Computer Science Department

CS504

Digital Logic & Computer Organization

Lecture 4

Lecture Outline (Chapter 3)

- Circuit Optimization (Section 3.1)
  - Literal Cost
  - Gate Input Cost
  - Boolean Function Optimization

- Karnaugh Map (K-Map) (Section 3.2)
  - Two-Variable K-Map
  - Three-Variable K-Map

- Four-Variable K-Map (Section 3.3)
- Product-Of-Sums (POS) Simplification (Section 3.4)

Circuit Optimization

- **Goal: To obtain the simplest implementation for a given function**
- **Optimization is a more formal approach to simplification that is performed using a specific procedure or algorithm**
- **Optimization requires a cost criterion to measure the simplicity of a circuit**
- **Distinct cost criteria we will use:**
  - **Literal cost (L)**
  - **Gate input cost (G)**
  - **Gate input cost including inverters (GN)**

Literal Cost

- Literal – a variable or it complement
- Literal cost – the number of literal appearances in a Boolean expression corresponding to the logic circuit diagram
- Example: Boolean expressions for F
  - $F = B\,D + A\,\overline{B}\,C + A\,\overline{C}\,\overline{D}$ L = 8
  - $F = B\,D + A\,\overline{B}\,C + A\,\overline{B}\,\overline{D} + A\,B\,\overline{C}$ L = 11
  - $F = (A+B)(A+D)(B+C+\overline{D})(\overline{B}+\overline{C}+D)$ L = 10
  - Which solution is best? First solution is best

Gate Input Cost

- Gate Input Cost: Count of total number of inputs to the gates in the logic circuit implementation
- Two gate input costs are defined:

  G = Count of gate inputs without counting Inverters

  GN = Count of gate inputs + count of Inverters
- For SOP and POS equations, the gate input cost can be found from the Boolean expression by finding the sum of:
  - All literal appearances
  - Number of terms excluding single literal terms (added to G)
  - Number of distinct complemented single literals (added to GN)
- Example:
  - $F = B\,D + A\,\overline{B}\,C + A\,\overline{C}\,\overline{D}$

L = 8 G = L+3 = 11

GN = G+3 = 14

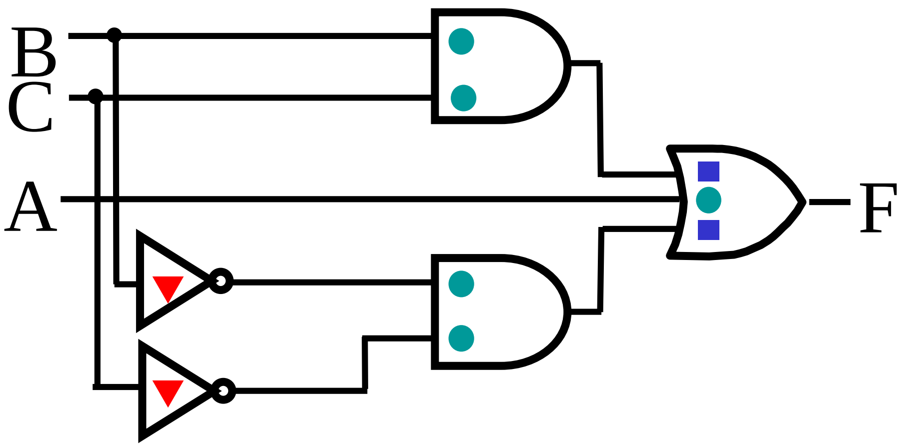
Cost Criteria (continued)

- Example 1:
- $F = A + B\,C + \bar{B}\,\bar{C}$

$GN = G + 2 = 9$

$L = 5$

$G = L + 2 = 7$



- L (literal count) counts the AND inputs and the single literal OR input.
- G (gate input count) adds the remaining OR gate inputs
- GN(gate input count with NOTs) adds the inverter inputs

Cost Criteria (continued)

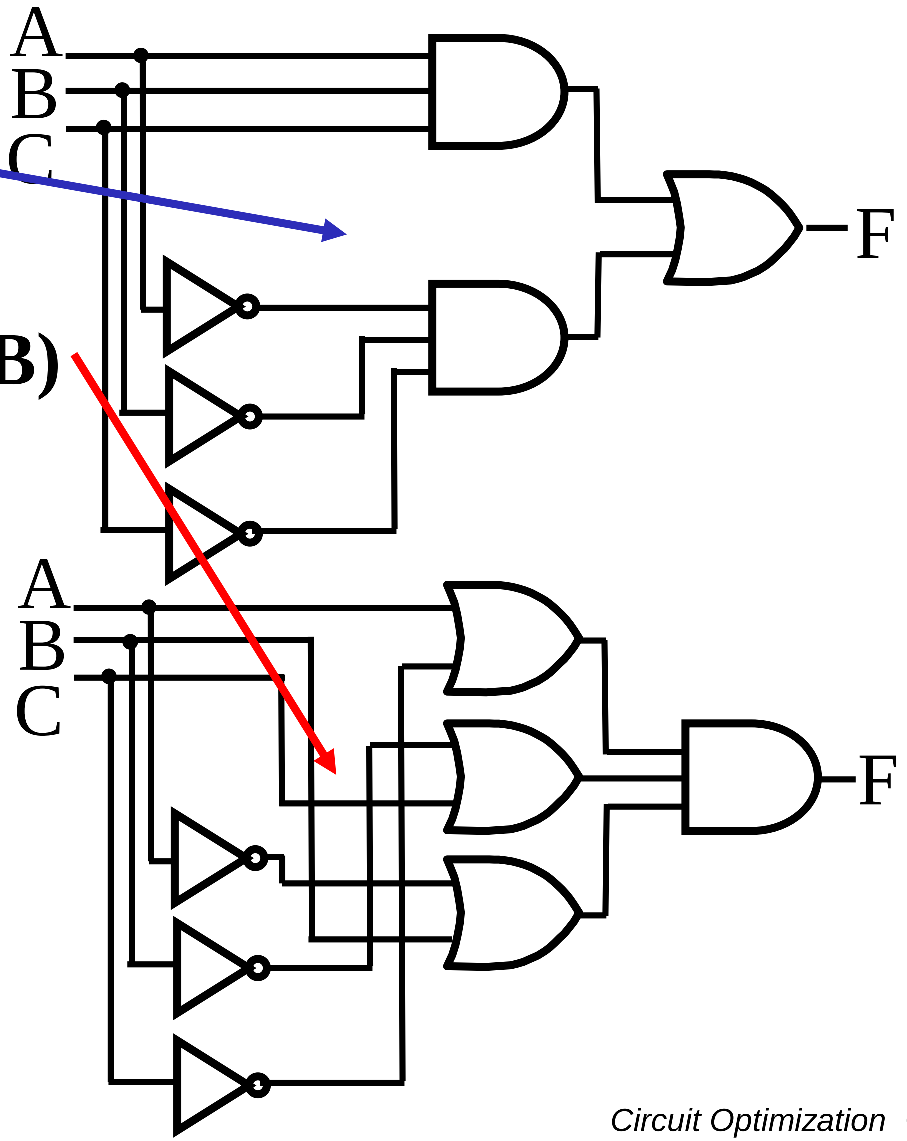
- **Example 2:**
- **$F = A\,B\,C + \bar{A}\,\bar{B}\,\bar{C}$**

  $L = 6 \quad G = 8 \quad GN = 11$

- **$F = (A + \bar{C})(\bar{B} + C)(\bar{A} + B)$**

  $L = 6 \quad G = 9 \quad GN = 12$

- **Same function and same literal cost**
- **But first circuit has better gate input count and better gate input count with NOTs**
- **Select first circuit!**



Cost Criteria Summary

- Literal Count:
  - Simple to evaluate by counting all literals
  - However, does not represent circuit complexity accurately in all cases
- Gate Input Cost (or Count):
  - Good measure of logic implementation
  - Proportional to the number of transistors and wires used in the implementation
  - Important when measuring cost of circuits with more than two levels

Boolean Function Optimization

- Minimizing the gate input (or literal) cost of a Boolean equation reduces the circuit cost
- We choose gate input cost
- Boolean Algebra and graphical techniques are tools to minimize cost criteria values
- Some important questions:
  - When do we stop trying to reduce the cost?
  - Do we know when we have a minimum cost?
- Treat optimum or near-optimum cost functions for two-level (SOP and POS) circuits first
- Introduce a graphical technique using Karnaugh maps (K-maps for short)

Karnaugh Map (K-map)

- A K-map is a collection of squares
  - Each square represents a minterm
  - The collection of squares is a graphical representation of a Boolean function
  - Adjacent squares differ in the value of one variable
  - Alternative algebraic expressions for the same function are derived by recognizing patterns of squares
- The K-map can be viewed as
  - A reorganized version of the truth table

Some Uses of K-Maps

- **Provide a means for:**
  - **Finding optimum or near optimum**
    - **SOP and POS standard forms**
    - **Two-level AND/OR and OR/AND circuits**

  for functions with small numbers of variables

  - **Visualizing concepts related to manipulating Boolean expressions, and**
  - **Demonstrating concepts used by computer-aided design programs to simplify large circuits**

Two-Variable K-Map

- A 2-variable Karnaugh Map:
  - Minterm m_0 and minterm m_1 are "adjacent" they differ in the value of variable y
  - Similarly, minterm m_0 and minterm m_2 are adjacent and differ in x
  - Also, m_1 and m_3 differ in the x variable
  - Finally, m_2 and m_3 differ in the y variable

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

K-Map and Truth Tables

- The K-Map is a different form of the truth table

Truth Table

Input Values (x,y)	Function Value F(x,y)
0 0	1
0 1	0
1 0	1
1 1	1

K-Map

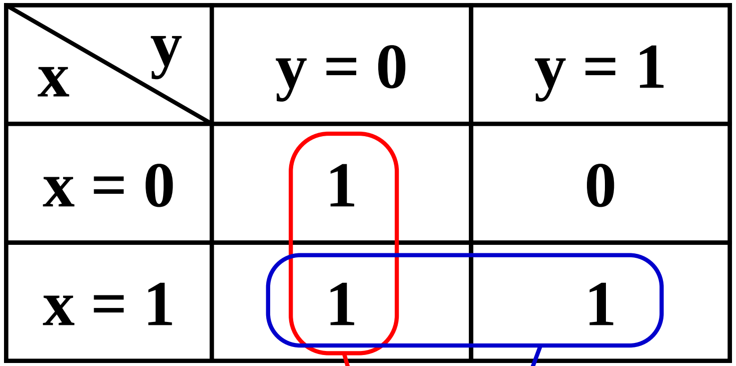
y x	y = 0	y = 1
x = 0	1	0
x = 1	1	1

K-Map Function Minimization

- $F(x,y) = m_0 + m_2 + m_3$

  $F = \bar{x}\,\bar{y} + x\,\bar{y} + x\,y$

K-Map



x \ y	y = 0	y = 1
x = 0	1	0
x = 1	1	1

- Two adjacent cells containing 1's can be combined using the Minimization Theorem

- $m0 + m2 = \bar{x}\,\bar{y} + x\,\bar{y} = (\bar{x} + x)\,\bar{y} = \bar{y}$
- $m2 + m3 = x\,\bar{y} + x\,y = x\,(\bar{y} + y) = x$
- Therefore, F can be simplified as $F = x + \bar{y}$

Three Variable K-Map

yz x	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

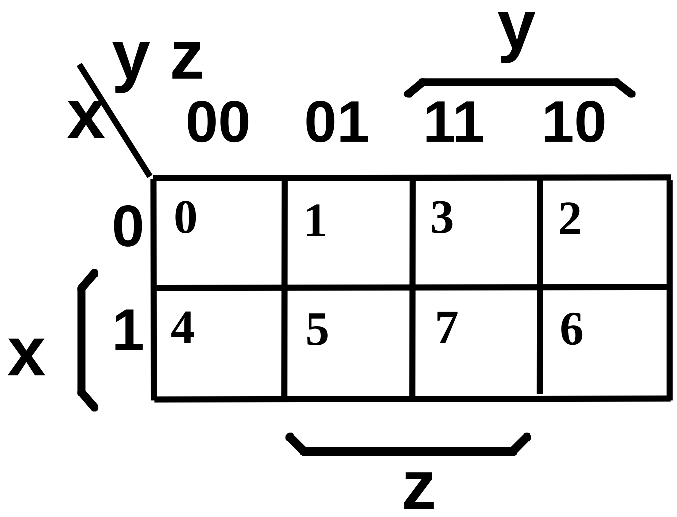
- **Where each minterm corresponds to the product terms:**

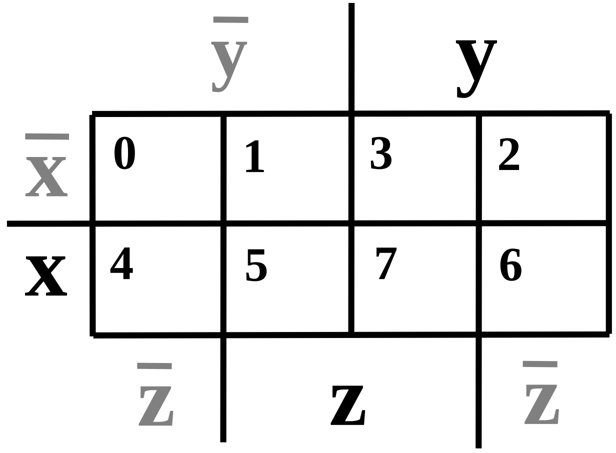
yz x	yz=00	yz=01	yz=11	yz=10
x=0	$\overline{x}\,\overline{y}\,\overline{z}$	$\overline{x}\,\overline{y}\,z$	$\overline{x}\,y\,z$	$\overline{x}\,y\,\overline{z}$
x=1	$x\,\overline{y}\,\overline{z}$	$x\,\overline{y}\,z$	$x\,y\,z$	$x\,y\,\overline{z}$

- **Note that if the binary value for an <u>index</u> differs in one bit position, the minterms are adjacent on the K-Map**

Alternative K-Map Labeling

- K-Map largely involves:
  - Entering values into the map, and
  - Reading off product terms from the map
- Alternate labelings are useful:

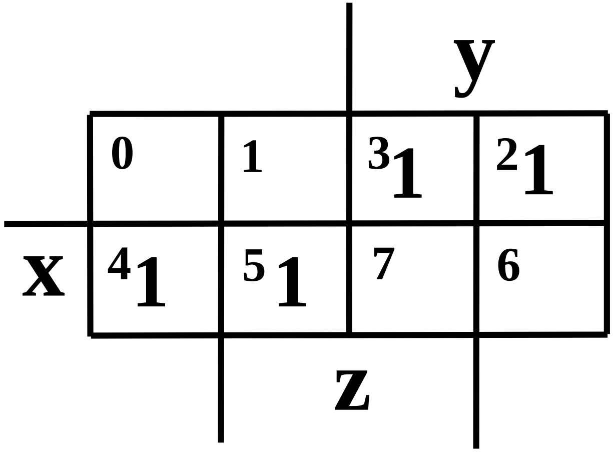




Example Functions

- By convention, we represent the minterms of F by a ‘1’ in the map and leave the entries that contain ‘0’ blank
- Example 1:

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

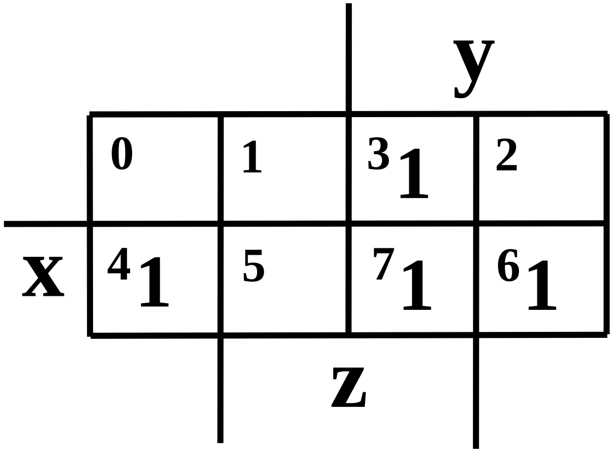


	^{0}	^{1}	$^{3}1$	$^{2}1$
x	$^{4}1$	$^{5}1$	^{7}	^{6}

- Example 2:

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

- Learn the locations of the 8 indices based on the variable order shown



	^{0}	^{1}	$^{3}1$	^{2}
x	$^{4}1$	^{5}	$^{7}1$	$^{6}1$

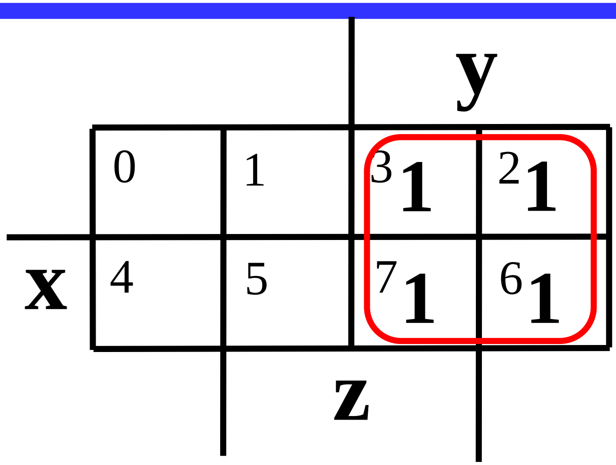
Combining Squares

- By combining squares, we reduce number of literals in a product term, thereby reducing the gate input cost
- On a 3-variable K-Map:
  - One square represents a minterm with 3 variables
  - Two adjacent squares represent a term with 2 variables
  - Four adjacent squares represent a term with 1 variable
  - Eight adjacent square is the function 1 (no variables)

Example: Combining Squares

- Example:

$$F = \sum(2, 3, 6, 7)$$



	0	1	3 1	2 1
x	4	5	7 1	6 1

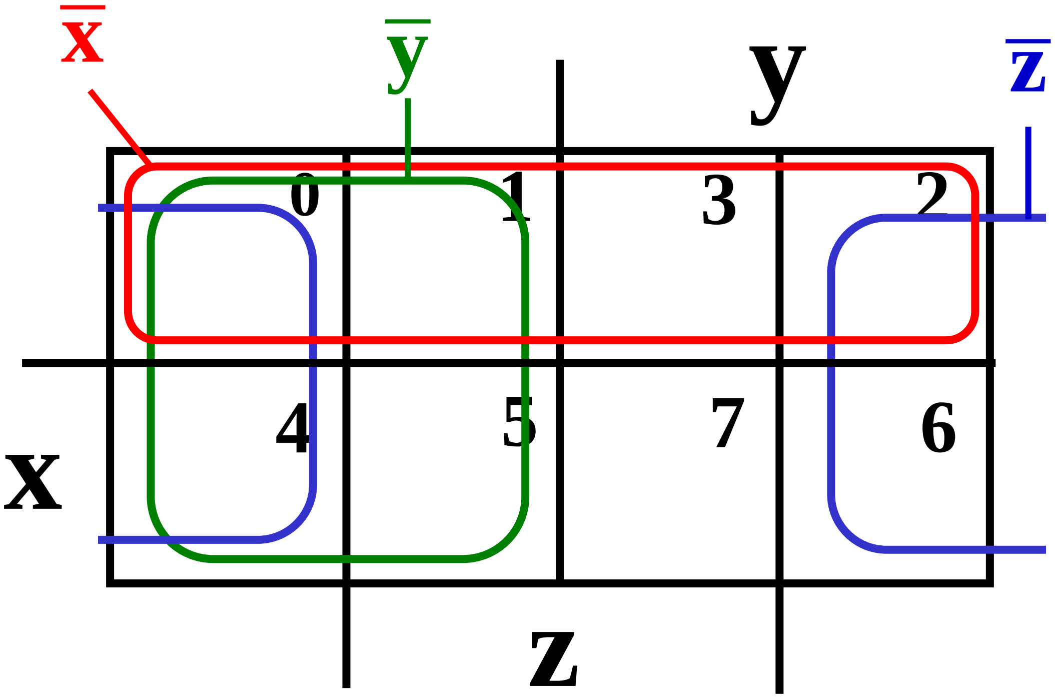
- Applying the Minimization Theorem 3 times:

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

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

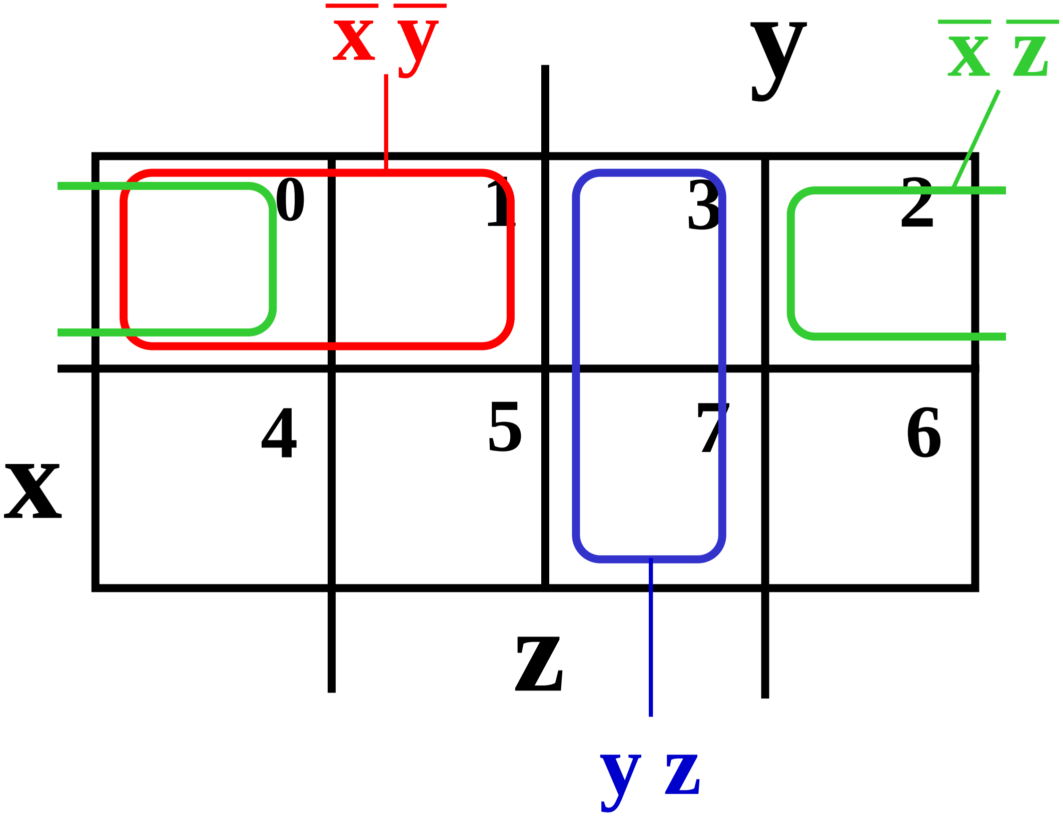
Combining Four Squares

- **Example Shapes of 4-square Rectangles:**



Combining Two Squares

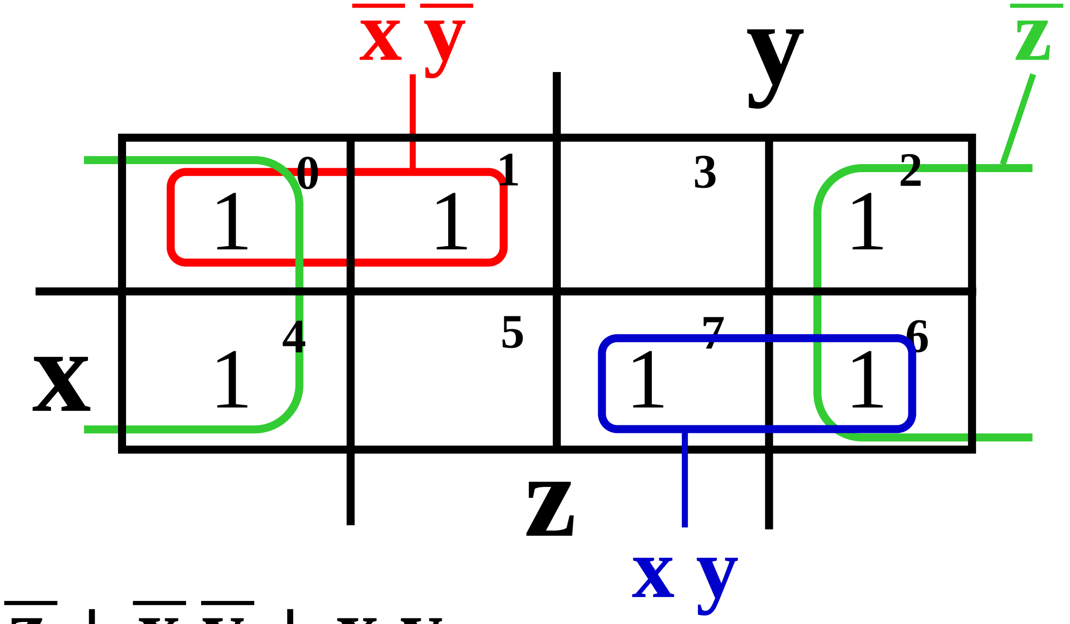
- Example Shapes of 2-square Rectangles:



Simplifying 3-Variable Functions

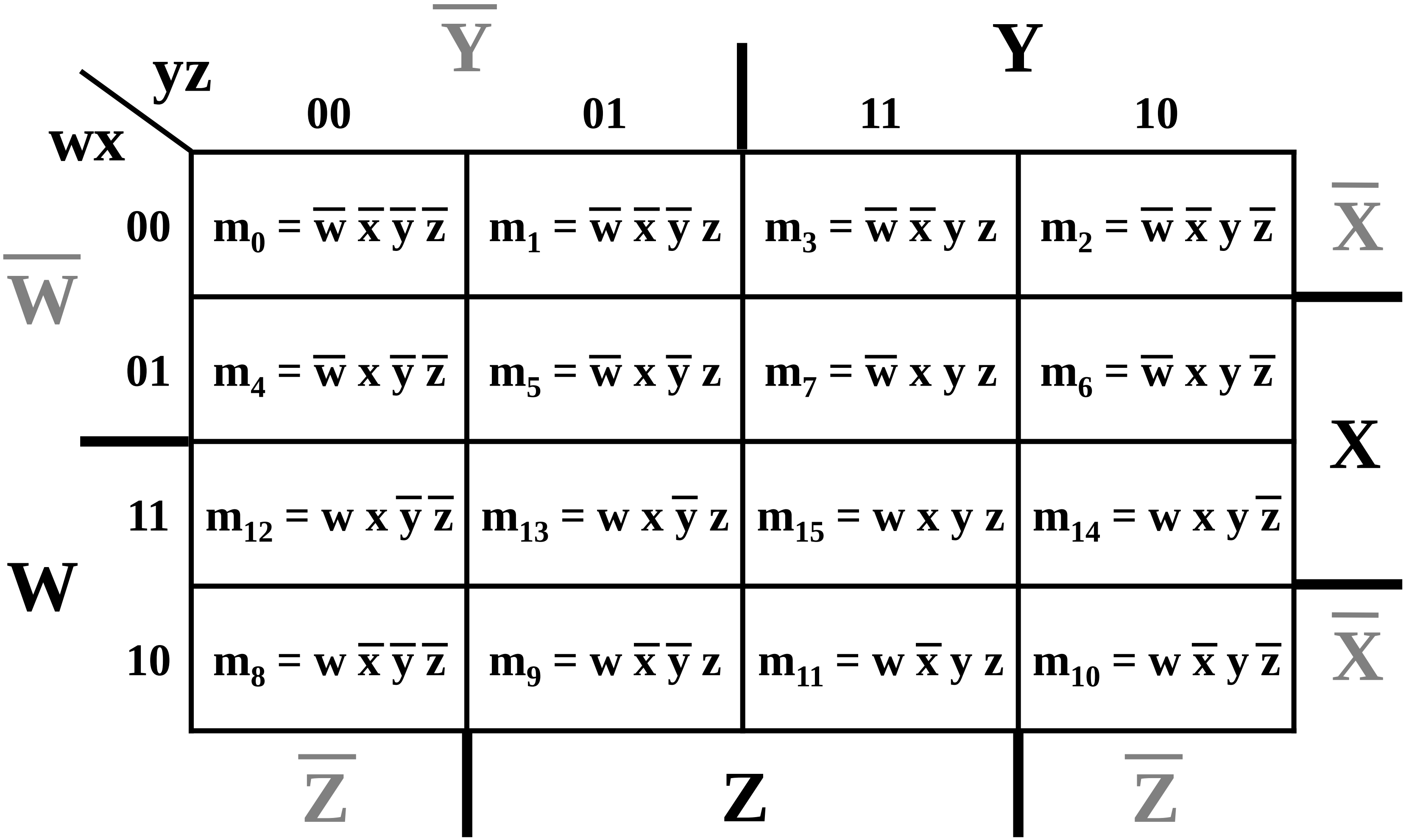
- K-Maps can be used to simplify Boolean functions
- Example: find an optimum SOP equation for

$F(x, y, z) = \sum (0, 1, 2, 4, 6, 7)$



$F = \overline{z} + \overline{x}\,\overline{y} + x\,y$

Four-Variable K-Map



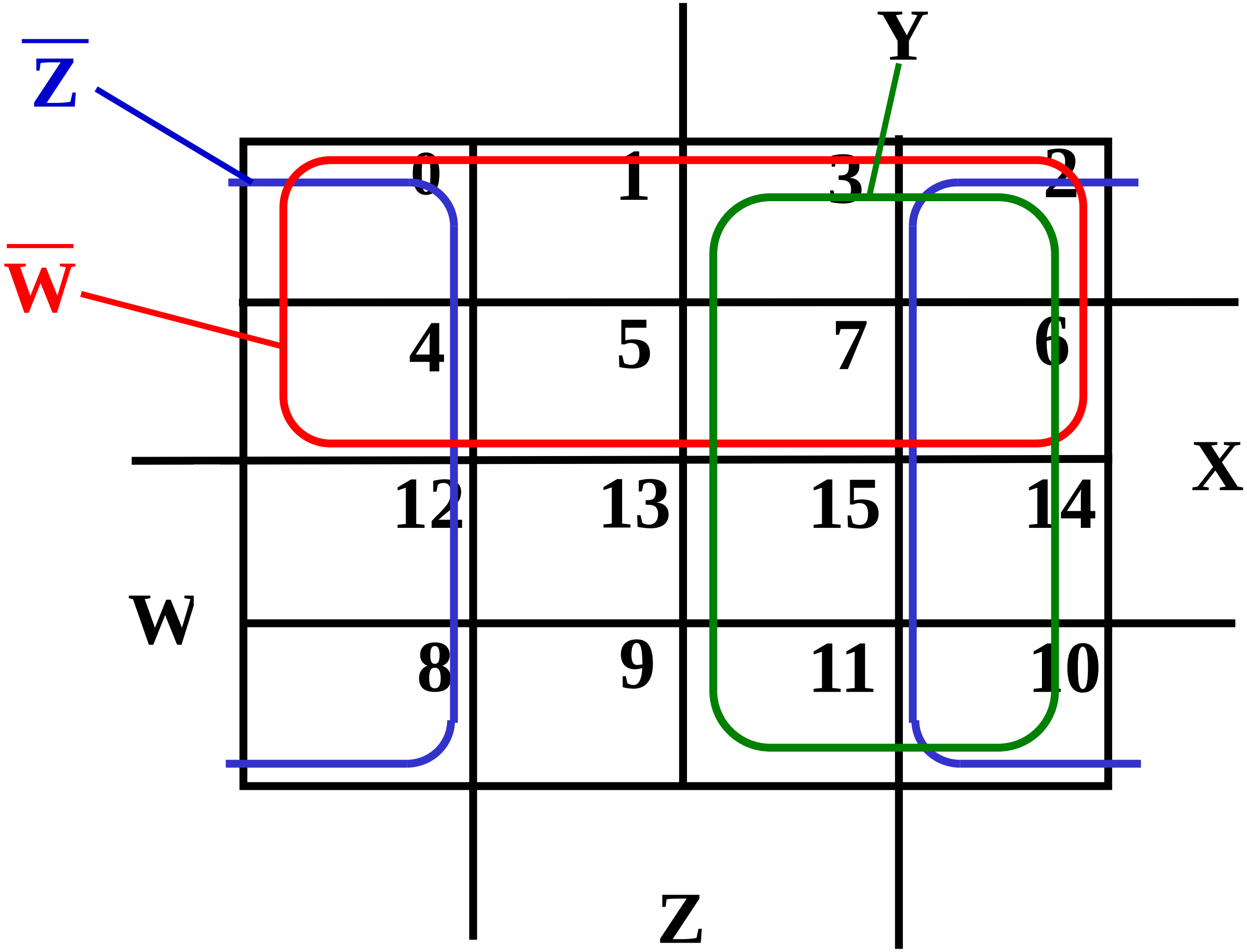
wx \ yz	00	01	11	10
00	$m_0 = \overline{w}\,\overline{x}\,\overline{y}\,\overline{z}$	$m_1 = \overline{w}\,\overline{x}\,\overline{y}\,z$	$m_3 = \overline{w}\,\overline{x}\,y\,z$	$m_2 = \overline{w}\,\overline{x}\,y\,\overline{z}$
01	$m_4 = \overline{w}\,x\,\overline{y}\,\overline{z}$	$m_5 = \overline{w}\,x\,\overline{y}\,z$	$m_7 = \overline{w}\,x\,y\,z$	$m_6 = \overline{w}\,x\,y\,\overline{z}$
11	$m_{12} = w\,x\,\overline{y}\,\overline{z}$	$m_{13} = w\,x\,\overline{y}\,z$	$m_{15} = w\,x\,y\,z$	$m_{14} = w\,x\,y\,\overline{z}$
10	$m_8 = w\,\overline{x}\,\overline{y}\,\overline{z}$	$m_9 = w\,\overline{x}\,\overline{y}\,z$	$m_{11} = w\,\overline{x}\,y\,z$	$m_{10} = w\,\overline{x}\,y\,\overline{z}$

4-Variable K-map Terms

- 4-variable K-maps can have rectangles corresponding to:
  - Single square = 4-variable minterm
  - 2 combined squares = 3-variable term
  - 4 combined squares = 2-variable term
  - 8 combined squares = 1 variable term
  - 16 (all) combined squares = constant ‘1’

Combining Eight Squares

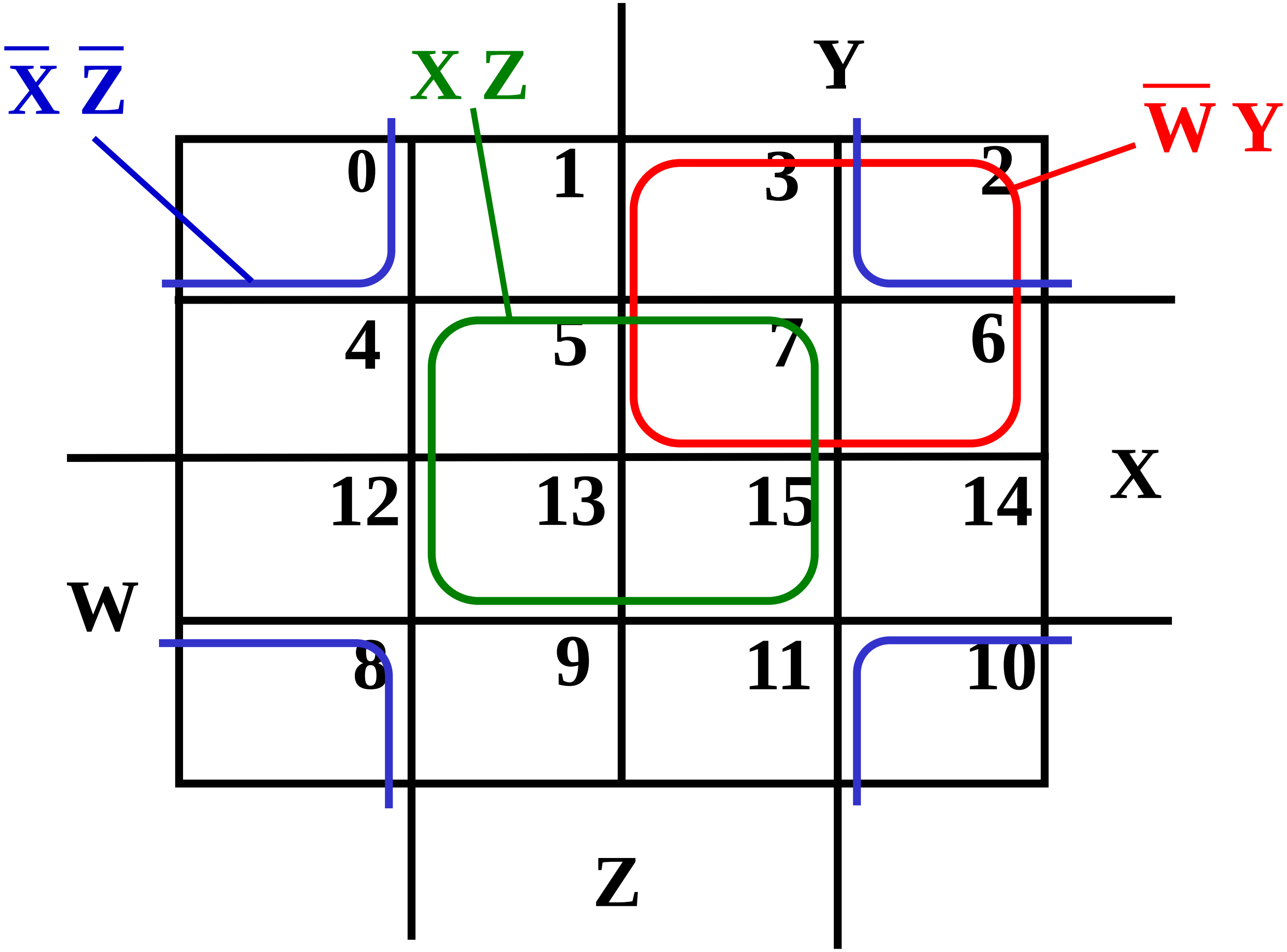
- **Examples of 8-square Rectangles:**



0	1	3	2
4	5	7	6
12	13	15	14
8	9	11	10

Combining Four Squares

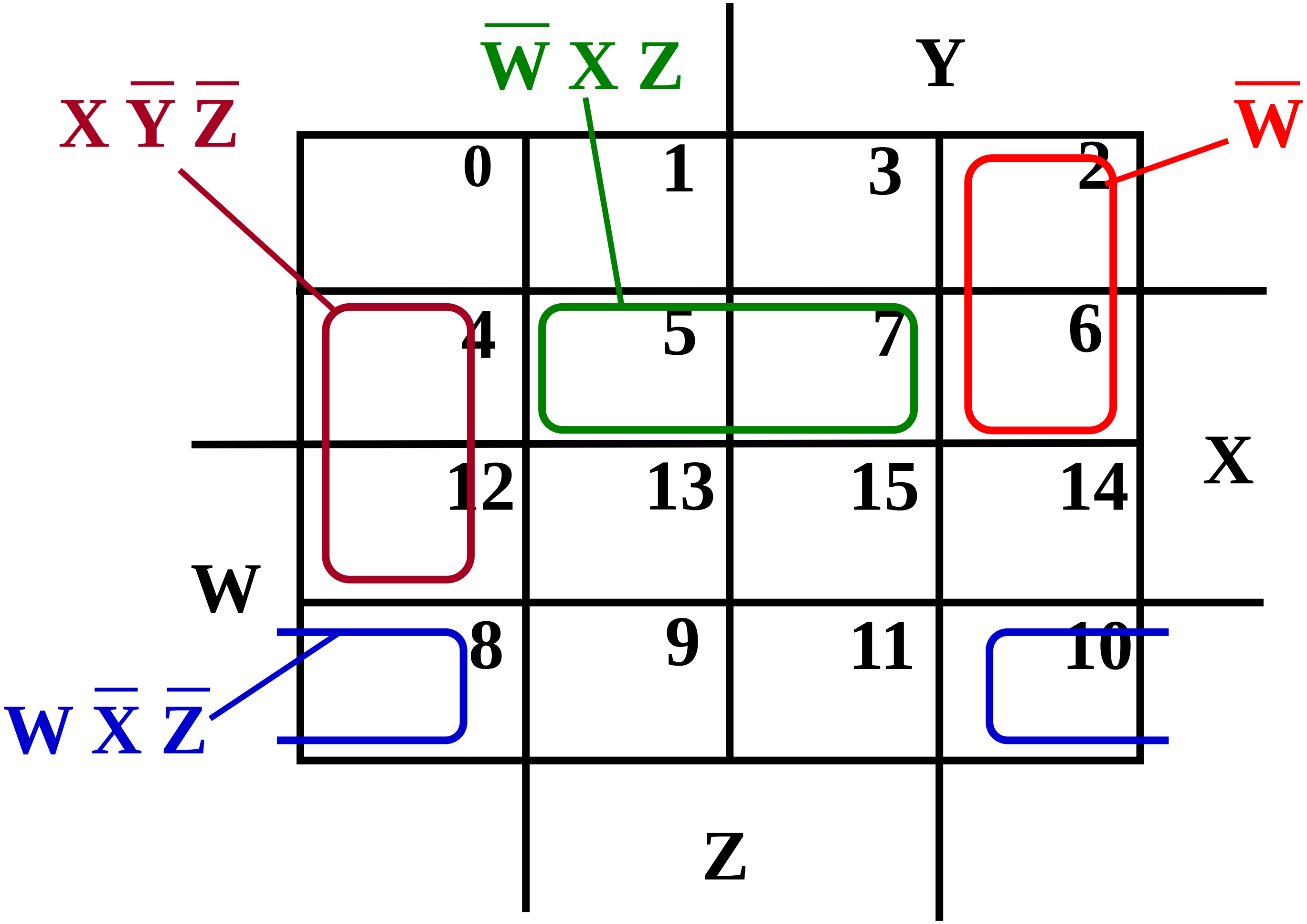
- Examples of 4-square Rectangles:



0	1	3	2
4	5	7	6
12	13	15	14
8	9	11	10

Combining Two Squares

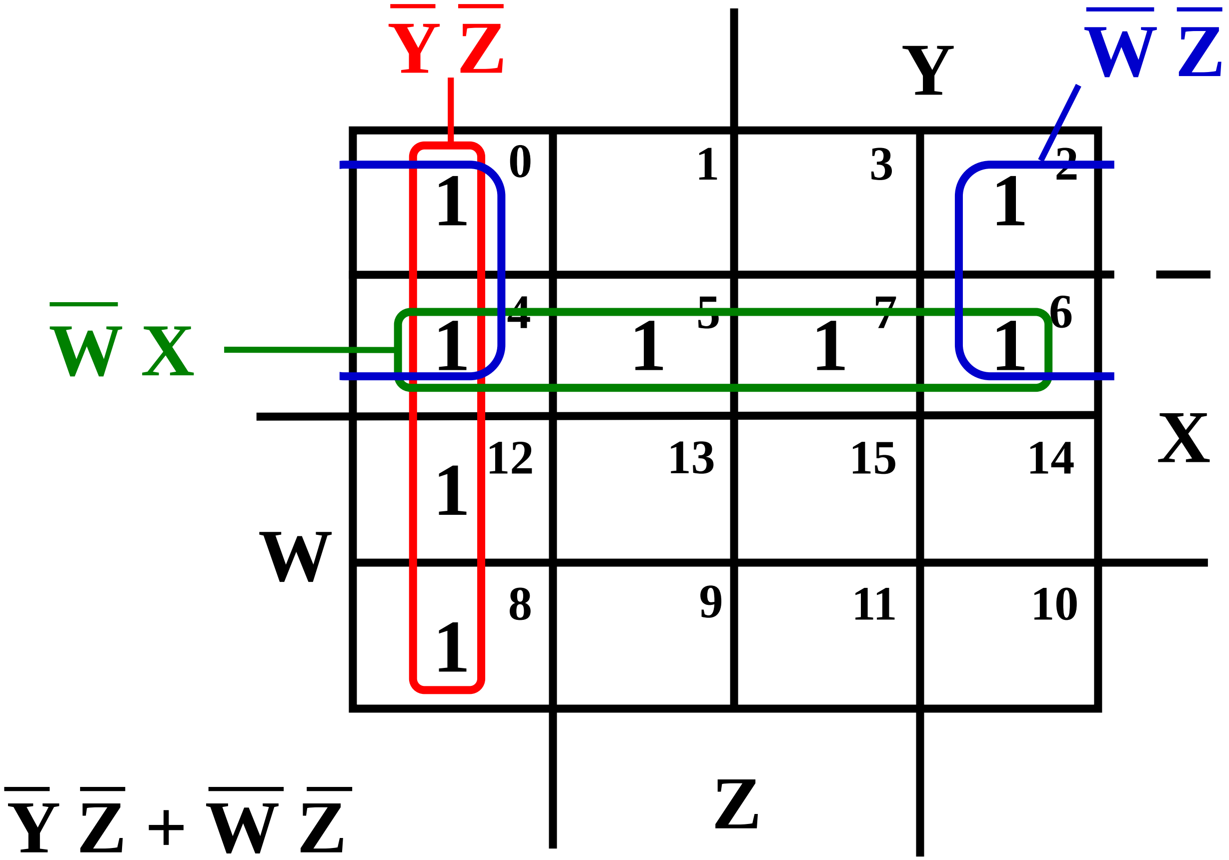
- **Examples of 2-square Rectangles:**


$\overline{W}XZ$
Y
$X\overline{Y}\overline{Z}$
$\overline{W}Y\overline{Z}$
X
W
$W\overline{X}\overline{Z}$
Z

0	1	3	2
4	5	7	6
12	13	15	14
8	9	11	10

Simplifying 4-Variable Functions

- $F(W, X, Y, Z) = \sum (0, 2, 4, 5, 6, 7, 8, 12)$



$$F = \overline{W}\,X + \overline{Y}\,\overline{Z} + \overline{W}\,\overline{Z}$$

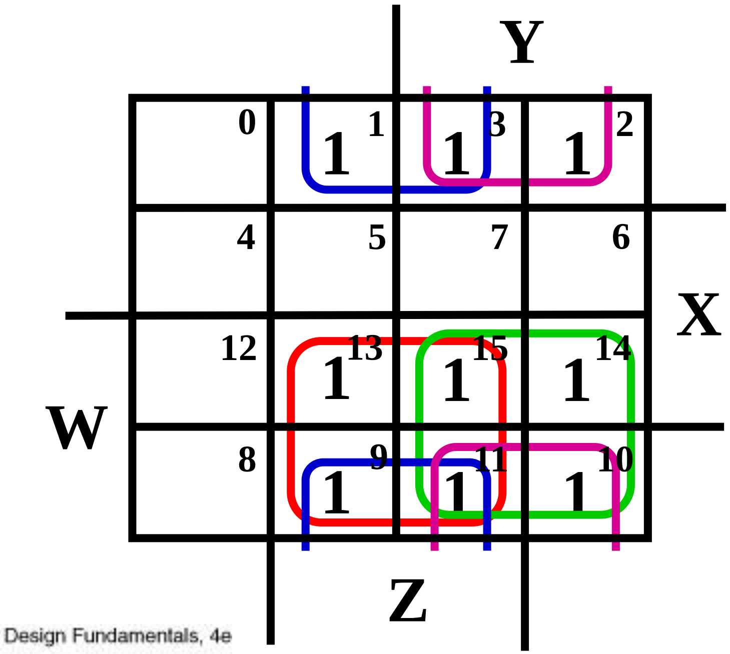
Product-of-Sums Simplification

- Step 1: Draw the K-map for $\overline{F}$, replacing the 0's of F with 1's in $\overline{F}$ and vice versa
- Step 2: Obtain a minimal Sum-of-Product (SOP) expression for $\overline{F}$
- Step 3: Use DeMorgan's Theorem to obtain $F = \overline{\overline{F}}$

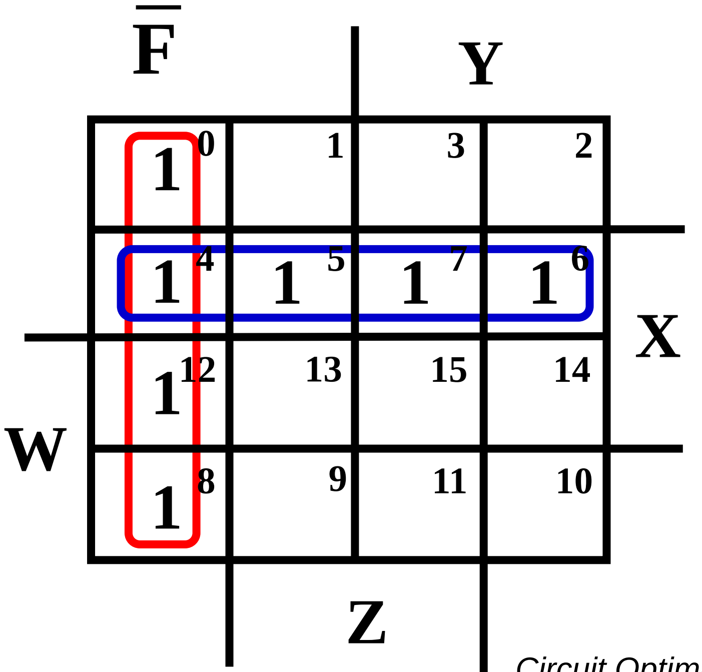
  The result is a minimal Product-of-Sum (POS) expression for F
- Step 4: Compare the cost of the minimal SOP and POS expressions to find which one is better

Product-Of-Sums Simplification

- $F(W, X, Y, Z) = \sum(1, 2, 3, 9, 10, 11, 13, 14, 15)$
- $F = W\,Z + W\,Y + \overline{X}\,Z + \overline{X}\,Y$ (G = 8+4 = 12)
- $\overline{F}(W, X, Y, Z) = \sum(0, 4, 5, 6, 7, 8, 12) = \overline{Y}\,\overline{Z} + \overline{W}\,X$
- $F = \overline{\overline{Y}\,\overline{Z} + \overline{W}\,X} = (Y + Z)(W + \overline{X})$ (G = 4+2 = 6)



0	1 1	3 1	2 1
4	5	7	6
12	13 1	15 1	14 1
8	9 1	11 1	10 1



0 1	1	3	2
4 1	5 1	7 1	6 1
12 1	13	15	14
8 1	9	11	10

The End

Questions?