Fundamentals of Electronics

ECE 101

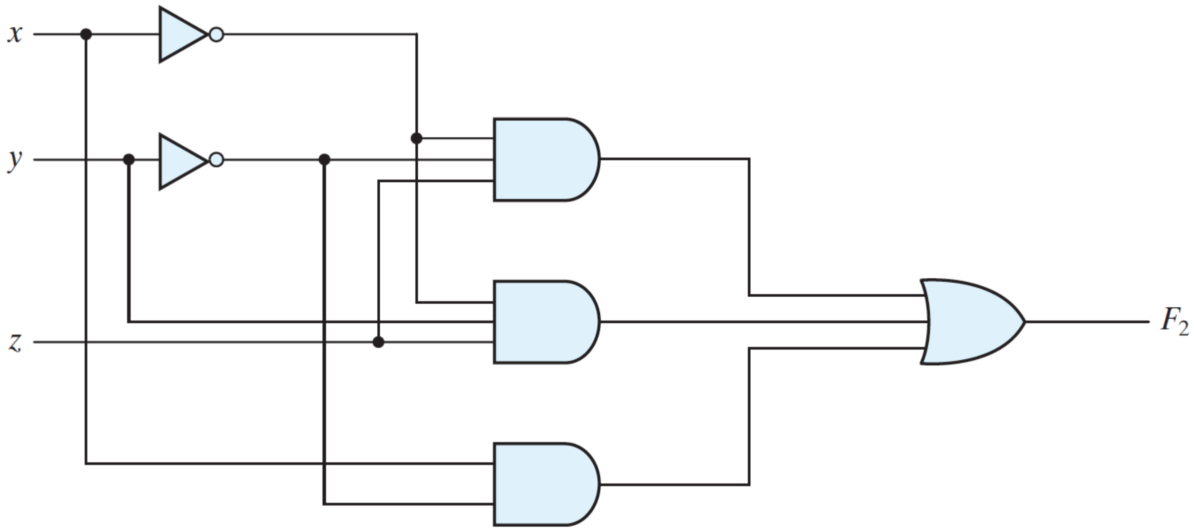
Digital Electronics

Boolean Functions

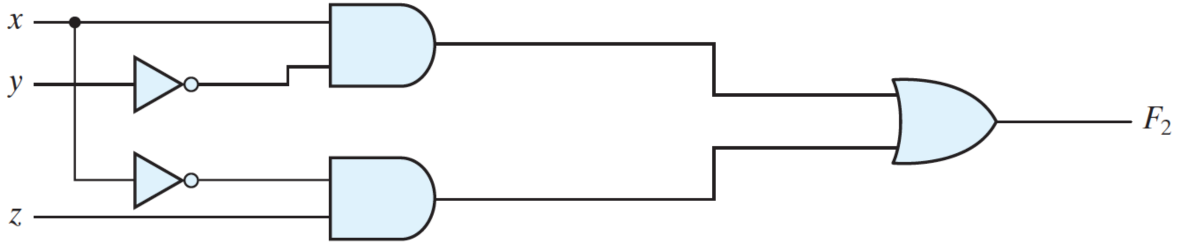
$$f : \{0,1\}^k \rightarrow \{0,1\}$$

$$F_2 = x'y'z + x'yz + xy'$$

$$F_2 = x'y'z + x'yz + xy' = x'z(y' + y) + xy' = x'z + xy'$$



(a) $F_2 = x'y'z + x'yz + xy'$



(b) $F_2 = xy' + x'z$

Other Boolean Functions

Truth Tables for the 16 Functions of Two Binary Variables

x	y	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Boolean Expressions for the 16 Functions of Two Variables

Boolean Functions	Operator Symbol	Name	Comments
$F_0 = 0$		Null	Binary constant 0
$F_1 = xy$	$x \cdot y$	AND	x and y
$F_2 = xy'$	x/y	Inhibition	x, but not y
$F_3 = x$		Transfer	x
$F_4 = x'y$	y/x	Inhibition	y, but not x
$F_5 = y$		Transfer	y
$F_6 = xy' + x'y$	$x \oplus y$	Exclusive-OR	x or y, but not both
$F_7 = x + y$	$x + y$	OR	x or y
$F_8 = (x + y)'$	$x \downarrow y$	NOR	Not-OR
$F_9 = xy + x'y'$	$(x \oplus y)'$	Equivalence	x equals y
$F_{10} = y'$	y'	Complement	Not y
$F_{11} = x + y'$	$x \subset y$	Implication	If y, then x
$F_{12} = x'$	x'	Complement	Not x
$F_{13} = x' + y$	$x \supset y$	Implication	If x, then y
$F_{14} = (xy)'$	$x \uparrow y$	NAND	Not-AND
$F_{15} = 1$		Identity	Binary constant 1

Canonical and Standard Forms

Minterms and Maxterms

Product (AND), Sum (OR)

- Each maxterm is obtained from an OR term of the n variables, with each variable being unprimed if the corresponding bit is a 0 and primed if a 1
- Each maxterm is the complement of its corresponding minterm and vice versa.

			Minterms		Maxterms	
x	**y**	**z**	**Term**	**Designation**	**Term**	**Designation**
0	0	0	$x'y'z'$	m_0	$x + y + z$	M_0
0	0	1	$x'y'z$	m_1	$x + y + z'$	M_1
0	1	0	$x'yz'$	m_2	$x + y' + z$	M_2
0	1	1	$x'yz$	m_3	$x + y' + z'$	M_3
1	0	0	$xy'z'$	m_4	$x' + y + z$	M_4
1	0	1	$xy'z$	m_5	$x' + y + z'$	M_5
1	1	0	xyz'	m_6	$x' + y' + z$	M_6
1	1	1	xyz	m_7	$x' + y' + z'$	M_7

Representation of Boolean Functions

Minterms: A Boolean function can be expressed algebraically from a given truth table by forming a minterm for each combination of the variables that produces a 1 in the function and then taking the OR of all those terms.

Example:

x	**y**	**z**	**Function f_1**	**Function f_2**
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$f_1 = x'y'z + xy'z' + xyz = m_1 + m_4 + m_7$$

$$f_2 = x'yz + xy'z + xyz' + xyz = m_3 + m_5 + m_6 + m_7$$

$$f_1' = x'y'z' + x'yz' + x'yz + xy'z + xyz'$$

$$f_1 = (x + y + z)(x + y' + z)(x' + y + z')(x' + y' + z) = M_0 \cdot M_2 \cdot M_3 \cdot M_5 \cdot M_6$$

$$f_2 = (x + y + z)(x + y + z')(x + y' + z)(x' + y + z) = M_0M_1M_2M_4$$

- **Form a maxterm for each combination of the variables that produces a 0 in the function,**
- **Then form the AND of all those maxterms.**

Boolean functions expressed as a sum of minterms or product of maxterms are said to be in canonical form

Sum of minterms

The minterms whose sum defines the Boolean function are those which give the 1's of the function in the truth table.

Product of maxterms

The maxterms whose product defines the Boolean function are those which give the 0's of the function in the truth table.

Sum of minterms representation:

$$F = A + B'C$$

$$A = A(B + B') = AB + AB'$$

$$\begin{aligned} A &= AB(C + C') + AB'(C + C') \\ &= ABC + ABC' + AB'C + AB'C' \end{aligned}$$

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

$$\begin{aligned} F &= A + B'C \\ &= ABC + ABC' + AB'C + AB'C' + A'B'C \end{aligned}$$

$$F = A + B'C$$

$$\begin{aligned} F &= A'B'C + AB'C + AB'C + ABC' + ABC \\ &= m_1 + m_4 + m_5 + m_6 + m_7 \end{aligned}$$

$$F(A, B, C) = \Sigma(1, 4, 5, 6, 7)$$

Product of maxterms representation:

$$F = xy + x'z$$

$$\begin{aligned} F = xy + x'z &= (xy + x')(xy + z) \\ &= (x + x')(y + x')(x + z)(y + z) \\ &= (x' + y)(x + z)(y + z) \end{aligned}$$

$$\begin{aligned} x' + y &= x' + y + zz' = (x' + y + z)(x' + y + z') \\ x + z &= x + z + yy' = (x + y + z)(x + y' + z) \\ y + z &= y + z + xx' = (x + y + z)(x' + y + z) \end{aligned}$$

$$\begin{aligned} F &= (x + y + z)(x + y' + z)(x' + y + z)(x' + y + z') \\ &= M_0 M_2 M_4 M_5 \end{aligned}$$

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

Conversion between minterm and maxterm forms

$$F(A, B, C) = \Sigma(1, 4, 5, 6, 7)$$

$$F'(A, B, C) = \Sigma(0, 2, 3) = m_0 + m_2 + m_3$$

$$F = (m_0 + m_2 + m_3)' = m_0' \cdot m_2' \cdot m_3' = M_0 M_2 M_3 = \Pi(0, 2, 3)$$

$$m_j' = M_j$$

$$F = xy + x'z$$

$$F(x, y, z) = \Sigma(1, 3, 6, 7)$$

Truth Table for $F = xy + x'z$

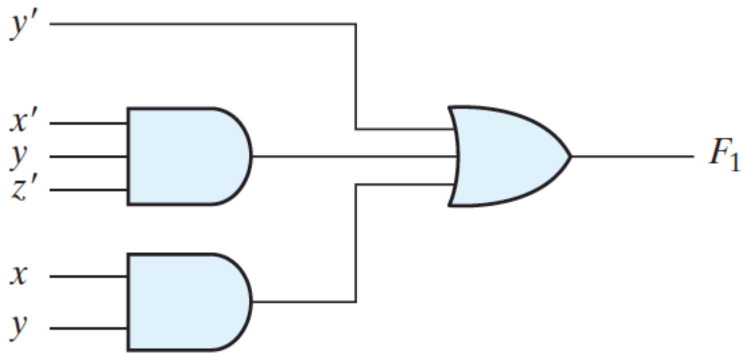
x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Minterms

Maxterms

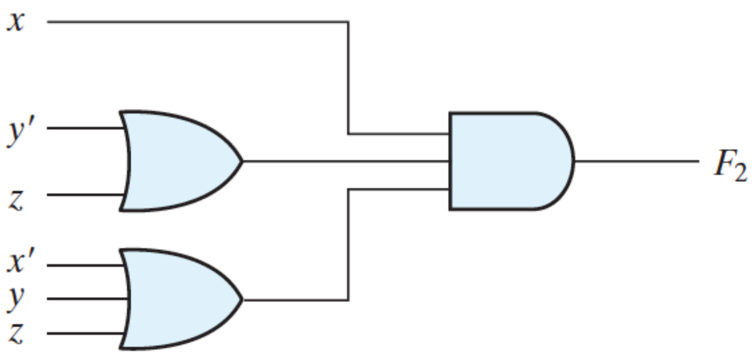
Standard Forms: SOPs and POSs

$$F_1 = y' + xy + x'yz'$$



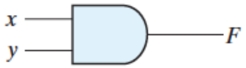
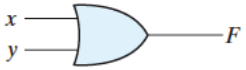
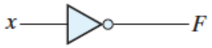
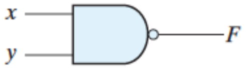
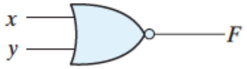
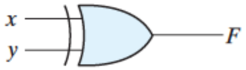
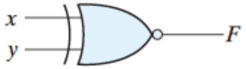
(a) Sum of Products

$$F_2 = x(y' + z)(x' + y + z')$$



(b) Product of Sums

Digital Logic Gates

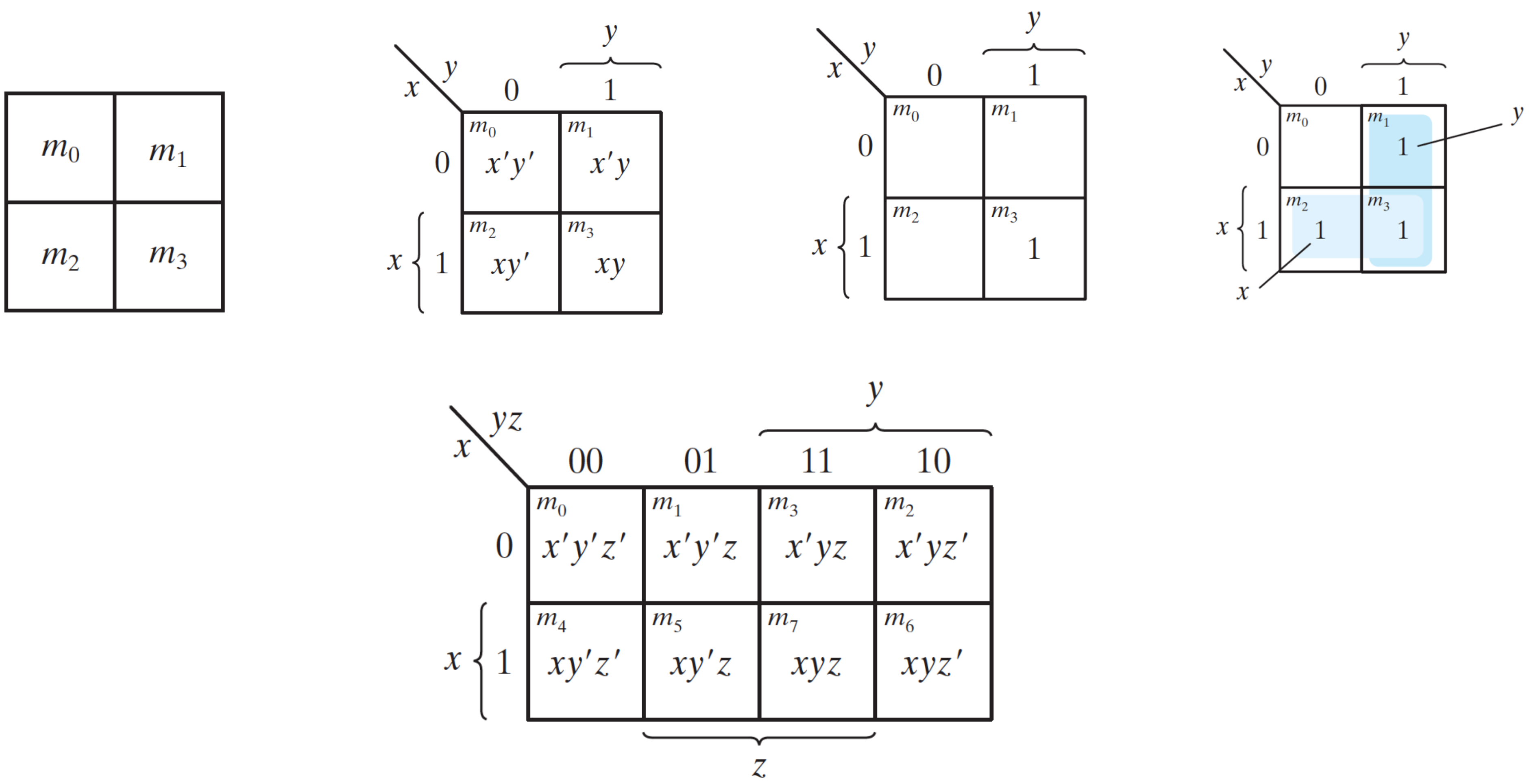
Name	Graphic symbol	Algebraic function	Truth table
AND		$F = x \cdot y$	x y | F 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR		$F = x + y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
Inverter		$F = x'$	x | F 0 | 1 1 | 0
Buffer		$F = x$	x | F 0 | 0 1 | 1
NAND		$F = (xy)'$	x y | F 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0
NOR		$F = (x + y)'$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0
Exclusive-OR (XOR)		$F = xy' + x'y$ $= x \oplus y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0
Exclusive-NOR or equivalence		$F = xy + x'y'$ $= (x \oplus y)'$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1

Gate Level Minimization

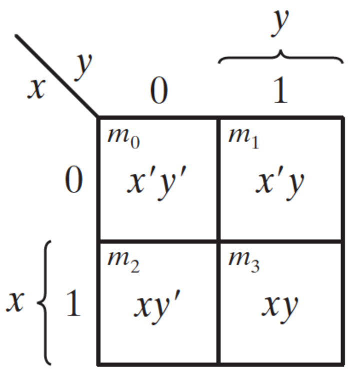
Finding an **optimal gate-level implementation** of the Boolean functions **describing a digital circuit**.

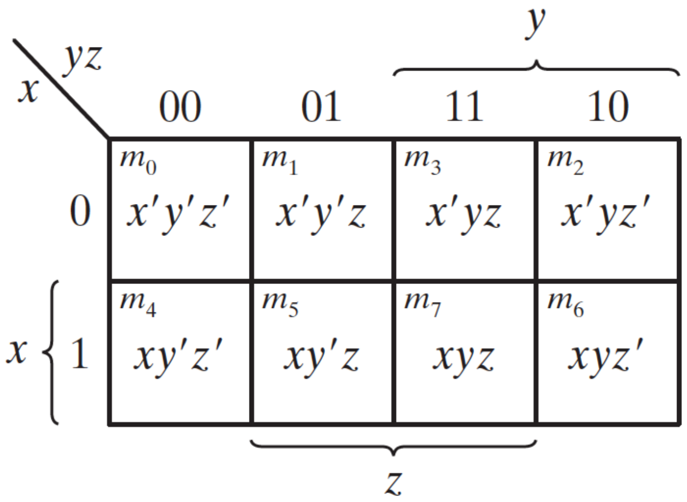
Karnaugh map or K-map

Pictorial form of the Truth Table



K-map





Any two adjacent squares in the map differ by only one variable, which is primed in one square and unprimed in the other – which of the terms are adjacent here?

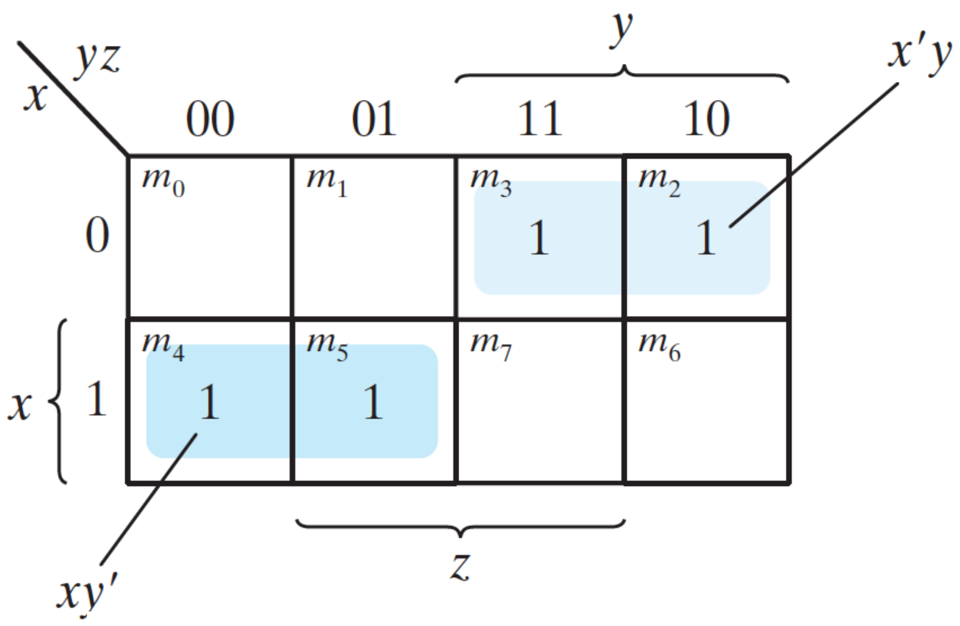
$$m_5 + m_7 = xy'z + xyz = xz(y' + y) = xz$$

Example:

Simplify this: $F(x, y, z) = \Sigma(2, 3, 4, 5)$

Example:

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



$$F = x'y + xy'$$

One square represents one minterm, giving a term with three literals.

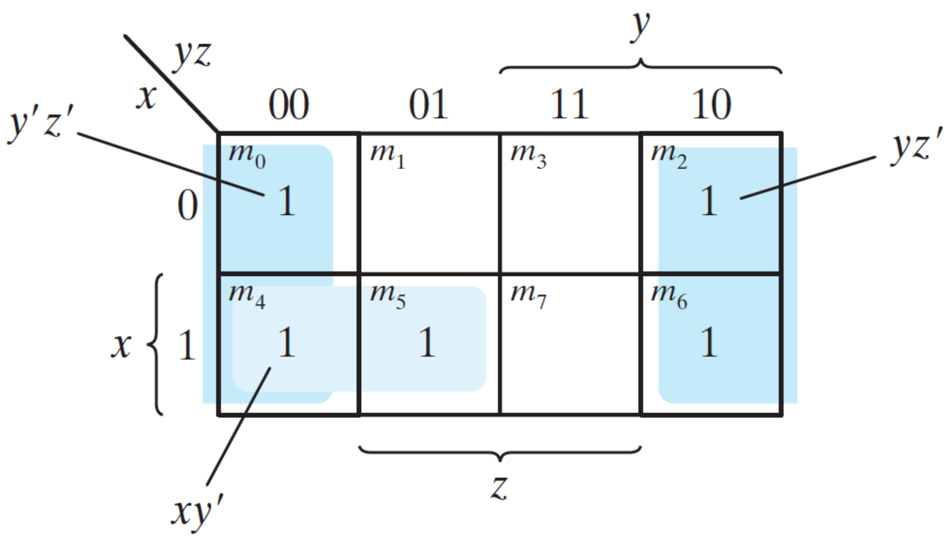
Two adjacent squares represent a term with two literals.

Four adjacent squares represent a term with one literal.

Eight adjacent squares encompass the entire map and produce a function that is always equal to 1.

Example: Simplify the following Boolean function

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



$$F = z' + xy'$$

4 variable K-map

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6
m_{12}	m_{13}	m_{15}	m_{14}
m_8	m_9	m_{11}	m_{10}

$wx \backslash yz$	00	01	11	10
00	m_0 $w'x'y'z'$	m_1 $w'x'y'z$	m_3 $w'x'yz$	m_2 $w'x'yz'$
01	m_4 $w'xy'z'$	m_5 $w'xy'z$	m_7 $w'xyz$	m_6 $w'xyz'$
11	m_{12} $wxy'z'$	m_{13} $wxy'z$	m_{15} $wxyz$	m_{14} $wxyz'$
10	m_8 $wx'y'z'$	m_9 $wx'y'z$	m_{11} $wx'yz$	m_{10} $wx'yz'$

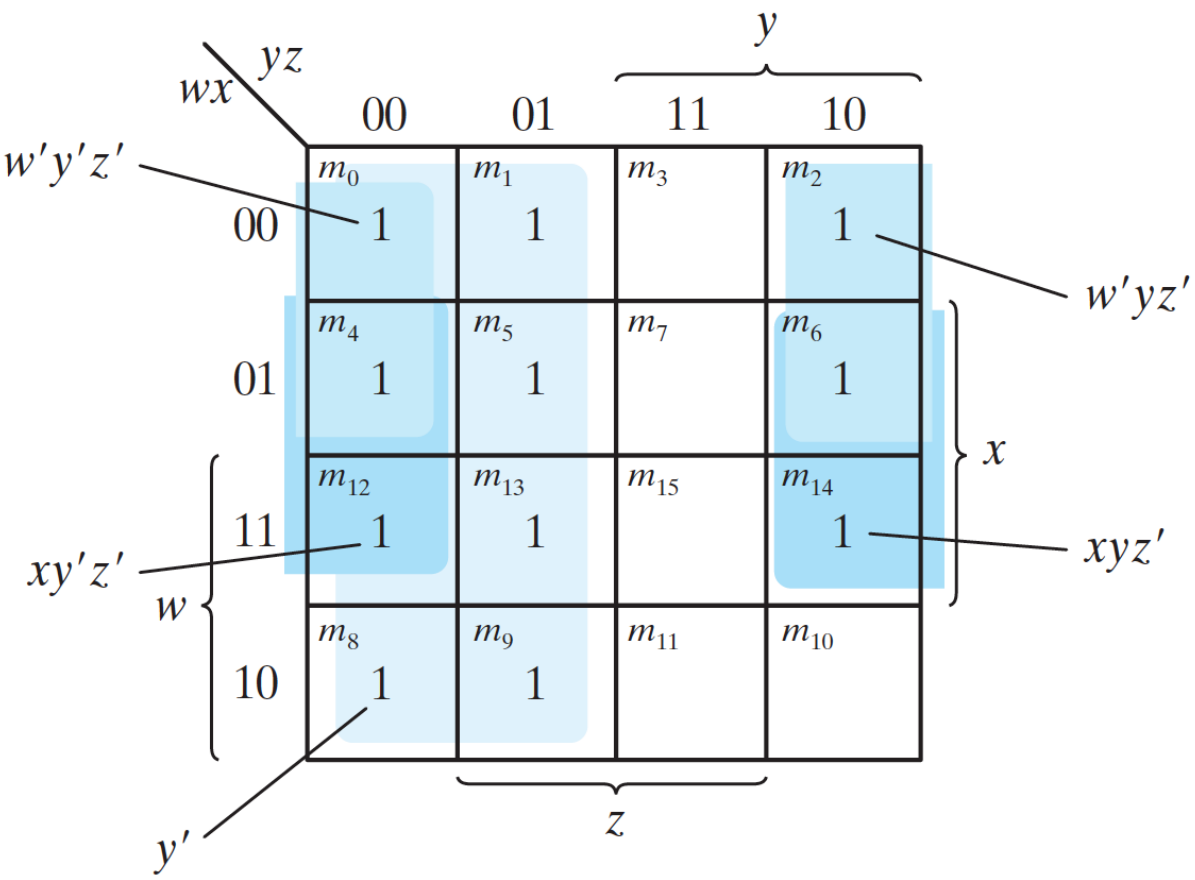
One square represents one minterm, giving a term with four literals.
Two adjacent squares represent a term with three literals.
Four adjacent squares represent a term with two literals.
Eight adjacent squares represent a term with one literal.
Sixteen adjacent squares produce a function that is always equal to 1.

Example: Simplify the following Boolean function

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

Example: Simplify the following Boolean function

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

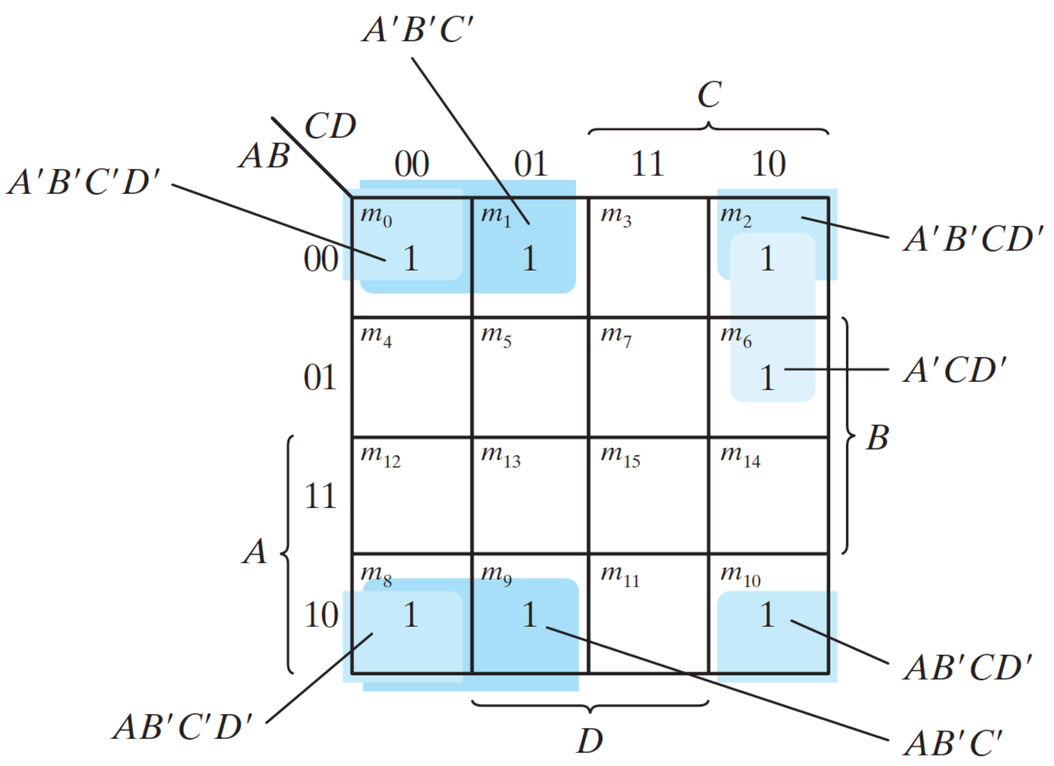


$$w'y'z' + w'yz' = w'z'$$
$$xy'z' + xyz' = xz'$$

$$F = y' + w'z' + xz'$$

Exercise: Simplify the following Boolean function

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



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

$$F = B'D' + B'C' + A'CD'$$

Thank you