Ollscoil na Gaillimhe
University of Galway

CT101 Computing Systems

Dr. Bharathi Raja Chakravarthi
Lecturer-above-the-bar
Email: bharathi.raja@universityofgalway.ie

University
ofGalway.ie

Ollscoil na Gaillimhe
University of Galway

Gate-Level Minimization

Gate-Level Minimization

- Gate-level minimization is the design task of finding an optimal gate-level implementation of the Boolean functions describing a digital circuit.
- This task is well understood but is difficult to execute by manual methods when the logic has more than a few inputs.
- Computer-based logic synthesis tools can minimize a large set of Boolean equations efficiently and quickly.

The Map Methods

- The complexity of the digital logic gates is directly related to the complexity of algebraic expression.
- The map method provides a simple 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

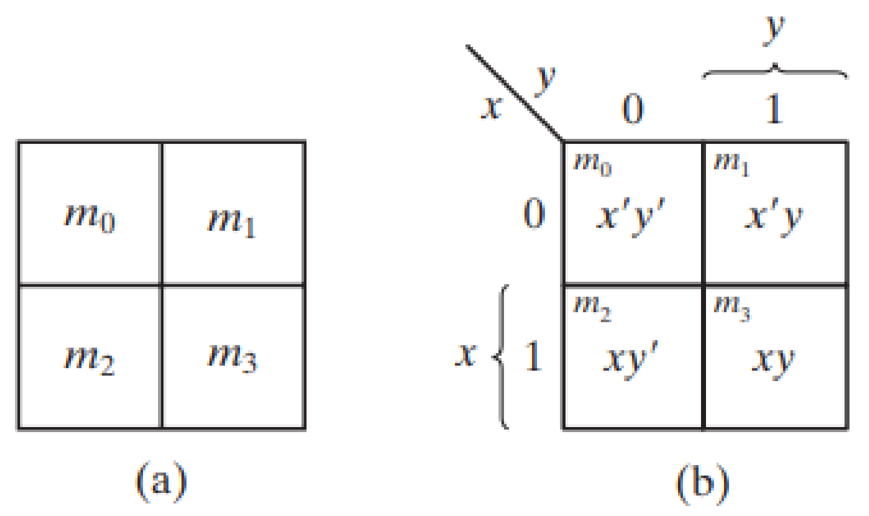
- A K-map is a diagram made up of squares, with each square representing one minterm of the function that is to be minimized.

- The map presents a visual diagram of all possible ways a function may be expressed in standard form.

- The simplified expressions produced by the map are always in one of the two standard forms:
    - sum of products or
    - products of sums

Two-Variable K-Map

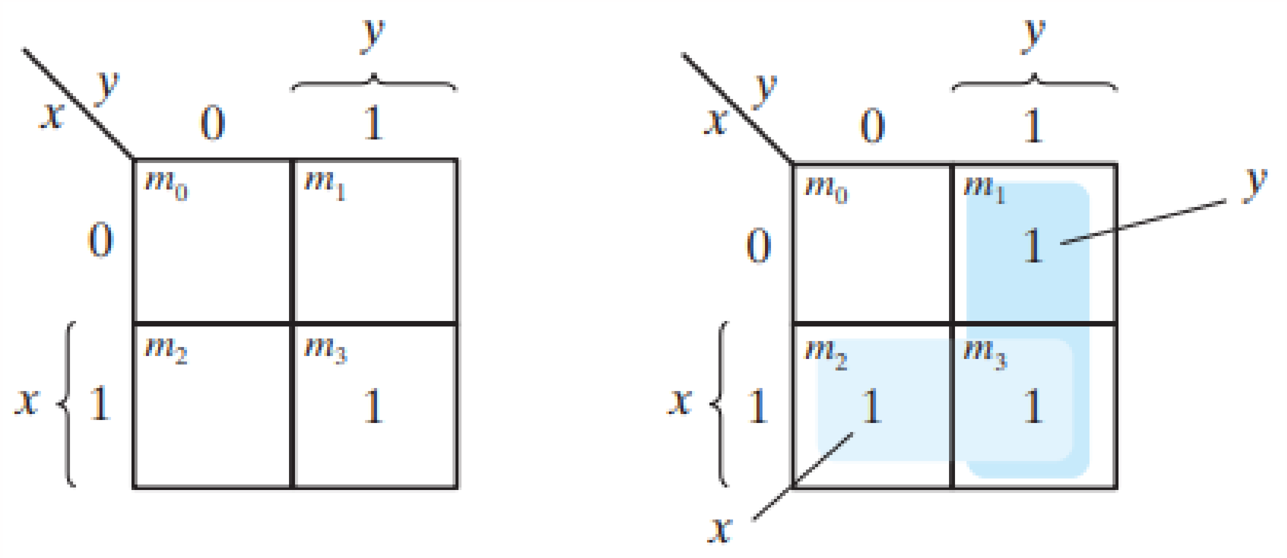
- The two-variable map is shown here.
- Four minterms for two variables.
- (b) shows the relationship between the squares and the two variables x and y .
- **0** and **1** designate the values of variables.
- Variable x appears primed in row 0 and unprimed in row 1.
- Similarly, y appears primed in column 0 and unprimed in column 1.



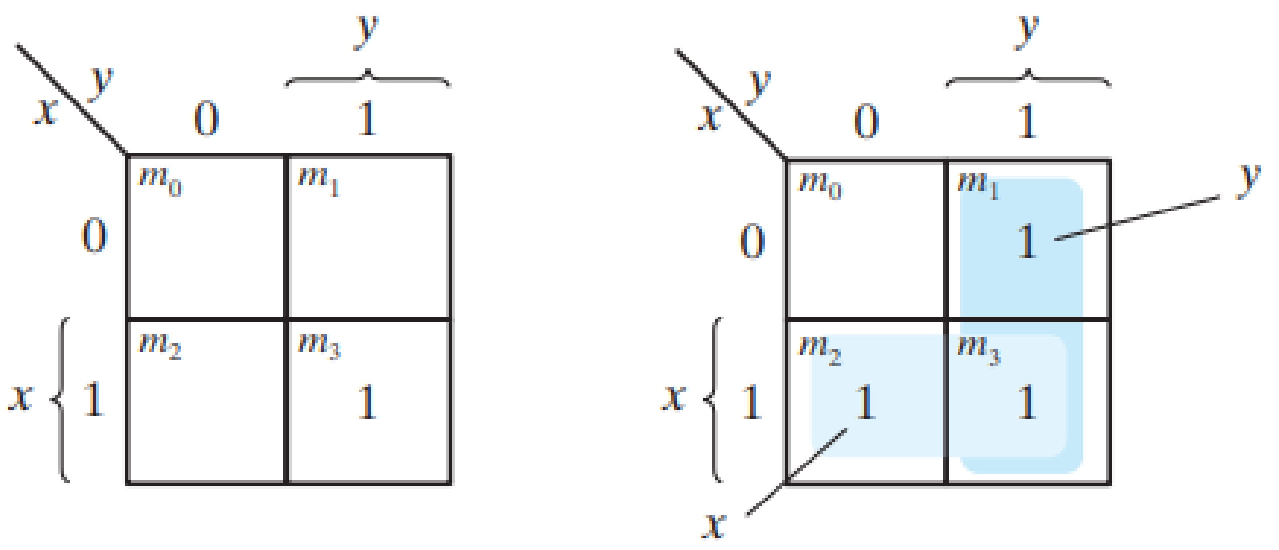
Two-Variable K-Map

- The two-variable map becomes another useful way to represent any one of the 16 Boolean functions of two variables.

- Example, the function xy is shown Fig. (a).



Two-Variable K-Map



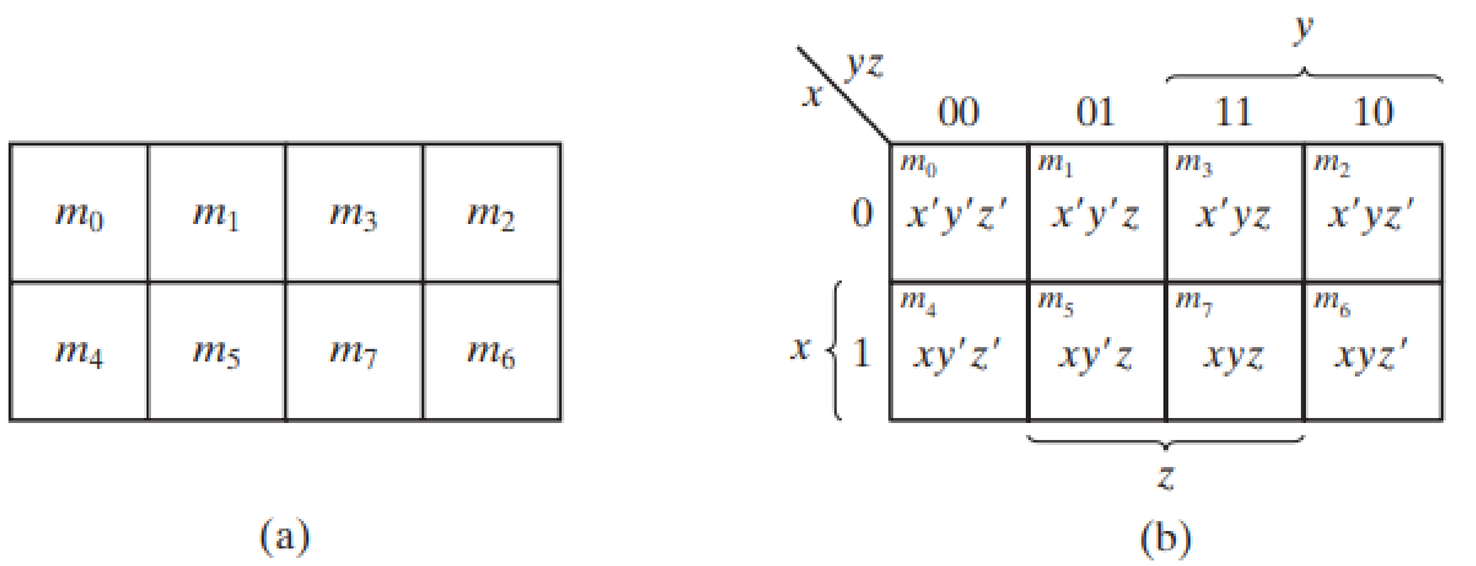
(a) xy (b) $x + y$

- Since **$xy = m_3$**, 1 is placed inside the square m_3.
- x + y function is represented in the map of Fig. (b) by three squares marked with 1's.
- These squares are found from the minterms of the function:

$$\mathbf{m_1 + m_2 + m_3 = x'y + xy' + xy = x + y}$$

Three-Variable K-Map

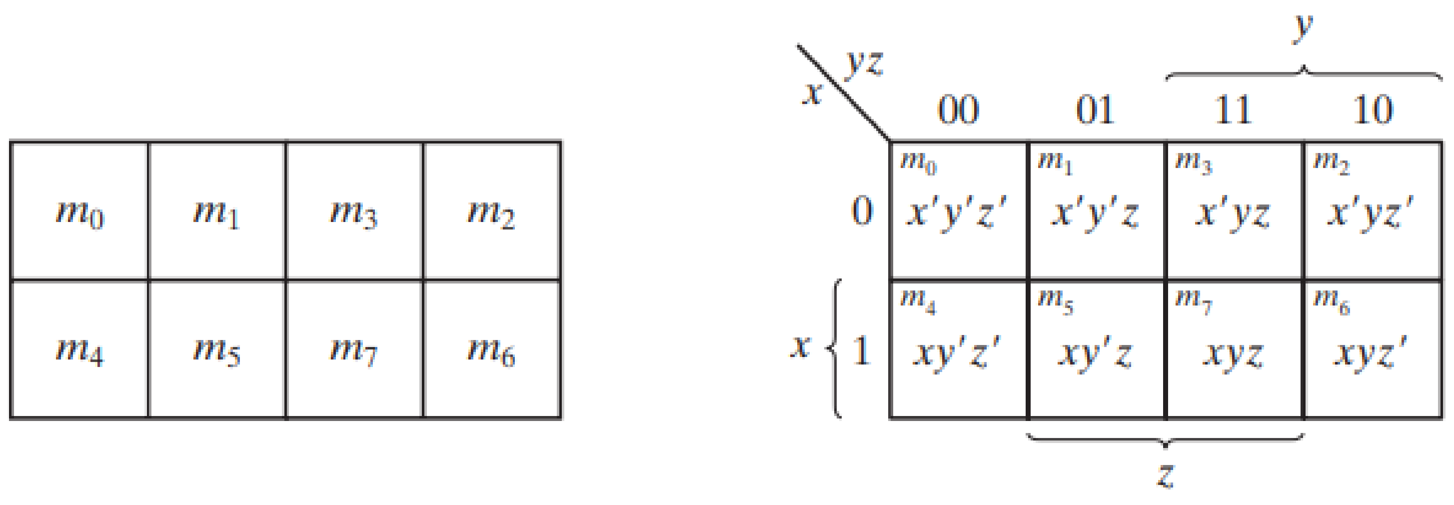
- A three-variable K-map is shown here.



- Eight minterms for three binary variables; the map consists of eight squares.

- Only one bit changes in value from one adjacent column to the next.

Three-Variable K-Map



(a) (b)

- 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.

- For example, the square assigned to m_5:
  row - 1 and column - 01.

- When these two numbers are concatenated, they give the binary number **101**, whose decimal equivalent is **5**.

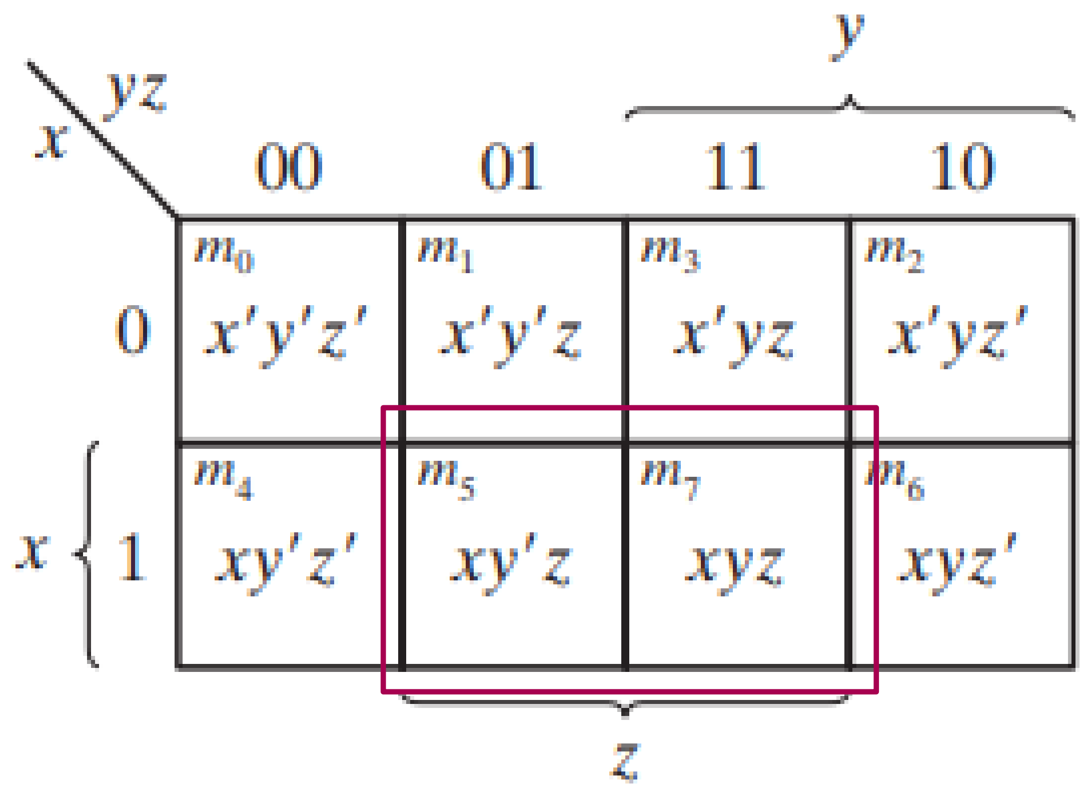
Three-Variable K-Map

- Another way of m_5 = xyz, consider it to be in the row marked x and the column belonging to yz (column 01).

- Variable appears unprimed in former four squares and primed in the latter.

- To understand the usefulness of the map in simplifying Boolean functions, we must recognize the basic property possessed by adjacent squares:

  - **Any two adjacent squares in the map differ by only one variable, which is primed in one square and unprimed in the other.**
  - For example, m_5 and m_7 lie in two adjacent squares.



Three-Variable K-Map

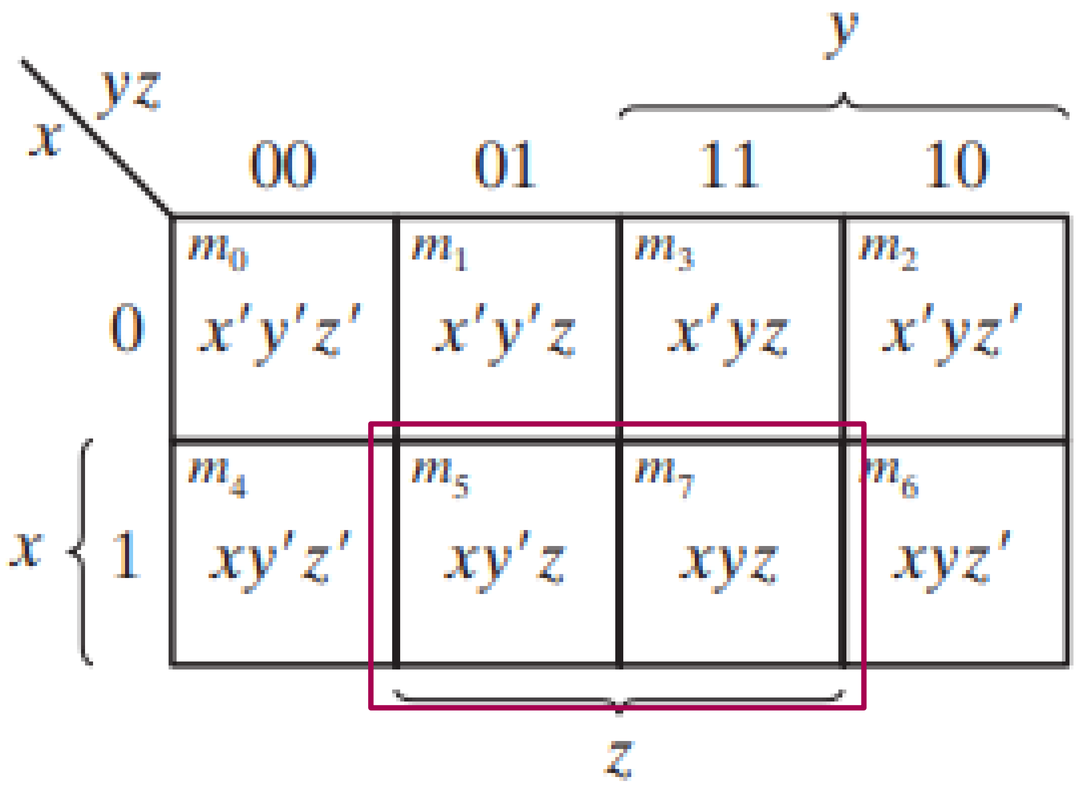
- Let's consider two squares, m_5, and m_7, where "y" is primed in m_5 and unprimed in m_7, but the other two variables are the same.

- According to the postulates of Boolean algebra, if we have the sum of two minterms in adjacent squares, we can simplify it to a single product term with only two literals.

- To illustrate this concept, let's take the sum of the two adjacent squares, m_5 and m_7:

$$\mathbf{m_5 + m_7 = xy'z + xyz = xz(y' + y) = xz}$$

- As we can see, we can simplify the sum of m_5 and m_7 to just xz.



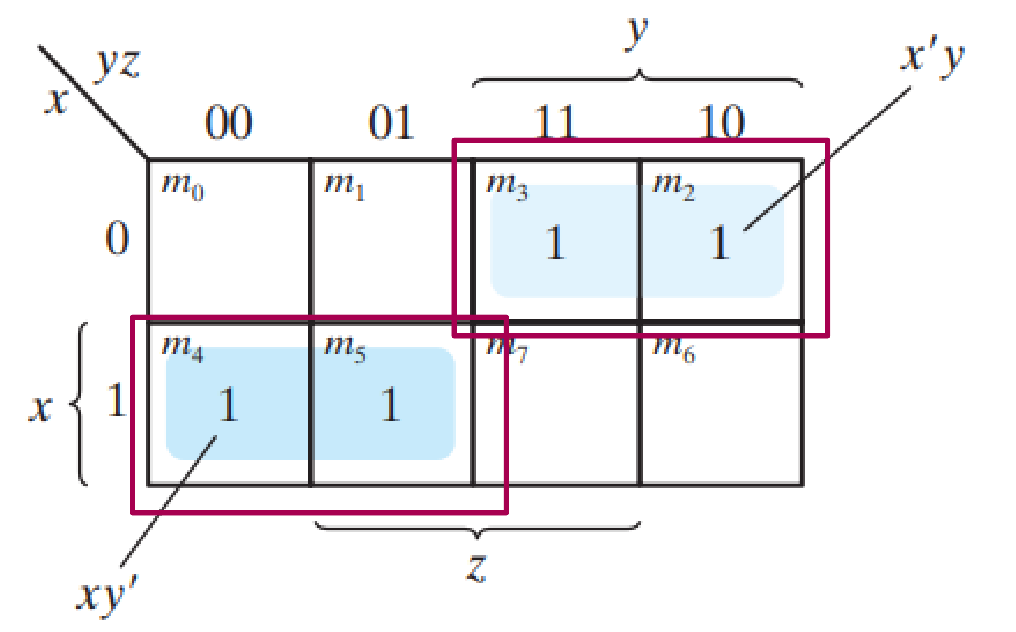
Ollscoil na Gaillimhe
University of Galway

Examples

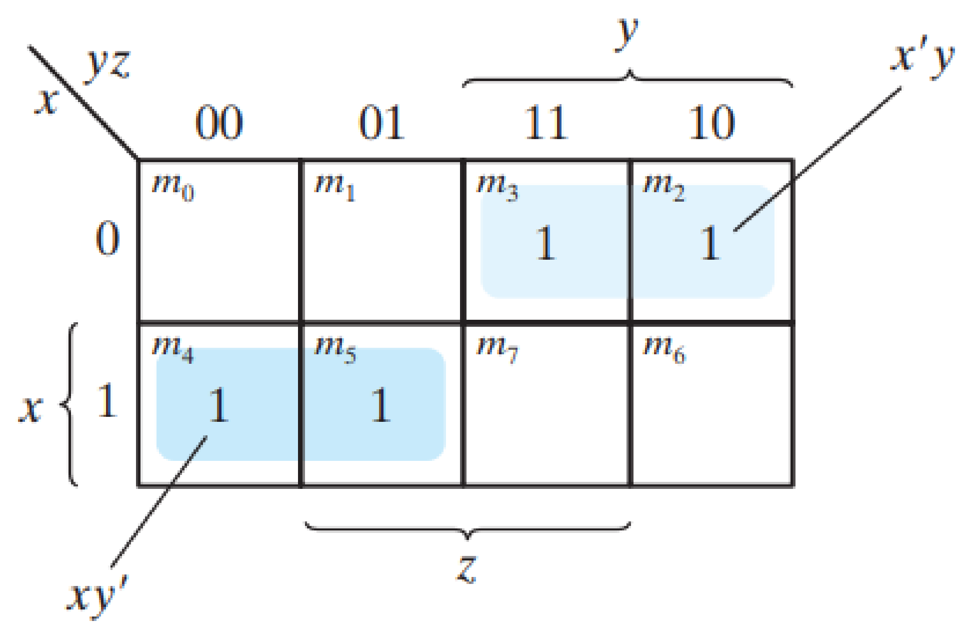
Simplifying the Boolean function **F(x, y, z) = ∑(2, 3, 4, 5):**

1. Begin with the Boolean function F(x, y, z) = ∑(2, 3, 4, 5).
2. Mark 1s in minterm squares that represent the function (minterms 010, 011, 100, 101).
3. Identify adjacent squares.
    1. Upper right rectangle: Represents **x'y** (complement of x and y).
    2. Lower left rectangle: Represents **xy'** (x and complement of y).

These simplifications help in reducing the complexity of the Boolean function for better representation and analysis.



Examples



4. Simplified Expression:
   - The sum of four minterms can be replaced by just two product terms.
   - Logical sum: **F = x'y + xy'**

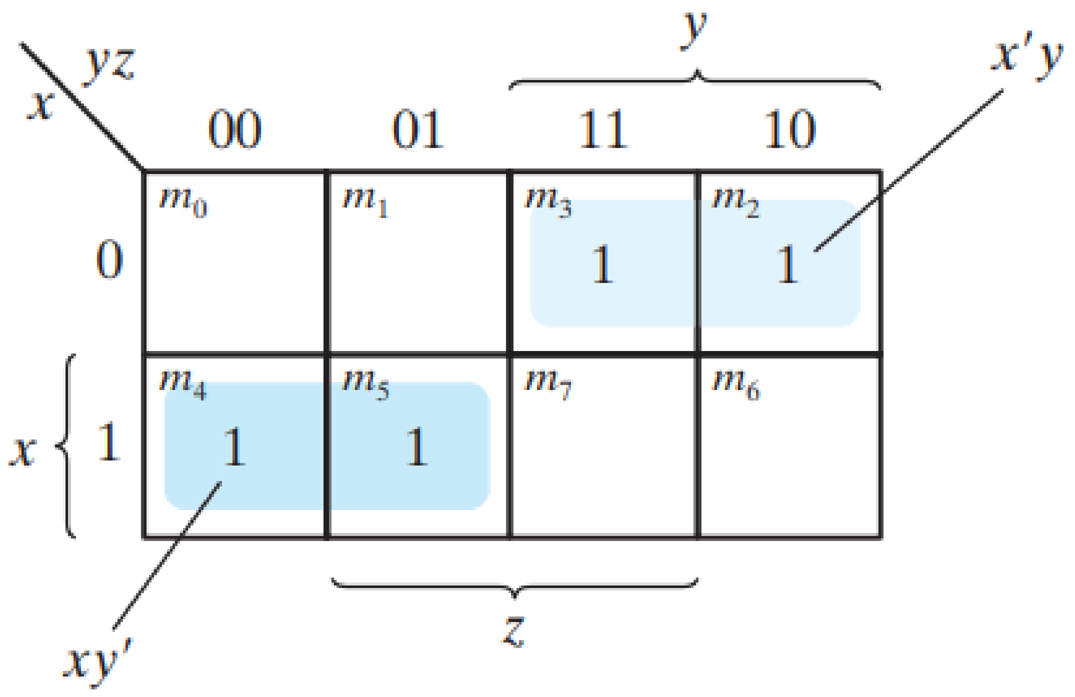
5. Expanding Adjacency:
   - In some cases, squares in the map are considered adjacent, even if they don't physically touch.
   - Verified algebraically:

$$\mathbf{m0 + m2 = x'y'z' + x'yz' = x'z'(y' + y) = x'z'}$$
$$\mathbf{m4 + m6 = xy'z' + xyz' = xz' + (y' + y) = xz'}$$

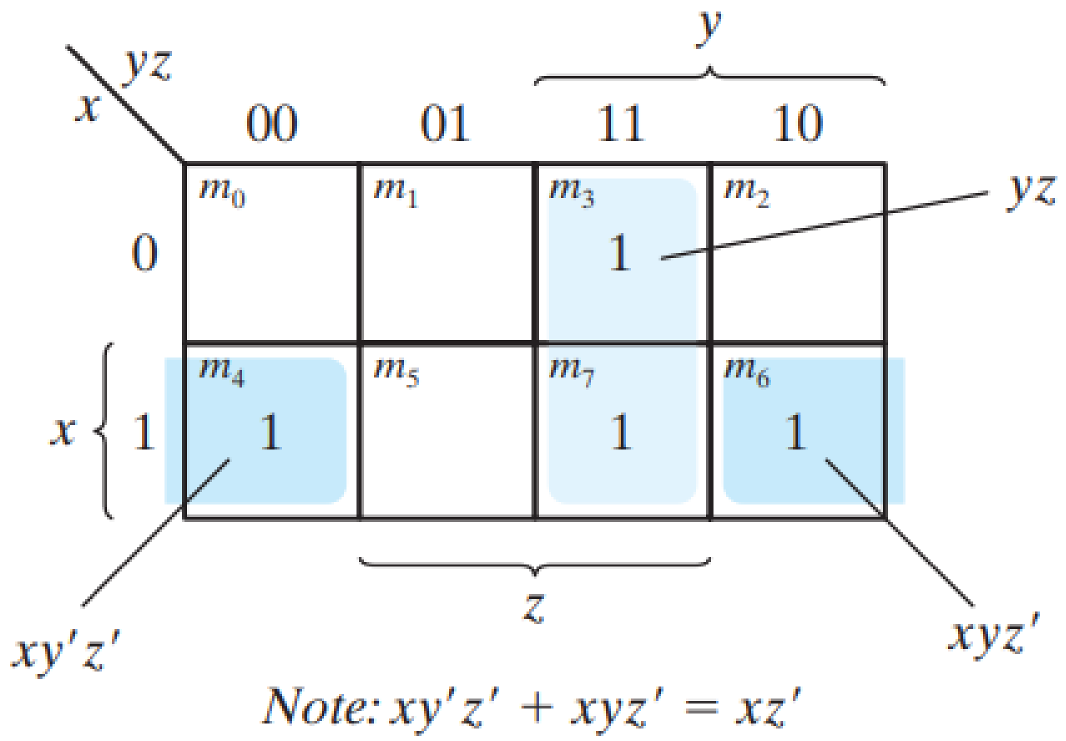
Examples

6. Modified Definition of Adjacency:

   - To include such cases, the map is considered as if it wraps around, touching left and right edges to form adjacent squares.



Examples



Simplify the Boolean function **F(x, y, z) = ∑(3, 4, 6, 7)**

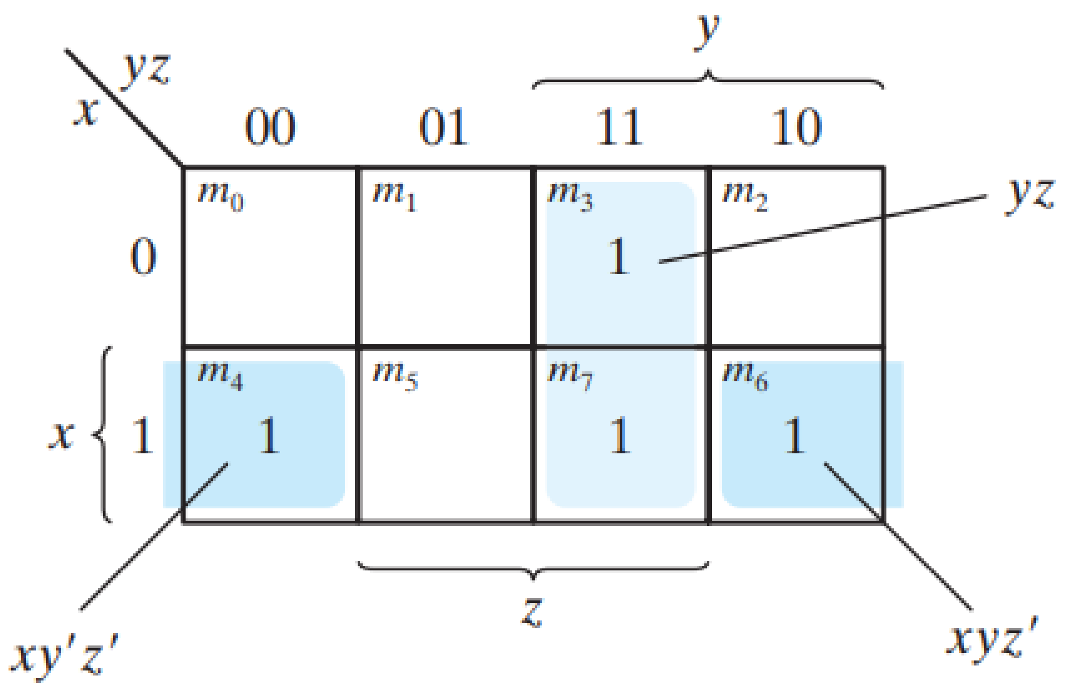
1. Mark squares with 1s in the Karnaugh Map (minterms 011, 100, 110, 111).
2. Two adjacent squares are combined in the third column to give a two-literal term **yz**.
3. The remaining two squares with 1's are also adjacent by the new definition.
4. These two squares, when combined, give the two-literal term **xz**.
5. Results the simplified function:

F = yz + xz'

Examples

1. Simplification Principle:
   - Four adjacent squares in a three-variable map represent the logical sum of four minterms, resulting in an expression with only one literal.
     - Example: The sum of adjacent minterms 0, 2, 4, and 6 simplifies to **z'**.
     - Consider the logical sum of the four adjacent minterms 0, 2, 4, and 6.
     - Simplified to the single literal term **z'**:

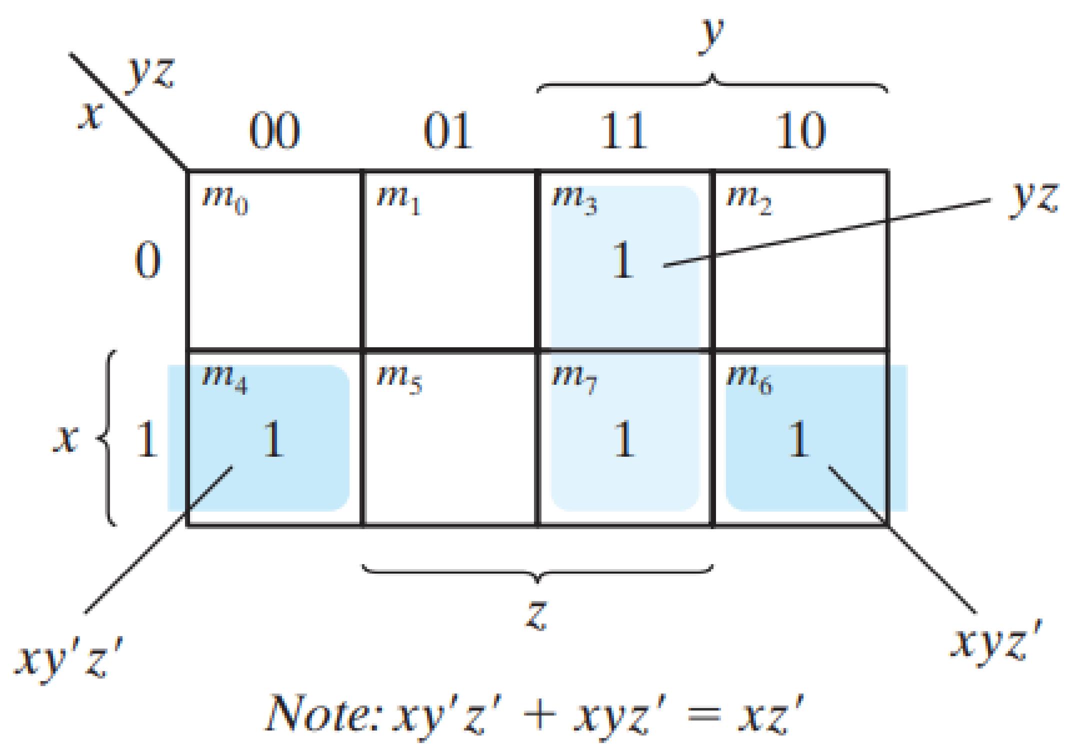
$$\begin{aligned} m_0 + m_2 + m_4 + m_6 &= x'y'z' + x'yz' + xy'z' + xyz' \\ &= x'z'(y' + y) + xz'(y' + y) \\ &= x'z' + xz' = z'(x' + x) = z' \end{aligned}$$



Note: $xy'z' + xyz' = xz'$

Examples

1. Number of Adjacent Squares and Literal Reduction:
   - The number of adjacent squares combined should always be a power of two (e.g., 1, 2, 4, 8).
   - One square represents a term with three literals.
   - Two adjacent squares represent a term with two literals.
   - Four adjacent squares represent a term with one literal.
   - Eight adjacent squares encompass the entire map and result in a function always equal to 1.

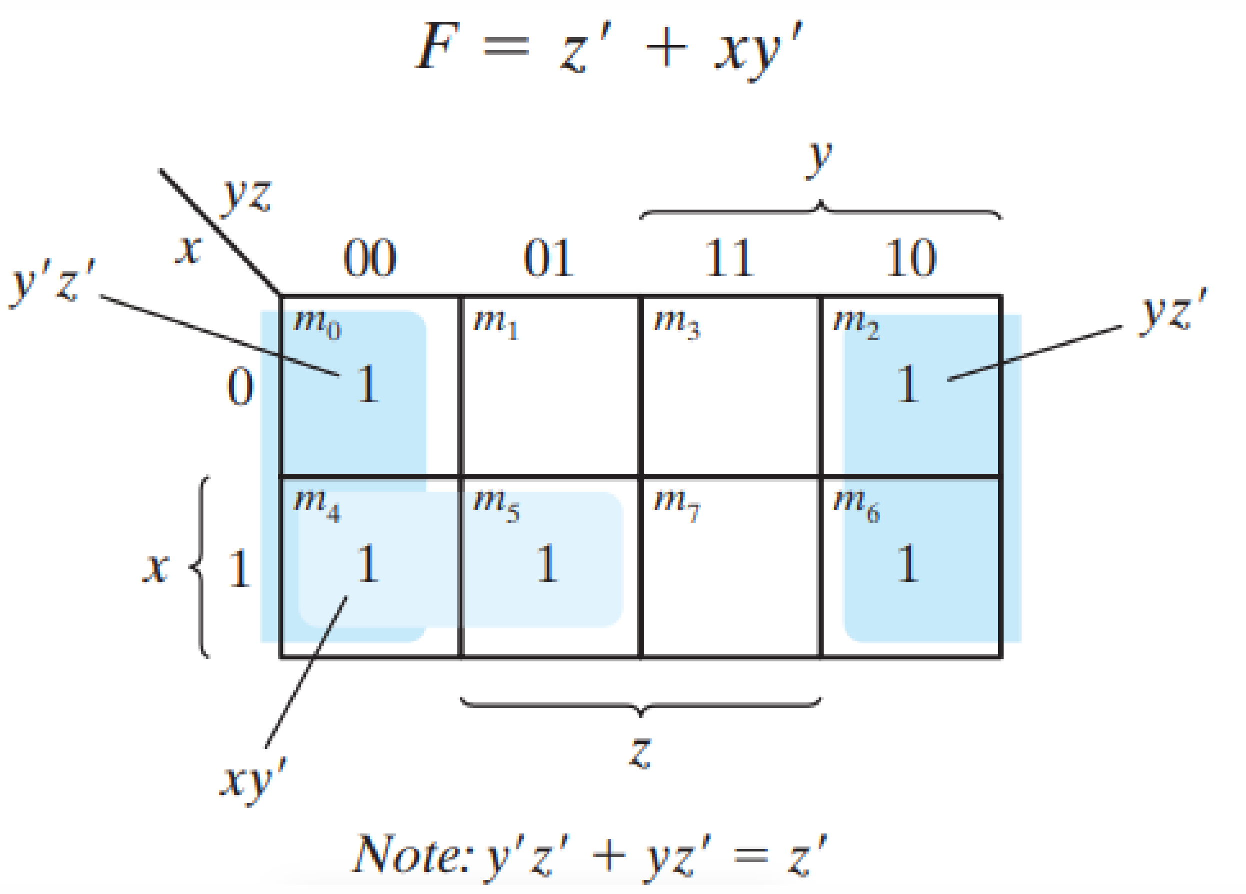


OLLSCOIL NA GAILLIMHE
UNIVERSITY OF GALWAY

Problem

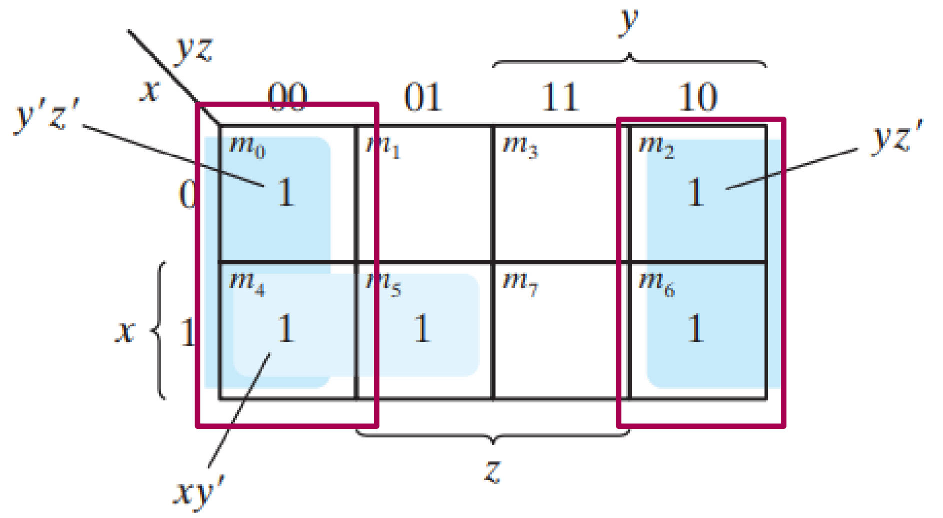
Simplify the Boolean function **$F(x, y, z) = \sum(0,2,4,5,6)$**

Solution



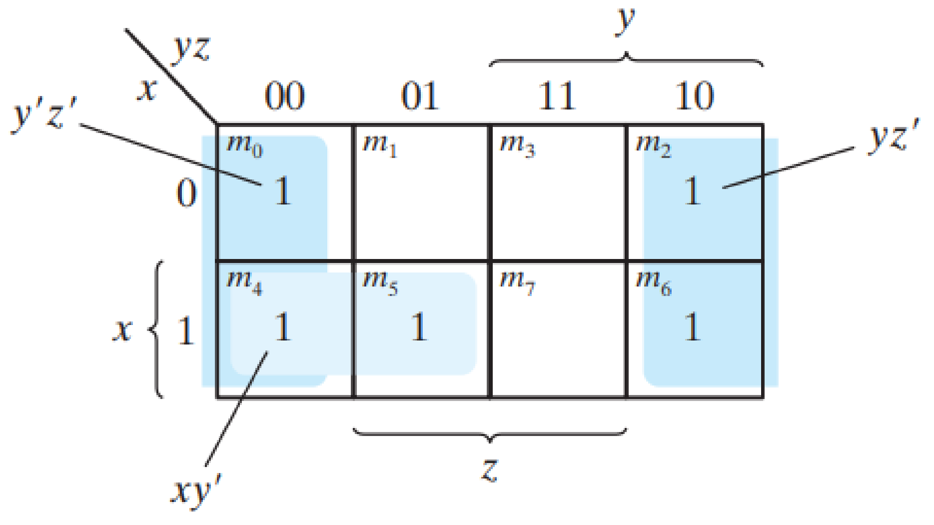
Simplify the Boolean function **F(x, y, z) = ∑(0,2,4,5,6)**

1. Utilize the map on right side for function F.
2. Start by combining the four adjacent squares in the first and last columns to obtain the single literal term **z'**.



3. Combine the single square representing minterm 5 with an adjacent square.

Solution



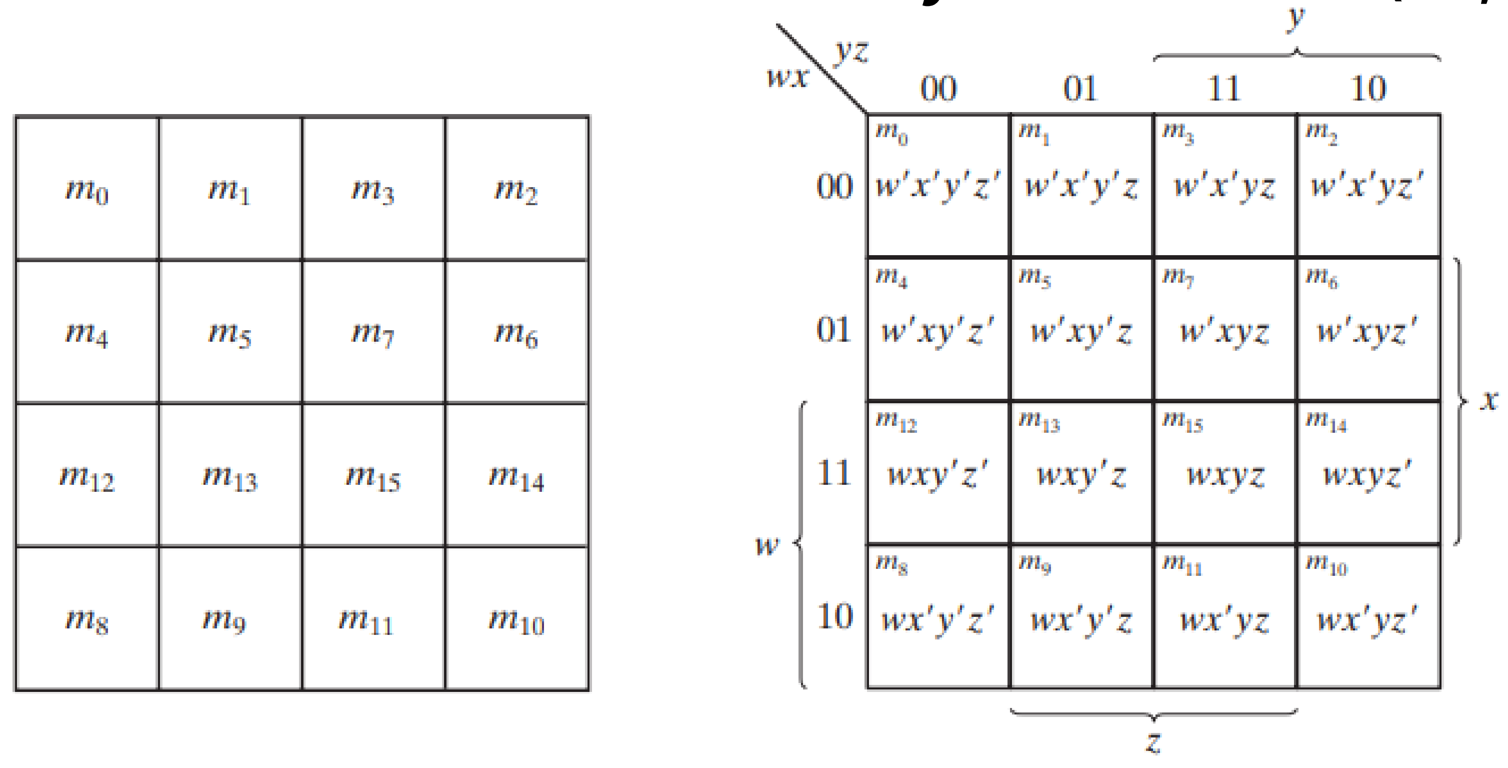
4. This combination results in:
    - Two-literal term: **xy'**
    - Three-literal minterm: **xy’z**

5. The simplified function becomes: **F = z' + xy’**

- If a function is not in sum-of-minterms form, the Karnaugh Map can help identify minterms and simplify it to an expression with minimal terms.
- Ensure that the resulting algebraic expression is in sum-of-products form.
- Plot each product term on the map in one, two, or more squares.
- Read the minterms directly from the map to simplify the function.

Four-Variable K-Map

- A map for Boolean functions with four binary variables (w, x, y, z) is presented in Figure.

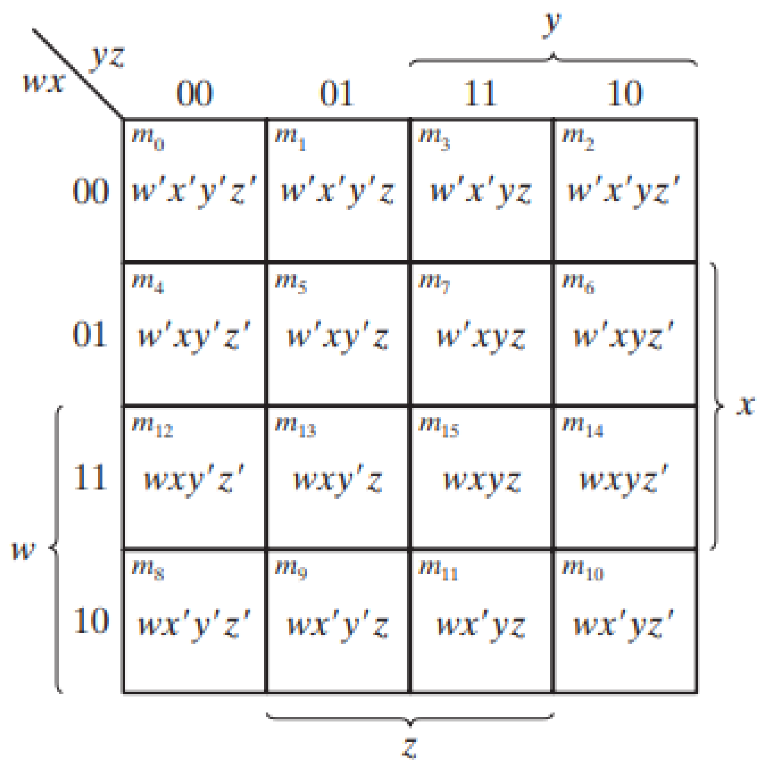


(a) (b)

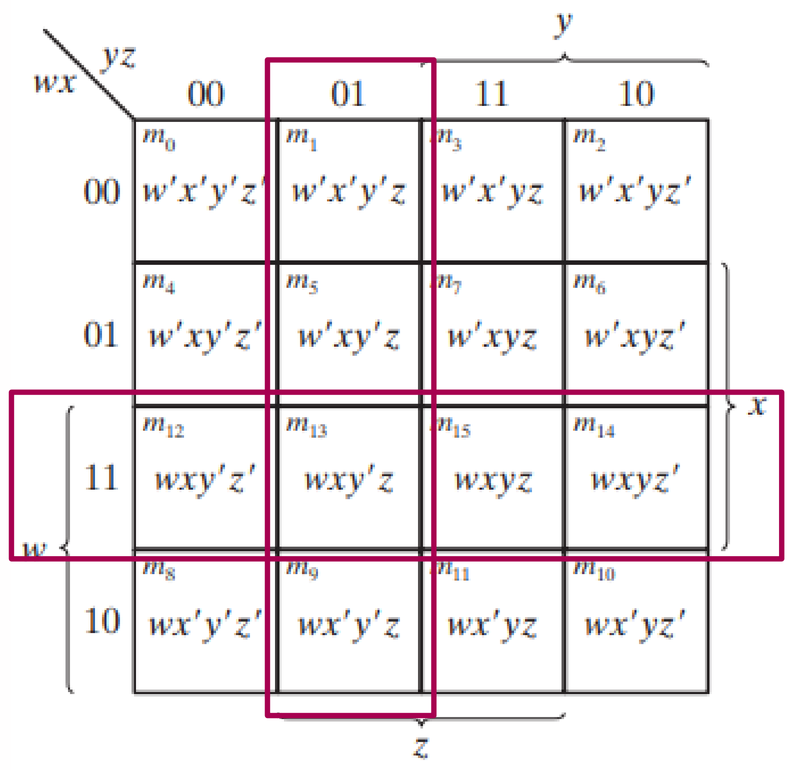
The 16 minterms are listed along with the squares assigned to each.

Illustrate the relationship between squares and the four variables.

Four-Variable K-Map

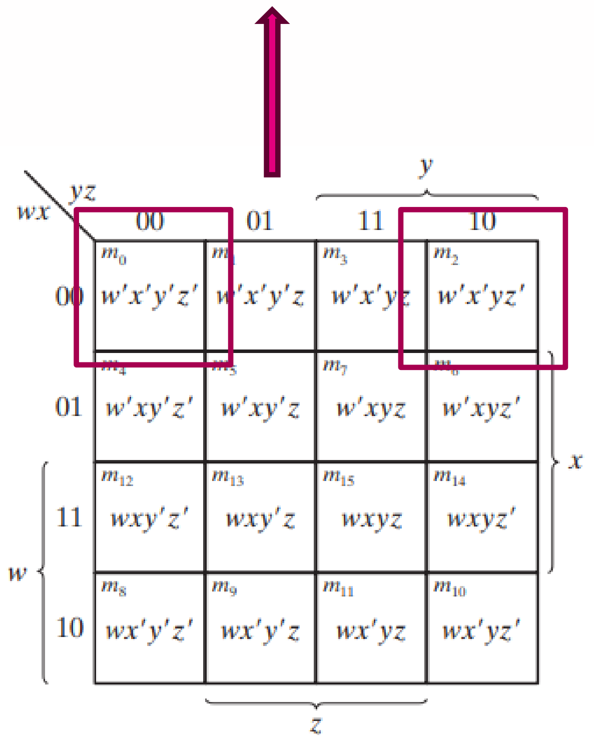


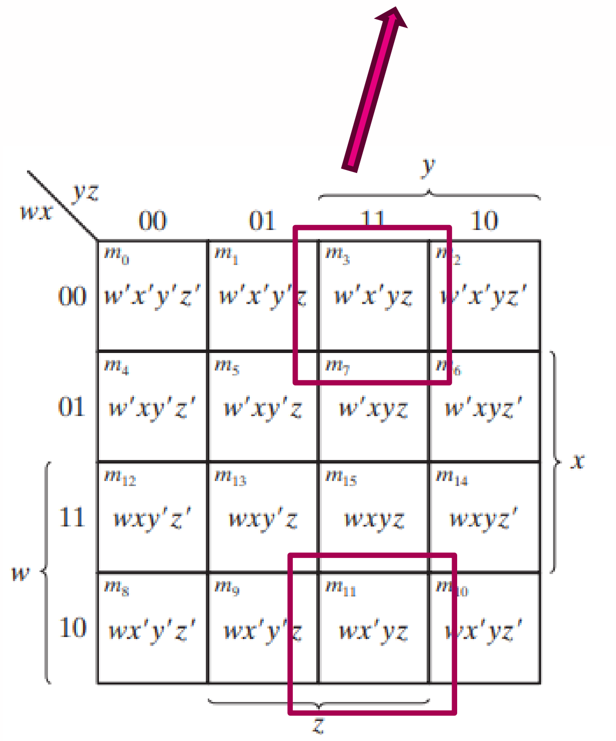
- Rows and columns are numbered in a Gray code sequence, with only one digit changing between adjacent rows or columns.
- The minterm for each square is obtained by concatenating the row and column numbers.
- For example, the third row (11) and the second column (01) when concatenated yield the binary number 1101, equivalent to decimal 13.
- Thus, the square in the third row and second column represents minterm $\mathbf{m_{13}}$.



Four-Variable K-Map

- Minimization of four-variable Boolean functions is similar to the method used for three-variable functions.
- Adjacent squares are defined as those next to each other.
- The map is considered to be on a surface where the top and bottom edges, as well as the right and left edges, touch each other to form adjacent squares.
- For example, m0 and m2 are adjacent squares, and so are m3 and m11.





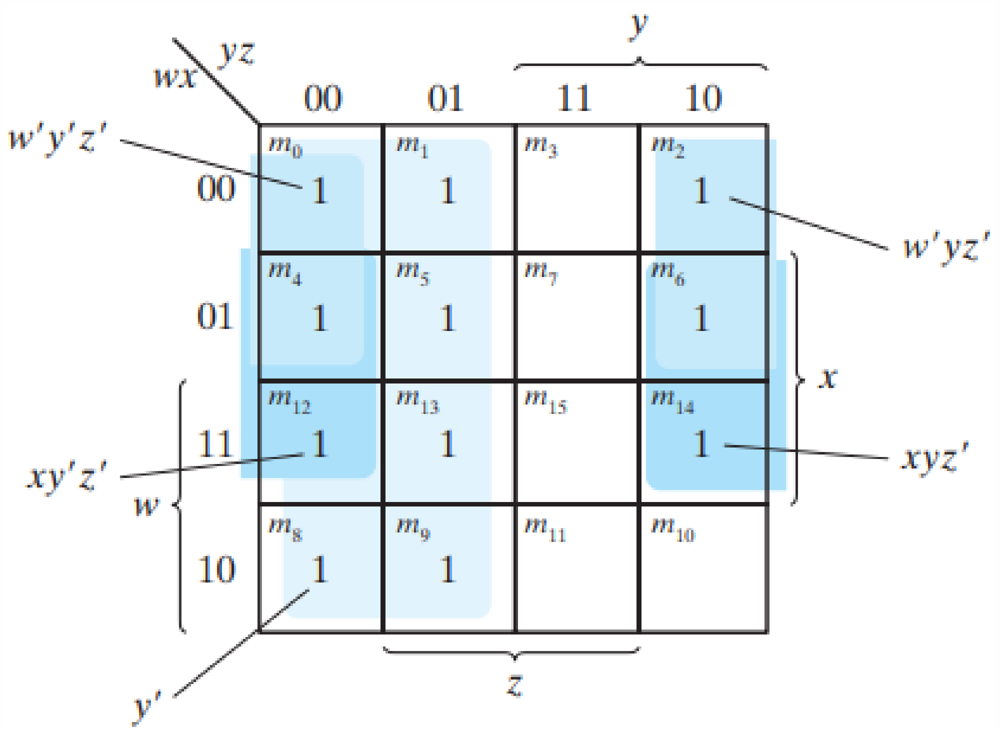
OLLSCOIL NA GAILLIMHE
UNIVERSITY OF GALWAY

Four-Variable K-Map

The combination of adjacent squares during simplification can be determined from inspection of the four-variable map:

- One square represents a minterm, resulting in 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 yield a function that is always equal to 1.

Example



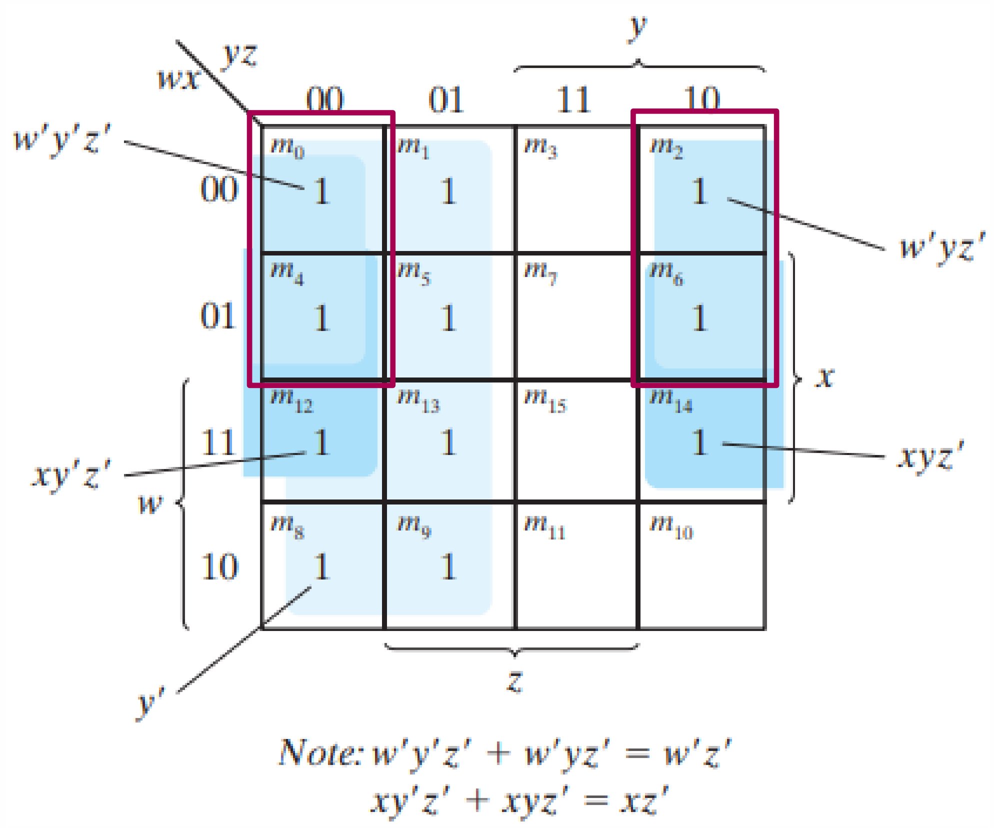
Note: $w'y'z' + w'yz' = w'z'$
$xy'z' + xyz' = xz'$

Simplify the Boolean function

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

1. Minterms are marked with 1's in a four-variable map.
2. Eight adjacent squares with 1's are combined to form the single literal term y'.
3. The remaining three 1's on the right cannot be combined to create a simplified term.
4. Instead, they are combined as two or four adjacent squares to reduce the number of literals in the term.

Example

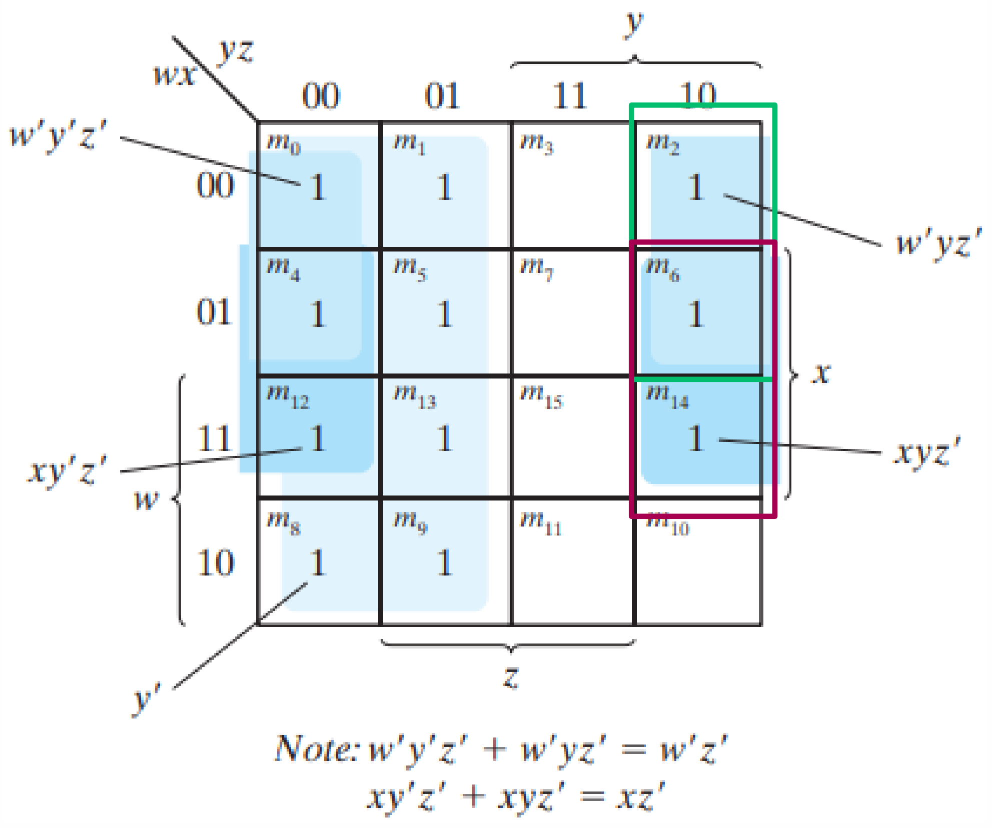


6. In this example, the top two 1's on the right are combined with the top two 1's on the left to create the term **w'z'**.

7. It's important to note that it's permissible to use the same square more than once in the process.

Example

6. A square marked with 1 in the third row and fourth column (square 1110) remains.
7. Instead of considering this square alone (which would result in a term with four literals), it is combined with squares already used to form an area of four adjacent squares.



8. These squares form the two middle rows and the two end columns, giving the term xz'.
9. The simplified function is now **F = y' + w'z' + xz'.**

Problem

Simplify the Boolean function

$$F = A'B'C' + B'CD' + A'BCD' + AB'C'$$

Solution

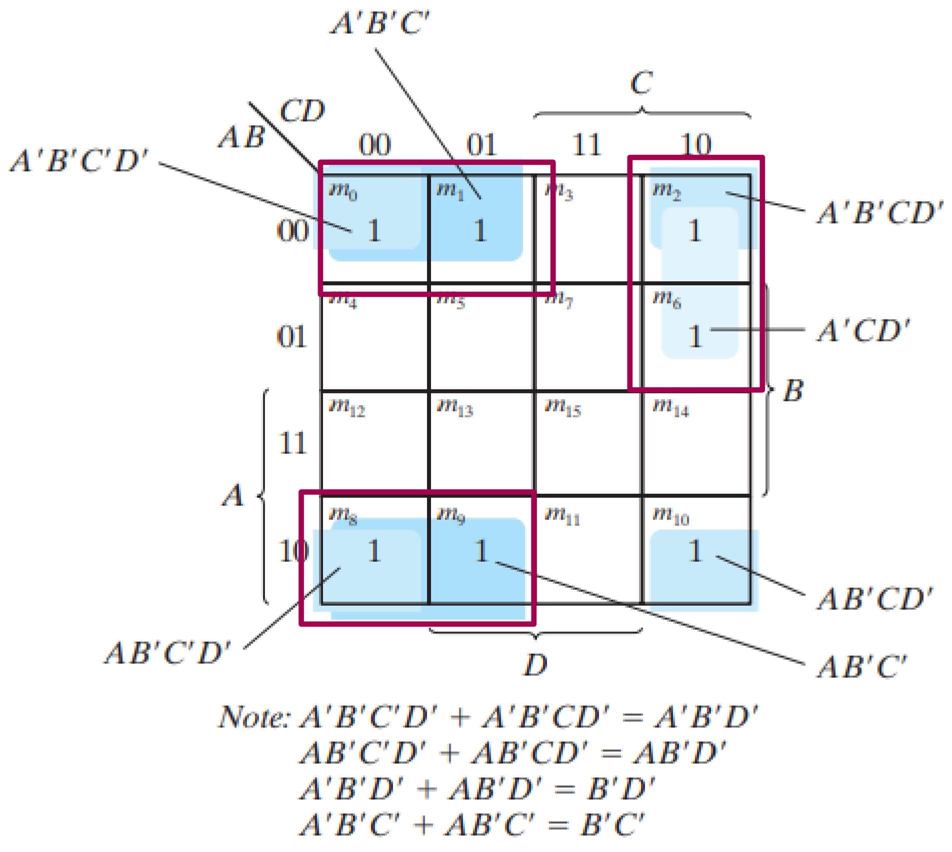
Simplify the Boolean function

F = A'B'C' + B'CD' + A'BCD' + AB'C'

1. The area in the map covered by this function includes squares marked with 1's in Figure right.

2. The function has four variables and consists of terms with three and four literals.

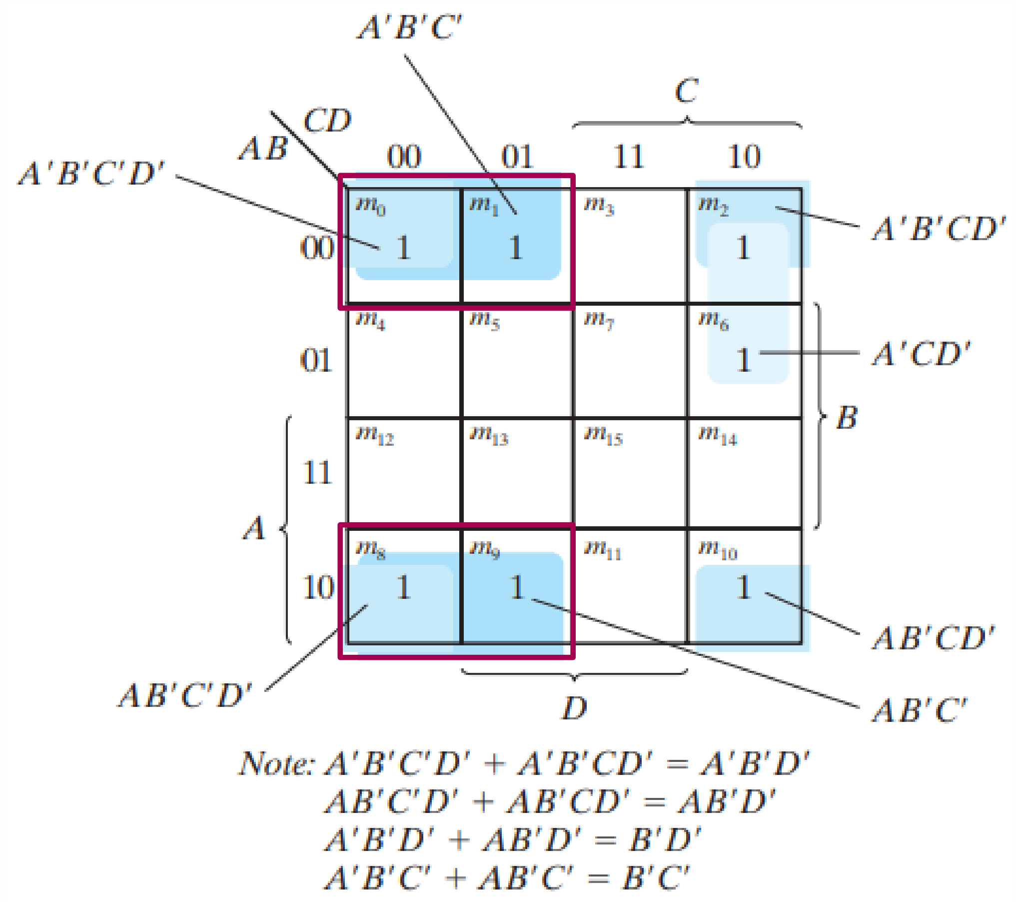
3. Each term with three literals is represented in the map by two squares.



Solution

4. To simplify, identify adjacent squares that can be combined to reduce the number of literals.

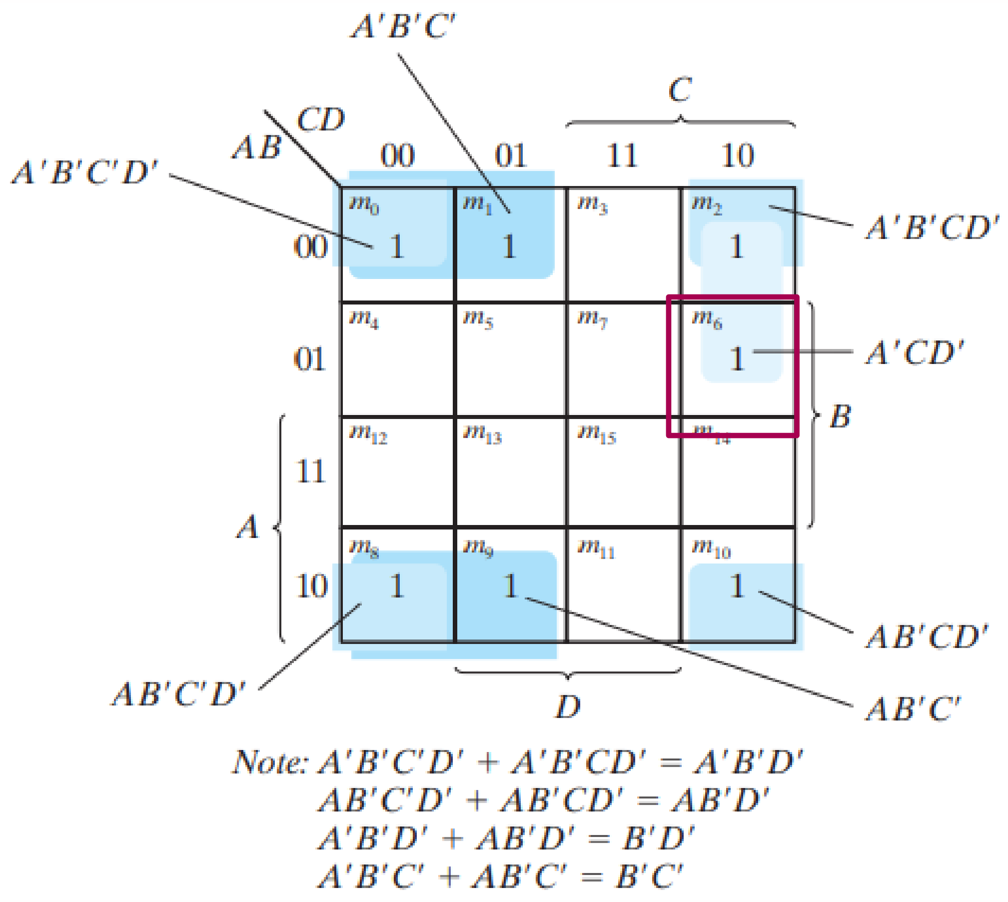
5. Take advantage of the adjacency, the two left-hand 1's in the top row are combined with the two 1's in the bottom row to form the term B'C'.



Solution

4. The remaining 1 may be combined in a two-square area to create the term A'CD’.

5. The simplified function is **F = B'D' + B'C' + A'CD'**.



Prime Implicants

Choosing adjacent squares in a map:

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

Prime Implicants

- 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.
- The prime implicants of a function can be obtained from the map by combining all possible maximum numbers of squares.

Prime Implicants

- A single 1 on the map represents a prime implicant if it's not adjacent to any other 1’s.
- Two adjacent 1's form a prime implicant, provided they are not within a group of four adjacent squares.
- Four adjacent 1's form a prime implicant if they are not within a group of eight adjacent squares, and so on.

Prime Implicants

Essential Prime Implicants:

- To find essential prime implicants, examine each square marked with a 1 and check how many prime implicants cover it.
- If a square is covered by only one prime implicant, that prime implicant is considered essential.

Prime Implicants

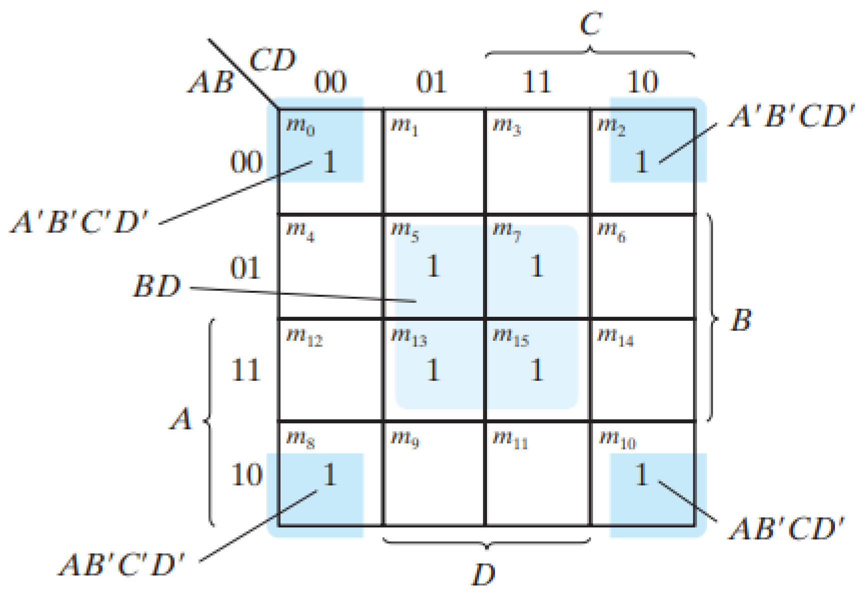
Simplifying a Four-Variable Boolean Function:

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

1. The function's minterms are marked with 1's in the maps.

2. A partial map shows two essential prime implicants, each formed by collapsing four cells into a term with only two literals. The terms are **B'D'** and **BD**.

3. These essential prime implicants cover **eight** minterms, but three (**m_3, m_9, m_{11}**) are omitted.



Note: $A'B'C'D' + A'B'CD' = A'B'D'$
$AB'C'D' + AB'CD' = AB'D'$
$A'B'D' + AB'D' = B'D'$

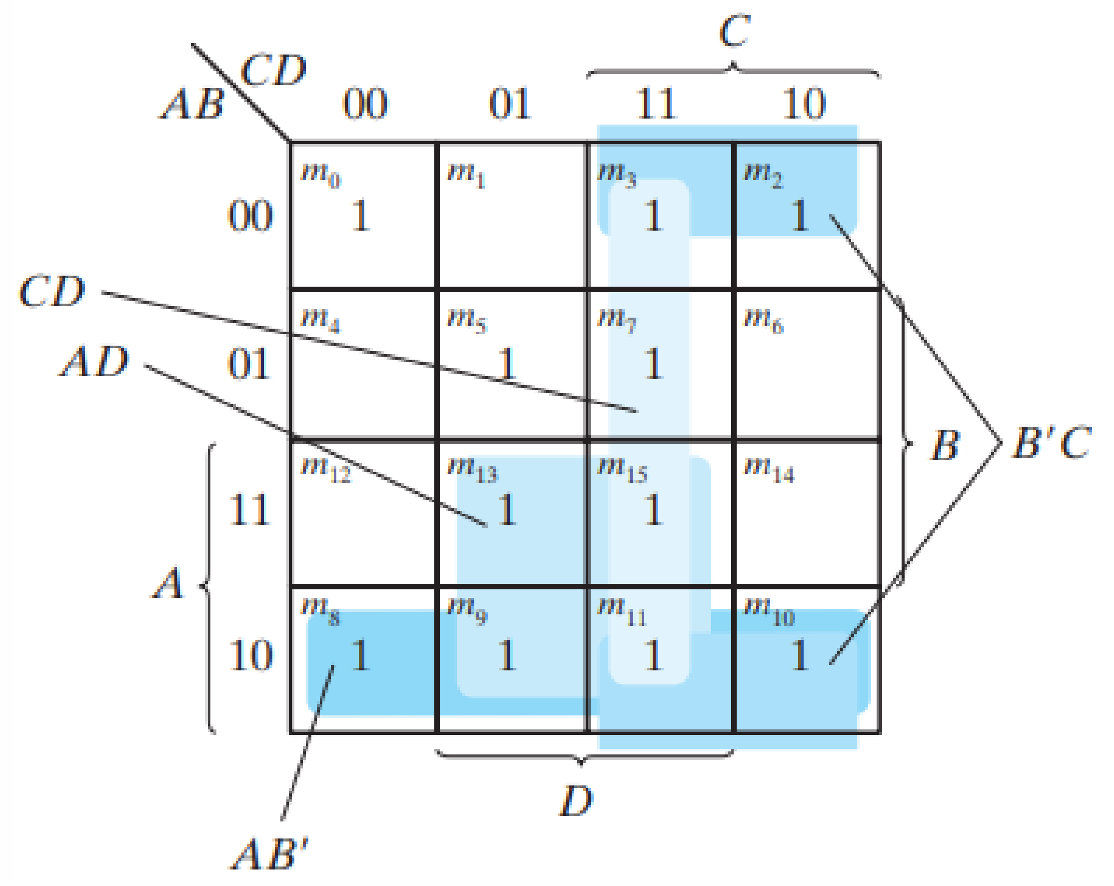
(a) Essential prime implicants BD and $B'D'$

Prime Implicants

4. All possible ways to cover the omitted minterms with prime implicants are shown.

   For example, m_3 can be covered by either CD or B'C.

5. The simplified expression is obtained from the logical sum of the two essential prime implicants and any two prime implicants that cover m_3, m_9, and m_{11}.



Prime Implicants

6. There are four possible expressions with four product terms of two literals each:

$$F = BD + B'D' + CD + AD$$
$$F = BD + B'D' + CD + AB'$$
$$F = BD + B'D' + B'C + AD$$
$$F = BD + B'D' + B'C + AB'$$

Five or More Variable Maps

- A five-variable map requires 32 squares, and a six-variable map needs 64 squares.
- As the number of variables increases, such as in a six-variable map with 64 squares, the number of squares becomes excessive.
- With more variables, the geometry for combining adjacent squares becomes more complex.
- Due to the difficulties and complexity involved, maps for more than four variables are not typically considered.

References

- Computer Organization and Architecture Designing for Performance Tenth Edition by William Stallings
- Digital Design With an Introduction to the Verilog HDL FIFTH EDITION by M Morris, M. and Michael, D., 2013.

Ollscoil na Gaillimhe
University of Galway

Thank *you*