Dr. SNS RAJALAKSHMI COLLEGE OF ARTS & SCIENCE (Autonomous)

Coimbatore -641049

Accredited by NAAC(Cycle–III) with ‘A+’ Grade
(Recognized by UGC, Approved by AICTE, New Delhi and
Affiliated to Bharathiar University, Coimbatore)

DEPARTMENT OF GRAPHIC & CREATIVE DESIGN AND DATA ANALYTICS

COURSE NAME : COMPUTER SYSTEM ARCHITECTURE (23UCU402)

I YEAR /I SEMESTER

Unit II- Logic Gates

Topic : K-MAP

Standard Forms

- **<u>Standard Sum-of-Products (SOP) form:</u>** equations are written as "AND" terms summed with "OR" operators.
- **<u>Standard Product-of-Sums (POS) form:</u>** equations are written as "OR" terms, all "ANDed" together.
- Examples:

  SOP: $A\,B\,C + \overline{A}\,\overline{B}\,C + B$

  POS: $(A + B) \bullet (A + \overline{B} + \overline{C}) \bullet (C)$

- These "Mixed" forms are <u>not SOP or POS</u>

  Wrong: $(A\,B + C)\,(A + C)$ or $A\,B\,\overline{C} + A\,C\,(A + B)$

Standard Sum-of-Products (SOP)

- **A <u>Sum of Minterms</u> form for *n* variables can be written down directly from a truth table.**
- **Implementation of this form is a two-level network of gates such that:**
  - **The first level consists of *n*-input AND gates, and**
  - **The second level is a single OR gate (with fewer than 2^n inputs).**
- **This form:**
  - **is usually <u>not</u> a minimum literal expression, and**
  - **leads to a more expensive implementation (in terms of two levels of AND and OR gates) than needed.**

Standard Sum-of-Products (SOP)

- **Therefore, we try to combine terms to get a lower literal cost expression, leading to a less expensive implementation.**
- **Example:** $F(A, B, C) = \sum(1,4,5,6,7)$
- **Simplifying**

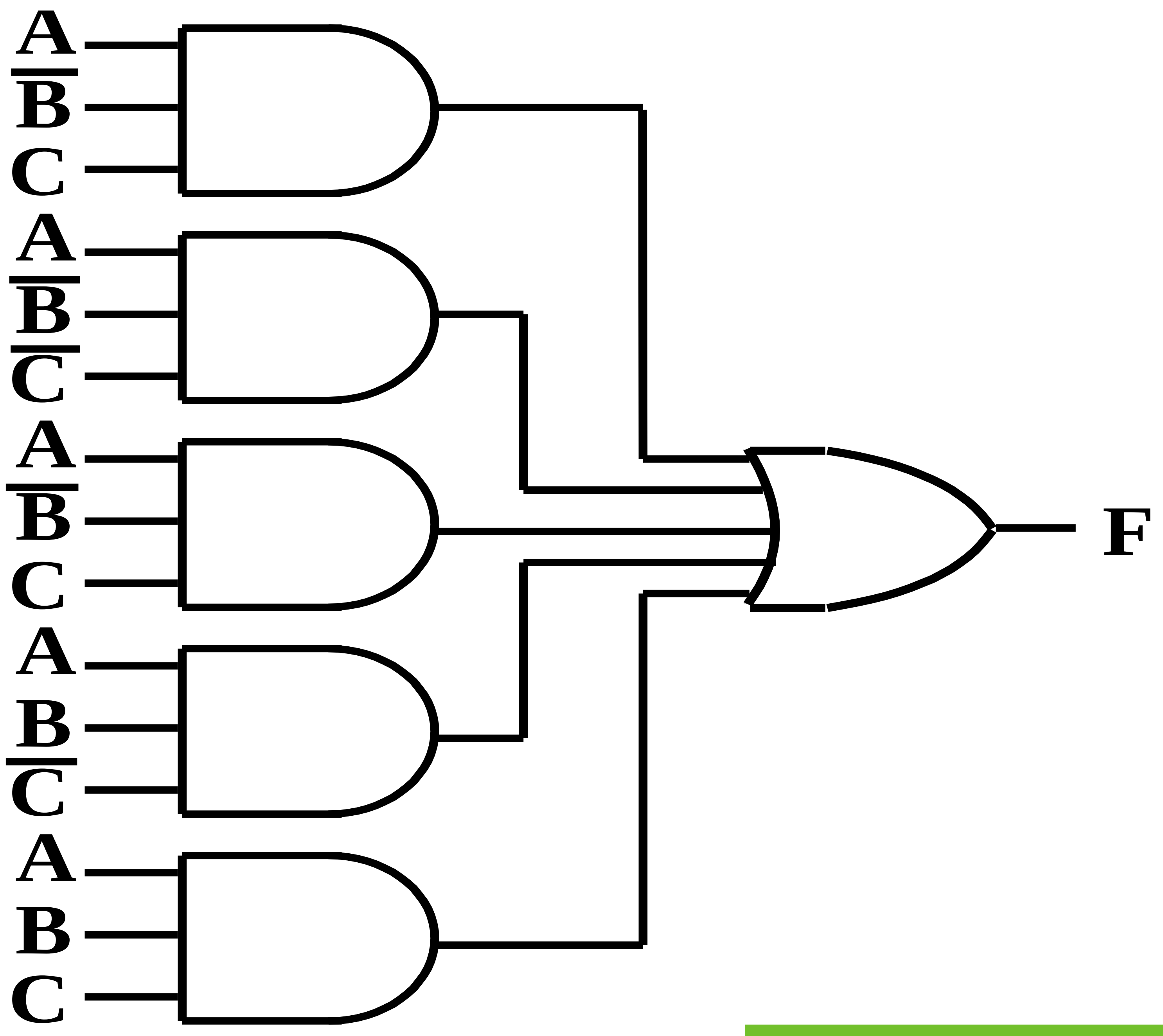
$$F = A B C + A B \overline{C} + A \overline{B} C + A \overline{B}\,\overline{C} + \overline{A}\,\overline{B} C$$
$$= A B (C + \overline{C}) + A \overline{B} C + A \overline{B}\,\overline{C} + A \overline{B} C + \overline{A}\,\overline{B} C$$
$$= A B + A \overline{B} (C + \overline{C}) + (A + \overline{A})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

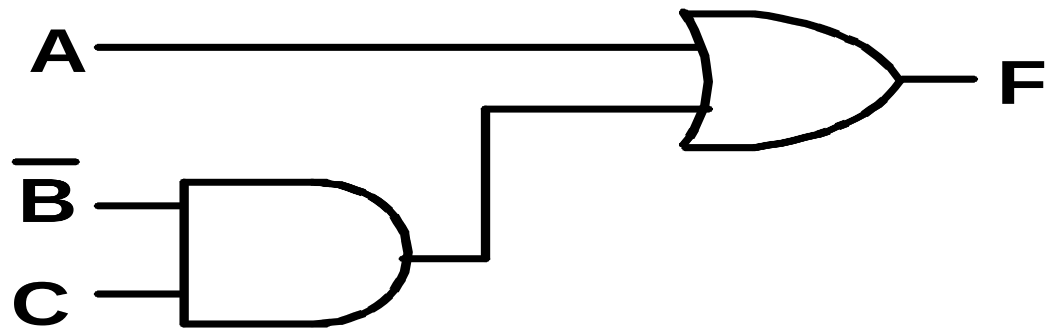
Note term $A\overline{B}C$ duplicated

The Canonical Sum-of-Minterms form has (5 * 3) = 15 literals and 5 terms. The reduced SOP form has 3 literals and 2 terms.

AND/OR Two-level Implementation of SOP Expression

- **The two implementations for F are shown below: (Which is simpler?)**





Standard Product-of-Sums (POS)

- **A <u>Product of Maxterms</u> form for *n* variables can be written down directly from a truth table.**
- **Implementation of this form is a two-level network of gates such that:**
  - **The first level consists of *n*-input OR gates, and**
  - **The second level is a single AND gate (with fewer than 2*n* inputs).**
- **This form:**
  - **is usually <u>not</u> a minimum literal expression, and**
  - **leads to a more expensive implementation (in terms of two levels of AND and OR gates) than needed.**

Standard Product-of-Sums (POS)

We can use Boolean algebra to minimize the literal cost of an expression for POS forms.

Example: $F = \prod(0,2,3)$

Becomes: F= (A+B+C)(A+B'+C)(A+B'+C')

(Note **duplicate** term) F= (A+C+B)(A+C+B')**(A+B'+C)**(A+B'+C')

= ((A+C)+BB')((A+B')+CC')

= ((A+C)+0)((A+B')+0)

= (A+C)(A+B')

Standard Product-of-Sums (POS)

- **Therefore, we try to combine terms to get a <u>lower</u> literal cost expression, leading to a <u>less</u> expensive implementation.**
- **Example:** $F = \prod(0, 2, 3)$
- **Simplifying**

Note term $A + \overline{B} + C$ duplicated

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

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

$$= ((A + C) + B\overline{B})((A + \overline{B}) + C\overline{C})$$

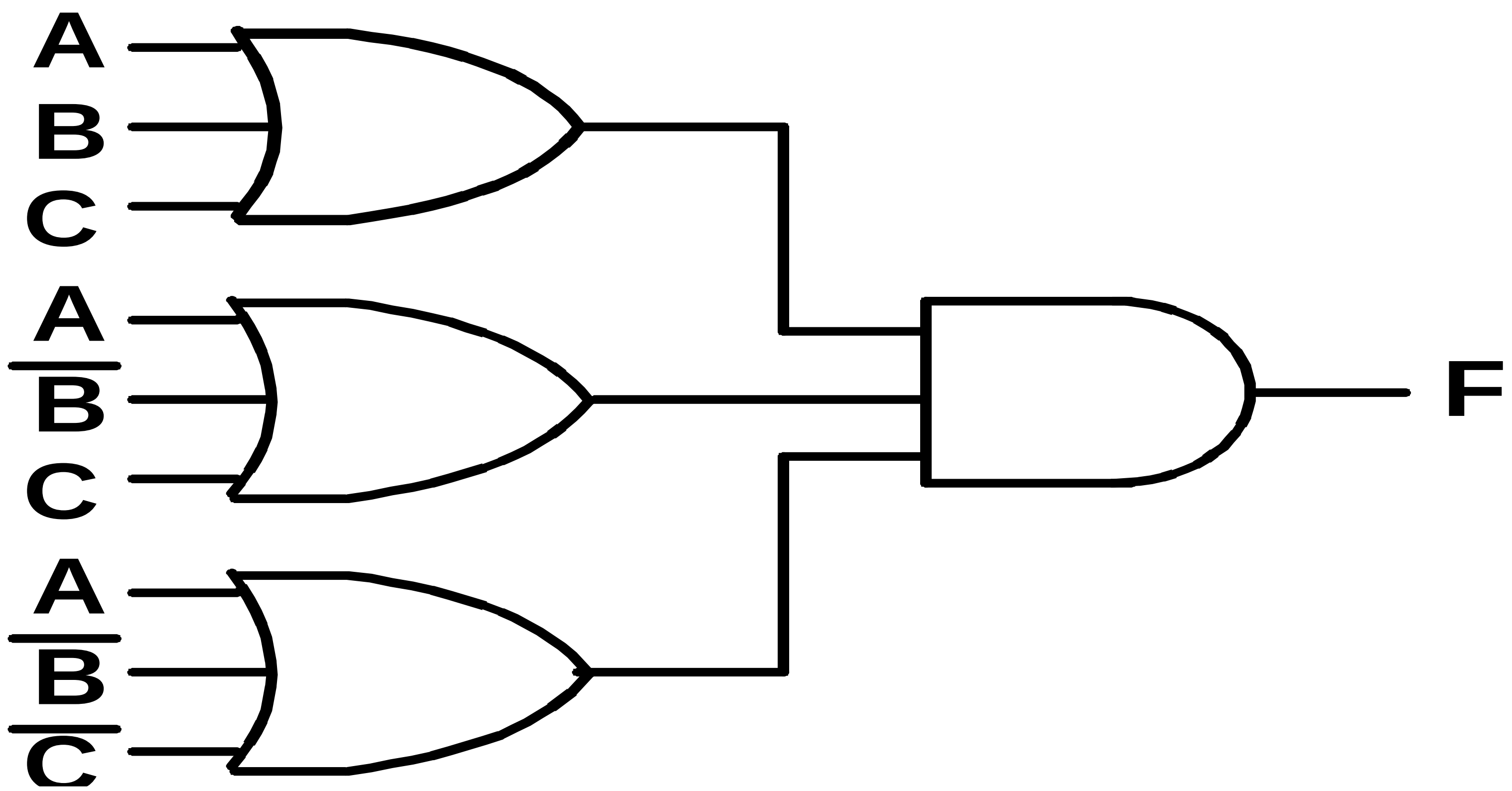
$$= ((A + C) + 0)((A + \overline{B}) + 0)$$

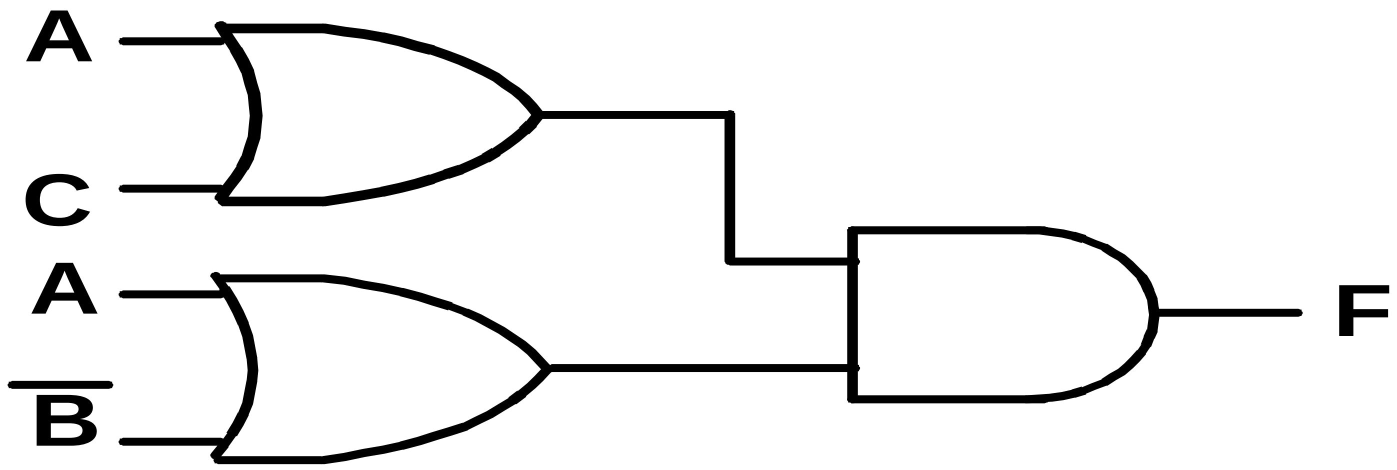
$$= (A + C)(A + \overline{B})$$

The Canonical Product-of-Maxterms form had (3 * 3) = 9 literals and 3 terms. The reduced POS form had 4 literals and 2 terms.

OR/AND Two-level Implementation

- **The two implementations for F are shown below: (Which is simpler?)**





SOP and POS Observations

- The previous examples show several things:
  - Canonical Forms (Sum-of-minterms, Product-of-Maxterms), or other standard forms (SOP, POS) can differ in literal cost.
  - Boolean algebra can be used to manipulate equations into simpler forms.
  - Simpler equations lead to simpler two-level implementations
- Questions:
  - How can we attain a minimum literal expression?
  - Is there only one minimum cost circuit?

Equivalent Cost Circuits

Given: $F(A,B,C) = \sum(0,2,3,4,5,7)$

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

By pairing terms differently at the start:

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

We arrive at:

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

BOTH HAVE THE SAME LITERALS COUNTS AND NUMBER OF TERMS !!

Boolean Function Simplification

- Reducing the literal cost of a Boolean Expression leads to simpler networks.
- Simpler networks are less expensive to implement.
- Boolean Algebra can help us minimize literal cost.
- When do we stop trying to reduce the cost?
- Do we know when we have a minimum?
- We will introduce a systematic way to arrive a a minimum cost, two-level POS or SOP network.

Karnaugh Maps (K-map)

- Diagram is a collection of squares
- Each square represents a minterm
- Collection of squares is a graphical representation of the Boolean function
- Adjacent squares differ in one variable
- Pattern recognition is used to derive alternative algebraic expressions for the same function
- The Karnaugh Map can be viewed as an extension of the truth table
- The Karnaugh Map can be viewed as a topologically warped Venn diagram as used to visualize sets

Uses of Karnaugh Maps

- Provide a means for finding optimum:
  - Simple SOP and POS standard forms, and
  - Small two-level AND/OR and OR/AND circuits
- Visualize concepts related to manipulating Boolean expressions
- Demonstrate concepts used by computer-aided design programs to simplify large circuits

Two Variable Maps

A Two variable Karnaugh Map:

- Note that minterm m_0 and minterm m_1 are "adjacent" and differ in the value of the variable y.
- Similarly, minterm m_0 and minterm m_2 differ in the x variable.
- Note that m_1 and m_3 differ in the x variable as well.
- Minterms m_2 and m_3 differ in the value of the variable y

	y=0	y=1
x=0	$m_0 = \overline{x}\,\overline{y}$	$m_1 = \overline{x}\,y$
x=1	$m_2 = x\,\overline{y}$	$m_3 = x\,y$

K-Map and Function Tables

- The K-Map is just a different form of the function table. For two variables, both are shown below. We choose a,b,c and d from the set {0,1} to implement a particular function, F(x,y).

Function Table

Input Values (x,y)	Function Value F(x,y)
0 0	a
0 1	b
1 0	c
1 1	d

K-Map

	y=0	y=1
x=0	a	b
x=1	c	d

K-Map Function Representations

Examples

F(x,y) = x

F=x	y=0	y=1
x=0	0	0
x=1	1	1

G(x,y) = x+y

G=x+y	y=0	y=1
x=0	0	1
x=1	1	1

- For function F(x,y), the two adjacent cells containing 1's can be combined using the Minimization Theorem:

$$F(x,y) = x\bar{y} + xy = x$$

- For G(x,y), two pairs of adjacent cells containing 1's can be combined using the Minimization Theorem:

$$G(x,y) = \left(x\bar{y} + xy\right) + \left(xy + \bar{x}y\right) = x + y$$

Duplicate x y

Three Variable Maps

- **A three variable Karnaugh Map is shown below:**

	yz=00	yz=01	yz=11	yz=10
x=0	m_0	m_1	m_3	m_2
x=1	m_4	m_5	m_7	m_6

- **Where each minterm corresponds to the product terms below:**

	yz=00	yz=01	yz=11	yz=10
x=0	$\overline{x}\,\overline{y}\,\overline{z}$	$\overline{x}\,\overline{y}\,z$	$\overline{x}\,y\,z$	$\overline{x}\,y\,\overline{z}$
x=1	$x\,\overline{y}\,\overline{z}$	$x\,\overline{y}\,z$	$x\,y\,z$	$x\,y\,\overline{z}$

- **Note that if the binary value for an index differs in one bit position, the minterms are adjacent on the Karnaugh Map**

Example Functions

- By convention, we represent the minterms by a "1" in the map and leave the other terms blank. A function table would also show the "0" terms.

Example:
$\sum$ m(2,3,4,5)

	yz=00	yz=01	yz=11	yz=10
x=0			1	1
x=1	1	1		

Example:
$\sum$ m(3,4,6,7)

	yz=00	yz=01	yz=11	yz=10
x=0			1	
x=1	1		1	1

Combining Squares

- By combining squares, we reduce the representation for a term, reducing the number of literals in the Boolean equation.
- On a three-variable K-Map:

  - **One square represents a minterm with three variables**
  - **Two adjacent squares represent a product term with two variables**
  - **Four “adjacent” terms represent a product term with one variable**
  - **Eight “adjacent” terms is the function of all ones (no variables) = 1.**

Combining Squares Example

- **Example: Let F = $\sum m(2,3,6,7)$**

F	yz=00	yz=01	yz=11	yz=10
x=0			1	1
x=1			1	1

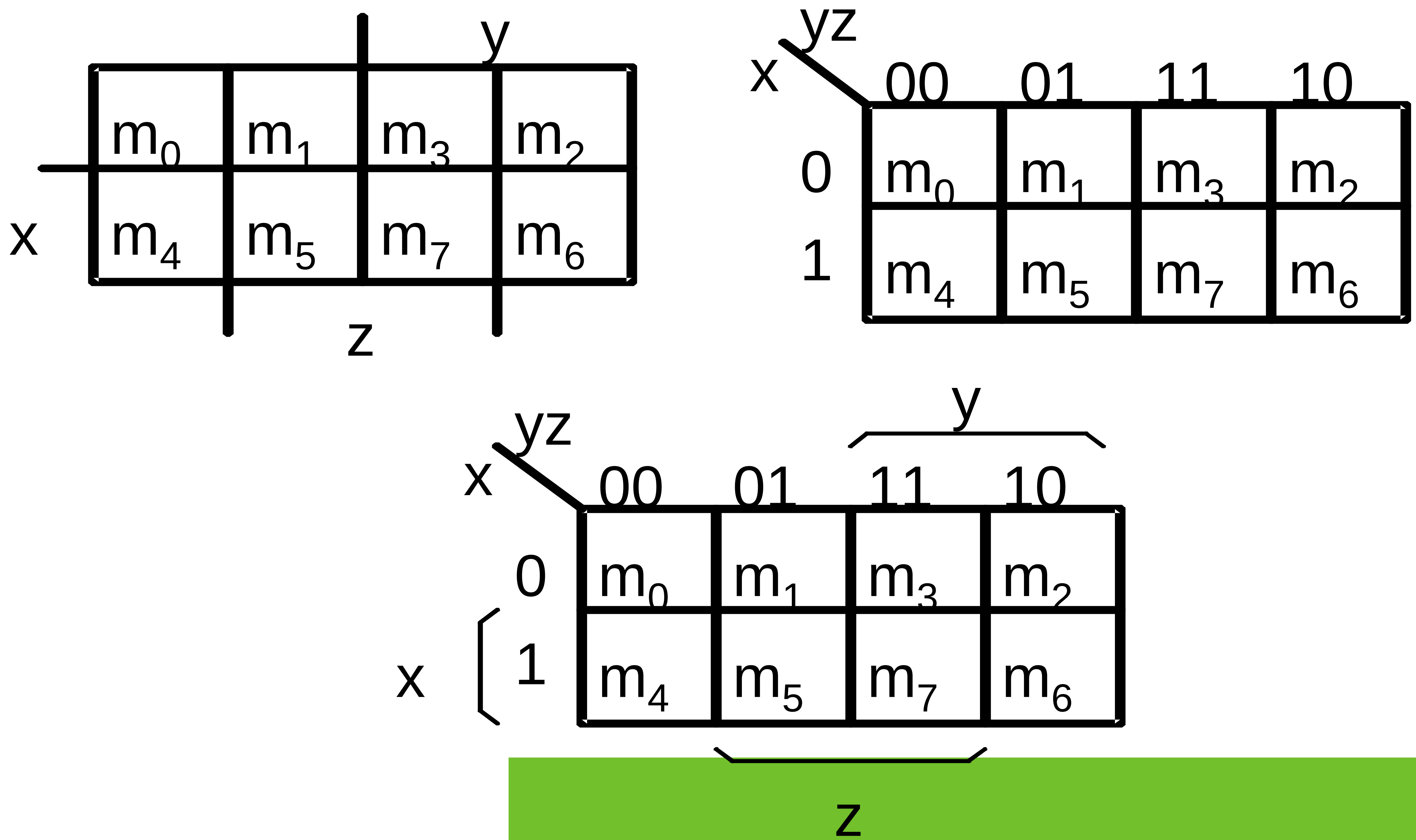
- **Applying the Minimization Theorem three times:**

$$\begin{aligned} F(x,y,z) &= \overline{x}\,y\,z + x\,y\,z + \overline{x}\,y\,\overline{z} + x\,y\,\overline{z} \\ &= yz + yz \\ &= y \end{aligned}$$

- **Thus the four terms that form a 2 × 2 square correspond to the term "y".**

Alternate K-Map Diagram

There are many ways to draw a three variable K-Map. Three common ways are shown below:



References

1. M.Morris Mano, “Computer System Architecture” 3rd Edition, Prentice Hall of India ,2000, ISBN-10: 0131663631
2. V.K. Puri, —DIGITAL ELECTRONICS CIRCUITS AND SYSTEMS” McGraw Hill Education (1 July 2017). ISBN-10: 9780074633175 , ISBN-13: 978-0074633175
3. William Stallings, “Computer Organization and Architecture, Designing for Performance” PHI/ Pearson Education North Asia Ltd., 10th Edition 2016, ISBN 978-0- 13-410161-3 — ISBN 0-13-410161-8.

Thank You