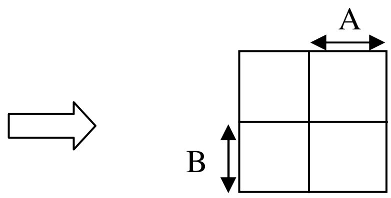
4. Boolean Minimisation Using Karnaugh Maps

4.1 Introduction to Karnaugh Maps

- The Karnaugh Map is a method for simplifying Boolean algebraic expressions in a structured manner.
- The Karnaugh Map (KM) was invented in 1952 by Edward Veitch, but later refined in 1953 by Maurice Karnaugh.
- It is a geometrical figure that has a ‘box’ or ‘cell’ to represent each row in a canonical truth table, i.e. each cell represents a canonical product term and the entire map represents all possible combinations of the input variables.
- In fact, the KM is simply a rearranged truth table such that adjacent cells differ by only one change of variable (as per the Gray code).
- For example, consider the 2-variable KM for the AND function:

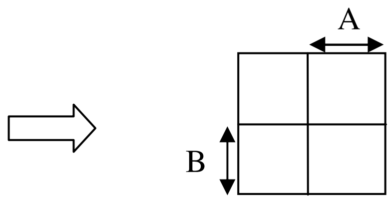
A	**B**	**f**
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**



A.B

- The 2-variable KM for the OR function is:

A	**B**	**f**
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**



A + B

- We will now look at the *3-variable* KM in more detail. This map (in detail) looks like:

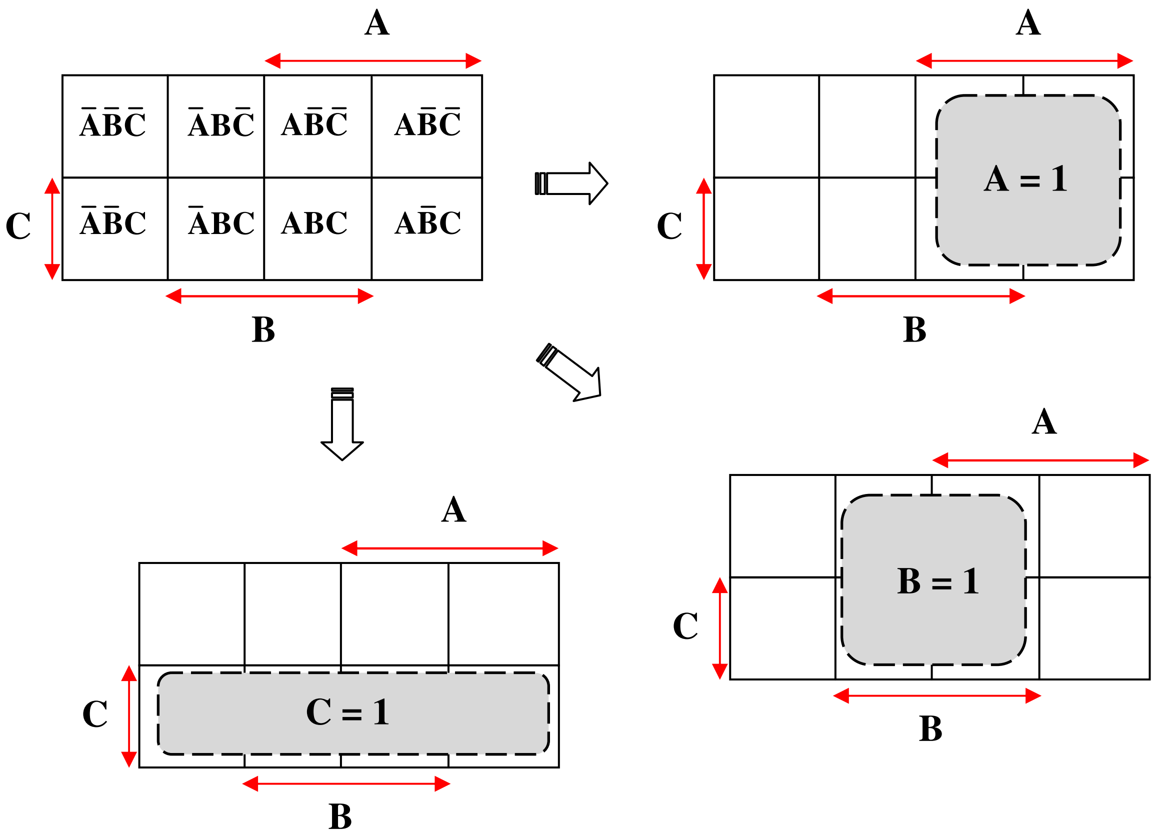
C \ **AB**	00	01	11	10
0	$\bar{A}\bar{B}\bar{C}$ (000)	$\bar{A}B\bar{C}$ (010)	$A\bar{B}\bar{C}$ (110)	$A\bar{B}\bar{C}$ (100)
1	$\bar{A}\bar{B}C$ (001)	$\bar{A}BC$ (011)	ABC (111)	$A\bar{B}C$ (101)

← *Note the Gray code here*

The function output (either 1 or 0) is entered in the appropriate cell location.

Each cell currently shows the combination of inputs associated with it.

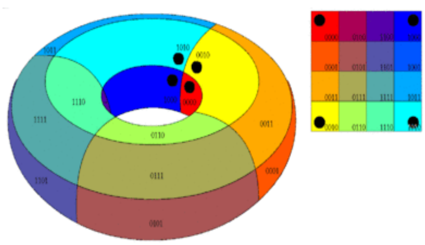
- More commonly:



- Note how each cell differs by only one change of variable (i.e. bit) from its adjacent cell in any vertical or horizontal direction (as per the Gray code system).

- It is important to note also that the concept of adjacency extends to cells along left and right edges.

- To aid this concept of adjacency, the top and bottom of the KM are regarded as being stitched together as are the left and right sides.



- Basically, ***we are looking at a 3-d model in 2-d space***.

- As already mentioned, the karnaugh map has a cell representing ***every possible canonical product term***.

- A canonical product term is more typically referred to as a **minterm**.

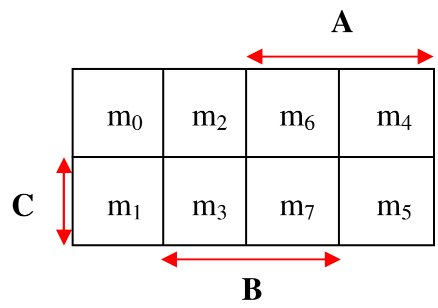
- Before we carry out minimisation using Karnaugh Maps, we need to first examine the concept of **minterms and maxterms** and understand the relationship between them.

4.2 Minterms and Maxterms

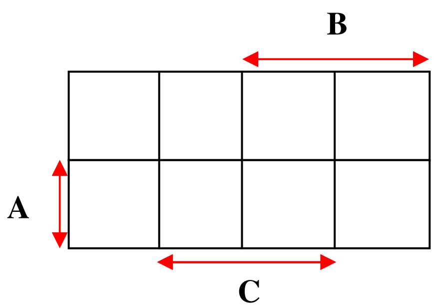
- A **minterm** of *n* variables is a logical **product** of all *n* literals.
- A **maxterm** of *n* variables is a logical **sum** of all *n* literals.
- Boolean functions can be represented using both minterms and maxterms.
- The list of minterms and maxterms for a 3-variable function are given below. This list can be easily extended for higher number of variables.

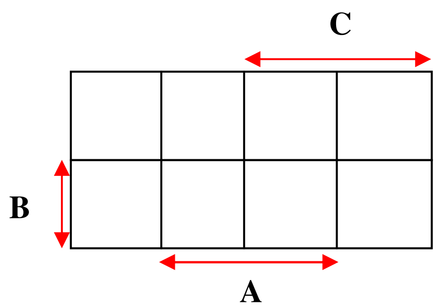
Decimal Notation	A B C	Minterm Term / designation	Maxterm Term / designation
0	0 0 0	$\bar{A}\,\bar{B}\,\bar{C}$ m_0	$A + B + C$ M_0
1	0 0 1	$\bar{A}\,\bar{B}\,C$ m_1	$A + B + \bar{C}$ M_1
2	0 1 0	$\bar{A}\,B\,\bar{C}$ m_2	$A + \bar{B} + C$ M_2
3	0 1 1	$\bar{A}\,B\,C$ m_3	$A + \bar{B} + \bar{C}$ M_3
4	1 00	$A\,\bar{B}\,\bar{C}$ m_4	$\bar{A} + B + C$ M_4
5	1 0 1	$A\,\bar{B}\,C$ m_5	$\bar{A} + B + \bar{C}$ M_5
6	1 1 0	$A\,B\,\bar{C}$ m_6	$\bar{A} + \bar{B} + C$ M_6
7	1 1 1	$A\,B\,C$ m_7	$\bar{A} + \bar{B} + \bar{C}$ M_7

- Using minterm notation, the KM now looks like:



- **Note:** The minterm pattern is determined by the position of the variables. When the position of the variables change, so too does the minterm position.





Relationship between Minterms and Maxterms …

- Consider the function:

$$f_{(A,B,C)} = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + A\overline{B}C + ABC$$

(Sum of Products)

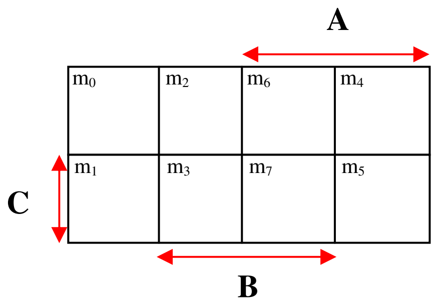
- This can be expressed as the canonical sum of minterms:

$$f_{(A,B,C)} = \Sigma\,(m_0, m_1, m_2, m_5, m_7)$$

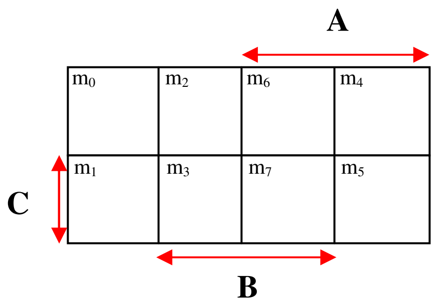
or

$$f_{(A,B,C)} = \Sigma\,(0, 1, 2, 5, 7)$$

- The Karnaugh Map for $f_{(A,B,C)}$ is:



- And the KM for $\overline{f_{(A,B,C)}}$ is:



- This $\overline{f_{(A,B,C)}}$ can be expressed as the canonical sum of minterms:

$$\overline{f_{(A,B,C)}} = \Sigma\,(3, 4, 6)$$

or

$$\overline{f_{(A,B,C)}} = \Sigma\,(m_3, m_4, m_6)$$

or

$$\overline{f_{(A,B,C)}} = m_3 + m_4 + m_6$$

- Therefore, $f_{(A,B,C)}$ can also be expressed as:

$$f_{(A,B,C)} = \overline{m_3 + m_4 + m_6}$$

- Using De Morgan's $\overline{(X + Y + Z)} = \overline{X}\overline{Y}\overline{Z}$, then:

$$f_{(A,B,C)} = \overline{m_3}.\overline{m_4}.\overline{m_6}$$

- But: $m_3 = \overline{A}BC$
  $= \overline{A + \overline{B} + \overline{C}}$ (*De Morgan's*)
  $= \overline{M_3}$

- Hence: $\overline{m_3} = M_3$ (*i.e. a minterm is the complement of the corresponding maxterm and vice versa*)

- Therefore:

$$f_{(A,B,C)} = M_3.\ M_4.\ M_6$$

or

$$f_{(A,B,C)} = \Pi\ (M_3, M_4, M_6)$$

or

$$f_{(A,B,C)} = \Pi\ (3, 4, 6)$$

or

$$f_{(A,B,C)} = (A + \overline{B} + \overline{C})\ (\overline{A} + B + C)\ (\overline{A} + \overline{B} + C)$$

- In summary:

  if $f_{(A,B,C)} = \Sigma\ (0, 1, 2, 5, 7)$

  then $f_{(A,B,C)} = \Pi\ (3, 4, 6)$

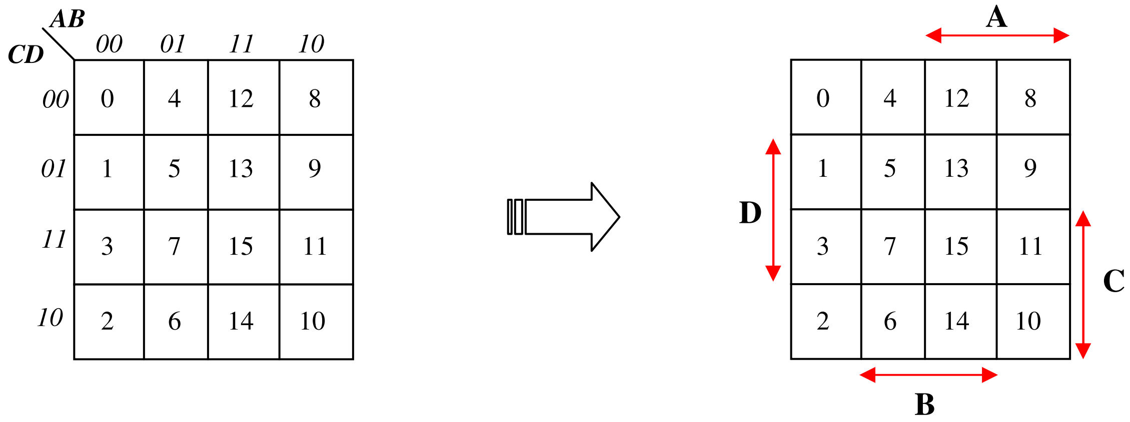
- For example:

  if $f_{(W,X,Y,Z)} = \Sigma\ (0, 1, 2, 3, 5, 7, 11, 13)$

  then $f_{(W,X,Y,Z)} =$

4.3 A 4-variable Karnaugh Map

- A *4-variable* KM is obtained as follows:



- Remember that all adjacent cells differ by only one change of variable.

- Note that the 'm' notation has been dropped for convenience.

4.4 Completing a Karnaugh Map

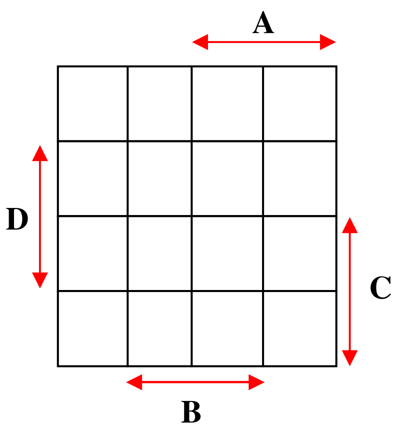
- A Karnaugh Map is completed by filling in 1's or 0's in each cell to indicate whether the output for that particular combination of inputs is high or low.

- When the function is expressed in terms of minterms, then filling the Karnaugh Map is straightforward as we have seen already. This is because the minterm locations are readily available.

- For example, fill a 4-variable Karnaugh Map to represent the function:

$$f_{(A,B,C,D)} = \sum (0, 1, 2, 5, 7, 11, 12)$$

- Thus we get the following KM:



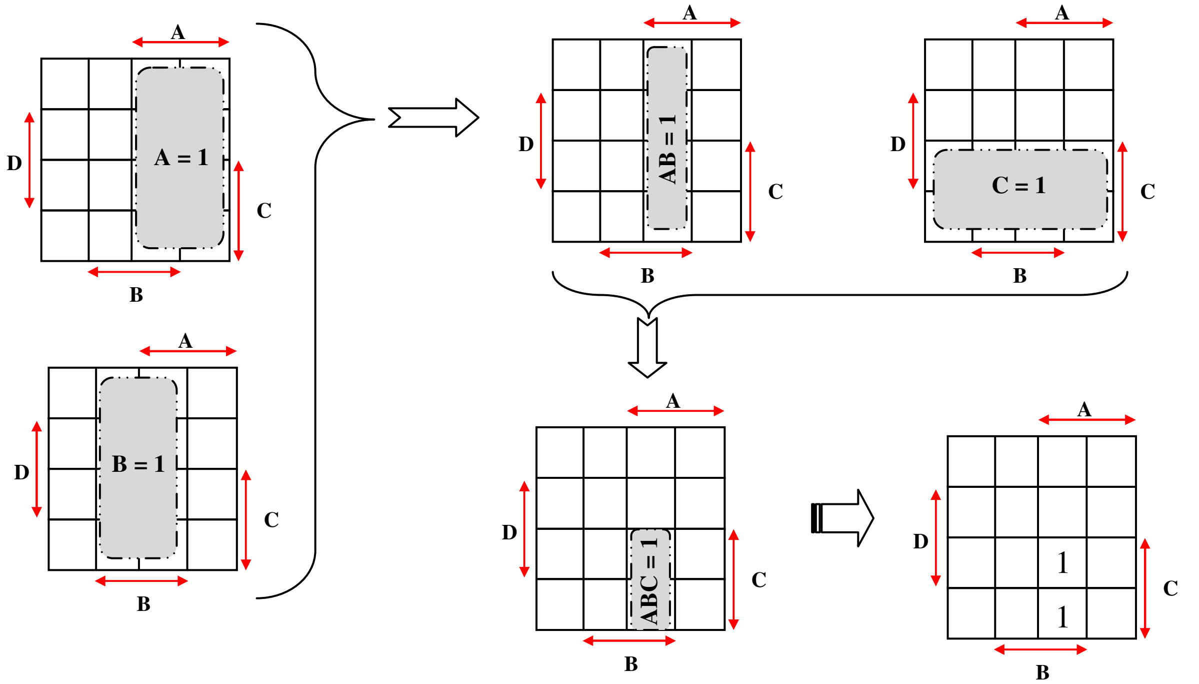
- If, however, the function is expressed in terms of its variables (and not necessarily in canonical form), then a little more care needs to be taken in order to complete the KM. However, this should still be a relatively straightforward task.

- For example, fill a 4-variable Karnaugh Map to represent the following function:

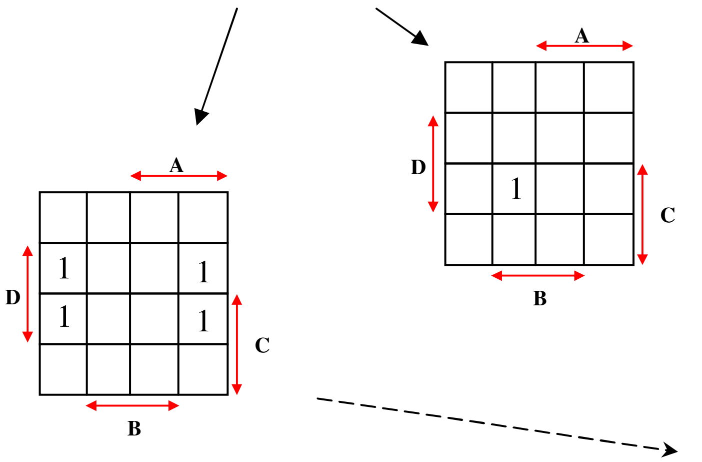
$$f_{(A,B,C,D)} = ABC + \overline{B}D + \overline{A}BCD$$

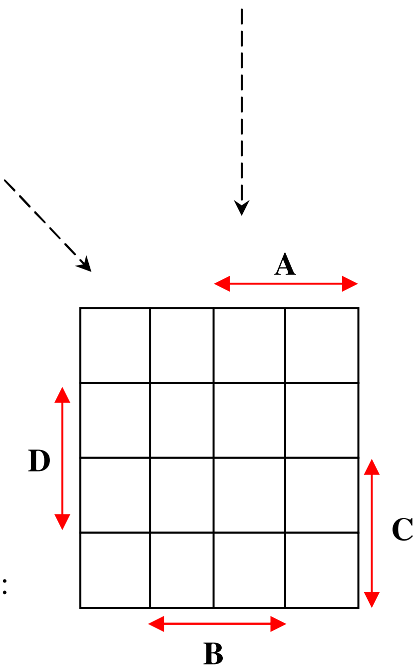
- Let's work through this in detail … just this once!

- Okay, let's take the term ABC …



- Do the same for the terms $\overline{B}D$ and $\overline{A}BCD$ to get:





- Hence, the final Karnaugh Map is:

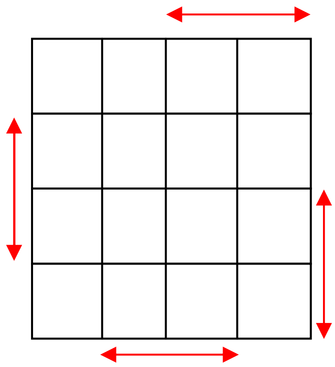
- ***Ex. 4.1 Obtain a Karnaugh Map representation for each of the following functions:***

  (i) $f_{(A,B,C,D)} = \Sigma(0, 1, 5, 15)$

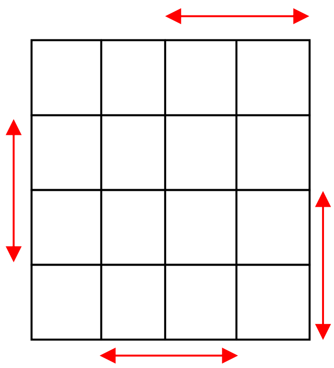
  (ii) $f_{(A,B,C,D)} = \bar{A}B\bar{D} + BC + A\bar{B}\bar{D}$

  (iii) $f_{(A,B,D)} = \bar{A}\bar{B}\bar{D} + BD + A\bar{B}\bar{D}$

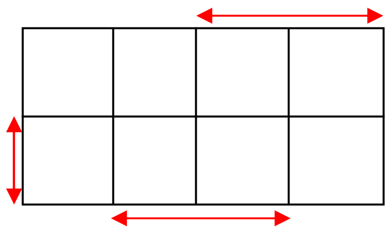
(i)



(ii)

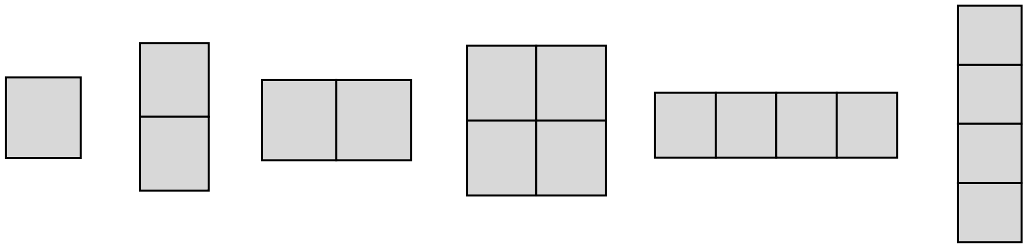


(iii)



4.5 Minimisation using Karnaugh Maps

- The '1' cells in a Karnaugh Map are grouped together into rectangular boxes.
- These must be **regular rectangles** and must **contain 2^n cells**. Permitted shapes include:



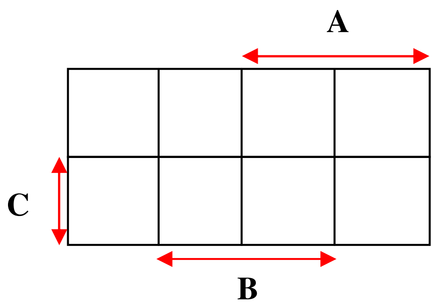
etc…

- Minimisation is achieved by enclosing all the 1's in the **minimum number of maximum size legitimate boxes**.
- **Boxes are permitted to overlap.** This is equivalent to using a minterm more than once.
- The best way to illustrate KM minimisation is through examples.

- ***Ex. 4.2 Minimise the following logic function:***

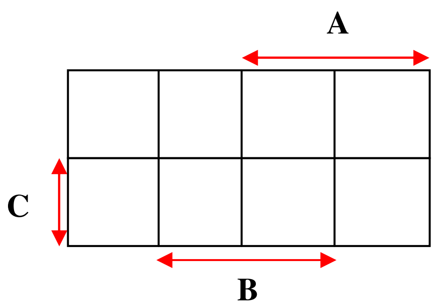
$$f_{(A,B,C)} = \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$$

(solved earlier in Ex 3.1 using Boolean algebra)



Hence:

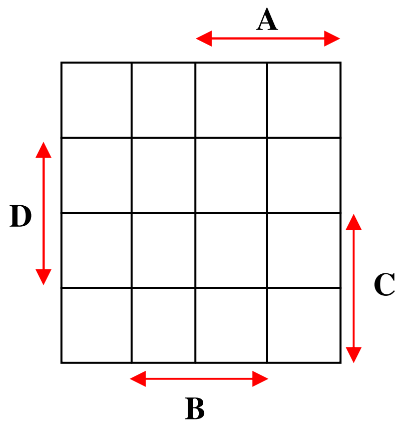
- ***Ex. 4.3 Minimise the logic function*** $f_{(A,B,C)} = \Sigma(0, 1, 2, 3, 4, 5)$



Hence:

- ***Ex. 4.4 Minimise the function given by:***

$$f_{(A,B,C,D)} = \sum(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$$



Hence:

- A simple check can be carried out for a 4-variable function, when obtaining the expression for a particular grouping:

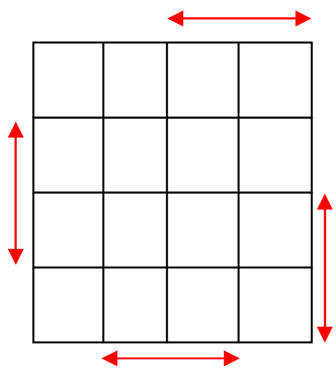
  8 squares = 1 variable (e.g. $\overline{A}$)
  4 squares = 2 variables (e.g. $\overline{B}\overline{C}$)
  2 squares = 3 variables
  1 square = 4 variables

- In the case of a 3-variable function:

  4 squares = 1 variable
  2 squares = 2 variables
  1 square = 3 variables

- ***Ex. 4.5 Minimise the following logic function:***

$$f_{(W,X,Y,Z)} = \overline{W}\overline{X}Y + \overline{W}Z + W\overline{Y}Z + \overline{W}Y\overline{Z}$$



Hence:

4.5 'Don't Care' Conditions

- Sometimes a situation arises where a certain combination of the inputs are not allowed or cannot occur.

- For example, in the Binary Coded Decimal (BCD) system which works for 0 to 9, the binary equivalent of the decimal values 10 to 15 are invalid or meaningless codes.

- In such cases, we design our circuit not caring if these conditions happen or not. In other words, we treat them as **don't care conditions** or don't care terms.

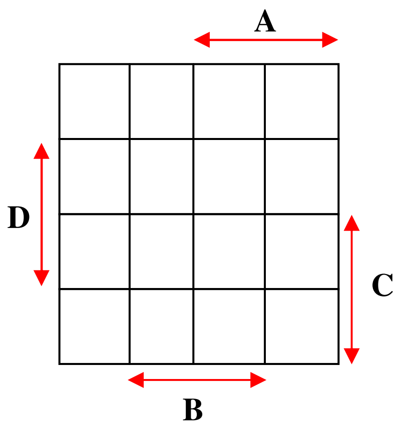
- We donate these terms with an 'X', as they can be treated as either a '1' or a '0' since they don't actually matter.

- We can then use such terms to our advantage in KM minimisation by treating them as either 1's or 0's to form larger groupings and, hence, more simplified functions.

- Don't care conditions are typically expressed using the notation d **(…)**, as illustrated in the next example.

- ***Ex. 4.6 Minimise the function defined as:***

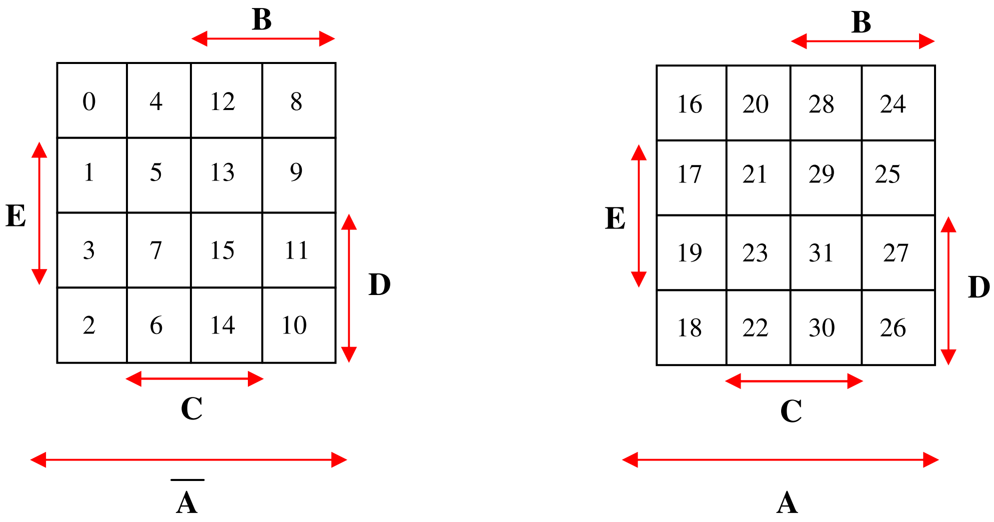
$$f_{(A,B,C,D)} = \sum(1, 3, 5, 7, 9) + \sum d(6, 12, 13)$$



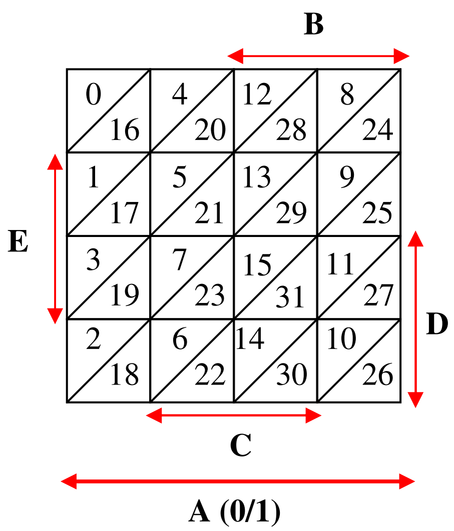
Hence:

4.6 Minimisation using 5-variable KMs

- A 5-variable KM consists of **two overlapping 4-variable KMs** as follows:



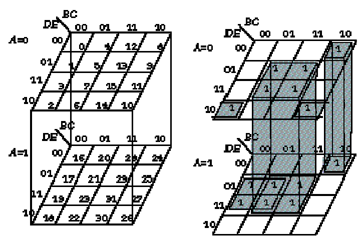
- An alternative representation is shown below:



- Remember to take the largest possible groupings across **both** maps.

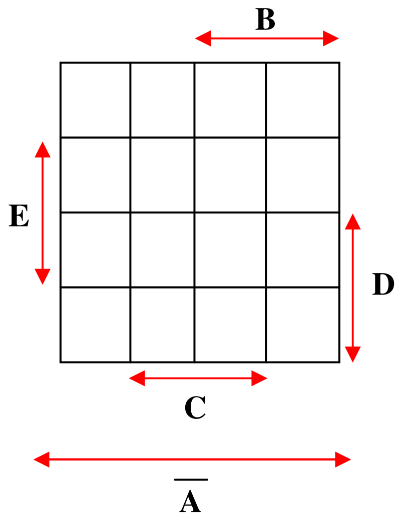
- As before, a simple check can be carried out for a 5-variable function, when obtaining the expression for a particular grouping:

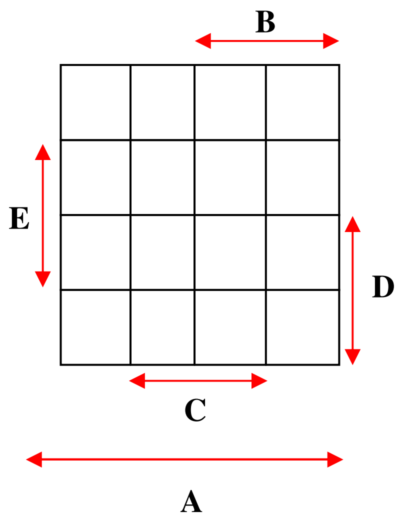
16 squares = 1 variable
8 squares = 2 variable
4 squares = 3 variables
2 squares = 4 variables
1 square = 5 variables



- *Ex. 4.7 Minimise the following algebraic expression:*

$$f_{(A,B,C,D,E)} = BDE + \bar{B}\bar{C}D + CDE + \bar{A}\bar{B}CE + \bar{A}\bar{B}C + \bar{B}\bar{C}\bar{D}\bar{E} + ABC\bar{D}\bar{E}$$

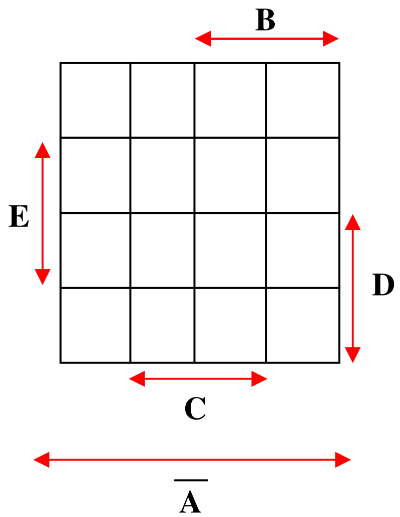


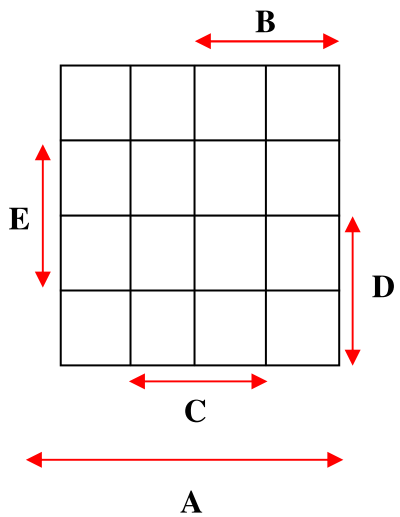


Hence:

- *Ex. 4.8 Minimise the following algebraic expression:*

$$f_{(A,B,C,D,E)} = \sum(2, 3, 4, 5, 6, 7, 11, 12, 13, 15, 18, 19, 21, 23, 26, 27, 28, 29, 30, 31)$$
$$+ \sum d(10, 14, 20, 22)$$





Hence: