Dynamic Reconfiguration in FPAA and its use in Education

Alejandro Macho
Electrical and Computer Engineering Department, DIEEC
Spanish University for Distance Education, UNED
Madrid, Spain
amacho@ieec.uned.es

Manuel Garcia Teruel
Electrical, Electronics, Automation and Communications Engineering Department
University of Castilla-La Mancha, UCLM
Albacete, Spain
manuel.garcia@uclm.es

Pablo Baizan
Electrical and Computer Engineering Department, DIEEC
Spanish University for Distance Education, UNED
Madrid, Spain
pbaizan@ieec.uned.es

Manuel Blazquez
Electrical and Computer Engineering Department, DIEEC
Spanish University for Distance Education, UNED
Madrid, Spain
mblazquez@ieec.uned.es

Felix Garcia-Loro
Electrical and Computer Engineering Department, DIEEC
Spanish University for Distance Education, UNED
Madrid, Spain
fgarcialoro@ieec.uned.es

Elio Sancristobal
Electrical and Computer Engineering Department, DIEEC
Spanish University for Distance Education, UNED
Madrid, Spain
elio@ieec.uned.es

Gabriel Diaz
Electrical and Computer Engineering Department, DIEEC
Spanish University for Distance Education, UNED
Madrid, Spain
gdiaz@ieec.uned.es

Rosario Gil
Electrical and Computer Engineering Department, DIEEC
Spanish University for Distance Education, UNED
Madrid, Spain
rgil@ieec.uned.e

Manuel Castro
Electrical and Computer Engineering Department, DIEEC
Spanish University for Distance Education, UNED
Madrid, Spain
mcastro@ieec.uned.es

***Abstract*— The purpose of this article is to study the dynamic reconfiguration in FPAA and, because of its potential, its academic applications. State driven and algorithmic reconfiguration methods have been considered during this work. Since these devices are not as well-known as FPGA, it is interesting to study its characteristics and abilities. The algorithmic method has been developed, obtaining conclusions about its use in education for technical and non-technical fields.**

***Index Terms*— FPAA, Dynamic Reconfiguration, Anadigm, Education, Lab**

I. Introduction

Reconfiguration is one of the most versatile and powerful techniques in the electronics field. Microcontrollers or FPGA can be reconfigured but, for instance, most of microcontrollers cannot be dynamically reconfigured. In other words, they need to be reset in order to work with the new configuration.

Dynamic reconfiguration avoids this issue and allows devices to change their functionality without a reset. FPAA, which are not as widely known as FPGA, can use this technology to reconfigure themselves, achieving different analog behaviors. It can be used in education, where this dynamic reconfiguration provides new tools and new possibilities to students and professors, and in several fields like avionics, signal processing, etc.

The importance of this technique in education resides in the ability of these devices in adapting themselves quickly to many situations while everything is under control in a safety environment.

This work tries to develop an example of educational application that could be applied to empower students to get more concepts while working at labs (or in remote labs). In order to achieve that, and with the aim to let other researchers or students to put this in practice, this work has used a simple methodology of six steps, that can be found on its own section. Anadigm devices have been used in this work, since it is the main manufacturer and its devices are cheap enough to be considered for educational purposes. Anadigm design and simulation software have been used but, as described during this report, simulation tools are sometimes insufficient. It is also an objective of this work to do an analysis about the different reconfigurations methods: the algorithmic method and the state driven method. The first one of these methods is the most powerful for educational and industrial applications, that is the reason why this method has been studied in a deeper way. Despite that some other works study the FPAA, this work aims to study its ability to dynamically reconfigure itself and introduce its application in education for technical and non-technical students. Analyzing, also, its benefits for students, professors and education centers. This practical work is the foundation of this article.

An educational development can be found in this article and it will show the different approaches that can support the learning of engineering students, electronic related studies or not, and even non-engineering students. Researchers can use

978-1-5090-5920-1/17/$31.00 ©2017 IEEE

this information to develop new tools and applications to many education fields, and it could be also applied to remote labs. Definitions, a briefly history introduction, a comparison of the two methods of reconfiguration, the limits of the simulation tools and, of course, the conclusions of the use in education of FPAA dynamic reconfiguration, compose this article. This publication is a result of the Master Thesis of the author [1].

II. Definition and Material

A. FPAA

Research in programmable analog devices has not been as wide as in logic devices. The term FPAA was first used by Lee, K. and Gulak, P. in 1991 [2], introducing the CAB concept. These CAB are units which can develop different functions and can be interconnected between them in order to perform complex tasks. The most common interconnection technology is the MOS transistor controlled by a digital memory [2-6].

A field-programmable analog array (FPAA) is an integrated device containing configurable analog blocks (CAB) which are interconnected between them. FPAA are usually application driven because of their different architectures and operation modes.

Nowadays, there are different architectures where other basic units are used instead of CAB or they change the way that CAB are routed [7]. These architectures can operate in continuous and discrete time.

B. Discrete Time FPAA

Analog devices usually work and operate in a continuous time environment where each input or output signals can exist in an unlimited range of values. In continuous time, FPAA switch matrix’s parasitic inductance, capacitance and noise contributions should be taken into account.

In order to avoid these problems while achieving more flexibility, discrete time devices are used. These FPAA use sample and hold circuits in each CAB output, obtaining a continuous signal but operating in discrete time with high frequency. In this kind of devices, switching noise and aliasing must be taken into consideration during the design phase [8].

The main technique used in discrete time FPAA is the switched capacitor technology. With this technology, a resistor can be obtained from a capacitor. Figure 1 illustrates how switched capacitors are configured as resistors. The resistance value will change according to the sampling frequency (f=1/T) [9].

This is one of the most important concepts of an FPAA because this technology allows for the creation of different resistors and can change its value on the fly. This is one of the fundamentals of dynamic reconfiguration.

C. State of the Art

FPAA are not as widely spread as their digital equivalent, FPGA, so there is not a lot of research in this field. There is also a device that uses mixed-signal processing, the FPMA [10]. These kinds of devices are even less used than FPAA. The reason is that hybrid FPGA-FPAA systems are achieving great results and it is complex to develop a new mixed technology like FPMA.

However, the robust capabilities of FPAA have been used in several disciplines for very specific functions like adaptive filters for the health field, in instrumentation [11-15] or hybrid FPGA-FPAA systems [16].

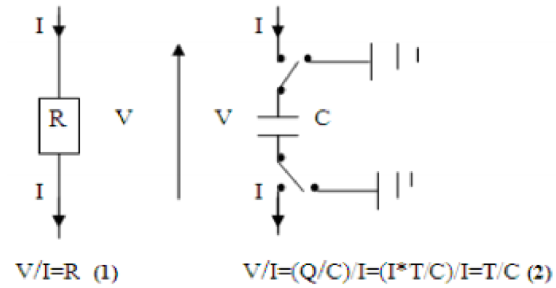

Figure 1. Switched Capacitor, [9].

D. Anadigm AN221E04 FPAA and AN221K04-V3

There are very few manufacturers of FPAA, with Anadigm being the principal developer. There is not a great amount of on-chip solutions, but because of its price and possibilities, the AN221E04 is the chip used in this development.

The AN221E04 [17] device, the first one of the second generation, consists of a 2x2 matrix of CABs that can be interconnected between them. These CABs are surrounded by analog input/output cells (4 configurable I/O cells and 2 output cells) with active elements like filters or amplifiers. The architecture scheme can be found in Figure 2.

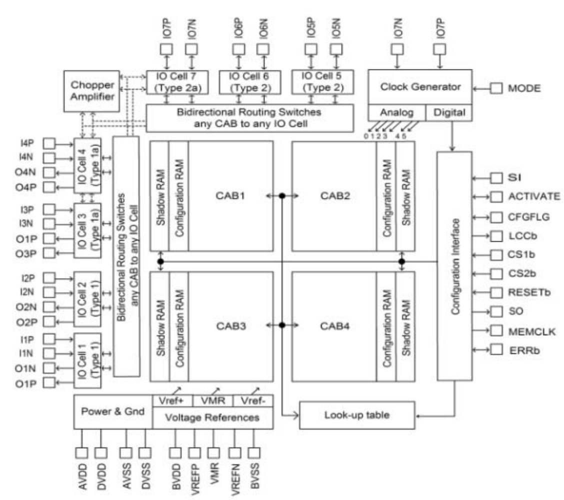

Figure 2. CAB structure of the AN221E04, [17]

This device includes 2 built-in memory blocks where configuration is stored. There is one SRAM block and a special memory block called Shadow RAM (SRAM). This special block allows the device to reconfigure itself dynamically and is the most important feature that makes dynamic reconfiguration possible.

The AN221E04 chip can be found in the AN221K04-V3 development board, among others, and it includes a PIC to program the FPAA and a good interface to test the device and to develop with it. The board also includes a serial port to connect the FPAA with a computer in order to program the chip. This FPAA-PC connection is very important in reconfiguration when used in educational environments.

E. AnadigmDesigner 2

Designing and implementing a circuit in the FPAA can be done through Electronic Design Automation (EDA) software. AnadigmDesigner2 [18] is the solution from the board manufacturer used in this project.

The process of implementing a design is made simple thanks to Configurable Analog Module (CAM). These modules can be dropped and connected in a visual interface in order to obtain the desired functionality. The software also provides an Assistant in order to facilitate the understanding of capabilities and limits of each CAM for the user. Figure 3. shows the AnadigmDesigner2 user interface.

This EDA software also includes different reconfiguration methods that can be easily used by the user. These methods depend on the chip and board used.

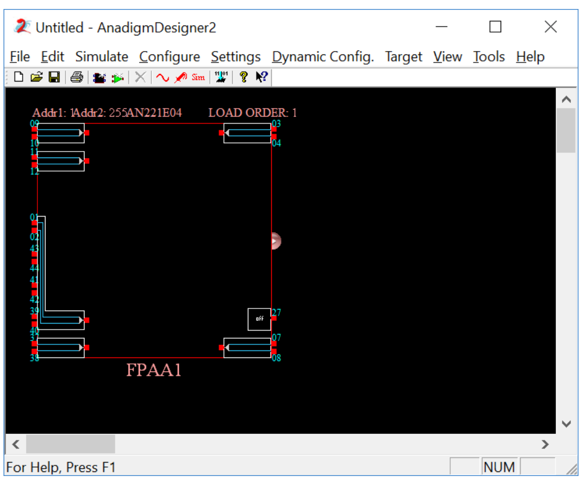

Figure 3. AnadigmDesigner2 user interface

III. Dynamic Reconfiguration Methods

Dynamic reconfiguration in FPAA modifies some parameters and functionalities of certain CAM. This reconfiguration is made on-the-fly because of Shadow SRAM memory blocks, as mentioned above. Anadigm allows the FPAA to reconfigured dynamically through two different processes: the state-driven method [19] and the algorithmic method [20, 21].

Operating under the state-driven method implies that all possible states are known during the design stage. These different states should be known, designed and stored in order to load them when necessary. This kind of reconfiguration is usually used in sound effects where transitions are set by time periods. In order to run this method, a microcontroller like a PIC [19] is necessary to store and load different FPAA configurations.

The algorithmic method is a more complex and has a bigger potential than the state driven method because of its functionalities. One of the main problems of the state driven method is memory limit, so it is not possible to store infinite transitions and configurations. The algorithmic technique avoids this problem since it uses a C library to generate a new configuration and, then, it loads the new configuration. So, it needs only memory to store the C library and the ability to execute it. A common PC or PIC are usually suitable to use this technique.

The aforementioned data lead us to believe that the algorithmic method is better and that it generates more opportunities and functionalities than the state driven method. It is true, but the state driven method is a great option and suitable in low-cost systems with low resources like memory or processing capacity.

IV. Methodology

Methodology and best practices are crucial in order to achieve good results and to guarantee a correct development. The methodology used in this project runs through the following stages.

A. System designing.

First of all, the system needs to be designed and the functional and technical requirements must be accomplished. This design can contain any operation that can be implemented in the FPAA with CAM. During this stage, FPAA electrical limitations like voltage limit should be taken into account.

B. AnadgimDesigner2 implementation.

Once the system is designed, this functionality should be implemented into the FPAA using CAM while wiring each one to another in regard to obtaining the desired behavior. Limitations of each CAM and its parameters range must be considered in next stages, so it is a good practice to implement the circuit taken care about parameters ranges.

C. C libraries.

AnadigmDesigner2 can export a C library with all functions related to the CAM used. Doing that as well is good practice because it minimizes the required memory size for the library in the PC or microcontroller.

D. Creating the control program.

The control program is in charge of all reconfiguration commands. In this project, the control program is implemented in a computer which is wired to the development board by the serial port. The C libraries should be included in this step and an appropriate interface has to be designed in order to facilitate the operation for users.

E. Implementing the design and configuration limits.

In analog systems, limits are the most important constraint. Suitable ranges, for instance, are ones of those limits that should be taken into account. CAM parameters can only work in a range of parameters and the entire FPAA cannot work with more than 4 V. These constraints should be implemented in the developed control software to avoid any problem during the reconfiguration. This is a critical stage and any error could possibly damage the entire board.

F. Compiling and testing.

Finally, we can compile the program and test the results. Visual Studio 2010 and Visual Studio 2015 have been used as compilers in this project. Please note that some interface features can vary from one version to another. Other C++ compilers can be used and libraries can be translated into other programming languages.

V. SIMULATING THE FPAA DESIGN

One of the most important steps during the design process is the simulation stage. This step is even more important in education, because students often do not have enough devices to test their designs and simulation allows them to save time and even to do a pre-work from home.

In order to simulate a design in a FPAA there are not many options but there are some generic ones like [22]. However, these solutions are not really focus on final applications and are not user friendly with most of students.

AnadigmDesigner2 offers a simulation solution for designs made for their devices. This solution offer a user-friendly interface with enough capabilities to test most of designs.

This simulation is made in a time domain, where the user is able to adapt the different clocks in the device and the input. However, there are also limitations, for example it is not possible to simulate a current input.

Simulation in FPAA is quite behind the FPGA or PIC simulation but current tools allow student to easily test different solutions and designs.

VI. EXAMPLE: IMPLEMENTING DYNAMIC RECONFIGURATION ADAPTING A TEMPERATURE SENSOR

Common problems while adapting sensors are resolution and precision. Most of times, a conditioner circuit is needed to adapt the values measured in order to be converted to a digital signal in a proper way.

A PT100 conditioner circuit can be implemented in a FPAA using only a few CAM (the scheme is shown in Figure 4). First, ranges and goals should be named, in this example the measured range goes from 0 °C to 40 °C and the output should be read by a PIC, which can only read tensions from 0 V to 3 V.

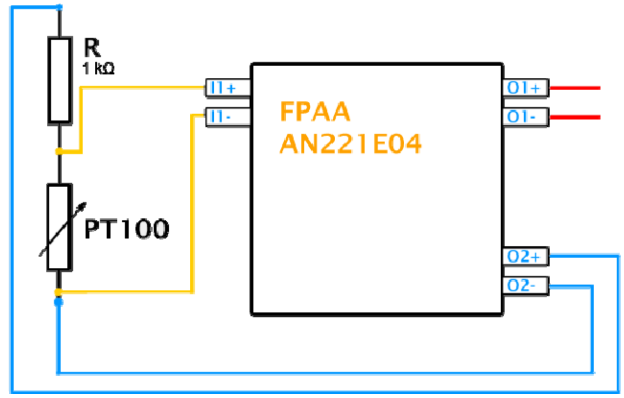

Figure 4. Scheme, PT100 conditioner design for FPAA

PT100 resistive values are:

$$PT100_{0\,°C} = 100\ \Omega$$
$$PT100_{40\,°C} = 115.54\ \Omega$$

The supply source is provided by the FPAA and the actual value should be read. In this case, the value applied to the voltage divider is 2.88 V. The real resistance value is 1001 Ω. Adding a pre-amplification stage of 5 V/V:

$$V_{PT100\,(0°C)} = \frac{2.88}{1001 + 100} \cdot 100 \cdot 5 = 1.3079\ V$$
$$V_{PT100\,(40°C)} = \frac{2.88}{1001 + 115.54} \cdot 115.54 \cdot 5 = 1.4901\ V$$

In order to adapt the signal to the range chosen (0 V, 3 V):

$$(1.3079 + y) \cdot x = 0\ V$$
$$(1.4901 + y) \cdot x = 3\ V$$

And, the solution is:

$$x = 16.47 \qquad y = -1.307$$

Note that the internal operational limit for the FPAA is 4 V, so any calculation should not exceed that limit in any stage of the process.

The design should be implemented in the FPAA as shown in Figure 5. A gain-limiter CAM has been used in order to protect the reader device of voltages higher than 3 V.

Finally, dynamic reconfiguration is applied using the algorithmic method, the C libraries and Visual Studio. A program has been developed and it allows the user to adapt the temperature and output ranges without making any analysis or calculation. Program, in Figure 6, reconfigures the board when it is running on a PC connected to the board.

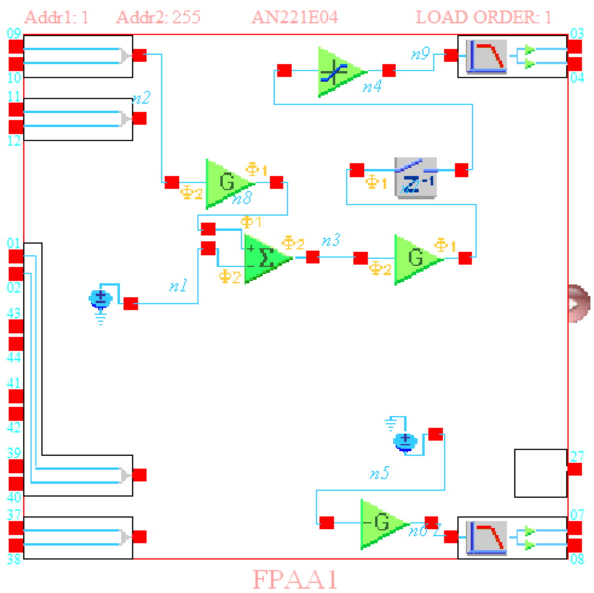

Figure 5. Implemented Design

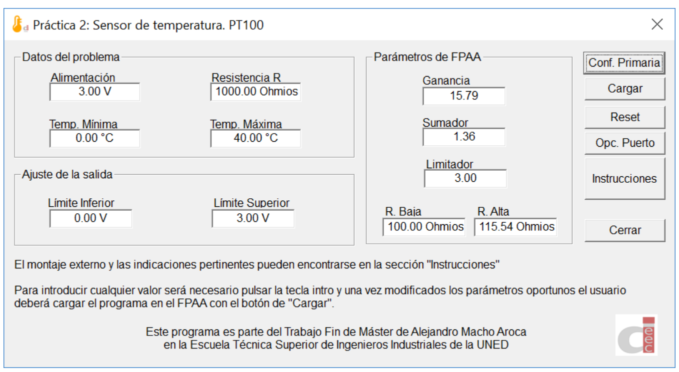

Figure 6. User interface of the developed program

VII. FPAA IN EDUCATION

FPAA allow student to test analog design in an easy way in a controlled and safety environment. This situation can lead in laboratories where FPAA is a main component of them.

A. Education in engineering

1) Electrical and Electronics Engineering

Students of electrical or electronics engineering are familiarized with analog concepts and they are able to create applications using discrete components. However, in a higher level, where inputs or analog processing is only a little part of their applications, they usually do not have enough time in labs to test different options or to fix errors.

In these cases, student can be supported by the FPAA [23] to generate inputs, noise, filters or to adapt their outputs in order to read them in another device. It will save time and components required to build a solution, but the most important thing is that it will allow the student to reach new knowledge and concepts using the same amount of time in university labs.

Students can use the dynamic reconfiguration to build robust systems and applications with many functionalities and options and, of course, it can be applied to other areas like control, where students usually think about microcontrollers [24].

2) Non-electrical or electronics engineering students

For example, mechanical engineering students can be interested in knowing how electrical concepts and circumstances can interfere in their works. These students usually do not have the knowledge to build custom solutions with discrete components in order to achieve their goals. In these situations, the FPAA can help them to experiment without the necessity of investing too much time acquiring the knowledge required to create this kind of applications. Professors can build solutions using dynamic reconfiguration to let their students to experiment with electrical concepts like frequencies, currents, voltages, integrators, etc. without having a deep understanding of these concepts. Using this option will lead in a more understanding of these practical concepts and to interact with them with only a few hours in labs, instead of investing hours in classes and accomplishing less goals.

B. Non-engineering education

There are many fields where electricals and electronics concepts are used and let students get a better understanding of their fields of study. For example, in medicine or in biology. Students can experiment with different frequencies and currents to know how they affect to a human being, to the heart or a tissue. However, these students usually do not have the technical background required to build a custom solution or to modify an existent one. Labs based on FPAA give a great amount of experiments where students can learn and get a deeper understanding of the electrical concepts involved and saving time to really get the concepts of their studies.

C. For educational institutions

Educational institutions can use FPAA based labs to save resources and spreading the opportunities they bring to students. Furthermore, if the FPAA are well designed and designed focusing in educational purposes, they are the perfect safety environment to study and learn, to grasp deep concepts and avoiding time consuming errors and knowledge barriers to non-technical students. These labs can be also deployed to server as a remote laboratory because of their dynamic reconfiguration capability students and, of course, large FPAA system could improve the features of these labs [25]. Thanks to this capability, students can perform different experiments, changing parameters in real time and observing the results.

VIII. RESULTS

A. Developed Programs

Four different programs have been developed in order to experiment with the possibilities of the algorithmic method in the dynamic reconfiguration: a lighting control program, a temperature measurement control, a light controlled trigger and a program to control two interconnected boards.

The programs reconfigure the FPAA in different ways. For example, when a slider bar is moved or when a new configuration, which has been checked by the user, is loaded. These developments work with a photodiode and a PT100 sensors and reconfigure the FPAA to operate under different situations with different goals. That shows how dynamic reconfiguration can be a tool to support students but also as a final application.

B. Dynamic Reconfiguration in Education

These programs have been used in four lab practices by students where dynamic reconfiguration has an active or passive role. The reconfiguration can be used to test more designs in less time while avoiding the probability of failure. For example, a filter CAM can be dynamically reconfigured to vary its pass frequency. In this case, the student will not need to modify the design he or she made.

Playing an active role, dynamic reconfiguration used by students can show them how to create applications for real situations and how to use resources, time and components, in a more efficient way. Following a series of instructions, students are able to create their own reconfiguration programs. With them, they can study a great deal of situations with only one circuit and without changing any components.

FPAA are not as known as FPGA, so this kind of techniques and methods help these devices be known by the scientific community and students. Its potential in several fields is outstanding and hybrid auto-reconfigurable systems (FPGA-FPAA, PIC-FPAA) [16] can make a difference in control issues.

IX. Conclusion

Dynamic reconfiguration in programmable analog devices is a new technique that allows the device to reconfigure themselves without changing their components and without a reset. Most analog systems present different issues because of temperature changes, different noises or tolerances. Dynamic reconfiguration avoids these problems because of the adaptability of the systems that uses this technique.

However, there are also limits in this technology and these limits depend on the reconfiguration method. In the state-driven method, limitations appear because of memory and non-adaptability constraints. It is also a problem that all reconfigurations should be known during the design stage, so it is not a true adaptive system. The algorithmic method needs a power controller or a PC to work, since it needs to execute the C libraries. Reconfiguration timing must be taken in account. Each CAM has its own timing. Due to that timing, performing a great quantity of reconfigurations in a short period of time can make the device reset itself. To avoid this kind of problems they should be considered during the design stage and it is not easy to know these constraints before testing.

A. Next Steps

Following the technique explained in this report the next step is to implement it in autonomous intelligent systems. A hybrid FPGA-FPAA system would be able to sense the environment and reconfigure itself in order to improve its performance. Research in this field can provide a good point for other investigations like predictive adaptive expert control, fuzzy logic, auto-calibrated measurements [12, 13], smart grids [11] or neural networks [2, 14, 26] science.

An interesting next step in the education field is that dynamic reconfiguration can be used in many fields, not only engineering. Developing simulated environments that can be reconfigured allows students and professors to learn and investigate with better and quicker tools.

Acknowledgments

The authors acknowledge the support of the eMadrid project (Investigación y Desarrollo de Tecnologías Educativas en la Comunidad de Madrid) - S2013/ICE-2715, VISIR+ project (Educational Modules for Electric and Electronic Circuits Theory and Practice following an Enquiry-based Teaching and Learning Methodology supported by VISIR), Erasmus+ Capacity Building in Higher Education 2015 nº 561735-EPP-1- 2015-1-PT-EPPKA2-CBHE-JP and PILAR project (Platform Integration of Laboratories based on the Architecture of visiR), Erasmus+ Strategic Partnership nº 2016-1- ES01-KA203-025327. And to the Education Innovation Project (PIE) of UNED, GID2016-17-1, “Prácticas remotas de electrónica en la UNED, Europa y Latinoamérica con Visir – PR-VISIR”, from the Academic and Quality Vicerectorate and the IUED (Instituto Universitario de Educación a Distancia) of the UNED.

This work is made within the Spanish University for Distance Education (UNED), in Spain, as part of the PhD Program on Advanced Electronics.

References

[1] A. Macho, "'La Reconfiguración Dinámica en FPAA, sus Aplicaciones y el Estudio de su Implementación en Sistemas de Educación," Master Thesis, Electrical and Computer Engineering Department. Spanish University for Distance Education, UNED, Madrid, Spain. Jul. 2016.

[2] E.K.F. Lee and P.G. Gulak, "'A CMOS field-programmable analog array," *IEEE J Solid State Circuits*, vol. 26, no. 12, pp. 1860-1867.

[3] D. Anderson, C. Marcjan, D. Bersch, H. Anderson, P. Hu, O. Palusinski, D. Gettman, I. Macbeth and A. Bratt, "'A field programmable analog array and its application," *Proceedings of CICC 97 - Custom Integrated Circuits Conference*, pp. 555-558.

[4] S.T. Chang, B.R. Hayes-Gill and C.J. Paull, "'Multi-function block for a switched current field programmable

analogue array," *Proceedings of the 39th Midwest Symposium on Circuits and Systems*, vol. 1, pp. 158-161 vol.1.

[5] V.C. Gaudet and P.G. Gulak, "'CMOS implementation of a current conveyor-based field-programmable analog array," *Conference Record of the Thirty-First Asilomar Conference on Signals, Systems and Computers (Cat. No.97CB36136)*, vol. 2, pp. 1156-1159 vol.2.

[6] B. Pankiewicz, M. Wojcikowski, S. Szczepanski and Y. Sun, "'A CMOS field programmable analog array and its application in continuous-time OTA-C filter design," *ISCAS 2001. The 2001 IEEE International Symposium on Circuits and Systems (Cat. No.01CH37196)*, vol. 1, pp. 5-8 vol. 1.

[7] S.H.K. Embabi, X. Quan, N. Oki, A. Manjrekar and E. Sánchez-Sinencio, "'A Current-Mode based Field-Programmable Analog Array for Signal Processing Applications," *Analog Integr.Cir.Signal Proc.*, vol. 17, no. 1, pp. 125-142.

[8] S.H. Tyson, "'Field Programmable Analog Arrays: A Floating Gate Approach," School of Electrical and Computer Engineering, Georgia Institute of Technology, 2004.

[9] M. Karmani, C. Khedhiri and B. Hamdi, "'Analog Integrated Circuit Design and Testing using the field programmable analog array technology," *IJAET*, vol. 1, no. 4, pp. 1-9.

[10] J. Stehlik and D. Becvar, "'Field Programmable Mixed-Signal Arrays (FPMA) Using Versatile Current/Voltage Conveyor Structures," *Personal Wireless Communications*, vol. 245, pp. 628-636.

[11] A.S. Deese and C.O. Nwankpa, "'Design and Testing of Custom FPAA Hardware With Improved Scalability for Emulation of Smart Grids," *IEEE Transactions on Smart Grid*, vol. 5, no. 3, pp. 1369-1378.

[12] M. Kouider, M. Nadi, J. Prado and D. Kourtiche, "'Embedded system design and implementation of standard auto-calibrated measurement chain,", pp. 377-382.

[13] D.P. Morales, A. Garcia, A.J. Palma, M.A. Carvajal, E. Castillo and L.F. Capitan-Vallvey, "'Enhancing ADC resolution through Field Programmable Analog Array dynamic reconfiguration," *2008 International Conference on Field Programmable Logic and Applications*, pp. 635-638.

[14] A. Zbrzeski, P. Hasler, F. Kolbl, E. Syed, N. Lewis and S. Renaud, "'A programmable bioamplifier on FPAA for in vivo neural recording," *2010 Biomedical Circuits and Systems Conference (BioCAS)*, pp. 114-117.

[15] M.A. Terry, J. Marcus, M. Farrell, V. Aggarwal and U. O'Reilly, "'GRACE: Generative Robust Analog Circuit Exploration," *Applications of Evolutionary Computing*, vol. 3907, pp. 332-343.

[16] L. Znamirowski, O.A. Palusinski and C. Reiser, "'Optimization Technique for Dynamic Reconfiguration of Programmable Analog/Digital Arrays," *Analog Integr.Cir.Signal Proc.*, vol. 31, no. 1, pp. 19-30.

[17] Anadigm, "'AN221E04 Datasheet Dynamically Reconfigurable FPAA With Enhanced I/O,", vol. DS030100-U006c.

[18] Anadigm.com. (2017). AnadigmDesigner®2 Software. [online] Available at: http://www.anadigm.com/anadigmdesigner2.asp [Accessed 20 Apr. 2017].

[19] Anadigm, "'App Note - 305 Application Note: State-Driven Control of a dpASP using a Microchip PIC.", vol. AN231010-U305.

[20] Anadigm, "'App Note - 208 Creating a DLL from An adigmDesigner2 C-code,", vol. AN221009-U208A.

[21] Anadigm, "'Apex Filter Demo Application Note,", vol. AN231K02-D001 B.

[22] A. Minocha, "'Simulation Tool for FPAA,", vol. 2016, no. 06-05, 2012.

[23] M. Collins, J. Hasler and S. Shah, "'An approach to using RASP tools in analog systems education," *FIE*, pp. 1.

[24] C. Choi, "'Comparing microcontroller and analog methods for tracking control experiments," *2015 IEEE Frontiers in Education Conference (FIE)*, pp. 1-4.

[25] C.M. Twigg and P.E. Hasler, "'Incorporating Large-Scale FPAAs Into Analog Design and Test Courses," *IEEE Transactions on Education*, vol. 51, no. 3, pp. 319-324.

[26] P. Rocke, B. McGinley, J. Maher, F. Morgan and J. Harkin, "'Investigating the Suitability of FPAAs for Evolved Hardware Spiking Neural Networks," *Evolvable Systems: From Biology to Hardware*, vol. 5216, pp. 118-129.