S E N S ' 2 0 0 6
Second Scientific Conference with International Participation
SPACE, ECOLOGY, NANOTECHNOLOGY, SAFETY
14 – 16 June 2006, Varna, Bulgaria

MULTIPLE-OUTPUT COMBINATIONAL SCHEMES DESIGN WEB-BASED TUTORING TOOL

Vladimir Mateev

Angel Kunchev University of Rousse
8, Studentska St., Rousse 7017, Bulgaria
e-mail: vmateev@ru.acad.bg

Keywords: ***Multiple-Output Networks, Boolean Function Minimization, Veitch Map, Digital Logic Design, e-learning.***

Abstract: *Web – based tutoring tool for multiple-output combinational networks design that has graphical user interface is presented in this paper. The tool allows the user to define his/her own task on multi-output combinational circuit design and leads him/her to the right decision. Task's definition involves functions and arguments number, their names and truth tables of the functions. The tool finds the decision and gives facilities to the user to express its own decision involving minimization of the functions, expressing them analytically and drawing the scheme. On each step of user's performance the user receives appropriate messages if there is a mistake in the decision. The tool is designed to improve students' skills in multiple-output schemes design giving them a possibility to practice as much as they want with tasks defined by them. Java programming language has been used. That allows to create the tool as an applet and to incorporate it in a Web document or to use it as a standalone application as well. The tool can be very useful for self-learning, in e-learning, virtual universities and distance education.*

INTRODUCTION

A lot of tasks involved in "Analysis and synthesis of logical schemes" subject for students of "Computer and Communication Systems and Technologies" and "Electronics" require the students to be able to design multiple-output combinational schemes. The students must be able to decide this problem like an isolated one or like a part of a task on synthesis of different logical devices that are both combinational and sequential circuits. This task is very important and it is a good idea a corresponding tool to be created which will be able to help students in their efforts to decide tasks like that. And it is better if this tool is Web located so it can be used not only in the laboratory but at any other place connected with the Internet, as well. So it can be used in e-learning, distance education and as a tutoring tool in virtual universities.

A lot of applets and pages about Boolean functions minimization and digital logic design can be found in Internet. Many of them are very interesting and useful. Some of them are applets for minimization by Karnaugh maps [6,7], some demonstrate Quine McCluskey method [8, 9] and present all minimal forms (both conjunctive and disjunctive minimal forms or just one of them). Some of them even draw the corresponding minimal scheme [6]. But the user can just see the simplified form of the function and the adjacent minterms while changing functions values or its analytical presentation but he cannot determine by himself the implicants in the function and to write corresponding minimal forms of the function. There are simple and very interesting tools for logical scheme design

[10] or multi-output scheme design [11], or even more complicated environments for design and simulation like “Digital simulator” [12] and “Digital Electronics Education and Design Suite” [13] which is a standalone application. Using the tools for scheme design the user can draw different schemes, can experiment using them but there is no connection between the design scheme and the given-in-advance set of functions that must be realized as a multi-output scheme. In all examples mentioned by now and many more the student can find a lot of explanations and possibilities to minimize his own function and to design a scheme and simulate its operation. But places where he can find tasks for logical schemes design and the necessary possibilities to design the task by drawing the corresponding scheme [3, 14] are not enough. And it was impossible for the author of the paper to find not only a web-based but even a standalone application where the student can define his own digital design task by giving a number of his own Boolean functions, defined by their truth tables and where he will find all necessary facilities to design a minimal multi-output scheme that realize the functions. Multi-output scheme design is a task that is very useful for both combinational and sequential logic networks design and a very useful skill that students must have and improve by as many tasks as they need in order to be able to design and realize more complex digital logic devices. This article shows the possibility of such a tool.

DESIGN OF MULTI-OUTPUT SCHEMES

In large digital systems, it is frequently necessary to realize several switching functions of the same input variables, so the solution of digital design problem often requires realization of several functions of the same variables. Although each function could be realized separately, it is generally more economical to realize the functions using a single network. There are different techniques for doing this. One of them consists in using some gates in common between two or more functions [4].

The first step in the design of multi-output combinational switching network is usually to set up truth tables that specify the outputs as functions of the input variables. The next step is to derive simplified algebraic expressions for the output functions using Veitch or Karnaught maps, the Quine-McCluskey procedure, or a similar method. In some cases, particularly if the number of variables is large and the number of terms is small, it may be desirable to go directly from the problem statement to algebraic equations, without writing down the truth table. The resulting equations can then be simplified algebraically. The simplified algebraic expressions are then manipulated into the proper form depending on the type of gates to be used in realizing the network.

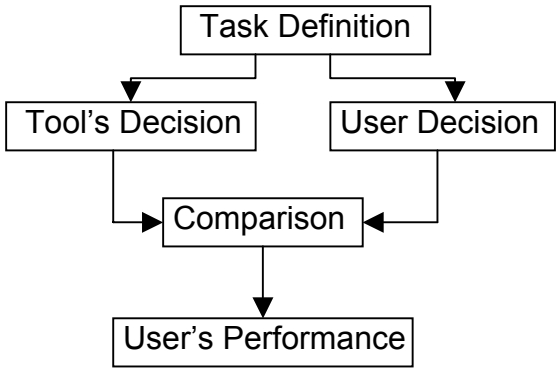


Fig.1. Architecture of the tool

For a two level network, Veitch maps of the output functions can be used to minimize them and to find common terms in order to reduce the total number of gates in the scheme.

TOOL’S ARCHITECTURE AND ITS REALISATION

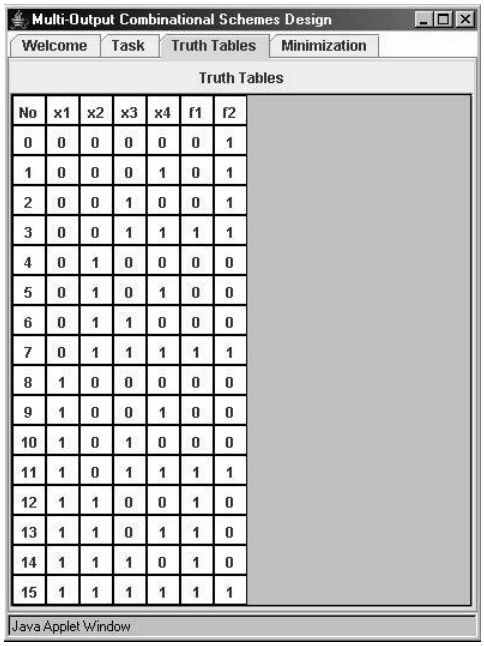
Java programming language and object oriented programming have been used to realize the tool [2, 5], which is an applet that can be started using a button, realized also like an applet involved in a Web page. The StartMultiOutputSchDesApplet class of multioutputschdesapplet package is an applet that present a button incorporated in a Web page. By pressing the button the user can start the tool. The main Graphic User Interface class is MultiOutputSchDesFrame from a multioutputschdes package.

A simplified architecture of the tool is presented in Fig. 1.

When the task is completely defined by the user the class TaskDecision starts a set of consecutive operations for finding the decision. A series of intermediate result are kept in order to be compared with the user operations while the user decides the task. The tool's decision contains all minimal sum of products form for all functions like arrays of implicants and all unique implicants needed for multi-output scheme design. The tool use Quine-McClusky method to minimize the functions, while the user must apply Veitch maps for this purpose, a skill that the user must improve. At every stage of user performance the tool gives appropriate messages when a mistake occurs.

DESCRIPTION OF THE TOOL AND ITS USAGE

In order to start the tool the user must have Java Runtime Environment (JRE) installed on its computer. If there is not JRE installed, when the user chose the Web page he/she will be prompted to do it and will be lead to the Sun facility to download necessary software. The tutoring tool is realized as an applet that can be invoked by pressing the button "Start Multi-Output Combinational Schemes Design Applet" involved in a Web page which URL is http://www.geocities.com/vmateev777/multi. In its mane frame there is a Swing component JTabbedPane. It gives a possibility to the user not only to go step by step forward to the final decision but to go back to a desire point in order to use a sort of information that has already been defined but that is necessary for some reason again. When the user starts the application just a "Welcome" page can be seen. There are some explanations to the user on it about the purpose of the tool. There is "Next" button on the page and after pressing it the user can move forward. Usually going forward is possible by pressing the "Next" button situated on the current page, which can be seen all the time or which appears immediately after a curtain operation has been accomplished by the user. The task's decision is separated in several stages. The stages which will be necessary again in a later time are panes in a JTabbedPane. The user can always see them again if he wants or he needs them. But he can not edit the information already involved in them. The page after "Welcome page" is a place where the user determines the numbers of functions and its arguments. There are two JTextFields in the page, where the user can involve just some numbers. Typing forbidden symbols is not allowed and a warning message is displayed. If the involved information is correct it become possible for the user to move to the next page. On it the user gives names for functions and their arguments. The names must be unique and every name must be a string of small Latin letters and digits where the first sign must be a letter. If some of these requirements are not kept the involved information is rejected and a warning message appears. When the names are typed correctly it becomes possible to go to the next page. The label of next page is "Fill Truth Table" and it is a place where the user defines truth tables of the functions. It happens by mouse click in the corresponding cell of the table and the value of the function changes alternatively. Filling truth tables is the last step in task definition. Truth tables of an example of two functions can be seen in Fig.2. What follows is minimization of the functions, its analytical presentation and scheme drawing.



No	x1	x2	x3	x4	f1	f2
0	0	0	0	0	0	1
1	0	0	0	1	0	1
2	0	0	1	0	0	1
3	0	0	1	1	1	1
4	0	1	0	0	0	0
5	0	1	0	1	0	0
6	0	1	1	0	0	0
7	0	1	1	1	1	1
8	1	0	0	0	0	0
9	1	0	0	1	0	0
10	1	0	1	0	0	0
11	1	0	1	1	1	1
12	1	1	0	0	1	0
13	1	1	0	1	1	0
14	1	1	1	0	1	0
15	1	1	1	1	1	1

Fig.2. Truth tables

In the tool presented Veitch maps are used for functions minimization. This method for minimization is chosen because it is involved in the syllabus in our university. The next stage in task decision is filling the Veitch maps. The user can start with every function. It is possible for the user to fill all the maps simultaneously as well. If the map is filled incorrectly, a warning message appears and the filling must continue. When the map is filled correctly the application gave the possibility for the user to go to the stage where to minimize the function, presented in the corresponding map. In a pane labeled “Minimization” which is for filling the maps and minimization of the functions using these maps, functions are presented by their names and after pressing the “Minimize function” button, situated next to the function name a frame for minimization appear. It can be seen in a process of minimization of a function in Fig.3. In a left upper site of the frame a Veitch map filled with functions values is situated. The user can not edit functions values in the map. Just like a truth table, the Veitch map of function specifies the value of the function for every combination of values of its arguments. On the figure a 4-variable Veitch map is shown.

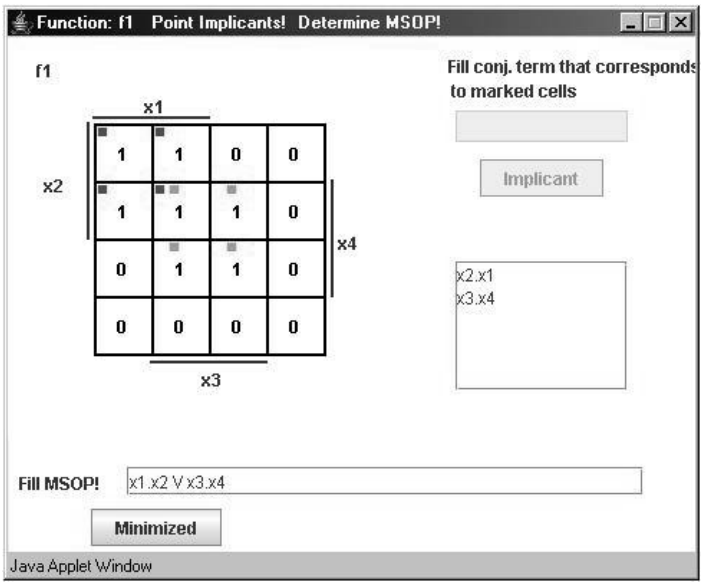


Fig.3. Function minimization

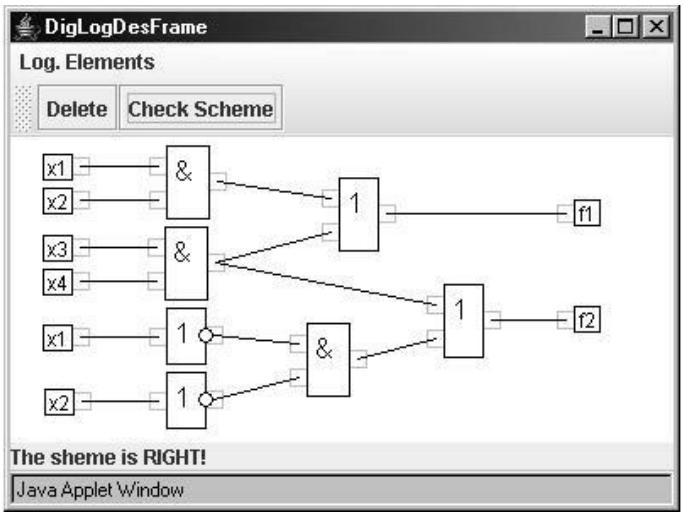


Fig.4. The drawing scheme frame

What the user must do is to combine adjacent groups of 1’s. Usually there are several groups. In all joined cells the function value must be 1 and the cells must be situated in a certain manner and their number must be 2 at the power of a whole number or 1. Every cell containing 1 must be joined to at least one group. Every group must contain as much as possible cells and the number of all groups must be as small as possible. To every joined cells corresponds an analytical expression. The user must mark by mouse click a group of adjacent cells. The function value in marked cell becomes red. Then the user must write in “Fill conj. Term that corresponds to marked cells” JTextField the analytical presentation of corresponding implicant and press “Implicant” button. If there are mistakes the user will see a warning message. Else the implicant will be written in the TextArea below. In this way the user must find all implicants needed for at least one form of the Minimal Sum Of Products (MSOP) form of the function. When all implicants are in the list the user must press the button “All”. It is situated under the text area containing implicants x2.x1 and x3.x4 and is invisible in fig. 3 because the figure shows the next stage of task decision. If there are no mistakes, the

button “All” disappears and a message to the user to write the corresponding MSOP of the function appears. The user can not mark sells and write implicants any more. He can only type the MSOP in the TextField next to the “Fill MSOP!” label. If the written form is correct this frame disappears and in a TabbedPane labeled “Minimization” the analytical MSOP form of the minimized function can be seen. In the same way the user must minimize all the functions and when it all is done correctly a button “Design Scheme” in “Functions” pane appears. By pressing it a frame “DigLogDesFrame” with facilities to draw the scheme can be used. The frame “DigLogDesFrame” has menu bar, tool bar, a large design area in the middle and status bar that appears at the bottom of the frame when it is necessary. The frame with a drawn scheme can be seen in Fig. 3. There are “Log. elements” menu, where the user finds necessary logical elements, arguments and function boxes to draw the scheme. The frame is very much like in [3] where can be found more explanations about its possibilities. The user can choose the elements for the scheme in the commands of “Log. Elements” menu. There are two submenus in it. The first one is for functions arguments. There are as much arguments as the user has defined in task definition stage and their names are names, given to them by the user. The user can use as much arguments as he wants in the scheme. The second submenu is for function boxes that present the functions in the scheme. The user can have just one box for every function. The elements have connecting areas for their inputs and outputs. Creation of a connection can be made by dragging the mouse from the output of one component to the desire input of the other. The user can move an element by dragging it. If there is a necessity of deleting an element or a connection it must be marked by clicking on it by the mouse and pressing the Delete button. The rightness of the scheme can be established by pressing the “Check scheme” button. Then an appropriate message appears in the status bar.

CONCLUSIONS

A two-level multiple-output combinational networks design web-based tutoring tool with a graphical user interface has been presented in this article. It shows the possibility of existence of a tool where the user can define a task on multi-output combinational circuit design determining a number of Boolean functions with their truth tables. The tool present facilities for the user to fulfill the defined task and gives a lot of messages about the user’s achievements. In the future the tool will be enriched with a possibility for the user to define Boolean functions with no cares values, to find not only MSOP but MPOS as well and the tool will make a model of a user and will give him not only an estimation of user’s skills but will propose him a number of task if the user needs them in order to be better in the area of multi-output combinational networks design.

REFERENCES

[1] Angel Smrikarov, "E-learning", Ruse, 2004
[2] Bruce Eckel: "Thinking in Java", 2000
[3] Mateev, V., P. Manoilov, M. Iliev, “Tutoring tool for Logical Schemes Design”, Proceedings of the International Conference on Computer Systems and Technologies, Rousse, Bulgaria, 17-18 June, 2004, pp. 1.9-1 - 1.9-5
[4] Roth, Charles. H.: "Fundamentals of logic design", West Publishing Company, 1992
[5] Simon Roberts, Philip Heller, Michael Ernest: "Complete Java 2 Certification", 2000
[6] http://kauai.theoinf.tu-ilmenau.de/~sane/projekte/karnaugh/embed_karnaugh.html
[7] http://members.cox.net/cyclone1980/KMapSimulation10Embedded.htm#KMap
[8] http://www.mathcs.bethel.edu/~gossett/DiscreteMathWithProof/QuineMcCluskey
[9] http://my.dreamwiz.com/jeehk/works/qm/index.html
[10] http://www.course.com/downloads/computerscience/aeonline/7/2/
[11] http://www.jhu.edu/~virtlab/logic/logic.htm
[12] http://www.eelab.usyd.edu.au/digital_tutorial/
[13] http://netpro.evtek.fi/team/wp6/papers/eunite-13567_P_PONTA.pdf
[14] http://www.geocities.com/vmateev777/logschenv