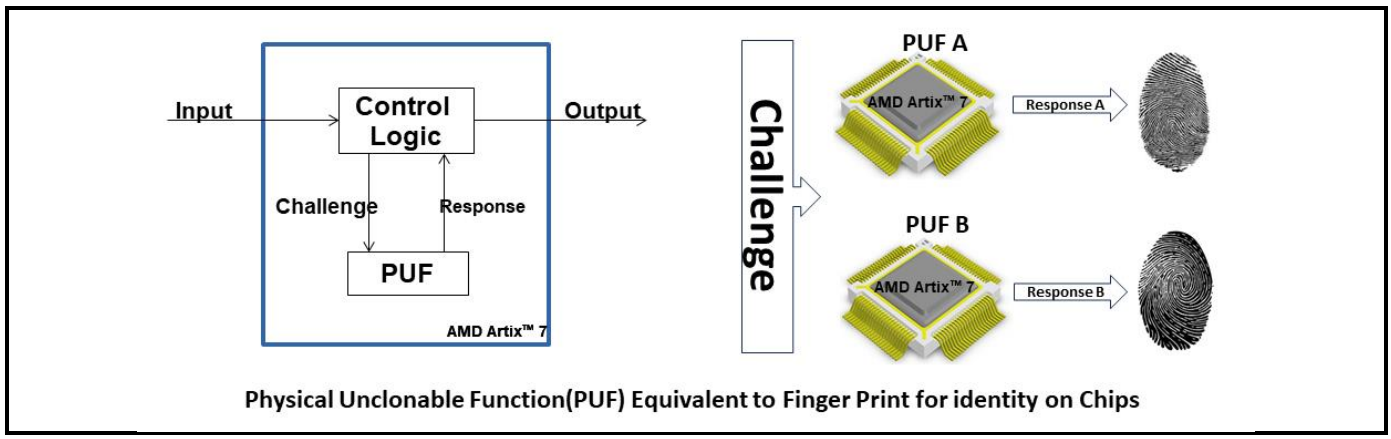
Design of Pseudo-Linear Feedback Shift Register (LFSR) based physical unclonable function

Jeeru Dinesh Reddy [1], Ajaykumar Devarapalli [2*], Radha RC [3], K Panduranga Vittal [4]
[1,2,3] *Department of ECE, B.M.S. College of Engineering, Bangalore, India.*
[4] *Department of EEE, NITK, Surathkal, India.*



Physical Unclonable Function(PUF) Equivalent to Finger Print for identity on Chips

ABSTRACT: Physical Unclonable Functions are used for authenticating hardware devices. This paper discusses implementing a pseudo-linear feedback shift register-based Physical Unclonable Function on an Artix-7 device within the Basys-3 Field Programmable Gate Array development board. The primary goal is to create an area-efficient Physical Unclonable Function that generates more challenge-response pairs than conventional pseudo-linear feedback shift register-based designs. The design relies on a linear feedback shift register but uses combinational circuits such as inverters and XOR gates instead of shift registers. The strength of the Physical Unclonable Function is based on the quantity of challenge-response pairs it generates, with a larger set indicating better security. The proposed design produces a large-bit response within a stipulated area, capturing n bits of response from a single n-bit challenge. Additionally, the mapping of challenge and response pairs can be varied without altering the hardware structure. Typically, Physical Unclonable Functions are implemented on Field Programmable Gate Arrays and Application Specific Integrated Circuits. This study details the design of a robust linear feedback shift register-based Physical Unclonable Function and how the modified design increases challenge-response pairs within a given area on Field Programmable Gate Array fabric.

Keywords: Physical Unclonable Function, Strong PUF, Weak PUF, CRPs, Constraints.

تصميم دالة مادية غير قابلة للاستنساخ قائمة على مسجل إزاحة التغذية الراجعة شبه الخطي

جيرو دينيش ريدي، اجاي كومار ديفارابالي*، رادا ر س، و باندوراجنا فيتال

الملخص: تُستخدم الوظائف الفيزيائية غير القابلة للاستنساخ لتوثيق الأجهزة المادية. يناقش هذا البحث تنفيذ وظيفة فيزيائية غير قابلة للاستنساخ تعتمد على سجل الإزاحة بتغذية راجعة شبه خطية على جهاز أرتيكس-7 ضمن لوحة تطوير مصفوفة البوابات المنطقية القابلة للبرمجة باسيس-3. الهدف الأساسي هو إنشاء وظيفة فيزيائية غير قابلة للاستنساخ ذات كفاءة مساحية، قادرة على إنتاج عدد أكبر من أزواج التحدي والاستجابة مقارنة بالتصميمات التقليدية التي تعتمد على سجل الإزاحة بتغذية راجعة شبه خطية. يعتمد التصميم على سجل إزاحة بتغذية راجعة خطية ولكنه يستخدم دوائر تركيبية مثل العواكس وبوابات الحصر XOR بدلاً من سجلات الإزاحة .تعتمد قوة الوظيفة الفيزيائية غير القابلة للاستنساخ على كمية أزواج التحدي والاستجابة التي تنتجها، حيث يشير العدد الأكبر إلى أمان أعلى. يتيح التصميم المقترح إنتاج استجابة ذات عدد كبير من البتات ضمن مساحة محددة، حيث يتم استخراج عدد **n** من البتات من الاستجابة من تحدٍ يحتوي على عدد **n** من البتات. بالإضافة إلى ذلك، يمكن تغيير خريطة أزواج التحدي والاستجابة دون تعديل هيكل الجهاز .عادةً ما يتم تنفيذ الوظائف الفيزيائية غير القابلة للاستنساخ على مصفوفات البوابات المنطقية القابلة للبرمجة أو الدوائر المتكاملة المخصصة. يشرح هذا البحث بالتفصيل تصميم وظيفة فيزيائية غير قابلة للاستنساخ تعتمد على سجل الإزاحة بتغذية راجعة خطية، ويوضح كيف يُساهم التصميم المعدل في زيادة أزواج التحدي والاستجابة ضمن مساحة معينة على مصفوفة البوابات المنطقية القابلة للبرمجة.

الكلمات المفتاحية: وظيفة فيزيائية غير قابلة للاستنساخ; وظيفة فيزيائية غير قابلة للاستنساخ قوية; وظيفة فيزيائية غير قابلة للاستنساخ ضعيفة; أزواج التحدي والاستجابة; قيود.

Corresponding author's e-mail: d.ajay402@gmail.com

DOI:10.53540/tjer.vol.21iss2pp97-104

1. INTRODUCTION

Field Programmable Gate Arrays (FPGA) have several applications in embedded system development because of their features like configurability and low cost as compared with Application Specific Integrated Circuits (ASIC). FPGA vendors provide security solutions to designers to protect sensitive data and intellectual properties, such as encrypting bitstreams, authenticating bitstreams, and protecting key memories. The keys used for encryption are stored in Non-volatile memories using Electrically erasable Programmable Read-only memory (EEPROM), Flash, battery-backed Static Random Access Memory (SRAM), etc. However, the major disadvantage of using these memory technologies is that they do not provide any guarantee about the security of sensitive information and keys. These keys are easily found in Non-volatile memory through physical attacks. For some of the FPGA applications, there are very limited hardware resources, and thus it will increase the hardware overhead by integrating all the security modules, which may lead to problems (Hou et al. 2019). Because of this reason the development of new lightweight hardware security modules and providing security services like authentication and key generation for several FPGA applications has become a significant research area (Gao et al. 2016). Due to some uncontrollable reasons while manufacturing digital blocks, the parameters like size, gate oxide thickness, and the threshold voltage of each device will not be the same, there will be many random deviations occurring such as process deviations. They neither affect the functionality nor precision of the circuit. However, the process deviations are extracted using different design methods to produce a unique "fingerprint" of the circuit, such that each circuit block can be identified accurately and prevent the chip and circuit from being altered. This structure is known as a Physical Unclonable Function (PUF). It has many advantages and various applications such as cryptography and hardware security.

As referred by (Suh et al 2007) Gassend and Pappu developed the first silicon PUFs in 2001 using intrinsic process variations in deep submicron integrated circuits. During manufacturing, they used intrinsic process variations of silicon devices to create unique arbitrary and unclonable responses and called them a physically random function. This non-repeatable physical circuit is called a PUF. PUFs should be unpredictable, unclonable, and robust.

A PUF provides a unique "digital fingerprint" of an integrated circuit based on a hardware device. These PUFs are used for security applications like secured access and authentication.

A challenge is an input given to the core logic of PUF and the output which is obtained from the core logic is called a response. These sets of inputs and outputs are termed Challenge-Response Pairs (CRPs) Since these PUFs provide many CRPs, they are regarded as a multiple-input (challenges) multiple-output (responses) function. This property makes it hard to predict the dependency between challenge-response pairs. The relationship between input and output appears like a random function. Since the PUF is derived from random process deviations, it is impossible to guess the response from the given challenge.

2. PHYSICAL UNCLONABLE FUNCTION(PUF)

PUFs are defined as physical circuits that are embedded in hardware devices and extract secrets from the physical features of an integrated circuit (IC). A PUF is defined as a "digital fingerprint" that can be used as a unique for semiconductor devices such as a microprocessor. They are based on physical variations that naturally occur during the manufacturing of semiconductor devices. PUFs are usually implemented in ICs and are typically used in applications such as cryptography which needs high security requirements as mentioned in (Guajardo et al. 2007).

The method of obtaining a response from the given challenge is known as the "Challenge-Response" mechanism. The set of challenge-response pairs is called CRP space. Because of this mechanism (Zhang et al. 2018), private information and security keys are produced in real-time without storing them in local memories which are easily accessed, further reducing the probability of the key being visible to attackers.

Based on the strength of PUF, they are mainly classified as weak PUF and strong PUF. The strength of PUF can be defined as the total number of CRP pairs it can provide. One of the most important criteria of this PUF implementation is that, when it is targeted on FPGA, PUF is constrained by the resources on FPGA, so area efficiency is a very important concern in such designs. Conventional Pseudo-LFSR PUF core that has a single select line, is now modified to achieve.

2.1. Strong PUF

Strong PUFs have a large set of CRPs. So even if an attacker has access to PUF, they cannot capture all the CRP pairs as the number is very large. These kinds of PUFs are more appropriate for authentication purposes (Hori et al. 2010). If some of the CRPs are taken randomly, the chances of an attacker recording the responses to the corresponding challenges are negligible. From this, it is clear that even if an attacker has access, only a user who has physical access to the PUF can deliver

the correct response and thus the designed PUF is said to be authenticated. In addition, many challenge-response pairs mean that each CRP should be used only once for authentication of the device. This mechanism protects an attacker against recording CRP and can provide safe communication using PUF.

2.2. Weak PUF

Weak PUFs comparatively provide very few challenge-response pairs. The main feature that distinguishes weak PUF from strong PUF is that it usually has only one challenge per PUF instance whereas strong PUF has many challenges. These are mainly used for authentication and key storage.

3. PSEUDO-LFSR PUF (PL-PUF)

The PUFs that are delay-based generally produce one-bit or many-bit response as an output at once and subsequently have low throughput. Whereas PUFs which are based on memory, produce several bits of output simultaneously but the values of output are fixed and PUFs which are addressable memory-based produce variable IDs as an output, but the size of the circuit becomes very large.

To overcome these disadvantages, they came up with another type of PUF named PL-PUF. Unlike the PUF that's referred to in (Ogasahara et al. 2016). The design of this PUF is dependent on LFSR but does not contain any shift register rather includes a combinational logic circuit. The PL-PUF produces a long-bit response as an output which is variable in nature. It adds up non-linearity in the structure in increases the strength of the PUF.

The structure of 128-bit Pseudo-LFSR PUF is shown in Figure 1. The derived primitive feedback polynomial of the proposed PUF is represented in equation (1) concerning the application note (George et al 2007).

$$x^{128}+x^{126}+x^{101}+x^{99}+1 \quad (1)$$

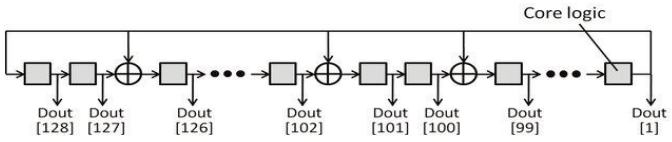


Figure 1. Pseudo-LFSR PUF(128-bit).

The core logic block of PL-PUF doesn't have a register but consists of an inverter, and thus PLPUF includes a large combinational circuit. Since the output of the last core logic (Dout [1]) is given as input to the first core logic (Dout [128]), the final PL-PUF output contains oscillations. The output of PL-PUF is based on the speed of the feedback signal, and this is affected by variations in the device. Thus, the PL-PUF output is dependent on the device. The core logic can be any combinational circuit that can extract the device variations. The core logic need not always be an inverter.

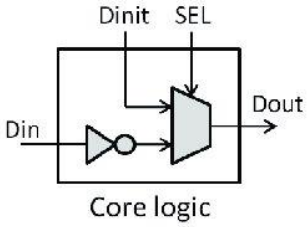


Figure 2. Core logic.

The 128-bit PL-PUF structure is shown in Figure 1. This structure of PL-PUF consists of 128 inverters and 3 XOR gates. The core logic has an inverter and a multiplexer (Figure 2). The following are the steps to obtain a response when the challenge is applied.

1. Make the "toggle" signal low.
2. Provide input to the "challenge" signal that is Dinit of each core logic.
3. Now make the "toggle" signal high to enable PUF for oscillations.
4. Note down the output of each core logic (i.e., the response bit of each core logic) when the challenge is given and store them in registers.

The PL-PUF usually realizes challenge-response pair-based authentication. The challenge is a 128-bit input value given as Dinit and the response is a 128-bit output value i.e. Dout. From this observation, we can make a note that only one 128-bit challenge is enough to produce a 128-bit response as an output. After each core logic is initialized with an initial value, this PUF is enabled for c-clock cycles. We get completely different IDs as an output for the same PL-PUF by varying the active duration of the clock cycle.

The following are the rules for the selection of feedback polynomial:

- The '1' present in the primitive feedback polynomial does not indicate tap, it signifies the input of the first bit.
- The powers of feedback polynomial indicate tapped bits, which are counted from the left. The input and output are connected to the first and last bits of the polynomial respectively.
- If the number of taps present in the polynomial is even, then LFSR is said to have a maximum

length.

- The set of taps not as pairs of elements, taken all together in the feedback polynomials should be relatively prime.

The features of PL-PUF are:

Efficiency: The proposed PL-PUF produces a 128-bit response from the single 128-bit challenge. It is a major benefit of this PL-PUF as compared with other types of PUF, whereas in other types of PUFs, several bit outputs are produced from a large bit challenge(Hori et al. 2011). For example, arbiter PUF needs 128 CRPs to acquire a 128-bit ID.

Multi-functionality: The proposed PL-PUF behaves like multiple PUFs because the output relies on the period of the clock. Accordingly, CRP mapping can be changed easily without changing its hardware architecture. This feature makes PUF unclonable because cloning all the combinations of CRPs is not possible practically.

Reliability: The reliability of PL-PUF means it should generate reproducible IDs that are unique to all devices. A good PUF should have high reproducibility for proper device authentication.

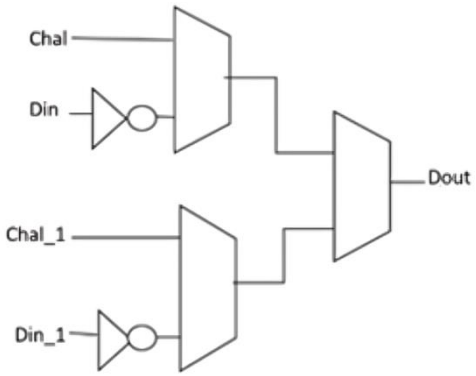


Figure 3. Modified core logic to increase the challenge-response pairs

4. PROPOSED DESIGN

The proposed work is taken from the Author's work on modelling Arbiter-based PUFs (APUFs) for attack resistance. The structure proposed in (Manchikanti Venkata et al. 2020) is on a regular multiplexer-based PUF. Here in this research, it is the modified structure of core logic as given in Figure 3, on PLPUF where it's a non-linear PUF. This structure is designed based on the structure of Basys3 Artix-7 FPGA. FPGAs are evolving with various architectures every year. Artix-7 is considered a benchmark in this work because it has the state of the art technology with a Processor System (PS) and Programmable Logic (PL). So, by utilizing the structurally enhanced architecture in Artix-7, the multiplexer structure is enhanced with additional F7AMUX, F7BMUX, and F8MUX in a modified structure. This structure improves the strict avalanche criteria (Hori et al. 2007) for Arbiter PUFs. As two 128-bit challenges are applied to the design at a time, there is an increase in the CRP pairs. This makes PUF stronger. The response bit of each core logic is stored in the register; here we are using the positive edge triggered D flip flop as a register. The elaborated design of 128-bit PL-PUF is shown in Figure 4.

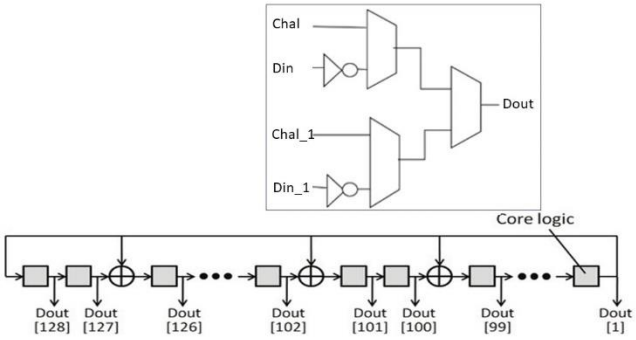


Figure 4. Design of 128-bit LFSR

5. RESULTS AND CONCLUSION

The proposed design was described using Verilog in the Xilinx Vivado tool. Symmetric placement of logic on the Look Up Tables (LUTs) inside the Configurable Logic Blocks (CLBs) is required to ensure that the process variation component is well extracted using Physical placement constraints. Xilinx Design Constraints (XDC) file or Tcl script file is used to load these constraints on the design. This process of specifying the constraints can be done using Register Transfer Logic (RTL) constraints or as a Xilinx Design Constraints (XDC) file. We can update new constraints with the help of Tcl commands in the Tcl console after the design is loaded into memory, or it can be done by using any one of the Integrated Development Environment (IDE) editing tools of Vivado Design Suite. Reference (Xilinx 2010) gives complete information about the synthesis constraints involved in this design. "KEEP_HIERARCHY" and "Create_Macro" are two important constraints that enable to retention of logical hierarchy and hard macro design respectively.

LOCK_PINS constraint is applied for mapping logical inputs of LUT (I0, I1, I2, ...) and physical input pins (A6, A5, A4, ...) of LUT. LOC constraint is used to place logical elements in a specific location on FPGA fabric and Basic element of logic (BEL) constraints are used to map the logic to specific LUTs.

The constraints BEL or LOC need to be mentioned in addition to Relative Location (RLOC) constraints. The BEL constraints should be used to align cells inside the

Relationally Placed Macro (RPM) set, for example, to align the LUTs with registers. Whenever BEL or LOC constraints must be specified, it is important not to mix the source of those constraints. These constraints must be entirely specified either through RTL or XDC constraints, but not a combination of the two.

Since the schematic of 128-bit LFSR is very large enough to fit into this report, the schematic of 3-bit LFSR-based PUF is shown in Figure 8.

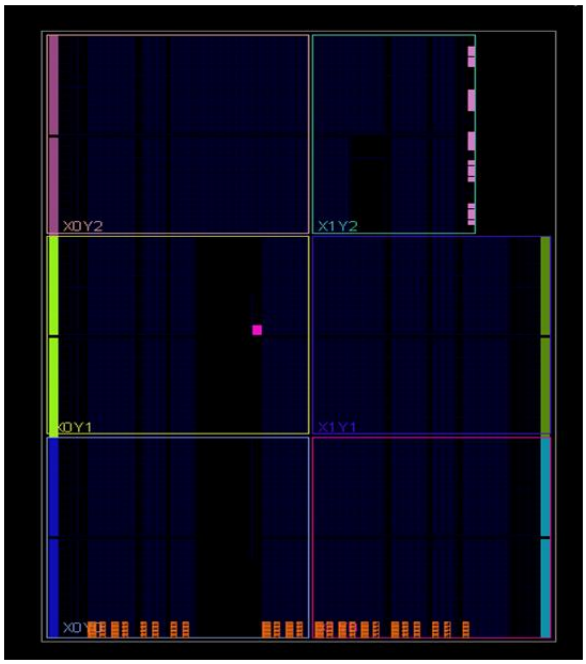


Figure. 5. Implemented Design.

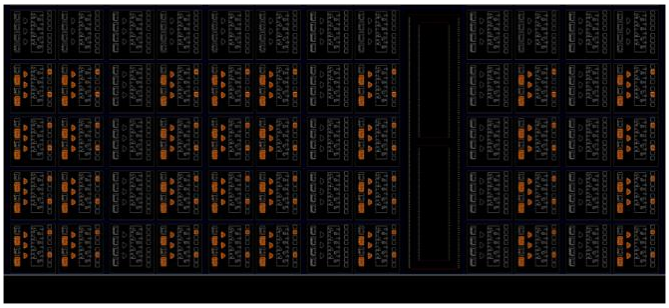


Figure. 6. Placement of 128-bit LFSR PUF.

Since the output of the last core logic is given as an input to the first core logic (feedback present in the design), the response (output) of PL-PUF contains continuous oscillations as shown in Figure 9. Every time we implement the design the placement of cells gets disturbed if we don't specify the placement constraints. Hence placement constraints are written and saved in an XDC file to secure the location of cells on the device. The implemented design of the proposed PL-PUF and the placement of 128-bit LFSR PUF is presented in Figure 5 and Figure 6 respectively. The elaborated structure of each slice is shown in Figure 7. At a time two 128-bit challenges are given as input to the device and each 128-bit challenge will produce a 128-bit response. Some of the sample challenge-response pairs are captured from the post-synthesis timing simulation as shown in Table 1. The response is captured on every positive edge of the clock when the enable signal is high.

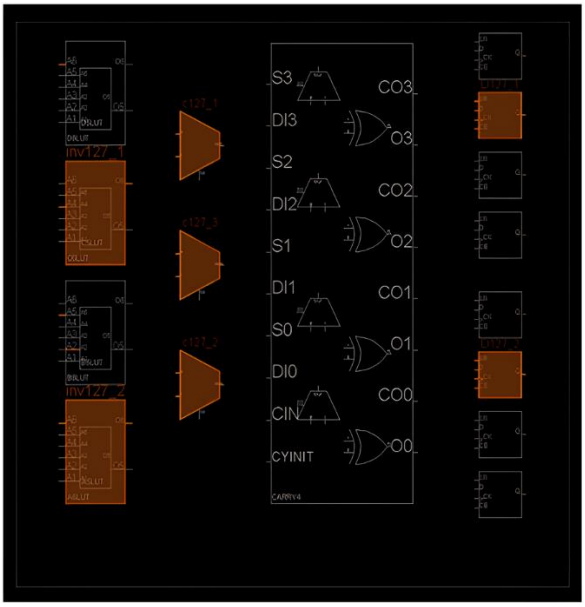


Figure 7. Structure of each SLICE.

We have designed a modified 128-bit pseudo-LFSR-based PUF to increase the challenge-response pairs. By placing the cells based on the Artix-7 architecture, two 128-bit challenges are given to the design at a time to increase the CRP pairs. In the basic design of pseudo-LFSR PUF, the number of CRPs is 2^{128}, whereas the proposed PUF has 2^{2*128} challenge-response pairs. This increase in the number of CRPs makes it hard to clone the PUF and becomes a more efficient technique to authenticate the device. Thus, even if an attacker has access, he can't record all the CRP pairs due large set of CRPs. Thus, the proposed PUF is considered a strong PUF.

The proposed design can be implemented on various FPGA devices to build device-specific challenge-response pairs. And, it is necessary to evaluate the performance parameters like uniqueness, randomness, and multi-functionality by collecting challenge-response pairs of each device as given in (Lim et al. 2005).

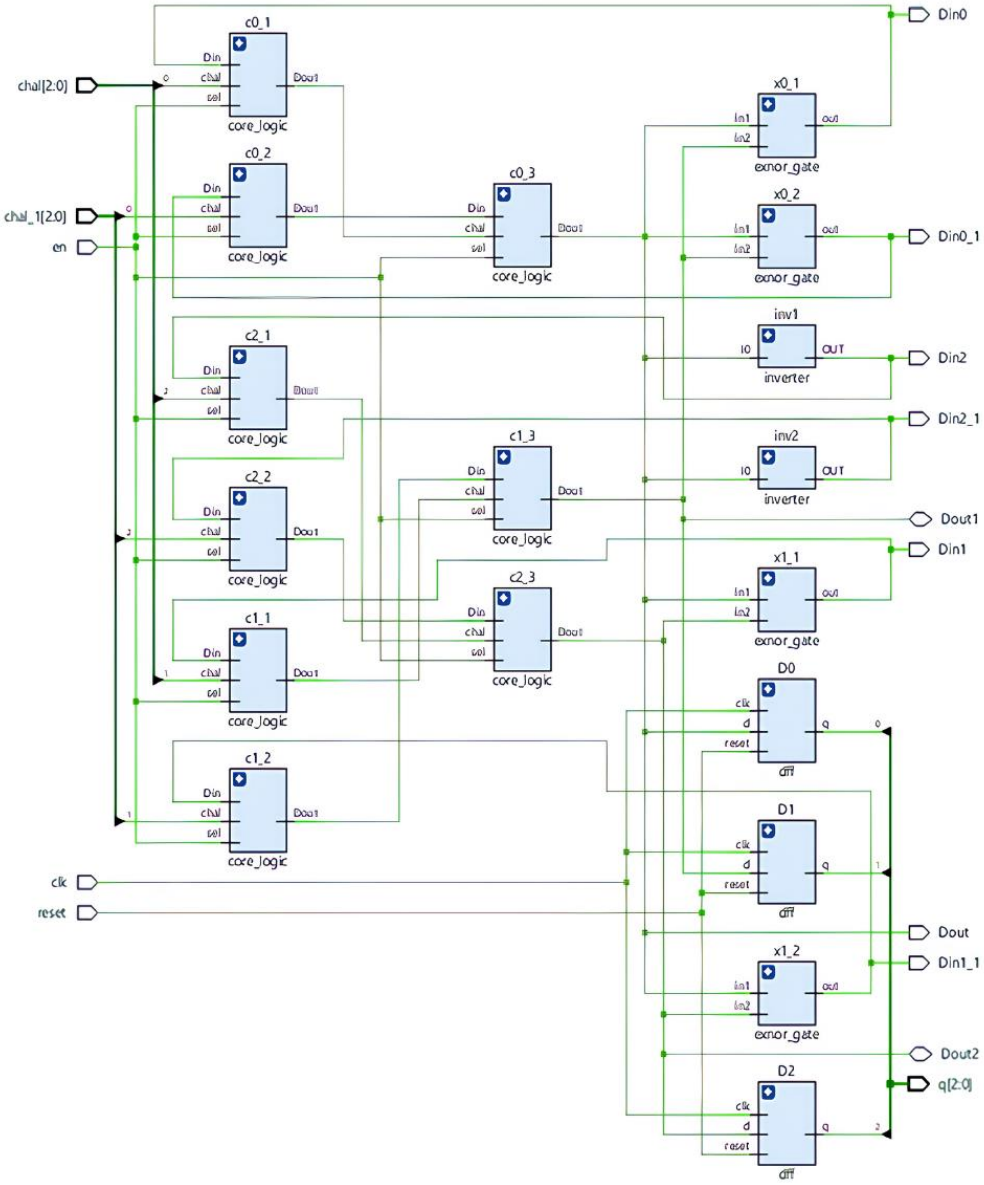


Figure 8. Schematic of 3-bit LFSR.

Figure 9. Simulation result of 128-bit LFSR PUF.

CONFLICT OF INTEREST

The authors declare that there are no conflicts of interest regarding this article.

FUNDING

The authors did not receive any funding for this research.

ACKNOWLEDGMENT

This project work is a part of research work guided by Dr. K.P. Vittal, Professor, Department of EEE, NITK. It was a great privilege to work under his guidance.

REFERENCES

Gao, Y., Ranasinghe, D. C., Al-Sarawi, S. F., Kavehei, O., & Abbott, D. (2016). Emerging physical unclonable functions with nanotechnology. IEEE Access, 4, 61-80.

George, M., & Alfke, P. (2007). Linear feedback shift register in Vertex devices. Xilinx Application Note XAPP210.

Guajardo, J., Kumar, S.S., Schrijen, GJ., Tuyls, P. (2007). FPGA Intrinsic PUFs and Their Use for IP Protection. In: Paillier, P., Verbauwhede, I. (eds) Cryptographic Hardware and Embedded Systems - CHES 2007. Lecture Notes in Computer Science, vol 4727. Springer, Berlin, Heidelberg.(63–80)

Hori, Y., Kang, H., Katashita, T., & Satoh, A. (2011). Pseudo-LFSR PUF: A compact, efficient and reliable physical unclonable function. In Proceedings of the 2011 International Conference on Reconfigurable Computing and FPGAs (223-228).

Hori, Y., Yoshida, T., Kinoshita, T., & Satoh, A. (2010). Quantitative and statistical performance evaluation of arbiter physical unclonable functions on FPGAs. In Proceedings of the 2010 International Conference on ReConFigurable Computing and FPGAs (298-303).

Hou, S., Guo, Y., & Li, S. (2019). A lightweight LFSR-based strong physical unclonable function design on FPGA. IEEE Access, 7, 64778-64787.

Lim, D., Lee, J. W., Gassend, B., Suh, G. E., van Dijk, M., & Devadas, S. (2005). Extracting secret keys from integrated circuits. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 13(10), 1200-1205.

Manchikanti Venkata, A., Jeeru, D. R., & Vittal, K. P. (2020). Design and model an attack on multiplexer-based physical unclonable function. International Journal of Engineering Trends and Technology, 68(6), 63-67.

Ogasahara, Y., Hori, Y., & Koike, H. (2016). Implementation of pseudo linear feedback shift register physical unclonable function on silicon. In Proceedings of the 2016 IEEE International Symposium on Circuits and Systems (ISCAS) (758-761).

Suh, G. E., & Devadas, S. (2007). Physical unclonable functions for device authentication and secret key generation. In Proceedings of the 44th ACM/IEEE Design Automation Conference (9-14).

Xilinx, Inc. (2010). Extended Spartan-3A FPGA family overview. Xilinx Synthesis Constraints.

Zhang, J., Tan, X., Wang, X., Yan, A., & Qin, Z. (2018). T2FA: Transparent two-factor authentication. IEEE Access, 6, 32677-32686.

APPENDIX

Table 1. Challenge Response Pairs of 128-Bit LFSR PUF.

Challenge	Response
123467589	ebc04fe01ff67fbced90a8bfffc79bff
12ab34cd56ef	2a00001b390000200f00002189372ba2
123467589	9e09c4178090a8bfffc79bffffc799b9
12ab34cd56ef	9f0882178891a8bfffc5bfffffc5bdb9
123467589	ebf74bef17fffffedb9646a2ffc01eee
12ab34cd56ef	ebfffdef1bfffffedb9646a2ffc11efa
123467589	2e47de0ae823fffeef83fffeef22591b
12ab34cd56ef	0c47fe0ae423ffffefc3ffffef6a591a
1234dcae	02490035002083000021a2cd2ba80000
abce453	004000b5012083880021a24d2ba00000
1234dcae	2b00200094000002000000248d372ba
abce453	2a0000065c40000200400002489372ba
98765abcd	d09ed5c8e9ef68d3ebae201f09fe201f
56437abcde	d09ed7e8e82f68d3caee000548fe0005
98765abcd	8a3dfff71f8824101234dcae803ffcdf
56437abcde	883cfff41f0000001234dcae803ffc87
98765abcd	8b5cf015682e201fc9fe201fc9eed37c
56437abcde	8b5cf00664ee000548fe0005482ed373
5647ad34ddec	2a0043f49b002122652ba3ac082003ac
12abcdef45987	2a0043e6ba00212a252ba3ee006003ee
5647ad34ddec	2a000273c38003801848d252e8c00002
12abcdef45987	280104e9870003801848f212e8c00002
123467589	ebc04fe01ff67fbced90a8bfffc79bff