International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 2, Issue 4, April 2014)

A Graphic Constructor for Logic Circuits Design

Hristo Kiskinov[1], Vilislav Radev[2], Maya Stoeva[3]

[1]Chief Assistant, [2]Assistant, at Faculty of Mathematics, Informatics and Information technology, Plovdiv University "Paisii Hilendarski", 4003 Plovdiv, 236 "Bulgaria" blvd., Bulgaria
[3] Like a part of PhD candidate- Assistant at Faculty of Mathematics, Informatics and Information technology, Plovdiv University "Paisii Hilendarski", 4003 Plovdiv, 236 "Bulgaria" blvd., Bulgaria

***Abstract* — The aim of this paper is the created by the authors graphic LC constructor – a designer tool for logic circuits. This constructor is a software system that enables graphical or tabular setting, drawing and calculation of logic circuits. Its user friendly interface has been built in compliance with the modern requirements and technology for design, allowing versatile use in teaching discrete mathematics.**

***Keywords*—Discrete mathematics, logic circuits, Boolean functions, graphic user interface, information technology, desktop software, UX design, responsible design, generalized model, flat design**

I. Introduction

In this article the authors make a view to their constructor of logic circuits - Logical Circuits (LC) for Boolean functions representation. It is created for educational purposes and is intended to help the study of the Boolean functions theory in lecture courses in discrete mathematics at universities and in specialized training in informatics in secondary schools. The clear and simple user interface vision also helps for that purpose. It is realized by following the basic principles in the design and building of interfaces [16].

On the one hand, together with the tables and formulas, the logic circuits are one of the most popular methods for representation of Boolean functions. On the other - they are mathematical models of actual electronic circuits. For that reason, the typical engineering pragmatism has led to stereotypes not typical to mathematical worldview. For example, in all considered by the authors graphic systems for creating logic circuits [3, 8, 11, 12, 13, 17, 19], there always exists a fixed set of basic logic gates, among which invariably participate AND, OR and NOT. The LC constructor has no fixed basic logic gates. All of the logic gates, without exception are created either tabular or by the help of logic circuit. Already built logic gates are used (if there are no restrictions with pedagogical purposes) by the creation of a logic circuit, which, if it is correct, in turn becomes a logic gate.

The LC constructor gives an automated drawing and interactive control of the process of designing the circuit, which not only ensures its correctness, but also helps for its creation.

By the software implementation of the constructor LC is taken into account the newest international standard for life-cycle processes through which a software passes in the development time – ISO/IEC 12207:2008 [9], which purpose is to define all necessary tasks related to the implementation and maintenance of the software, following certain patterns. The used model here is the generalized model for Interactive multimedia tools development [14]. Additional information about the LC constructor can be found on the project site: www.lc.myplovdiv.com [21].

II. Description Of The Functional Abilities Of The Graphic Constructor LC

The workspace for constructing logic circuits is a rectangular grid. Only one logic gate may be disposed in each cell of this grid. There exist a title bar, a menu bar with buttons and a toolbar with all available logic gates (see Fig. 1).

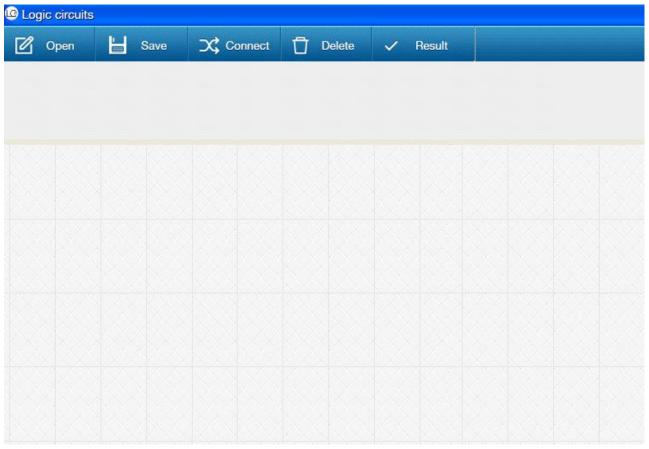


Fig. 1 - LC workspace –rectangular grid with a title bar, a menu bar with buttons and the bar with all available logic gates

When creating a logic circuit first you have to set the number of inputs (i.e. the number of the variables of the Boolean function) (see Fig. 2). On the first row of the grid automatically are drawn logic gates, which are representing the respective identities (see Fig. 4).

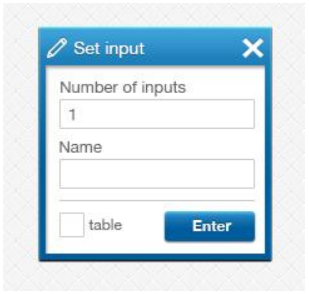
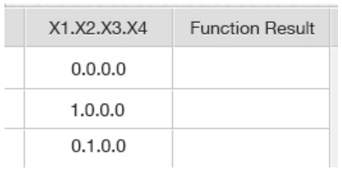

Set input
Number of inputs: 1
Name:
table
Enter

Fig. 2 - Setting the number of inputs, the logic circuit name and the designing type (tabular or graphical)

Further the name of the logic circuit is introduced, which will be also the name of the logic gate, defined with this logic circuit. After that follows the question how the logic circuit itself will be defined– with a table (Fig. 3) or graphically (Fig. 4).



X1.X2.X3.X4	Function Result
0.0.0.0	
1.0.0.0	
0.1.0.0	

Fig. 3 - Tabular definition

A. Designing logic circuit with a table

Defining with a table means that a logic gate from type Black box is created, i.e. such an element (logic gate), which realization is not important to us, but we will use it in the design of future logic circuits. This gate is drawn automatically on the second grid row, under the identity-gates, like a new logic gate with corresponding number of inputs and name and is connected with them. A window opens, in which the Boolean function, realized by this logic circuit can be defined with truth table (Fig. 3). After finishing with the whole table filling the logic circuit is saved as a new logic gate. It is recommended, but not mandatory, at least one full system of Boolean functions to be realized in this way, that will guarantee the possibility for graphical realization of any logic circuits in future.

For example, if we define with tables the logic gates AND, OR, NOT, which correspond to the full Boolean function system - conjunction, disjunction and negation, then we will obtain as a special case the well-known situation (shown on Fig. 4) of available basic gates.

B. Graphically designing of logic circuits

The main purpose of the LC constructor is graphical defining of a logic circuit (Fig. 4). This is done by dragging available items (gates) from the toolbar with the logic gates on the workspace grid. Since the first row is busy with the identities, the possible deployment on the grid is from the second row down. As is known, one of the possible errors in the construction of a logic circuit is the so called loop – as input of a gate to set a gate, which depends on one of its inputs from the output of the first gate. Such loop in a circuit, constructed with LC is impossible, because of the limitation every gate in a grid's row to have its own output, which connects only to gates, drawn below. This restriction not only makes impossible the existence of a loop in the circuit, but also improves its appearance and is very helpful for its construction. After dragging a gate from the toolbar to the grid, the constructor automatically positions it in the center of the selected grid cell. The user points the connection from the output of a gate to an input of another gate by mouse click. The system checks and does not allow connection to an already connected input. The drawing of this connection is made automatically by the LC constructor and ensures perspicuousness of the drawing, i.e. without overlapping parts of the connecting lines. This is one of the largest advantage of this constructor, because it allows the user to focus only on the logic of connecting gates, and not on their location and drawing. The constructor lets move on already placed element by automatically re-drawing its links, which are previously made. When the user is ready with his logic circuit, he can save it. The LC constructor checks the correctness of the circuit calculating the truth table of the realized Boolean function. Because the creation of loops is prevented at the stage of positioning and connecting elements, the only possible error, which can arise, is a missing connection to the gate's input. In that case, all unconnected inputs are marked and the process of graphically setting of the logic circuits continues.

If the logic circuit is correct, then it is saved and appears as a new available gate in the toolbar with the logic gates.

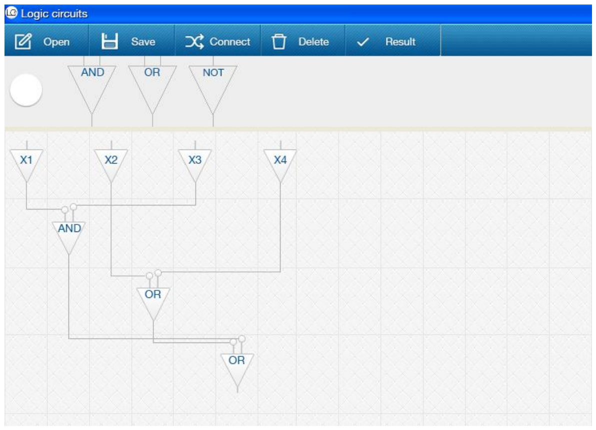


Fig. 4 - Sample logic circuit drawing

C. *Review of already created logic circuits*

At any time of working with the constructor it is possible in a new window detailed information about each of the logic gates, saved in the constructor, to be requested. This information includes the logic circuit, designed for this element and the table of the related Boolean function. The possibility in separate windows the logic circuits of the gates, involved in concrete logic circuit, to be visualized allows the creation and the use of “subcircuits”. These subcircuits can be embedded later in other circuits and so on. This avoids the creation of immense circuits containing a lot of gates.

D. *Resources for supporting the learning process (education tools)*

Because the main purpose of LC is for education, it has also resources, helping in learning process. The constructor offers two levels of access to its functional abilities – for teachers and for students. From teacher’s level it is possible to determine which of the logic gates in the toolbar will be accessible and which will not be. Such limiting access to available gates allows the presenting of many and varied tasks to be solved by the students. For example, one of the possible learning tasks can be constructing of new circuits only with a limited number of specific gates. Other additional possibility, given to the teacher, is the automatically checking of the circuit, constructed by the student – i.e. comparing the truth table of its corresponding Boolean functions with a target-table, set by the teacher.

III. Software Implementation

The LC software implementation is according to the international standard, responsible for the proper development of a software product - ISO/IEC 12207:2008 [9]. „Generalized model for Interactive multimedia tools development“ was chosen here [14]. It is nonlinear and that feature helps for easier interface design building, because in the time of LC development, was needed to revisit some visual and functional software possibilities.

The design of the user interface is the basis of building a unique vision of the graphic constructor LC. It is functional, has a clear, but attractive vision and it is oriented to the user in the sense of UX design definition [10]. All main principle for building of interfaces like: aesthetics, transparency, consistency, availability, configuration, control, management, efficiency, awareness, flexibility, predictability and simplicity are kept [16].

The software interface design is made in Adobe Photoshop following the so actual in the moment and easy to distinguish flat design style [6]. This flat trend in the design of advanced interfaces fully embraced by Google and Microsoft, and required also in Apple iOS 7 is characterized by the absence of gradients or if there are any of them - they contain close colors' tints. The characteristics minimalism, from where comes the introduced by Allen Grinshtein term flat-design, is achieved by using no more than three colors with one of their tints [6].

In the LC graphical constructor 2 main colors are used, chosen according the meaning of blue and dark grey, close to the clear back tint, described in the Goin’s book [5], plus one tint of dark grey and one of the blue – light grey.

The choice of user controls is intuitive, and the redesign is made carefully, not to lose their identity or confuse the users. For faster recognition, every button has a label and an icon.

The main spaces are separated graphically and with different color areas: a title bar, a toolbar with buttons, canvas/grid for drawing and saving of logic circuits, a table with functions and results.

The web site, created for users of the LC project [21], keeps the principles of responsive design [4]. Its goal is an optimal and good looking vision of web site or application, suitable under different devices with different display resolutions and other software and hardware differences.

This allows LC constructor's web site to be visible well and available not only for the desktop, but also for lately so popular mobile devices.

For the LC constructor a unique logo is made, also in the colors from the LC constructor interface design, which can be used in the application as an icon and in the site (Fig. 5).

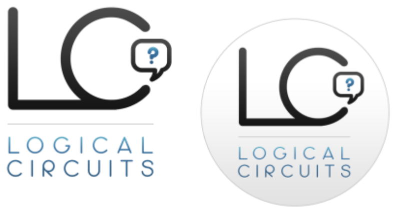


Fig. 5 - Logo and application icon of the graphic constructor LC

For the web site implementation HTML5/CSS3 technology is used, but in combination with jQuery - a javascript library, to provide more consistency under different browsers and devices.

For the software development of the LC constructor Macrosoft Visual Studio 2010 framework is used. The application uses Net 4.0 platform and it is necessary to be installed on the computer, on which LC will be executed. The program development language is C#. In the highest degree the power of Drawing from .Net 4.0 platform is used, in which GDI graphic functionality is available. Drawing2D is the library, part from Drawing, which is used for drawing of 2D graphic elements, supporting complex 2D graphic objects. Also Xml.Serialization library for gates saving on the disc drive and for their later opening from there is used.

IV. ALGORITHMIC SOLUTIONS

A. Algorithms for positioning and drawing

As already mentioned the workspace of LC is a rectangular grid. All gates, participating in the current logic circuit, are placed in it. On the first row (level one) of the grid automatically are arranged so many identity-gates as are the number of variables. On this first row the deployment of other gates is not allowed. The arrangement of the various gates in the grid and the connections between their inputs and outputs are made from the user so that during the placement of the gates the process is controlled not more than one gate in each column of the grid to be placed. The element is dragged to the desired cell and is centered from the constructor. Besides positioning of the various gates it is also necessary to indicate the connections between them.

Every gate's output can be connecting with one or more inputs of other gates, located in the lower level. After the user makes a connection via clicking on the output of one element and on the input of the other, this connection is automatically drawn. To avoid overlapping of connection lines parts, the LC constructor draws the line in three stages: First, starting from the output of the element, a line is drawn vertically until reaching the grid row of the gate, owner of the input. In the top part of every grid row is formed a “virtual highway”, on which all the horizontal connecting lines pass. To avoid overlapping of different ones, this "virtual highway" is divided into so many "tracks" as the sum of the inputs of the gates in that row. Then, on the corresponding track is made the horizontal part of the connecting line until reaching the point, standing above the input to which inclusion is carried. Finally a vertical line from this point to the corresponding input is drawn.

B. Controls, ensuring the correctness of the logic circuit

In the connecting gates process the following controls are provided:

1) *- Controlling all outputs to be connected only with inputs from lower level gates.*
2) *- It is not allowed an output of gate to be connected with input, which has already been connected with output of other gate.*

Before one logic circuit to be saved always the following is checked:

3) *- In the lowest level only one gate to be placed. That condition guarantees the existence of a single output of the entire logic circuit.*
4) *- Whether any output (except the bottom gate) is connected to an input gate. This controls and precludes unnecessary and unused parts of the circuit.*
5) *- Whether any input of a gate (except with identities in the top row) is connected with a gate's output, (for which it is already controlled that can be only one).*

C. Algorithm for moving a gate

The movement of an already existing gate in the circuit differs significantly from the placing of a new one, because of the possibility, its inputs and outputs to be already connected to other gates. For that reason during the gate's moving, simultaneously the controls for the gate's placing, and the controls for all of its connections separately, are made.

If any of these controls fail, the movement action is rejected, and the reason, causing that (for example, because of which connection this movement is denied) is marked. If the gate's movement is acceptable, besides the gate drawing, a redraw of the whole circuit is done, because the parameters of the "virtual highways" (the number of the tracks in them) with all horizontal connecting lines in have been changed.

D. Algorithm for calculating the truth table of the corresponding Boolean function

The algorithm for calculating the truth table of the corresponding Boolean function is standard: On the identities input are given all possible ordered "n"-s from zeros and ones as arguments and consistent, level by level; the results, received from this level located elements are calculated. In that way, on the element's output from the lowest level the result, which corresponding Boolean function gives for the submitted arguments, is shown. The results are saved in the Boolean function's table.

V. Conclusion

The created constructor of logic circuits combines power, simplicity and its own vision. These are unique features, which differ this software from all other graphical tools for creating logic circuits. The graphic systems Logisim [12], TkGate [17], HADES [8], Logic Sim [13], Digital Simulator [3] are intended mainly for construction of electronic components and, due to their complexity, are not suitable for education purposes. But in some aspects the LC constructor even surpasses them. For example, the lack of a limited set of fixed basic gates and the ability to define tabular gates of type „black box". LC has an advantage of all above described systems with the automatic way of drawing the circuits too.

From the graphic systems known by the authors for education purposes are xLogicCircuits [19] and JLS [11]. The first one is primitive and inferior to LC by all components. JLS [11] is a very powerful system, which however makes it too complex – something, which prevents its broad use in learning of discrete mathematic. The constructor LC is better than both of them on several indicators: First – it not only controls, but also helps the process of constructing, by making placing the gates easy and by making automatic the drawing of the connecting lines between the gates. So in the end the logic circuit looks good and maximum tidy. And second – LC doesn't allow constructing of wrong circuit. For example, both - [19] and [11] allow drawing a circuit with loops – something, which LC doesn't.

Without any doubts LC surpasses all of the reviewed graphic systems also with its unique look, thanks to the specially designed interface.

The constructor LC is intended primarily for education. Its possibilities were tested via using the system during exercises on the theory of the Boolean functions based on the books [2], [7], [15] and [18].

The Authors believe that the constructor LC could easily take its place among the existing graphical tools for logic circuits designing.

References

[1] Cwalina K., Abrams B., Framework Design Guidelines, Second Edition, Addison-Wesley, Boston, 2008, ISBN 978-0321545619.

[2] Denev J., Pavlov R., Demetrovich Y., Discrete Mathematics, Nauka i Iskustvo, Sofia 1984 (bulgarian)Tavel, P. 2007 Modeling and Simulation Design. AK Peters Ltd.

[3] Digital Simulator: http://web.mit.edu/ara/www/ds.html, (Last visited 18.10.2013).

[4] Frain B., Responsive Web Design with HTML5 and CSS3, Packt Publishing, Birmingham 2012, ISBN 978-1-84969-318-9.

[5] Goin L., Design for Web Developers – Colour and Layouts for the Artistically Overwhelmed, DMXzone.com, PA Ensched, Netherland 2006, ISBN 90-77397-04-3.

[6] .Grinshtein A., LayerVault, http://layervault.tumblr.com/post/32267022219/flat-interface-design.

[7] Grossman, J. W. :Discrete Mathematics. Macmillan, New York 1990, ISBN 0-02-348331-8.

[8] HADES: http://tams-www.informatik.uni-hamburg.de/applets/hades/webdemos/index.html, (Last visited 18.10.2013).

[9] http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43447, ISO (Last visited 18.12.2013).

[10] http://msdn.microsoft.com , Design Guidelines, UX Design, April 2010.

[11] JLS: http://www.cs.mtu.edu/~pop/jlsp/bin/JLS.html, (Last visited 18.10.2013)

[12] Logisim: http://ozark.hendrix.edu/~burch/logisim/, (Last visited 18.10.2013).

[13] LogicSim: http://www.tetzl.de/java_logic_simulator.html, (Last visited 18.10.2013).

[14] Stoeva M., „Generalized model for Interactive multimedia tools development, which are used for software interface implementation",to appear.

[15] Manev K., An Introduction to Discrete Mathematics, KLMN Sofia 2005, ISBN 954-535-136-5 (bulgarian).

[16] Rahnev A., M. Stoeva, Principles and technologies for building user interfaces of web and desktop applications, "Education in the information society" National conference, Plovdiv, Bulgaria, 27-28 of May 2010, p. 308-316.

[17] TkGate: http://www.tkgate.org/index.html, (Last visited 18.10.2013).

[18] Wuttke H.-D., Henke K.: Schaltsysteme – Eine automatenorientierte Einfuehrung. Pearson Studium, Muenchen 2003, ISBN 3-8273-7035-3.

[19] xLogicCircuits: http://math.hws.edu/TMCM/java/xLogicCircuits/, (Last visited 18.10.2013).

[20] Zeldman J. Designing with Web Standards – Second Edition, New riders, Berkeley 2007, ISBN 0-321-38555-1.

[21] www.lc.myplovdiv.com – site of the project.