EFFICIENT PUFs BASED ON EXTENDED HAMMING CODE

[1]N.M.Surekha, [2]S.Nanda Kishor
[1]PG Scholar, Department of ECE, Kuppam Engineering College, Kuppam, Andhra Pradesh, India.
[2]Associate professor, Department of ECE, Kuppam Engineering College, Kuppam, Andhra Pradesh, India

Abstract: **Reconfigurable systems often require secret keys to encrypt and decrypt data. Applications requiring high security commonly generate keys based on physical unclonable functions (PUFs), circuits that use random manufacturing variations to produce secret keys that are unique to each device. Implementing PUFs on field-programmable gate arrays (FPGAs) is usually difficult, because the designer has limited control over layout, and each PUF system requires a large area overhead to correct errors in the PUF response bits. The state of the art for FPGA-based weak PUFs is extended by using a novel methodology of per-device configuration and a new PUF variant derived from the popular FPGA-specific Anderson PUF. The design has several advantages over existing work including the Anderson PUF on which it is based. It is tunable to minimize the response bias and can be implemented using the common SLICEL components on Xilinx FPGAs along with an efficient per-device configuration that reduces bit error rate by over 10× at room temperature and improves response stability by over 2× across all temperatures. The Proposed PUF reduces the delay when compared to Existing PUFs. The designs are modeled in Verilog HDL and are simulated in Xilinx ISE 14.5 Tool. The functional Verification is Performed using Xilinx ISE Simulation Tool.**

***Index Terms*: Cryptography, Error Detection and Correction, Extended Hamming Code, Field Programmable Gate Arrays, Physical Unclonable Functions.**

1. INTRODUCTION

FPGAs are used for an increasingly large number of applications that require security. Due to their volatile nature, SRAM-based field-programmable gate arrays (FPGAs) require security at multiple levels. Bitstream encryption is often used to protect the configuration bits that define application implementation. Additionally, secure encrypt/decrypt cores are often implemented as part of a user's design to allow for the confidential processing of application data. These cores require secret keys that are often customized on a per-device basis. Securing these keys can be problematic.

Common SRAM-based FPGAs do not have on-chip nonvolatile or battery-backed key storage that a user can access, but if the keys are stored off-chip, then they are susceptible to the many attacks that have been demonstrated against bit stream encryption [1]. Physical unclonable functions (PUFs) represent a device tied method of generating secret keys on-chip without reliance on secured nonvolatile memory or battery-backed storage. PUF-based keys are uniquely tied to each device and are not directly compromised by attacks that are able to recover decrypted bit streams. These characteristics make PUFs well suited to key generation for FPGA-based applications. For example, PUF-based keys are currently available in reconfigurable devices, including Microsemi IGLOO2 [2] and Intel Stratix 10 [3] FPGAs, but these keys are generated by specialized blocks and not from circuitry created from the user accessible FPGA fabric.

A PUF's response should solely rely on the inherent process variation of its components. Therefore, most PUF designs rely on differential circuits in which two paths are designed with identical logic and matched routing, so that the difference only comes from process variation. This level of matching can be easily done in application-specific integrated circuits, because the designer has the freedom to easily control the layout of the PUF. However, FPGA designers are limited in this freedom and must work within the constraints of the unmovable lookup tables (LUTs) and routing tracks in the FPGA fabric. Based on the design, different sections of logic and routing resources in the FPGA can be used, leading to non identical paths. Therefore, different approaches must be considered for FPGA PUF design that considers the available resources and their abundancy. This paper details about the PUFs that are developed for cryptographic applications.

This paper is organized as section II describes Existing System and in section III discussed Proposed Method and Section IV describes in Simulation results and Section V concludes the paper followed by references.

2. EXISTING SYSTEM

Cryptographic keys must be generated repeatedly over time. The outputs of PUFs are noisy and thus cannot be used directly as key bits. Fuzzy extractors are cryptographic primitives designed for deriving reliable key values from noisy biometric data and are widely used with PUFs. When a PUF is first enrolled with a fuzzy extractor, a key is derived from the PUF and helper data are produced to facilitate generation of the same key at a later time. By using a code-offset fuzzy extractor construction with BCH codes are a family of error-correcting codes where each code is described by a tuple (n, k,t); parameter n is the block size or, equivalently, the size in bits of each codeword, parameter k is the number of information bits in each codeword, and parameter t is the number of correctable errors in each codeword.

The basic circuit of the Anderson PUF [4] is shown in Fig. 1. This PUF does not require the use of hard macros and is designed purely at the register-transfer level with the required constraints described in VHDL. The PUF uses two adjacent SLICEM blocks of a Xilinx FPGA.

In SLICEM blocks, each LUT can be used as either a 6-input combinational logic function or a 16-bit shift register, and the Anderson PUF uses the shift-register functionality. As shown in Fig. 1, the outputs of Shift reg 1 and Shift reg 2 are connected to the select lines of multiplexers that are separated by a chain of multiplexers that have their select inputs stuck at 1 as shown in fig.1.

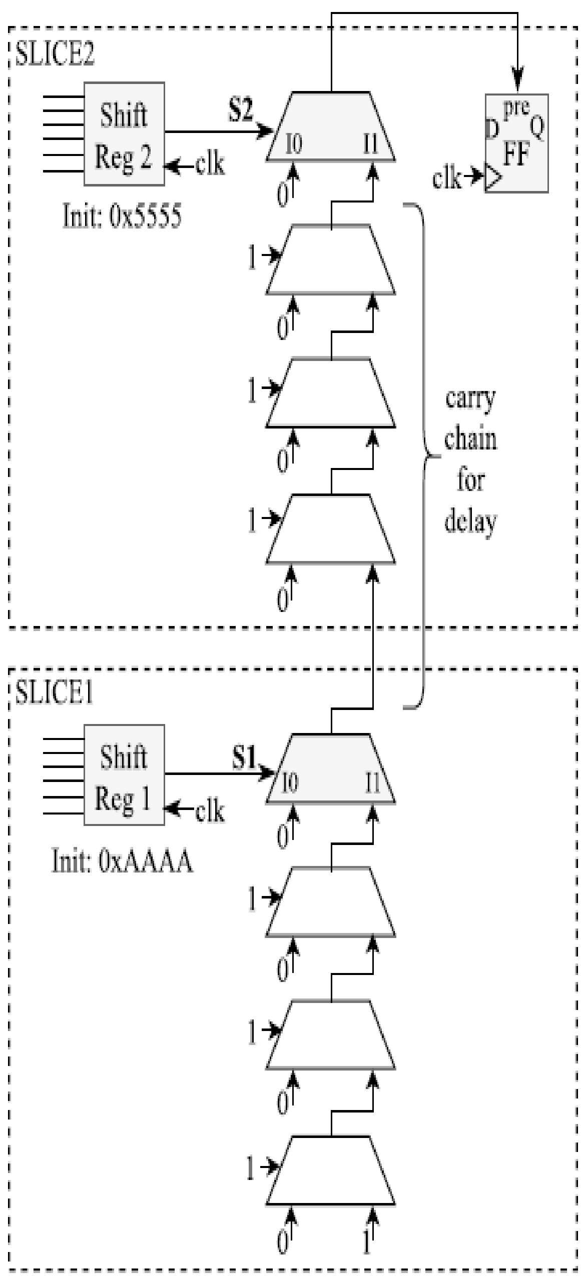

Figure 1: Anderson PUF (Existing Design - 1)

The Anderson PUF requires has the drawback of nine identical FPGAs to evaluate PUFs i.e., large area is required.

The Anderson PUF is redesigned as existing dsign-2 to improve on its limitations with modifications that allow for the creation of PUFs using LUT-based SLICEL elements that cannot be used as shift registers. These elements are more plentiful in FPGAs than the SLICEM elements that the Anderson design uses. The modification provides the capability for fine-grained PUF tuning to adjust a PUF's Hamming weight, thereby optimizing its uniqueness.

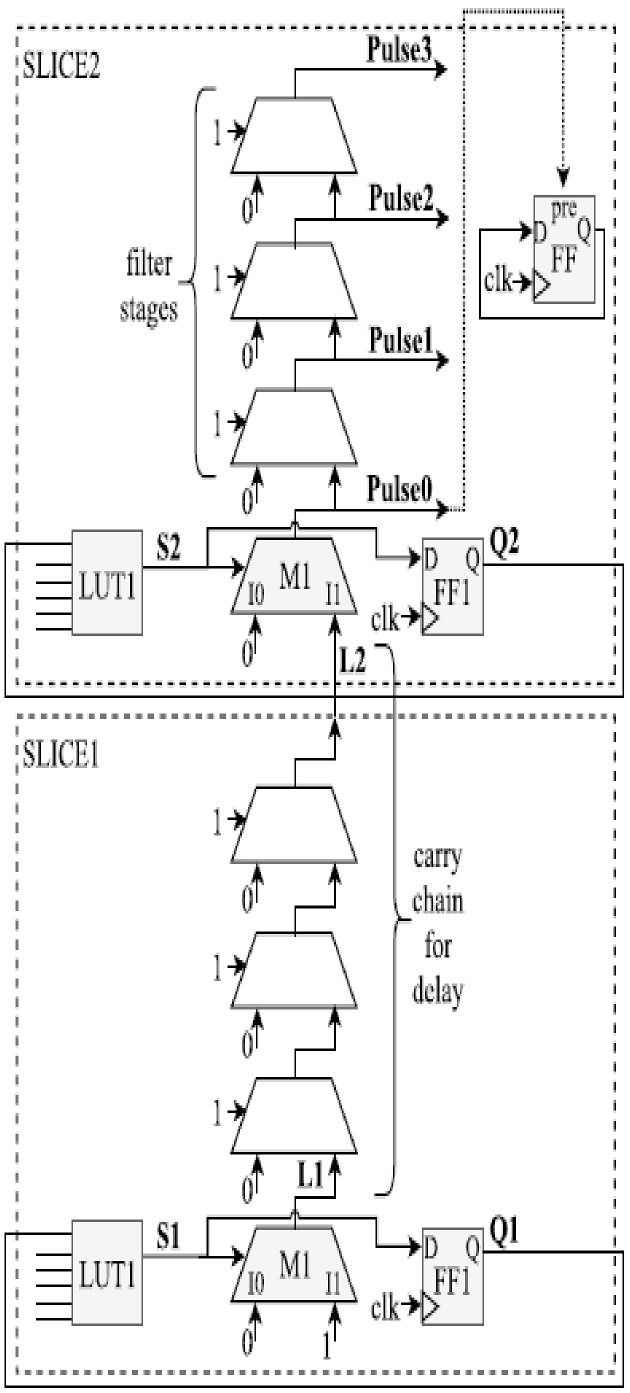

Figure 2: Redesigned Anderson PUF (Existing Design - 2)

The redesigned Anderson PUF is improvement on its limitations of existing design -1 as shown in fig. 2. But this design is Device Dependent where the mismatch in delay of feedback paths affects its performance.

3. PROPOSED METHOD

The PUFs can be developed based on Extended Hamming Code to improve reliability and Error Detection and Correction Capability. For each SLICE the internal design can be developed as shown in fig. 3.

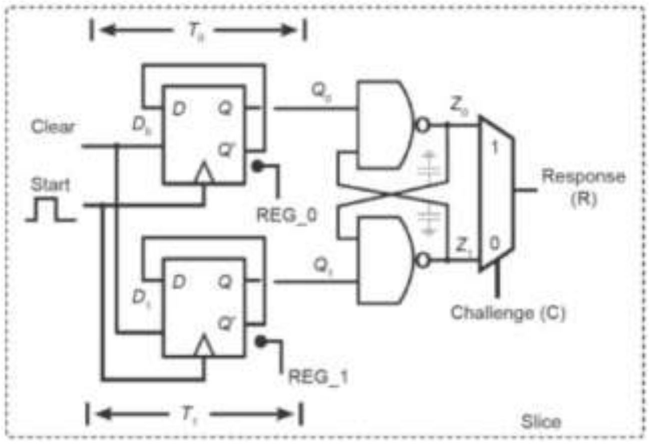

Figure 3: Proposed PUF per SLICE

The design provides advantages like error detection and correction capability, less area, etc and thereby improves reliability of the designs. Also The proposed PUF increases the Fanout for the designs and reduces the delay in acquiring the key.

4. SIMULATION RESULTS

The designs are modelled in Verilog HDL and are functionally verified by using Xilinx ISIM Simulation Tool. The designs are synthesized for Spartan3E FPGA by suing Xilinx ISE 14.5 Tools for the device XC3S500E with a package of FG320 and a speed grade of -5.

The simulation waveform for existing design -1 is shown in figure 4 where clearly the PUF operates with the designed specifications. The Existing design - 1 is converted to register transfer logic by the Xilins synthesizer as shown in figure 5 and to 90nm CMOS technology based LUT mapped schematic as shown in figure 6. The existing design - 1 extracted for FPGA along with its sample routing without imposing constraints is shown in figure 7.

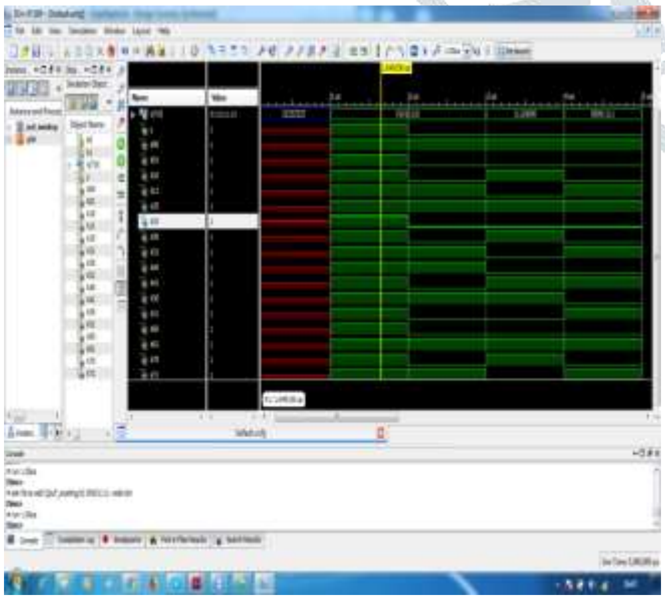

Figure 4: Simulation result of Anderson PUF

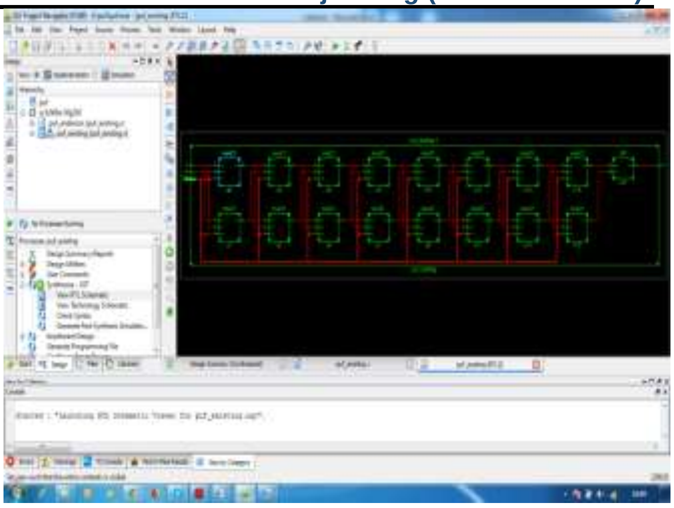

Figure 5: RTL Schematic for the Existing Design - 1.

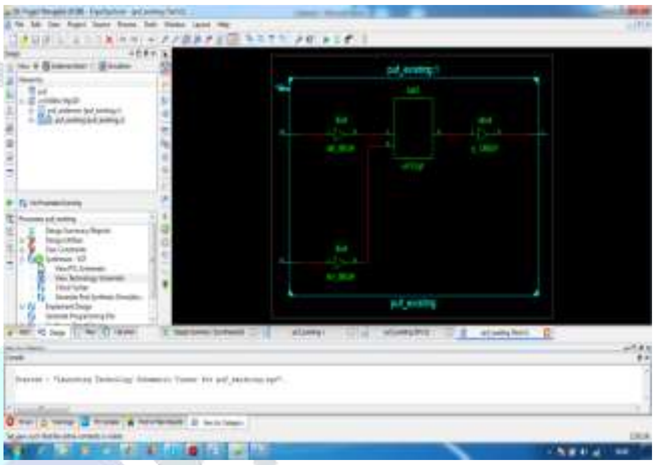

Figure 6: Technology Schematic for the Existing Design - 1.

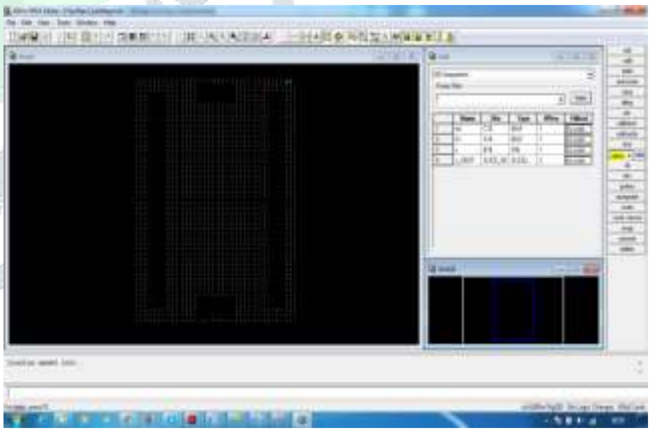

Figure 7: FPGA Implementation of the Existing Design - 1.

The simulation waveform for existing design - 2 is shown in figure 8 where clearly the PUF operates with the designed specifications. The Existing design - 2 is converted to register transfer logic by the Xilins synthesizer as shown in figure 9 and to 90nm CMOS technology based LUT mapped schematic as shown in figure 10. The existing design - 1 extracted for FPGA along with its sample routing without imposing constraints is shown in figure 11.

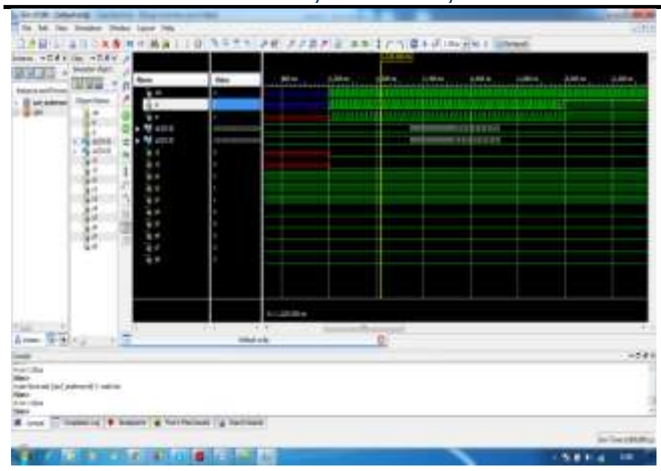

Figure 8: Simulation Result of Existing Design - 2.

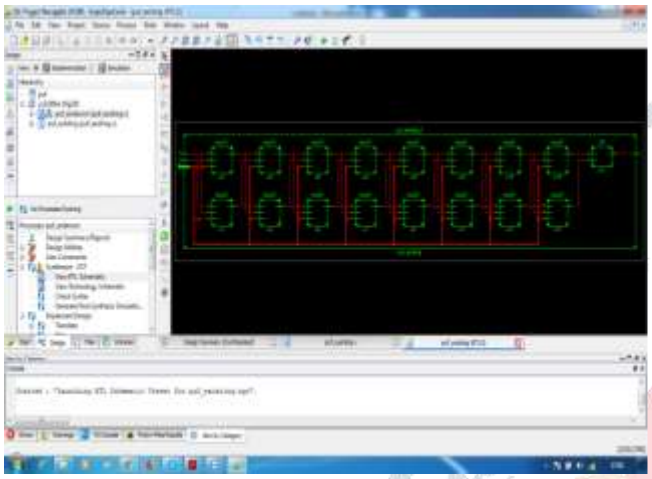

Figure 9: RTL Schematic of Existing Design - 2.

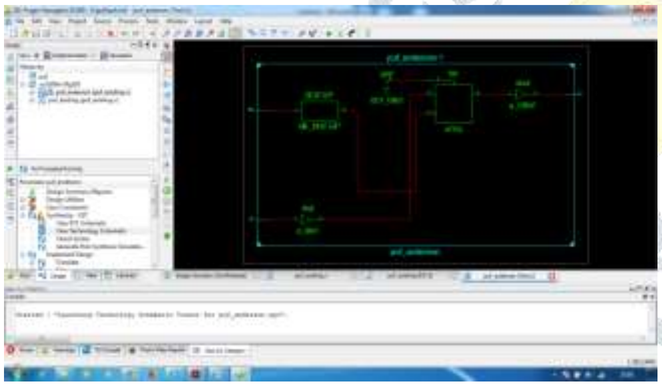

Figure 10: Technology Schematic of Existing Design - 2.

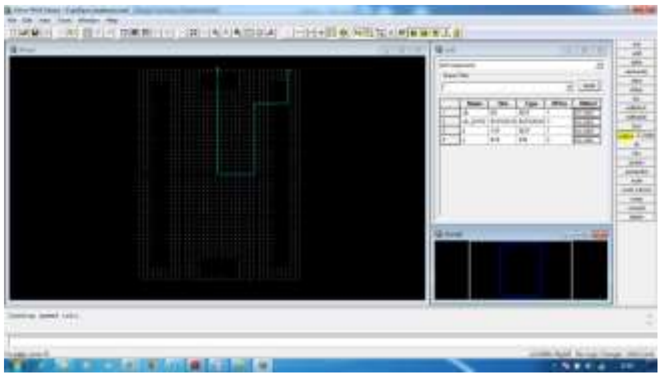

Figure 11: FPGA Implementation of Existing Design - 2.

The simulation waveform for Proposed design is shown in figure 12 where clearly the PUF operates with the designed specifications. The proposed design is converted to register transfer logic by the Xilins synthesizer as shown in figure 13 and to 90nm CMOS technology based LUT mapped schematic as shown in figure 14. The proposed design extracted for FPGA along with its sample routing without imposing constraints is shown in figure 15.

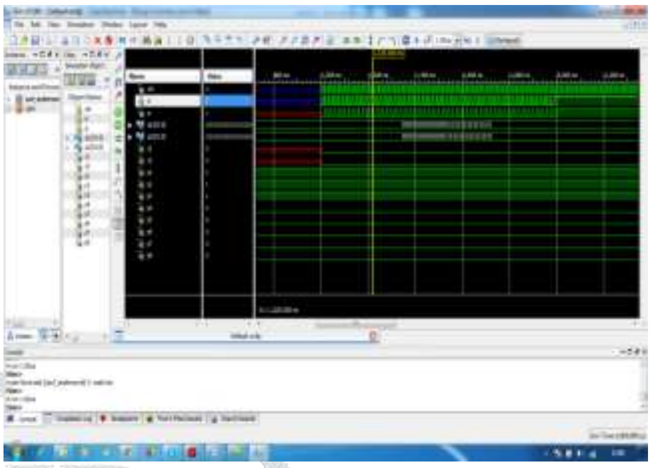

Figure 12: Simulation Result of Proposed Design

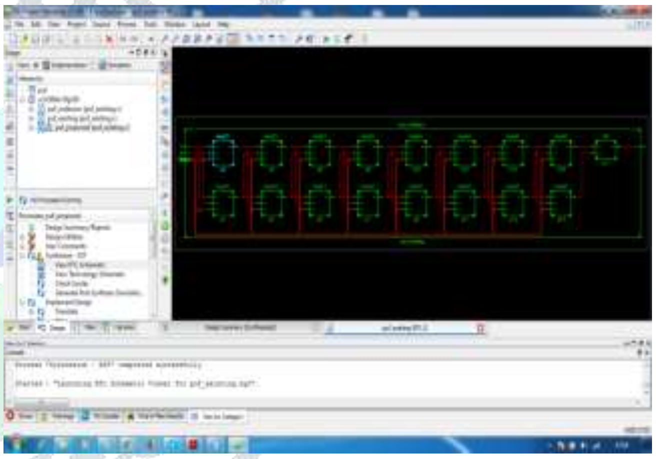

Figure 13: RTL Schematic of Proposed Design

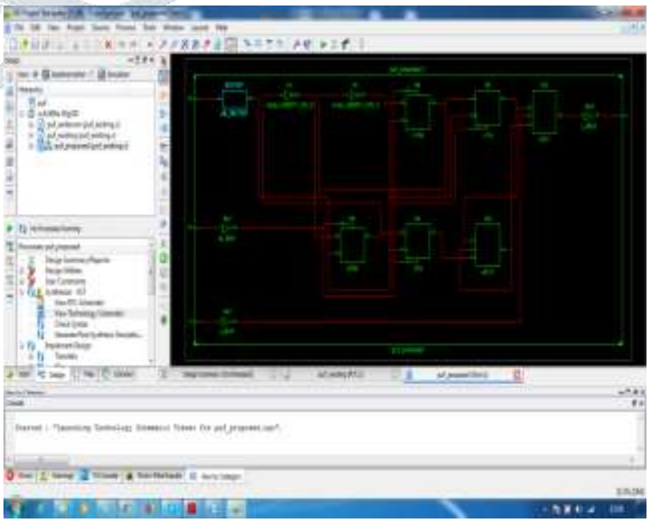

Figure 14: Technology Schematic of Proposed Design

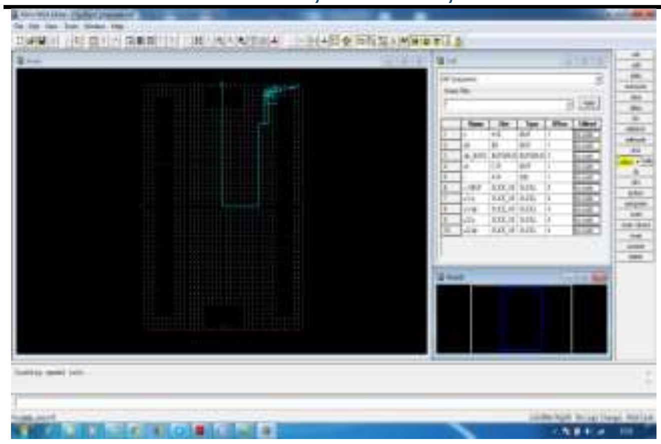

Figure 15: FPGA Implementation of Proposed Design.

Table - I: Comparison of designs for various Parameters

Parameters	PUF_Existing_1	PUF_Existing_2	PUF_Proposed
No. of slices	1 out of 4656	0 out of 4656	2 out of 4656
No. of 4-input LUTs	1 out of 9312	3 out of 9312	4 out of 9312
Combinational Path Delay	5.753ns	-	5.670ns
Average Fanout	1.00	1.00	1.88
Total Power	0.082W	0.081W	0.082W
Logic Power	0.0000	0.0000	0.0000
Signal Data Power	0.0000	0.0000	0.00001
I/O Power	0.00106	0.00005	0.00108

From Table - I, the proposed PUF increases the Fanout for the designs and reduces the delay in acquiring the key.

5. CONCLUSION

Reconfigurable high security systems often require secret keys to encrypt and decrypt data which operate based on physical unclonable functions (PUFs) which use random manufacturing variations to produce secret keys that are unique to each device. As the implementation of PUFs on field-programmable gate arrays (FPGAs) is usually difficult, because the designer has limited control over layout, and each PUF system requires a large area overhead to correct errors in the PUF response bits. The state of the art for FPGA-based weak PUFs is extended by using a novel methodology of per-device configuration and a new PUF variant derived from the popular FPGA-specific Anderson PUF. The design has several advantages over existing work including the Anderson PUF on which it is based. It is tunable to minimize the response bias and can be implemented using the common SLICEL components on Xilinx FPGAs along with an efficient per-device configuration that reduces bit error rate by over 10× at room temperature and improves response stability by over 2× across all temperatures. The Proposed PUF reduces the delay when compared to Existing PUFs along with error detection an correction capability. The designs are modeled in Verilog HDL and are simulated in Xilinx ISE 14.5 Tool. The functional Verification is Performed using Xilinx ISE Simulation Tool. The proposed design proves to be more effective as it could increase the fanout of design with less delay.

REFERENCES

1. Moradi, D. Oswald, C. Paar, and P. Swierczynski, “Side-channel attacks on the bitstream encryption mechanism of Altera Stratix II: Facilitating black-box analysis using software reverse-engineering,” in *Proc. ACM/SIGDA Int. Symp. Field Program. Gate Arrays*, Feb. 2013, pp. 91–100.
2. Introduction to Implementing Design Security With Microsemi SmartFusion2 and IGLOO2 FPGAs, Microsemi Corp., Aliso Viejo, CA, USA, Nov. 2013.
3. *Stratix 10 Device Data Sheet*, Intel Corp., Santa Clara, CA, USA, Jan. 2018.
4. J. H. Anderson, “A PUF design for secure FPGA-based embedded systems,” in *Proc. Asia South Pacific Design Autom. Conf.*, 2010, pp. 1–6.
5. J.-J. Wu and R. Huang, “A FPGA-based wireless security system,” in *Proc. Int. Conf. Multimedia Inf. Netw. Secur.*, 2011, pp. 512–515. [6] *Product Brief: AES-XTS Cores for FPGA*, Helion Corp., Omaha, NE, USA, Jan. 2016.
6. S. S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls, “Extended abstract: The butterfly PUF protecting IP on every FPGA,” in *Proc. IEEE Int. Workshop Hardw.-Oriented Secur. Trust*, Jun. 2008, pp. 67–70.
7. E. Simpson and P. Schaumont, “Offline hardware/software authentication for reconfigurable platforms,” in *Cryptographic Hardware and Embedded Systems—CHES*. Berlin, Germany: Springer, 2006, pp. 311–323.
8. K. Hu, T. Wolf, T. Teixeira, and R. Tessier, “System-level security for network processors with hardware monitors,” in *Proc. IEEE/ACM Design Autom. Conf.*, Jun. 2014, pp. 1–6.
9. P. Sedcole and P. Y. K. Cheung, “Within-die delay variability in 90 nm FPGAs and beyond,” in *Proc. Int. Conf. Field Program. Technol.*, 2006, pp. 97–104.
10. L. Cheng, J. Xiong, L. He, and M. Hutton, “FPGA performance optimization via chipwise placement considering process variations,” in *Proc. Int. Conf. Field Program. Logic Appl.*, Aug. 2006, pp. 1–6.
11. A. M. Bsoul, N. Manjikian, and L. Shang, “Reliability- and process variation-aware placement for FPGAs,” in *Proc. Conf. Design, Autom. Test Eur.*, 2010, pp. 1809–1814.
12. Gassend, D. Clarke, S. Devadas, and M. van Dijk, “Silicon physical random functions,” in *Proc. ACM Conf. Comput. Commun. Secur.*, 2002, pp. 148–160.
13. M. Majzoobi, F. Koushanfar, and S. Devadas, “FPGA PUF using programmable delay lines,” in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, Dec. 2010, pp. 1–6.
14. S. Keshavarz and D. Holcomb, “Threshold-based obfuscated keys with quantifiable security against invasive readout,” in *Proc. 36th Int. Conf. Comput.-Aided Design*, 2017, pp. 57–64.
15. J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, “FPGA intrinsic PUFs and their use for IP protection,” in *Cryptographic Hardware and Embedded Systems—CHES*. Berlin, Germany: Springer, 2007, pp. 63–80.
16. O. Sander, B. Glas, L. Braun, K. D. Müller-Glaser, and J. Becker, “Intrinsic identification of Xilinx Virtex-5 FPGA devices using uninitialized parts of configuration memory space,” in *Proc. Int. Conf. Reconfigurable Comput. FPGAs*, Dec. 2010, pp. 13–18.