



O N L I N E

L E A R N I N G

Digital Electronics (SKEE1223)

Karnaugh Maps

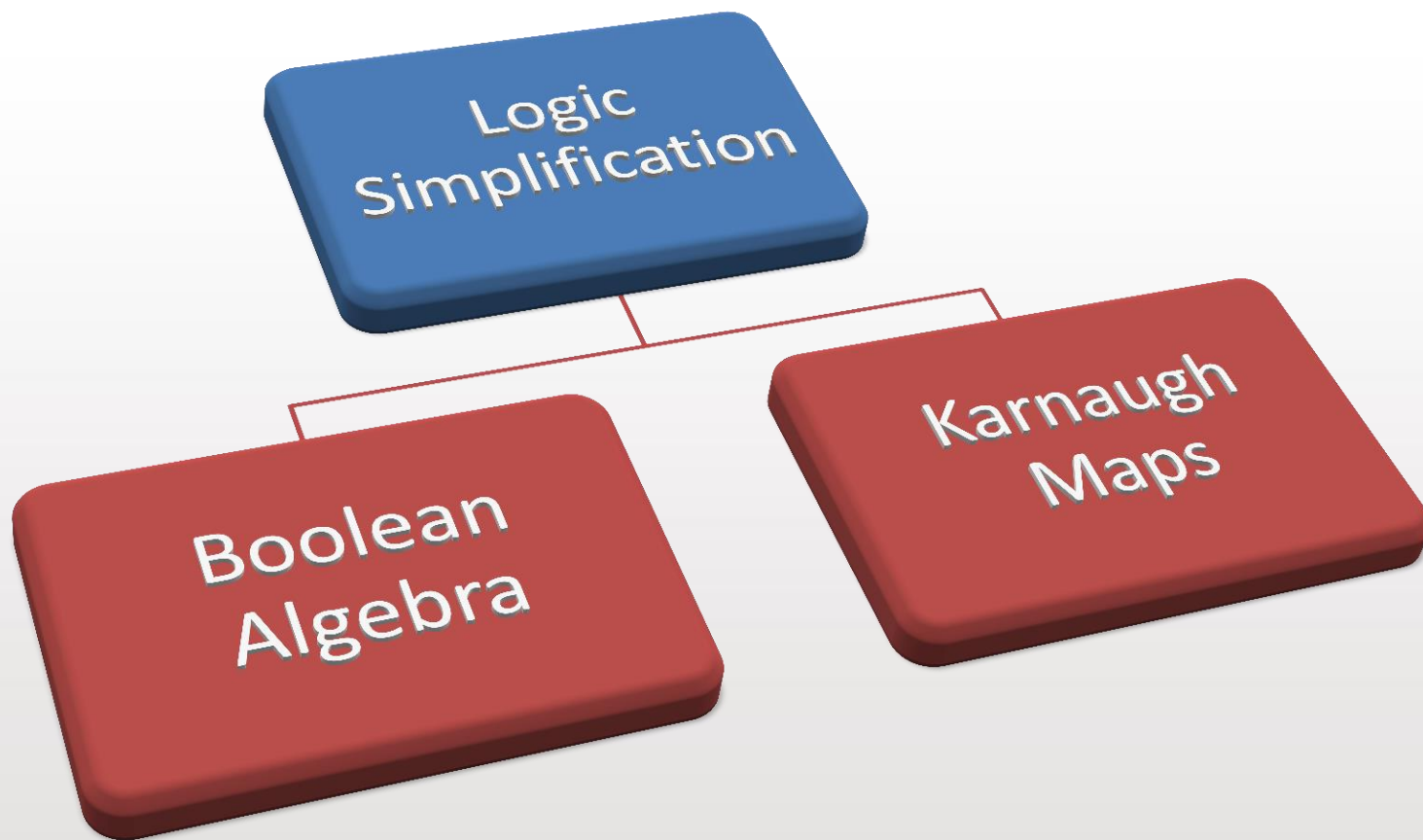
Muhammad Arif Abd Rahim
Muhammad Mun'ím Ahmad Zabidi
Ab Hadi Abd Rahman

Faculty of Electrical Engineering



Introduction

- A logic circuit can be built using logic gates connected according to its Boolean function.
- This type of circuit is known as **combinational logic circuit**.
- The best design is the simplest circuit derived from the simplest Boolean expression.
- Simplifying a Boolean function allows us to remove redundant gates.





Logic Simplification

- **Boolean Algebra**
 - Not guaranteed to obtain simplest solution
- **Karnaugh Maps**
 - Introduced by Maurice Karnaugh
 - A graphical method of writing the truth table
 - Quickly simplify Boolean functions up to 6 variables



Karnaugh Maps for 2-4 Variables

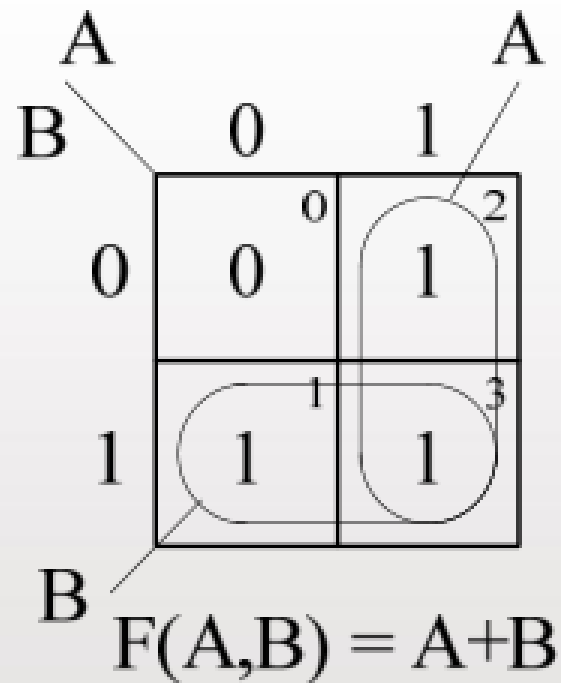
- Number of cells = 2^n , where n = number of inputs

		A	
		0	1
B	0	0	2
	1	1	3

		AB			
		00	01	11	10
C	0	0	2	6	4
	1	1	3	7	5

		AB			
		00	01	11	10
CD	00	0	4	12	8
	01	1	5	13	9
	11	3	7	15	11
	10	2	6	14	10

2-variable K-map



3-variable K-map

		AB		B. \bar{C}		
	C	00	01	11	10	
0		0 ⁰	1 ²	1 ⁶	0 ⁴	
1		0 ¹	0 ³	0 ⁷	0 ⁵	

$$F(A,B,C) = B.\bar{C}$$

3-variable K-map

		AB			
		00	01	11	10
C	0	0 ⁰	0 ²	0 ⁶	0 ⁴
	1	1 ¹	1 ³	1 ⁷	1 ⁵

C

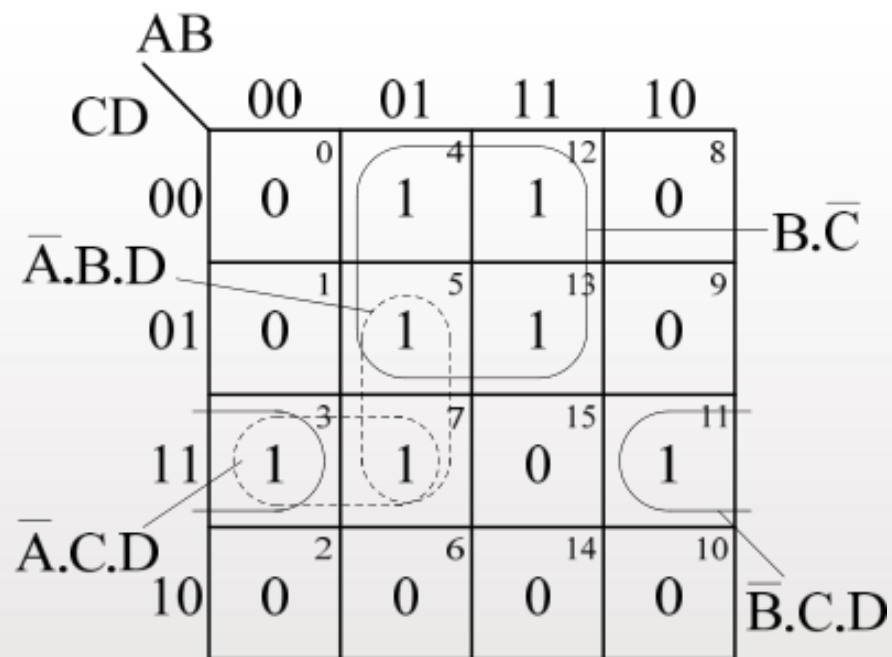
$$F(A,B,C) = C$$

3-variable K-map

		AB		B. \bar{C}	
	C	00	01	11	10
0	0	0 ⁰	1 ²	1 ⁶	0 ⁴
1	1	1 ¹	0 ³	0 ⁷	1 ⁵

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

4-variable K-map

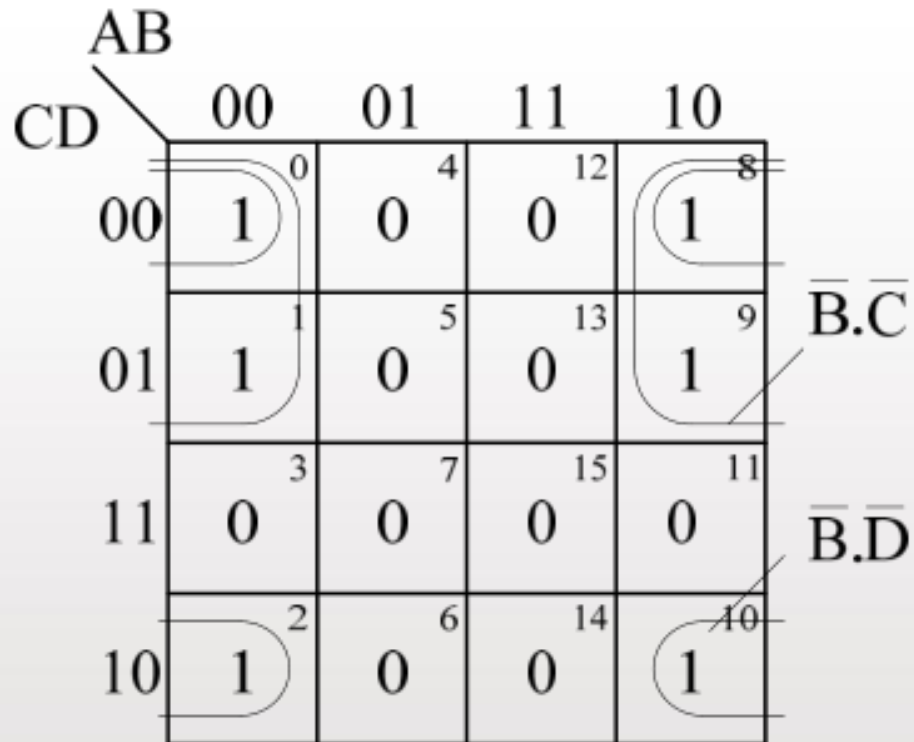


$$F(A,B,C,D) = B.\bar{C} + \bar{B}.C.D + \bar{A}.C.D$$

$$= B.\bar{C} + \bar{B}.C.D + \bar{A}.B.D$$

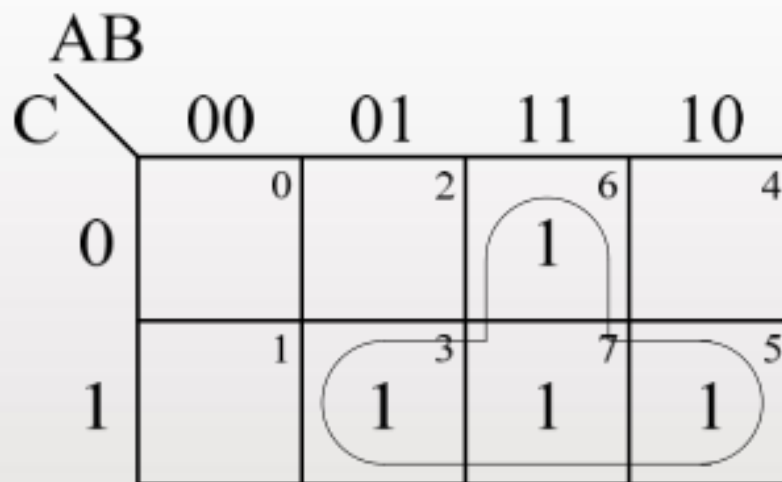
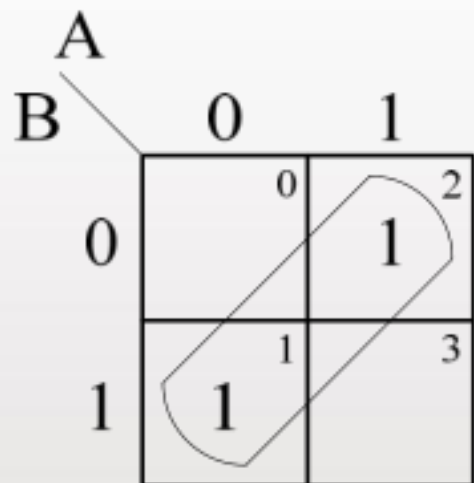


4-variable K-map



$$F(A,B,C,D) = \bar{B}.C + \bar{B}.D$$

Examples of Illegal Loops





Examples of Illegal Loops

CD \ AB	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

The table shows a 4x4 grid of cells. Each cell contains a number (0-15) and a '1'. The numbers are arranged in a 4x4 grid: 0, 4, 12, 8; 1, 5, 13, 9; 3, 7, 15, 11; 2, 6, 14, 10. The '1's are located in the following cells: (00, 11), (00, 10), (01, 01), (01, 11), (01, 10), (11, 01), (11, 11), (11, 10). There are two loops drawn around the '1's: one loop encircling the '1's at (00, 11), (00, 10), (01, 11), and (01, 10); another loop encircling the '1's at (01, 01), (01, 11), (11, 01), and (11, 11).

Looping 0's for POS Expressions

AB		$\bar{B} + \bar{C}$		$\bar{A} + \bar{B}$	
		00	01	11	10
C	0	1 ⁰	1 ²	0 ⁶	1 ⁴
	1	1 ¹	0 ³	0 ⁷	1 ⁵

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



Looping 0's for POS Expressions

	AB		B+C		
CD	00	01	11	10	
00	0 ⁰	1 ⁴	1 ¹²	0 ⁸	
01	0 ¹	1 ⁵	1 ¹³	0 ⁹	
11	1 ³	1 ⁷	0 ¹⁵	1 ¹¹	$\bar{A} + \bar{B} + \bar{C}$
10	0 ²	0 ⁶	0 ¹⁴	0 ¹⁰	$\bar{C} + D$

$$F(A,B,C,D) = (B+C).(\bar{C}+D).(\bar{A}+\bar{B}+\bar{C})$$

Don't Care Conditions

- Certain input combinations do not exist or are not allowed.
- They are labelled as X in the K-map.
- X can be assumed to be either 0 or 1.
- Can be used to simplify Boolean function by forming larger loops of 1's or 0's.
- It is unnecessary to loop the unused “don't cares”



Don't Care Conditions

	AB		B.C̄		
CD	00	01	11	10	
00	0 ⁰	1 ⁴	1 ¹²	0 ⁸	
01	0 ¹	1 ⁵	1 ¹³	X ⁹	A.D
11	1 ³	1 ⁷	X ¹⁵	1 ¹¹	C.D
10	0 ²	0 ⁶	0 ¹⁴	0 ¹⁰	

$$F(A,B,C,D) = B.C̄ + C.D$$