

# **Karnaugh Maps (K maps)**

---

# What are Karnaugh<sup>1</sup> maps?

---

- Karnaugh maps provide an alternative way of simplifying logic circuits.
- Instead of using Boolean algebra simplification techniques, you can transfer logic values from a Boolean statement or a truth table into a Karnaugh map.
- The arrangement of 0's and 1's within the map helps you to visualise the logic relationships between the variables and leads directly to a simplified Boolean statement.

---

<sup>1</sup>Named for the American electrical engineer Maurice Karnaugh.

# Karnaugh maps

- Karnaugh maps, or K-maps, are often used to simplify logic problems with 2, 3 or 4 variables.

**Cell =  $2^n$  ,where  $n$  is a number of variables**

**For the case of 2 variables, we form a map consisting of  $2^2=4$  cells as shown in Figure**

		A	
		0	1
B	0	$A + B$	$\bar{A} + B$
	1	$A + \bar{B}$	$\bar{A} + \bar{B}$

Maxterm

		A	
		0	1
B	0	00 <span style="color: red;">0</span>	10 <span style="color: red;">2</span>
	1	01 <span style="color: red;">1</span>	11 <span style="color: red;">3</span>

		A	
		0	1
B	0	$\bar{A}\bar{B}$	$A\bar{B}$
	1	$\bar{A}B$	$AB$

Minterm

# Karnaugh maps

---

- 3 variables Karnaugh map

$$\text{Cell} = 2^3 = 8$$

<div>AB C</div>		AB			
		00	01	11	10
C	0	<div>0</div> $\overline{A}\overline{B}\overline{C}$	<div>2</div> $\overline{A}B\overline{C}$	<div>6</div> $AB\overline{C}$	<div>4</div> $A\overline{B}\overline{C}$
	1	<div>1</div> $\overline{A}\overline{B}C$	<div>3</div> $\overline{A}BC$	<div>7</div> $ABC$	<div>5</div> $A\overline{B}C$

# Karnaugh maps

---

- 4 variables Karnaugh map

		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

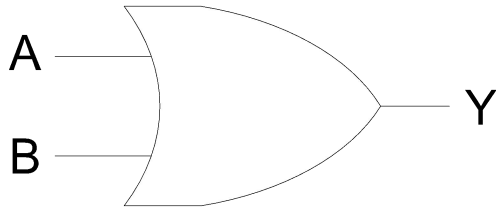
# Karnaugh maps

---

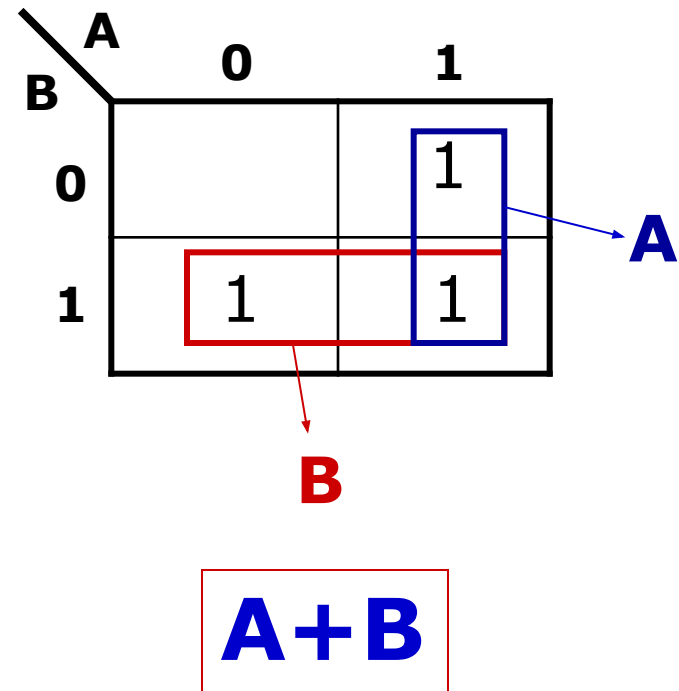
- The Karnaugh map is completed by entering a '1' (or '0') in each of the appropriate cells.
  - Within the map, adjacent cells containing 1's (or 0's) are grouped together in twos, fours, or eights.
-

# Example

2-variable Karnaugh maps are trivial but can be used to introduce the methods you need to learn. The map for a 2-input OR gate looks like this:

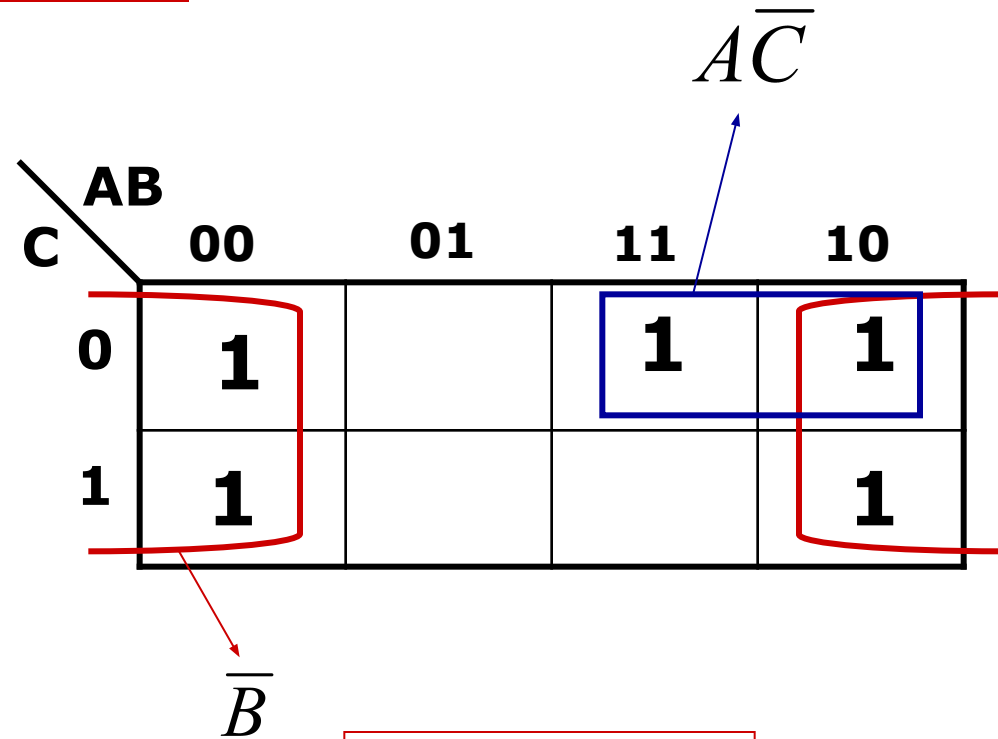


A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1



# Example

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0



$$\overline{B} + A\overline{C}$$



# Exercise

---

- Let us use Karnaugh map to simplify the follow function.

$$F_1 = m_0 + m_2 + m_3 + m_4 + m_5 + m_6 + m_7$$

$$F_2 = m_0 + m_1 + m_2 + m_5 + m_7$$

- Answer
-

# Exercise

---

**Given the truth table, find the simplified SOP and POS form.**

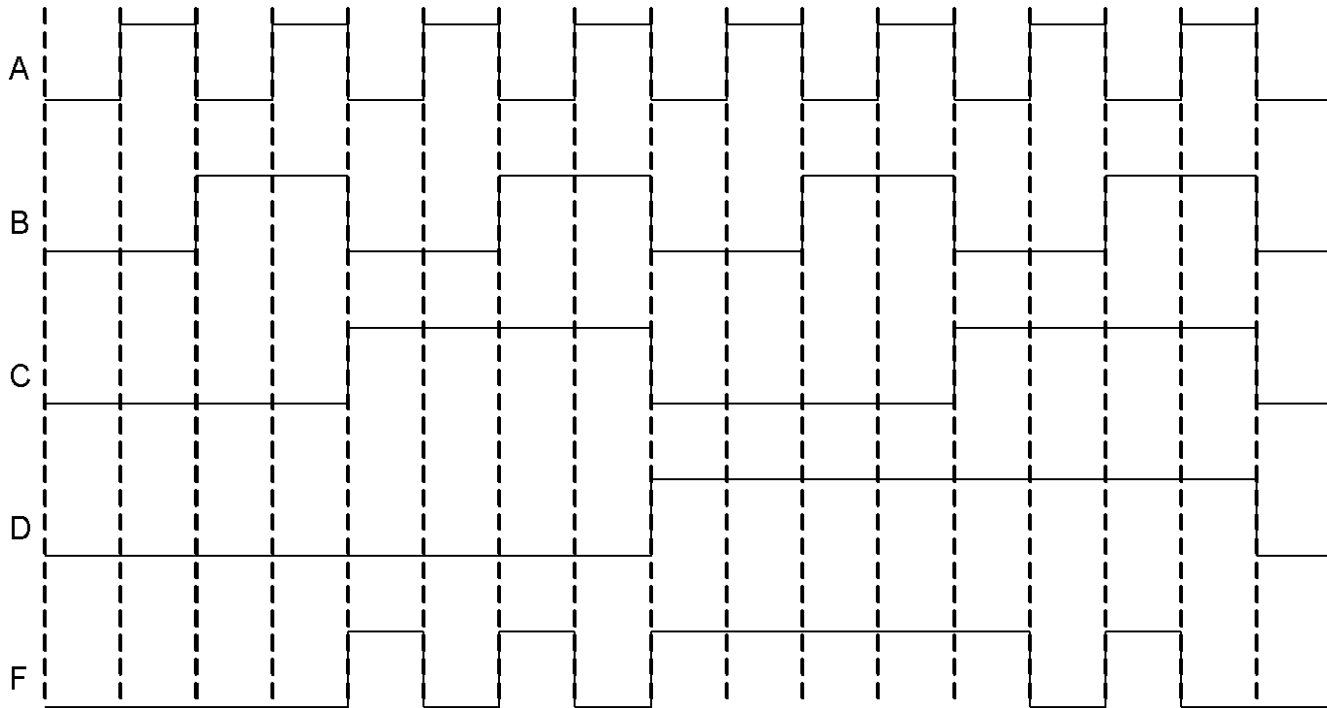
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

---

# Exercise

---

- Design two-level NAND-gate logic circuit from the follow timing diagram.



# Don't care term

A	B	C	D	X
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	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
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

		AB			
		00	01	11	10
CD	00			X	
	01			X	1
	11			X	X
	10			X	X

**AD**

# Exercise

---

- Design logic circuit that convert a 4-bits binary code to Excess-3 code

A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	x	x	x	x
1	0	1	1	x	x	x	x
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	X	X	X
1	1	1	1	X	X	x	x