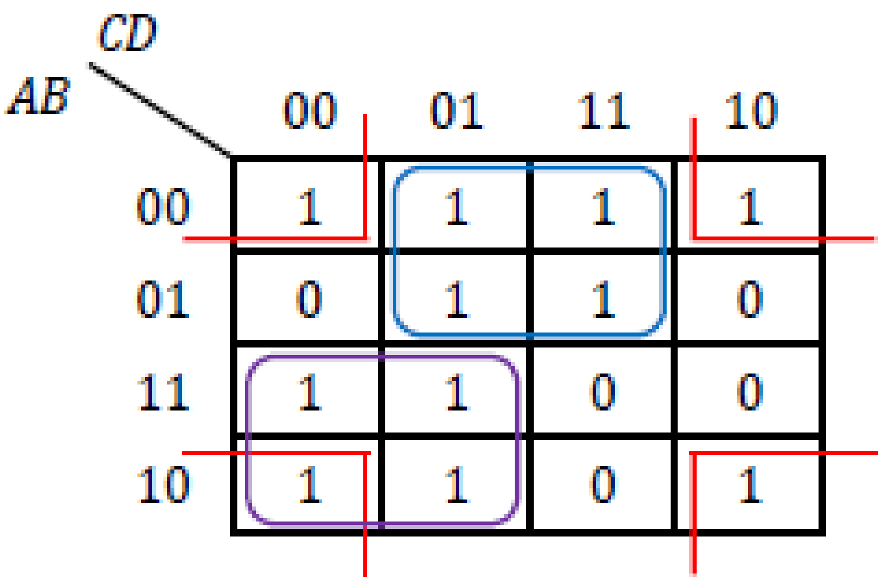
AB \ CD	00	01	11	10
00	0	1	1	0
01	1	1	1	1
11	1	1	1	1
10	1	1	0	0

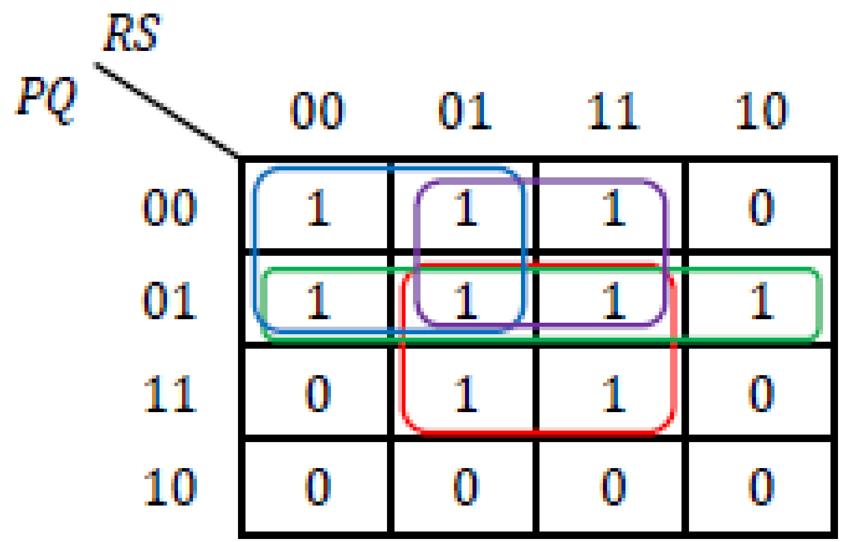
$F(A, B, C, D) = B' + D'$

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

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

$F(A, B, C, D) = \sum(0, 1, 2, 3, 5, 7, 8, 9, 10, 12, 13)$ $F(P, Q, R, S) = \sum(0, 1, 3, 4, 5, 6, 7, 13, 15)$





$F(A, B, C, D) = B'D' + A'D + AC'$ $F(P, Q, R, S) = P'R' + P'S + P'Q + QS$

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

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

wx \ 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 \ 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 following Boolean functions, using four-variable maps:

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

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

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

Here *BD* is a **Redundant group**.

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

$F(A, B, C, D) = \sum(0, 2, 4, 5, 6, 7, 8, 10, 13, 15)$

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

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

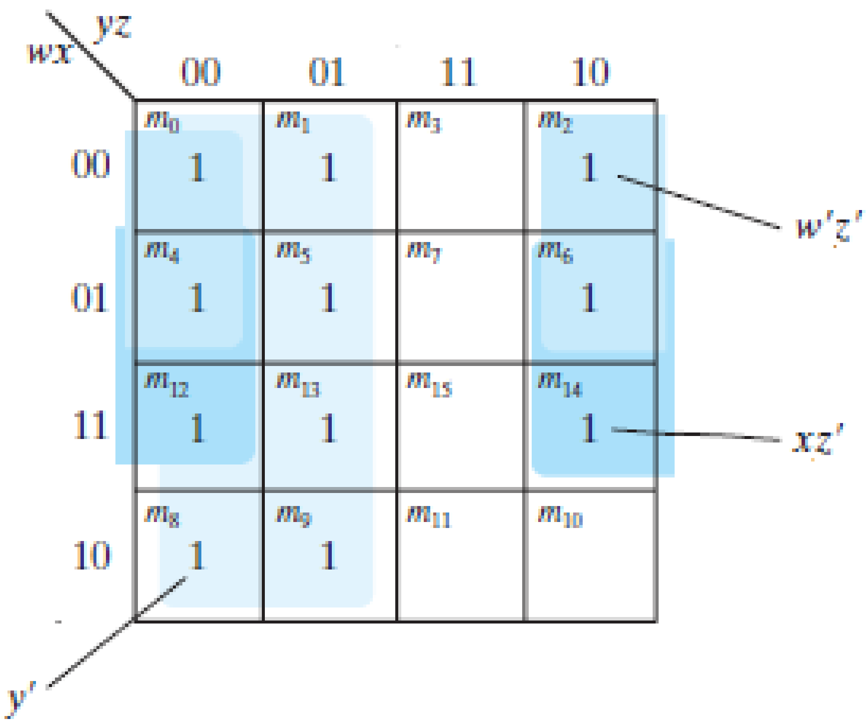
or

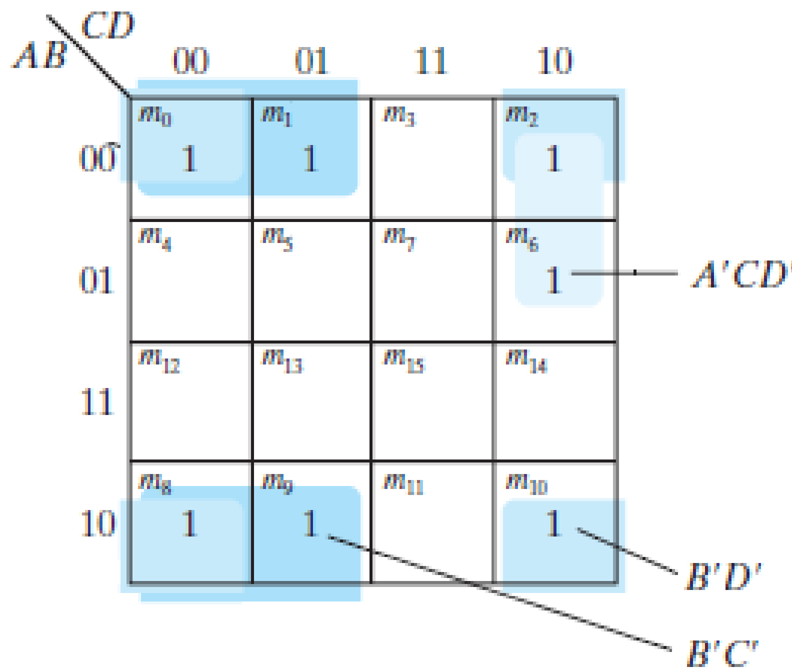
$F(A, B, C, D) = B'D' + BD + A'D'$

Example: Simplify the Boolean functions

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

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





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

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

Prime Implicants

- In choosing adjacent squares in a map, we must ensure that

1. all the minterms of the function are covered when we combine the squares,
2. the number of terms in the expression is minimized, and
3. there are no redundant terms (i.e., minterms already covered by other terms).

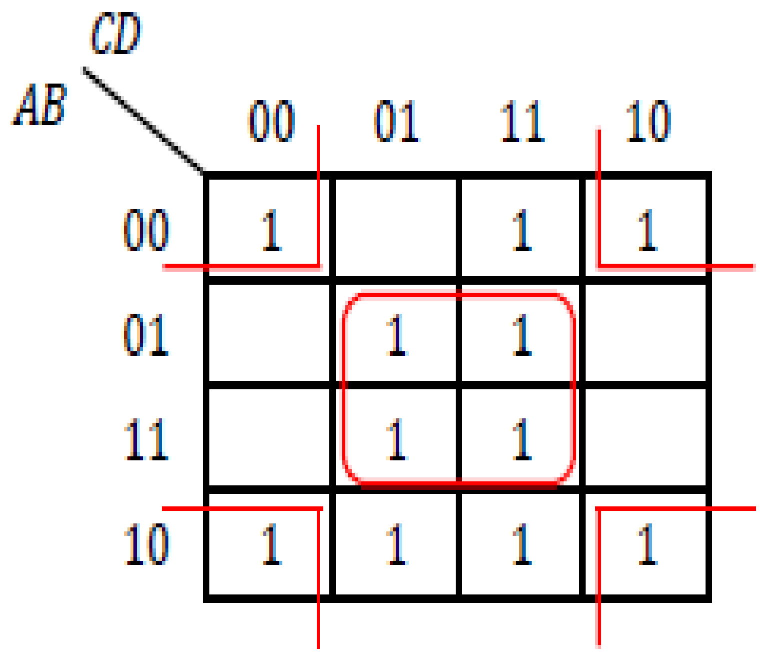
- Sometimes there may be two or more expressions that satisfy the simplification criteria.

- The procedure for combining squares in the map may be made more systematic if we understand the meaning of two special types of terms.
- A *prime implicant* is a product term obtained by combining the maximum possible number of adjacent squares in the map.
- If a minterm in a square is covered by only one prime implicant, that prime implicant is said to be *essential*.

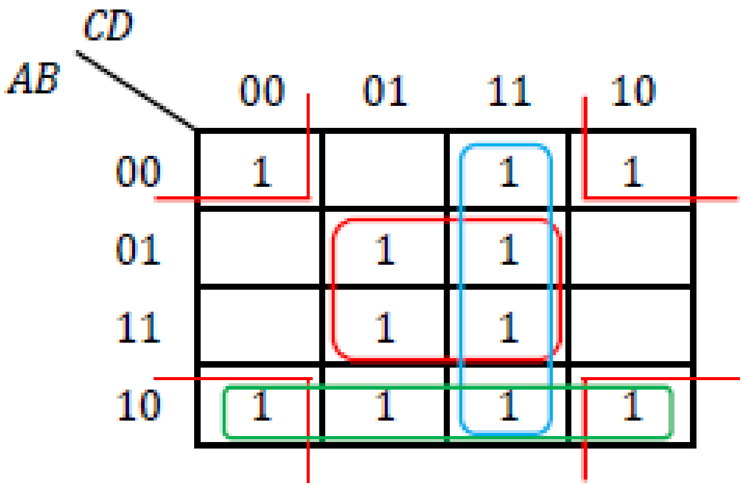
Consider the following four-variable Boolean function:

$F(A, B, C, D) = \sum(0, 2, 3, 5, 7, 8, 9, 10, 11, 13, 15)$

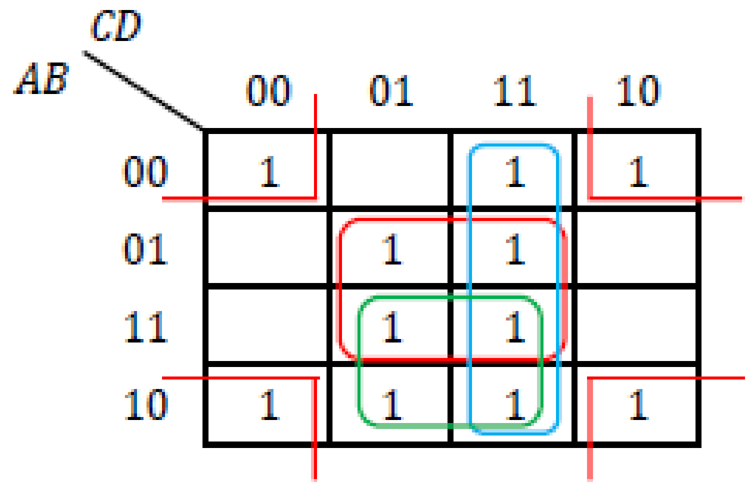
- There is only one way to include minterm m_0 within four adjacent squares. These four squares define the term $B'D'$.
- Similarly, there is only one way that minterm m_5 can be combined with four adjacent squares, and this gives the second term BD.
- The two essential prime implicants cover eight minterms.



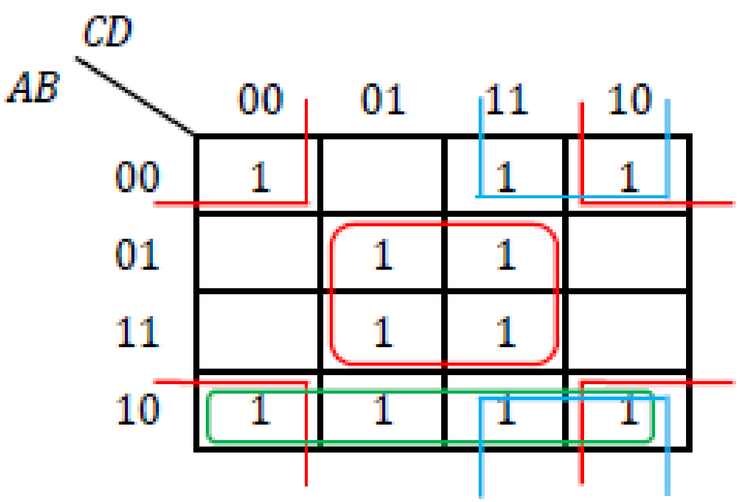
Essential prime implicants $B'D'$ and BD



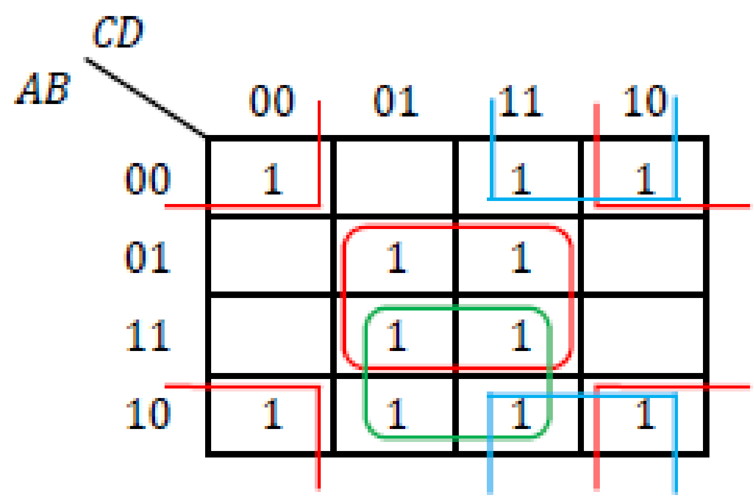
$F(A, B, C, D) = B'D' + BD + CD + AB'$



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



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



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

- Figure shows all possible ways that the three minterms (m_3, m_9, and m_{11}) can be covered with prime implicants.
- Minterm m_3 can be covered with either prime implicant CD or prime implicant $B'C$.
- Minterm m_9 can be covered with either AB' or AD.
- Minterm m_{11} is covered with any one of the four prime implicants.
- The simplified expression is obtained from the logical sum of the two essential prime implicants and any two prime implicants that cover minterms m_3, m_9, and m_{11}.
- There are four possible ways that the function can be expressed with four product terms of two literals each:

$$\begin{aligned} F &= B'D' + BD + CD + AB' \\ &= B'D' + BD + CD + AD \\ &= B'D' + BD + B'C + AB' \\ &= B'D' + BD + B'C + AD \end{aligned}$$

Find all the prime implicants for the following Boolean functions, and determine which are essential:

***F* (*A*, *B*, *C*, *D*) = ∑ (0, 4, 5, 10, 11, 13, 15)**

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

F* = *A'C'D'* + *AB'C* + *BC'D* + *ABD

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

F* = *A'C'D'* + *AB'C* + *BC'D* + *ACD

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

F* = *A'C'D'* + *AB'C* + *A'BC'* + *ABD

- Essential prime implicants *A'C'D'* and *AB'C*
- Prime implicants *BC'D*, *ABD*, *ACD* and *A'BC'*

DON'T-CARE CONDITIONS

- In some applications the function is not specified for certain combinations of the variables.
- As an example, the four-bit binary code for the decimal digits has six combinations that are not used and consequently are considered to be unspecified.

- Functions that have unspecified outputs for some input combinations are called *incompletely specified functions*.
- For this reason, it is customary to call the unspecified minterms of a function *don’t-care conditions*.
- These don’t-care conditions can be used on a map to provide further simplification of the Boolean expression.

- A don’t-care minterm 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 *F* for the particular minterm.

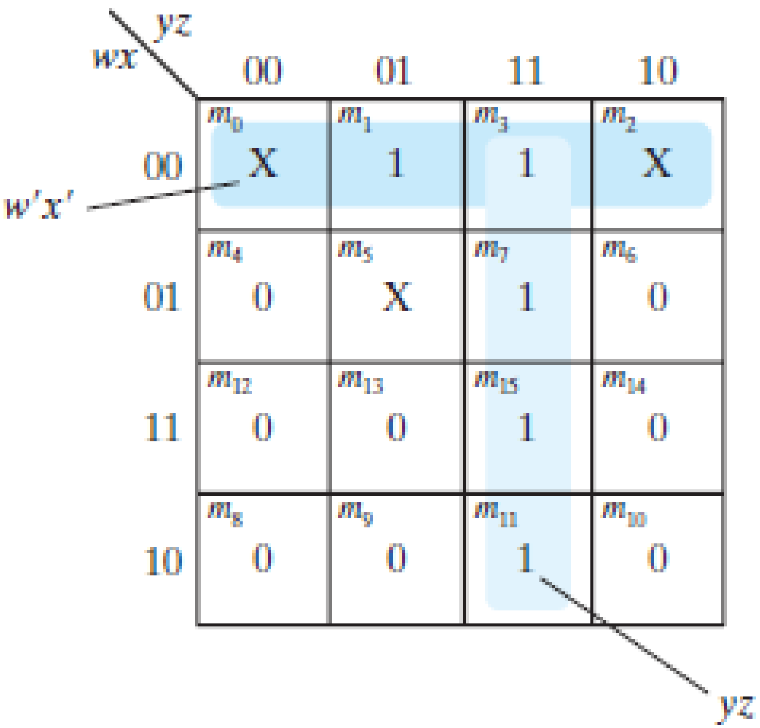
- In choosing adjacent squares to simplify the function in a map, the don’t-care minterms may be assumed to be either 0 or 1.
- When simplifying the function, we can choose to include each don’t-care minterm with either the 1’s or the 0’s, depending on which combination gives the simplest expression.

Example: Simplify the Boolean function

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

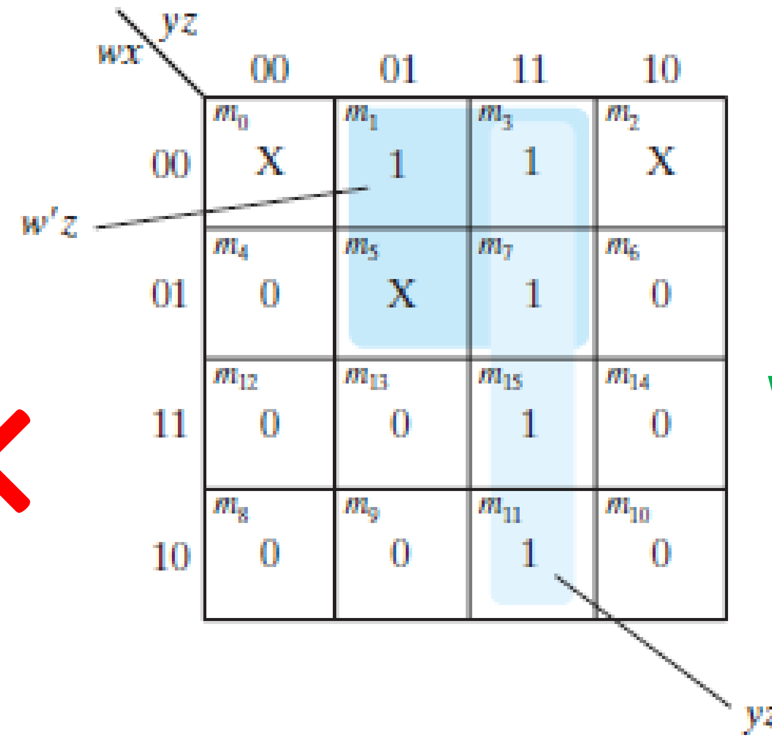
which has the don't-care conditions

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



✗

$F = yz + w'x'$



√

$F = yz + w'z$

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

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

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

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

PRODUCT-OF-SUMS SIMPLIFICATION

Example: Simplify the following Boolean functions in POS form.

$F(x, y, z) = \prod(0, 1, 3, 7)$

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

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

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

$F(A, B, C) = \prod(0, 1, 2, 3, 4, 7)$

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

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

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

Example: Simplify the following Boolean functions in POS form.

$F(w, x, y, z) = \prod(0, 4, 6, 7, 8, 12, 13, 14, 15)$

$F(A, B, C, D) = \prod(2, 8, 9, 10, 11, 12, 14)$

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

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

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

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

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

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