

Title:

TEACHING DIGITAL SYSTEMS DESIGN WITH A NEW DIDACTIC ENVIRONMENT
THROUGH THE INTERNET

Scope:

Poster Session of the IUT Conference

Authors:

* Prof. Héctor Pomares

Department of Computer Architecture & Computer Technology

University of Granada

ETS Ing. Informática y de Telecomunicación

E-18071

Granada

Spain

** Author who will be the primary contact*

Teaching Digital Systems Design with a New Didactic Environment through the Internet

H.Pomares, I.Rojas, M.Damas, J.González, J.P.Florido, J.Urquiza

Department of Computer Architecture & Computer Technology, University of Granada, Spain

ABSTRACT

In this work we present the basics of a new didactic environment that has been built during the last few years to facilitate the way students can learn digital systems design. This environment comprises a set of JAVA programs which the students can execute through the Internet and independently of the platform they are using (PC, Macintosh, Solaris, etc.) or the operative system (Windows, Linux, etc.). From our experience with this environment, we can conclude that this system has boosted the motivation of the students since they can readily check if the designs they made for the different problems proposed in the course are correct and, in case they are not, find where they have made a mistake. Thus, this environment eliminates the sensation of frustration that many students feel when they just don't know whether they are doing the correct thing or when it is complex to test if what they are doing is correct. In some cases, this process can take even more time than the design method itself.

Teaching Digital Systems Design with a New Didactic Environment through the Internet

H.Pomares, I.Rojas, M.Damas, J.González, J.P.Florido, J.Urquiza

Department of Computer Architecture & Computer Technology, University of Granada, Spain

Abstract

In this work we present the basics of a new didactic environment that has been built during the last few years to facilitate the way students can learn digital systems design. This environment comprises a set of JAVA programs which the students can execute through the Internet and independently of the platform they are using (PC, Macintosh, Solaris, etc.) or the operative system (Windows, Linux, etc.). From our experience with this environment, we can conclude that this system has boosted the motivation of the students since they can readily check if the designs they made for the different problems proposed in the course are correct and, in case they are not, find where they have made a mistake. Thus, this environment eliminates the sensation of frustration that many students feel when they just don't know whether they are doing the correct thing or when it is complex to test if what they are doing is correct. In some cases, this process can take even more time than the design method itself.

Introduction

E-learning platforms are becoming widespread among educational institutions worldwide, especially as a support tool to Technology-related degree courses. However, most learners require reinforcement tools to increase retention of the course material and advance the learning process (Anisetti 2007, Dormido 2008). Digital systems design is a very important subject in the first courses of electrical and computer engineering. In these courses, students learn Boolean algebra, number systems and codes, combinational logic minimization, the properties and applications of latches and flip-flops, synchronous finite-state-machine design, as well as sequential components such as register files, counters and memories (Tocci 2004, Floyd 2005). The major difficulty they encounter arises when trying to learn the different stages an engineer must carry on in order to design digital systems, both combinational and sequential. And this problem becomes even worse because generally there is more than one possible solution to them, and checking that a given solution is good normally requires a big effort by the students that can provoke a certain anxiety and discourage them to tackle new problems in the course. Being aware of this problem, in this work we propose a new Internet-based didactic environment which helps the student to:

- Learn step by step the different stages necessary to design combinational logic, from the specification of the problem until the attainment of two-level minimized circuits of the form: AND-OR, OR-AND, NAND-NAND and NOR-NOR.
- Check if the solutions they have found are correct without the need to make complex operations which could even take more time than the solving of the problem itself.
- Learn how to design combinational circuits using Read-Only Memories (ROM).
- Understand the multi-functional simplification methodologies used in PLAs, so that they can check that the solutions found by them cannot be improved.

All these features are obtained using a platform-independent environment and without the need to install any specific software (Capilano 1996). The student can use the type of computer they desire (PC, Macintosh, Solaris, etc.) and the operating system they like most (Windows, Linux, etc.). Besides, we have put special emphasis on the reliability, robustness, easiness of use, portability and appealing visual appearance. All these attributes have been considered

indispensable in order to make the student work comfortably and never let it apart because of its complexity, for having errors or for any other reason.

The rest of the paper is organized as follows. After this introduction, in Section 2 we explain how the proposed environment can be used for two-level synthesis of combinational logic. Section 3 and 4 deal with programmable combinational logic, namely, ROM design and PLA design. Finally, some conclusions and future advances about this project are drawn in the last Section.

Synthesis of Switching Functions

Figure 1 shows the main panel of the applications, the one in charge of the synthesis of switching functions. This panel is sub-divided into five fully-related sub-panels:

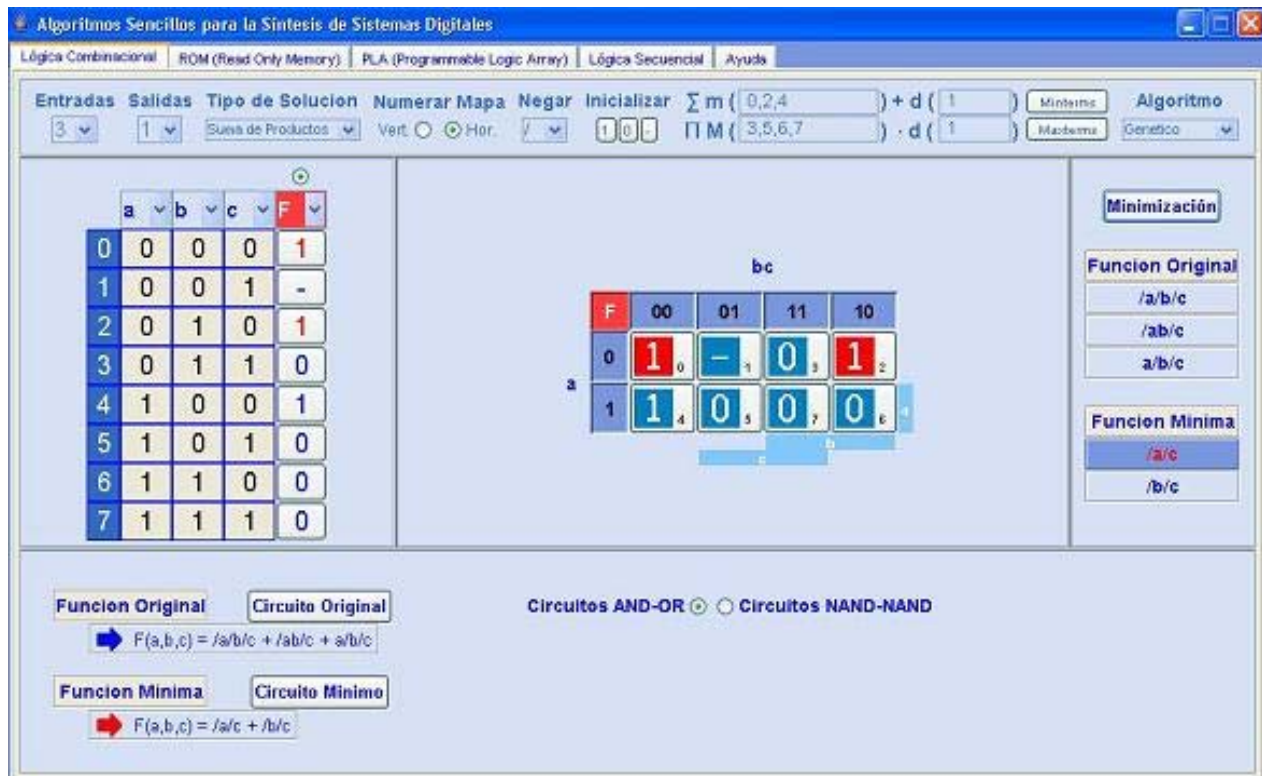


Figure 1. Main panel showing the AND-OR synthesis tools for switching functions

Upper sub-panel: Selection and initialization of combinational functions. In this sub-panel the student can select the number of input variables of the combinational circuit, the number of outputs (i.e. logic functions of the combinational circuit), and the type of solution to be found: AND-OR, OR-AND, NAND-NAND or NOR-NOR synthesis (Nelson 1996, Brown 2000). Besides, there are some other helpful buttons to define these logic functions: initialization using all 0s, all 1s or all “don’t cares”. It also provides a way of fully specifying the whole combinational system by providing the expression of each logic function as a sum of minterms or a product of maxterms (including don’t cares). As the student validates these fields, the rest of the sub-panels are updated accordingly. There are still two more interesting features added to this sub-panel. One is the possibility to choose one out of two different ways of displaying the Karnaugh Map (Holder 2005): with the most significant bit to the right side or to the left side. In the literature we can find these two ways of drawing the Karnaugh Map and it is very important that the students use the one they are accustomed to. The other extra feature of this sub-panel is that the student can choose the symbol they want to use to express the NOT logic operator: some common symbols in the literature are: '/', '¬', '|', '¯'. Finally, the students can select the minimization algorithm to find prime implicants: the classical method or a method based on

genetic algorithms (Michalewicz 1996) which are less computational expensive than classical methods when the number of input variables is large.

Left sub-panel: Truth table, with as many input and output variables as selected in the upper sub-panel. The student can choose the name of each input and output variables and introduce/change any value by just clicking the corresponding cell. This makes the program very robust and easy-to-use. Every time any value is changed in this truth table, this new change is immediately updated in all the sub-panels.

Central sub-panel: Karnaugh map, in which the student can also modify the value of the logic functions by left-clicking on every cell of the map. In the Karnaugh map the student can visualize the different sum or product terms composing the logic function as well as the different prime implicants considered in the minimization of the function.

Right sub-panel: Logic minimization. When the button “minimization” is pressed, the process of automatic synthesis of digital systems with two levels of gates is started. When this process is finished (either using classical methods or genetic algorithms according to the upper sub-panel) the student can visualize all minimal cost solutions. All optimal solutions are presented to the student and not just one of them, as is the case with other minimization tools. This is very important since it is possible that the student has found also an optimum solution and seeing that the program displays a different solution, they can think that they are wrong or, at least, the gain a little sense of frustration. This is, therefore, avoided in our proposed didactic environment. The minimization tool is completed with two additional and not less important didactic features: One of them is visualization of the prime implicants obtained by the minimization process: placing the mouse over the prime implicants the corresponding cells are lighted up in full colours in both the truth table and the Karnaugh map. Likewise, placing the mouse over the terms of the final boolean expression of each function, the corresponding cells are correspondingly lighted up.

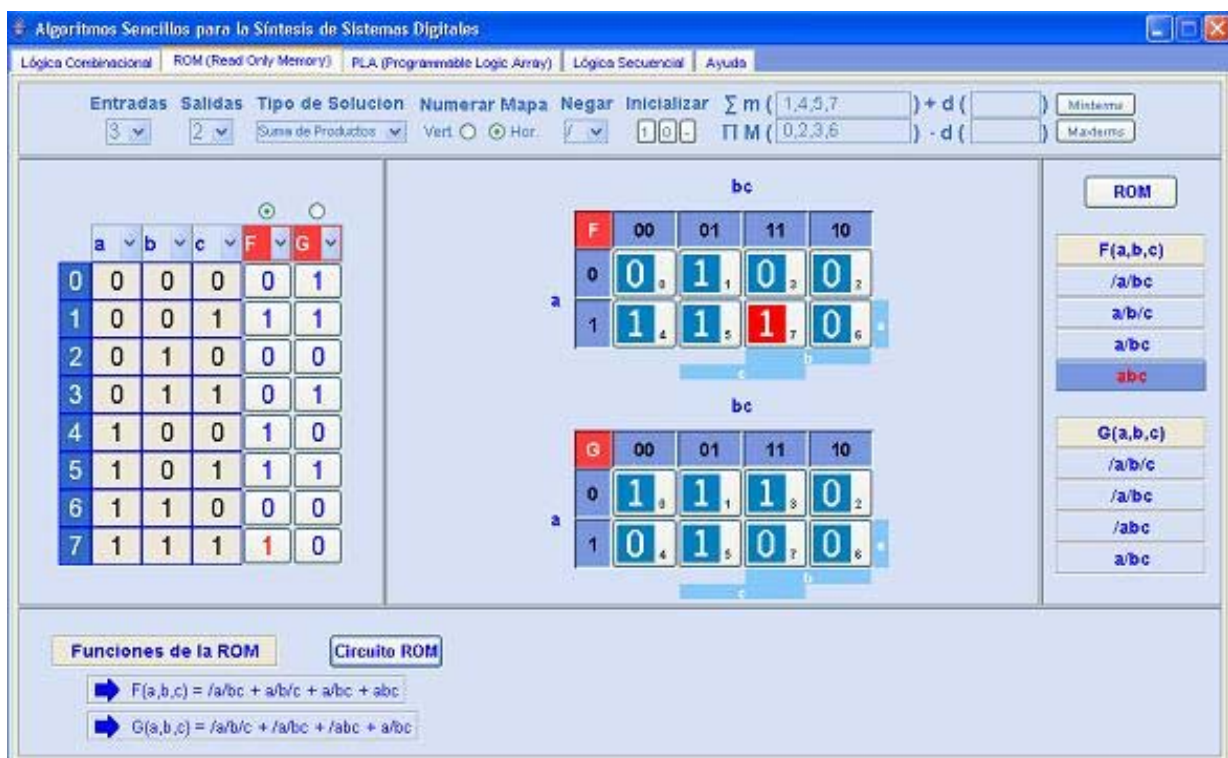


Fig. 2. ROM design panel

Bottom sub-panel: Boolean expressions and circuits. In this sub-panel the student can obtain the boolean expression of the minimized function as sum of products, product of sums, NAND-NAND or NOR-NOR, depending on the kind of solution they wish to obtain (chosen in the upper sub-panel). In the case of existing different equally optimum solutions, all are displayed one after the other. Finally, the student can visualize the corresponding hardware circuits.

Design of Read-Only Memories (ROM)

There are some designs that should be implemented using a ROM memory with optimized size. ROMs can be considered as special cases of two-level AND-OR circuits, where the AND plane implements each and every one of the possible minterms of the input variables, i.e. a binary decoder (Nelson 1996). In that case, the design process must be modified in order to comply with the new specifications. Figure 2 shows the “ROM design panel” of the proposed didactic environment.

As can be seen from the figure, the appearance of the panel is very similar to that of the previous section, which makes it easier for the student to understand how it works. In this case, we have similar interfaces to specify the ROM size, the ROM contents, etc. The program designs the ROM, gives the student the equations of every output value in form of a sum of minterms and graphs a schematic of the final ROM, once the OR plane is correctly programmed (see Figure 3).

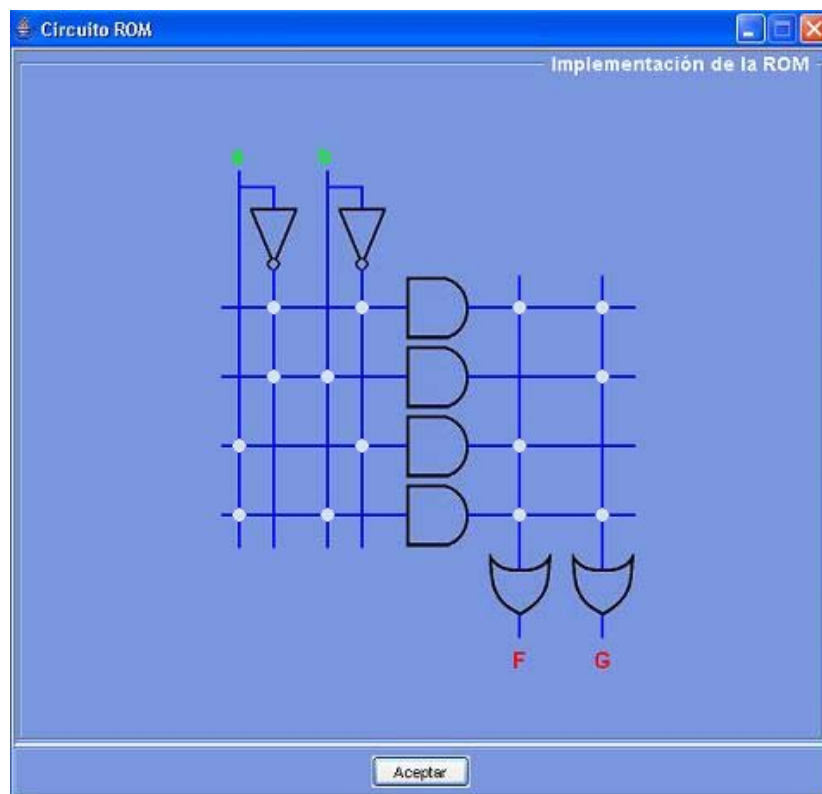


Fig. 3. Schematic of the internal configuration of a certain simple ROM

Programmable Array Logic (PLA) Design

In practical cases, when a digital system designer must synthesize some combinational logic, generally for implementing some interface function (glue logic), this combinational system is implemented in a PLA or a PLA-like structure (Brown 2000). A PLA is composed of an AND plane and an OR plane, but in this case, both are programmable. This entails that every AND gate can be shared by any OR gate, and thus, the design mechanism is more complex.

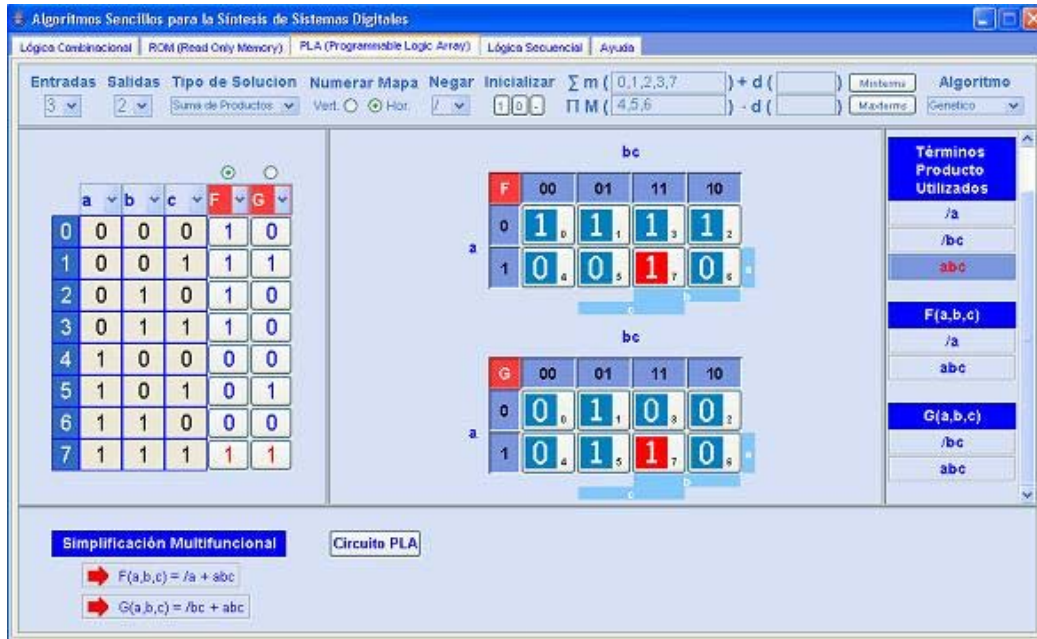


Fig. 4. PLA design sub-panel

Figure 4 shows the PLA sub-panel of our didactic environment. Again the appearance is similar to the other sub-panels so the student does not need any effort to use the different sub-options. The software is capable of performing a multi-functional simplification so that the number of AND gates of the PLA is minimized. Besides, every time the mouse moves over a certain implicant, the corresponding association is highlighted in the Karnaugh maps of those functions that share this implicant. Finally, the PLA can be depicted, as shown in Figure 5.

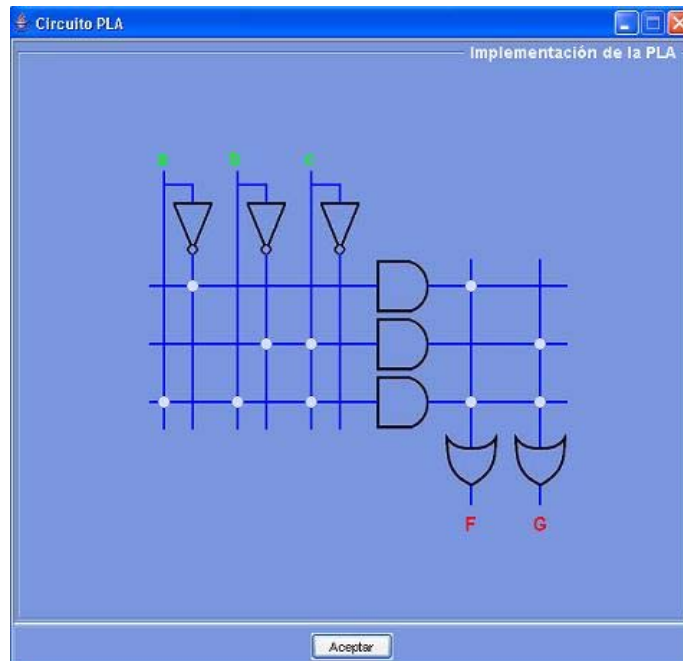


Fig. 5. PLA example

Finally, given the excellent reception our proposed environment has achieved from our students, we can conclude that this tool has proved to be a real help for the suitable development of our course.

Conclusions and Future Work

In this work we have presented a software tool to help students learn digital systems design based on a new didactic environment that has been built during the last few years within the department of computer architecture and computer technology of the University of Granada, in Spain. This environment comprises a set of JAVA programs which the students can execute wherever they are, and whenever they want, through the Internet and independently of the platform they are using (PC, Macintosh, Solaris, etc.) or the operative system (Windows, Linux, etc.). From our experience with this environment, we can conclude that this system has boosted the motivation of the students since they can readily check if the designs they made for the different problems proposed in the course are correct and, in case they are not, find where they have made a mistake.

Thus, this environment eliminates the sensation of frustration that many students feel when they just don't know whether they are doing the correct thing or when it is complex to test if what they are doing is correct. We are already working on improving the features of this environment by inserting a module for the design of sequential systems and the inclusion of a virtual blackboard with which the teacher can interact with the student in order to solve doubts in which digital circuits should be drawn.

References

- M. Anisetti et al, "Learning Computer Networking on Open Paravirtual Laboratories", IEEE Trans. on Education, vol. 50, no. 4, november 2007.
- R. Dormido et al, "Development of a Web-Based Control Laboratory for Automation Technicians: The Three-Tank System", IEEE Trans. on Education, vol. 51, no. 1, february 2008
- Capilano Computing Systems Ltd., "Logic Works 3", Addison Wesley 1996.
- Michel E. Holder, "A Modified Karnaugh Map Technique", IEEE Transactions on Education, vol. 48, no. 1, february 2005.
- S. Brown, Zvonko Vranesic, "Fundamentals of Digital Logic with VHDL Design", McGraw-Hill, 2000.
- V.P.Nelson, H.Troy Nagle, Bill D. Carroll, J.David Irwin, "Digital Logic Circuit Analysis and Design", Prentice Hall, 1996.
- R.J.Tocci, N.S.Widmer, G.L.Moss, "Digital Systems: Principles and Applications", 9th edition, Prentice Hall, 2004.
- T. L. Floyd, "Digital Fundamentals", 9th edition, Prentice Hall, 2005.
- Z. Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs", Springer-Verlag, 1996.