Comparison of TERO-cell implementations and characterisation on SRAM FPGAs

Cédric Marchand, Lilian Bossuet
Laboratoire Hubert Curien, UMR CNRS 5516
University of Lyon
Saint-Etienne, France
{cedric.marchand,lilian.bossuet}@univ-st-etienne.fr

Abdelkarim Cherkaoui
TIMA laboratory
Grenoble, France
abdelkarim.cherkaoui@imag.fr

***Abstract*—Physical unclonable functions (PUF) are a promising approach in design for trust and security. A PUF derives a unique identifier for different similar dies using some of their physical characteristics, so it can be used to authenticate chips and to fight against counterfeiting and theft of devices. The transient effect ring oscillator (TERO) PUF is based on the extraction of the entropy of the process variations by comparison between TERO cells characteristics. This TERO cells need to be carefully designed in order to construct a PUF. This task need to be done with precision especially in the size of used gates and in the delays of all connections inside the element which is particularly challenging in FPGA. This paper presents the design of TERO cells in two FPGA families: Xilinx Spartan 6 and Altera Cyclone V. In addition, the result of the characterization of the TERO-PUF are compared for the two technologies.**

***Keywords*—Physical unclonable function, PUF design, FPGA, PUF characterization.**

I. Introduction

A physical unclonable function (PUF) is a salutary hardware [2] used to identification. It is also used to secure lightweight entity authentication [7] or to secure access control for the internet of things (IoT) [6]. Many architectures of PUFs are presented in the related literature and they are divided into groups [13]. One of these groups includes memory based PUF such as SRAM PUF [10]. Another group includes delay based PUF such as arbiter PUF [15], ring oscillator (RO) PUF [14], loop PUF [4] and RS latch PUF [9]. Many studies has shown that the RO-PUF is the best candidate for FPGAs ([16], [11], [12]). Unfortunately, the RO-PUF has a security problem: it is possible to clone it by using electromagnetic analysis. The transient effect ring oscillator (TERO) PUF has been proposed to solve this drawback [3]. The TERO-PUF is close to RO-PUF, but the TERO-PUF uses cells with transient oscillations.

In this article, the implementation of TERO cells are presented for two different FPGA families: Xilinx Spartan 6 and Altera Cyclone V. In addition, characterization results and comparison of the PUF on these two families are also presented in this paper.

The article is organized as follow : The next section presents the TERO cells and details its implementation for the two FPGA families. Section three describes the overall system used for the characterization and section four gives characterization results and compare them for the two technologies. Finally, section five summarizes and concludes the paper.

II. The TERO cell and its designs

A. The TERO Cell

This section describes the TERO cell and its characteristics. The TERO cell is a metastable structure first presented in [17]. This structure has been used as true random number generator in the first place but has also good characteristics to be used as PUF ([18], [3]). Figure 1 presents the generic structure of the TERO basic cell.

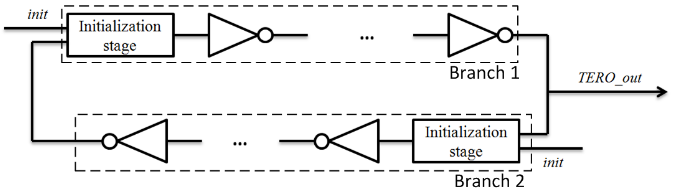

Figure 1. Generic Structure of a TERO cell

In this schematic, it is possible to see that when the cells are initialized (init at $'1'$), two events start their propagation inside the ring. These two events will move inside it until one catch up the other. That is why this ring presents only a finite number of oscillations. In theory, if all gates and all paths inside the ring are perfectly equal, the cell would oscillate infinitely but due to manufacturing process variations, this is an extremely rare case. In addition to this, the number of transient oscillations strongly depend on the number of delay elements in each branch as shown in [5].

This structure has already been studied as a prototype PUF in [3]. In this work, authors use Altera Cyclone III FPGA. In this article, two other FPGA family devices are used and the TERO cell need to be specially designed for these new technologies. Indeed, a PUF derives identifiers or keys from physical characteristics of each device so for each new technology, the basic element used to create the PUF needs to be reimplemented.

Let us review which are the parameters involved in the design of the TERO cell. First, the odd number of elements need to be the same in the two branches (figure 1). Secondly, the two branches need to be symmetric which means that all paths in the two branches have to be pairwise identical. These two first conditions are quite easy to guaranty since it is possible to constrain the placement of each elements involved in the cell. The last condition is that the two paths connecting the two branches of the TERO cell have to be of the same length. This is more difficult to achieve, especially for FPGAs and such custom designs need to be done by hand.

B. Design on Xilinx Spartan 6 FPGA

Xilinx Spartan 6 FPGAs ($45nm$ CMOS) are composed of an array of configurable logic blocks (CLB). each CLB contains two elements called slices. There are three types of slices respectively called slice_L, slice_M and slice_X, each slices contains four Look up tables (LUT) with 6 inputs and 2 outputs. To implement the TERO cell inside this FPGA technology, only slices_X will be used because they represent 50% of the FPGA. In addition, using the library of Xilinx components, it is possible to use LUTs directly in VHDL files by instantiating LUT6 component. The LUT6 component specifications are described in [1].

According to the properties of the TERO cells and to the structure of Xilinx Spartan 6 FPGA, the design need to follow some constraints. First, a LUT can be used for one and only one gate. Then, it is necessary to use the minimum number of slices allowing the cross paths to be equal. So, the first choice is to use four slices to implement one TERO cell. Furthermore, using four slices allows the designer to create TERO cell with 1, 3, 5 or 7 elements per branch. In this paper, 7 elements are implemented per branch. In the Figure 2, the version with one inverters per branch is shown with a simplified schematic.

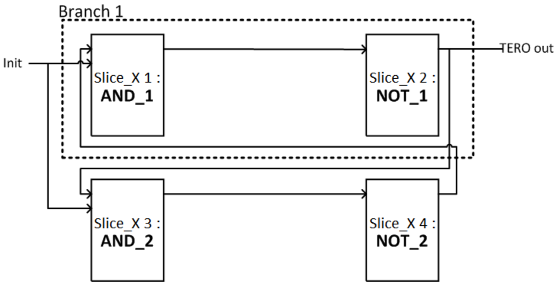

Figure 2. Simplified schematic of the first design of the TERO cell with one inverter by branch

In order to create the and gate between the inputs $I0$ and $I5$ using the LUT6 component, the initialization vector needs to be set to x"AAAAAAAA00000000". To create the function $Not(I5)$, this vector needs to be set to x"00000000FFFFFFFF". The final step to finalize the cell is to force the placement of each elements of the ring. To do it, a user file constraints (ucf) is created and each element placed according to the figure 5. To see the design, the software FGPA editor is used. This software also provides an estimation of the delays between each element. According to this Xilinx tool, all paths of the presented structure are pairwise identical (Table I). So all properties described in section II-A are respected.

It is now possible to add the rest of the elements to create the final design of the TERO cell which has 7 inverters per branch. Once this final design is done, it is really interesting for practical uses to create a hard macro of the cell. Indeed, a hard macro is an object which can be used as component inside vhdl files and more importantly, this component is never changed during optimization phase. Xilinx tool for synthesis see hard macros as black boxes which are just replicated around one reference component placed in the ucf file. The advantage of this method is that it is possible to copy and paste the TERO cell all over the FPGA.

Table I. LENGTH OF TERO CELL PATH ACCORDING TO XILINX SOFTWARE

From	To	Delay (ns)
AND_1	NOT_1	0.595
NOT_1	AND_2	0.378
AND_2	NOT_2	0.595
NOT_2	ANd_1	0.378

C. Design on Altera Cyclone V FPGA

The structure of Altera Cyclone V FPGAs ($28nm$ CMOS) is completely different from Xilinx Spartan 6 FPGAs. Indeed, Altera Cyclone V FPGAs are composed of an array of logic array blocks (LAB). Each LAB contains ten adaptive logic modules (ALM) which contains two LUT with 6 inputs and 2 outputs. It is possible to implement LUT inside vhdl files but using to component taken from the Altera component library. One component is Lut_input and represents one input of a LUT, the other is Lut_output and represents one output. The logical equation inside the LUT is between these elements. Unfortunately, Altera synthesis tool always optimize logic function and merges some LUTs together even if constraints are set. To overcome this problem, it exist one delay element called LCELL which is not optimised by the tool. According to this, the TERO cell is slightly different for Altera Cyclone V FPGAs related to the figure 1. Indeed, only one inverter per branch is used and delays elements are added between the and and the not as it is shown in figure 3.

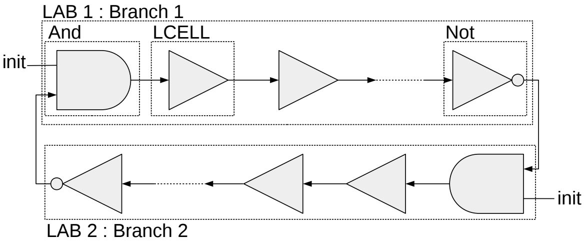

Figure 3. TERO cells for Altera Cyclone V FPGAs

This cell has only one inverter but 6 delay elements, so the total number of delays per branch is the same as the one of the TERO cell implemented in Xilinx Spartan 6 FGPAs. In addition two other LCELLs are added and merged with the and gate and the not gate. Once the TERO loop is designed in vhdl, it needs to be placed in such a way that all delays are pairwise equal. The first idea to design the cell is to try to use only one LAB. However, after testing all possible configurations of the TERO elements, the two branches still have a non negligible difference of their delays (more than $1ns$). Thus, two LABs need to be used. These two LABs can be placed side by side or one above the other.

After testing possible placements using two LABs, the best configuration gives a total difference of $0.035ns$ as determined using the Timing quest analyser. This configuration is shown in figure 4. It uses two LABs side by side with a particular arrangement of the TERO elements inside the two LABs. In figure 4, each square corresponds to one LUT and each row to one ALM. Because the And and Not gates are merged with LCELLs, they use two LUTs in this design.

LAB 1		LAB 2	
LCELL 5	LCELL 6	LCELL 5	LCELL 6
NOT		NOT	
AND		AND	
LCELL 1	LCELL 2	LCELL 1	LCELL 2
LCELL 3	LCELL 4	LCELL 3	LCELL 4

Figure 4. Final configuration of the design of the TERO cell for Altera Cyclone V FPGAs

Once the cell is designed, it needs to be placed inside the FPGA. To do this, a Quartus II setting file (qsf) is used and each element of the 256 TERO cells is placed in this file. Finally, a logic lock region is created to be sure that no logic is placed in TERO LABs by the tool during the synthesis flow.

III. TERO PUF SYSTEM

A. TERO PUF architecture

In order to compare the TERO-PUF between different technologies, it is very important to use a common framework. In this paper, the Evariste II system presented in [8] is used. In this system, users can add their own design inside of two blocks of the project. The first block is the interface between the common part and the application, the second part is the application. In this study, only the TERO cells are different depending on the technology. The interface which controls the TERO-PUF and the application stay the same for both Xilinx Spartan 6 and Altera Cyclone V FPGAs. The global TERO-PUF system presented in figure 5 does not show the whole Evariste II system but just the TERO-PUF part.

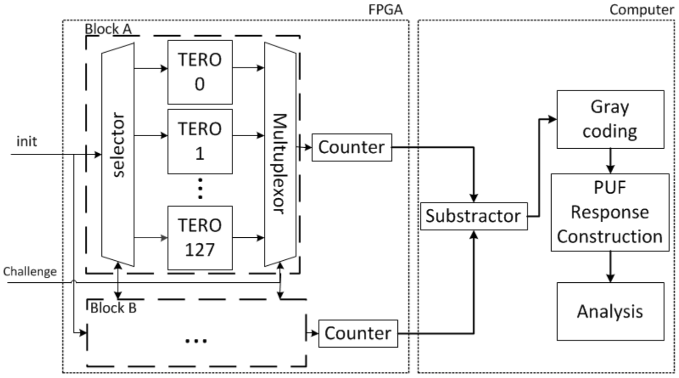

Figure 5. Hardware/software architecture of the TERO-PUF FPGA used for the characterization

The first important thing is the separation of the TERO cells into two blocks because of security. Indeed, without doing this, first order dependencies can appear inside generated signature depending on which cells contribute to the response. The second is that for characterization purpose, subtraction, Gray coding and signature generation are computed in software. The hardware part gives only the numbers of oscillations of two TERO cells. Finally, it is possible to configure the time of acquisition which corresponds to the time the cells are able to oscillate.

Each block of TERO cells contains exactly 128 cells for this characterization. Thus, the number of possible challenges (pairs of TERO cells) is $128 * 128 = 16384$ and the number of completely independent set of 128 challenges is 128. Nevertheless, it is possible to generated more signatures but they will have some common subset of challenges, considering that, the total number of all possible signatures corresponds to the number of bijections of a set in himself.

IV. COMPARISON OF CHARACTERIZATION RESULTS

This section compares the result of the characterization of the TERO-PUF implementation on Xilinx Spartan 6 and Altera Cyclone V FPGAs. First of all, let us describe what are the metrics used to analyse the TERO-PUF responses. All responses are generated using one, two or three bits of the difference between the number of oscillations returned by two TERO cells which are oscillating at the same time (a challenge). This difference between the number of oscillations is Gray coded and the used bits are finally concatenated. In addition, to generate a signature of 128 bits, 128 challenges are needed if only one bit is used, 64 challenges are needed if two bits are used, and so on.

To analyse the responses, the steadiness and the uniqueness are computed. The best result for steadiness is 0% and the best result for the uniqueness is 50%. These metrics are used in the same way as in [5]. The characterization is done at nominal voltage for temperature variations and at 25°C for voltage variations. Furthermore, a steadiness constraint is set to 10% in order to create utilization range in temperature and voltage. The last parameter of this characterization is the acquisition time, which is set to $0.67\mu s$. This parameter has a huge impact on the steadiness of the responses but not on their uniqueness. Nevertheless, setting this time too short will result in stopping all TERO cells and the advantage of transient oscillations will be lost.

All results are summarized in Table II. They are taken over 30 FPGAs for Xilinx Spartan 6 (Sp 6) and 18 FPGAs for Aletra Cyclone V (Cyc V) FPGAs. It is possible to the remark that results at nominal temperature and voltage are similar in both steadiness and uniqueness for the two FPGA families. Indeed, even if more than one bit is used to generate the PUF responses, results stay very close. In addition, the voltage ranges encompass the entire FPGAs specifications using one or two bits to generate the 128 bits signatures for Xilinx Spartan 6 and for Altera Cyclone V. Temperature variations characterization are available only for Xilinx FPGAs and they give a steadiness bellow 10% between 2 and 70°C using two bits of the differences. The fact that the TERO PUF can extract more than one bit per challenge is a great improvement for PUF. Indeed, only two block of 64 cells are needed two build 128 bits signatures using two bits and less if three bits are used.

Table II. Comparison of the results of the characterization of the TERO-PUF on Xilinx Spartan 6 and Altera Cyclone V FPGAs

response bit	Steadiness mean		Uniqueness		range	T°range		Voltage range	
per challenge	Sp 6	Cyc V	Sp 6	Cyc V	constraint	Sp 6	Cyc V	Sp 6	Cyc V
1	2.63%	1.80%	48.46%	47.62%	10%	2°C to 70°C	na	1.10V to 1.27V	1,05V to 1,15V
2	2.36%	2.66%	47.22%	48.58%	10%	2°C to 70°C	na	1.14V to 1.27V	1,05V to 1,15V
3	3.56%	3.73%	45.52%	47.39%	10%	5°C to 48°C	na	1.16V to 1.25V	1,06V to 1,13V

V. Conclusion

In this paper, TERO PUF implementations are proposed for two different FPGA families, Xilinx Spartan 6 and Altera Cyclone V. These implementations are described at the lower lever accessible for both families and the TERO cell for Xilinx FPGAs is, according to the design tool, fully balanced. For Altera FPGAs a little difference remains between the two branches of the TERO cell. In addition, characterization results are given for the two implementations. The characterization is described and has been done using the exact same set up and the exact same global system. This make the characterization fair. The comparable results prove that the TERO-PUF is reliable and not very sensitive to temperature and voltage variations. It is a promising PUF for authentication.

References

[1] "http://www.xilinx.com/support/documentation/sw_manuals/xilinx13_1/spartan6_hdl.pdf."

[2] L. Bossuet and D. Hely, "Salware: Salutary hardware to design trusted ic." in *Workshop on Trustworthy Manufacturing and Utilization of Secure Devices, (TRUDEVICE)*, 2013.

[3] L. Bossuet, X. T. Ngo, Z. Cherif, and V. Fischer, "A puf based on a transient effect ring oscillator and insensitive to locking phenomenon," *IEEE Transactions on Emerging Topics in Computing*, 2014.

[4] Z. Cherif, J.-L. Danger, S. Guilley, and L. Bossuet, "An easy-to-design puf based on a single oscillator: The loop puf," in *Euromicro Conference on Digital System Design (DSD)*, 2012.

[5] A. Cherkaoui, L. Bossuet, and C. Marchand, "Design, evaluation and optimization of physical unclonable functions based on transient effect ring oscillators," Cryptology ePrint Archive, Report 2015/623, 2015.

[6] A. Cherkaoui, L. Bossuet, L. Seitz, G. Selander, and R. Borgaonkar, "New paradigms for access control in constrained environments," in *9th International Symposium on Reconfigurable and Communication Centric Systems-on-Chip (ReCoSoC)*, 2014.

[7] J. Delvaux, D. Gu, D. Schellekens, and I. Verbauwhede, "Secure lightweight entity authentication with strong pufs: Mission impossible?" *Cryptographic Hardware and Embedded Systems (CHES)*, 2014.

[8] V. Fischer, F. Bernard, and P. Haddad, "An open-source multi-fpga modular system for fair benchmarking of true random number generators," in *Field Programmable Logic and Applications (FPL), 2013 23rd International Conference on*, 2013.

[9] B. Habib, J. Kaps, and K. Gaj, "Efficient sr-latch PUF," in *Applied Reconfigurable Computing - 11th International Symposium, ARC 2015, Bochum, Germany, April 13-17, 2015, Proceedings*, 2015, pp. 205–216.

[10] D. E. Holcomb, W. P. Burleson, and K. Fu, "Power-up sram state as an identifying fingerprint and source of true random numbers," *IEEE Transactions on Computers*, 2009.

[11] A. Maiti, J. Casarona, L. McHale, and P. Schaumont, "A large scale characterization of RO-PUF," in *In Proc. of the 2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, 2010.

[12] A. Maiti, V. Gunreddy, and P. Schaumont, "A framework for the evaluation of physical unclonable functions," *in Proc. of NIST Work. on Crypto. For Emerging Tech. and Appl.*, 2011.

[13] ——, "A systematic method to evaluate and compare the performance of physical unclonable functions," *IACR Cryptology ePrint Archive*, 2011.

[14] A. Maiti and P. Schaumont, "Improved ring oscillator puf: An fpga-friendly secure primitive," *Journal of Cryptology*, 2011.

[15] M. Majzoobi, F. Koushanfar, and M. Potkonjak, "Lightweight secure pufs," in *In Proc. of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2008.

[16] S. Morozov, A. Maiti, and P. Schaumont, "An analysis of delay based PUF implementations on FPGA," in *In Proc. of the 6th International Symposium on Reconfigurable Computing: Architectures, Tools and Applications (ARC)*, 2010.

[17] L. M. Reyneri, D. D. Corso, and B. Sacco, "Oscillatory metastability in homogeneous and inhomogeneous flip-flops," *IEEE Journal of Solid-State Circuits*, 1990.

[18] M. Varchola, M. Drutarovský, and V. Fischer, "New universal element with integrated puf and trng capability," in *International Conference on Reconfigurable Computing and FPGAs (ReConFig)*, 2013.