

KARMA: A DIDACTIC TOOL FOR KARNAUGH MAPS

Carlos E. Klock, Felipe R. Schneider, Mateus V. N. Gomes, Renato R. Ribas, André I. Reis

Nangate Research Lab – Instituto de Informática, UFRGS

Av. Bento Gonçalves, 9500 – CEP 91501-970, Porto Alegre, Brazil

{ceklock, felipers, mvngomes, rpribas, andreis}@inf.ufrgs.br

ABSTRACT

This paper presents a didactic tool for logic synthesis called Karma (KARnaugh MAPs). The main objective of this tool is to help users to have a better understanding of Karnaugh maps, truth-tables, Boolean functions and many concepts of logic synthesis. Through a friendly graphical interface the user can interact with the software in many ways like assigning a Boolean function and running Quine-McCluskey algorithm to get the prime implicants and see how they are positioned in the Karnaugh map.

1. INTRODUCTION

Karnaugh Maps [1][2] are useful structures to minimize Boolean functions. Although not being the best way, computationally, to minimize Boolean functions, these structures are very important to understand how adjacent terms of equal value, when grouped, can reduce the number of minterms that comprehend the optimized final solution. Karnaugh maps are widely used in computer science, in areas that include logic, binary arithmetic and circuit synthesis. This way, it is possible to say that Karnaugh maps are part of the basis of computer science.

Karma is a software that was made to help in the process of learning Karnaugh maps and the concepts that involve it. It has many interesting features to accomplish this objective, divided in two modes of operation: normal mode and teaching mode (that will be explained below). Some of the features included in Karma are the visualization of prime implicants in the map, step-by-step solution of the Quine-McCluskey algorithm, change of the position of the variables in the Karnaugh map and a teaching mode with options to help the users to learn concepts about Karnaugh maps and logic synthesis.

Karma was developed in the Java language as part of the Lagarto (LAYOUT GenerAtOR TOol) project and is available for download at its home page [3].

2. NORMAL MODE

This is the mode that is opened when the software starts. The first thing that the user will notice in this mode is the truth-table and the Karnaugh map that are displayed in the window (see Fig.1). The user has freedom to change the values of the minterms of the Karnaugh map and truth-table and it is possible to see that they are synchronized: when a value is changed in the map it is changed in the truth-table too.

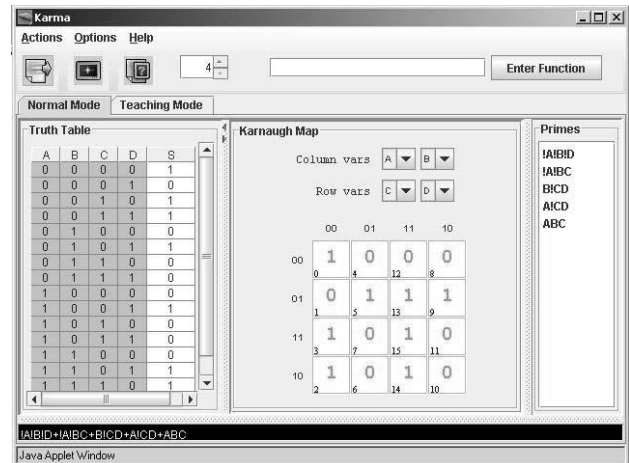


Figure 1: first window when software starts (normal mode)

The user is able to enter a function of at least two and maximum eight variables. The function can be entered in three different ways: Karnaugh map, truth-table or the function itself. After the function is defined it is possible to run the Quine-McCluskey algorithm [4] and check the steps in a window that shows a log of the procedures (see Fig.2) that were done by the algorithm. After running the algorithm a window is displayed to show the prime implicants that were found. The user can click on the prime implicants to see them highlighted in the K-map.

The cells of the Karnaugh map and truth-table display the logic values zero, one and don't-care. Each time the user clicks over a cell its value is changed cyclically.

It is possible to change the position of the variables in the Karnaugh map. This means that the cells of the map will change their position depending on the position that was selected for the variables.

3. TEACHING MODE

This mode was developed for users to review basic concepts of logic synthesis through exercise options. The available exercises cover the learning of adjacency of minterms, cubes, prime implicants, function cover and Quine-McCluskey's cover chart.

The exercises contain a brief description of the proposed problem, and a Karnaugh map to help finding the answer in a visual way. The given solution is evaluated to check its correctness. If it is not correct, the correct solution is displayed.

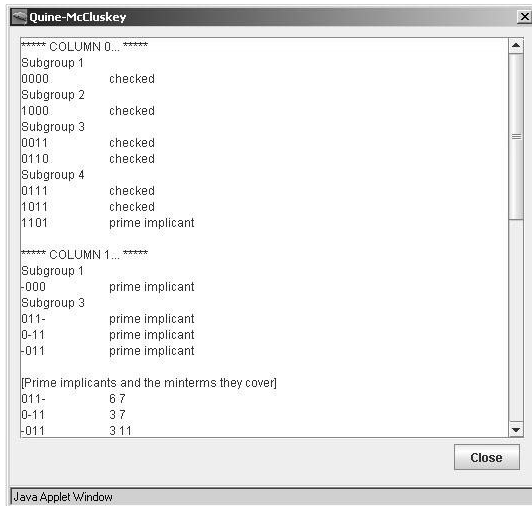


Figure 2: Log showing steps of the Quine-McCluskey algorithm

For the exercises the window is divided in two parts, where the left part shows the description of the exercise and the right part shows the Karnaugh map.

The available exercises were implemented because they are the most common and basic exercises about Karnaugh maps. A description of each exercise group is given below.

3.1. Minterms

Two options are available to cover the concept of minterm. In “*view adjacent minterms*” the user is proposed to select a minterm in the Karnaugh map, and the adjacent minterms are highlighted. In “*find adjacent minterms*”, a minterm is selected and the user is asked to point its adjacent minterms. If a correct minterm is selected, it is displayed in green color, otherwise the minterm is displayed in red color.

3.2. Groups

In the “*what group is*” exercise, the user is asked to select minterms in the map that form a group. When the selection is finished, a message displays what cube was formed.

3.3 Cubes

Three exercises are proposed to review the concept of a cube. In “*find cube*”, the user is given a cube and is asked to find the minterms that form it in the function displayed on the map. In “*select cube*” the user must select a cube of the function displayed. Information on the cube type (implicant, prime implicant or not a cube) is given. This information must be entered by the user in the “*cube is prime*” exercise.

3.4. Function cover

In this exercise a list of cubes is displayed and the user

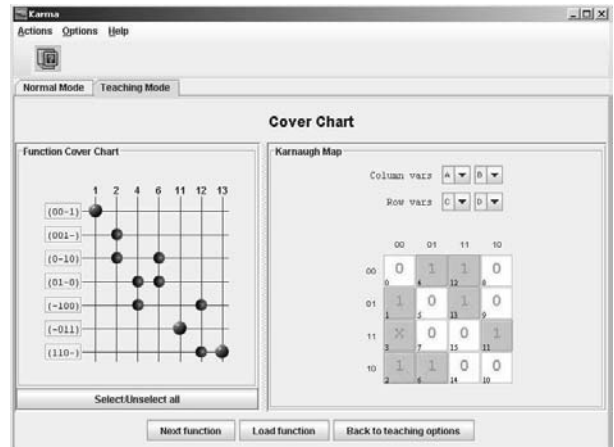


Figure 3: the final cover of a function in the teaching mode

must select, for each cube, the correct answer about its state inside the function. There are hints about the type of function cover, according to the given answers.

3.5. Cover chart

This is the most illustrative of the exercises. It shows the cover chart of a function, and the user can interact with the cover chart to find the essential prime implicants by selecting cubes in the chart and observing what minterms they cover in the map (see Fig.3). When a selected cube is essential prime implicant, a red ball is displayed in the chart at the same row, otherwise just blue balls are displayed. This exercise is interesting because the user can try to find the minimum cover of a function.

4. CONCLUSION AND FUTURE WORK

The acquisition of more advanced knowledge about logic synthesis depends on a solid knowledge about the basic concepts. The tool that was presented in this paper is a good instrument to learn the basis, offering the possibility to understand important concepts of logic functions, truth-tables, Karnaugh maps, cubes, and other logic synthesis subjects.

Some features not yet implemented would make the use of Karma an even more interesting experience. These features include the possibility of renaming variables, and the input of a logic function through a BLIF file.

5. REFERENCES

- [1] Wagner, F. R., A. I. Reis, R. P. Ribas, *Fundamentos de Circuitos Digitais*, Sagra Luzzatto, Porto Alegre, 2006.
- [2] Katz, R. H., *Contemporary Logic Design*, Benjamin/Cummings Publishing Company, United States, 1995.
- [3] *LAGARTO Project*, <http://www.inf.ufrgs.br/lagarto>
- [4] Gerez, S. H., *Algorithms for VLSI Design Automation*, John Wiley & Sons, England, 1999.