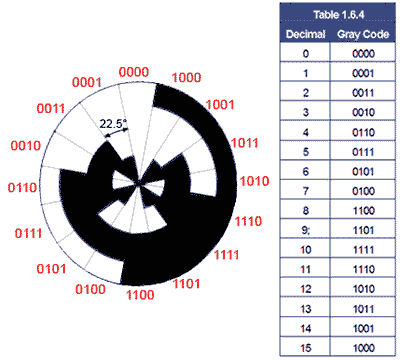
**KARNAUGH MAPS**

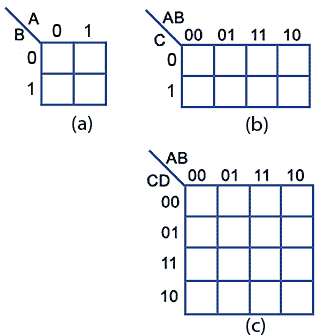
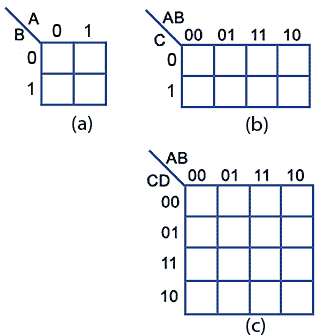
# Why Karnaugh Maps?

Karnaugh Maps offer a graphical method of **reducing a digital circuit to its minimum number of gates** by graphical means rather than by algebra.

Karnaugh maps can be used on small circuits having two or three inputs. On more complex circuits having up to 6 inputs, it can provide **quicker** and **simpler** **minimisation** **than Boolean algebra**.

# Constructing Karnaugh Maps

The shape and size of the map is dependent on the number of binary inputs. The map needs one cell for each possible binary word applied to the inputs.



*Fig a*: 2 input circuits with inputs A and B require maps with 22 = 4 cells

*Fig b*: 3 input circuits with inputs A B and C require maps with 23 = 8 cells

*Fib c*: 4 input circuits with inputs A B C and D require maps with 24 = 16 cells

**Note**: The way to number the top and left map’s edge does not follow the normal binary counting sequence like the Truth Table, but uses a **Gray Code**. Get the sequence wrong and the map will not work!

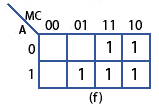
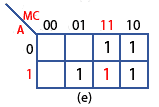
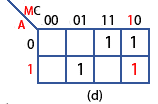
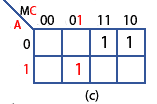
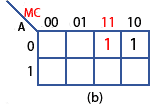
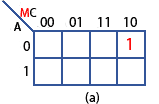
# Example of Using the Karnaugh Map

The figure beside is the truth table for the **3 inputs** (A, M, C) and **1 output** (X).

This table will serve to show the process of transferring the data into the cells of the Karnaugh map.

**Step 1: Create a K-Map**

The process is shown step by step in the figures below (from a to f).

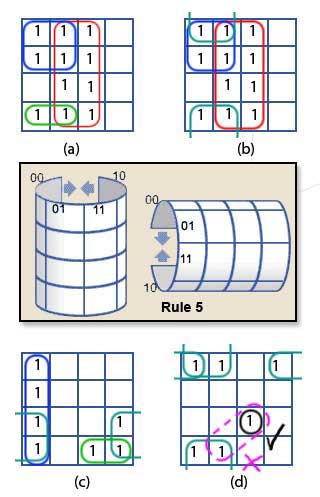


We can create a K-map based on **Minterm** (*Sum of Product*) or **Maxterm** (*Product of Sum*). If we use Minterm, **values of 1** are put into the map, and inputs’ relationship is **multiplication** while groups’ relationship is **addition**. On the other hand, if we use Maxterm, **values of 0** are put into the map, inputs’ relationship is addition and groups’ relationship is multiplication.

# Simplifying Karnaugh Maps

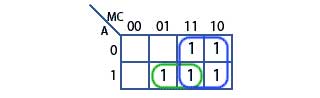
Circuit simplification in any Karnaugh map is achieved by combining the cells containing 1 to make groups of cells. In grouping the cells it is necessary to follow **six rules**:

* If using Minterm, groups should contain **as many ‘1’ cells as possible** and **no ‘0’ cells**, and vice versa for Maxterm.
* Groups can **only** contain 1, 2, 4, 8, 16 or 32... cells (**powers of 2**).
* A ‘1’ cell can only be grouped with **adjacent** ‘1’ cells that are immediately above, below, left or right of that cell; **no diagonal grouping**. Similar condition for ‘0’ cells.
* Groups of ‘1’ cells (or ‘0’ cells) **should overlap**. This helps make smaller groups as large as possible, which is an advantage in finding the simplest solution.
* The top/bottom and left/right **edges** of the map are considered to be **continuous**, so larger groups can be made by grouping cells across the top and bottom or left and right edges of the map.
* There should be **as few groups as possible**.



Map (a) follows rules 2, 3 and 4 and shows three groups containing 8, 4 and 2 cells. This will simplify the circuit being produced, but it is not optimum.

Map (b) shows an improvement, still with 3 groups but they now contain 8, 4 and 4 cells. This map takes advantage of rule 5 by joining the 2 cells ringed in green in Map (a) with the top 2 cells in the blue group. Thus, this way forms a group of 4 instead of 2. The map now conforms to all 6 rules.

***Back to our example***

**Step 2: Group the K-Map cells**

Here is the result with just 2 groups

**Step 3: Convert the K-Map to a simple Boolean equation**

Converting the two groups in the Karnaugh map to Boolean expressions is done by **maintaining which input (A, M or C) does NOT change within each group.**

Firstly, taking the blue group. It spans two rows vertically, so it contains rows A=0 and A=1, which means A changes within the group, so it cannot appear in the expression. The blue group also spans two columns (MC=11 and MC=10), in which the only input that does not change in the blue group is M, so the Boolean expression for the blue group is simply M.

Secondly, looking at the green group. A does not change but MC changes from 01 to 11 (M changes, C does not). Therefore, there are two non-changing inputs in this group A and C.

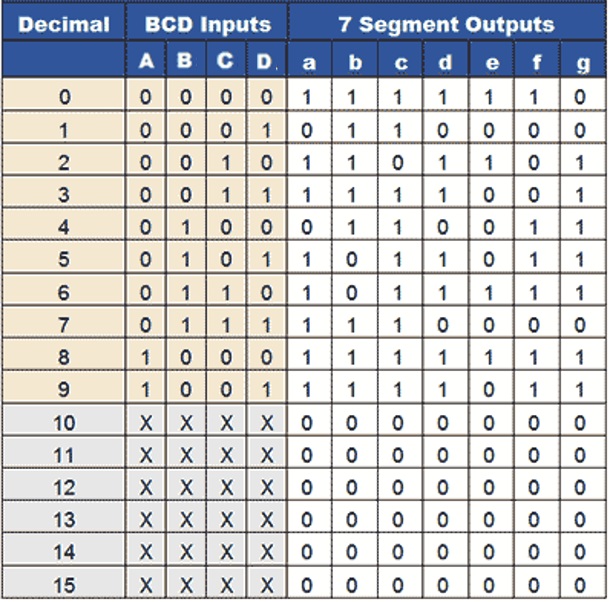
Putting the results of the simplification together by ‘**ANDing’ any non-changing inputs within a single group,** and **‘ORing’ the different groups**, produces the simplified Boolean equation for the whole circuit:

**X = M + A•C**

# One More Example: Designing the BCD to 7 Segment Decoder

The purpose of this circuit is to illuminate the LEDs (or activate the LCD segments) that make up typical numerical displays.

The table below shows an example of a truth table for a BCD to 7 segment decoder. In columns a to g, an output of logic 1 lights one particular segment of the display. Logic 0 turns it off. An X output is called a ‘Don’t Care’ as it does not matter what the possible binary value would be in the BCD inputs.



Nên thay các giá trị 0 này thành x

**Step 1 + 2 + 3**: Create a K-Map, group its cells and convert to Boolean equations

Không bắt buộc phải nhóm x, nhưng nếu có x thì nên nhóm để equations đơn giản nhất

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AB \ CD** | **00** | **01** | **11** | **10** |
| **00** | 1 | 0 | 1 | 1 |
| **01** | 0 | 1 | 1 | 1 |
| **11** | x | x | x | x |
| **10** | 1 | 1 | x | x |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AB \ CD** | **00** | **01** | **11** | **10** |
| **00** | 1 | 1 | 1 | 1 |
| **01** | 1 | 0 | 1 | 0 |
| **11** | x | x | x | x |
| **10** | 1 | 1 | x | x |

**a = A + C + BD + b = + + CD**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AB \ CD** | **00** | **01** | **11** | **10** |
| **00** | 1 | 0 | 1 | 1 |
| **01** | 0 | 1 | 0 | 1 |
| **11** | x | x | x | x |
| **10** | 1 | 1 | x | x |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AB \ CD** | **00** | **01** | **11** | **10** |
| **00** | 1 | 1 | 1 | 0 |
| **01** | 1 | 1 | 1 | 1 |
| **11** | x | x | x | x |
| **10** | 1 | 1 | x | x |

**c = B + + D d = + C + BD + C + A**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AB \ CD** | **00** | **01** | **11** | **10** |
| **00** | 1 | 0 | 0 | 1 |
| **01** | 0 | 0 | 0 | 1 |
| **11** | x | x | x | x |
| **10** | 1 | 0 | x | x |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AB \ CD** | **00** | **01** | **11** | **10** |
| **00** | 1 | 0 | 0 | 0 |
| **01** | 1 | 1 | 0 | 1 |
| **11** | x | x | x | x |
| **10** | 1 | 1 | x | x |

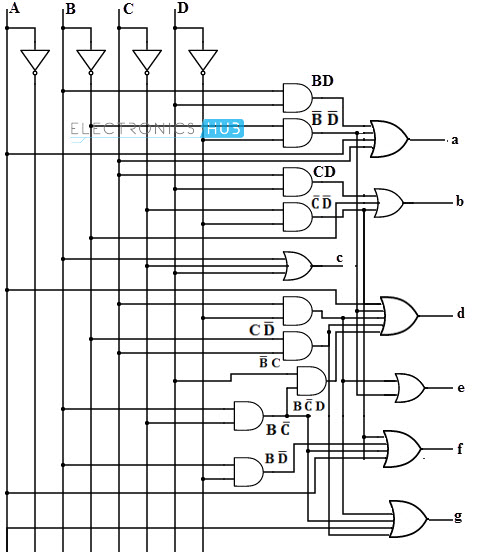
**e = + C f = A + + B + B**

Ví dụ này t sử dụng Minterm, tức nhóm các giá trị 1 vì ta thấy 1 xuất hiện rất nhiều. Nếu không thích, t có thể nhóm các giá trị 0 (maxterm)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AB \ CD** | **00** | **01** | **11** | **10** |
| **00** | 0 | 0 | 1 | 1 |
| **01** | 1 | 1 | 0 | 1 |
| **11** | x | x | x | x |
| **10** | 1 | 1 | x | x |

**g = C + C + B + A**

**Step 4:** Draw the circuit



References:

<http://www.learnabout-electronics.org/Digital/dig24.php>

<http://www.electronicshub.org/bcd-7-segment-led-display-decoder-circuit/>

<https://www.youtube.com/watch?v=KGDRddDQhFI&nohtml5=False> 🡪 Excellent tutorial video

<http://www.32x8.com> **🡪 Online Automatical K-Map Tool 🡪 Very optimum**