Design and Implementation of PUFs for Automotive Applications

Alikhan Kapar, B. Eng

Submitted in fulfillment of the requirements for the degree of Master of Science in Electrical and Computer Engineering

NAZARBAYEV UNIVERSITY

School of Engineering and Digital Sciences
Department of Electrical and Computer Engineering
Nazarbayev University
53 Kabanbay Batyr Avenue, Astana, Kazakhstan, 010000

Supervisor: Associate Professor Akhan Almagambetov

Co-supervisor: Associate Professor Mohammad Hashmi

April 2024

Declaration

I hereby declare that this manuscript, entitled "Design and Implementation of PUFs for Automotive Applications", is the result of my own work except for quotations and citations which have been duly acknowledged.

I also declare that, to the best of my knowledge and belief, it has not been previously or concurrently submitted, in whole or in part, for any other degree or diploma at Nazarbayev University or any other national or international institution.

Alikhan Kapar

April 2024

Abstract

The automotive industry has undergone remarkable changes in recent times, characterized by an emphasis on the integration of complex electronic systems into vehicles. The goal of this evolution is to improve user experience, security and performance. However, this has also brought new challenges in ensuring the safety and reliability of automotive systems.To overcome these obstacles, this thesis focuses on creating Physical Unclonable Functions (PUFs) based on Field-Programmable Gate Arrays (FPGAs) that are specifically suited for automotive applications. Since PUFs are inherently unpredictable and unique, they present a promising option for safe hardware authentication and protection against a wide range of security risks.

This study investigates the architectural layout, methods of application and the fundamental ideas behind FPGA-based PUFs. The proposed PUF design produces 128-bit distinct and unclonable identifiers (IDs) by utilizing the inherent manufacturing variances of FPGAs and provides a strong security measure. Thorough testing that extends the three standard performance metrics of uniqueness, uniformity, and reliability shows how well the implemented PUF design satisfies the demanding security and performance standards of automotive systems.

Resource utilization analysis highlights the design's suitability for the resource-constrained automotive applications by revealing its efficiency in terms of power and FPGA resource consumption. The thesis also suggests future directions for investigation, such as the development of advanced error correction methods to increase the dependability of PUF responses, scalability improvements, and the integration of the PUF design with different automotive subsystems.

This thesis adds to the body of knowledge in the field of automotive security by providing a novel method for utilizing FPGA-based PUFs to improve the security infrastructure of contemporary vehicles through a thorough investigation and empirical evaluation. The

study's findings will have a major impact on the automotive industry and help create safer, more reliable and trustworthy automotive systems.

Acknowledgements

First and foremost, I want to sincerely thank Professor Mohammad Hashmi and Professor Akhan Almagambetov, who have been my supervisors. I have been able to effectively complete this master's program because of their guidance and supervision during my studies. The course of my research has been greatly influenced by their patient, knowledgeable, and insightful comments, and I am appreciative of their unwavering support.

I also want to express my gratitude to Dr. Nalla Anandakumar for all of his assistance throughout my research, as well as for his insightful remarks and important contributions. His vast knowledge, skill, and mentoring have been really helpful to me in deepening my understanding of the research problem.

Finally, I want to express my gratitude to my wife and my family in particular for their continuous encouragement and support during the program.

Contents

List of Abbreviations and Symbols

ECUs	Electronic Control Units
ADAS	Advanced Driver Assistance Systems
FPGA	Field-Programmable Gate Array
PUFs	Physical Unclonable Functions
V2X	Vehicle-to-Everything
VHSIC	Very High-Speed Integrated Circuit
VHDL	VHSIC Hardware Description Language
V2V	Vehicle-to-Vehicle
RS-LPUF	RS Latch-based PUF
A-PUF	Arbiter-based PUF
PDLs	Programmable Delay Lines
TMV	Temporal Majority Voting
CRPs	Challenge-Response Pairs
LUTs	Look-Up Tables
CLBs	Configurable Logic Blocks
ASICs	Application-Specific Integrated Circuits
IP	Intellectual Property
CAD	Computer-Aided Design
CRRPs	Challenge-Response-Reliability Flag Pairs
ECCs	Error Correction Codes
ROs	Ring Oscillators
SCM	Statistical Complexity Measure
NIST	National Institute of Standards & Technology

LFSRs	Linear Feedback Shift Registers
CAN	Controller Area Network
HSMs	Hardware Security Modules
TPMs	Trusted Platform Modules
RKE	Remote Keyless Entry
SRAM	Static Random-Access Memory
IoV	Internet of Vehicles
IoT	Internet of Things
V2G	Vehicle-to-Grid
EVs	Electric Vehicles
RO-PUF	Ring Oscillator-based PUF
PRNG	Pseudo-Random Number Generator
BCH	Bose, Chaudhuri, Hocquenghem
PPUFs	Public PUFs
V2I	Vehicle-to-Infrastructure
VANETs	Vehicular Ad Hoc Networks
MA	Mutual Authentication
PIDs	Pseudo IDs
CAN-FD	Controller Area Network Flexible Data-Rate
SIDH	Supersingular Isogeny Diffie-Hellman
UART	Universal Asynchronous Receiver Transmitter

List of Figures

Introduction

1.1 Evolution of automotive systems

The automotive sector has experienced an important transformation in the last century, progressing from basic mechanical devices to complicated electronic and digital marvels. This development began in the early 1900s, when the majority of an automobile's systems were mechanical, with only a few electrical parts, like the lighting and ignition system. As technology developed, electronic features such as electronic ignition and fuel injection were introduced in the 1960s and 1970s, which initiated the incorporation of electronics into automotive design.

However, real change began with the introduction of digital technology in the late 20th and early 21st centuries. A wide range of electronic control units (ECUs), sensors and actuators are now standard on modern vehicles. They are all working together to improve efficiency, safety and user experience [1]. These systems regulate almost every aspect of the car's functioning such as infotainment, advanced driver assistance systems (ADAS), braking and engine management.

This digital revolution has brought new features like electrification, autonomous driving, and connected car technologies in addition to enhancements in safety and performance. Nowadays cars are a part of a wider ecosystem and they are linked to other cars and the Internet. Also, they can receive updates over the air, just like smartphones and other modern digital devices [2].

However, this transition to more electronic and digital systems has also presented new difficulties in terms of reliability and security [3]. The increased connectivity of vehicles raises the possibility of cyber threats. Therefore, sophisticated security measures to prevent unwanted access and guarantee the confidentiality and safety of users are required.

1.2 Hardware security of automotive systems

The automotive industry has heavily prioritized hardware security enhancements due to the substantial risks involved. While traditional security measures such as software-based and encryption are critical, they may not be sufficient to defeat sophisticated hardware-based attacks [4]. For example, attackers can bypass software security measures or introduce malicious firmware by exploiting hardware flaws. Hardware-based security mechanisms are useful in this situation. They are more resistant to tampering and exploitation since they are made to be an essential part of the physical construction of the device. Some examples of such safeguards are hardware-based authentication systems, cryptographic keys kept in hardware, and secure hardware modules. They offer a strong base for security, guaranteeing that the underlying hardware stays safe even if the software is corrupted [5]. Ensuring that vehicles function safely and securely in the face of evolving cyber threats is the priority of trust and reliability. The significance of hardware security will become more and more evident as the automobile sector innovates and incorporates cutting-edge technologies. This phenomenon creates an opportunity and necessity to investigate cutting-edge security solutions like Physical Unclonable Functions (PUFs) [6] which present a novel method for hardware authentication and protection of the automotive industry.

1.3 The role of FPGAs in the automotive industry

Field-Programmable Gate Arrays (FPGAs) have become a key technology in the automotive sector due to the changing nature of automotive systems and the increasing demand for strong hardware security as stated above. FPGAs are flexible, reconfigurable circuits that have a broad range of programming options. They are extremely versatile and fit for a wide range of automotive applications. They play a crucial role in modern cars by filling the gap between requirement for being flexible and being able to be updated, as well as, being fixed-functionality hardware [7]. Since they are reconfigurable, the hardware can be

updated or modified after implementation and this feature helps to accommodate new standards, security fixes, or functionality without having to be replaced. This is especially helpful in the automobile sector, where the longer lifespan of cars makes component longevity and adaptability an important factors.

FPGAs can be used in a variety of automotive applications such as multimedia, power management, vehicle-to-everything (V2X) communications, as well as ADAS [8]. Because of their ability to fast and parallel computing, they are perfect for real-time applications like processing sensor data or making decisions in autonomous driving systems. In addition, FPGAs provide a high degree of determinism and reliability, which is necessary for safety-critical operations in vehicles. FPGAs' perspective security capabilities are yet another important benefit for the automotive sector. Security features like hardware-based authentication, encryption, and secure boot can all be implemented using FPGAs. They offer a strong basis for safeguarding against both physical and cyber hazards, which makes them a desirable option for building secure automotive systems. the use of FPGAs in automotive design highlights the industry's move towards more adaptable, updatable, and secure electronic systems. And this is happening in line with the larger concepts of digital transformation and cybersecurity in the automobile industry. It will become clearer how important and promising FPGAs are for developing new security features for hardware authentication as we go into the details of PUFs in the following section. This is a major step forward for automotive hardware security.

1.4 Rise of PUF as a novel security feature for hardware authentification

PUFs can become a revolutionary security feature in the context of modern automotive applications, changing automotive systems, and the urgent need for better hardware security. PUFs are able to produce cryptographic keys or unique identifiers by taking advantage of the inherent physical variances that naturally arise during the manufacturing process in semiconductor devices [11]. Because of their intrinsic uniqueness, PUFs are the perfect

choice for secure key generation and hardware authentication since it is nearly impossible to replicate or predict the PUF response.

Within the automotive domain, the incorporation of PUF technology into FPGA-based systems offers a strong strategy for tackling the security issues that come with cars going digital. FPGAs play a key role in many automotive systems, ranging from connectivity and autonomous operations to controlling critical functions. Therefore, adding PUFs strengthens the security infrastructure by offering reliable methods to verify hardware components and secure communication between them.

PUFs are beneficial in automobile security applications in a number of ways. Firstly, because the unique characteristics utilized for authentication are essentially attached to the device's physical construction and cannot be changed without significantly altering its operation, they offer a high degree of tamper resistance. Second, by eliminating the requirement for non-volatile memory storage and enabling the production of cryptographic keys on demand, PUFs lower the danger of key disclosure or theft [17]. Finally, hardware-based attestation and secure boot processes are supported by PUF-based systems, guaranteeing that the device runs only trusted software and that the hardware is authentic and unaltered.

The car industry is well-positioned to progress toward a future in which hardware authentication and security are tightly integrated into the fabric of automotive technology by combining the adaptability and performance of FPGAs with the intrinsic security features of PUFs. By taking a novel approach, automotive systems' security position is strengthened and the foundation for an entirely novel type of reliable, secure, and trustworthy vehicles is established.

1.5 Aims and objectives

This thesis' main goal is to create a PUF circuit for FPGAs that is especially suited for use in automotive applications. This section establishes the particular goals that direct this research project, guaranteeing that the suggested solution properly addresses the distinct

needs and obstacles related to automotive hardware security. The objectives of this thesis are:

- ***Design a PUF Circuit for FPGA.*** Create a PUF circuit design that's suitable for MAX10 DE-10 Lite FPGA board and make use of the its inherent physical properties to produce unique IDs. The design should take into account the limitations and capabilities of the FPGA architecture to achieve optimal performance and reliability. Additionally, the PUF design have to be flexible and be able to support a range of automotive applications such as simple vehicle identification systems and intricate secure communication protocols.

- ***Implement the PUF Design Using VHDL.*** Utilize VHDL (Very High-Speed Integrated Circuit (VHSIC) Hardware Description Language) to implement the PUF circuit on MAX10 DE-10 Lite FPGA board. The implementation must follow best practices in hardware description and guarantee that the design is effective. Also, it should be robust and reliable to mitigate the impacts of aging and environmental changes that may affect the consistency of the PUF's response throughout vehicle's life.

- ***Validate Uniqueness, Uniformity and Reliability of Generated IDs.*** Perform in-depth tests to determine whether IDs produced by the PUF are distinct among various FPGA implementations. Statistical analysis must be used in this process to guarantee that every PUF instance generates a unique identifier that can be used as a safe hardware fingerprint. Also, thesis must examine how well the PUF reacts to different operating circumstances, such as fluctuations in temperature and voltage changes. It must be done in order to replicate the environmental conditions found in vehicle circumstances.

- ***Demonstrate Application of PUF-generated IDs in Automotive Security.*** Present the real-world use of the PUF-generated IDs in a relevant automotive security circumstance.

By achieving these goals, this thesis aims to make significant contribution to understanding of current issues regarding automobile system security in the digital era and offer reliable solutions.

1.6 Thesis outline

This thesis is systematically divided into multiple important sections. The introduction section presents background information by going over the development of automotive systems, the importance of hardware security, the role of FPGAs, and the novel application of PUFs. Next, the related works section offers a critical analysis of previous research to place this study in the context of the larger field. The research strategy and methods used to design the PUF circuit and implement it on FPGA board are described in the methodology section. Details on the PUF circuit design and implementation are provided in the proposed design and implementation section. The post-processing and setup section covers the processing of the raw ID generated by the proposed design to remove noise and describes the overall setup utilized. The performance evaluation and resource utilization part then goes into how the design was realized on the FPGA in terms of resources. Also, this part discusses how various experiments were used to analyze the reliability and effectiveness of the proposed design. The conclusions and future work section serves as the thesis's conclusion, providing a summary of the results, a discussion of the implications and recommendations for future study directions.

Literature Review

The research articles were carefully examined and classified into three subcategories that are relevant to FPGA-based PUFs in automotive industry. Every subsection is intended to cover specific aspects of the subject. Aspects include theoretical ideas and design principles, useful implementation techniques and actual automotive security applications.

2.1 Foundations and Design Principles of FPGA-based PUFs

This subsection includes articles that explain different PUF architectures and give a theoretical basis for PUF technology. It also go into detail about the design principles unique to FPGA implementations. The foundation for comprehending the inherent qualities and design considerations necessary for creating robust and reliable PUFs on FPGA platforms is laid by these papers.

A novel method for improving hardware-based security is presented in the paper [10]. This method involves creating a lightweight, space-efficient hybrid PUF using FPGA technology. The RS Latch-based PUF (RS-LPUF) and Arbiter-based PUF (A-PUF) in this design are combined with programmable delay lines (PDLs) and Temporal Majority Voting (TMV). They are used to support performance metrics such as uniqueness, consistency, and reliability. The suggested architecture is classified as a weak PUF because it uses so few Challenge-Response Pairs (CRPs). Nevertheless, it greatly improves PUFs' performance characteristics, which are conventionally determined by bit-aliasing, uniqueness, uniformity, and reliability. These characteristics are frequently prone to deterioration. The reason for that is correlated or systematic process variations and outside noise. PDLs and TVM are hybridized and integrated to offer a robust solution for key generation and device ID generation applications in order to get around these limitations.

The main idea behind this design is to increase the uniqueness, randomness, and unpredictable nature of responses. To achieve this two different randomness sources inside a PUF are utilized. This method entails XORing RS-Latch cell outputs with Arbiter PUF instances, with a focus on using hard macro techniques, careful routing, and placement. All of this is done to minimize delay skew and bias caused by design. This approach makes sure that the PUF's response is solely dependent on the inherent differences found in FPGA Look-Up Tables (LUTs) so that generated keys or IDs remain secure and intact. The hybrid PUF is implemented using the Configurable Logic Blocks (CLBs) and slices of the FPGA. The design was created with the Xilinx ISE design suite and coded in Verilog HDL. The design

and implementation process is described in great detail in the paper. It also shows how the combination of PDLs and TMV improves the reliability metric and solves bit-aliasing and uniformity problems by using hard macro design techniques.

The research paper [11] offers a thorough analysis of PUFs in the context of FPGA devices. It emphasizes their crucial role in cryptographic protocols and security architectures. The authors describe the flexibility that FPGAs offer in terms of design, validation, and short time-to-market, making them ideal for a wide range of applications including military, automotive, consumer electronics, and more. They also discuss the market's rapid growth, which has been fueled by the rising costs of producing Application-Specific Integrated Circuits (ASICs). The article classifies FPGA-based PUF designs, describes their unique characteristics, and assesses their effectiveness. Diverse PUF varieties, including delay- and memory-based PUFs, as well as hybrid, composite, and double arbiter PUFs, are investigated for their special qualities and uses in hardware security. The study also discusses the vulnerability of FPGA-based PUFs to various attacks, including modeling, physical, side-channel, and cloning attacks. In addition, it suggests appropriate countermeasures to lessen these risks.

In order to highlight the flexibility and significance of PUF technology in improving FPGA security, the paper goes into great detail about the applications of FPGA-based PUFs in areas such as random number generation, identification, authentication, Intellectual Property (IP) protection, logic obfuscation, secret key generation, and key sharing. To progress PUF technology, the authors highlight the present obstacles and recommend potential areas of study. They stress the importance of creating workable solutions for entropy estimation, security concerns, Computer-Aided Design (CAD) frameworks, environmental factors, and hardware efficiency.

The study [12] introduces a Bit-Self-Test (BST) strategy to improve both uniqueness and reliability, which are critical metrics for PUF effectiveness. Result is a novel arbiter PUF design known as BST-APUF. A delay detection circuit is integrated into the traditional ar-

biter PUF architecture in the BST-APUF design. This circuit is responsible for automatically calculating PUF response's delay deviation for each bit. The bit is marked with reliability flag to indicate its dependability if the deviation is greater than predetermined threshold. In cryptographic applications where stable keys must be recovered or extracted from PUF responses, this method produces a large number of Challenge-Response-Reliability Flag Pairs (CRRPs), which is useful helper data.

The use of reliability flag to show the dependability of each PUF response upon input of particular challenge is significant innovation of this work. This feature sets the BST-APUF apart from conventional PUF designs by offering an extra layer of information (the reliability flag) in addition to the typical challenge-response pair (CRP). This allows for a more sophisticated evaluation of the PUF's functionality and applicability for different applications.

Testing results showed significant improvement in the uniqueness and reliability of the selected responses when the BST-APUF was implemented on a Xilinx Artix 7 FPGA. This meets requirements for secure cryptographic key generation and device authentication in systems with limited resources. Notably, the BST-APUF has less hardware resource requirements than current state-of-the-art mechanisms, which makes it desirable option for applications requiring lightweight security.

The creation of BST-APUF tackles two major issues with PUF design: increasing uniqueness, which is essential to PUFs' efficacy as secure identifiers and key generators, and improving response reliability without requiring the high overhead associated with error correction codes (ECCs). BST-APUF design strikes an delicate balance between performance and resource efficiency by adding a delay detection circuit that evaluates and flags the dependability of each response bit, providing a promising development in the field of PUF technology.

The research article [13] explores how using PUFs based on ring oscillators (ROs) in FPGAs can improve security of cryptographic systems. The study discusses critical function

that PUFs play in producing distinct, non-replicable responses that come from the physical properties of the chip itself. These responses are used in cryptographic applications for secure key generation and authentication.

The primary goal of the research is apply RO-based PUFs to FPGA environments in order to utilize the inherent variability brought about by manufacturing procedures and produce true random numbers, which are crucial for strong cryptographic security. In order to strengthen the PUFs against different attacks, the paper carefully investigates the application of challenges with both periodic and non-periodic structures to increase the randomness of the generated numbers.

The thorough statistical analysis of responses generated by PUF constitutes a substantial portion of the research. To assess the randomness and security characteristics of the responses the study uses a number of tests such as the scale index method, statistical complexity measure (SCM), auto-correlation tests, and the National Institute of Standards and Technology (NIST) statistical test suite. These analyses are essential for determining whether the PUF-generated numbers are suitable for use in cryptography applications and that they satisfy the strict criteria for unpredictability and true randomness.

The outcomes of the experiment show how well the suggested RO-based PUF design generates secure, random numbers. High-quality randomness is demonstrated by responses generated from non-periodic challenges. They are from non-linear combination generators and Linear Feedback Shift Registers (LFSRs), which perform better in passing the NIST tests. Moreover, the SCM results highlight how well the non-periodic challenge structures achieve the desired randomness properties.

In the context of contemporary vehicle systems, Carson the paper [14] offers a thorough summary of the cybersecurity issues and possible solutions. The integration and application of PUFs and Hardware Security Modules (HSMs) within automotive systems—especially ECUs, which are essential for vehicle operation and communication—are the main areas of focus.

The authors highlight how the complexity and interconnectedness of vehicular systems are increasing, especially with the introduction of autonomous vehicles, and how strong security measures are required to guard against possible cyberattacks. Because native encryption and authentication protocols are absent from the Controller Area Network (CAN) bus system—which is used for inter-ECU communication—its inherent vulnerabilities are emphasized as a major cause for concern. Because the CAN bus system is broadcast in nature, malicious actors could be able to compromise a vehicle's operation by using standard diagnostic ports like OBD-II, which emphasizes the need for more robust security frameworks.

A novel approach to vehicle security is presented: PUFs use the distinctive physical properties of electronic components to create a "fingerprint" specific to a particular device. They also offer a degree of security that is challenging for adversaries to duplicate or get around. Strong PUFs have the ability to generate an exponential number of CRPs, which would improve the security posture of vehicular systems. PUFs have the capacity to generate a large number of CRPs.

The article addresses the function of HSMs in automotive cybersecurity in addition to PUFs. HSMs are cryptographic co-processors that enable secure data storage and communication inside of automobiles. They were inspired by Trusted Platform Modules (TPMs), which are utilized in traditional computer systems. The development of specialized vehicular HSMs is necessary due to the distinct operational and resource constraints of automotive systems. These HSMs may only include a portion of the cryptographic functionalities present in standard TPMs. The goal of these specialized HSMs is to protect external and internal vehicle communication networks without placing an undue strain on resources or money.

In order to improve vehicle security, the authors also highlight a number of HSM and PUF implementations and applications. For example, the adaptability and potential effectiveness of these hardware security solutions in defending against cyber threats in automotive contexts is demonstrated by the vehicular HSM proposal of the EVITA project, which includes various variations tailored to different security needs within the vehicle.

The paper [15] addresses common security issues like replay and RollJam attacks by introducing a PUF-based mutual authentication (MA) mechanism specifically made for Remote Keyless Entry (RKE) systems in cars. This mechanism is novel because it makes use of PUFs to produce distinct, unclonable responses for authentication, greatly strengthening the security posture of RKE systems.

Due to their reliance on fixed or rolling codes, RKE systems—which let users unlock their cars without physical keys—have historically been subject to a variety of attacks. Replay attacks are common occurrence for fixed code systems where attacker records and retransmits the unlock signal. Even though rolling code systems are more secure, they are still vulnerable to advanced RollJam attacks. In these kind of attacks the attacker jams original signal to keep it from reaching the vehicle while intercepting and storing the rolling code signal for later use. The proposed authentication mechanism uses a novel approach that takes advantage of PUFs' inherent security properties in an attempt to mitigate these vulnerabilities.

The main contribution of this work is creation of a MA framework that is both lightweight and secure. Also it uses PUFs to prevent impersonation, replay, and RollJam attacks. A formal security analysis supporting the mechanism shows that it is resistant to common attack vectors that target RKE systems. Additionally, a comparative analysis with current schemes highlights the efficiency of the mechanism demonstrating a notable decrease in computational costs. Because of its effectiveness, the suggested method is especially well-suited for implementation in contemporary automobiles where there is a constant need for strong security measures and limited computational resources.

2.2 Implementation Strategies and Challenges on FPGAs

Articles in this subsection explore details of implementing PUFs on FPGA hardware. They cover deployment procedures, optimization techniques, and obstacles faced in practical PUF implementations. With an emphasis on the complexities of FPGA platforms, this section emphasizes the practical aspects of putting PUF designs from theory to reality.

The study [16] discusses difficulties and solutions involved in creating scalable, effective, and lightweight PUFs for FPGAs. PUFs generate n-bit binary strings that act as a digital fingerprint for hardware devices, giving them a distinct identity. Each PUF is distinct due to these strings, which are a result of the devices' inherent manufacturing variations. A major difficulty with FPGA-based PUFs has been striking a balance between controlling the amount of hardware resources used and maintaining a high enough level of uniqueness and reliability. When implemented on FPGAs, many current PUF implementations find it difficult to effectively meet these requirements. To address these problems the authors of this article present a novel PUF identification (ID) generator circuit. It is made especially for FPGAs. The compact design of this system makes it stand out because it provides good uniqueness and reliability without consuming a lot of hardware resources.

This work makes a significant contribution by introducing a novel post-characterisation methodology that aims to increase PUF reliability without requiring extra hardware resources. This approach is flexible and can be used to improve any FPGA-based PUF design's reliability. It is shown that the suggested PUF ID generator is very effective in terms of using hardware resources; on an inexpensive Xilinx Spartan-6 LX9 FPGA and 0.81% on a Xilinx Artix-7 FPGA, it only uses 8.95% of the hardware resources. With no bit-aliasing and outstanding performance in terms of uniqueness, consistency, and reliability, the experimental results show that this is a promising solution for secure identification in FPGA applications.

The study goes on to address relevant research in the area of PUF designs, highlighting several PUF varieties with pros and cons, including Static Random-Access Memory (SRAM) PUFs, latch PUFs, flip-flop PUFs, and others. The authors analyze these designs critically and offer their own solution to address some of the found drawbacks, especially with regard to FPGA implementations.

The creation of an improved RO PUF suited for applications in the Internet of Vehicles (IoV) is explored in the paper [17]. The paper highlights the increasing interconnectedness made possible by the Internet of Things (IoT) and how it extends into specialized fields

like IoV, which aims to improve vehicle communication for increased efficiency and safety. Because data exchange in these networks is so vital, the paper emphasizes the need for strong security measures. PUFs are positioned as workable solution because of their capacity to produce distinct, device-specific signatures that improve hardware security.

The paper provides a framework for the suggested modified RO PUF design by outlining a thorough analysis of conventional RO PUFs and their drawbacks. The study adds a 32-ring oscillator configuration with XOR and inverter gates to improve the PUF's performance in terms of uniqueness, reliability, and uniformity. Also this design seeks to address the drawbacks of traditional RO PUFs. Achieving a uniqueness of 49.83%, reliability of 99.93%, and uniformity of 49.75%, the implementation shows notable improvements in these areas and indicates the design's potential to provide efficient authentication mechanisms for IoT devices. The architecture of the suggested RO PUF design is explained in detail, along with how it varies from conventional PUF designs and its working principle. A dual-path oscillator system with a complex set of logic gates and multiplexers is part of the modified structure. It is optimized to take advantage of manufacturing variances for increased randomness and security outcomes. The unpredictable responses of the PUF, which are essential for key generation and authentication in security-critical applications like IoV, are made possible in part by this complexity.

The implementation section describes the RO PUF's FPGA deployment in more detail and emphasizes the design's versatility and effective use of available resources. Through performance metrics, the paper presents empirical evidence of the proposed design's efficacy. In addition, paper shows that it is superior to other modified PUF designs and traditional designs in terms of uniqueness, reliability, and uniformity. These metrics highlight how well-suited the design is for secure key generation in environments with limited resources, which are common to IoT devices.

A novel design and implementation of an XOR Arbiter (XOR APUF) on FPGAs is proposed in the paper [18] with a focus on IoT applications. The goal of the implementation

is to make XOR APUF more resistant to modeling attacks while maintaining a high level of uniqueness and randomness in the PUF responses. The first section of the paper highlights the importance of PUFs in the context of IoT security. It emphasizes how PUFs can be used to generate cryptographic keys or identifiers that are specific to a given device and difficult to copy or predict because of variations in manufacturing processes. Because of its ability to generate a large number of CRPs, the XOR APUF is seen as particularly promising among the various types of PUFs for challenge-response pair (CRP)-based authentication applications. This provides reasonable security. The authors do note that there are difficulties in implementing high-quality XOR APUFs on FPGAs, mainly because of the rigid and fixed interconnect structure of FPGAs, which can make it difficult to achieve the necessary levels of security and uniqueness. The authors suggest an XOR APUF architecture that combines a number of cutting-edge methods to address these issues. To improve security against modeling attacks, the design first uses discrete programmable delay logic (PDL) configurations to add variability and randomness. The internal challenge bits are obscured by the permutation of input challenges, which is based on a cryptographic algorithm and makes the task more difficult for possible attackers. To lessen response instability and raise attack complexity, the paper also introduces a majority voting mechanism before the XOR operation. The effectiveness of the suggested XOR APUF design is shown by the experimental results reported in the paper. The uniqueness, consistency, and dependability of the PUF responses were the evaluation metrics, and the design was executed on Xilinx Artix-7 FPGAs. In comparison to earlier XOR APUF implementations, the suggested design demonstrated notable improvements in uniqueness and security features, yielding encouraging results. These outcomes were made possible by the use of discrete PDL configurations, obfuscated challenges, and majority voting prior to the XOR operation.

In order to provide a strong framework that not only guarantees physical and cyber security but also protects user identity and location privacy, paper [19] suggests PUF-based secure authentication protocol designed specifically for Vehicle-to-Grid (V2G) communica-

tions. In order to protect against a wide range of potential attacks such as eavesdropping, message alteration, impersonation, replay attacks, Denial of Service, physical attacks, and threats to identity and location privacy, the research introduces a novel approach within the V2G network. This network consists of Electric Vehicles (EVs), Charging Stations, and a Grid Server. One noteworthy advantage of the suggested protocol is its thorough security analysis, which shows that it is resistant to every attack that has been thought of. This is especially important when it comes to V2G communications, where transaction security and privacy are crucial. The performance evaluation highlights the low computational cost of the protocol, which is particularly helpful considering the restricted computational resources found in EVs. This further emphasizes the protocol's efficiency. As part of the protocol, each EV is only needed to complete three one-way hash functions, demonstrating how lightweight the scheme is.

Furthermore, the extensive coverage of security and functional features provided by the suggested scheme is highlighted by the paper's feature-based comparison with current protocols. It is a major improvement over earlier solutions in that it effectively handles crucial issues like data confidentiality, integrity, two-way communication enablement, and MA.

In order to overcome the shortcomings of current models, the paper [20] presents a novel cross-trusted authority authentication and key agreement protocol for the IoV that makes use of elliptic curve cryptography and blockchain technology. The suggested protocol seeks to address the security flaws found in the blockchain-based Roadside Unit (RSU)-assisted authentication and key agreement protocol for IoV developed by Xu et al. These flaws include identity guessing attacks, vehicle forgery attacks, and weaknesses in known session key secrecy and session key security.

The protocol is a major step forward in protecting IoV communications, as it makes use of PUFs and biometric keys to improve security against RSU capture attacks and Onboard Unit (OBU) intrusion attacks. The authors show by formal security proofs and comparative analysis that their protocol is capable of withstanding a broad range of known attacks with

reduced computational complexity, which makes it a workable solution for safe and effective cross-domain identity authentication and key agreement in the context of the IoV.

In addition, the paper's conclusion highlights how the protocol uses blockchain to enable cross-domain vehicle authentication while also providing lightweight anonymous identity authentication and key agreement. The system is further strengthened against possible security breaches from both internal and external threats by the integration of PUF and biometric keys. The protocol's superior security and efficiency are attested to by the thorough security analysis and comparative evaluations, which present a promising strategy for improving the security infrastructure of IoV systems.

2.3 Automotive Security Applications of FPGA-based PUFs

This subsection focuses on how FPGA-based PUFs are used in the automotive sector to improve security. The use of PUFs for secure authentication, anti-counterfeiting, secure communication, and other security measures in automotive systems is examined in these articles. The benefits and practical applications of PUF technology in resolving the security issues that contemporary automotive systems face are illustrated in this subsection.

The research paper [21] explores and analyzes two particular FPGA-implemented PUFs: the RS-LPUF and the Ring Oscillator-based PUF (RO-PUF). Due to the TMV scheme, the strategic integration of PDLs, and the use of placed macro techniques for PUF unit routing and placement, these PUFs exhibit notable performance improvements over previous models.

PUFs are divided into two types based on their operating principles: memory-based and delay-based. While delay-based PUFs like Arbiter PUF and RO-PUF use digital race conditions or frequency variations, memory-based PUFs like SRAM-PUF and RS-LPUF rely on the instability of volatile memory cells. Based on their challenge-response pair (CRP) spaces, PUFs are further divided into strong and weak PUFs. Strong PUFs have a large set of CRPs that are appropriate for applications involving direct authentication, while weak

PUFs have a small number of CRPs that are perfect for applications involving key generation and pseudo-random number generators (PRNGs).

Both internal and external factors affect PUF performance, which is measured by bit-aliasing, reliability, uniqueness, and uniformity. In addition to highlighting the detrimental effects of environmental fluctuations on these metrics, the article describes how the TMV scheme, hard/placed macro techniques, PDLs, and the combination of PUF outputs to improve uniformity and security all contributed to improved performance.

The article [22] presents a novel method of improving security in automotive systems through the integration of Bose, Chaudhuri, Hocquenghem (BCH) error-correcting codes and PUFs for Electronic Control Unit (ECU) authentication over the CAN protocol. Modern cars, sometimes called "computers on wheels," have many ECUs for different functions, such as airbags and emergency braking, among other safety-critical systems. However, there are serious security risks because the CAN protocol, which enables real-time ECU communications, lacks built-in authentication mechanisms. The urgent need for reliable ECU authentication techniques is highlighted by the possibility that malicious or unauthorized ECUs could compromise vehicle safety systems.

To ensure the authenticity of each ECU, the proposed authentication system creates unique signatures for each one using PUFs' unclonable nature. Similar to digital "fingerprints," PUFs react to a given set of challenges in a variety of unpredictable ways. In order to reduce noise in PUF responses, improve reliability, and guarantee accurate authentication even in the face of changing environmental factors like temperature swings and device aging, the system makes use of BCH codes. Securing ECU communications against potential threats like impersonation or replay attacks is a critical challenge that is addressed by this combination of BCH codes for error correction and PUFs for uniqueness.

By preventing unwanted access and control of vehicle systems, the adoption of this authentication system in the automotive industry could greatly improve the security of communications between vehicles and infrastructure. The system reduces the possibility of mali-

cious activity by limiting the number of verified ECUs that can join the network, improving overall vehicle safety and security.

By addressing a critical security issue with contemporary automotive systems and offering a workable solution that strikes a balance between efficacy and computational efficiency, this work advances the field. A viable path toward creating safe, robust automotive communication systems that can withstand a variety of cyberattacks is the integration of PUFs and BCH codes.

The work [23] explores the use of Public PUFs (PPUFs) and conventional PUFs as a substitute for traditional HSMs in the creation of safe pseudonyms in vehicular networks. In order to guarantee security and user acceptance in Intelligent Transportation Systems, it is imperative that vehicles be authenticated while maintaining their privacy. This calls for the use of pseudonymity.

The conventional method of creating pseudonyms entails a backend infrastructure that issues many pseudonyms to vehicles, usually a Certification Authority or Pseudonym Provider. However, this procedure presents a number of difficulties, such as managing and securely storing a sizable quantity of private key material. The authors suggest using PUFs to improve the security and efficiency of pseudonym generation and to streamline credential management. PUFs take advantage of the inherent manufacturing variances of devices to generate unique identifiers.

PUFs are superior to conventional techniques in a number of ways. Because they can create secure keys based on a device's physical characteristics, each PUF is distinct and challenging to duplicate. By enabling vehicles to self-generate their own pseudonyms, PUFs have the potential to reduce the need for secure key storage, ease the burden on pseudonym providers, and enhance system scalability. The importance of stability in the generated keys and the function of fuzzy extractors in achieving this stability are emphasized in the paper's discussion of key extraction from PUF responses.

Additionally, the study examines a number of pseudonym schemes, emphasizing the

significance of secure key generation in each method, including those based on group signatures, symmetric cryptography, identity-based cryptography, and asymmetric cryptography. The study provides a thorough analysis of PUFs' potential to improve the security and privacy of automotive networks while also offering a cutting-edge remedy for some of the current problems with pseudonym generation and management.

An extensive examination of the potential and application of PUFs in the automotive sector can be found in the paper [24]. As IT-based systems are integrated into cars more and more, security issues like software piracy, component counterfeiting, and illegal tampering with digital data in ECUs are becoming more and more prevalent. Promising answers to these problems can be found in PUFs, which are known for their distinct and unclonable qualities brought about by differences in the manufacturing process.

The paper begins by laying out the background information and highlighting the critical need for security and safety in the context of contemporary automobiles, which are complex systems embedded with multiple processors and large amounts of software code. It draws attention to the threat that fake parts pose to a variety of industries, including the automotive one, and emphasizes the negative effects these activities have on the economy and public safety.

The concept of PUFs, their characteristics, and the presumptions guiding their security are covered in great detail throughout the paper. PUFs are described as mappings from physical stimuli (challenges) to digital responses, which are hard to duplicate because every device has different physical properties. The article describes in detail the noisy character of PUF responses and presents the idea of fuzzy extractors, also known as helper data algorithms, which are crucial for extracting reliable and secure keys from PUFs.

The paper then shifts to the particular use of PUFs in automotive systems, going over a number of security goals like data integrity, confidentiality, and hardware component uniqueness. It suggests that PUFs are a good fit for security modules in cars because they are tamper-evident and cannot be replicated, which could make security solutions already in

place simpler.

The study also examines the use of PUFs in insurance applications, secure key storage, component identification, and IP protection in the automotive industry. For example, PUFs could make it difficult for attackers to compromise key material by enabling secure on-the-fly key generation for cryptographic operations. In a similar vein, PUF integration with public-key infrastructure may enable software to be bound to particular hardware, providing a strong IP protection mechanism.

In conclusion, the paper proposes a novel application of PUFs as insurance seals, providing unquestionable confirmation of the caliber of parts installed in cars after collisions. Because this application guarantees the use of certified components, it could have a significant impact on the aftermarket parts industry, improving vehicle integrity and safety.

The use of PUFs and post-quantum cryptography within the Controller Area Network Flexible Data-Rate (CAN-FD) to improve vehicular security is examined in the research paper [25]. Securing ECUs against potential quantum computing threats and guaranteeing long-term security are critical as modern vehicles become more interconnected and dependent on ECUs for various functions.

The vulnerabilities in the conventional CAN used in automobiles, which can be used to obtain unauthorized control over delicate vehicular subsystems, are the driving force behind this study. Existing cryptographic techniques—particularly asymmetric cryptography—run the risk of being compromised by the development of quantum computing. Additionally, the paper emphasizes how post-quantum cryptography techniques limit the payload size and efficiency of standard CAN, thereby promoting the use of CAN-FD, which provides higher transmission speeds and payload capacities.

In order to improve security for vehicle communication systems, this paper contributes a new framework called PUF-PQC-CANFD that combines PUFs with post-quantum cryptography methods. The framework is made to reduce message transmission overhead, minimize performance overhead, and comply with CAN standards. Through the use of PUFs,

the framework guarantees the unclonability and dependability of authentication signatures within the cryptosystems while leveraging high-entropy, consistent responses to generate cryptographic material.

Supersingular Isogeny Diffie-Hellman (SIDH) and AES-256-GCM are used in the proposed PUF-PQC-CANFD framework for cryptographic operations, optimizing for smaller post-quantum key sizes and reduced storage and transmission requirements. It is demonstrated that this strategy protects against a variety of cryptography-based attacks, including quantum attacks, while transmitting a notably smaller number of messages than current pre- and post-quantum frameworks.

A privacy-preserving authentication protocol based on PUFs is presented in the research article [26]. It is intended for use in vehicular ad hoc networks (VANETs) and allows for simultaneous authentication of multiple vehicles. The study's primary goal is to address the privacy and security issues that arise in VANETs, which are becoming an increasingly important part of the infrastructure of intelligent transportation systems and smart cities.

The goal of the suggested protocol is to offer a vehicle-to-infrastructure (V2I) communication solution that is lightweight, secure, and effective. PUF technology is used to improve physical security and fend off potential threats. PUFs are important because they are nearly impossible to copy or clone because of their special capacity to produce safe cryptographic keys based on the physical differences of the hardware itself.

The study starts by describing the significant computational and communication overheads connected to conventional authentication protocols in VANETs, which are frequently weak points for impersonation, replay, and eavesdropping attacks. In order to address these problems, the authors present a novel solution that creates a secure communication channel without requiring a lot of processing power by utilizing the physical security features of the PUF.

The proposed protocol's ability to drastically cut down on overall computation and communication overhead is one of its main features. This is accomplished by reducing the

need for complex cryptographic operations and optimizing the authentication procedure. The authors go into detail about the computational overhead related to different cryptographic functions that are used in the protocol, including multiplication of elliptic curve points, symmetric encryption/decryption, and one-way hash functions. They offer a comparative analysis that demonstrates how much less computational work the suggested protocol requires overall than the protocols that are currently in use.

It also discusses the communication overhead, which is another important aspect of authentication protocols' effectiveness. The authors show that, in comparison to other protocols, their protocol requires fewer bits for successful authentication by analyzing the volume of data transmitted during the authentication process. In addition to expediting the authentication process, this decrease in communication overhead also reduces bandwidth consumption, which is advantageous in the resource-constrained VANET environment.

The paper provides a detailed security comparison of the suggested protocol with current solutions, emphasizing the improved security aspects of their methodology. The protocol's robustness in securing V2I communication is highlighted by its resilience to multiple attacks, such as desynchronization and physical attacks.

In addition, a simulation run on the NS-3 simulator is included in the study to assess how well the suggested protocol performs on the network. The simulation results validate the effectiveness and reliability of the protocol in a real-world VANET environment, taking into account metrics like packet delivery ratio, throughput, and end-to-end delay. These findings support the protocol's ability to enable safe and effective communication in the quickly developing fields of intelligent transportation systems and smart cities.

The paper [27] offers a thorough investigation into improving the security of VANs, which are essential for contemporary smart transportation systems. The main objective is to create a strong foundation for safe key generation, authentication procedures, and trust metric definition for vehicles in these networks by utilizing PUFs.

The creative application of PUFs, which are essentially electronic circuits with ran-

dom physical variations inherent in their manufacturing process and capable of producing unique responses to particular challenges, lies at the core of the suggested solution. Because of their distinctiveness, PUFs are a great option for developing a dynamic, secure authentication system that is hard to copy or duplicate, which will serve as a solid basis for secure communication and vehicle authentication.

The development of multimodal protocols covering authentication, anonymity, proof of presence, and V2V exchange of ephemeral session keys is described in detail in the paper. These protocols are designed to deal with the dynamic nature of VANs, where vehicles join and exit the network frequently. This makes it necessary to have an effective and safe method for quickly establishing secure communication channels and building trust.

The introduction of PUF-based trust metrics—which are used to determine the integrity and authenticity of nearby vehicles—is a key contribution of this work. The suggested framework also protects the privacy of vehicle identities during communication by encrypting and hiding PUF responses. This is especially significant for VANs, where maintaining vehicle anonymity is essential to safeguarding drivers' and passengers' privacy.

Important elements of the suggested security framework, the V2I and V2V authentication protocols, are also presented in detail in this paper. These protocols use dynamic key exchange mechanisms and PUF-based authentication to enable secure communication between vehicles and infrastructure as well as within them.

The suggested protocols' resistance to tampering, replay, eavesdropping, impersonation, man-in-the-middle, and physical attacks is highlighted by an informal security analysis. The protocols can successfully fend off these attacks thanks to the usage of PUFs, dynamic session keys, and nonces, which offer a high degree of security for VAN communications.

A novel method for boosting the security and privacy of V2G networks is presented in the research paper [28]. Effective energy use and trading between EVs and the power grid depend on V2G networks. The paper highlights how important it is for V2G systems to have a secure, lightweight, and privacy-preserving protocol because of the physical security

breaches that could occur from EVs and charging stations being accessible to the general public.

Utilizing PUFs to enable a two-step MA process between an EV and the Grid Server is a key component of the suggested solution. PUFs are renowned for their capacity to offer a high degree of protection against physical attacks by utilizing CRPs that take advantage of the hardware's special physical properties, making it difficult to copy or duplicate.

In order to improve the privacy of the car owner by preventing the tracking of specific vehicles, the protocol introduces the idea of Pseudo IDs (PIDs) to mask the identity of the vehicle. This method makes sure that the participants' anonymity and privacy are protected during communication inside the V2G network.

Additionally, the protocol creates two unique session keys: one for the grid and the EV and another for the aggregator (charging station). Further strengthening the system's resistance to physical tampering, these session keys are exclusive to the PUFs installed on the corresponding devices, securing communication and eliminating the need to store secret keys within the EVs and aggregators.

By utilizing quick cryptographic operations, the protocol is shown in the paper to be both energy-efficient and computationally light-weight. This is especially significant when it comes to V2G networks, where prompt information sharing is essential for efficient energy trading and grid management.

In terms of security features like MA, identity protection, message integrity, and resistance to different attacks like man-in-the-middle, impersonation, and replay attacks, the RAMA protocol is examined and contrasted with other protocols that are currently in use. The comparison highlights RAMA's superiority in guaranteeing the privacy and security of V2G communications, as well as its extensive security coverage.

A novel method for securing V2G communications is presented in the research paper [29]. The authors of the paper concentrate on creating a lightweight protocol that uses the special qualities of PUFs to enable MA and secure key agreement in a V2G environment.

The energy and transportation sectors both see substantial reductions in greenhouse gas emissions thanks to the rapidly developing technology known as EVs. When EVs are incorporated into smart city infrastructure, V2G technology performs better, enabling two-way electrical energy exchange between EVs and the grid. This exchange optimizes EV charging and discharging to increase grid efficiency and lower carbon emissions, in addition to supporting the grid during periods of high demand. But the widespread use of V2G technology brings with it new difficulties, especially with regard to security. A strong security solution is required due to the physical vulnerability of user devices and the transmission of sensitive information over public channels.

The authors suggest a lightweight authentication and key agreement protocol that makes use of PUF technology in order to overcome these issues. PUFs are very resistant to physical attacks and cloning because they provide a safe and effective way to generate cryptographic keys based on a device's inherent physical properties. Without imposing a large amount of computational or communication overhead, the protocol seeks to establish a secure communication channel between different entities in the V2G network, including energy servers, charging stations, and EVs.

The paper offers a thorough security analysis that shows the protocol's resistance to common security risks like replay attacks, impersonation, and man-in-the-middle attacks using both formal and informal techniques. The protocol's benefits in terms of communication and computational efficiency are also demonstrated by comparing its performance to those of current solutions.

Methodology

The methodical process used to develop and implement FPGA-based PUFs for use in automotive applications is described in this section. The concept, design, implementation, testing, and application phases are the different parts of the methodology. Ensuring that the

developed PUFs meet the required security and reliability standards for automotive systems needs careful consideration in each phase. In order to comprehend the current state of PUF technology, particularly in relation to FPGA implementations and automotive applications, the first phase involves a thorough literature review. The primary characteristics of PUFs that make them appropriate for security applications—such as their unpredictability, uniqueness, and resilience to cloning—are highlighted in this review. Current PUF designs are examined in detail, along with their advantages, disadvantages, and compatibility for use in automotive systems. The design phase concentrates on conceptualizing PUF architectures that are well-suited for FPGA platforms used in automotive systems, relying on insights from the literature review. This involves choosing the best PUF architecture (such as an arbiter PUF, SRAM PUF, or RO PUF) based on factors like implementation complexity, response uniqueness, and reliability. Determining the PUF design specifications also takes into account the intended security level, response generation speed, and environmental stability. As a result of the design phase, the RO was chosen as a fundamental building block of the proposed design by this thesis. Also, because of the fact that the RO contains combinatorial loops, it is impossible to do a simulation. The PUF design has been coded and deployed on FPGA platforms during the implementation phase. The main tasks involve selecting an FPGA board that satisfies the performance, power consumption, and environmental tolerance requirements for automotive applications, setting up the development environment with the required tools and software, writing the VHDL code for the PUF design and synthesizing it onto the chosen FPGA board, and integrating the PUF with other automotive system components to guarantee smooth operation within the vehicle environment. For the proposed design, the TerAsic DE-10 Lite development board with Intel MAX10 FPGA was chosen as a platform. The Quartus Prime Lite software was used as a development environment. To assess the effectiveness and dependability of the deployed PUFs, extensive testing is carried out. This includes performance benchmarking (measuring the response generation time, resource utilization on the FPGA, and power consumption to ensure the PUFs meet automotive industry standards),

and uniqueness and reliability testing (evaluating the uniqueness of PUF responses and their reliability over time and across different environmental conditions).

Proposed design and Implementation

The PUF ID produced by the suggested approach is exclusive to the bitstream and one particular FPGA. On a different FPGA of the same type, the same bitstream will result in a different PUF ID. The PUF ID of a particular FPGA will change if the bitstream of that particular FPGA is altered (by altering the design logic). Our design was built on a MAX10 FPGA, and Intel Quartus Prime software was used for verification. However, it can be implemented on any FPGA because it is constructed in a technology-neutral manner without requiring any device-specific macros, primitives, or properties.

The proposed design creates 128 bits of unique, unclonable identifier or ID. Each bit is generated by an individual PUF element. Simple ROs serve as the foundation for each asynchronous component that makes up a PUF element. The RO is made up of one inverter. The output of the inverter is connected back to its input through a D latch. If the D latch is enabled, the RO starts oscillating, otherwise RO stops and the D latch stores its last state. Additionally, the D latch has a "reset" input and it is able to bring the PUF element to a defined state.

Each oscillator's frequency is determined by the routing that corresponds to its logic cell mapping. Additionally, the intrinsic semiconductor properties of the chip randomly change the frequency. These changes are brought on by minor production variations. For instance, fluctuations in oxide thickness can alter a routing wire's capacitance or delay. This change, however, is constant for a given arrangement. Consequently, each RO has its own fixed frequency. As a result, sampling repeatedly within a predetermined time frame will always result in the same output state. All oscillators are affected by temperature in essentially the same way, and post-processing must remove or correct it.

The asynchronous PUF element runs the danger of being eliminated by the synthesis toolchain or optimized into a single PUF element because the suggested architecture does not make use of any device-specific features or primitives. Therefore, in our design, each PUF cell's latch enable and reset timings are independently and gradually controlled using a shift register.

The shift register contains 129 bits. The "enable" and "reset" signals of the PUF elements are connected to this shift register. At the beginning of the ID generation process, a logic high is applied to the least significant bit of the shift register. This high bit travels until the 129th bit during the process. The *kth* bit of the shift register drives the "reset" signal of the *kth* PUF cell and the "enable" signal of the *(k - 1)th* PUF cell. It means that the "reset" signal of the cell is active for one clock cycle and in the next clock cycle the cell is enabled exactly for one clock cycle allowing for oscillations. Finally, in the next clock cycle, the D latch is disabled and the last state of the RO is recorded. As you can see, at any time instance only one PUF element is active. Therefore, the work of one PUF cell can not affect the output of another cell.

When the 129th bit of the shift register is "high", the ID generation process is completed and outputs of all 128 PUF elements are sampled to another register. This register provides a 128-bit raw ID.

The proposed design was coded using VHDL. The top entity is called *puf*. Figure 1 shows the inputs and outputs of this entity. The full code and explanations for *puf* entity are given in Appendix A.

The entity *puf* uses two components: *puf_cell* and *UART_TX*.

The *puf_cell* entity describes the PUF element that consists of a RO, D latch and sampling D FF. Figure 2 shows the inputs and outputs of this entity. The full code and explanations for *puf_cell* entity are given in Appendix B.

The PUF cell number 73 (as an example) is shown in Figure 3. This figure is taken from the Intel Quartus Prime RTL viewer. One PUF element utilizes one LUT and one FF.

```
entity puf is
   generic (
      ID_SIZE  : natural := 128; -- the size of the generated ID
      N_SAMPLES   : natural := 131072 -- number of samples
   );
   port (
      clk_i   : in  std_logic; -- clock line
      rstn_i  : in  std_logic; -- reset, low-active
      busy_o  : out std_logic; -- busy if HIGH (sampling the ID)
      tx_o  : out std_logic -- UART TX
   );
end puf;
```

Figure 1: The puf entity

```
entity puf_cell is
  port (
    clk_i    : in  std_ulogic; -- clock signal for synchronization
    reset_i  : in  std_ulogic; -- signal for resetting the cell
    latch_i  : in  std_ulogic; -- latch enable signal
    sample_i : in  std_ulogic; -- signal that triggers the sampling
    data_o   : out std_ulogic  -- output signal from the cell(raw ID bit)
  );
end puf_cell;
```

Figure 2: The puf_cell entity

The LUT is required by the RO and D latch, while FF is required for the sampling register.

The *UART_TX* entity describes the Universal Asynchronous Receiver Transmitter (UART) transmitter module used for serial communication between FPGA and the PC. Figure 4 shows the inputs and outputs of this entity. The full code and explanations for *UART_TX* entity are given in Appendix C.

Post-Processing and Setup

There were noticeable noises in certain parts of the raw PUF ID. Determining an ID that is constant over time and for fluctuations of operating circumstances, such as temperature, becomes more difficult as a result of these noises. Numerous methods exist in the literature to address this problem. ECCs are a viable method for stabilizing determined raw ID.

For the proposed design, the simplest error correction algorithm is used. The raw ID

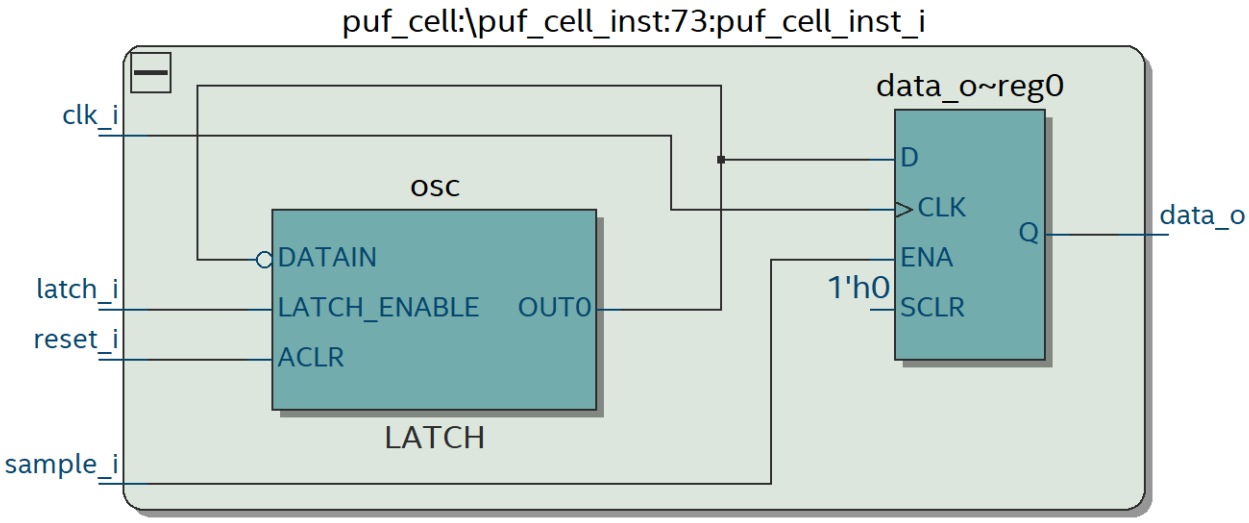

Figure 3: PUF cell number 73

```
10 entity UART_TX is
11   generic (
12     g_CLKS_PER_BIT : integer := 434     -- Needs to be set correctly
13     );
14   port (
15     i_Clk       : in  std_logic; -- clock line
16     i_TX_Tr     : in  std_logic; -- transmission trigger
17     i_TX_Byte   : in  std_logic_vector(7 downto 0); -- byte to be transmitted
18     o_TX_Active : out std_logic; -- LOW when idle state
19     o_TX_Serial : out std_logic; -- TX line
20     o_TX_Done   : out std_logic -- HIGH for one clock cycle when transmission done
21     );
22 end UART_TX;
```

Figure 4: UART_TX entity

is generated several times (for example 131072 times) and after every generation, each bit from 128 bits is checked. If most of the time (more than the threshold value) bit was set, it is considered to be 1, otherwise, it is considered to be 0.

The excerpt of the top entity architecture code with the post-processing algorithm can be seen in Figure 5 below.

After the post-processing algorithm is done, the ready 128-bit PUF ID is sent to a terminal of the PC using USB to TTL-level UART converter and UART protocol. Figure 6 shows the setup.

As a terminal emulator, an open-source, free software called Yet Another Terminal is used. The screenshot of the terminal with generated ID can be seen in Figure 7. The 128-bit

```
94    when S_SAMPLE => -- sample latch states
95       counter <= counter + 1; -- increase the number of sampled IDs
96       for i in 0 to ID_SIZE-1 loop
97          if (id(i) = '1') then
98             arr(i) <= arr(i) + 1;
99          end if;
100      end loop;
101      if (counter = N_SAMPLES - 1) then -- if enough number of IDs are sampled
102         arbiter.state <= S_FINISH;
103      else
104         arbiter.state <= S_IDLE;
105      end if;
106
107   when S_FINISH => -- post-processing
108      for i in 0 to ID_SIZE-1 loop
109         if (arr(i) >= hyst_high) then
110            f_id(i) <= '1';
111         else
112            f_id(i) <= '0';
113         end if;
114      end loop;
115      arbiter.state <= S_TX;
```

Figure 5: The excerpt of the code with the post-processing algorithm

generated ID is depicted in the hex format.

Performance Evaluation and Resource Utilization

Conventional performance metrics such as uniqueness, reliability and uniformity [30] are calculated for the generated PUF ID.

6.1 Uniqueness

The PUF's uniqueness is a measure of its ability to produce unique IDs on various devices or bitstreams. It is essential for guaranteeing the unique identification of every PUF instance. To evaluate uniqueness, the Hamming Distance is frequently employed. It is computed between the PUF responses of several devices or bitstream configurations in the same environment [31, 32].

$$Uniqueness = \frac{2}{d(d-1)} \sum_{i=1}^{d-1} \sum_{j=i+1}^{d} \frac{HD(R_i, R_j)}{n} \times 100\% \quad (1)$$

Here d is the number of devices or instances, *HD(Ri, Rj)* denotes the Hamming Distance, *R_i* and *R_j* are the responses from the *ith* and *jth* PUF instances respectively,

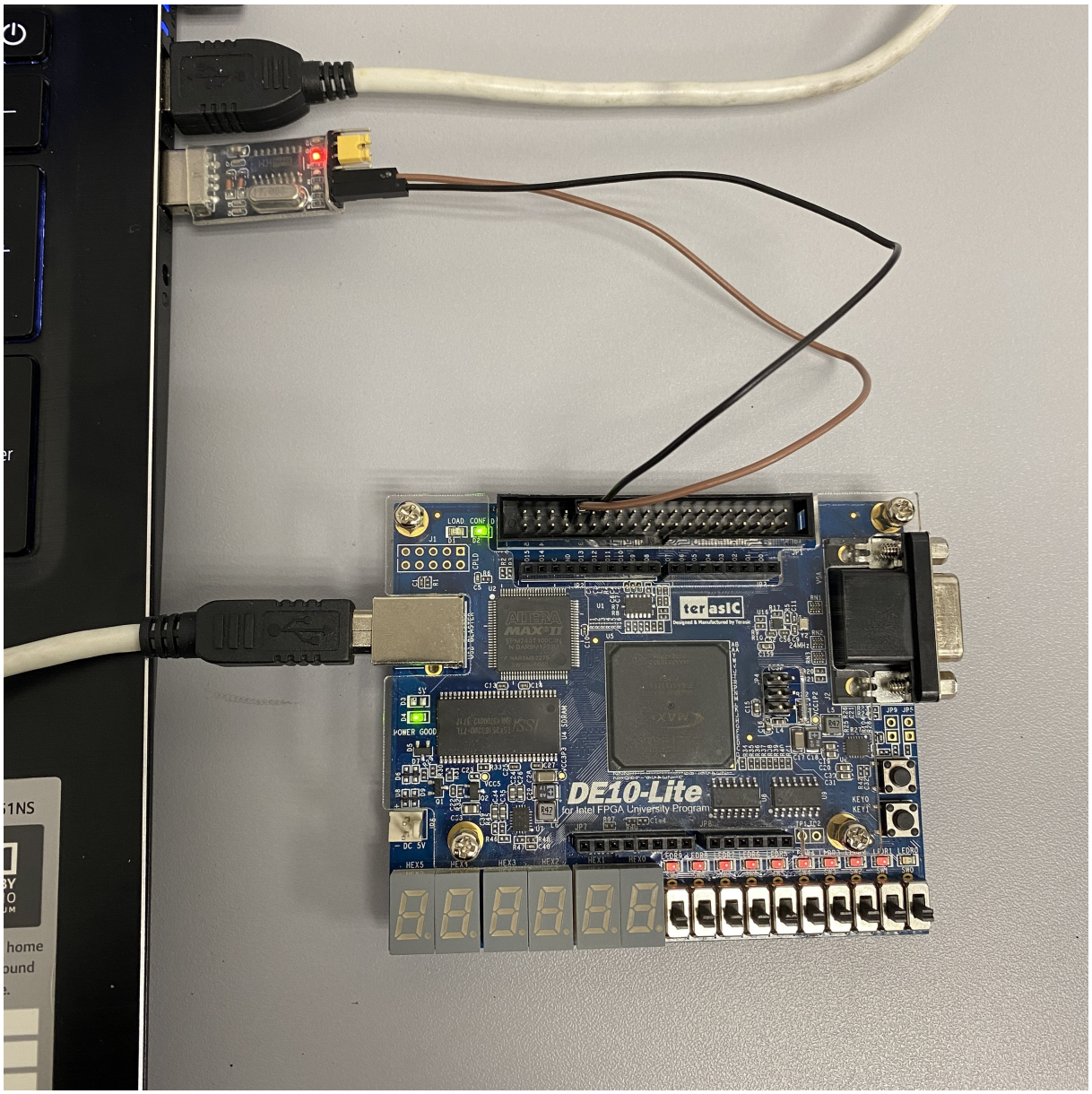

Figure 6: The FPGA to PC connection setup

and n is the length of the response in bits. A 50% Hamming distance is what a perfect PUF should have.

For evaluation of our proposed PUF ID generator, 100 IDs with 128 bit response were generated under normal operating conditions by implementing 100 different bit files for the same FPGA. As a result, the proposed design has an uniqueness of 48.61%, which is close to the ideal value.

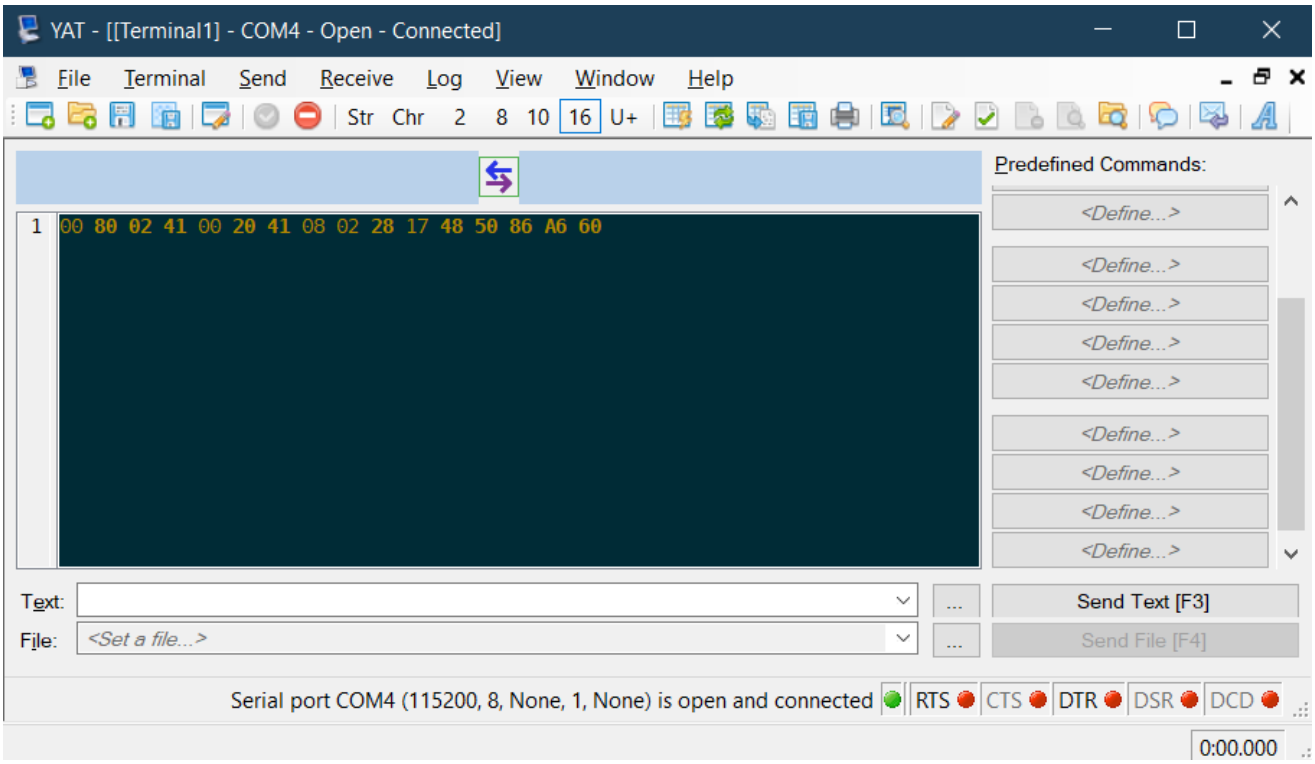

Figure 7: The screenshot of the terminal with generated ID received using UART

6.2 Reliability

The PUF's reliability is evaluated based on its ability to replicate the same ID in different scenarios, including temperature variations, voltage fluctuations, and aging. The ideal value is 100%. By calculating the Hamming distance between replies produced by the same PUF design under various circumstances but using the same device and bitstream, you may assess reliability [33].

$$Reliability = \frac{1}{x}\sum_{y=1}^{x} HD(R_i, R_j) \times 100\% \quad (2)$$

Here *HD(Ri, Rj)* denotes the Hamming Distance between the generated IDs by the same bitstream under different conditions, x is the number of ID samples.

For the proposed PUF ID generator, the reliability was calculated by obtaining 200 responses at varying temperature and supply voltage. The temperature range of -15°C to

24°C was achieved and voltage supply range of 3.5V to 5V was achieved using regulated power supply. According to the results, the proposed design achieve the reliability of 95.89%.

6.3 Uniformity

The tendency of some PUF bits to prefer a given value (0 or 1) across several PUF instances is known as uniformity. A balanced bit-aliasing, in which every bit in the PUF answer has an equal chance of being 0 or 1, is a feature of a well-designed PUF. This may be measured by counting the proportion of ones in every bit location across several PUF instances, and then comparing the results to the optimal 50% threshold [32, 33].

$$Uniformity = \frac{1}{k}\sum_{j=1}^{k}\sum_{i=1}^{n}\frac{R_{i,j}}{n} \times 100\% \tag{3}$$

Here $R\ i,j$ means the j-th bit of n-bit ID generated from PUF i and k represents the number of PUF instances.

The proposed design generates ID with uniformity of 52.78% which is close to ideal value.

6.4 Resource Utilization

The resource utilization of the proposed design can be seen in Figure 8. It should be noted that the suggested design has a low hardware overhead for the utilized area, which leads to effective and optimum system performance. Additionally, the estimated power consumption of the proposed PUF ID generator is about 0.63W. This implies that the suggested approach is particularly useful for circuits with low power consumption.

Top-level Entity Name	puf
Family	MAX 10
Device	10M50DAF484C7G
Timing Models	Final
Total logic elements	2,756 / 49,760 (6 %)
Total registers	2625
Total pins	4 / 360 (1 %)
Total virtual pins	0
Total memory bits	0 / 1,677,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 288 (0 %)
Total PLLs	0 / 4 (0 %)
UFM blocks	0 / 1 (0 %)
ADC blocks	0 / 2 (0 %)

Figure 8: The resource utilization of the proposed design on MAX10 FPGA

Conclusions and Future Work

This thesis focuses on improving security features in the automotive industry by providing an extensive analysis of the development and application of FPGA-based PUFs for automotive applications. With the increasing integration of digital and electronic systems in contemporary vehicles, there is a growing need for strong hardware security solutions, which is fueling the research effort. This study is primarily noteworthy for creating a new FPGA-based PUF architecture. As a fundamental component of hardware authentication, the suggested design takes advantage of the intrinsic manufacturing variances in FPGAs to produce distinct, unclonable IDs. Along with adhering to security standards, this design also meets the high standards set by the automotive industry for dependability and performance in a variety of environmental settings.

Key metrics, including uniqueness, reliability, and uniformity, showed satisfactory results in the performance evaluation of the implemented PUF design. The metric of uniqueness, which is essential to guarantee that every PUF instance produces a unique identifier, was found to be highly correlated with the ideal value, suggesting a high degree of individuality among PUF instances. The design's resilience to environmental fluctuations—a crucial feature for automotive applications—was demonstrated by the reliability metric, which high-

lighted the design's consistent performance across a range of operating conditions. To further attest to the design's resilience, the uniformity metric verified that the PUF responses had a balanced distribution of "0s" and "1s." An analysis of resource utilization showed that the suggested PUF design is effective in both saving FPGA resources and performing the intended security functions. Given that automotive systems are resource-constrained, this feature is especially beneficial. The PUF design's low power consumption is also consistent with the automotive industry's continuous endeavors to optimize vehicle energy consumption.

The encouraging results of this study open up a ways of possibilities for further investigation with the goal of improving and expanding the utility of FPGA-based PUFs in the automotive industry. Integrating the suggested PUF design with different automotive subsystems, such as infotainment and ECUs, is one such approach. A layer of hardware authentication can be incorporated into a variety of components to greatly improve the overall security of automotive ecosystems. Future research must also focus on the PUF design's scalability and adaptability. The architecture's applicability and efficacy in the automotive industry could be greatly increased by modifying it to support various FPGA platforms as well as the various scales and complexities of automotive systems. Furthermore, even though the current design includes a reasonably simple error correction mechanism, exploring more advanced methods may improve the accuracy of PUF responses even further. These kinds of developments are especially important considering the harsh environmental conditions that are frequently found in automotive settings. Another crucial area for future research is the crossover from controlled environments and theoretical frameworks to practical applications. Thorough testing of the PUF design in real-world automotive settings would provide priceless information about areas that need improvement and how well it works in practice. This shift could be facilitated by cooperative efforts with auto suppliers and manufacturers, guaranteeing that the suggested fixes adhere to industry norms and specifications. Moreover, the security architecture of contemporary cars may undergo a complete transformation with the creation of security protocols that take advantage of the inherent qualities of PUFs for safe

key storage, authentication, and communication within vehicular networks. These protocols have the potential to provide strong answers to some of the most important security issues that the automotive industry is currently facing by utilizing the special qualities of PUFs.

To sum up, this thesis has created a basic framework for using FPGA-based PUFs to improve the security of automobiles. Future research has exciting prospects in the areas of advanced error correction, scalability, integration strategies, real-world deployment, and PUF-centric security protocol development. In addition to promising to progress the field of automotive security, these initiatives also support the overarching objective of building safer, more secure automobiles for the digital age.

Appendices

8.1 Appendix A

```
1   library ieee;
2   use ieee.std_logic_1164.all;
3   use ieee.numeric_std.all;
4
5   entity puf is
6      generic (
7         ID_SIZE  : natural := 128; -- the size of the generated ID
8         N_SAMPLES   : natural := 131072 -- number of samples
9      );
10     port (
11        clk_i   : in  std_logic; -- clock line
12        rstn_i  : in  std_logic; -- reset, low-active
13        busy_o  : out std_logic; -- busy if HIGH (sampling the ID)
14        tx_o  : out std_logic -- UART TX
15     );
16  end puf;
17
18  architecture rtl of puf is
19
20     constant c_CLKS_PER_BIT : integer := 434;
21     constant hyst_high      : integer := (N_SAMPLES - (N_SAMPLES/8));
22     type state_t is (S_IDLE, S_RUN, S_SAMPLE, S_FINISH, S_TX);
23
24     type arbiter_t is record
25        state       : state_t;
26        sreg        : std_logic_vector(ID_SIZE downto 0);
27        sample      : std_logic;
28     end record;
29
30     signal arbiter    : arbiter_t;
31
32     type t_array is array (ID_SIZE-1 downto 0) of integer range 0 to (N_SAMPLES - 1);
33
34     signal arr     : t_array := (others => 0);
35
36     signal id         : std_logic_vector(ID_SIZE-1 downto 0) := (others => '0');
37     signal f_id       : std_logic_vector(ID_SIZE-1 downto 0) := (others => '0');
38     signal counter    : integer range 0 to (N_SAMPLES - 1)   := 0;
39     signal trig       : std_logic := '0';
40     signal tx_active  : std_logic;
41     signal done       : std_logic;
42     signal byte_counter  : integer range 0 to (ID_SIZE/8) + 1   := 0;
43     signal data       :  std_logic_vector(7 downto 0);
44
45     component puf_cell
46        port (
47           clk_i    : in  std_logic;
48           reset_i  : in  std_logic;
49           latch_i  : in  std_logic;
50           sample_i : in  std_logic;
51           data_o   : out std_logic
52        );
53     end component;
```

Figure A.1: First part of the code of entity puf

```
55      component UART_TX
56         generic(
57            g_CLKS_PER_BIT : integer := 434
58         );
59         port(
60            i_Clk       : in  std_logic;
61            i_TX_Tr     : in  std_logic;
62            i_TX_Byte   : in  std_logic_vector(7 downto 0);
63            o_TX_Active : out std_logic;
64            o_TX_Serial : out std_logic;
65            o_TX_Done   : out std_logic
66         );
67      end component;
68
69      begin
70
71      sampling: process(clk_i)
72      begin
73         if rising_edge(clk_i) then -- latch control SREG: control reset and transparent mode --
74            arbiter.sreg(arbiter.sreg'left downto 1) <= arbiter.sreg(arbiter.sreg'left-1 downto 0);
75            arbiter.sreg(0) <= '0'; -- default
76
77            if (rstn_i = '0') then -- reset
78               arbiter.state <= S_IDLE;
79               counter <= 0;
80               arr <= (others => 0);
81               byte_counter <= 0;
82               trig <= '0';
83            else
84               case arbiter.state is
85                  when S_IDLE =>
86                     arbiter.sreg(0) <= '1';
87                     arbiter.state   <= S_RUN;
88
89                  when S_RUN => -- reset & open latches for one cycle - one by one
90                     if (arbiter.sreg(arbiter.sreg'left) = '1') then -- if MSB of shift register is HIGH
91                        arbiter.state <= S_SAMPLE;
92                     end if;
93
94                  when S_SAMPLE => -- sample latch states
95                     counter <= counter + 1; -- increase the number of sampled IDs
96                     for i in 0 to ID_SIZE-1 loop
97                        if (id(i) = '1') then
98                           arr(i) <= arr(i) + 1;
99                        end if;
100                    end loop;
101                    if (counter = N_SAMPLES - 1) then -- if enough number of IDs are sampled
102                       arbiter.state <= S_FINISH;
103                    else
104                       arbiter.state <= S_IDLE;
105                    end if;
106
```

Figure A.2: Second part of the code of entity puf

```
107                 when S_FINISH => -- post-processing
108                    for i in 0 to ID_SIZE-1 loop
109                       if (arr(i) >= hyst_high) then
110                          f_id(i) <= '1';
111                       else
112                          f_id(i) <= '0';
113                       end if;
114                    end loop;
115                    arbiter.state <= S_TX;
116
117                 when S_TX => -- ID transmitting
118                    if (tx_active = '0') then
119                       if (byte_counter < ID_SIZE/8) then
120                          data <= f_id((((byte_counter + 1) * 8) - 1) downto (byte_counter * 8));
121                          trig <= '1';
122                       end if;
123                    else
124                       trig <= '0';
125                    end if;
126                    if (done = '1') then
127                       byte_counter <= byte_counter + 1;
128                    end if;
129                 when others => -- undefined
130                    arbiter.state <= S_IDLE;
131
132              end case;
133           end if;
134        end if;
135     end process sampling;
136
137     arbiter.sample <= '1' when (arbiter.state = S_SAMPLE) else '0'; -- sample PUF data
138
139     busy_o <= '1' when (arbiter.state = S_FINISH) else '0'; -- busy flag
140
141     puf_cell_inst: -- PUF elements
142     for i in 0 to ID_SIZE-1 generate
143        puf_cell_inst_i: puf_cell
144        port map (
145           clk_i    => clk_i,
146           reset_i  => arbiter.sreg(i), -- reset one cycle before enabling latch
147           latch_i  => arbiter.sreg(i+1), -- enable latch for one cycle
148           sample_i => arbiter.sample,
149           data_o   => id(i)
150        );
151     end generate;
152
153     uart_inst: UART_TX
154     generic map (
155        g_CLKS_PER_BIT => c_CLKS_PER_BIT
156     )
157     port map (
158        i_Clk       => clk_i,
159        i_TX_Tr     => trig,
160        i_TX_Byte   => data,
161        o_TX_Active =>    tx_active,
162        o_TX_Serial => tx_o,
163        o_TX_Done   =>    done
164     );
165
166  end rtl;
```

Figure A.3: Third part of the code of entity puf

8.2 Appendix B

```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity puf_cell is
  port (
    clk_i    : in  std_ulogic; -- clock signal for synchronization
    reset_i  : in  std_ulogic; -- signal for resetting the cell
    latch_i  : in  std_ulogic; -- latch enable signal
    sample_i : in  std_ulogic; -- signal that triggers the sampling
    data_o   : out std_ulogic  -- output signal from the cell(raw ID bit)
  );
end puf_cell;

architecture rtl of puf_cell is

  signal osc : std_ulogic;

begin

  oscillator: process(osc, reset_i, latch_i) -- Oscillator element
  begin
    if (reset_i = '1') then -- reset oscillator to defined state
      osc <= '0';
    elsif (latch_i = '1') then -- enable ring-oscillator; keep current state when disabled
      osc <= not osc;
    end if;
  end process oscillator;

  cap_reg: process(clk_i)  -- Output Capture Register
  begin
    if rising_edge(clk_i) then
      if (sample_i = '1') then
        data_o <= osc;
      end if;
    end if;
  end process cap_reg;


end rtl;
```

Figure B.1: The code of entity puf_cell

8.3 Appendix C

```
1  -- Set Generic g_CLKS_PER_BIT as follows:
2  -- g_CLKS_PER_BIT = (Frequency of i_Clk)/(Frequency of UART)
3  -- Example: 10 MHz Clock, 115200 baud UART
4  -- (10000000)/(115200) = 87
5  --
6  library ieee;
7  use ieee.std_logic_1164.all;
8  use ieee.numeric_std.all;
9
10 entity UART_TX is
11   generic (
12     g_CLKS_PER_BIT : integer := 434     -- Needs to be set correctly
13     );
14   port (
15     i_Clk       : in  std_logic; -- clock line
16     i_TX_Tr     : in  std_logic; -- transmission trigger
17     i_TX_Byte   : in  std_logic_vector(7 downto 0); -- byte to be transmitted
18     o_TX_Active : out std_logic; -- LOW when idle state
19     o_TX_Serial : out std_logic; -- TX line
20     o_TX_Done   : out std_logic -- HIGH for one clock cycle when transmission done
21     );
22 end UART_TX;
23
24
25 architecture RTL of UART_TX is
26
27   type t_SM_Main is (s_Idle, s_TX_Start_Bit, s_TX_Data_Bits,
28                      s_TX_Stop_Bit, s_Cleanup);
29   signal r_SM_Main : t_SM_Main := s_Idle;
30
31   signal r_Clk_Count : integer range 0 to g_CLKS_PER_BIT-1 := 0;
32   signal r_Bit_Index : integer range 0 to 7 := 0;  -- 8 Bits Total
33   signal r_TX_Data   : std_logic_vector(7 downto 0) := (others => '0');
34   signal r_TX_Done   : std_logic := '0';
35
36 begin
37
38
39   p_UART_TX : process (i_Clk)
40   begin
41     if rising_edge(i_Clk) then
42
43       case r_SM_Main is
44
45         when s_Idle =>
46           o_TX_Active <= '0';
47           o_TX_Serial <= '1';         -- Drive Line High for Idle
48           r_TX_Done   <= '0';
49           r_Clk_Count <= 0;
50           r_Bit_Index <= 0;
51
52           if i_TX_Tr = '1' then
53             r_TX_Data <= i_TX_Byte;
54             r_SM_Main <= s_TX_Start_Bit;
55           else
56             r_SM_Main <= s_Idle;
57           end if;
58
59         -- Send out Start Bit. Start bit = 0
60         when s_TX_Start_Bit =>
61           o_TX_Active <= '1';
62           o_TX_Serial <= '0';
```

Figure C.1: First part of the code of entity UART_TX

```
64            -- Wait g_CLKS_PER_BIT-1 clock cycles for start bit to finish
65            if r_Clk_Count < g_CLKS_PER_BIT-1 then
66              r_Clk_Count <= r_Clk_Count + 1;
67              r_SM_Main   <= s_TX_Start_Bit;
68            else
69              r_Clk_Count <= 0;
70              r_SM_Main   <= s_TX_Data_Bits;
71            end if;
72
73          -- Wait g_CLKS_PER_BIT-1 clock cycles for data bits to finish
74          when s_TX_Data_Bits =>
75            o_TX_Serial <= r_TX_Data(r_Bit_Index);
76
77            if r_Clk_Count < g_CLKS_PER_BIT-1 then
78              r_Clk_Count <= r_Clk_Count + 1;
79              r_SM_Main   <= s_TX_Data_Bits;
80            else
81              r_Clk_Count <= 0;
82
83              -- Check if we have sent out all bits
84              if r_Bit_Index < 7 then
85                r_Bit_Index <= r_Bit_Index + 1;
86                r_SM_Main   <= s_TX_Data_Bits;
87              else
88                r_Bit_Index <= 0;
89                r_SM_Main   <= s_TX_Stop_Bit;
90              end if;
91            end if;
92
93          -- Send out Stop bit. Stop bit = 1
94          when s_TX_Stop_Bit =>
95            o_TX_Serial <= '1';
96
97            -- Wait g_CLKS_PER_BIT-1 clock cycles for Stop bit to finish
98            if r_Clk_Count < g_CLKS_PER_BIT-1 then
99              r_Clk_Count <= r_Clk_Count + 1;
100             r_SM_Main   <= s_TX_Stop_Bit;
101           else
102             r_TX_Done   <= '1';
103             r_Clk_Count <= 0;
104             r_SM_Main   <= s_Cleanup;
105           end if;
106
107         -- Stay here 1 clock
108         when s_Cleanup =>
109           o_TX_Active <= '0';
110           r_TX_Done   <= '0';
111           r_SM_Main   <= s_Idle;
112
113         when others =>
114           r_SM_Main <= s_Idle;
115
116       end case;
117     end if;
118   end process p_UART_TX;
119
120   o_TX_Done <= r_TX_Done;
121
122 end RTL;
```

Figure C.2: Second part of the code of entity UART_TX

References

[1] V. Kalyani, M. Bhatnagar, L. Shivnani, and E. Vijay, "Significance of electronics and luxury car sophistication through ecu: A progressive study of evolution in automobile industry," *Journal of Management Engineering and Information Technology*, vol. 3, 10 2016.

[2] D. K. Nilsson and U. E. Larson, "Secure firmware updates over the air in intelligent vehicles," in *ICC Workshops - 2008 IEEE International Conference on Communications Workshops*, 2008, pp. 380–384.

[3] M. Bozdal, M. Samie, S. Aslam, and I. Jennions, "Evaluation of can bus security challenges," *Sensors*, vol. 20, pp. 16–17, 04 2020.

[4] A. Alrabady and S. Mahmud, "Analysis of attacks against the security of keyless-entry systems for vehicles and suggestions for improved designs," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 1, pp. 41–50, 2005.

[5] F. Sagstetter, M. Lukasiewycz, S. Steinhorst, M. Wolf, A. Bouard, W. R. Harris, S. Jha, T. Peyrin, A. Poschmann, and S. Chakraborty, "Security challenges in automotive hardware/software architecture design," in *2013 Design, Automation and Test in Europe Conference and Exhibition (DATE)*, 2013, pp. 458–463.

[6] J. Yang, Z. Duan, M. Wang, J. Mahmood, Y. Xiao, and Y. Yang, "An authentication mechanism for autonomous vehicle ecu utilizing a novel slice-based puf design," *Journal of New Media*, vol. 2, pp. 157–165, 01 2020.

[7] G. Aishwarya, B. Patil, P. V. Joshi, K. M. Sudarsham, K. Vaidyanathan, P. Parandkar, and A. Dsouza, "A survey on use of fpga in automotive system," in *2022 International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, 2022, pp. 51–56.

[8] J. Peng, L. Tian, X. Jia, H. Guo, Y. Xu, D. Xie, H. Luo, Y. Shan, and Y. Wang, "Multi-task adas system on fpga," in *2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2019, pp. 171–174.

[9] M. Gabrick, R. Nicholson, F. Winters, B. Young, and J. Patton, "Fpga considerations for automotive applications," 04 2006.

[10] N. N. Anandakumar, M. S. Hashmi, and S. K. Sanadhya, "Efficient and lightweight fpga-based hybrid pufs with improved performance," *Microprocessors and Microsystems*, vol. 77, p. 103180, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0141933120303471

[11] N. N. Anandakumar, M. S. Hashmi, and M. Tehranipoor, "Fpga-based physical unclonable functions: A comprehensive overview of theory and architectures," *Integration*, vol. 81, pp. 175–194, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167926021000766

[12] Z. He, W. Chen, L. Zhang, G. Chi, Q. Gao, and L. Harn, "A highly reliable arbiter puf with improved uniqueness in fpga implementation using bit-self-test," *IEEE Access*, vol. 8, pp. 181 751–181 762, 2020.

[13] E. Avaroğlu, "The implementation of ring oscillator based puf designs in field programmable gate arrays using of different challenge," *Physica A: Statistical Mechanics and its Applications*, vol. 546, p. 124291, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378437120300868

[14] C. Labrado and H. Thapliyal, "Hardware security primitives for vehicles," *IEEE Consumer Electronics Magazine*, vol. 8, no. 6, pp. 99–103, 2019.

[15] R. P. Parameswarath and B. Sikdar, "A puf-based lightweight and secure mutual authentication mechanism for remote keyless entry systems," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 2022, pp. 1776–1781.

[16] C. Gu, N. Hanley, and M. O'neill, "Improved reliability of fpga-based puf identification generator design," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 10, no. 3, may 2017. [Online]. Available: https://doi.org/10.1145/3053681

[17] M. A. Hamza, H. H. Issa, and S. Eisa, "Fpga-based modified ring oscillator physical unclonable function for internet of vehicles," in *2023 40th National Radio Science Conference (NRSC)*, vol. 1, 2023, pp. 208–216.

[18] N. N. Anandakumar, M. S. Hashmi, and M. A. Chaudhary, "Implementation of efficient xor arbiter puf on fpga with enhanced uniqueness and security," *IEEE Access*, vol. 10, pp. 129 832–129 842, 2022.

[19] M. Kaveh, D. Martín, and M. R. Mosavi, "A lightweight authentication scheme for v2g communications: A puf-based approach ensuring cyber/physical security and identity/location privacy," *Electronics*, vol. 9, no. 9, 2020. [Online]. Available: https://www.mdpi.com/2079-9292/9/9/1479

[20] Q. Xie, Z. Sun, Q. Xie, and Z. Ding, "A cross-trusted authority authentication protocol for internet of vehicles based on blockchain," *IEEE Access*, vol. 11, pp. 97 840–97 851, 2023.

[21] N. N. Anandakumar, M. Hashmi, and S. Sanadhya, "Design and analysis of fpga based pufs with enhanced performance for hardware-oriented security," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 18, 02 2022.

[22] Aishwarya, F. Syed, J. Nupur, A. Vichare, and A. Mishra, "Authentication of electronic control unit using arbiter physical unclonable functions in modern automobiles," in *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*, ser. ICTCS '16. New York, NY, USA: Association for Computing Machinery, 2016. [Online]. Available: https://doi.org/10.1145/2905055.2905328

[23] J. Petit, C. Bösch, M. Feiri, and F. Kargl, "On the potential of puf for pseudonym generation in vehicular networks," in *2012 IEEE Vehicular Networking Conference (VNC)*, 2012, pp. 94–100.

[24] M. Asim, J. Guajardo, S. S. Kumar, and P. Tuyls, "Physical unclonable functions and their applications to vehicle system security," in *VTC Spring 2009 - IEEE 69th Vehicular Technology Conference*, 2009, pp. 1–5.

[25] T. Cultice and H. Thapliyal, "Puf-based post-quantum can-fd framework for vehicular security," *Information*, vol. 13, no. 8, 2022. [Online]. Available: https://www.mdpi.com/2078-2489/13/8/382

[26] S. Awais, W. Yucheng, K. Mahmood, M. Akram, S. Hussain, A. K. Das, and Y. Park, "Puf-based privacy-preserving simultaneous authentication among multiple vehicles in vanet," *IEEE Transactions on Vehicular Technology*, vol. PP, pp. 1–14, 01 2023.

[27] F. Gebali and M. K. Elhadad, "Pufguard: Vehicle-to-everything authentication protocol for secure multihop mobile communication," *Computers*, vol. 12, no. 11, 2023. [Online]. Available: https://www.mdpi.com/2073-431X/12/11/233

[28] G. Bansal, N. Naren, and V. Chamola, "Rama: Real-time automobile mutual authentication protocol using puf," in *2020 International Conference on Information Networking (ICOIN)*, 2020, pp. 265–270.

[29] A. G. Reddy, P. R. Babu, V. Odelu, L. Wang, and S. AP Kumar, "V2g-auth: Lightweight authentication and key agreement protocol for v2g environment leveraging physically unclonable functions," *IEEE Transactions on Industrial Cyber-Physical Systems*, vol. 1, pp. 66–78, 2023.

[30] H. Ning, F. Farha, A. Ullah, and L.-F. Mao, "Physical unclonable function: Architectures, applications and challenges for dependable security," *IET Circuits, Devices and Systems*, vol. 14, 03 2020.

[31] S. Hou, Y. Guo, and S. Li, "A lightweight lfsr-based strong physical unclonable function design on fpga," *IEEE Access*, vol. 7, pp. 64 778–64 787, 2019.

[32] A. Aguirre, M. Hall, T. Lim, J. Trinh, W. Yan, and F. Tehranipoor, "A systematic approach for internal entropy boosting in delay-based ro puf on an fpga," in *2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2020, pp. 623–626.

[33] Y. Cui, C. Gu, C. Wang, M. O'Neill, and W. Liu, "Ultra-lightweight and reconfigurable tristate inverter based physical unclonable function design," *IEEE Access*, vol. 6, pp. 28 478–28 487, 2018.