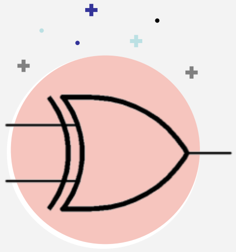
V1 (@2025)



Digital Circuits

Lecture 4:

Karnaugh Map

By: M.Razeghizadeh

Start now!

Contents of This Slide

- Boolean Function Minimization
- The Karnaugh Map (K-Map)
- Two, Three, and Four-Variable K-Maps
- Prime and Essential Prime Implicants
- Minimal Sum-of-Products and Product-of-Sums
- Don't Cares
- Five and Six-Variable K-Maps
- Multiple Outputs

Boolean Function Minimization

- Complexity of a Boolean function is directly related to the complexity of the algebraic expression
- The truth table of a function is unique
- However, the algebraic expression is not unique
- Boolean function can be simplified by algebraic manipulation
- However, algebraic manipulation depends on experience
- Algebraic manipulation does not guarantee that the simplified Boolean expression is minimal

Example: Sum of Minterms

Truth Table

x y z	f	Minterm
0 0 0	0	
0 0 1	1	$m_1 = x'y'z$
0 1 0	1	$m_2 = x'yz'$
0 1 1	1	$m_3 = x'yz$
1 0 0	0	
1 0 1	1	$m_5 = xy'z$
1 1 0	0	
1 1 1	1	$m_7 = xyz$

Focus on the '**1**' entries

$$f = m_1 + m_2 + m_3 + m_5 + m_7$$

$$f = \sum(1, 2, 3, 5, 7)$$

$$f = x'y'z + x'yz' + \\ x'yz + xy'z + xyz$$

- Sum-of-Minterms has 15 literals ➔ Can be simplified

Algebraic Manipulation

- **Simplify:** $f = x'y'z + x'yz' + x'yz + xy'z + xyz$ (15 literals)

$f = x'y'z + x'yz' + x'yz + xy'z + xyz$	(Sum-of-Minterms)
$f = x'y'z + x'yz + x'yz' + xy'z + xyz$	Reorder
$f = x'z(y' + y) + x'yz' + xz(y' + y)$	Distributive $\cdot$ over $+$
$f = x'z + x'yz' + xz$	Simplify (7 literals)
$f = x'z + xz + x'yz'$	Reorder
$f = (x' + x)z + x'yz'$	Distributive $\cdot$ over $+$
$f = z + x'yz'$	Simplify (4 literals)
$f = (z + x'y)(z + z')$	Distributive $+$ over $\cdot$
$f = z + x'y$	Simplify (3 literals)

Drawback of Algebraic Manipulation

- No clear steps in the manipulation process
  - Not clear which terms should be grouped together
  - Not clear which property of Boolean algebra should be used next
- Does not always guarantee a minimal expression
  - Simplified expression may or may not be minimal
  - Different steps might lead to different non-minimal expressions
- However, the goal is to minimize a Boolean function
- Minimize the **number of literals** in the Boolean expression
  - The **literal count** is a good measure of the **cost** of logic implementation
  - Proportional to the number of transistors in the circuit implementation

Karnaugh Map

- Called also K-map for short
- The Karnaugh map is a diagram made up of squares
- It is a reorganized version of the truth table
- Each square in the Karnaugh map represents a minterm
- Adjacent squares differ in the value of one variable
- Simplified expressions can be derived from the Karnaugh map
  - By recognizing patterns of squares
- Simplified sum-of-products expression (AND-OR circuits)
- Simplified product-of-sums expression (OR-AND circuits)

Two-Variable Karnaugh Map

- Minterms m_0 and m_1 are adjacent (also, m_2 and m_3)
  - They differ in the value of variable y
- Minterms m_0 and m_2 are adjacent (also, m_1 and m_3)
  - They differ in the value of variable x

Two-variable K-map

$x \backslash y$	0	1
0	m_0	m_1
1	m_2	m_3

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

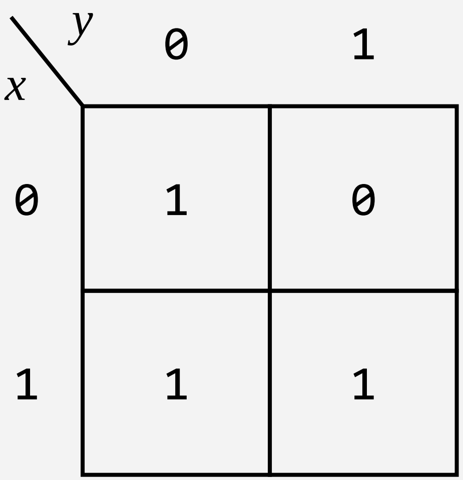
From a Truth Table to Karnaugh Map

- Given a truth table, construct the corresponding K-map
- Copy the function values from the truth table into the K-map
- Make sure to copy each value into the proper K-map square

Truth Table

x y	f
0 0	1
0 1	0
1 0	1
1 1	1

K-map

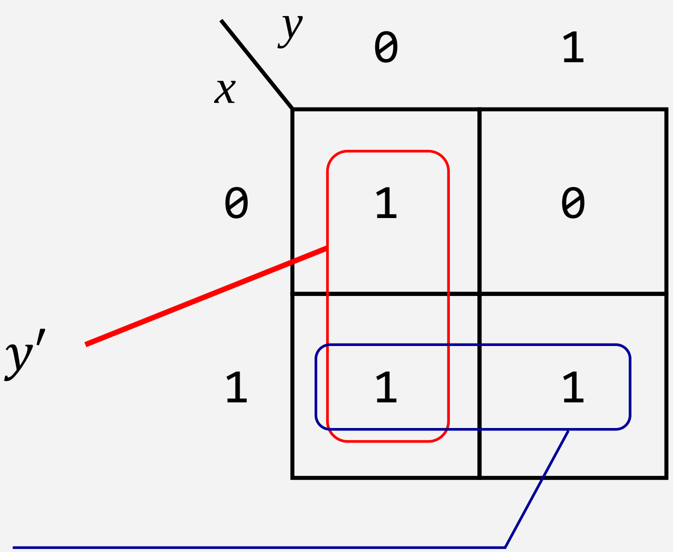


$x \backslash y$	0	1
0	1	0
1	1	1

K-Map Function Minimization

- Two adjacent cells containing 1's can be combined
- $f = m_0 + m_2 + m_3$
- $f = x'y' + xy' + xy$ (6 literals)
- $m_0 + m_2 = x'y' + xy' = (x' + x)y' = y'$
- $m_2 + m_3 = xy' + xy = x(y' + y) = x$
- Therefore, f can be simplified as: $f = x + y'$ (2 literals)

K-map



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

Three-Variable Karnaugh Map

- Have eight squares (for the 8 minterms), numbered 0 to 7
- The last two columns are not in numeric order: 11, 10
  - Remember the numbering of the squares in the K-map
- Each square is adjacent to three other squares
- Minterms in adjacent squares can always be combined
  - This is the key idea that makes the K-map work
- Labeling of rows and columns is also useful

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

		y'		y	
x \ yz		00	01	11	10
x'	0	$x'y'z'$	$x'y'z$	$x'yz$	$x'yz'$
x	1	$xy'z'$	$xy'z$	xyz	xyz'
		z'	z		z'

Simplifying a Three-Variable Function

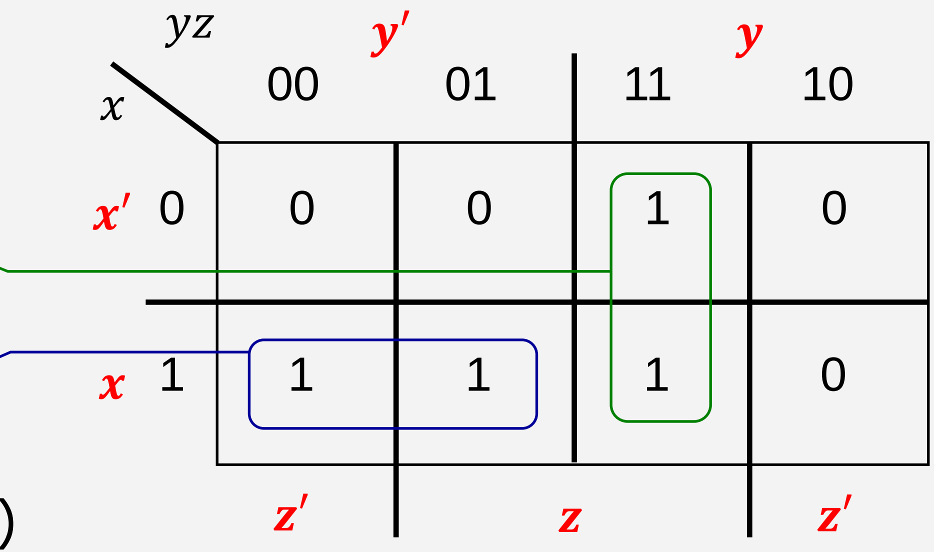
Simplify the Boolean function: $f(x, y, z) = \sum(3, 4, 5, 7)$

$f = x'yz + xy'z' + xy'z + xyz$ (12 literals)

1. Mark '**1**' all the K-map squares that represent function f
2. Find possible adjacent squares

$x'yz + xyz = (x' + x)yz = yz$

$xy'z' + xy'z = xy'(z' + z) = xy'$



x \ yz	00	01	11	10
x' 0	0	0	1	0
x 1	1	1	1	0

Therefore, $f = xy' + yz$ (4 literals)

Simplifying a Three-Variable Function (2)

Here is a second example: $f(x, y, z) = \sum(3, 4, 6, 7)$

$f = x'yz + xy'z' + xyz' + xyz$ (12 literals)

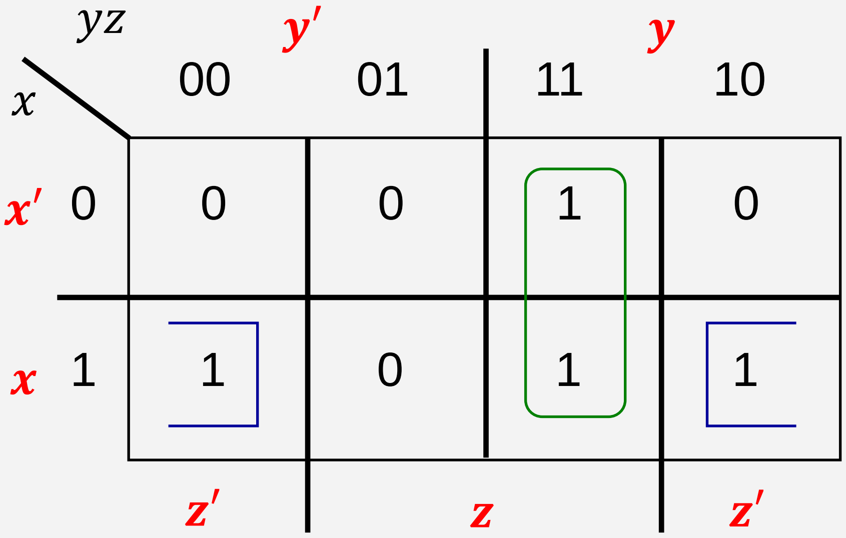
Learn the locations of the 8 indices based on the variable order

$x'yz + xyz = (x' + x)yz = yz$

Corner squares can be combined

$xy'z' + xyz' = xz'(y' + y) = xz'$

Therefore, $f = xz' + yz$ (4 literals)



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

Combining Squares on a 3-Variable K-Map

- By combining squares, we reduce number of literals in a product term, thereby reducing the 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 constant '**1**' (no variables)

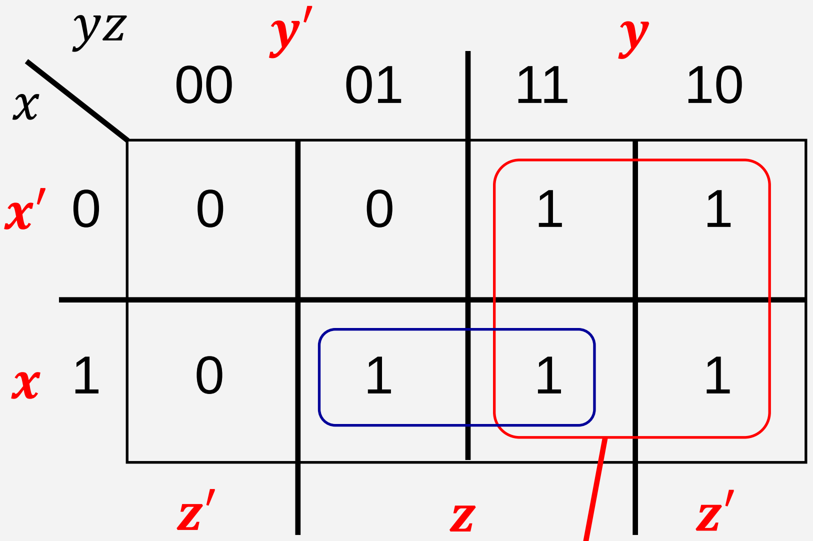
Example of Combining Squares

- Consider the Boolean function: $f(x, y, z) = \sum(2, 3, 5, 6, 7)$
- $f = x'yz' + x'yz + xy'z + xyz' + xyz$

❖ The four minterms that form the 2×2 red square are reduced to the term y

❖ The two minterms that form the blue rectangle are reduced to the term xz

❖ Therefore: $f = y + xz$



x \ yz	00	01	11	10
x' 0	0	0	1	1
x 1	0	1	1	1

$$x'yz + x'yz' + xyz + xyz'$$
$$= x'y(z + z') + xy(z + z')$$
$$= x'y + xy = (x' + x)y = y$$

Minimal Sum-of-Products Expression

Consider the function: $f(x, y, z) = \sum(0, 1, 2, 4, 6, 7)$

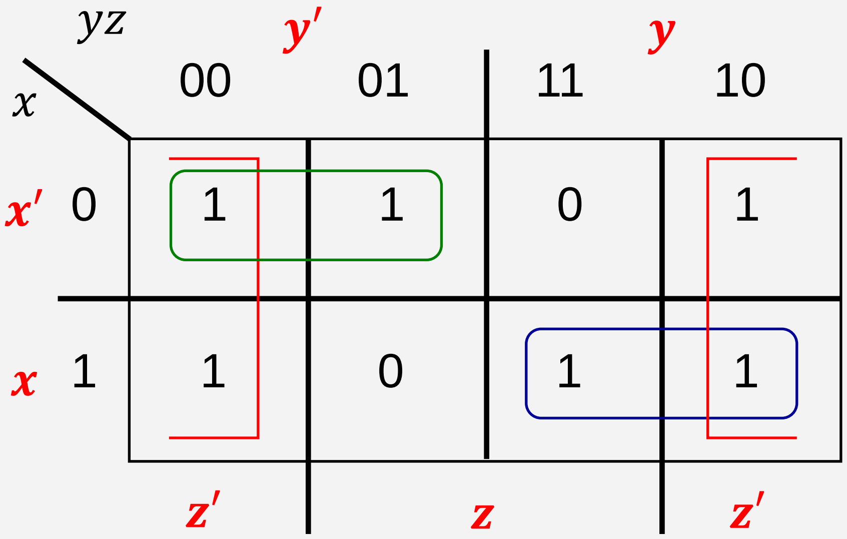
Find a minimal sum-of-products (SOP) expression

Solution:

Red block: term = z'

Green block: term = $x'y'$

Blue block: term = xy



x \ yz	00	01	11	10
x' 0	1	1	0	1
x 1	1	0	1	1

Minimal sum-of-products: $f = z' + x'y' + xy$ (5 literals)

Four-Variable Karnaugh Map

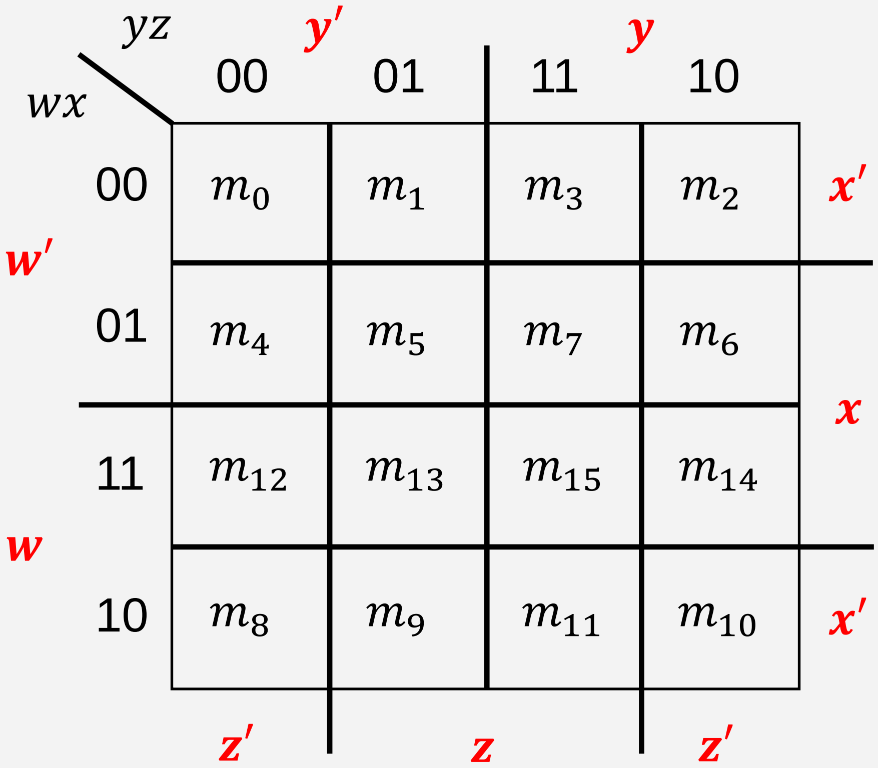
Notice the order of Rows 11 and 10
and the order of columns 11 and 10

4 variables ➔ 16 squares

Remember the numbering of the squares in the K-map

Each square is adjacent to four other squares

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

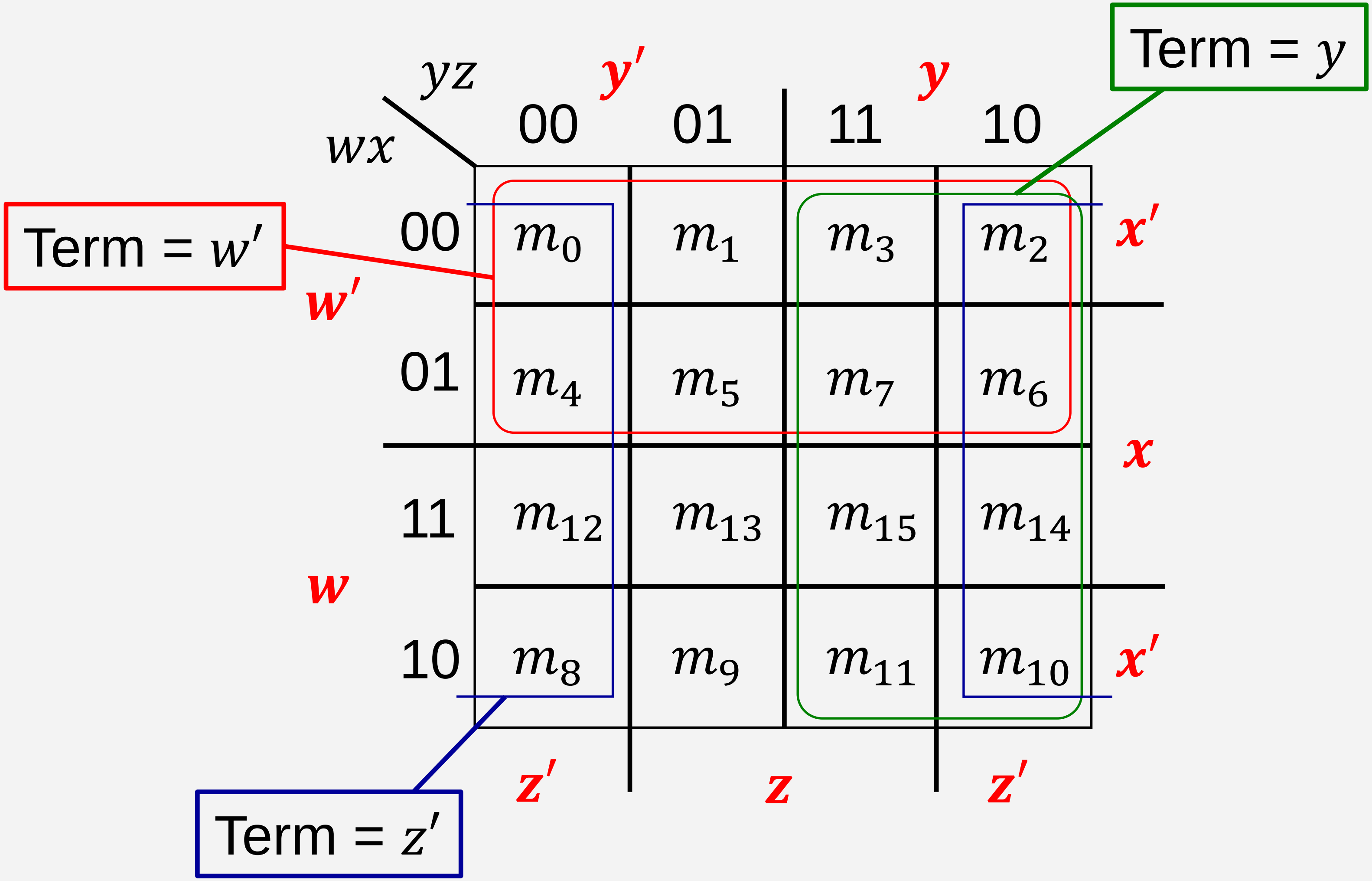


wx \ yz	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

Combining Squares on a 4-Variable K-Map

- On a 4-variable K-Map:
    - One square represents a minterm with 4 variables
    - Two adjacent squares represent a term with 3 variables
    - Four adjacent squares represent a term with 2 variables
    - Eight adjacent squares represent a term with 1 variable
    - Combining all 16 squares is the constant **'1'** (no variables)

Combining Eight Squares



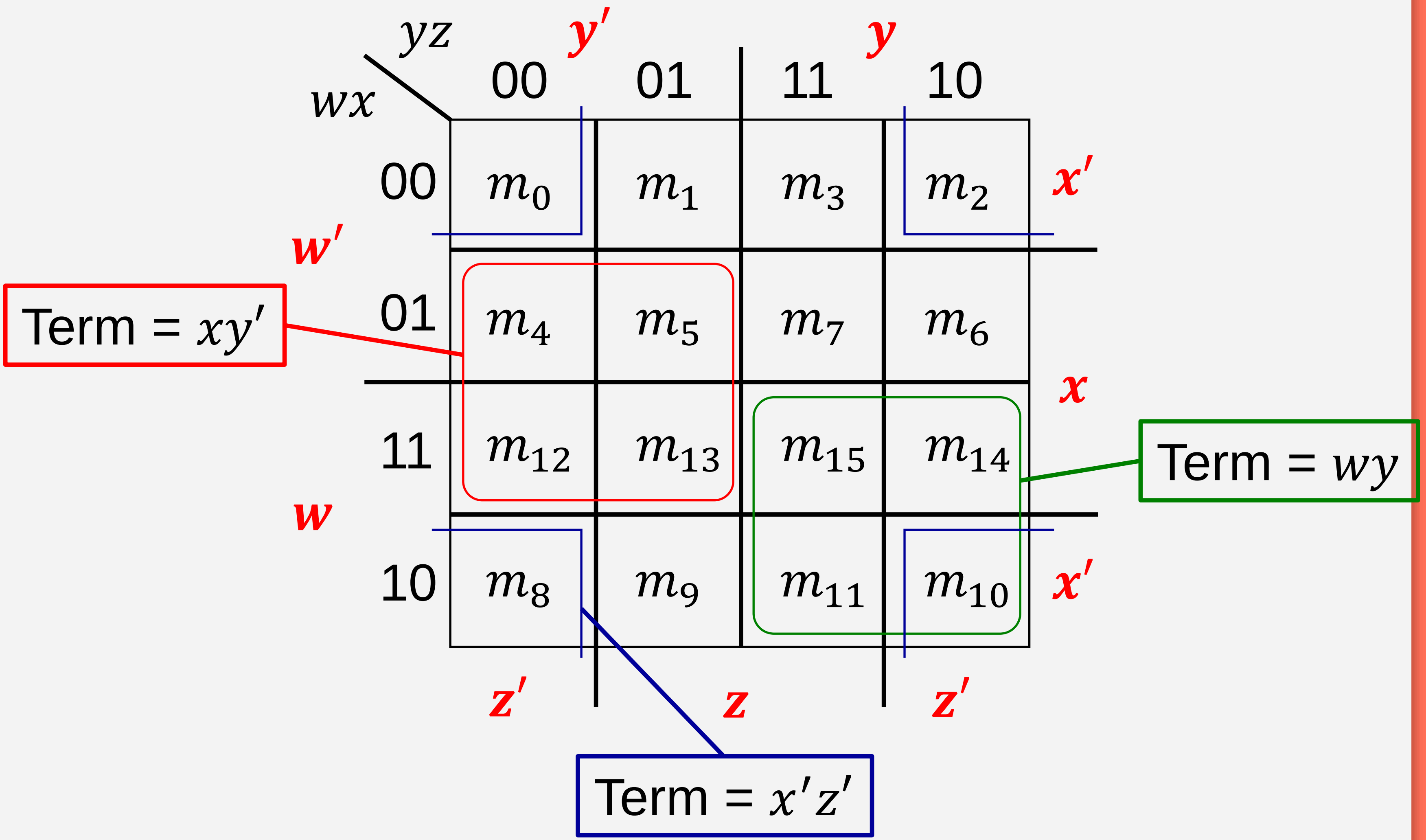
$wx \backslash yz$	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

Term = y

Term = w'

Term = z'

Combining Four Squares



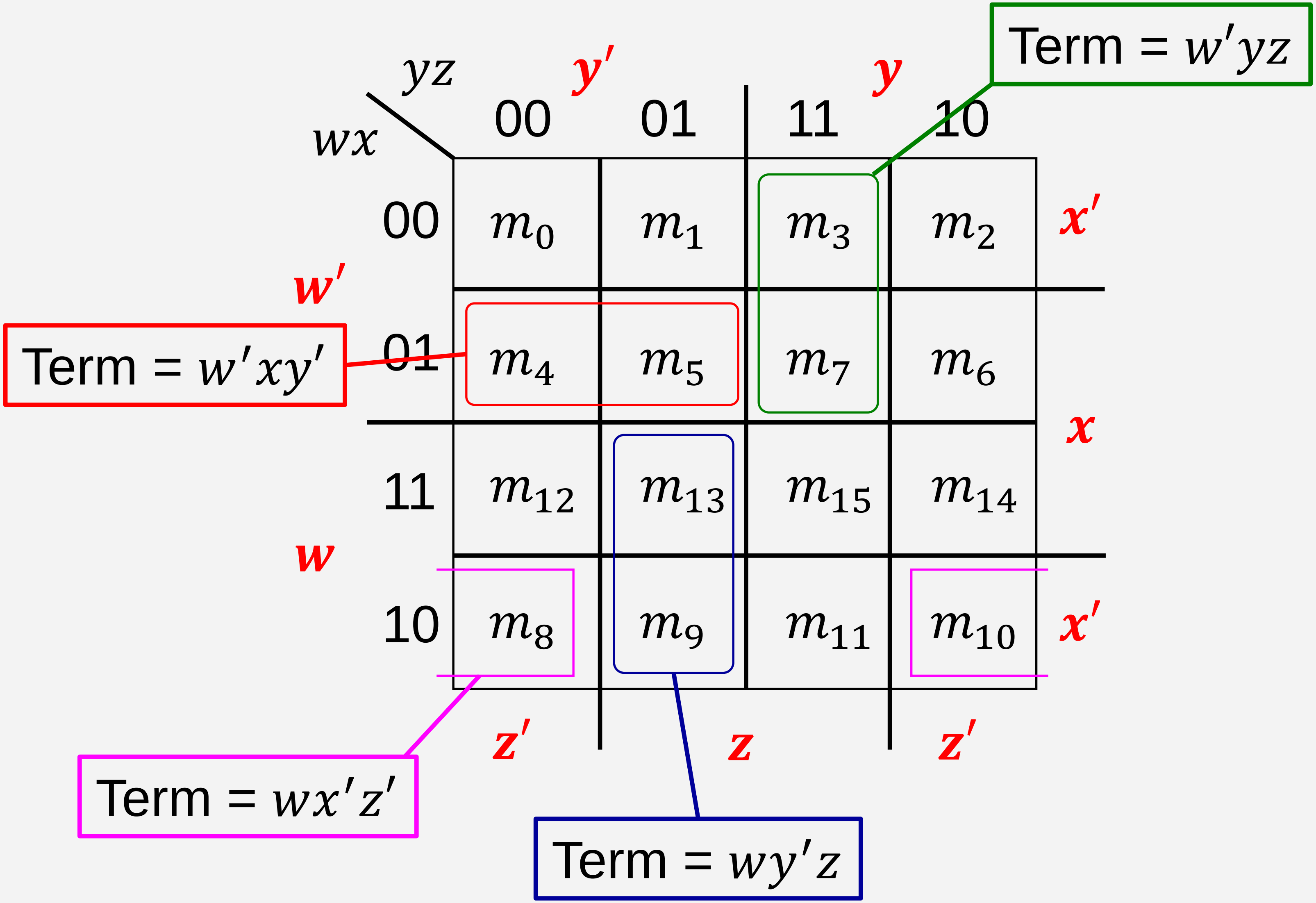
$wx \backslash yz$	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

Term = xy'

Term = wy

Term = $x'z'$

Combining Four Squares



$wx \backslash yz$	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

Term = $w'yz$

Term = $w'xy'$

Term = $wx'z'$

Term = $wy'z$

Simplifying a 4-Variable Function

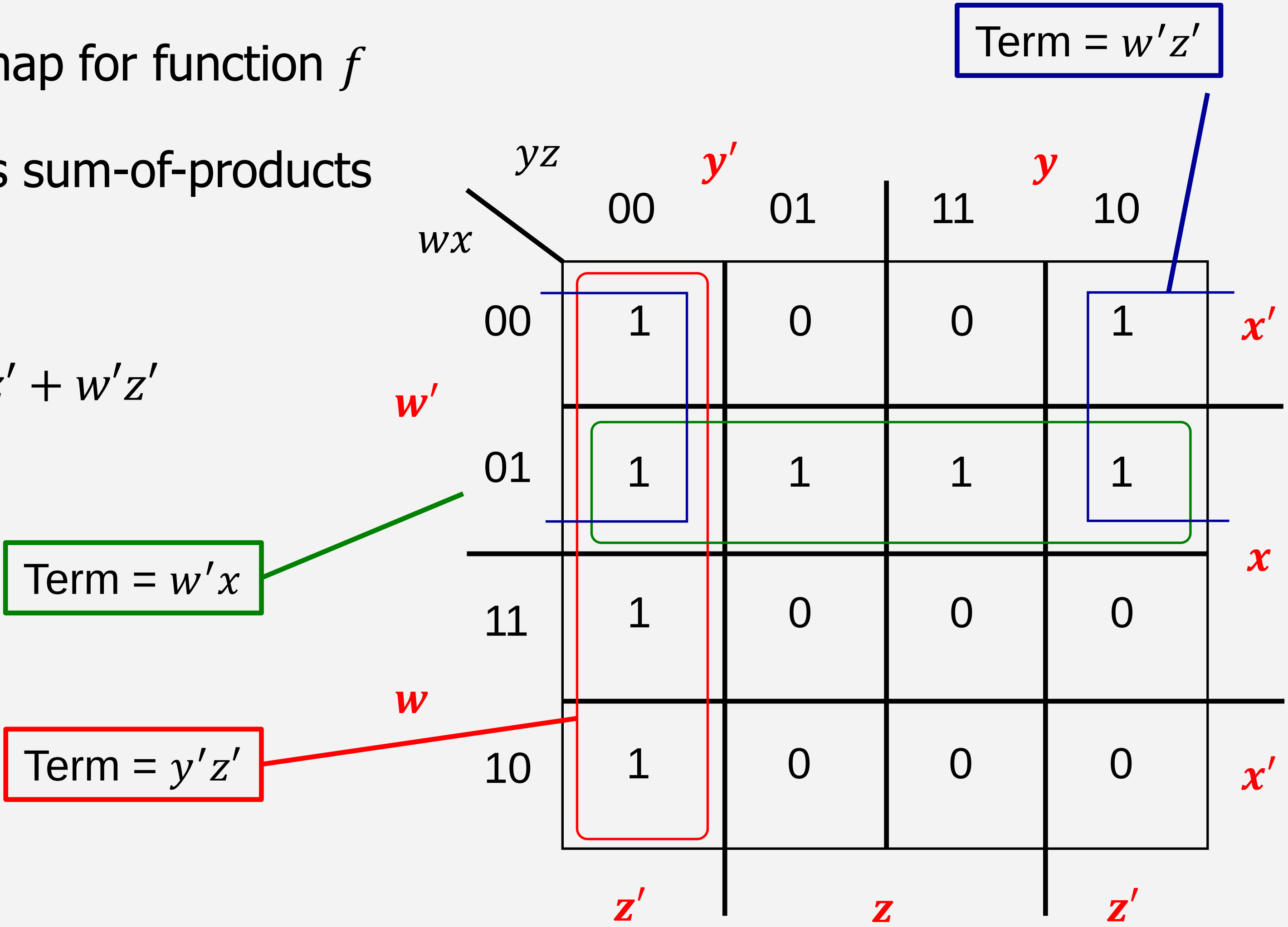
Given $f(w, x, y, z) = \sum(0, 2, 4, 5, 6, 7, 8, 12)$

Draw the K-map for function f

Minimize f as sum-of-products

Solution:

$f = w'x + y'z' + w'z'$



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

Term = $w'z'$

Term = $w'x$

Term = $y'z'$

Prime Implicants

- **Prime Implicant:** a product term obtained by combining the **maximum number of adjacent squares** in the K-map

- The number of combined squares must be a **power of 2**

- **Essential Prime Implicant:** is a prime implicant that covers at least one minterm not covered by the other prime implicants

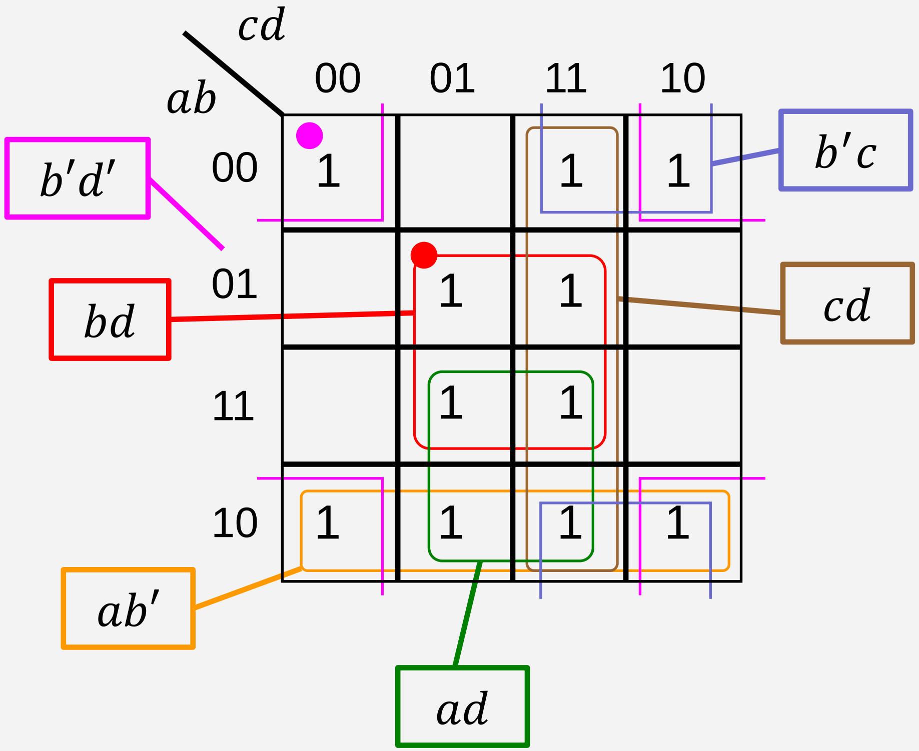
- The prime implicants and essential prime implicants can be determined by inspecting the K-map

Example of Prime Implicants

Find all the prime implicants and essential prime implicants for:

$$f(a,b,c,d) = \sum(0,2,3,5,7,8,9,10,11,13,15)$$

K-Map



ab \ cd	00	01	11	10
00	1		1	1
01		1	1	
11		1	1	
10	1	1	1	1

Six Prime Implicants

bd, $b'd'$, ab', ad, cd, $b'c$

Only Two Prime Implicants are essential

bd and $b'd'$

Simplification Procedure Using the K-Map

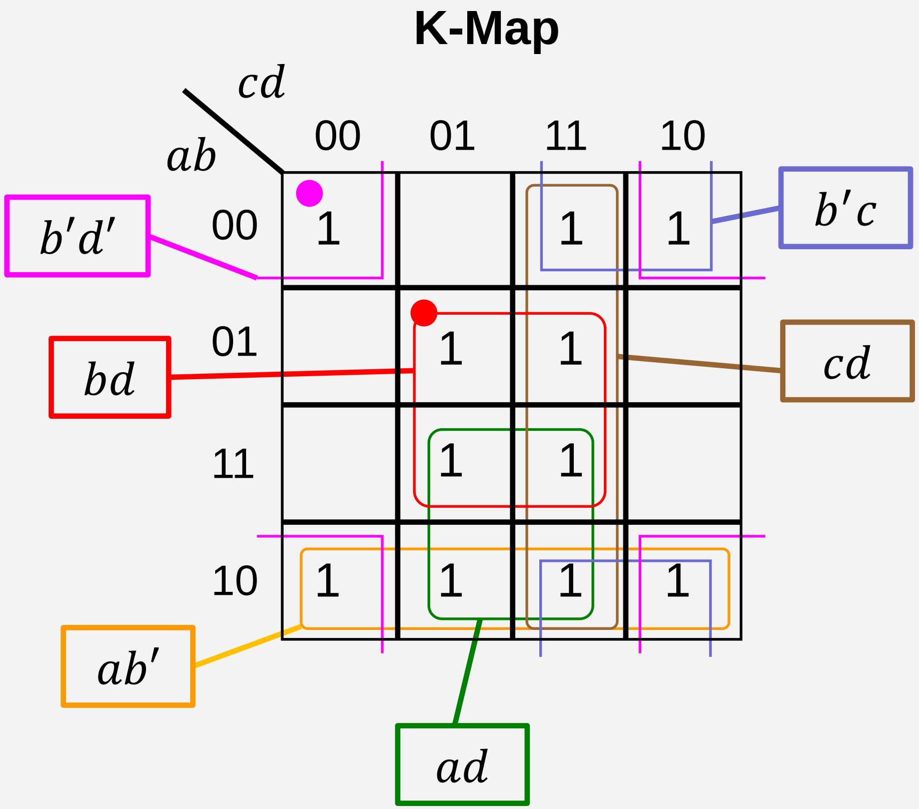
1. Find all the essential prime implicants
   - Covering maximum number (power of 2) of 1's in the K-map
   - Mark the minterm(s) that make the prime implicants essential
2. Add prime implicants to cover the function
   - Choose a minimal subset of prime implicants that cover all remaining 1's
   - Make sure to cover all 1's not covered by the essential prime implicants
   - Minimize the overlap among the additional prime implicants

- Sometimes, a function has multiple simplified expressions
  - You may be asked to list all the simplified sum-of-product expressions

Obtaining All Minimal SOP Expressions

Consider again: $f(a,b,c,d) = \sum(0,2,3,5,7,8,9,10,11,13,15)$

Obtain all minimal sum-of-products (SOP) expressions



Two essential Prime Implicants: bd and $b'd'$

Four possible solutions:

$f = bd + b'd' + cd + ad$

$f = bd + b'd' + cd + ab'$

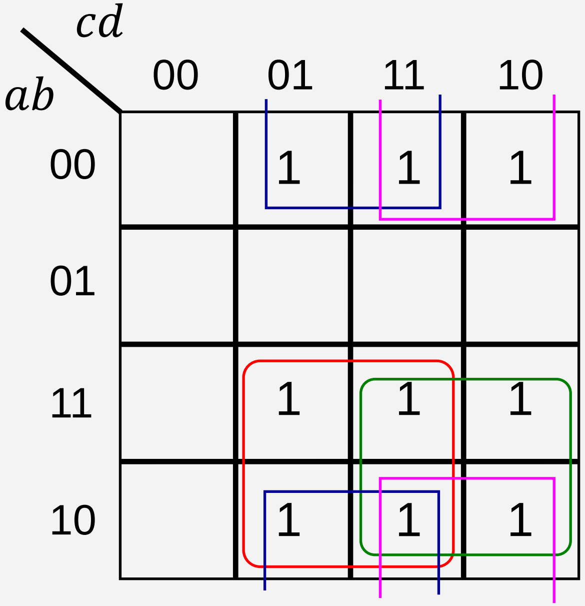
$f = bd + b'd' + b'c + ab'$

$f = bd + b'd' + b'c + ad$

Product-of-Sums (POS) Simplification

- All previous examples were expressed in Sum-of-Products form
- With a minor modification, the Product-of-Sums can be obtained
- Example: $f(a,b,c,d) = \sum(1,2,3,9,10,11,13,14,15)$

K-Map of f



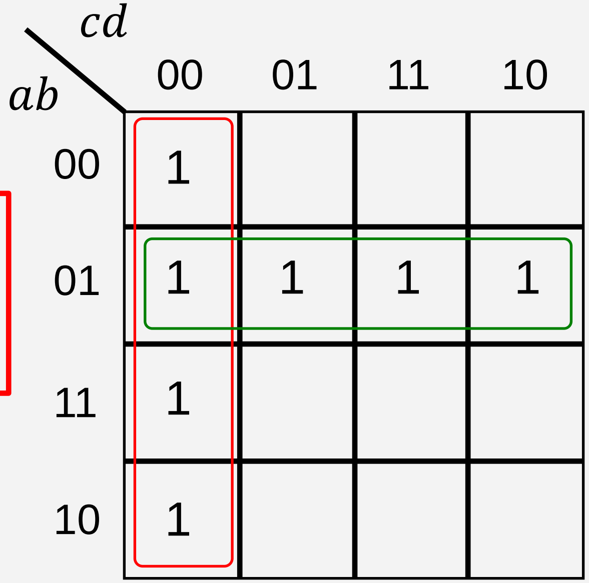
ab \ cd	00	01	11	10
00		1	1	1
01				
11		1	1	1
10		1	1	1

$$f = ad + ac + b'd + b'c$$

Minimal Sum-of-Products = 8 literals

All prime implicants are essential

K-Map of f'



ab \ cd	00	01	11	10
00	1			
01	1	1	1	1
11	1			
10	1			

$$f' = c'd' + a'b$$

$$f = (c+d)(a+b') \rightarrow$$

Minimal Product-of-Sums = 4 literals

Product-of-Sums Simplification Procedure

1. Draw the K-map for the function f
   - Obtain a minimal Sum-of-Products (SOP) expression for f
2. Draw the K-map for f', replacing the 0's of f with 1's in f'
3. Obtain a minimal Sum-of-Products (SOP) expression for f'
4. Use DeMorgan's theorem to obtain $f = (f')'$
   - The result is a minimal Product-of-Sums (POS) expression for f
5. Compare the cost of the minimal SOP and POS expressions
   - Count the number of literals to find which expression is minimal

Don't Cares

- Sometimes, a function table may contain entries for which:
  - The input values of the variables will never occur, or
  - The output value of the function is never used
- In this case, the output value of the function is not defined
- The output value of the function is called a **don't care**
- A don't care is an **X** value that appears in the function table
- The **X** value can be later chosen to be **0 or 1**
  - To minimize the function implementation

Example of a Function with Don't Cares

- Consider a function f defined over BCD inputs
- The function input is a BCD digit from 0 to 9
- The function output is 0 if the BCD input is 0 to 4
- The function output is 1 if the BCD input is 5 to 9
- The function output is X (don't care) if the input is 10 to 15 (not BCD)
- $f = \underbrace{\sum_m (5,6,7,8,9)}_{\text{Minterms}} + \underbrace{\sum_d (10,11,12,13,14,15)}_{\text{Don't Cares}}$

Truth Table

a	b	c	d	f
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

Minimizing Functions with Don't Cares

Consider: $f = \sum_m(5,6,7,8,9) + \sum_d(10,11,12,13,14,15)$

If the don't cares were treated as 0's we get:

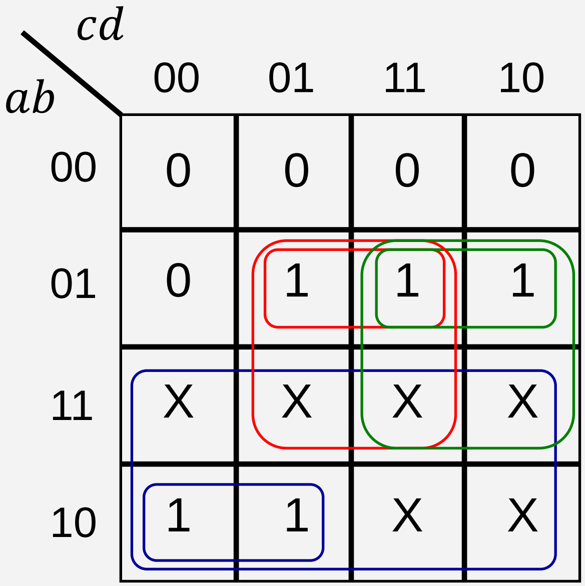
$f = a'bd + a'bc + ab'c'$ (9 literals)

If the don't cares were treated as 1's we get:

$f = a + bd + bc$ (5 literals)

The don't care values can be selected to be either 0 or 1, to produce a minimal expression

K-Map of f



$ab \backslash cd$	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

Simplification Procedure with Don't Cares

1. Find all the essential prime implicants
   - Covering maximum number (power of 2) of 1's and X's (don't cares)
   - Mark the 1's that make the prime implicants essential
2. Add prime implicants to cover the function
   - Choose a minimal subset of prime implicants that cover all remaining 1's
   - Make sure to cover all 1's not covered by the essential prime implicants
   - Minimize the overlap among the additional prime implicants
   - You need not cover all the don't cares (some can remain uncovered)

- Sometimes, a function has multiple simplified expressions

Minimizing Functions with Don't Cares (2)

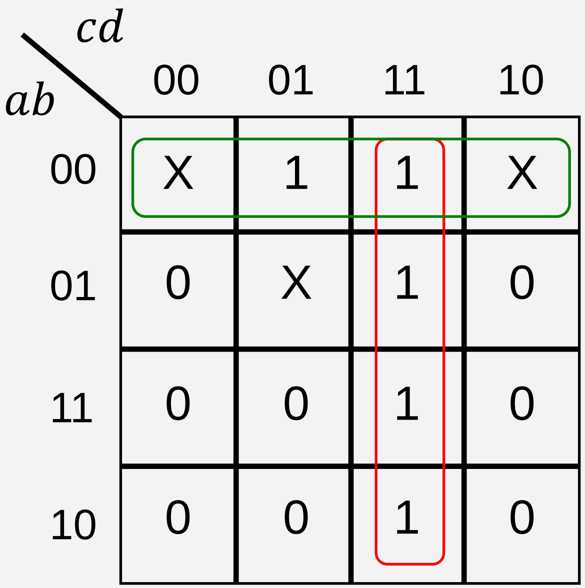
Simplify: $g = \sum_m (1,3,7,11,15) + \sum_d (0,2,5)$

Solution 1: $g = cd + a'b'$ (4 literals)

Solution 2: $g = cd + a'd$ (4 literals)

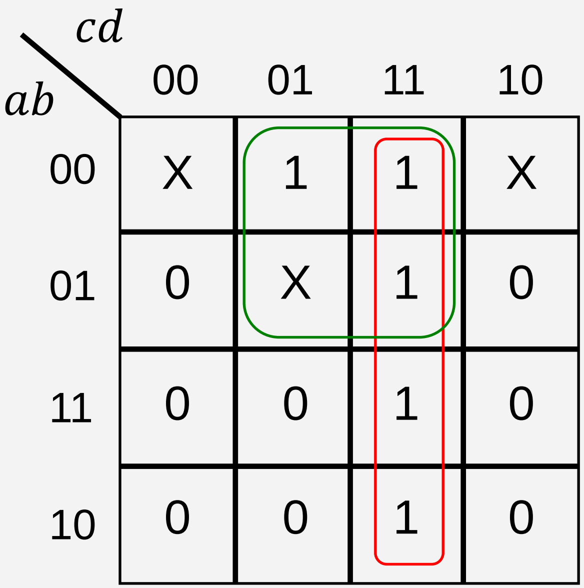
Prime Implicant cd is essential

K-Map of g



$ab \backslash cd$	00	01	11	10
00	X	1	1	X
01	0	X	1	0
11	0	0	1	0
10	0	0	1	0

K-Map of g



$ab \backslash cd$	00	01	11	10
00	X	1	1	X
01	0	X	1	0
11	0	0	1	0
10	0	0	1	0

Not all don't cares need be covered

Minimal Product-of-Sums with Don't Cares

Simplify: $g = \sum_m(1,3,7,11,15) + \sum_d(0,2,5)$

Obtain a product-of-sums minimal expression

Solution: $g' = \sum_m(4,6,8,9,10,12,13,14) + \sum_d(0,2,5)$

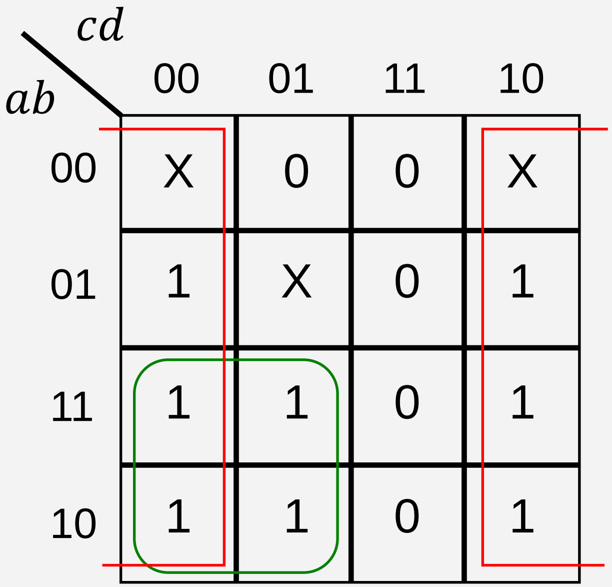
Minimal $g' = d' + ac'$ (3 literals)

Minimal product-of-sums:

$g = d(a' + c)$ (3 literals)

The minimal sum-of-products expression for g had 4 literals

K-Map of g'



$ab \backslash cd$	00	01	11	10
00	X	0	0	X
01	1	X	0	1
11	1	1	0	1
10	1	1	0	1

Five-Variable Karnaugh Map

- Consists of $2^5 = 32$ squares, numbered 0 to 31
  - Remember the numbering of squares in the K-map
- Can be visualized as two layers of 16 squares each
- Top layer contains the squares of the first 16 minterms $(a = 0)$
- Bottom layer contains the squares of the last 16 minterms $(a = 1)$

$a = 0$

bc \ de	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

$a = 1$

bc \ de	00	01	11	10
00	m_{16}	m_{17}	m_{19}	m_{18}
01	m_{20}	m_{21}	m_{23}	m_{22}
11	m_{28}	m_{29}	m_{31}	m_{30}
10	m_{24}	m_{25}	m_{27}	m_{26}

Each square is adjacent to
5 other squares:
4 in the same layer and
1 in the other layer:
m_0 is adjacent to m_{16}
m_1 is adjacent to m_{17}
m_4 is adjacent to m_{20} …

Example of a Five-Variable K-Map

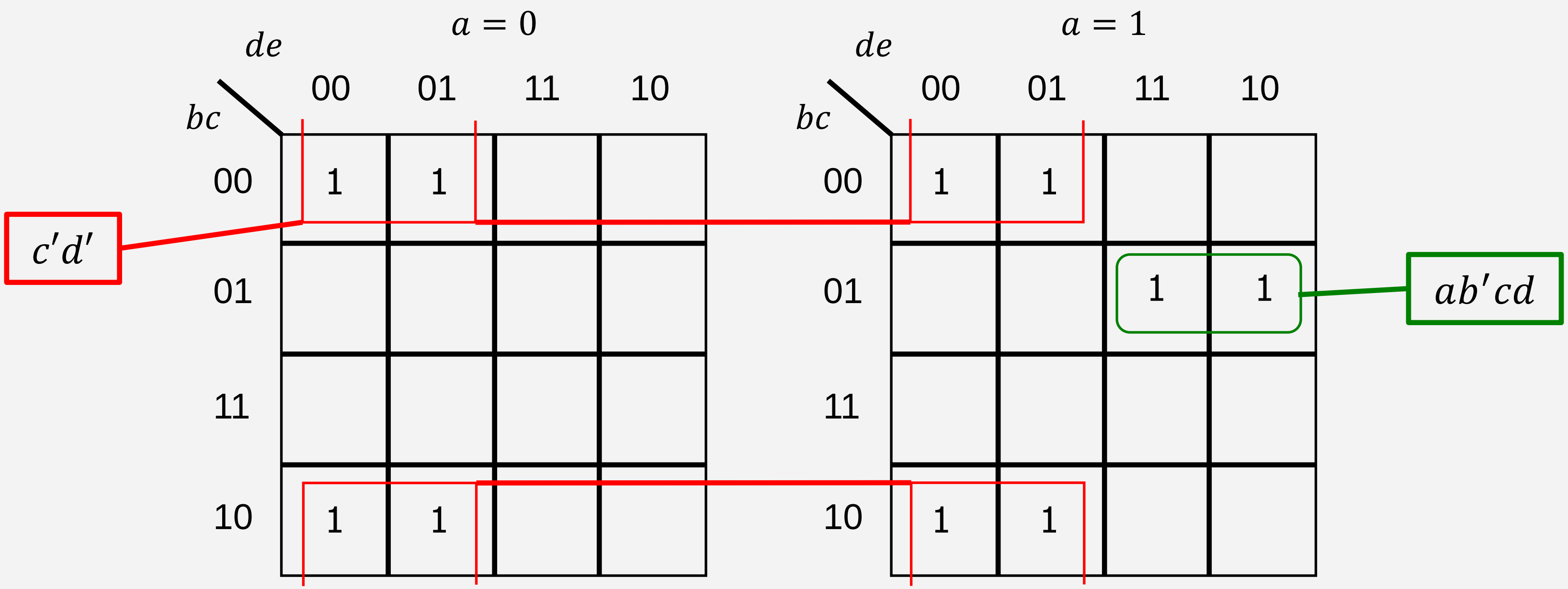
Given: $f(a,b,c,d,e) = \sum(0,1,8,9,16,17,22,23,24,25)$

Draw the 5-Variable K-Map

Obtain a minimal Sum-of-Products expression for f

Solution: $f \ = \ c'd' + ab'cd$ (6 literals)

5-Variable K-Map



$a = 0$

$bc \backslash de$	00	01	11	10
00	1	1		
01				
11				
10	1	1		

$a = 1$

$bc \backslash de$	00	01	11	10
00	1	1		
01			1	1
11				
10	1	1		

Five-Variable K-Map with Don't Cares

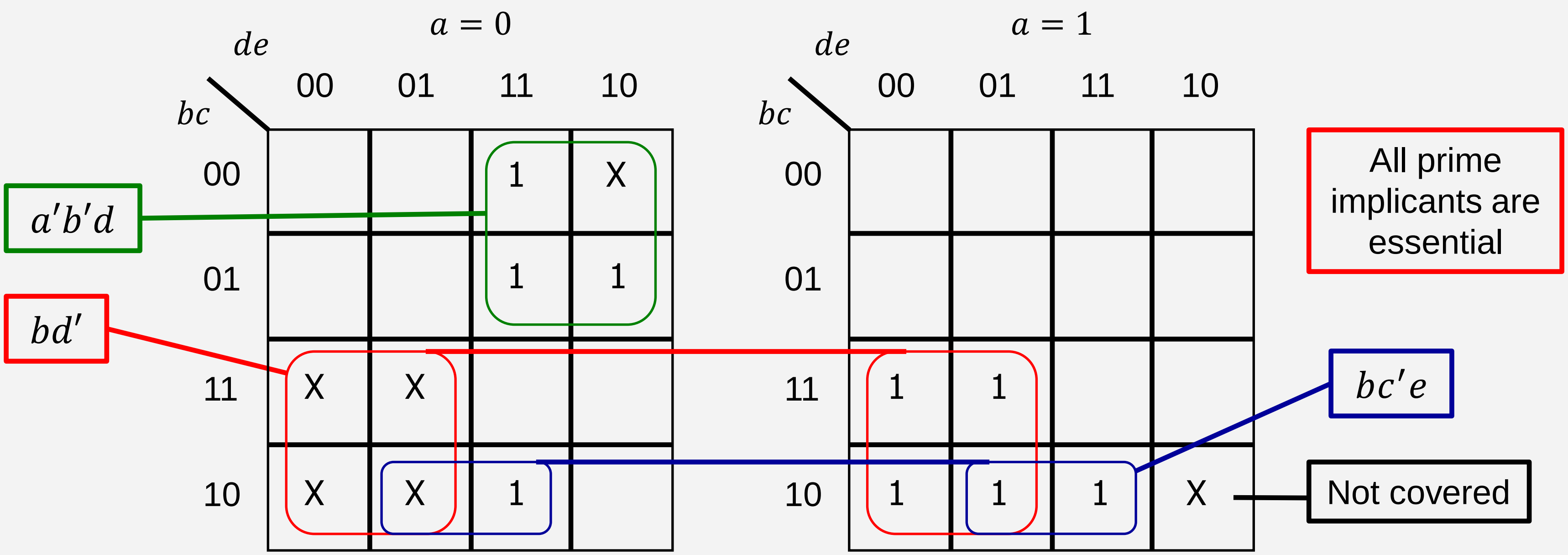
$g(a,b,c,d,e) = \sum_m(3,6,7,11,24,25,27,28,29) + \sum_d(2,8,9,12,13,26)$

Draw the 5-Variable K-Map

Obtain a minimal Sum-of-Products expression for g

Solution: $g = bd' + a'b'd + bc'e$ (8 literals)

5-Variable K-Map



$a = 0$

$bc \backslash de$	00	01	11	10
00			1	X
01			1	1
11	X	X		
10	X	X	1	

$a = 1$

$bc \backslash de$	00	01	11	10
00				
01				
11	1	1		
10	1	1	1	X

Six-Variable Karnaugh Map

- Consists of $2^6 = 64$ squares, numbered 0 to 63
- Can be visualized as four layers of 16 squares each
  - Four layers: $ab = 00, 01, 11, 10$ (Notice that layer 11 comes before 10)
- Each square is adjacent to 6 other squares:
  - 4 squares in the same layer and 2 squares in the above and below layers

$ab = 00$

cd \ ef	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

$ab = 01$

	00	01	11	10
00	m_{16}	m_{17}	m_{19}	m_{18}
01	m_{20}	m_{21}	m_{23}	m_{22}
11	m_{28}	m_{29}	m_{31}	m_{30}
10	m_{24}	m_{25}	m_{27}	m_{26}

$ab = 11$

	00	01	11	10
00	m_{48}	m_{49}	m_{51}	m_{50}
01	m_{52}	m_{53}	m_{55}	m_{54}
11	m_{60}	m_{61}	m_{63}	m_{62}
10	m_{56}	m_{57}	m_{59}	m_{58}

$ab = 10$

	00	01	11	10
00	m_{32}	m_{33}	m_{35}	m_{34}
01	m_{36}	m_{37}	m_{39}	m_{38}
11	m_{44}	m_{45}	m_{47}	m_{46}
10	m_{40}	m_{41}	m_{43}	m_{42}

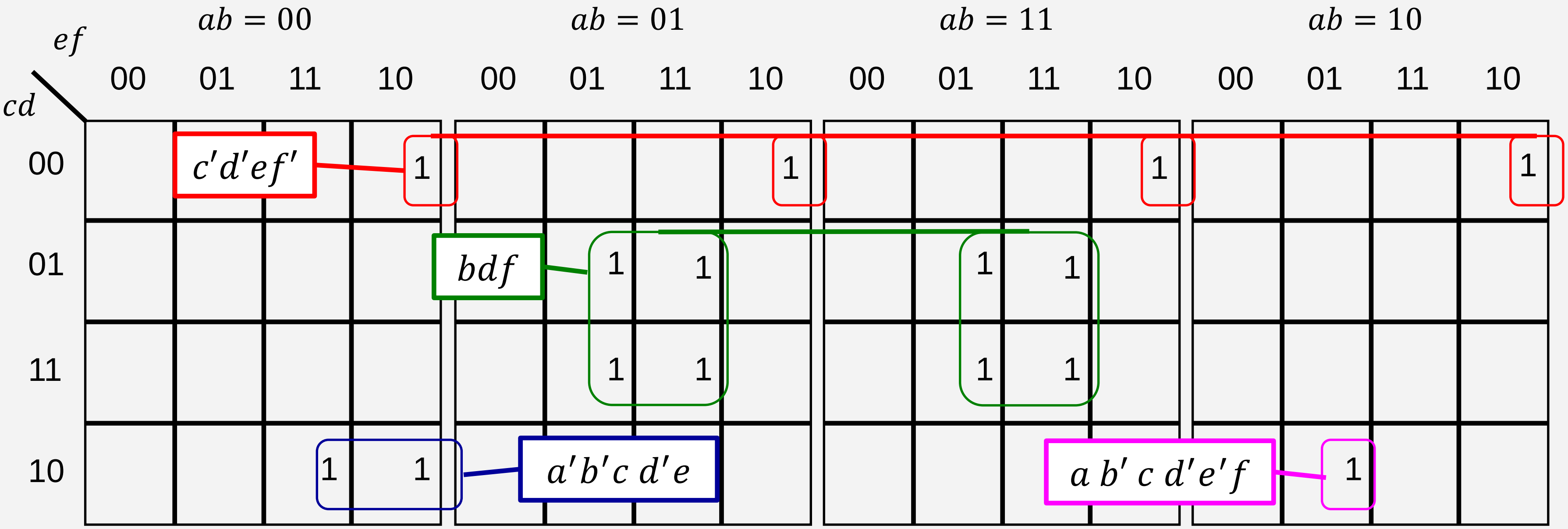
Example of a Six-Variable K-Map

$h(a,b,c,d,e,f) = \sum(2,10,11,18,21,23,29,31,34,41,50,53,55,61,63)$

Draw the 6-Variable K-Map

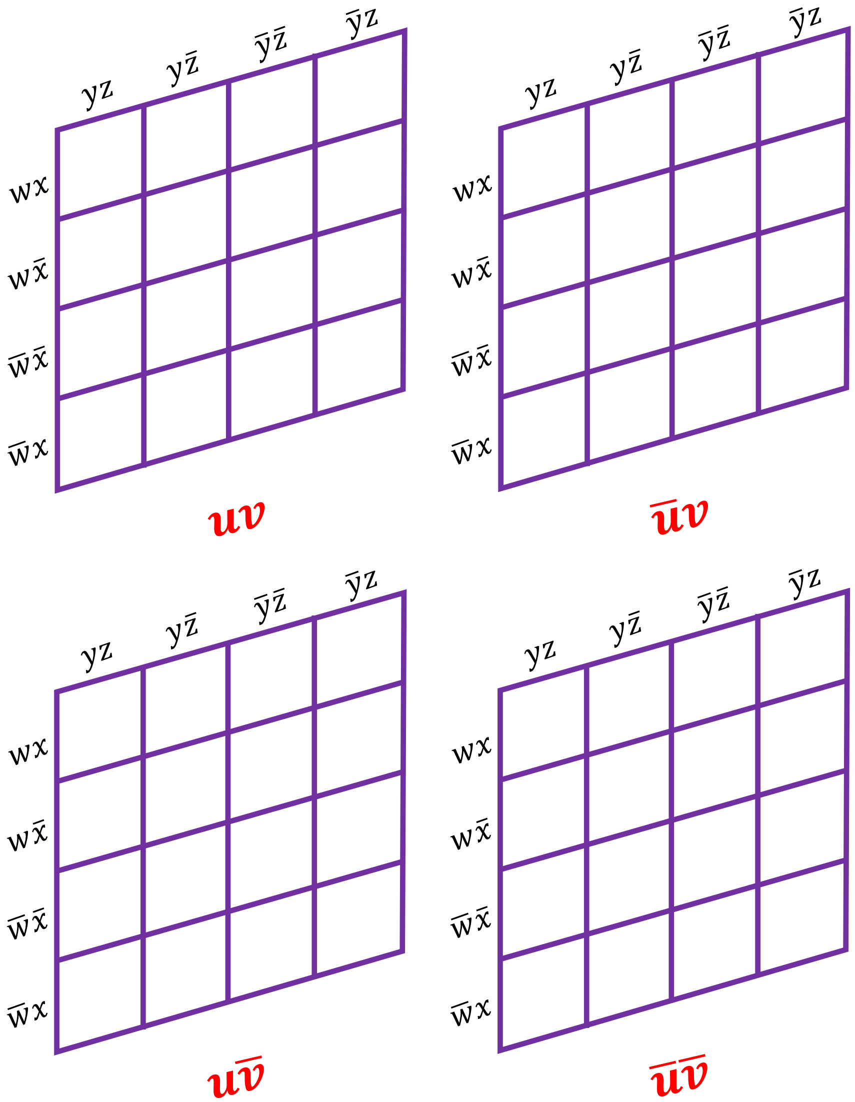
Obtain a minimal Sum-of-Products expression for h

Solution: $h = c'd'ef' + b\,d\,f + a'b'c\,d'e + a\,b'\,c\,d'e'f$ (18 literals)



	ab = 00				ab = 01				ab = 11				ab = 10			
cd \ ef	00	01	11	10	00	01	11	10	00	01	11	10	00	01	11	10
00				1				1				1				1
01						1	1			1	1					
11						1	1			1	1					
10			1	1										1		

Six-Variable Karnaugh Map



	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx				
$w\bar{x}$				
$\bar{w}\bar{x}$				
$\bar{w}x$				

uv

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx				
$w\bar{x}$				
$\bar{w}\bar{x}$				
$\bar{w}x$				

$\bar{u}v$

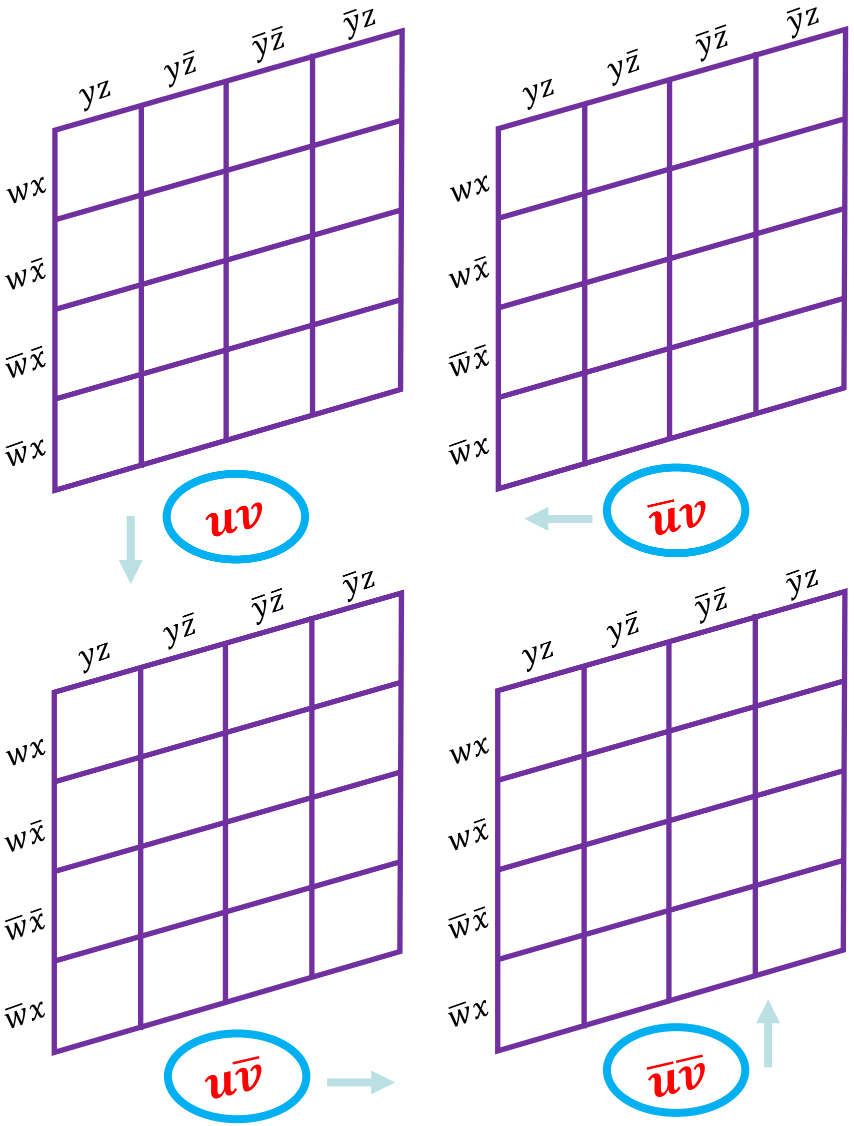
	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx				
$w\bar{x}$				
$\bar{w}\bar{x}$				
$\bar{w}x$				

$u\bar{v}$

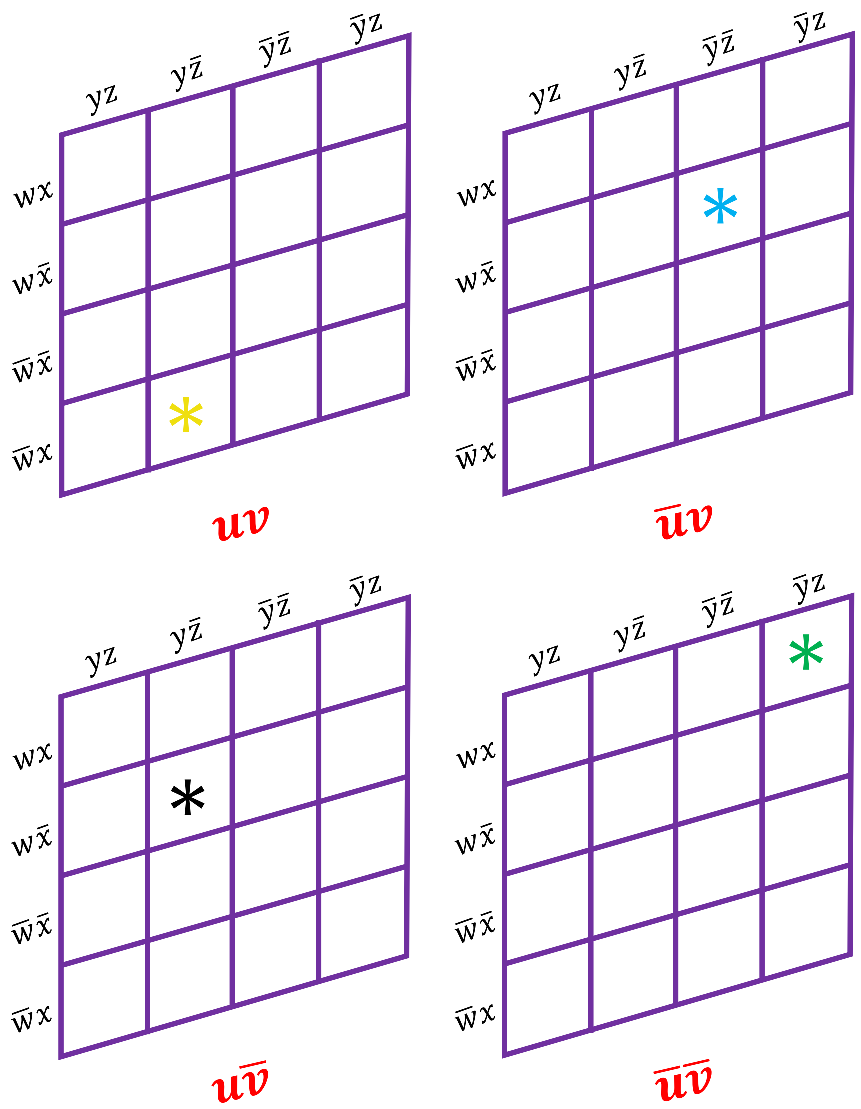
	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx				
$w\bar{x}$				
$\bar{w}\bar{x}$				
$\bar{w}x$				

$\bar{u}\bar{v}$

Six-Variable Karnaugh Map



Six-Variable Karnaugh Map

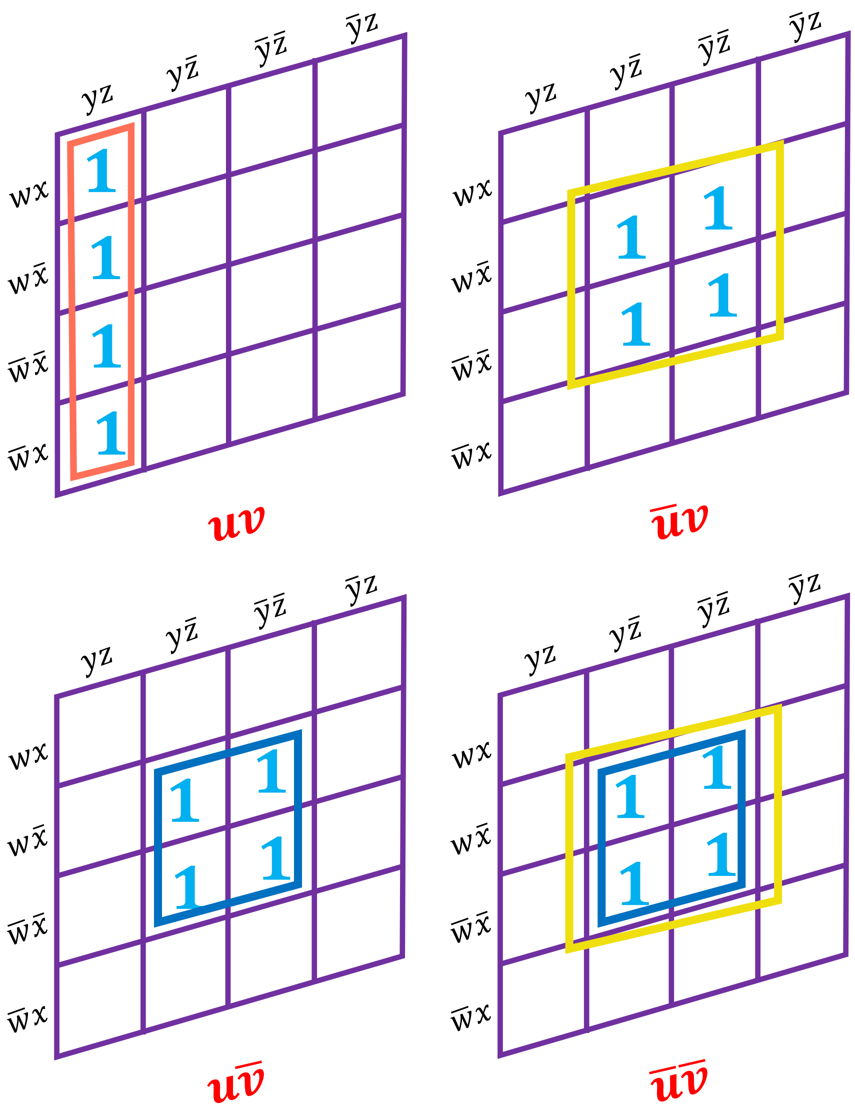


Identify the Stars?

Solution:

*	$\bar{u}v\,\bar{w}\,\bar{x}\,y\,\bar{z}$
*	$uv\,w\,\bar{x}\,y\,\bar{z}$
*	$u\bar{v}\,\bar{w}\,\bar{x}\,\bar{y}\,\bar{z}$
*	$\bar{u}\,\bar{v}\,w\,x\,\bar{y}\,z$

Six-Variable Karnaugh Map



	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx	1			
$w\bar{x}$	1			
$\bar{w}\bar{x}$	1			
$\bar{w}x$	1			

uv

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx				
$w\bar{x}$		1	1	
$\bar{w}\bar{x}$		1	1	
$\bar{w}x$				

$\bar{u}v$

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx				
$w\bar{x}$		1	1	
$\bar{w}\bar{x}$		1	1	
$\bar{w}x$				

$u\bar{v}$

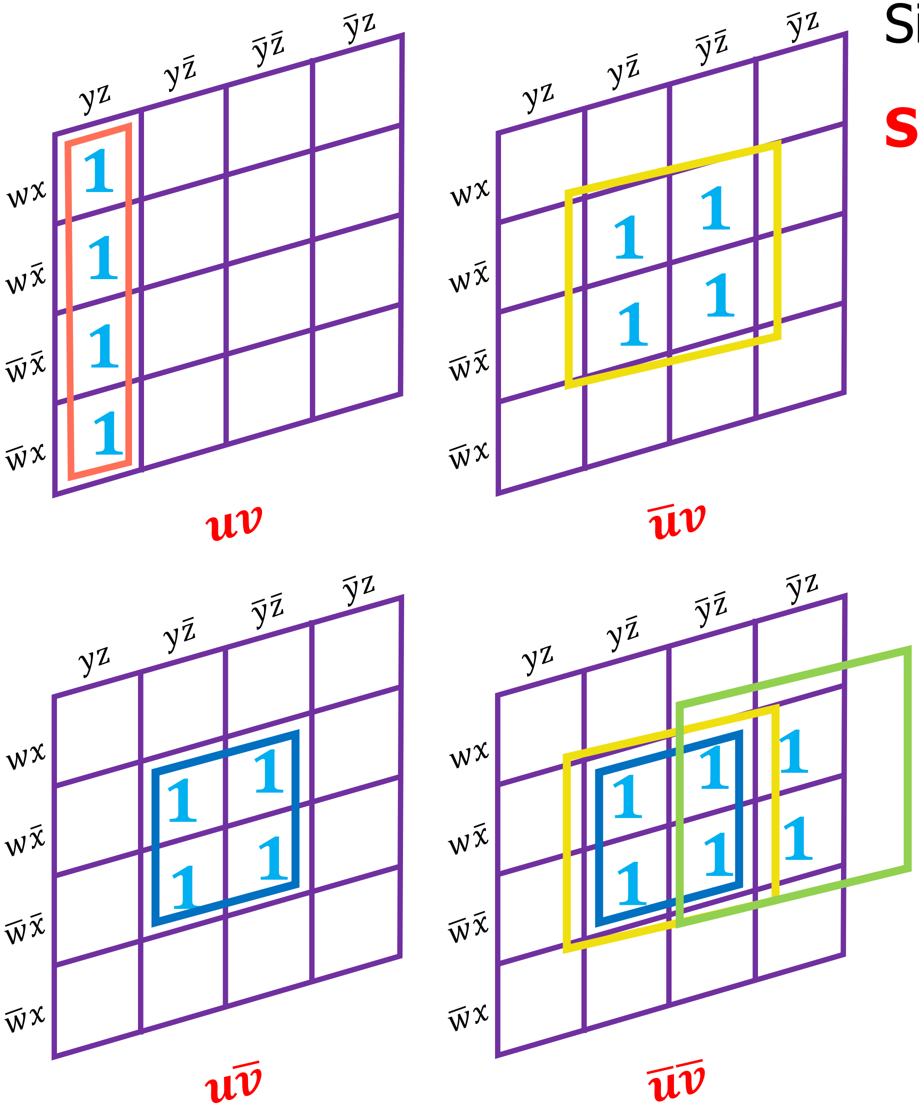
	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx				
$w\bar{x}$		1	1	
$\bar{w}\bar{x}$		1	1	
$\bar{w}x$				

$\bar{u}\bar{v}$

Simplify the mapped expression?

Solution:

Six-Variable Karnaugh Map



uv

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx	1			
$w\bar{x}$	1			
$\bar{w}\bar{x}$	1			
$\bar{w}x$	1			

$\bar{u}v$

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx				
$w\bar{x}$		1	1	
$\bar{w}\bar{x}$		1	1	
$\bar{w}x$				

$u\bar{v}$

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx				
$w\bar{x}$		1	1	
$\bar{w}\bar{x}$		1	1	
$\bar{w}x$				

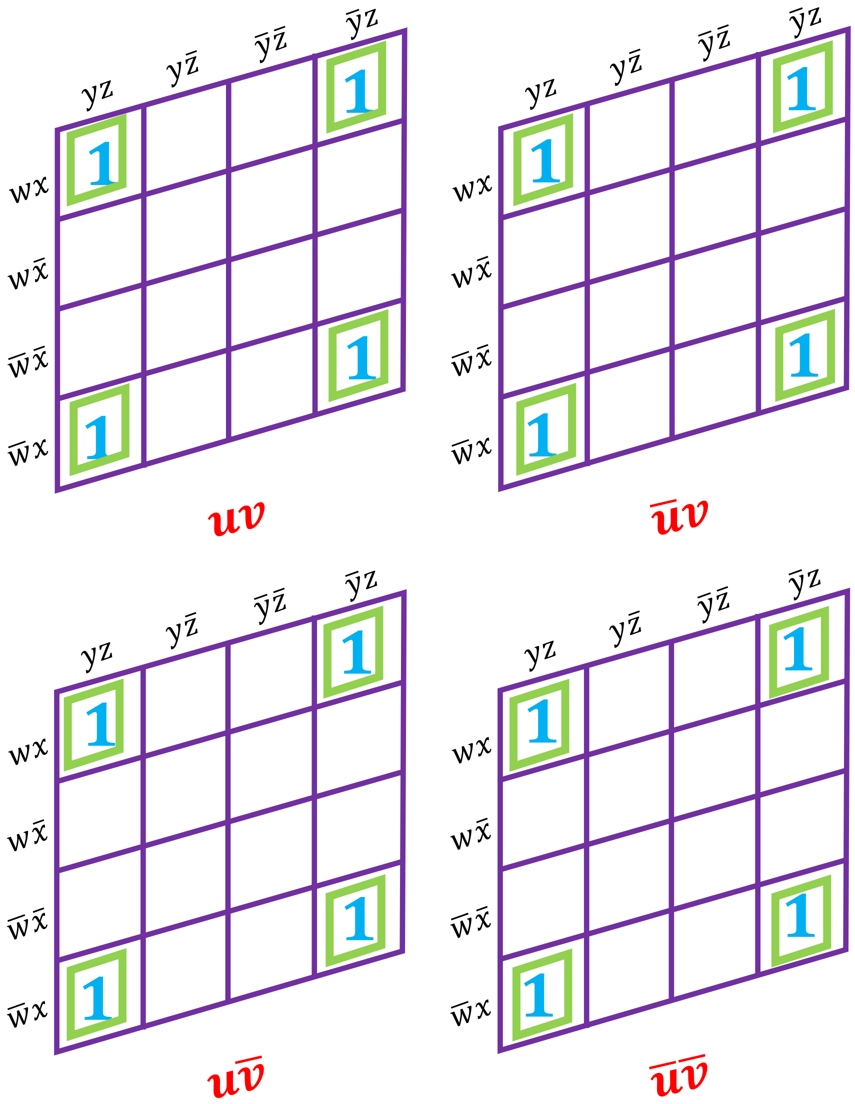
$\bar{u}\bar{v}$

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx				
$w\bar{x}$		1	1	1
$\bar{w}\bar{x}$		1	1	1
$\bar{w}x$				

Simplify the mapped expression?

Solution:

Six-Variable Karnaugh Map



uv

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx	1			1
$w\bar{x}$				
$\bar{w}\bar{x}$				1
$\bar{w}x$	1			

$\bar{u}v$

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx	1			1
$w\bar{x}$				
$\bar{w}\bar{x}$				1
$\bar{w}x$	1			

$u\bar{v}$

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx	1			1
$w\bar{x}$				
$\bar{w}\bar{x}$				1
$\bar{w}x$	1			

$\bar{u}\bar{v}$

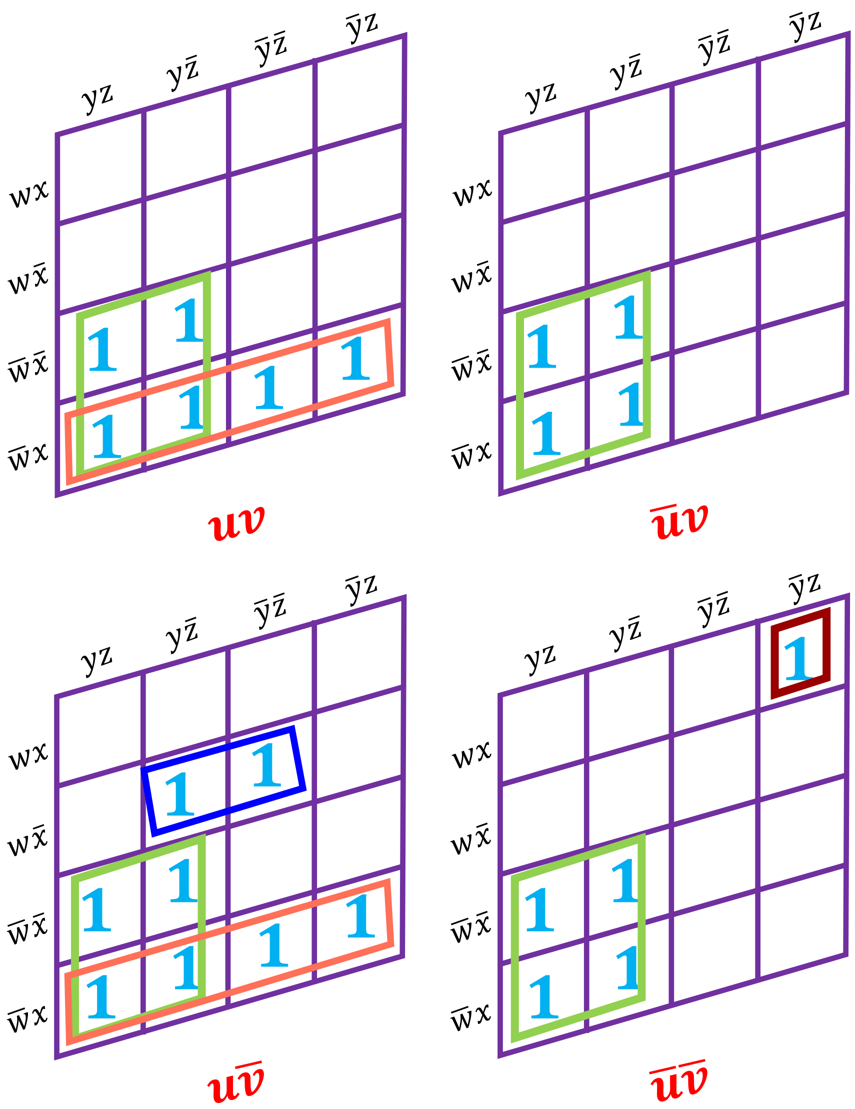
	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx	1			1
$w\bar{x}$				
$\bar{w}\bar{x}$				1
$\bar{w}x$	1			

Simplify the mapped expression?

Solution:

xz

Six-Variable Karnaugh Map



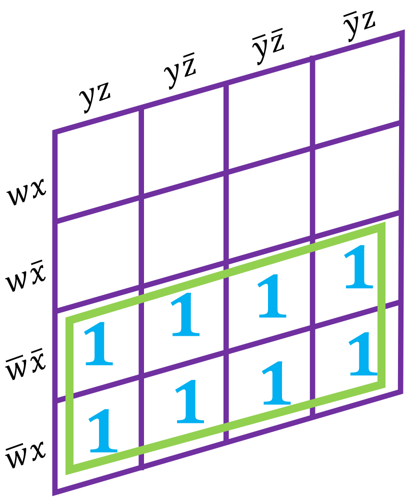
Simplify the mapped expression?

Solution:

$$\overline{w}y + u\overline{w}x + u\overline{v}w\,\overline{x}\,\overline{z}$$

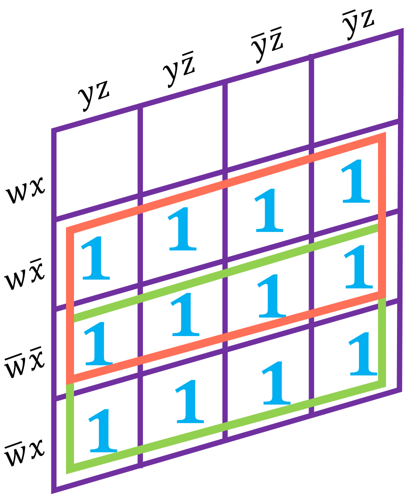
$$+\,\overline{u}\,\overline{v}\,wx\overline{y}\,z$$

Six-Variable Karnaugh Map



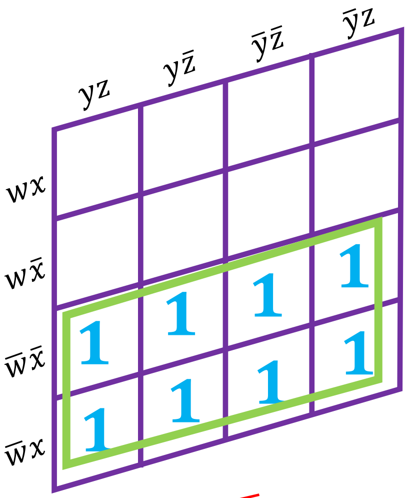
	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx				
$w\bar{x}$				
$\bar{w}\bar{x}$	1	1	1	1
$\bar{w}x$	1	1	1	1

uv



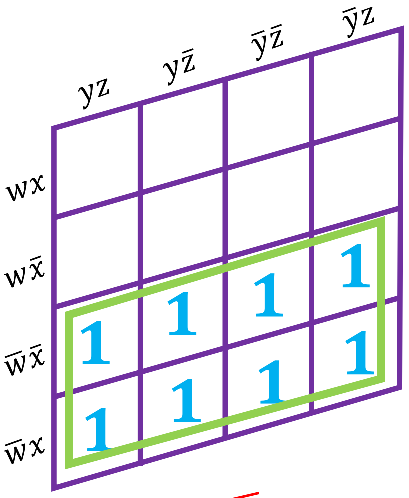
	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx				
$w\bar{x}$	1	1	1	1
$\bar{w}\bar{x}$	1	1	1	1
$\bar{w}x$	1	1	1	1

$\bar{u}v$



	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx				
$w\bar{x}$				
$\bar{w}\bar{x}$	1	1	1	1
$\bar{w}x$	1	1	1	1

$u\bar{v}$



	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx				
$w\bar{x}$				
$\bar{w}\bar{x}$	1	1	1	1
$\bar{w}x$	1	1	1	1

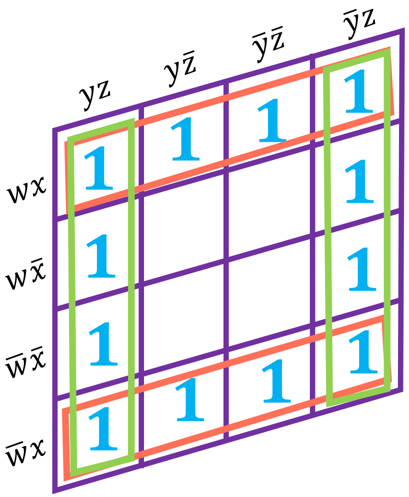
$\bar{u}\bar{v}$

Simplify the mapped expression?

Solution:

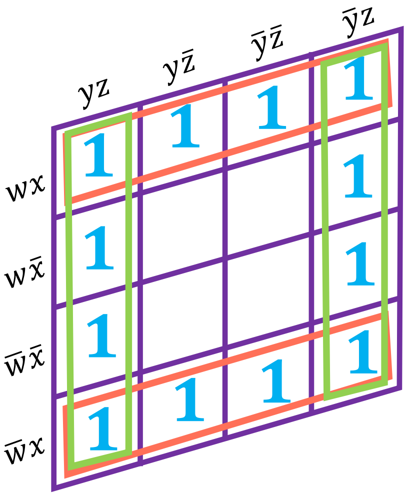
$$\overline{w} + \overline{u}\, v\, \overline{x}$$

Six-Variable Karnaugh Map



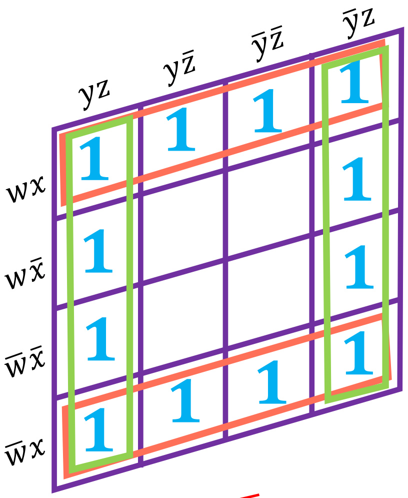
	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx	1	1	1	1
$w\bar{x}$	1			1
$\bar{w}\bar{x}$	1			1
$\bar{w}x$	1	1	1	1

uv



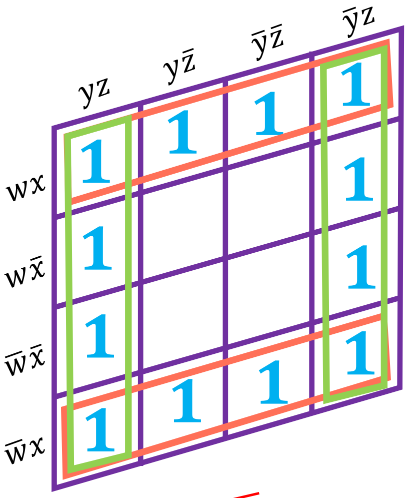
	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx	1	1	1	1
$w\bar{x}$	1			1
$\bar{w}\bar{x}$	1			1
$\bar{w}x$	1	1	1	1

$\bar{u}v$



	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx	1	1	1	1
$w\bar{x}$	1			1
$\bar{w}\bar{x}$	1			1
$\bar{w}x$	1	1	1	1

$u\bar{v}$



	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx	1	1	1	1
$w\bar{x}$	1			1
$\bar{w}\bar{x}$	1			1
$\bar{w}x$	1	1	1	1

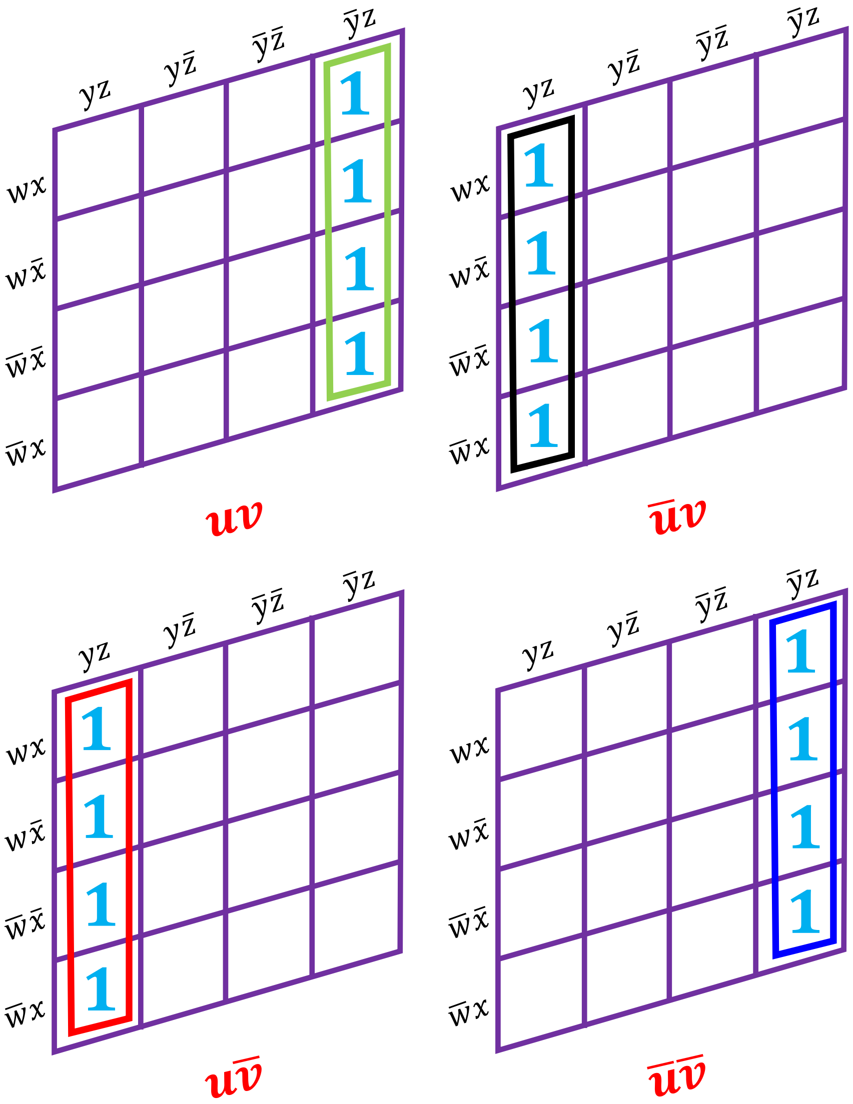
$\bar{u}\bar{v}$

Simplify the mapped expression?

Solution:

$$x + z$$

Six-Variable Karnaugh Map



uv

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx				1
$w\bar{x}$				1
$\bar{w}\bar{x}$				1
$\bar{w}x$				1

$\bar{u}v$

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx	1			
$w\bar{x}$	1			
$\bar{w}\bar{x}$	1			
$\bar{w}x$	1			

$u\bar{v}$

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx	1			
$w\bar{x}$	1			
$\bar{w}\bar{x}$	1			
$\bar{w}x$	1			

$\bar{u}\bar{v}$

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx				1
$w\bar{x}$				1
$\bar{w}\bar{x}$				1
$\bar{w}x$				1

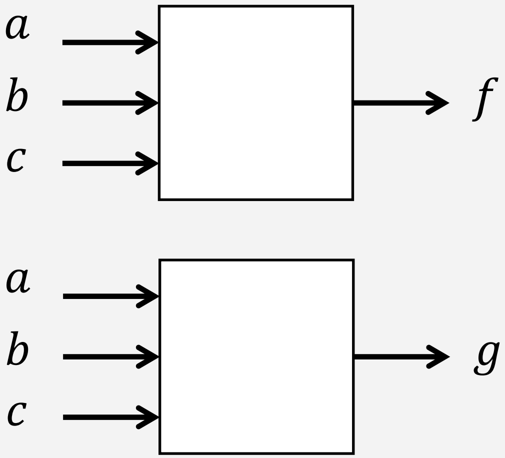
Simplify the mapped expression?

Solution:

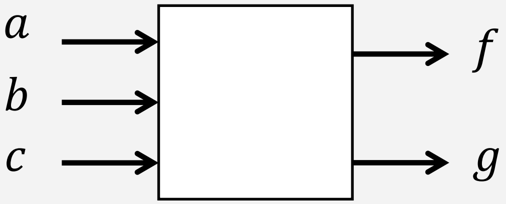
$$\bar{u}vyz + uv\bar{y}z + u\bar{v}yz + \bar{u}\,\bar{v}\,\bar{y}\,z$$

Multiple Outputs

- Suppose we have two functions: $f(a, b, c)$ and $g(a, b, c)$
- Same inputs: a, b, c, but two outputs: f and g
- We can minimize each function separately, or
- Minimize f and g as one circuit with two outputs
- The idea is to share terms (gates) among f and g



Two separate circuits



One circuit with Two Outputs

Multiple Outputs: Example 1

Given: $f(a,b,c) = \sum(0,2,6,7)$ and $g(a,b,c) = \sum(1,3,6,7)$

Minimize each function separately

Minimize both functions as one circuit

K-Map of f

a \ bc	00	01	11	10
0	1	0	0	1
1	0	0	1	1

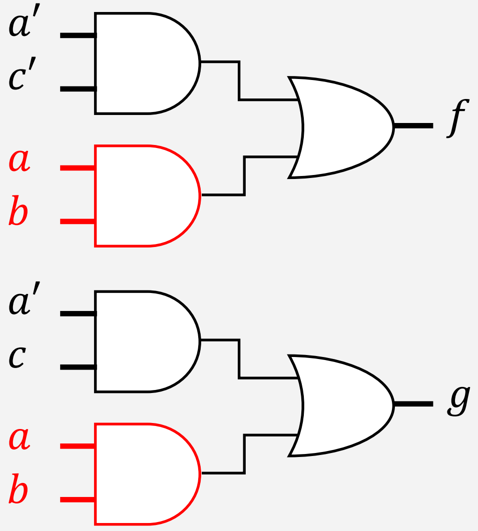
$f = a'c' + ab$

K-Map of g

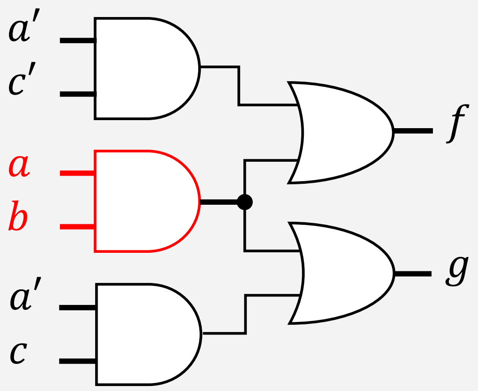
a \ bc	00	01	11	10
0	0	1	1	0
1	0	0	1	1

$g = a'c + ab$

Common Term = ab



One circuit per function



One circuit with two Outputs

Multiple Outputs: Example 2

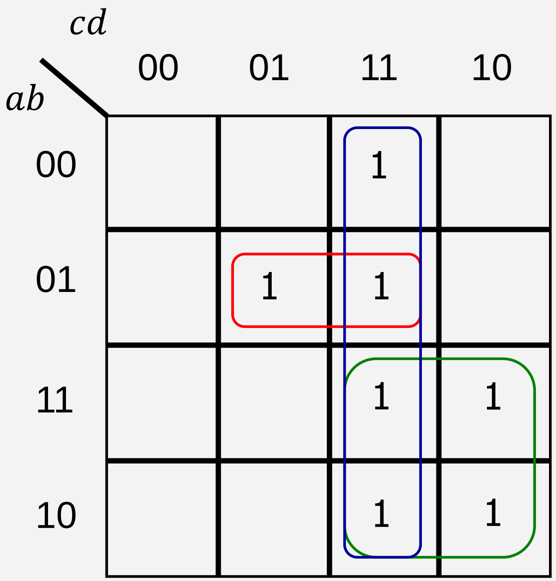
$f(a,b,c,d) = \sum(3,5,7,10,11,14,15)$, $g(a,b,c,d) = \sum(1,3,5,7,10,14)$

Draw the K-map and write minimal SOP expressions of f and g

$f = a'bd + ac + cd$ $\quad\quad$ $g = a'd + acd'$

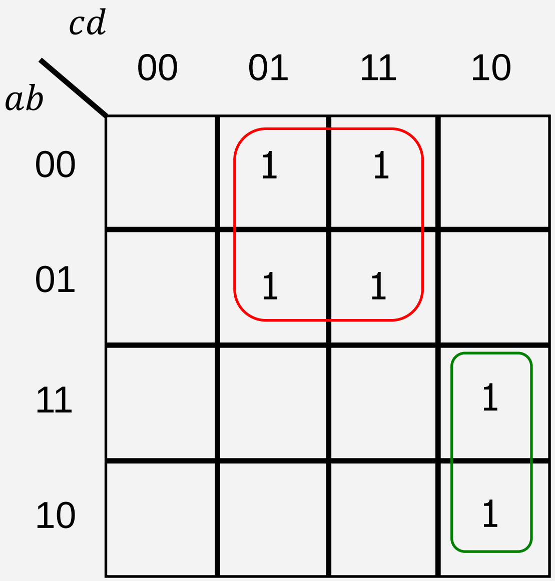
Extract the common terms of f and g

K-Map of f



ab \ cd	00	01	11	10
00			1	
01		1	1	
11			1	1
10			1	1

K-Map of g



ab \ cd	00	01	11	10
00		1	1	
01		1	1	
11				1
10				1

Common Terms

$T_1 = a'd$ and $T_2 = ac$

Minimal f and g

$f = T_1 b + T_2 + cd$

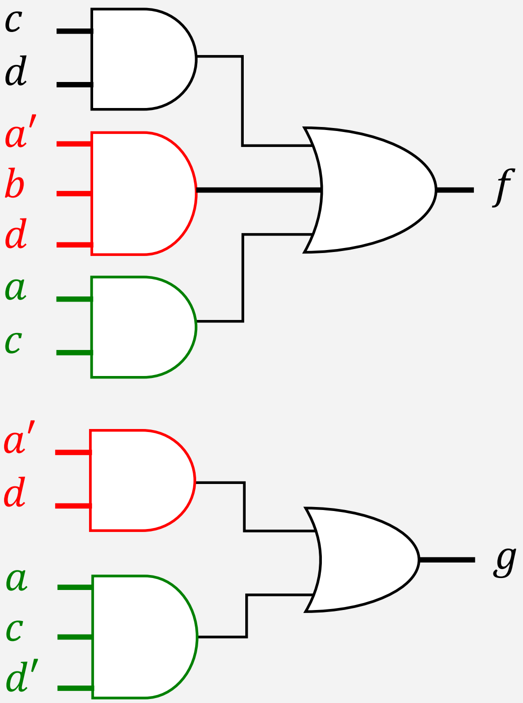
$g = T_1 + T_2 d'$

Common Terms ⇔ Shared Gates

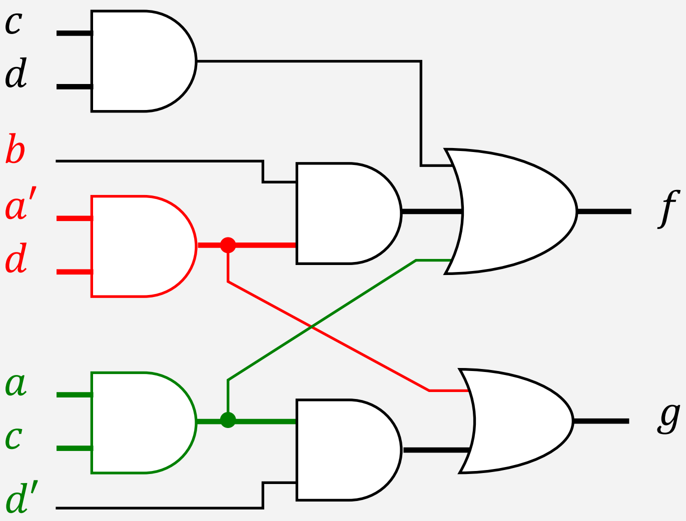
Minimal $f = a'bd + ac + cd$ Minimal $g = a'd + acd'$

Let $T_1 = a'd$ and $T_2 = ac$ (shared by f and g)

Minimal $f = T_1 b + T_2 + cd$, Minimal $g = T_1 + T_2 d'$



NO Shared Gates



One Circuit

Two Shared Gates

Implementing a Function Using NAND and NOR Gates

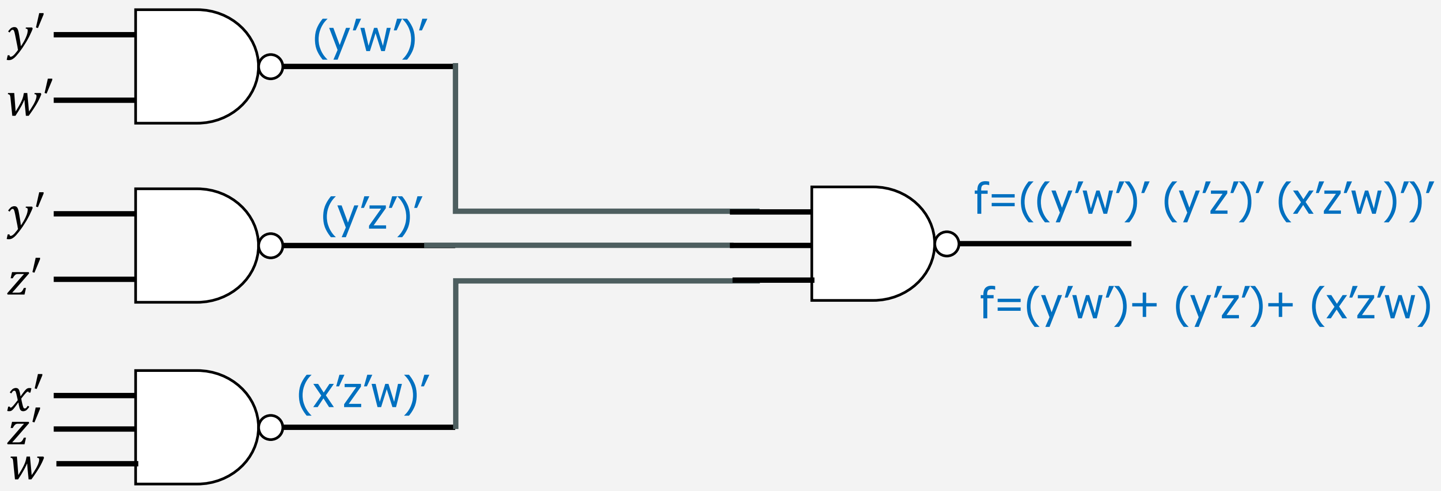
Due to the simplicity of manufacturing NAND and NOR gates compared to other logic gates, they are more commonly used in digital circuits.

- Rule for Implementing a Function Using NAND Gates (Two-Level Implementation):
  1. The function must be simplified and converted into Sum of Products (SOP) form.
  2. For each product term that includes at least two variables, draw a NAND gate. The variables in each term are connected to the inputs of the NAND gate. These gates form the first level.
  3. In the second level, draw another NAND gate whose inputs are the outputs of the first-level NAND gates.
  4. A product term consisting of only one variable in the first level requires an inverter (a single-input NAND gate).

Implementing a Function Using NAND and NOR Gates

Example:

f(x,y,w,z) = y′w′ + y′z′ + x′z′w



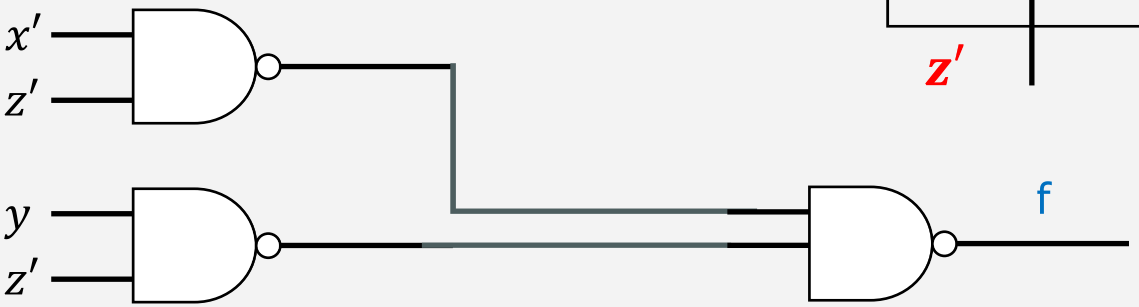
Implementing a Function Using NAND and NOR Gates

Example: Implement the following function using NAND gates.

$$f(x, y, z) = \sum(0,2,6)$$

f = y z′ + x′ z′

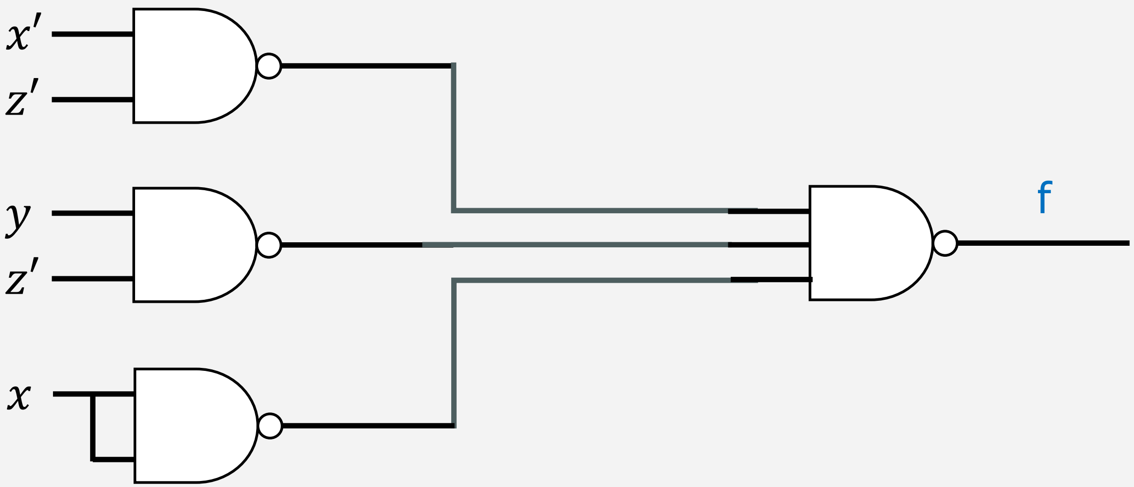
		y'		y	
x \ yz		00	01	11	10
x'	0	1	0	0	1
x	1	0	0	0	1
		z'	z		z'



Implementing a Function Using NAND and NOR Gates

Example: Implement the following function using NAND gates.

$$f = y\,z' + x'\,z' + x$$



Implementing a Function Using NAND and NOR Gates

- Rule for Implementing a Function Using NOR Gates (Two-Level Implementation):

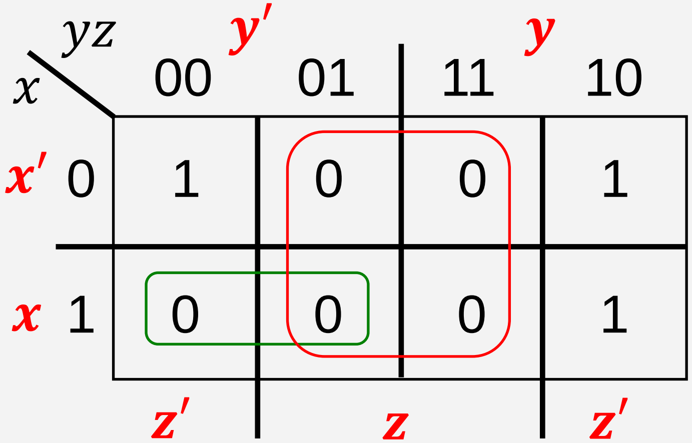
1. The function must be simplified and converted into Product of Sums (SOP) form.
2. For each product term that includes at least two variables, draw a NOR gate. The variables in each term are connected to the inputs of the NOR gate. These gates form the first level.
3. In the second level, draw another NOR gate whose inputs are the outputs of the first-level NAND gates.
4. A product term consisting of only one variable in the first level requires an inverter (a single-input NOR gate).

Implementing a Function Using NAND and NOR Gates

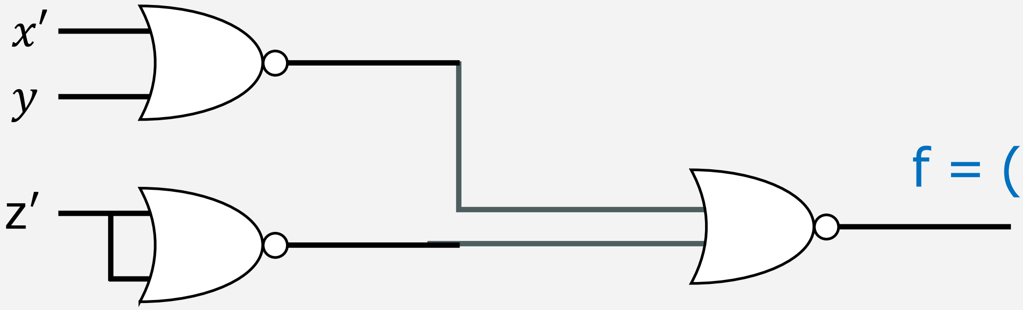
Example: Implement the following function using NOR gates.

$$f(x,y,z) = \sum(0,2,6)$$

f = (z′) . (x′ + y)



x \ yz	00	01	11	10
x' 0	1	0	0	1
x 1	0	0	0	1



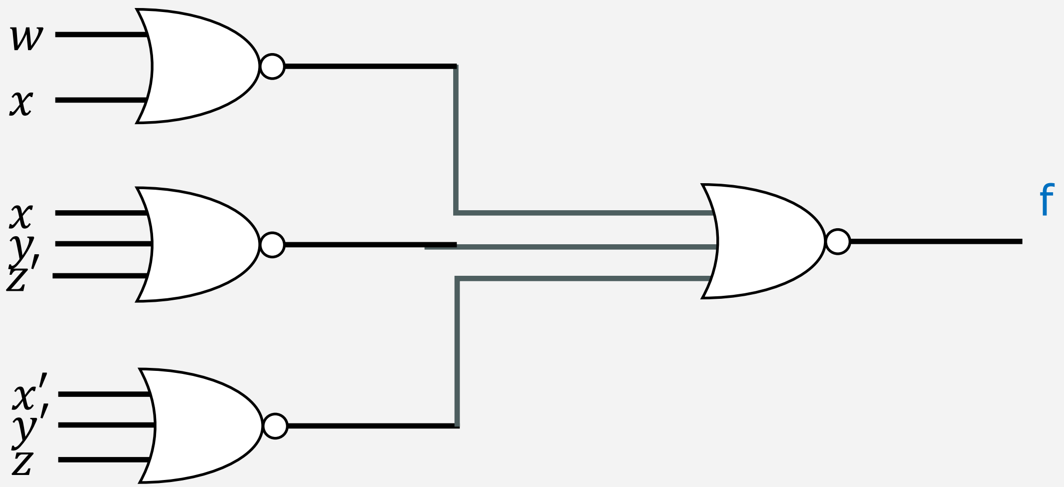
f = (z′) . (x′ + y)

Implementing a Function Using NAND and NOR Gates

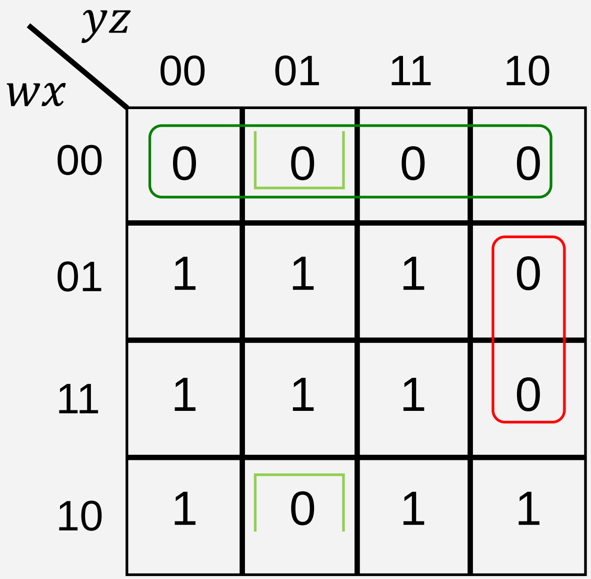
Example: Simplify the following function using the POS method.

$$f(w,x,y,z) = \prod(0,1,2,3,6,9,14)$$

$$f = (w + x) \cdot (x + y + z') \cdot (x' + y' + z)$$



K-Map of f



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

razeghizade@gmail.com

Razeghizade.pudica.ir

CREADITS: This presentation was created by M.Razeghizadeh
Please keep this slide for attribution