University of Windsor Scholarship at UWindsor

Electronic Theses and Dissertations

Theses, Dissertations, and Major Papers

2018

Physical Unclonable Function Reliability on Reconfigurable Hardware and Reliability Degradation with Temperature and Supply Voltage Variations

Manpreet Kaur University of Windsor

Follow this and additional works at: https://scholar.uwindsor.ca/etd

🔮 Part of the Electrical and Computer Engineering Commons

Recommended Citation

Kaur, Manpreet, "Physical Unclonable Function Reliability on Reconfigurable Hardware and Reliability Degradation with Temperature and Supply Voltage Variations" (2018). *Electronic Theses and Dissertations*. 7487.

https://scholar.uwindsor.ca/etd/7487

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

Physical Unclonable Function Reliability on Reconfigurable Hardware and Reliability Degradation with Temperature and Supply Voltage Variations

By

Manpreet Kaur

A Thesis

Submitted to the Faculty of Graduate Studies

through the Department of Electrical and Computer Engineering

in Partial Fulfillment of the Requirements for

the Degree of Master of Applied Science

at the University of Windsor

Windsor, Ontario, Canada

2018

© 2018 Manpreet Kaur

Physical Unclonable Function Reliability on Reconfigurable Hardware and Reliability Degradation with Temperature and Supply Voltage Variations

by

Manpreet Kaur

APPROVED BY:

J. Defoe

Department of Mechanical, Automotive & Materials Engineering

H. Wu

Department of Electrical and Computer Engineering

R. Muscedere, Co-Advisor

Department of Electrical and Computer Engineering

R. Rashidzadeh, Co-Advisor

Department of Electrical and Computer Engineering

June 14, 2018

DECLARATION OF CO-AUTHORSHIP / PREVIOUS PUBLICATION

I. Co-Authorship

I hereby declare that this thesis incorporates material that is the result of my research studies, as follows: Chapter 3 of this thesis includes an accepted paper for publication. The key ideas, primary contributions, experimental designs, data analysis, interpretation, and writing were performed by both the author and coauthor.

I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my thesis, and have obtained written permission from each of the co-author(s) to include the material(s) in my thesis.

I certify that, with the above qualification, this thesis, and the research to which it refers, is the product of my own work.

II. Previous Publication

This thesis includes one original paper that has been accepted for publication in a peer reviewed conference, as follows:

Chapter	Publication Title	Publication
		Status
Chapter -3	M. Kaur, R. Rashidzadeh, and R. Accepted	
	Muscedere, "Reliability of	

Physical Unclonable Function	
under Temperature and Supply	
Voltage Variations" in 61st IEEE	
International Midwest	
Symposium on Circuits and	
Systems (MWSCAS), 2018.	

I certify that I have obtained a written permission from the copyright owner(s) to include the above published material(s) in my thesis. I certify that the above material describes work completed during my registration as a graduate student at the University of Windsor.

III. General

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owners to include such materials in my thesis.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

A hardware security solution using a Physical Unclonable Function (PUF) is a promising approach to ensure security for physical systems. PUF utilizes the inherent instance-specific parameters of physical objects and it is evaluated based on the performance parameters such as uniqueness, reliability, randomness, and tamper evidence of the Challenge and Response Pairs (CRPs). These performance parameters are affected by operating conditions such as temperature and supply voltage variations. In addition, PUF implementation on Field Programmable Gate Array (FPGA) platform is proven to be more complicated than PUF implementation on Application-Specific Integrated Circuit (ASIC) technologies. The automatic placement and routing of logic cells in FPGA can affect the performance of PUFs due to path delay imbalance.

In this work, the impact of power supply and temperature variations, on the reliability of an arbiter PUF is studied. Simulation results are conducted to determine the effects of these varying conditions on the CRPs. Simulation results show that \pm 10% of power supply variation can affect the reliability of an arbiter PUF by about 51%, similarly temperature fluctuation between -40 °C and +60 °C reduces the PUF reliability by 58%. In addition, a new methodology to implement a reliable arbiter PUF on an FPGA platform is presented. Instead of using an extra delay measurement module, the Chip Planner tool for FPGA is used for manually placement to minimize the path delay misalignment to less than 8 ps.

DEDICATION

Education is not the learning of the facts,

But the training of the mind to think!

-Albert Einstein

I dedicate my work to my loving parents, Mr. Bhupinder Singh and Mrs. Harjeet Kaur; my brothers, Manipal Singh and Damanpal Singh; and my friend, Preet Inder Singh. Thank you for always being there and their love. Thank you for the indestructible wealth of all time. I would also like to thank my supervisor, Dr. Rashidzadeh for his guidance and support.

ACKNOWLEDGEMENTS

I would like to sincerely thank my supervisor, Dr. Rashid Rashidzadeh, for his guidance and support in successfully completing my thesis. I am deeply grateful for his involvement, guiding, mentoring and providing any help that I needed to complete my degree. It is an honor to have worked under his supervision.

I am grateful to my co-supervisor, Dr. Roberto Muscedere for his support and valuable comments which helped in completing this thesis.

I would also like to thank my committee members, Dr. Huapeng Wu and Dr. Jeff Defoe for their encouragement, constructive comments and positive criticism which in fact, improved my ideas and solutions.

I would like to extend my gratitude to my colleagues at the Research Centre for Integrated Microsystems (RCIM). I appreciate their friendship, support, encouragement, their constant involvement and valuable feedbacks.

Finally, I would like to thank the research and financial support received from Natural Sciences and Engineering Research Council (NSERC) of Canada and Canadian Microelectronics Corporation (CMC) Microsystems.

TABLE OF CONTENTS

DECLARATION OF CO-AUTHORSHIP / PREVIOUS PUBLICATION iii
ABSTRACTvi
DEDICATIONvii
ACKNOWLEDGEMENTSviii
LIST OF FIGURESxii
LIST OF TABLExvi
LIST OF ABBREVIATIONS/SYMBOLSxvi
Chapter-1 Introduction1
1.1 Motivation
1.2 Security and Test
1.2.1 Software Security
1.2.2 Hardware Security
1.3 What is PUF?
1.3.1 The PUF Concept7
1.3.2 Challenge and Response7
1.4 Research Objective
1.5 Outline of Thesis9
1.6 References
Chapter-2 Physical Unclonable Function Concept and Background12
2.1 Previous Work12

2.2 PUF Construction	14
2.3 PUF Properties	14
2.3.1 Uniqueness	14
2.3.2 Reliability	15
2.3.3 Randomness	16
2.3.4 Tamper Evidence (Security)	17
2.4 PUF Hamming Distance	17
2.4.1 Intra-distance	17
2.4.2 Inter-distance	
2.5 Classification of PUFs	
2.6 PUF Implementation	20
2.6.1 Arbiter PUF	20
2.6.2 Lightweight Secure PUF	21
2.6.3 Feed Forward Arbiter PUFs	22
2.6.4 Ring Oscillator PUFs	23
2.6.5 SRAM PUFs	24
2.7 PUF Applications	26
2.7.1 Low-Cost Authentication	26
2.7.2 Cryptographic Key Generation	
2.8 Summary	29
2.9 References	
Chapter-3 Reliability of Physical Unclonable Function under Tem	perature and
Supply voltage variations	
3.1 Introduction	

VITA A	AUCTORIS	
APPEN	DIX: IEEE PERMISSION TO REPRINT	71
5.3 F	uture Work	70
5.2 C	onclusion	70
5.1 S	ummary	
Chapte	r-5 Conclusions and Future Work	69
4.7 R	eferences	
4.6 C	onclusion	67
4.5 E	xperimental Results	
4.4 D	esign Requirements and Implementation	
4.3 C	yclone II LE Architecture	53
4.2 A	ltera EP2C5	53
4.1 Ir	troduction	
Chapte	r-4 APUF Implementation on FPGA Platform	
3.6	References	
3.5	Conclusion	
3.4	Simulation Results	
3.3	Effect of Supply voltage variation on PUF Reliability	
3.2	Effect of Temperature Variation on PUF Reliability	

LIST OF FIGURES

Figure 1.1 Hardware threat points
Figure 1.2 Challenge and response pair system
Figure 2.1 Unique response for even same challenge applied15
Figure 2.2 Reliability to achieve same response for same challenge applied under
fluctuating environmental conditions16
Figure 2.3 Classification of PUFs19
Figure 2.4 APUF basic model
Figure 2.5 Lightweight Secure PUF structure
Figure 2.6 Feed-forward arbiter PUF23
Figure 2.7 RO PUF structure
Figure 2.8 (a) Logic circuit of SRAM PUF (b) Electrical circuit of SRAM cell25
Figure 2.9 Applications of PUF26
Figure 2.10 Unclonable RFIDs approach to authentication27
Figure 2.11 Cryptographic key generation using PUF [29]28
Figure 3.1 Effect of setup/hold time on proper operation of arbiter
Figure 3.2 (a) A two port network representing a multiplexer. (b) Its equivalent noise
free circuit. (c) The equivalent circuit connected to a source

Figure 3.3 Effect of noise on propagation delay of a multiplexer. (a) Without noise
where the propagation delay is fixed. (b) With noise where the propagation delay
varies between Min and Max values
Figure 3.4 Noise response of a multiplexer. (a) Power spectral density. (b) Noise
nower (c) rms noise voltage 42
Figure 3.5 Simulation results to evaluate the effect of process variations on the
propagation delay of a multiplexer. (a) Output response of a MUX to an input
indicating different propagations delay for FF, TT and SS corners. (b) Output
response of a delay line containing ten multiplexers indicating propagation delay
variations at different corners
Figure 3.6 Effect of supply voltage variation on propagation delay of logic gates. (a)
A multiplexer. (b) A delay line containing ten multiplexers
Figure 3.7 Reliability reduction for an arbiter PUF due to 10% supply voltage
variation46
Figure 3.8 Propagation delay versus temperature for a delay line containing 10
multiplexers
Figure 3.9 Reduction in area due to temperature variation47
Figure 3.10 Simple circuit of supply-independent
Figure 4.1 Cyclone II logic element structure [4]

Figure 4.2 (a) Elementary symbol of delay cell (b) Symbol compact pair of 2-1 MUX
whose inputs are cross connected and selectors are connected together55
Figure 4.3 RTL view of 64-bit challenge with 1- bit response
Figure 4.4 Resource property editor window view of delay cell and DFF57
Figure 4.5 Effect of routing at different sections
Figure 4.6 Random placing of MUX and arbiter element on the chip planner59
Figure 4.7 Design after partition in LLR60
Figure 4.8 Result of one manual placement61
Figure 4.9 Cyclone II FPGA board used for testing
Figure 4.10 Result of propagation delay for random and fixed 16-bit PUF63
Figure 4.11 Probability of observing '1' as response after locking them at fixed
places for five PUFs
Figure 4.12 Probability of observing '1' as response bit for random challenges
versus 5 response bits
Figure 4.13 Hamming distance of output response bits

LIST OF TABLE

Table 1: Comparison	with pervious v	work)
---------------------	-----------------	------	---

Abbreviations/Symbols	Description
API	Application Programming Interface
APUF	Arbiter PUF
С	Challenge
CPUF	Controlled PUF
CRP	Challenge Response Pair
DFF	Delay Flip-Flop
DL	Delay Line
DP	Design Partition
ECC	Error Correcting Code
FF	Fast-Fast
FPGA	Field-Programmable Gate Array
HD	Hamming Distance
IC	Integrated Circuit
I/O	Input/Output
IP	Intellectual Property

LIST OF ABBREVIATIONS/SYMBOLS

IoTs	Internet of Things
JTAG	Joint Test Active Group
LABs	Logic Array Blocks
LE	Logic Element
LFSR	Linear Feedback Shift Register
LLR	Logic Locked Region
LUT	Look Up Table
Max	Maximum value
Min	Minimum value
MUX	Multiplexer
OS	Operating System
PAR	Placement-and-Routing
PDL	Programmable Delay Lines
РОК	Physically Obfuscated Key
PUF	Physical Unclonable Functions
PVT	Process Voltage Temperature
R	Response

RFIDs	Radio Frequency Identifications
ROPUF	Ring Oscillator PUF
rms	Root Mean Square
RTL	Register Transfer Level
SRAM	Static Random Access Memory
SS	Slow-Slow
TERO	Transient Element Ring Oscillators
TT	Typical-Typical
USB	Universal Serial Bus

Chapter -1

Introduction

Internet of Things (IoTs) is a network of physical devices which enables data exchange in real-time between objects. IoT has emerged as one of the promising technologies that has the potential to affect the lives of billions of people. However, the potential of IoT technology will not be fully materialized if a robust solution is not developed for IoT security. Due to the large number of connecting IoT devices, their wide applications and impact on everyday life, even a minor security breach can be a major problem. For example, if a connected car is hacked by a third party, it can be remotely controlled which may result in unwanted acceleration and deceleration and in worst cases, loss of life. The security needs to be built inside the design and each IoT device has to be uniquely identified [1].

There is at present a rapid increase in susceptibilities per devices due to the dependency on the technology such as software installed on devices. The main concerns are listed as follow [2]:

1. **Privacy concerns:** Research reports indicate that about 90% of devices hold a minimum of one piece of private information either due to the device or the applications installed on it. It can include name, address, date of birth, usernames, passwords, and credit card information.

- 2. Insufficient Authentication: Most of the people save their passwords in their devices and these passwords can get stored on the cloud. These passwords are usually user's name or date of birth which can easily be hacked by simple techniques.
- **3.** Transport Encryption: Encryption is a technique to encode the information, however, about 70% of devices use unencrypted network services.
- **4. Web Interface:** It is the interface between a user and the software running on the server. Signing web pages using default credentials can help hackers to identify valid credentials and take advantage of them.

The software based security alone may not be sufficient to handle security concerns of IoT enabled networks. Hardware of a product at the design stage needs to be secure and be reinforced by software level security.

1.1 Motivation

The cost of in-house fabrication is too expensive and most companies rely on a foreign semiconductor foundries for IC fabrication. This potentially poses a security threat since the factories are provided with the design details and they can secretly add hardware infection to the main design. Encryption is a known method which is widely used for secure data transmission [3]. A private key is used for encryption, however, it is susceptible to malicious attacks and the private key can be retrieved by adversaries through timing and power consumption analysis known as side channel attacks [4].

A solution for the above mentioned problem is to generate the key on the fly to make the system secure and resilient against attacks. Physical Unclonable Function (PUF) is a hardware security module which can be used to generate secret keys for authentication. By

using these secure keys, PUF can also be used to develop security solutions for portable wireless communication devices including IoT sensors. This can be done by not storing the secrets in the smart meters and instead generating them at the time when authentication is required using PUF technology [5].

1.2 Security and Test

Security of communication networks is a top priority. In the modern digital world, the traditional security techniques are no longer sufficient to adequately enable trusted interaction [6]. The need for security among information processing and communication systems have risen exponentially because confidential information is stored over the Internet and other networks.

1.2.1 Software Security

Software security is a system which involves building secure mechanisms and making system robust. In other words, security is implemented to protect the software against hacking or other malicious attacks. Malicious harmful files or a programs are used to suspect the user's information, which includes virus, worms, trojan and spyware.

A Virus is a piece of code which copies itself and becomes part of the code. It spreads from one device to another, it can damage the data of the device and also get attached to executable files. Worms are similar to viruses and cause damage to data. Worms are standalone software, so do not require a host program to run. Trojans are another type of malware which can delete or modify data, steal personal information, or activate other malware. They can also create hidden doors for an attacker to access the system. Spyware is like a trojan which collects the user information without his/her awareness. Once installed on a device, it can monitor user's activities and gather credentials.

In addition, these malicious attacks can affect the normal operation of a system and if the infected system is in a network, other systems in that network are susceptible to an attack through the infected system. However, there are many security solutions against these malwares such as antivirus, encryption, firewall and spyware removal software [7]. This security software is limited to software-based attacks and cannot resist the hardware attacks which can significantly affect the operation of a system and requires the study and need to secure hardware systems.

1.2.2 Hardware Security

Hardware security refers to the protection of physical system(s) against malicious attacks. A hardware security module is a physical system used to safeguard and maintain strong authentication. It is believed that hardware is more secure as compared to software. Some of the reasons are as follows:

- 1. Hardware is well tested and certified in laboratories.
- 2. There is a limited access to the system and it is strictly controlled by internal rules.
- 3. It has security-focused operating system. OS is a software which manages the hardware and software of the device and also provide some common services.

However, if an attacker can disassemble the hardware module, then important information can be extracted. A number of attacks on hardware are reported through micro probing, reverse engineering and side channel analysis. Moreover, outsourcing presents a security threat as well.



Figure 1.1 Hardware threats points

As mentioned in Section 1.1 that due to high cost of fabrication, design is sent out of company. Due to the outsourcing of the design, trust on the security of hardware built can be lost at any point. For example, to complete the hardware module, design moves from one point to another as shown in Figure 1.1. IP core is a block of data or logic which provides more options for designing. However, the security issue comes across the third-party IP cores where it can be misused. A system integrator is a person or a company that build systems for customers by combining hardware, software, and products from multiple vendors. Security of the system integrator risks due to the smart devices present on the floor through which information can be leaked to unwanted systems. Manufacturer is an arrangement to produce goods like integrated circuits. Manufacturer can fabricate more number of ICs or can modify the design by adding wanted circuit which can leak the

information. A large number of side-channel attacks and trojan injection has been reported at these points. Hence, it is necessary to prevent ICs from the third party theft, piracy and overproduction.

However, there are some methods such as watermarking and hardware metering to stop the overproduction of chips and to make them secure against malicious attacks. These techniques and their respective disadvantages are explained below.

- Hardware metering was introduced against the overproduction and piracy of the IC's. It is a set of security protocols enabled by IP owner to achieve post-fabrication control. It helps the owner to track and identify the design post fabrication. Using the active metering, not only the identification can be performed but also the designer can lock or unlock the functionality. The drawback of passive metering method is that it is not strong enough to protect designs against overproduction [8].
- **Digital Watermark** is a mark used to identify the ownership or to verify the authenticity of a carrier signal. However, this technique is vulnerable to attacks. A possible threat is that the same data can be watermarked multiple times and distributed. Fraudulent data can be created to combine different watermarked copies [6].
- **Hardware Trojan** has emerged as a major concern for IC manufacturing. It is a hidden circuit inserted by adversaries in a chip to modify existing circuits and affect their functionalities, reduce the reliability or extract valuable information. A hardware module which is infected by a Trojan can leak confidential information [9].

1.3 What is PUF?

A physical unclonable function (PUF) is a hardware security module which is generally considered as a random variable. It relies on physical characteristics of integrated circuits (ICs) to generate unique signatures. It is a one-way function in which an output is produced by an input, but the input cannot be determined from the output. PUF was first introduced by Pappu in 2002, when he observed the unique speckle patterns on a transparent epoxy wafer filled with bubbles upon shining it with a laser [10]. Silicon based PUF was then introduced by Gassend et al. [11].

1.3.1 The PUF Concept

The fundamental concept behind the PUF technology is the variation that occurs in an IC characteristic due to the process variations during the manufacturing. The manufacturing process variations are minute, unavoidable, completely random and difficult to control. PUF takes advantage of these random variations to generate a unique and random set for each fabricated device. PUF behaves like a random function which generates random values. These random values are unpredictable for an attacker even if they have physical access.

1.3.2 Challenge and Response

The input of a PUF is called challenge and the output generated is called response. PUF is interrogated by a set of challenges to generate a unique set of responses. The set of challenge-response pairs (CRPs) is used to uniquely identify a device. Figure 1.2 shows the block diagram of a PUF. PUF core circuit is used to generate output when input is applied to it, where input and output randomizer are to add security. An error correction

scheme is implemented to correct the errors of the raw responses. These error correction scheme can be hamming codes or syndrome codes [12].



Figure 1.2 Challenge and Response pair system

1.4 Research Objective

The objective of this research is to analyze the reliability of an arbiter PUF under supply and temperature variation. In addition, the effect of rigid FPGA structure of PUF implementation is studied.

The key contributions from this thesis are:

- 1. The impact of supply voltage and temperature variations cannot be ignored as it can affect the performance parameters of PUF significantly. Moreover, these variations can also reduce the number of reliable CRPs of a PUF to less than 42%.
- 2. To implement a circuit like PUF on FPGA requires symmetry, which can be done by manual placing the circuit elements. In addition, the location and the delay imbalance can be observed using Chip Planner instead of using an extra hardware module. The positions of the logic elements can be fixed using LogicLock Region and Design Partition within the Chip Planner.

1.5 Outline of Thesis

This thesis is organized as follows:

- In Chapter 2, The published research works in the field of PUF and its implementation methods are covered. Then PUF structure, different properties of PUF, classification of existing PUFs along with their circuits and its application are explained.
- In Chapter 3, A delay based arbiter PUF is implemented in the Cadence Virtuoso environment using 0.18µm CMOS technology to study the effects of supply and temperature variations on PUF performance parameters. The simulation results are presented in the form of meaningful waveforms and plots for PUF performance evaluation. Additionally, we analyzed the reduction of challenge response pairs (CRP) set under the varying operating conditions. The results out of this work has been accepted for publication in the 61st IEEE International Midwest Symposium on Circuits and Systems (MWSCAS) 2018.
- In Chapter 4, An APUF is implemented on an FPGA platform to analyze the effects of FPGA rigid structure on its performance parameters. To minimize the effect, a new practically feasible and easy to implement method has been presented.
- In Chapter 5, This chapter summarizes the results and presents potential area of research for future work.

1.6 References

 Tankard, Colin, "The Security Issues of the Internet of Things," Computer Fraud & Security, vol. 2015, no. 9, 2015, pp. 11–14.

[2] P. V. Liesdonk, S. Sedghi, J. Doumen, P. Hartel, and W. Jonker, "Computationally Efficient Searchable Symmetric Encryption," Lecture Notes in Computer Science Secure Data Management, pp. 87–100, 2010.

[3] M. Kassner, "Computer-stored encryption keys are not safe from side-channel attacks," TechRepublic. [Online]. Available: https://www.techrepublic.com/article/computer-stored-encryption-keys-are-not-safe-from-side-channel-attacks/. [Accessed: 02-April-2018].

[4] R. Maes, Physically Unclonable Functions Constructions, Properties and Applications. Berlin: Springer Berlin, 2016.

[5] M. Delavar, S. Mirzakuchaki, M. H. Ameri, and J. Mohajeri, "PUF-Based Solutions for Secure Communications in Advanced Metering Infrastructure (AMI)," International Journal of Communication Systems, vol. 30, no. 9, Dec. 2016.

[6] "What is the Difference: Viruses, Worms, Trojans, and Bots?" Cisco, 20 Nov. 2017, https://www.cisco.com/c/en/us/about/security-center/virus-differences.html.

[7] F. Koushanfar, "Hardware Metering: A Survey," Introduction to Hardware Security and Trust, pp. 103–122, Aug. 2011.

[8] M. Tehranipoor, F. Koushanfar (2010) A survey of hardware Trojan taxonomy and detection. IEEE Design Test Comput 27:10-24.

10

[9] S. Bhunia, M. S. Hsiao, M. Banga, and S. Narasimhan, "Hardware Trojan Attacks: Threat Analysis and Countermeasures," Proceedings of the IEEE, vol. 102, no. 8, pp. 1229–1247, 2014.

[10] B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas, "Silicon physical random functions," in CCS, 2002, pp. 148-160.

[11] Circuit Cellar. "Protect IoT Designs with PUF Circuitry." Circuit Cellar, Circuit Cellar, 28 Mar. 2018, circuitcellar.com/cc-blog/protect-iot-designs-with-puf-circuitry/.

[12] S. Devadas, and M. Yu, "Secure and Robust Error Correction for Physical Unclonable Functions." IEEE Design & Test of Computers, 2010, pp. 48-65.

Chapter -2

Physical Unclonable Function

Concept and Background

PUF is an expression of an inherent and unclonable instance-specific feature of a physical object. It is a physically disordered system which is intentionally designed to become a function of process variation to produce random response when offered an input challenge. The security of a PUF is based on wire delays, gate delays, and quantum mechanical fluctuations. PUF response values are observed at three dimensions in an array: -

- (i) Responses from different PUF instances,
- (ii) Responses from the same PUF but on different challenges, and
- (iii) Responses from the same PUF on same challenges.

2.1 Previous Work

The idea of using complex unclonable features of a physical system using mesoscopic physics of coherent light transport for security measures was discovered in 2002 by Pappu et al. in [1]. Then Gassend et al. in [2] exploited delay variations and measured transient responses to generate multiple CRPs for identification and authentication of an IC. They proposed linear arbiter PUF which operates based on the race between the rising edges and this is more explained in Section 2.6.1. However, arbiter PUF was vulnerable to numerous attacks, for instance, invasive attacks, in which attacker tries to remove the package and layers of IC, or non-invasive attacks, in which adversary determines the key by observing

the power or by simulating the IC. Further work done by Majzoobi et al. in [3] introduced the formal methodology for testing the security of PUFs using four different test methods such as (1) predictability, (2) collision, (3) sensitivity, and (4) reverse-engineering. The offline and software-based testing paved the way for understanding the PUF. Maiti et al. in [4] extended the work by adding evaluation of parameters such as reliability, bit-aliasing and probability of misidentification.

PUFs are not only implemented on Application Specific Integrated Circuit (ASIC) [5] but also on FPGA platform. The advantage of implementing design on FPGA platform is that it is easy to modify. Morozov et al. in [6] and Majzoobi et al. in [7] implemented delaybased PUFs on FPGAs and found the implementation challenging due to the routing constraints and arbiter element violation. Majzoobi et al. in [8] proposed a novel approach of using programmable delay lines (PDL) against asymmetries in routing on FPGA. For the delay measurement, they used timing characterization circuit, by sweeping the clock frequency they monitored the rate of timing error then added tuning blocks to cancel out the biasing caused by routing constraints and achieved 9ps on average resolution for each inverter. Takanori et al. in [9] proposed 3-1 double arbiter PUF to improve the uniqueness of responses to approximately 50%. To improve the reliability under noisy condition, Yuejiang et al. in [10] used machine learning algorithm by selectively choosing CRPs. They successfully improved the reliability to 96.91% under the same setting.

However, the detail analysis on how various operating conditions such as temperature and supply voltage variations, can affect the reliability and CRP's of PUF has not been reported. Therefore, we performed a detailed analysis on these operating conditions and their effect on PUF performance. The result of this work has been presented in Chapter 3. Literature presented PUF implementation on FPGA platform, which requires an extra hardware module to measure the delay imbalance. What if, the measurement delay module is needs calibration or is not ideal. Therefore, we presented a new methodology to implement APUF on FPGA platform in chapter 4, without the requirement of an extra delay measurement module.

2.2 PUF Construction

The claim to construct unclonable instance is discovered in the fabrication technique limitations of physical objects. These variations occurring at (sub) microscopic level with high accuracy can be used to distinguish physical objects by generating random signatures [3]. PUF takes the advantage of these variations to generate unique random values and secret keys.

2.3 PUF Properties

PUF performance is evaluated based on four important parameters, namely uniqueness, reliability, randomness, and security (tamper evident). These performance parameters are discussed below.

2.3.1 Uniqueness

Uniqueness is a basic property of PUF which is used to ensure no two PUF chips and their CRP's are identical even if they have the same design layout and technology. Figure 2.1 indicates that even if the same challenge sets are applied to different PUFs, the response generated by them will be different. Ideally, the CRPs need to be 100% uncorrelated across the chip. The hamming distance (HD) between the responses obtained from different



instances of PUF is used to evaluate a PUF. For an ideal PUF circuit, the hamming distance need to be 0.5 (i.e. 50%) between the obtained responses. To achieve a high level of uniqueness, large process variation is required with minimum systematic biasing to make it unpredictable [11], [12], [13].

2.3.2 Reliability

Reliability is the ability to generate the same response for a known challenge even in the presence of noise and other operating condition variations as shown in Figure 2.2. The probability of getting the same response ideally needs to be 100%. The delay and the power consumption of a circuit are a function of the supply voltage and temperature fluctuations, which can affect the CRPs. These fluctuations can result in a different response set for the same challenge applied to a given PUF instance. Reliability can be measured by determining the intra-Hamming Distance (HD) of a PUF because intra-distance is calculated for the same challenge applied to the same PUF instance. Ideally, inter-chip HD



Figure 2.2 Reliability to achieve same response for same challenge applied under fluctuating environmental conditions

should be zero, it means that the total number of bit errors rate for CRP should be zero at any stage even at varying operating conditions [14], [15].

2.3.3 Randomness

Randomness is a measure of unpredictability of a PUF response. It can be measured based on how biased the response bits are towards '1' or '0' in the entire response data set and in the different slices of the data set. Minimum systematic biasing in the circuit results in a high value of randomness. Hence, the probability of achieving '1' or '0' at the output needs to be 50%.

2.3.4 Tamper Evidence (Security)

The key concept of PUF is that it is impossible to build a duplicate of a PUF instance. Under tampering, permanent changes can be made to the integrity of a PUF entity. Tamper evident for PUF stands for the fact that if reverse engineering or micro-probing is done on a PUF, the PUF gets damaged to such an extent that it starts producing wrong responses for the same set of input challenges.

2.4 PUF Hamming Distance

Hamming Distance (HD) is a parameter to calculate the number of different elements of two strings of the same length. Inter-device and intra-device distances are the two important metrics which are used to categorize the uniqueness and robustness of a PUF responses. Variation in surrounding conditions such as temperature, supply voltage and aging can affect the inter-device and intra-device distances between the PUF responses. Large inter-device and small intra-device distances are required to achieve an ideal PUF [16].

2.4.1 Intra-hamming distance

A PUF response intra-hamming distance is a distance between two PUF responses from the same PUF instance using same challenge. In an ideal case, HD between the responses should differ by 50% of the total response set, on changing a bit in the challenge set. The intra-HD can be calculated by:
$$HD_{intra} = \frac{1}{k} \sum_{i=1}^{k-1} \frac{HD(R_{i,1}, R_{i,2})}{n} \times 100\%$$
(2-1)

Where k and n are the number of chips and response bits, respectively. $R_{i,1}$ and $R_{i,2}$ are the responses for challenges C1 and C2 from chip 'i', correspondingly.

2.4.2 Inter-distance

A PUF response inter-distance is a distance between two PUF responses from different PUF instances using the same challenge. Inter-hamming distance between responses is used to measure the uniqueness of responses. Inter-hamming distance can be calculated by:

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

Where n is the number of bits (responses), $R_{i,j}$ are the response vectors of two chips, *i* and *j*; and *k* is the number of experiments.

2.5 Classification of PUFs

Figure 2.3 shows the classification of PUFs. Based on the number of challenge-response pairs (CRPs) set, PUFs are classified into weak and strong PUFs. A weak PUF is a type of PUF which is interrogated with a small set of challenges. An extreme case of a weak PUF is the one which has only a single challenge such as physically obfuscated key (POK). Whereas, a strong PUF refers to a PUF which supports an exponentially large number of challenge set. It is practically infeasibility to build an accurate model of PUF based on observed CRPs. In this case, if an adversary is given unlimited access to a PUF instance



Figure 2.3 Classification of PUFs

for a prolonged period of time, it is still almost impossible for an attacker to determine the PUF responses.

Weak PUFs offer better mechanism to generate secret keys and are hard to attack using invasive techniques. Typical examples of weak PUFs are SRAM-PUFs [17] and Coating PUFs [16]. On the other side, strong PUFs are not susceptible for modeling attacks and therefore, are ideally appropriate for IC identification, fingerprinting and secret key generation. Typical example of this type of PUFs are arbiter PUF [18], feed-forward arbiter PUF [19], lightweight secure PUF [20] and optical PUF [16].

Whereas, Controlled PUF (CPUF) is a type of PUF which can only be accessed through a specific Application Programming Interface (API). The limitation of the strong PUFs are that adversary can freely apply the challenge to get the response. CPUF resolve this limitation by restricting the access by using control algorithm. The existing PUF technology has successfully solved the authentication and secure key generation, but still

has some untapped potential such as sure bootstrapping which has been solved under emerging PUF concept by timed authentication PUF, public models PUF [22].

2.6 PUF Implementation

2.6.1 Arbiter PUF

Arbiter PUF (APUF) is a type of delay-based silicon PUF. The idea behind the arbiter PUF is to exploit the propagation delay variation of delay lines to produce a unique response. It is composed of two parallel delay lines with N number of delay cells as shown in Figure 2.4. Each delay cell is implemented as a pair of 2-1 multiplexers connected in series whose selected lines are connected together. It operates based on a race condition between the rising edges in the delay paths. The arbiter element, which is usually D Flip-Flop, determines which signal arrived first and correspondingly respond with 0 or 1.



Figure 2.4 APUF basic model

If both paths are designed to have nearly identical normal delays, the result of the race and arbiter element cannot be unambiguously determined due to the effect of random silicon process variations on the delay parameters. In case if both the delays are nearly identical and two edges are applied simultaneously on both paths which will reach the arbiter element at the same moment. This condition causes the arbiter circuit to go into a metastable state, i.e. the logic output of the arbiter circuit is temporarily undermined but after a time when arbiter leaves metastable state and respond with random binary value which is independent of the outcome of the race. This condition can cause unreliability of the responses of an arbiter.

2.6.2 Lightweight Secure PUF

The lightweight secure PUF is similar to arbiter PUF but the challenge bit passes through complicated mapping to increase security. The structure is composed of an input logic network, interconnect network, parallel arbiter PUFs and an output logic network, as shown in Figure 2.5. The input network consists of XOR gates to generate different combinations of challenge bits to each of the PUFs. Similarly, the output network also consists of XOR gates to combine responses from different PUFs. The advantage of the lightweight secure PUF is that it is resistive to reverse engineering and emulation attacks due to the confusion and diffusion properties of hash functions.



Figure 2.5 Lightweight Secure PUF structure

2.6.3 Feed Forward Arbiter PUFs

Feed forward arbiter PUF works similar to arbiter PUF by exploiting the delay variations. The difference in feed forward arbiter PUF is that some of the challenges are the result of racing conditions and are determined with the help of arbiters. As shown in Figure 2.6 shows the output of intermediate MUXs on the signal paths are the input to so-called feed-forward arbiter. The output of this arbiter is then fed to the input of another MUX forward on the signal path. It overcomes the limitation of an arbiter PUF due to the feed-forward arbiter as it is susceptible to software modeling attacks [21].



Figure 2.6 Feed-forward arbiter PUF

2.6.4 Ring Oscillator PUFs

Figure 2.7 shows the typical block diagram of a ring oscillator. It consists of two multiplexers to select ring oscillators for pairwise comparisons. The basic principle of the ring oscillator is to compare the frequencies of different on chip ring oscillators to generate CRPs. The output of the MUX is provided to the counter. After comparing the value of the two counters, the response bit is generated. If the frequency of the first ring oscillator becomes greater than the second ring oscillator, then the response bit is considered as '1', otherwise '0'. The limitation of RO PUF is the limited number of CRPs. Therefore, it is restricted to secret key generation for ICs where the response bits are used internally.



Figure 2.7 RO PUF structure

2.6.5 SRAM PUFs

SRAM-PUF is a type of weak PUF which utilizes deep submicron variations that occurs during semiconductor fabrication. The model of SRAM PUF is shown below, Figure 2.8 (a) shows two cross-coupled inverters each built from two MOSFETs, one from p-MOS and one from n-MOS. The logic memory functionality of SRAM cell is used from these inverters where Figure 2.8 (b) shows the common CMOS implementation with six MOSFETs transistors. Due to these variations, transistors properties become random. When an SRAM is turned ON, every time it has its own preferred state due to the random difference in the threshold voltages of transistors. However, an adversary can obtain the fingerprints by exposing SRAM array to a high voltage and temperature as presented in [16].



(a)



Figure 2.8 (a) Logic circuit of SRAM PUF (b) Electrical circuit of SRAM cell

2.7 PUF Applications

Based on the properties of PUF, a PUF can be used for reliable identification, authentication, key storage, and other security applications such as Internet of Things (IoT) device authentication as shown in Figure 2.9 [22], [23].



Figure 2.9 Applications of PUF

2.7.1 Low-Cost Authentication

Figure 2.10 shows the scenario of a chip authentication using PUF. The Challenge Response Pairs (CRPs) are securely stored in a database of each instance of PUF. To authenticate an IC, a random set of CRP is selected from the database and applied to the

IC. The response generated is compared with the response stored in the database to authenticate the IC. To prevent man-in-middle attacks, challenges are only used once in the system and then deleted from the database. Therefore, it is necessary to have either a large number of CRPs or a platform to regenerate new CRPs for the system.



Figure 2.10 Unclonable RFIDs approach to authentication

Mutual authentication to secure RFID tags by utilizing PUF and Linear Feedback Shift Register (LFSR) to identify both readers and tags successfully is presented in [24]. To securely activate IC and user authentication, PUF is used as proposed in [25]. Pier et al. in [26] presents PUF-based RFID tags for authentication and used error-correcting codes (ECC) to support the use of real PUF tags.

2.7.2 Cryptographic Key Generation

A cryptographic primitive requires every bit of a key to remain constant. However, PUF cannot guarantee 100% reproducibility of a response for an applied challenge under varying environmental conditions. Figure 2.11 shows firstly, an ECC technique used to ensure that a PUF produces the same output under environmental condition variation. Secondly, the cryptographic key is generated by converting the PUF output response using a key generation.



Figure 2.11 Cryptographic key generation using PUF [27]

In the first step of initialization, an output response is generated from PUF circuit and then the error encoding syndrome for the generated response is computed and saved. To regenerate the key, firstly response is generated from PUF circuit and this response is fed to the ECC decoding along with the syndrome from initialization step to correct any error if required. In addition, syndrome reveals PUF delay circuit output information. For cryptographic operations, the output of the ECC can be simply hashed to desired length of k to generate key.

2.8 Summary

In this chapter, we have discussed the properties and implementation of different PUF circuits. We provide simplified explanation of the work which has been reported in literature. The study shows that PUF is subjected to operating conditions such as power supply, temperature and noise. However, a detail analysis was required to conclude that how much these conditions can affect the performance parameters and CRPs of a PUF. To understand the PUF at practical stage, it requires fabrication of a chip. To avoid the long fabrication process, study analyzed the PUF and its performance on FPGA platform. Due to the rigid constrains of FPGA, literature highlighted the need of delay measurement module to measure the imbalance of the implemented circuit. The issue with this measurement module is that it can also be affected by the rigid constrains of the FPGA and can lower the accuracy of the system. Hence, to resolve the issue of rigid constrains and measurement module, we presented a new easy method to implement PUF circuit on FPGA platform without the requirement of an extra module. We used the available tools of the FPGA to read and minimize the misalignment.

2.9 References

[1] S. Morozov, A. Maiti, and P. Schaumont, An analysis of Delay based PUF Implementations on FPGA. Springer, 2010, pp. 382-387.

[2] B. Gassend, D. Clarke, M.Van Dijk, and S. Devadas, "Silicon physical random functions," Proc. Ninth ACM Conf. Computer and Comm. Security (CCS '02), pp. 372-373, 2002.

[3] A. Moradi, A. Barenghi, T. Kasper, and C. Paar, "On the vulnerability of FPGA bitstream encryption against power analysis attacks," in ACM Conference on Computer and Communications Security, 2011, pp. 111-124.

[4] A. Maiti, V. Gunreddy, and P. Schaumount, "A systematic method to evaluate and compare the performance of physical unclonable functions," in Embedded Systems Design with FPGAs. Springer, 2013, pp. 245-267.

[5] M. Majzoobi, F. Koushanfar, and M. Potkonjak, "Techniques for design and implementation of secure reconfigurable PUFs," TRETS, vol. 2, no. 1, pp. 1-33, 2009.

[6] P. Chen, Y. Hsiao, Y. Chung, W. X. Tsai, and J. Lin, "A 2.5-ps Bin Size and 6.7ps Resolution FPGA Time-to-Digital Converter Based on Delay Wrapping and Averaging," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 1, pp. 114-124, 2017.

[7] M. Majzoobi, F. Koushanfar, and S. Devdas, "FPGA PUF using programmable delay lines," in IWIFS, 2010, pp. 1-6.

[8] T. Machida, D. Yamamoto, M. Iwamoto, and K. Sakiyama, "A New Mode of Operation for Arbiter PUF to Improve Uniqueness on FPGA," Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, 2014.

30

[9] Y. Wen, and Y. Lao, "Enhancing PUF Reliability by Machine Learning," 2017IEEE International Symposium on Circuits and Systems (ISCAS), 2017.

[10] R. Maes, Physically Unclonable functions Constructions, Properties and Applications, Springer Berlin, 2016.

[11] A. Vijaykumar, V. C. Patil, and S. Kundu, "On testing Physically Unclonable Functions for Uniqueness," in 2016 17th International Symposium on Quality Electronics design (ISQED), 2016.

[12] D. Yamamoto, K. Sakiyama, M. Iwamoto, K. Ohta, T. Ochiai, M. Takenaka, and K. Itoh, "Uniqueness Enhancement of PUF Responses Based on the Locations of Random Outputting RS Latches," Cryptographic Hardware and Embedded Systems – CHES 2011, pp. 390-406, 2011.

[13] M. Parusiński, S. Shariati, D. Kamel, and F. Xavier-Standaert, "Strong PUFs and Their (physical) Unpredictability," Proceedings of the Workshop on Embedded Systems Security - WESS 13, 2013

[14] MD. T. Rahman, F. Rahman, D. Forte, and M. Tehranipoor, "An Aging-Resistant RO-PUF for Reliable Key Generation," IEEE Transactions on Emerging Topics in Computing, vol. 3, pp. 335-348, 2016.

[15] Y. Wen, and Y. Lao, "Enhancing PUF Reliability by Machine Learning," 2017IEEE International Symposium on Circuits and Systems (ISCAS), 2017.

[16] M. Tehranipoor, and C. Wang, "Security based on Physical Unclonability and Disorder," Introduction to Hardware Security and Trust, Sringer, 2011.

[17] A. Roelke, and M. R. Stan, "Attacking an SRAM-based PUF through Wearout,"2016 IEEE Computer Society Annual Symposium on VLSI, pp. 206-211, 2016.

[18] A. Spenke, R. Breithaupt, and R. Plaga, "An Arbiter PUF Secured by Remote Random Reconfigurations of an FPGA," Trust and Trustworthy Computing Lecture Notes in Computer Science, 2016, pp. 140-58.

[19] R. Kumar, and W. Burleson, "Side-Channel Assisted Modeling Attacks on Feed-Forward Arbiter PUFs Using Silicon Data," Radio Frequency Identification. Security and Privacy Issues Lecture Notes in Computer Science, 2015, pp. 53-67.

[20] M. Majzoobi, F. Koushanfar, and M. Potkonjak, "Lightweight Secure PUFs," in Proc. IEEE/ACM Int. Conf. Comput-Aided Des., 2008, pp. 670-673.

[21] T. Idriss, H. Idriss, and M. Bayoumi, "A PUF-based Paradigm for IoT Security,"2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), pp. 700-705, 2016.

[22] C. Herder, M. Yu, F. Kousanfar, and S. Devdas, "Physical Unclonable Functions and Applications: A Tutorial," Proceedings of the IEEE, vol. 102, no. 8, 2014, pp. 1126-1141.

[23] G.E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in ACM/IEEE Design Automation Conference, pp. 9-14, 2007.

[24] L. Kulseng, Z. Yu, Y. Wei and Y. Guan, "Lightweight Mutual Authentication and Ownership Transfer for RFID Systems," in IEEE INFOCOMM 2010.

[25] J. Huang and J. Lach, "IC Activation and User Authentication for Security-Sensitive systems," in IEEE International Workshop on Hardware-Oriented Security and Trust, HOST 2008.

[26] P.F. Cortese, F. Gemmiti, B. Palazzi, M. Pizzonia and M. Rimondini, "Efficient and Practical Authentication of PUF-Based RFID Tags in Supply Chains," in IEEE International Conference on RFID-technology and Applications, June 2010.

32

[27] G.E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in ACM/IEEE Design Automation Conference, pp. 9-14, 2007

Chapter -3

Reliability of Physical Unclonable Function under Temperature

and Supply Voltage Variations

3.1 Introduction

Security for communication networks has been a top priority for system designers and policy makers. In the past, the hardware was considered the source of trust while the application layer presumed the main source of security concerns. Security measures were commonly implemented using software to protect systems. However, the notion that hardware is the source of trust is not valid anymore due to the outsourcing of semiconductor manufacturing services. Physical layer security will become increasingly important as the threats of infected hardware continue to grow.

Physical Unclonable Function (PUF) has emerged as a basic hardware security primitive. A PUF is intentionally designed to become a function of process variation to generate a unique signature. It is interrogated by a set of challenges to generate a set of unique responses. Every challenge set for a particular PUF has to generate a unique signature which depends primarily on the fabrication process randomness. PUF can be utilized for reliable identification, authentication, key storage and other security applications [1, 2] such as Internet of Things (IoT) authentication.

The design objective for a PUF may not be fully met due to fluctuation of supply voltage, temperature and aging. Among the factors affecting PUF behavior, supply voltage and

temperature variations are important and cannot be ignored. Supply voltage variation over time for certain applications such as IoT sensors is unavoidable. PUF performance is evaluated based on several parameters such as unclonability, unpredictability, randomness, robustness, sensitivity, and reliability [3-5]. The effects of varying temperature on the unpredictability and stability of a PUF are discussed in [6]. A Programmable Delay Lines (PDL) is proposed in [6] to reduce the noise impact on the PUF responses. The effects of temperature variation and aging on PUF stability and reliability are discussed in [7-9]. However, a detailed analysis to quantify how temperature and supply voltage fluctuations affect a PUF reliability has not been presented in these works.

Different types of PUF have been reported in the literature such as Arbiter-PUF [10], Ring-Oscillator-PUF [10, 11], and SRAM-PUF [12]. A typical arbiter PUF exploits the propagation delay variation to produce a unique response. An arbiter PUF is composed of two parallel delay lines as shown in Figure 2.4. It operates based on a race between the speeds of rising edges in the delay paths. The arbiter element determines which signal arrived first and correspondingly respond with 0 or 1. In addition, PUF reliability depends on the performance parameters of the arbiter. For instance, the setup-time and the holdtime of the arbiter have a considerable impact on the reliability. As shown in Figure 3.1, for a reliable response the data has to be stable for a certain period of time known as setup time before the rising edge of the clock and must remain stable for a time period known as hold-time after the rising edge.



Figure 3.1 Effect of setup/hold time on proper operation of arbiter

It is a common practice to design and fabricate a PUF first and then take necessary measures to compensate the effects of supply and temperature variations on its performance parameters. However, such a method cannot always solve the problem. If the effects of supply voltage and temperature variations become comparable to the effects of process variations on propagation delay, the available measures to compensate them cannot easily solve the problem. A PUF has to be designed to minimize the effects of supply voltage and temperature to ensure its reliability.

The rest of the paper is organized as follows. The analytical analysis of temperature effect on PUF reliability is covered in section 3.2. Section 3.3 presents the impact of supply voltage variations on PUF reliability. Simulation results are presented in section 3.4 and section 3.5 summarizes the conclusions.

3.2 Effect of Temperature Variation on PUF Reliability

The propagation delay of a delay cell changes with temperature due to circuit noise. Consequently, the reliability of a PUF is affected by temperature variations. To clarify how circuit noise affects the behavior of a PUF, a multiplexer which is the building block of an arbiter PUF is analyzed. Figure 3.2 (a) shows a noisy multiplexer and its equivalent two port network in which the sources of noise are referred to the input. The spectral power density of the thermal noise in Figure 3.2 (b) is represented by a current source, $\overline{I_n^2}$, and a voltage source, $\overline{v_n^2}$. If a source, V_s , with resistance of R_s is connected to the multiplexer as indicated in Figure 3.2 (c), the input voltage can be calculated from:

$$\overline{v_{in}^2} = \left(\overline{v_s^2} + \overline{v_n^2} + \overline{I_n^2} R_s^2\right) \left(\frac{R_{in}}{R_{in} + R_s}\right)^2$$
(3-1)

where R_{in} is the input resistance of the two port network. Thus, the equivalent noise power spectral density, $\overline{v_{n,eq}^2}$, can be determined from:

$$\overline{v_{n,eq}^2} = \left(\overline{v_n^2} + \overline{I_n^2} R_s^2\right) \left(\frac{R_{in}}{R_{in} + R_s}\right)^2 \tag{3-2}$$

To determine the power of noise at the multiplexer's input, the equivalent noise has to be integrated over the bandwidth. Assuming parasitic capacitance of C_p at the multiplexer's input, the noise power, P_n , is given by $P_n = \int_0^\infty \overline{v_{n,eq}^2} df$ from which the noise root mean square, $v_{n,rms}$, can be calculated. The presence of noise voltage at the multiplexer's input corrupts the switching time and alters the propagation delay.



Figure 3.2 (a) A two port network representing a multiplexer. (b) Its equivalent noise free circuit. (c) The equivalent circuit connected to a source

In an ideal case, for a logic gate, the switching happens when the input signal crosses the threshold of $V_{DD}/2$ as indicated in Figure 3.3 (a). However, in the presence of noise, as shown in Figure 3.3 (b), the switching can happen in an interval defined by:

$$V_{min} < V_{switching} < V_{max} \tag{3-3}$$

where

$$V_{min} = \frac{V_{DD}}{2} - v_{n,peak} \text{ and } V_{max} = \frac{V_{DD}}{2} + v_{n,peak}$$
(3-4)

 $v_{n,peak}$ is the peak value of the noise voltage. Assuming normal distribution, the noise peak value and rms value are related by:

$$v_{n,peak} \approx 3v_{n,in,rms} \tag{3-5}$$

Therefore, the propagation delay for the multiplexer cannot be defined by a fixed value. It can take a random value within a time interval depending on the instantaneous level of the noise voltage.

Figure 3.4 shows the result of noise simulation for a multiplexer implemented in Cadence environment using CMOS 0.18µm technology. Figure 3.4 (a) shows the power spectral density of the noise at the multiplexer output. The power of the noise, which is obtained through integration of the noise spectral density, is shown in Figure 3.4 (b) and the rms noise voltage is indicated in Figure 3.4 (c). It can be seen that the rms noise voltage for the implemented multiplexer exceeds 1mV which can affect the propagation delay considerably.



Figure 3.3 Effect of noise on propagation delay of a multiplexer. (a) Without noise where the propagation delay is fixed. (b) With noise where the propagation delay varies between Min and Max values



(b)



(c)

Figure 3.4 Noise response of a multiplexer. (a) Power spectral density. (b) Noise power. (c) rms noise voltage.

3.3 Effect of Supply Voltage Variation on PUF Reliability

In this section, the effect of supply voltage variations on propagation delay of delay cells in an arbiter PUF is evaluated. A basic multiplexer and a delay line are used to show propagation delay variation with \pm 10% supply voltage fluctuations. The output resistance of the multiplexer varies significantly during the input transition. Neglecting the second order effects, the average output resistance can be determined by [13]:

$$R_{eq} \approx \frac{3}{4} \frac{V_{DD}}{I_{DSAT}} \left(1 - \frac{7}{9} \lambda V_{DD}\right)$$
(3-6)

where

$$I_{DSAT} = k' \frac{W}{L} \left((V_{DD} - V_T) V_{DSAT} - \frac{V_{DSAT}^2}{2} \right)$$
(3-7)

It can be seen from (6) that the average output resistance is a function of V_{DD} . During the output transition, the parasitic capacitance at the output node, C_p , is charged through V_{DD} with a time constant defined by $R_{eq} C_p$. Simulation results in section 3.4 indicate that the effect of supply voltage variations can be comparable to the effects of process variations on propagation delay of logic gates.

3.4 Simulation Results

To analyze the effect of temperature and supply voltage variations on PUF reliability, a basic arbiter PUF with 10 delay cells was implemented in Cadence environment using CMOS 0.18µm technology to evaluate the PUF reliability. A module consisting of 10 delay cells is enough to perform simulations and estimate the average delay per cell. Further, the result is verifies through the normal distribution.

The implemented arbiter PUF shown in Figure 2.4, was used to perform simulations and to evaluate the effect of process variations. Figure 3.5 presents the results of corner analysis at different corners of fast-fast (FF), typical-typical (TT), and slow-slow (SS), where fast and slow corner exhibit higher and lower carrier mobilities than typical, i.e. normal. It can be seen that the propagations delay varies up to 31.504ps for a single multiplexer and 378.2ps for the delay line. The circuit used to evaluate the effect of process variations on the propagation delay was used to determine the effects of temperature and supply variations. Figure 3.6 shows simulation results for $\pm 10\%$ supply voltage variation. It can



Figure 3.5 Simulation results to evaluate the effect of process variations on the propagation delay of a multiplexer. (a) Output response of a MUX to an input indicating different propagations delay for FF, TT and SS corners. (b) Output response of a delay line containing ten multiplexers indicating propagation delay variations at different corners.



Figure 3.6 Effect of supply voltage variation on propagation delay of logic gates. (a) A multiplexer. (b) A delay line containing ten multiplexers.

be seen that the propagation delay of the delay line varies by 87.426ps. Assuming a normal distribution for the propagation delay due to the process variation with $6\sigma = 378$ ps, the effect of supply variation on the reliability can be calculated. It can be seen in Figure 3.7 that the area under the Gaussian distribution which indicates the PUF reliability reduces to less than 49% due to 10% supply voltage variation. Therefore, supply voltage fluctuations can reduce the set of reliable PUF CRPs considerably.



Figure 3.7 Reliability reduction for an arbiter PUF due to 10% supply voltage variation

Simulation results for temperature variations in Figure 3.8 indicate that the propagation delay increases as the temperature rises. It can be seen that the propagation delay of delay line varies by 101.7ps when the temperature rises from -40°C to +70°C. The effects of temperature variations on the PUF reliability can be determined by using the same method used to calculate the effects of supply voltage variation. As shown in Figure 3.9, in this case the area under the Gaussian distribution drops to lower than 42% which indicates that the PUF reliability is effects by temperature variations significantly.



Figure 3.8 Propagation delay versus temperature for a delay line containing 10

multiplexers.



Figure 3.9 Reduction in area due to temperature variation

Simulation results indicate that both temperature and supply voltage variations can undermine the reliability of an arbiter PUF considerably to the degree that they cannot be easily compensated after fabrication. To minimize these effects, the PUF circuity has to be designed properly. The first stage in a delay line has the maximum effect on the overall output noise. Therefore, it is reasonable to utilize low noise design techniques to implement a low noise delay cell as the first stage in a delay line for an arbiter PUF. There are also known circuit design methodologies to implement supply voltage independent circuits.

Figure 3.10 shows a simple circuit of supply-independent biasing, which includes two current mirrors using PMOS and NMOS transistors. The idea is if I_{REF} does not change with respect to V_{DD} , then I_{out} remain independent of supply voltage [14].



Figure 3.10 Simple circuit of supply-independent

Using both negative Temperature Coefficient (TC) circuits and positive TC circuits [14] a temperature independent gate can be designed. Such temperature independent gates can be utilized to design reliable PUFs.

3.5 Conclusion

This paper presents a detail analysis on how supply voltage and temperature variations can undermine PUF reliability and reduce the set of reliable challenges and responses. Simulation results using Cadence environment indicate that the set of CPRs can decrease by more than 51% if the supply varies by $\pm 10\%$. Likewise, temperature variations from - 40° C to $\pm 70^{\circ}$ C reduces a PUF reliability significantly by more than 58%. Such a considerable reduction of reliability cannot be readily fixed after fabrication. Circuit design techniques have to be employed to address the problem prior to fabrication. To lower the effect of temperature, the first stage of delay lines in a PUF has to be designed as a low noise delay cell.

3.6 References

[1] M. Tehranipoor, and C. Wang, "Security based on Physical Unclonability and Disorder," Introduction to Hardware Security and Trust, Sringer, 2011.

[2] B. Halak, M. Zwolinsi, and M. Syafiq Mispan, "Overview of PUF-Based Hardware Security Solutions for the Internet of Things," IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS), 2016.

[3] Maiti, V. Gunreddy, and P. Schaumont, "A systematic method to evaluate and compare the performance of physical unclonable functions," Proc. Embedded System Design FPGAs, 2013, pp. 245-267.

[4] Y. Hori, T. Yoshida, T. Katashita, and A. Satoh, "Quantitative and statistical performance evaluation of arbiter physical unclonable functions on FPGAs," Proc. IEEE Int. Conf. Reconfigurable Comput. FPGAs, 2010, pp. 298-303.

[5] T. Bryant, S. Chowdhury, D. Forte, M. Tehranipoor, and N. Maghari, "A Stochastic
 Approach to Analog Physical Unclonable Function," IEEE 59th International MWSCAS,
 2016.

[6] S. U. Hussain, M. Majzoobi, and F. Koushanfar, "A Built-in-Self-Test Scheme for Online Evaluation of Physical Unclonable Functions and True Random Number Generators," IEEE Transactions on Multi-scale Computing Systems, vol. 2, no. 1, 2016.

[7] G. Brussenskiy, and J. S. Yuan, "Robust PUF Circuit Design against Temperature Variations and Aging Effects," International Conference on Security and Management (SAM), 2015, pp. 211-216.

50

[8] T. Xu, and M. Potkonjak, "Stable and Secure Delay-based Physical Unclonable Functions using device aging," 2015 IEEE International Symposium on Circuits and Systems (ISCAS), 2015, pp. 33-36.

[9] C. Gu, N. Hanley, and M. O'Neill, "FPGA-based strong PUF with increased uniqueness and entropy properties," 2017 IEEE International Symposium on Circuits and Systems (ISCAS), 2017, pp. 1-4.

[10] G. E. Suh, and S. Devadas, "Physical Unclonable Functions for Device Authentication and Secret Key Generation" Design Automation Conference, 44th ACM/IEEE, 2007, pp. 9-14.

[11] Z. Chen, Y. Cai, Q. Zhou, and G. Qu, "An efficient framework for configurable RO
PUF," 2016 IEEE International Symposium on Circuits and Systems (ISCAS), 2016, pp.
742-744.

[12] C. Boit, C. Helmeier, D. Nedospasov, and J. P. Seifert "Cloning Physically Unconable Functions," IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), 2013, pp. 1-6.

[13] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, "The CMOS Inverter," Digital Integrated Circuits, Pearson, 2002.

[14] B. Razavi, Design of Analog CMOS Integrated Circuits, Thomas Casson, 2001.

[15] Crepaldi, Paulo Cesar, et al. Low-Voltage, Low-Power Vt Independent VoltageReference for Bio-Implants. INTECH Open Access Publisher, 2012.

Chapter-4

APUF Implementation on FPGA Platform

Implementing PUF on FPGAs platform has been addressed in many research papers. To the greatest, PUF performance parameters for different types of PUF have been studied. However, few papers focus on a practical solution for PUF implementation on FPGAs. In this chapter, we present an implementation method of APUF on an Altera FPGA without using an extra measurement module. Experiments are performed using FPGA technology to determine how much misalignment can be tolerated to achieve a high reliability.

4.1 Introduction

To implement an APUF on FPGA, we need to create a delay chain consisting of a number of MUXs as shown in Figure 2.4. However, it is hard to implement a symmetrical structure on FPGAs since FPGA synthesis tools are designed to optimize implemented circuits and remove redundant circuits. To overcome this problem, LCELL are used which are not optimized by Altera synthesis tool, hence it can be used to design delay lines composed of delay cells. This approach fits well for RO-PUF and TERO-PUF but design of APUF requires MUXs.

PUF implementation on an FPGA is challenging due to the placement and routing (PAR) tools in the FPGA rigid fabric and interconnection structure. A symmetric routing in most cases is infeasible and asymmetry in routing can lead to bias and predictable responses which reduces the response entropy [1].

4.2 Altera EP2C5

An APUF was implemented on Altera Cyclone II (EP2C5T144C8N) in this work. This board allows an easy way to design and test a PUF. This board is widely used to design advanced applications. It has a large number of logic elements (LE) along with I/O pins. The board is composed of flash, 50MHz active on-board oscillator, Joint Test Active Group (JTAG) and Universal Serial Bus (USB) blaster port, and I/O pins including LEDs, push button [2].

4.3 Cyclone II LE Architecture

Figure 4.1 shows the LE structure of Cyclone II, which consists of four-input LUT, programmable register, carry chain, and register chain. This LUT is a function generator of four variables, used to design different functions. The LE operates in either the normal mode or in the arithmetic mode. The normal mode best suits for general logic applications and combinational functions. Whereas, the arithmetic mode is used for designing adders, counter, accumulator and comparators [3].


Figure 4.1 Cyclone II logic element structure [4]

4.4 Design Requirements and Implementation

To implement APUF on FPGA, we utilized the diagram/schematic file. To design a delay line, it is required to copy the same circuit; for this an elementary symbol of a delay cell is created as shown in Figure 4.2 (a). A symbol consisting of a pair of 2-1 MUX in parallel whose inputs are cross connected and the select lines are connected was created as presented in Figure 4.2 (b). The delay lines were designed using the delay cell symbol as shown in Figure. 2.4.



(a)



(b)

Figure 4.2 (a) Elementary symbol of a delay cell (b) Symbol compact pair of 2-1 MUX whose inputs are cross connected and selectors are connected together

In this work, a 64-bit challenge PUF was implemented to generate 16-bit responses. Figure 4.3 shows the Register Transfer Level (RTL) view for 1-bit response of the implemented design.



Figure 4.3 RTL view of 64-bit challenge with 1- bit response



Figure 4.4 Resource property editor window view of delay cell and DFF



Figure 4.5 Effect of routing at different sections

Figure 4.4 shows the LE view for the last delay cell and a Delay Flip-Flop (DFF) in the resource property editor window, where MUX_U and MUX_L are the last multiplexers of upper and lower delay lines, correspondingly. Here MUX function is implemented using Look Up Table (LUT) whose output is connected to a DFF to generate the response.

PAR of logic elements on an FPGA has to be constrained since it can affect the output result. In addition, it can affect the PUF efficiency at four different sections listed below and shown in Figure 4.5.

- 1. Routing before the first delay cell,
- 2. Routing inside the delay cells,
- 3. Routing between the delay cells, and
- 4. Routing after the last delay cell.

The result of the automatic PAR is represented in Figure 4.6. It has to be noted that the PUF building blocks in the chip planner are randomly placed in LUTs of a number of LEs. This automatic PAR causes imbalance in the delay lines and makes the circuit biased and predictable. However, it is hard to calculate the exact percentage of imbalance as it depends on the type of FPGA board, software version and circuit design.

To overcome the undesired effects of PAR on PUF performance, an extra measurement module is commonly utilized to fix the imbalance by adding extra delay cells in the top delay line. This measurement module increases the area and complexity of implementation. In this work, we presented a new and easy solution to the imbalance created by PAR. Instead of adding an extra module, we exploited the chip planner tool available in the FPGA platform to check the PAR at the initial stage to fix the imbalance.



Figure 4.6 Random placing of MUX and arbiter element on the chip planner

Chip planner is a tool provided by Altera which allows a quick view of the logic cells on the chip. It provides visual display of post-place-and-route design mapped to the device structure. This tool allows users to create, move and delete logic cells.

This tool is used to manually place the logic cells in the desired location to minimize the imbalance between the lines. LogicLock Region (LLR) and Design Partition (DP) from



Figure 4.7 Design after partition in LLR



Figure 4.8 Result of one manual placement

Chip Planner tool are consumed to define specific regions to delay cells on the chip. The design is initially divided into individual partitions at DP level based on the system complexity and user need; then logic cells are locked in the desired LEs using LLR.

To minimize the misalignment between the upper and lower delay line, we placed the logic cells next to each other manually to test the uniqueness and reliability parameters of APUF. Figure 4.7 represents the physical allocation of 16-bits response APUF which is placed at different locations in the chip planner to check the propagation delay difference between PUFs with respect to the location, where Figure 4.8 presents the closer look of placed 1-bit PUF instance.

4.5 Experimental Results

In this section, the results are observed by burning the code on Cyclone II (EP2C5T144C8N) board. Figure 4.9 shows the utilized board.



Figure 4.9 Cyclone II FPGA board used for testing

The presented solution resulted in a PUF with delay imbalance of less than 8ps on average. The plot in Figure 4.10 shows the imbalance between 16-bit responses, where the blue and red colors indicate the results of imbalances between the upper and the lower delay lines before the arbiter for automatic placement and after placing the delay cells manually, correspondingly. It is clear from the plot that automatic PAR can result in minimum and maximum misalignment which cannot be ignored. The results of the proposed solution are between 5ps and 17ps for misalignment. Hence, this misalignment can also affect the reliability which has been examined further.



Figure 4.10 Result of propagation delay for random and fixed 16-bit PUF



Figure 4.11 Result of observing '1' as response after locking them at fixed places for five PUFs

To test the reliability parameter, the same challenge was applied 100 times and the response bits equal to '1' were counted. Figure 4.11 shows the result of observing '1' for 5 PUF responses. It can be seen that at most the response is '1' but for 2 PUF instances, at the end, response switched from '1' to '0'.

We also applied 100 random challenges and repeated each for 10 times to test the reliability and randomness of the proposed design. When the response bit is '1' for more than half of the challenges, the response was considered as '1'. Figure 4.12 shows the probability of obtaining '1' for 5 responses. It can be seen that the distribution is unbiased towards '1'. The probability of these responses for random challenges is around 50%, which is close to an ideal case.



Figure 4.12 Probability of observing '1' as response bit for random challenges versus 5 response bits

To ensure the uniqueness of the responses, the hamming distance is calculated and shown in Figure 4.13. Table 4.1 compares the results of this work with the previous work. The proposed solution has less misalignment as compared to the reported solution and can be implemented without the requirement of an extra hardware module. Moreover, it is easy to implement and do not require tuning block.



Figure 4.13 Hamming distance of output response bits

ruole il companson with pervious work	Table	1:	Compariso	n with	pervious	work
---------------------------------------	-------	----	-----------	--------	----------	------

	[21]	Proposed Method
Misalignment	Average 9 ps	Average 8 ps
Extra hardware module	Yes	No
Need tuning	Yes	No
Reliability	Close to 50 %	Average 50 %
Implementation	Complicated	Easy

4.6 Conclusion

A proper placement-and-routing of logic cells on FPGA platform is very important for PUF implementation, as this can cause a delay imbalance and in worst condition can affect the functionality of the design. To measure the imbalance of the implemented circuit on FPGA, an extra module is commonly used, however this extra hardware module can also add a delay imbalance to the circuit. In this work, we presented a method to measure and fix the imbalance without adding an extra module. Instead of automatic PAR, we manually placed the logic cells on the chip to ensure symmetrical PUF delay line and PUF reliability. The misalignment was successfully controlled using chip planner tool. This work was conducted for APUF, however other types of PUF can also be implemented using the same methodology. Experimental results indicate that the implementation of the circuit at different parts of the chip changes the response set.

4.7 References

[1] M. Majzoobi, F. Koushanfar, and S. Devdas, "FPGA PUF using programmable delay lines," in IWIFS, 2010, pp. 1-6.

 [2] "Introduction", Altera.com. (2008). [online] Available at: https://www.altera.com/en_US/pdfs/literature/hb/cyc2/cyc2_cii51001.pdf [Accessed: 05-May-2018].

[3] Cyclone II Device Handbook, Volume 1. San Jose, CA: Altera Corporation,2008, pp. 1-470.

 [4] "Cyclone II Architecture", Altera.com, 2007. [Online]. Available: https://www.altera.com/en_US/pdfs/literature/hb/cyc2/cyc2_cii51002.pdf. [Accessed: 05-May- 2018].

Chapter -5

Conclusions and Future Work

5.1 Summary

Hardware security and trust has become a challenge, due to increasing vulnerable to adversary attacks. In this thesis, we looked at the design of an Arbiter Physical Unclonable Function that enable reliable authentication of integrated circuits. Further, we studied the properties which help to derive the PUF design such as uniqueness, reliability, randomness and tamper evidence. In addition, we studied the designs of PUF circuits and their applications. PUF functionality depends on the performance metrics of the CRPs and these parameters are influenced by environmental conditions. To test the accuracy of APUF, reliability and uniqueness simulations were performed under fluctuating power supply and temperature using Cadence environment 0.18µm technology. We also analyzed how these factors reduce the database of CRPs.

Another way to test the accuracy of the PUF is by implementing it on FPGA environment but there are several challenges. It is important to make sure that the implemented PUF is balanced and has same symmetry which requires to measure the propagation delay of the delay lines. To measure the propagation delay an extra hardware module is proposed by the literature. However, this measurement module can also affect the accuracy of the PUF. To test the accuracy of APUF on FPGA platform, we utilized Chip Planner software available inside the FPGA tool instead of an extra hardware module.

5.2 Conclusion

The results conducted using Cadence tool indicated that more than 23% of the CRPs reduced from the database due to $\pm 10\%$ of power supply variations. In addition, around 26% of the reduction in CRPs was noticed in the case of temperature variation from -40°C to +70°C. Such a reduction was huge and cannot be ignored. Therefore, it is imperative that if supply or temperature variations may occur, circuit elements specifically designed to be robust to such variations be employed in the design of PUFs. The result of this work has been accepted in MWSCAS-2018 for publication.

In addition, we implemented APUF on Altera FPGA platform by using Chip Planner tool available in FPGA and achieved to 8ps of misalignment with 5ps from the best case and 17ps for the worst case. Hence, this approach of implementing PUF designs on FPGAs provide better performance and is much easier to implement. The results have been prepared to be submitted in journal for publication in the field of security.

5.3 Future Work

In this thesis, measurement results for the FPGA implementation were conducted under the same environmental conditions. As presented in chapter 3, varying operating conditions can affect the performance of PUF. Measurement results under varying operating conditions are required. Moreover, performance evaluation for a PUF designed using delay cells containing temperature and voltage resilient circuits can be a good topic for future works.

APPENDIX: IEEE PERMISSION TO REPRINT

In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of University of Windsor products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to https://www.ieee.org/publications/rights/reqperm.html to learn how to obtain a License from RightsLink.

VITA AUCTORIS

NAME: Manpreet Kaur

PLACE OF BIRTH: Patiala, India

YEAR OF BIRTH: 1992

Education:

Master of Applied Science (Electrical Engineering)	June 2018
University of Windsor	Windsor, ON
Bachelor of Engineering (Electrical and Electronics)	August 2014
Asia Pacific Institute of Information Technology	Panipat, India