miniFPGA: An Educational App for teaching Partitioning, Placement and Routing on Android Devices

Adrian Moreno-Villalón, Ángel Guerra-Martín, and Eduardo Boemo
DSLab, School of Engineering
Universidad Autónoma de Madrid
28049 Madrid, Spain
eduardo.boemo@uam.es

Abstract— *In this paper is presented an educational application for teaching ideas related to partitioning, placement and routing (PPR) of digital circuit in a generic FPGA (Field-Programmable Gate Arrays) architecture. Tapping in the tactile screen, the student can construct different blocks. They can fill the contents of LUTs (Look-up Tables) and wire them activating segments of a programmable interconnection network. The program works on Android telephones and tablets. The application includes a tutorial, a set of exercises, and auxiliary tools for checking the results or sends them by e-mail.*

Keywords—Android, FPGA, Digital Design, Look-up Tables, Technology Mapping, Shannon Co-factoring

I. INTRODUCTION

Using smartphones as a platform for teaching, technologist interested on education have advantages in terms of time, cost, and reliability that are unthinkable departing from the design of a tailored hardware or software. From the user side, the final product (an interactive tutorial, a book, an exam, etc.) is always available, considering that a smartphone is part of the dairy dressing of any person. For the professor, the model of application store guarantees a worldwide distribution as well as a direct way to update the material.

This work is focused on PPR concepts of RAM-based FPGA technology. These devices are nowadays the principal option to materialize digital circuits, if power consumption is not a limitation. Introduced in the market in 1985, the theme is today part of any curricula of Electrical or Computer Engineering. In education, reprogrammability has been a central advance. It allows the professor to propose laboratory works where student can simulate and construct complex circuits with a minimum cost. All the material is reusable. Teaching electronics with FPGAs is a well-known idea: it has been present at technical conference over the last twenty years [1-7] at least.

Training on FPGA technology at university level is based on the utilization of actual products, normally Xilinx or Altera chips [8], [9]. They let the student to practice with state-of-the art integrated circuits, tools, and documents. However, some drawback exists from the professor side. The complexity of these products sometimes hides the concepts to be taught. Moreover, the fast change of them requires an important effort in upgrading the educational material, which invariantly looks old fashion every few months.

One distinctive difference between a primer course on logic design and FPGA-oriented classes is the use of look-up tables (LUTs) instead of gates. Nowadays, students still learn to construct and optimize circuit thinking on 2-input gates, following the TTL logic design style. However, actual implementations are based on LUTs, not gates. Other important difference in FPGA technology is the emphasis on fan-out and wiring delay reduction, rather than the concept of logic depth minimization.

In this paper, we present a prototype of an Android application designed to experiment simple PPR problems related to FPGAs. Tapping in the tactile screen of a smartphone or a tablet, students read a problem, and then try different solutions. He/she can open LUTs and fill their memory to map different part of the circuit. Finally, these LUTs can be wired using a programmable interconnection and matrix boxes. The tool also gives some tips for checking the results.

II. FPGA ARCHITECTURE

An educational FPGA to be fitted in a mobile phone must be necessarily small. In our case, we designed a structure with only 9 LUTs, 5 files and 4 columns of wires, and 20 interconnection matrix boxes. There are 8 inputs and four outputs available. The flow of data is similar to many commercial FPGAs: from North-West to South-East (Fig.1). Fundamentals of FPGA architectural trade-off have been stated in [10]-[11].

The proposed LUT block is created using 8-1 multiplexers instead the usual 16-1 to 64-1 ones. There is only a flip-flop (FF) associated to each LUT and two outputs: one direct and another registered. The block is completed by a clock signal (Fig.2). Clock tree is global and it is connected to all FF. It is considered a pre-defined part, so that it is not shown in the wiring area of the FPGA. The interconnection matrix allows the designer route any pair of terminal. In Fig.2 it is shown an example.

978-1-4799-6848-0/14/$31.00 ©2014 IEEE

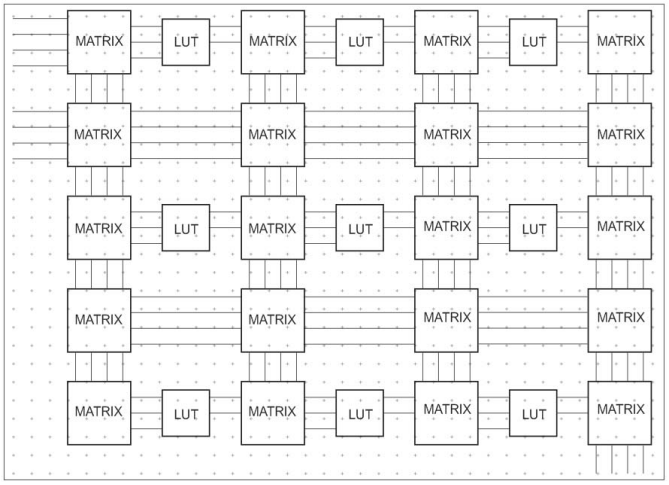

Fig. 1. *FPGA Architecture*

The student manually maps the proposed circuits by tapping in the memory of the LUT or in the interconnection matrix. The size of each problem is designed to fit well with the few available resources.

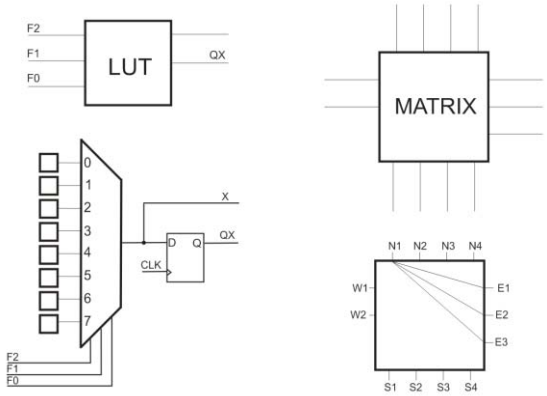

Fig. 2. *LUT Block and Interconnection Matrix details.*

III. EXERCISES AND CONCEPTS

An 8-1 multiplexer is the minimum size of LUT useful for educational purposes. It allows to pack more than 2-input gates and it is also adequate to apply Shannon co-factoring algorithm to map 4 or 5 variable functions. More than 5 variables lead to sizes too big for a mobile phone screens. Some examples of the proposed exercises are:

A. Mapping of basic gates:

The main idea of LUT is that the delay is constant independently of the logic function. Different memory contents emulate different functions, but the hardware is the same (Fig.3)

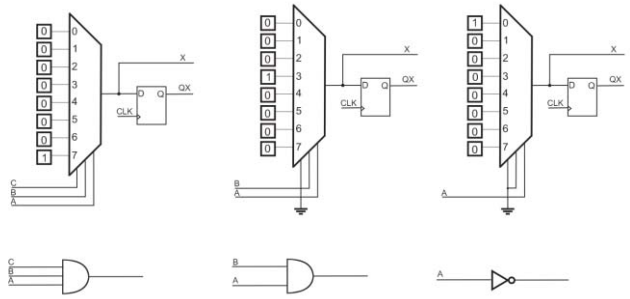

Fig. 3. *LUT utilization for basic functions.*

B. Routing through a LUT:

LUTs not only map function. In some cases, they can be utilized to connect adjacent blocks, saving both wiring resources and extra delay (Fig.4). This option exists on commercial FPGAs; for example, it is called *through-clb* in Xilinx.

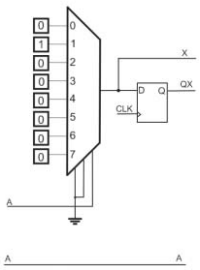

Fig. 4. *LUT Block as a wire.*

C. Shannon Cofactoring Teorem:

Claude Shannon exposed in 1947 several rules of Boolean Logic in [12]. One of them transforms an n-variable function in two functions of (n-1) variables that are finally multiplexed by the eliminated variable. The procedure fit perfect with the problem of mapping functions that have more variables than LUT inputs [13]. For the educational architecture proposed, the power of Shannon idea is evidenced by any function of four variables and four minterms. For example, in Fig.5 is shown a non-optimal map in 3-LUTs of the expression:

$$F(DCBA) = DCBA + /DC/BA + DC/B/A + /D/C/B/A \quad (1)$$

The sign / is utilized to denote variable negation and 3-LUTs is the standard nomenclature to name a LUT of 3 inputs (Fig.5).

In Fig.6 is shown the application of Shannon cofactoring to simplify the original function. In the resultant circuit, the logic depth is reduced by a half, and the number of 3-LUT is lowered from 10 to 3. There is no cost in terms of extra fan-out. Students also can practice other partitioning and mapping strategies.

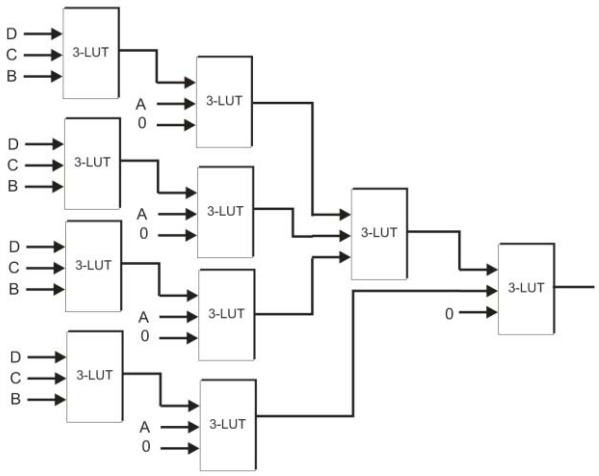

Fig. 5. *Not optimized mapping of a 4-variable fuction with 4 minterms in a 3-input LUT FPGA architecture.*

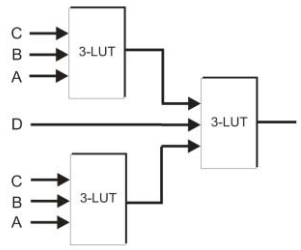

Fig. 6. *Mapping of a 4-variable fuction with 4 minterms in a 3- input LUT FPGA architecture after applying Shannon Cofactoring.*

D. Utilization Common Subexpression.

The proposed FPGA also allows the student to practice with the idea of minimize LUT count by reusing LUTs with common expression. In this case, a cost in terms of extra-fan-out exists.

E. Hardware replication to decrease fan-out

This type of exercises shows the reverse idea of the previous point: LUT duplication to half the fan-out of each LUT output. This option is part of any tool of logic synthesis.

IV. SOME ANDROID HINTS

A disadvantage of Android respect to Apple iOS developments is the variety of devices, screen size, and resolutions on the first. In order to make the app compatible with most of the telephones and tablets, each figure and its clickable area must be parameterized according to the width and height (w x h in pixels) of the particular screen. This information is retrieved using the function:

```
Display display = getWindowManager().getDefaultDisplay();

                  DisplayMetrics metrics = new DisplayMetrics();
                  getWindowManager().getDefaultDisplay().getMetrics
(metrics);

                  int w = metrics.widthPixels;
                  int h = metrics.heightPixels;
```

In order to draw the FPGA layout, the *OnDraw* function must be utilized. Then, a paint object is created to draw the *canvas*, which will be the background of the app:

```
public void onDraw(Canvas canvas)
paint = new Paint();
```

After that, the following functions must be utilized to draw the LUTs, wires and interconnection boxes. The x-y coordinates of the start and end points of each line or rectangle are variable:

```
canvas.drawLine(x, y2+i, x2, y2+i, paint);
canvas.drawRect(x7, yi+i, x7+dc, yf+i, paint);
```

Finally, to pass data from the 3 different screens (main FPGA, LE, and IB) preferences must be utilized. A file is generated to store the name of each variable and its value. In this way, the connections are saved and then can be drawn in the screen of the FPGA layout. The main functions involved are:

```
public void SavePreferences(SharedPreferences multiplexor)
{
      SharedPreferences.Editor editor = multiplexor.edit();
      editor.putInt("Out",valor);
      editor.commit();
}


public void LoadPreferences(SharedPreferences multiplexor) {

      valor = multiplexor.getInt("PreferencesFile", 2)
}
```

The following figures show the final aspect of the application. Short acronyms LE (Logic Element) and IB (Interconnection Blocks) are utilized to indicate LUT blocks (that include an output flip-flop) and interconnection matrix.

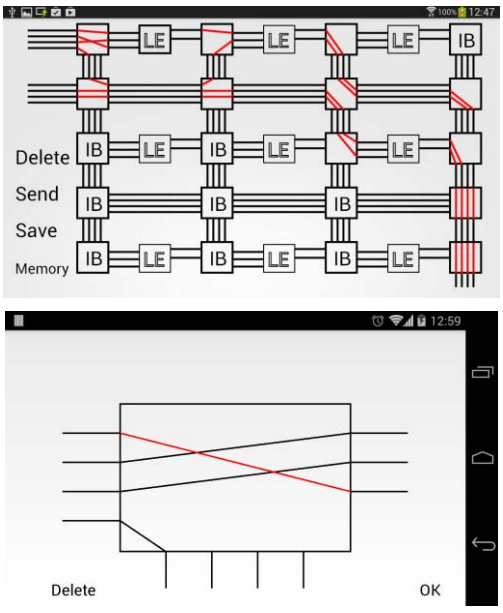

Fig. 7. *Aspect of the application in a 4x7 inches telephone.*

Tapping in the LE or IB, new screens are presented. The student can fills the LUT memory in the LE screen, or indicates pair of interconnection in the IB one.

The current version support Android from Froyo (2.2) hasta Kitkat (4.4), screens Small, Normal, Large y Xlarge, and densities densidades ldpi, mdpi, hdpi, xhdpi, xxhdpi.

V. CONCLUSIONS

In this paper, we have presented an Android application of a small educational FPGA. It can be utilized to practice different trade-offs of the partitioning, placement and routing process. The circuits have been scaled to fit in the screen of different mobile phones. A previous version has been finished [14] and a new version with improved routing capabilities is under test. The app is part of a program to create a set of free educational smartphone-based learning tools in the area Digital Design [15].

ACNOWLEDGEMENTS

Devices and tools for iOS and Android programming have been supported by grants of Banco Santander and Fundación Vodafone España. Additional funds have been obtained from Euroform Polo Español.

REFERENCE

[1] Boemo E. and Meneses J., "Learning VLSI using FPGAs", *Proc. 1993 IEEE Frontiers in Education Conference*, pp.422-425, NY: IEEE Press, 1993.

[2] Meyer, D.G. ; Burke, E.P. "An FPGA-based pedagogical microprocessor for introductory computer engineering courses", *Proc. 1994 Frontiers in Education Conference*, Page(s): 383- 387, NY: IEEE Press, 1994 .

[3] Areibi, S. , "A first course in digital design using VHDL and programmable logic", *Proc 2001 Frontiers in Education Conference*, Vol.1, Page(s): TIC-19-23 vol.1, NY: IEEE Press, 2001.

[4] Wood, S.L. ; Dick, C., "Included in Your Digital Subscription Concepts of parallelism in an introductory computer architecture courses with FPGA laboratories" *Proceedings 2004 Frontiers in Education Conference*, 2004. Page(s): F4C- 15-21, Vol. 2, NY: IEEE Press, 2004.

[5] Quintans, C. ; Valdes, M.D. ; Moure, M.J. ; Fernandez-Ferreira, L. ; Mandado, E., "Digital electronics learning system based on FPGA applications", *Proc. 2005 Frontiers in Education*, Page(s): S2G- 7, NY: IEEE Press, 2005.

[6] Weng, T.; Yi Zhu ; Chung-Kuan Cheng, "Digital design and programmable logic boards: Do students actually learn more?", *Proc. 2008 Frontiers in Education Conference*, Page(s): S1H-1, NY: IEEE Press, 2008.

[7] Xiaofang, W., "Multi-core system education through a hands-on project on FPGAs, *Proc. 2011 Frontiers in Education Conference*, Page(s): F2G-1- F2G-6, NY: IEEE Press, 2011.

[8] Xilinx Inc. "Xilinx University Program", http://www.xilinx.com /support/ university.html

[9] Altera Corp. "Altera University Program—Learning Through Innovation", http://www.altera.com/education/univ/unv-index.html

[10] Rose, J. ; El Gamal, A. ; Sangiovanni-Vincentelli, A. "Architecture of field-programmable gate arrays", *Proceedings of the IEEE*, Vol. 81, Issue: 7, Page(s): 1013- 1029, 1993.

[11] Brown, S., "FPGA architectural research: a survey", *IEEE Design & Test of Computers*, Vol. 13, Issue: 4 , Page(s): 9-15, 1996.

[12] Shannon, C., "The Synthesis of Two Terminal Switching Circuits", *Bell System Technical Journal*, v28: i1, Jan. 1949.

[13] Francis, R. "A tutorial on logic synthesis for lookup-table based FPGAs", *1992 IEEE/ACM Computer-Aided Design (ICCAD-92)*, Page(s): 40-47, 1992.

[14] Guerra Martín, A. *"Aplicación Android para simular una FPGA educativa"*, Master Thesis School of Engineering, Universidad Autónoma de Madrid, June 2013 (in Spanish).

[15] DSLab, *"Sequential and Combinational Circuits",* availables at Google Play at: https://play.google.com/store/apps/developer?id=DSLab+UAM