

Today:

- Gate-level minimization – part 2
 - Four variable K-maps
 - POS implementation using K-maps
 - Incompletely specified functions & Don't care
 - Ex-OR being an “Odd function”
 - Application: Parity generator & checker

Four 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}

(a)

		y				
		yz	00	01	11	10
w	x	00	$w'x'y'z'$	$w'x'y'z$	$w'x'yz$	$w'x'yz'$
	01	$w'xy'z'$	$w'xy'z$	$w'xyz$	$w'xyz'$	
	11	$wxy'z'$	$wxy'z$	$wxyz$	$wxyz'$	
	10	$wx'y'z'$	$wx'y'z$	$wx'yz$	$wx'yz'$	

z

(b)

(b)

Remarks

- One square represents one minterm, giving a term of four literals.
- Two adjacent squares represent a term of three literals.
- Four adjacent squares represent a term of two literals.
- Eight adjacent squares represent a term of one literal.
- Sixteen adjacent squares represent the function equal to 1.

Example

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

		yz		y	
		00	01	11	10
w	x				
00	00	1	1		1
01	01	1	1		1
11	11	1	1		1
10	10	1	1		

Diagram illustrating the Karnaugh map for the function $F(w, x, y, z)$. The map is a 4x4 grid with rows labeled w and x (00, 01, 11, 10) and columns labeled y and z (00, 01, 11, 10). The function is 1 for the minterms 0, 1, 2, 4, 5, 6, 8, 9, 12, 13, and 14. The map shows three groups of 1s: a group of 4 cells (00, 01, 11, 10) for w and x (00, 01, 11, 10), a group of 4 cells (00, 01, 11, 10) for w and z (00, 01, 11, 10), and a group of 4 cells (00, 01, 11, 10) for x and z (00, 01, 11, 10).

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

Example

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

		CD		C	
AB		00	01	11	10
A	00	1	1		1
	01				1
	11				
	10	1	1		1

$\underbrace{\hspace{10em}}_D$
 $\left. \begin{array}{c} \text{ } \\ \text{ } \\ \text{ } \\ \text{ } \end{array} \right\} B$

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

Remarks

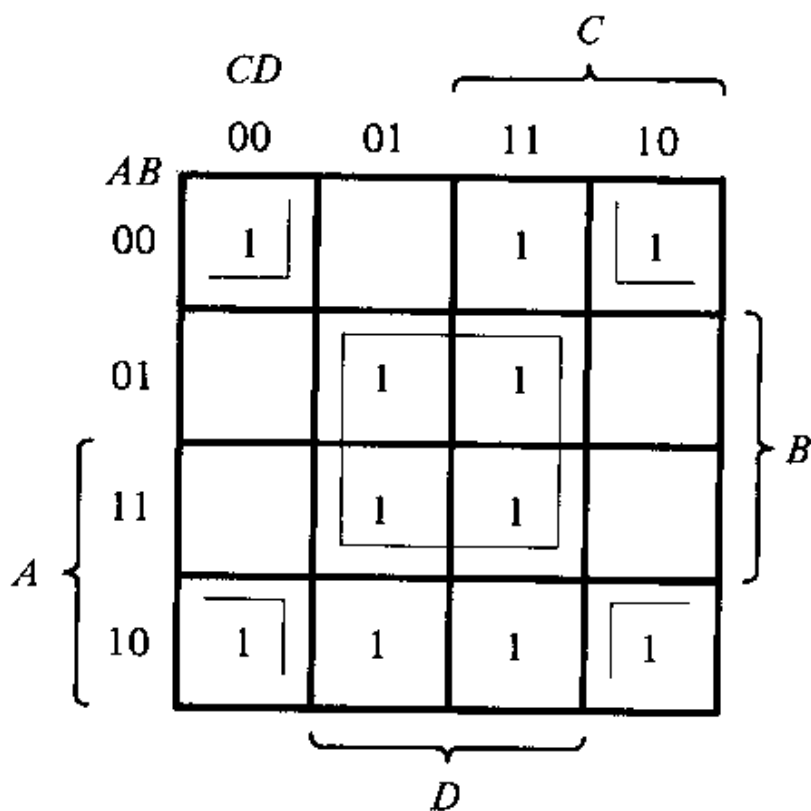
- Ensure following while choosing adjacent squares in a K-map:
 1. All the minterms of the function are covered when we combine the squares.
 2. The number of terms in the expression is minimized.
 3. There are no redundant terms, i.e., minterms already covered by other terms.
- Note:** Sometimes there may be two or more expressions that satisfy the simplification criteria.

A more systematic way to combine the squares

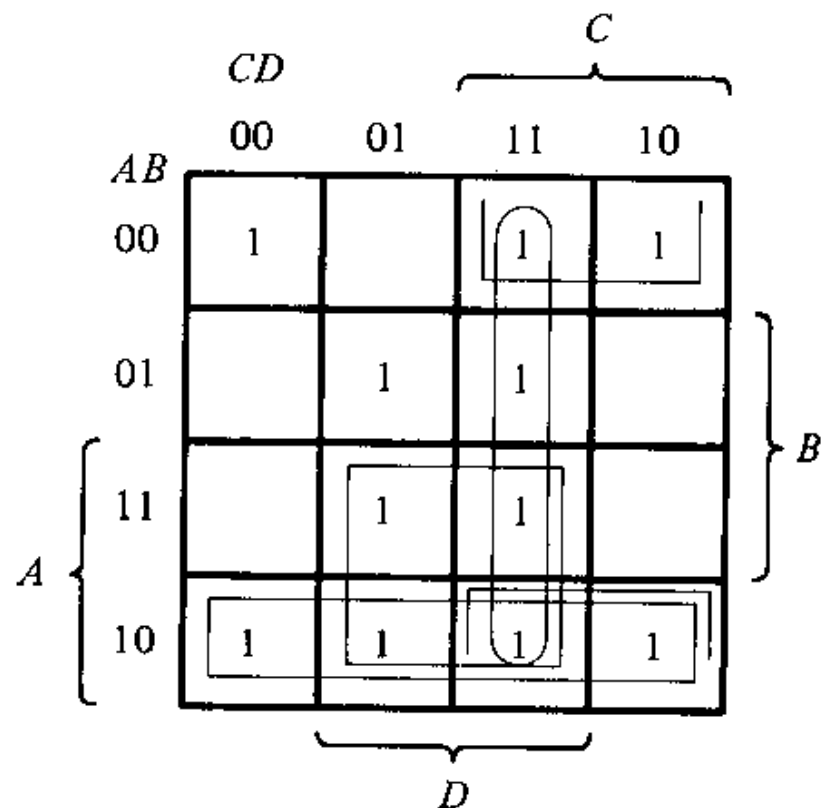
- *Prime implicant* is a product term obtained by combining the maximum possible number of adjacent squares in the K-map.
- If a minterm in a square is covered by only one prime implicant, that prime implicant is said to be *essential*.

Example

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



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



(b) Prime implicants CD , $B'C$, AD , and AB'

...continued

Four possible ways to express the corresponding function:

$$\begin{aligned} F &= BD + B'D' + CD + AD \\ &= BD + B'D' + CD + AB' \\ &= BD + B'D' + B'C + AD \\ &= BD + B'D' + B'C + AB' \end{aligned}$$

Product-of-Sums (POS)

Implementation using K-map

- Hint:

Use DeMorgan's property in the K-map and we can obtain the POS implementation instead of the SOP form.

Find POS implementation

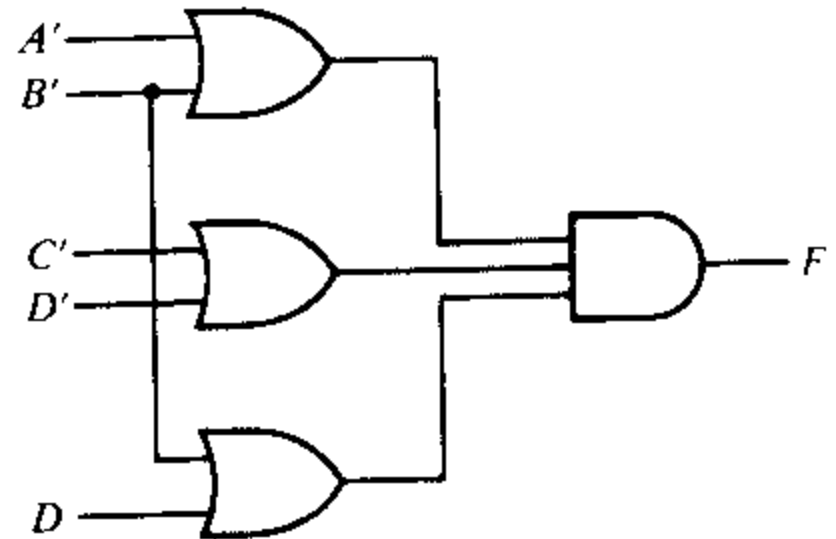
See that it is
SOP equation

$$F(A, B, C, D) = \Sigma(0, 1, 2, 5, 8, 9, 10)$$

		<i>CD</i>		<i>C</i>	
		00	01	11	10
<i>AB</i>	00	1	1	0	1
	01	0	1	0	0
	11	0	0	0	0
	10	1	1	0	1

A { 11, 10 } *B* { 01, 11 } *D* { 00, 10 }

$$(A' + B')(C' + D')(B' + D)$$



Find the SOP and POS implementations for given Truth-table

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

K-map representation

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

SOP expression

$$F(x, y, z) = \Sigma(1, 3, 4, 6)$$

POS expression

$$F(x, y, z) = \Pi(0, 2, 5, 7)$$

Incompletely specified functions

- Functions that have **unspecified** outputs for some input combinations are called *incompletely specified functions*.

meaning of **unspecified**:

e.g., in a design for an application, some combinations of the inputs are **not used** and hence considered as **unspecified**.

Don't care conditions

- The unspecified minterms of a function called as *don't-care conditions*.

meaning:

we simply don't care what value is assumed by such minterms, either 0 or 1.

- Symbol in the K-map: "X"
- Use in simplifying the Boolean functions.

Ex: function with don't-care conditions

$$F(w, x, y, z) = \Sigma(1, 3, 7, 11, 15) \quad \text{with} \quad d(w, x, y, z) = \Sigma(0, 2, 5)$$

		y			
		yz			
		00	01	11	10
w	00	X	1	1	X
	01	0	X	1	0
	11	0	0	1	0
	10	0	0	1	0

Diagram (a) shows a 4x4 Karnaugh map for function F. The horizontal axis is labeled yz (00, 01, 11, 10) and the vertical axis is labeled wx (00, 01, 11, 10). The map contains 1s at (00,11), (01,11), (11,11), (10,11), (01,01), (11,01), (10,01), and (10,00). Don't-care conditions (X) are at (00,00) and (10,00). A group of four 1s is circled, representing the term $w'x'$. A group of four 1s is also circled, representing the term yz . The function is $F = yz + w'x'$.

(a) $F = yz + w'x'$

		y			
		yz			
		00	01	11	10
w	00	X	1	1	X
	01	0	X	1	0
	11	0	0	1	0
	10	0	0	1	0

Diagram (b) shows a 4x4 Karnaugh map for function F. The horizontal axis is labeled yz (00, 01, 11, 10) and the vertical axis is labeled wx (00, 01, 11, 10). The map contains 1s at (00,11), (01,11), (11,11), (10,11), (01,01), (11,01), (10,01), and (10,00). Don't-care conditions (X) are at (00,00) and (01,00). A group of four 1s is circled, representing the term $w'z$. A group of four 1s is also circled, representing the term yz . The function is $F = yz + w'z$.

(b) $F = yz + w'z$