2-13-2014

A Physical Unclonable Function Based on Inter-Metal Layer Resistance Variations and an Evaluation of its Temperature and Voltage Stability

Jing Ju

Jing Ju

Candidate

Electrical & Computer Engineering

Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

Dr. James F. Plusquellic , Chairperson

Dr. Payman Zarkesh-Ha

Dr. Ryan Helinski

Dr. Fernando Perez-Gonzalez

A Physical Unclonable Function Based on Inter-Metal Layer Resistance Variations and an Evaluation of its Temperature and Voltage Stability

BY

Jing Ju

B.E., Communication Engineering, Hunan University, 2006
M.E., Information &Communication Engineering, Hunan University, 2009

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Engineering

The University of New Mexico
Albuquerque, New Mexico

December, 2013

Dedication

This work is dedicated to my parents and husband who have made this possible through their love and support.

Acknowledgments

I would like to thank all who helped me directly or indirectly in making this work possible. First of all, I thank my husband Huiwen Xu who encourage me to pursue my Phd degree in the United States. I owe gratitude to all my family members for their support and encouragement, without which I would't have been able to come this far.

I heartily acknowledge Dr. Jim Plusquellic, my advisor and dissertation chair for his guidance, support, and contributions throughout this work. He is the main source of inspiration for me throughout this research work. I am so grateful for all the opportunities that he offered me. I also thank my committee members, Dr. Payman Zarkesh-Ha, Dr. Ryan Helinski and Dr. Fernando Perez-Gonzalez, for their valuable advice to this research work.

I also thank my colleague Charles Lamech, Jim Aarestad and Fareena Saqib, for all their help and support through the daily discussion.

To everyone at the Electrical & Computer Engineering Department at the University of New Mexico who made this possible, thank you.

A Physical Unclonable Function Based on Inter-Metal Layer Resistance Variations and an Evaluation of its Temperature and Voltage Stability

by

Jing Ju

B.E., Communication Engineering, Hunan University, 2006
M.E., Information & Communication Engineering, Hunan University, 2009
Ph.D., Engineering, University of New Mexico, 2013

ABSTRACT

Keying material for encryption is stored as digital bistrings in non-volatile memory (NVM) on FPGAs and ASICs in current technologies. However, secrets stored this way are not secure against a determined adversary, who can use probing attacks to steal the secret. Physical Unclonable functions (PUFs) have emerged as an alternative. PUFs leverage random manufacturing variations as the source of entropy for generating random bitstrings, and incorporate an on-chip infrastructure for measuring and digitizing the corresponding variations in key electrical parameters, such as delay or voltage. PUFs are

designed to reproduce a bitstring on demand and therefore eliminate the need for on-chip storage.

In this dissertation, I propose a kind of PUF that measures resistance variations in inter-metal layers that define the power grid of the chip and evaluate its temperature and voltage stability. First, I introduce two implementations of a power grid-based PUF (PG-PUF). Then, I analyze the quality of bit strings generated without considering environmental variations from the PG-PUFs that leverage resistance variations in: 1) the power grid metal wires in 60 copies of a 90 nm chip and 2) in the power grid metal wires of 58 copies of a 65 nm chip. Next, I carry out a series of experiments in a set of 63 chips in IBM's 90 nm technology at 9 TV corners, i.e., over all combination of 3 temperatures: -40^{O}C, 25^{O}C and 85^{O}C and 3 voltages: nominal and +/-10% of the nominal supply voltage. The randomness, uniqueness and stability characteristics of bitstrings generated from PG-PUFs are evaluated. The stability of the PG-PUF and an on-chip voltage-to-digital (VDC) are also evaluated at 9 temperature-voltage corners.

I introduce several techniques that have not been previously described, including a mechanism to eliminate voltage trends or 'bias' in the power grid voltage measurements, as well as a voltage threshold, Triple-Module-Redundancy (TMR) and majority voting scheme to identify and exclude unstable bits.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

As electronic devices are becoming ubiquitous and interconnected around people's daily life, security, trustworthy computing, and privacy protection have emerged over the past decades as hardware design objectives of great significance. For many emerging applications that need exceptional security in identifying and authenticating users, such as intellectual property (IP) protection, system security is traditionally based on the protection of secret keys. Conventionally, these secret keys are based on battery-backed RAM or NVM such as ROMs, fuses, or flash/EEPROM. These secret keys need to be well managed and stored to avoid releasing information to attackers. Over the past several years, many kinds of attacks for exacting, estimating, or cloning secret keys that are stored digitally in NVM have been successfully developed and reported. When the adversaries have the full and direct access to ICs, the situation is getting especially problematic and worse. Also, integrating a NVM for an IC incurs additional costs and fabrication overhead.

The described problem has become more intense recently, and this motivated the

idea of using inherent random manufacturing variations to serve as identifiers for ICs. It provides a promising alternative which can address the present challenges of traditional security that were described earlier. This idea is not new and a kind of circuits called Physical Unclonable Functions, which store secret keys in silicon circuits by exploiting uncontrollable randomness due to manufacturing process variations, have been proposed and successfully implemented to overcome some of the problems faced by traditional techniques. PUFs are easy to build and can be implemented without any additional manufacturing steps since they are based on the existing resources of physical variations of the IC. Due to the inherent properties and advantages of PUFs, the research of them has becoming more and more attractive in recent years.

1.2 Introduction to PUFs

When ICs are manufactured, there will be inevitable inherent process difference even these ICs are fabricated from the same lot or wafer. Precise control over the fabrication of IC components is becoming more difficult in advanced technology generations, resulting in a wide range of electrical variations among and within the replicated copies of the chip. Due to the existing manufacturing process variations, it is difficult or impossible to create two exact ICs. A PUF is such an embedded structure that are designed to be sensitive to variations in the printed and implanted features of wires and transistors on the IC. Signal variations that occur within the IC are the source of entropy for the PUF.

Usually, the physical variations of manufacturing process manifest as analog variations in the chips' parametric properties, such as wire delay, threshold voltage.

Therefore, a PUF is designed to measure and 'digitize' these analog electrical variations. When characterizing the PUF in a more understandable way, it can be considered as a function that maps challenges to responses. The basic operation is shown in Fig. 1.1: when a **Challenge** (C) is applied to the PUF, it reacts with a **Response** (R). And the Challenge together with its associated response is known as Challenge-Response Pair (CRP).

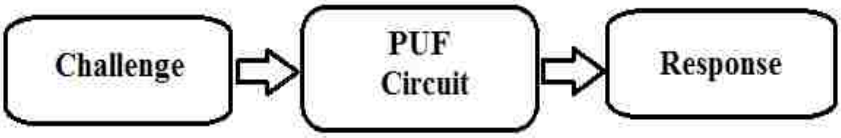


Fig. 1.1: CRP Protocol.

By applying a set of 'challenges' to the PUF, it can produce a large, ideally exponential, set of unique bitstring, which can be considered as the fingerprint or DNA of a chip. Since the challenges are typically 'digital' and therefore they can be generated on-chip from a pseudo-random number generator such as a linear feedback shift register (LFSR).

Nothing in manufacturing process of a chip is exact, and therefore, all fabricated physical components, e.g., wires and transistors, on the chip vary from their nominal characteristics. Although it is possible to measure these physical variations directly, it is extremely difficult or impossible to do so without sophisticated process and equipment. The analog electrical and parametric variations that result, on the other hand, can be measured and processed more easily, and in many cases, this can be done using on-chip

instrumentation. Many proposed PUF-based systems are defined in this manner, and differentiated by the type of electrical and/or parametric variations they leverage. The magnitude and stability of variations in, e.g., transient current delay, leakage, resistance, capacitance, etc. are dependent on the technology and environment, and therefore, some PUF systems can better meet the important criteria described above than others.

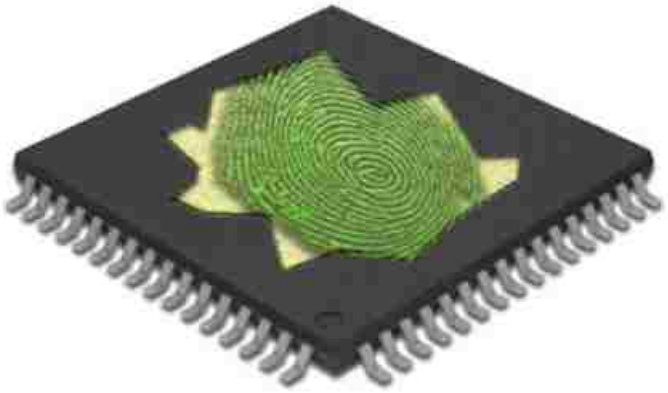


Fig. 1.2: The fingerprint of devices.

Due to the inherent and particular characteristics, PUFs are promising components in the next generation of IC security and the bitstring generated from PUFs can be used in security applications such as encryption, authentication, feature activation, metering, etc. Since the bitstrings are generated on-the-fly using dedicated hardware primitives and processing engines, and thereby avoid the need for storage in on-chip NVMs. This feature not only improves their resilience to invasive attacks designed to steal the secret key material, but it also reduces the cost of manufacturing the IC. The latter is true because, in many cases, PUFs are designed using components that can be fabricated using standard

complementary metal-oxide-semiconductor (CMOS) processing steps, and therefore, the cost of integrating non-standard components, such as NVM, is eliminated. Another important characteristic of the PUF as a next generation security mechanism is its potential for generating large numbers of repeatable random bits. This feature offers new opportunities for software processes to strengthen security mechanisms, for example, by allowing frequent rekeying in encrypted communication channels and by allowing a large, changing set of shared keys to be utilized among multiple communicating entities.

From the above description of PUFs, we know PUFs can provide security advantages over traditional key generation mechanisms because they are volatile and very difficult to clone. Unfortunately, the analog nature of the underlying random variable make the PUF sensitive to environmental variations such as temperature and power supply noise. For some important applications such as keys for encryption, they require that the PUF must produce the same bit string for a fixed challenge under varying environment. Therefore, in order to effectively move from theoretical exercises to commercial products, the 'quality' of the bitstring produced can be measured against many statistical metrics, but needs to meet three important criteria: 1) the bit string is unique for each chip, and thereby able to distinguish each chip in the population, 2) the bit string is random for each chip, and therefore is difficult or impossible to model and predict by an adversary, and 3) the bit string is stable, i.e., it remains constant for a given chip over time, and under varying environmental conditions. A PUF that is able to meet these requirements can be used in applications related to security including system identification, key generation for encryption, authentication via CRPs, IP protection,

remote service etc.

In this dissertation, Inter-chip hamming distance (HD) is used to determine the uniqueness of the bitstrings among the population of chips. Similarly, the National Institute of Standards and Technology (NIST) statistical test suite is used to evaluate the randomness of the bitstrings produced by each chip. And intra-chip HD can be used to evaluate stability of the bitstrings, i.e., the ability of each chip to reproduce the same bitstring time-after-time, under varying temperature and voltage conditions.

1.3 Contributions

In this dissertation, I investigate two implementations of a PUF that measures resistance variations in metal wires that define the power grid of the chip, and we call it PG-PUF. There are several benefits of the PG-PUF. First, resistance of metal wires varies linearly with temperature and can easily be designed to be resistant to aging effects such as electromigration. Second, resistance can be measured using a simple direct current (DC) process, which can improve signal-to-noise significantly over PUFs that leverage alternating current (AC) characteristics, such s delay. Third, metal components are ubiquitous on a chip, with the power grid consuming a large fraction of the metal resources, e.g., 15-25% is typical in most commercial power grid designs. Fourth, the power grid is a stacked structure, offering a 3rd dimension in which to leverage entropy. Last, the interconnected structure of the wires in the power grid complicates the interaction among variations in resistance that occur, thereby increasing the complexity of model building attacks.

In order to fully leverage metal resistance variations, I design a PUF that can measure them in each of the metal layers of the power grid. In order to reduce bias issues and correlations that exist in the V_{DD} and GND grids, inter-metal layer voltage drops/rises, which are computed by subtracting the voltages measured from consecutive metal layers, are introduced. By this method, the independent resistance variations that occur in each of the metal layers of the power grid can be leveraged.

First, the experiments are carried out on a set of chips fabricated in IBM's 90nm, 9 metal layer bulk silicon process, and on a set of 65 nm chips fabricated in IBM's silicon-on-insulator (SOI) process at room temperature. The quality of the bit strings are evaluated using inter-chip and intra-chip HD, as well as a suite of statistical tests available from NIST [1].

Then, I focus on determining the temperature and voltage (TV) stability of the PG-PUF. A significant benefit of using metal structure is that “noise-related” variations, such as those introduced by TV variations, result in linear changes to the measured voltages. This linear scaling characteristic allows the relative magnitude of two voltages to remain consistent across changes in temperature and voltage, which, in turn, improves the stability of the PUF to bit-flips, when compared for example to PUFs which leverage transistor-based variations. In the experiments, I evaluate the PG-PUF at 9 TV corners, i. e., over all combinations of 3 temperatures; -40^{o}C, 25^{o}C, 85^{o}C, and 3 voltages; nominal and +/-10% of nominal. The evaluation is carried out on a set of chips fabricated in IBM's 90 nm, 9 metal layer bulk silicon bulk silicon process. The stability of the bitstring is measured using intra-chip HD and 'probability of failure' techniques. Randomness and

uniqueness are also evaluated using NIST test suite and inter-chip HD methods.

I also investigate an on-chip voltage-to-digital converter (VDC) for measuring voltage variations (which reflect resistance variations in the metal wires) and its stability across the 9 TV corners. The VDC is used to digitize the voltage drops produced from each metal layer of the power grid.

Besides, I introduce several noise resilient bit-flip avoidance schemes, which are used to reduce the probability of a failure to reproduce the bitstring to less than 1E-9 under varying environmental variations. One technique is called thresholding mechanism, which derives a voltage threshold from a chip's voltage drop distribution profile that is used to decide whether a given voltage comparison generate a strong bit or a weak bit. The second scheme is a bitstring replication method that mimics a popular scheme used in fault tolerance called Triple-Module-Redundancy (TMR). TMR is proposed for fixed-length bitstrings that can further improve the stability of the bitstrings. Although these techniques discard a large fraction of bits, they provide several advantages and can be used as an alternative to error correction and Helper Data schemes [2][3]. The helper data (public data) associated with these methods reveals nothing about the secret bitstring since every bit is independent from others. Also, the helper data is simple data for both of these techniques and its size can be easily reduced by using a certain compression technique. What's more, these methods increase the difficulty of model-building since bitstrings are constructed using only a subset of the possible voltage pairings. However, for error correction scheme, the complex algorithm implementation will require significant resources and overhead. And also the error correction scheme will release

some secret information. Therefore, both the thresholding and TMR are easier and cheaper to implement on chip when compared with the error correction schemes. That's the main reason we avoid using the error correction scheme to fix unstable bits issues for PUFs. In this dissertation, both techniques are demonstrated to provide a significant improvement to inter-chip HD and the results obtained from NIST statistical tests.

1.4 Organization of the Dissertation

This dissertation is organized as follows:

In Chapter 1, the motivation of using PUFs for security applications is briefly described. Then the basic idea of PUFs and their advantages over conventional methods are given. Finally, the requirements for PUFs in order to be used for practical security applications and some methods of evaluating the quality of them are discussed.

In Chapter 2, the development of definition for PUFs are described. Then the previous work about PUFs and classification of the popular PUFs are covered. Also, the applications of PUFs are discussed. Besides, several statistical methods for evaluating the quality of bitstrings generated from PG-PUFs are briefly described. Finally, some techniques used to improve the stability of PUFs are given.

In Chapter 3, two implementations of PG-PUFs for both IBM 90 nm and 60 nm technologies are introduced and experiments are carried out without considering environmental variations. The output bitstrings are analyzed and evaluated by using inter-chip HD, intra-chip HD and NIST tests suite.

In Chapter 4, experiments are carried out across different temperatures and voltages in order to evaluate the stability of PG-PUFs against environmental variations. Several methods of improving the stability of PG-PUFs are demonstrated. The quality of bitstrings are also evaluated using statistical tests.

In Chapter 5, some future work is proposed.

Finally, the findings and contributions of this dissertation are concluded.

Chapter 2

Background

Study on manufacturing process variations and PUFs has been gaining increasing interest and attention since the concept of PUFs was firstly introduced in [4]. Over the past couple of years, the attention on PUFs has risen substantially, making them a hot topic in the field of hardware security and leading to an expansion of published results. This chapter will provide an overview of PUFs. The basic definition will be given in Section 2.1, different types of PUFs will be classified in Section 2.3, many important applications will be discussed in Section 2.4. At the end of this chapter, several methods used for evaluating the quality of PUFs will be listed. And many techniques used to improve the stability of PUFs are also discussed.

2.1 Definition of PUFs

It is widely accepted that the significance and complexity of process variations is increasing as technologies are aggressively scaled [5][6]. The idea of using these manufacturing process variations as identifiers for ICs has showed up for a long time. However, the concept of PUFs didn't appear until recent years. There have already been several proposals for the definition of required PUF properties. Firstly, they were called

Physical One-Way Function [4], and later renamed to Physical Random Functions [7], but eventually resolved to Physical Unclonable Functions (PUFs). The exact definition for this kind of circuits is still under discussion. The most popular definition of PUFs is described in [8] by Gassend et al.:

"A physical Random Function (PUF) is a function that maps challenges to responses that is embodied by a physical device, and that verifies the following properties:

- Easy to evaluate: The physical device is easily capable of evaluating the function in a short amount of time.
- Hard to predict: From a polynomial number of plausible physical measurements, an attacker who no longer has the device, and who can only use a polynomial amount of resources can only extract a negligible amount of information about the responses to a random chosen challenge."

There are also other descriptions of PUFs, e.g., in [9], the authors described that "Physical Unclonable Functions consist of inherently unclonable physical systems. They inherit their unclonability from the fact that they consist of many random components that are present in the manufacturing process and cannot be controlled. When a stimulus is applied to the system, it reacts with a response".

2.2 PUF Extensions

From the description in section 1.2, we know that PUF is a function that can map challenges to responses. Therefore, according to how many CRPs different PUFs can

generate with the same hardware resources, there are two different situations: strong PUFs and weak PUFs. Usually, Strong PUFs can produce an exponential number of CRPs which are more difficult for attackers to model and learn, while weak PUFs only have a linear number of them. Besides trying to generate as many CRPs as possible, people have also developed some other methods to increase the security of PUFs. Some people use pre-processing or post-processing schemes to obtain more robust bitstrings. Some people combine PUFs with a particular algorithm which can make model-building attacks more difficult. In the following part, I will discuss more details on how to obtain these improvements.

2.2.1 Controlled PUFs

The concept of Controlled PUFs or (CPUFs) was introduced by Gassend et al. in [2], which combine a PUF with other primitives. The CPUFs can only be accessed via an algorithm that is physically bound to the PUF in an inseparable way, and any attempt to break the link between the PUF and the algorithm will definitely lead to the destruction of the PUF. Therefore, by using controls to the PUFs, it is possible to make a PUF more robust and reliable. In [7], the authors summarized several controls that can be placed around the PUF to improve its function. Fig. 2.1 shows the full details of these controls.

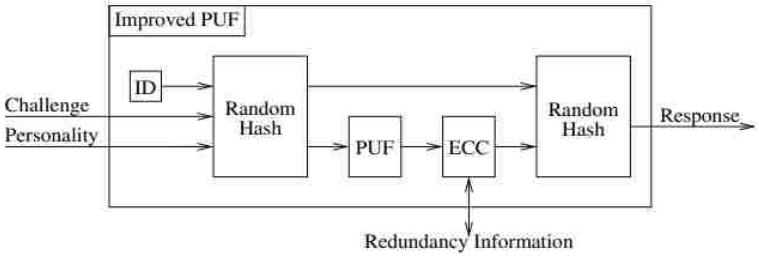


Fig. 2.1: Using controls to improve a PUF[7].

2.2.2 Reconfigurable PUFs

In 2009, the authors in [10] defined a new primitive, called reconfigurable PUF (rPUF). The rPUF is a PUF with a mechanism, which can transform the original PUF into a new PUF with a new unpredictable and uncontrollable challenge-response behavior. Meanwhile, the rPUF can still preserve all the security properties of the original PUF. In this literature, two practical instantiations of rPUFs were presented. One is based on the optical PUF. When the structure of the optical PUF is irradiated with a laser beam outside the normal operating conditions, the structure will change its internal configuration. This change will cause a rearrangement of the optical scatters, which will lead to a completely new random CRP behavior. Another is based on phase change memory. By rewriting the memory cells, a new random CRPs can also be generated .

Later in 2011, more novel structures of the reconfigurable PUFs were discussed in [11]. The first method is adding reconfigurable elements before the challenges applied to the PUF, while the original properties and performance of the PUF will still be preserved; another idea is to add extra reconfigurable components to process the output before using it as an authentication key. The authors in the literature also stated that instead of only making CPRs reconfigurable by processing the challenge and responses directly, they can also alter the main PUF circuit to update the challenge-response behavior.

2.3 PUF Implementations

Since the introduction of concept of PUFs, many studies on PUFs have been reported.

There have been a large variety of PUF implementations, and the list still keeps getting larger. In this section, I will mainly discuss the following several types of PUFs: 1) Non-electronic PUF. (2) Delay-based PUF. (3) Memory-based PUF. (4) Power-grid PUF. 5) Misc Category.

2.3.1 Non-Electronic PUFs

An earlier non-electronic PUF is a so-called optical PUF [4], which use microscopic refractive glass spheres in a small transparent epoxy plate. It is stimulated using a helium-neon laser and the speckle pattern represents the response. The basic implementation and operation of an optical PUF is shown in Fig. 2.2 [12]. It is obvious that the optical PUF has a really complex structure and therefore cannot be used for large scale production.

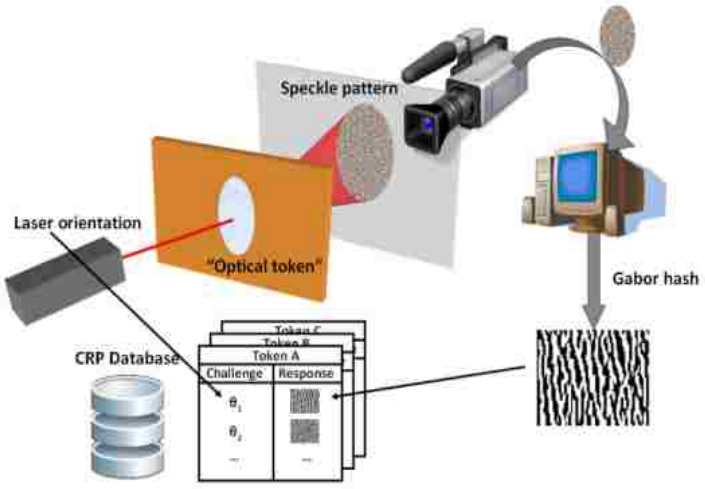


Fig. 2.2: Basic operation of an optical PUF [12].

In [13], the authors introduced a technique which can make use of the reflection of a focused laser beam by the irregular fiber structure of a paper document. The particular

pattern of the reflection can be used as fingerprint of that paper and therefore can be used as anti-counterfeiting strategy for currency notes. In [14], the authors proposed a new technique for exacting unique fingerprints from identical CDs. It was observed that the manufacturing variations of the lengths of lands and pits on a regular CD are large enough to be measured from the electrical signal of the photodetector in a regular CD player. In [15], a kind of PUFs called magnetic PUFs were introduced, which use the inherent uniqueness of the particular patterns in magnetic media, e.g., for credit cards. Other non-electronic PUFs including RF-DNA [16] and acoustical PUFs [17] were also proposed.

2.3.2 Delay-Based PUFs

Delay-based PUFs utilize the variations in propagation delay of identical circuits to derive a secret response. There are mainly two popular types of delay-based PUFs: arbiter PUFs and ring oscillator (RO) PUFs.

The basic idea of the arbiter PUF was proposed in [18][19], which exploits the statistical delay variations of wires and transistors across ICs to introduce a digital race condition on two paths on a chip and to be implemented with an arbiter to decide which path won the race. The basic structure of the arbiter PUFs is shown in Fig. 2.3. In this structure, a sequence of switches is connected in series and each of them is controlled by one bit of the challenge. At the end, an arbiter decides which path is faster and assigns a digital '1' or '0' accordingly. In real world, the two delay paths are 'lay-out' in an identical fashion. However, due to the uncontrollable manufacturing processing variations, the

delays of these two delay paths vary randomly across different chips, which can allow the generation of unique responses for every IC.

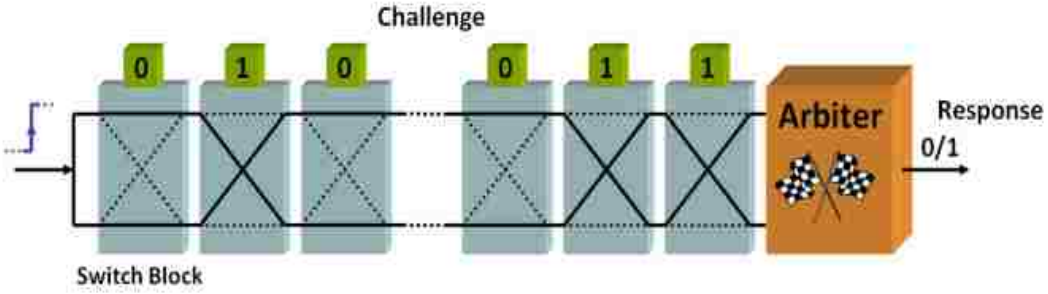


Fig. 2.3: Basic operation of an arbiter PUF [12]

From the structure of the arbiter PUF, it is known that the total delay of the chain should be the sum of the delay of the separate blocks. Therefore, the arbiter PUF is susceptible to model-building attacks. By observing a number of CPRs, the attackers can build a mathematical model of the PUF that predicts the response to an unseen challenge. In order to make the model-building attacks to the arbiter PUF more difficult, in [19], a feed-forward arbiter PUF was depicted as shown in Fig. 2.4. In this structure, multiple challenge bits are determined by the racing result in intermediate stages instead of being provided by a user as that for the basic arbiter PUFs [8][18]. Since the internal feed-forward bits are hidden to an adversary, it is difficult to build a precise model of a feed-forward arbiter PUF.

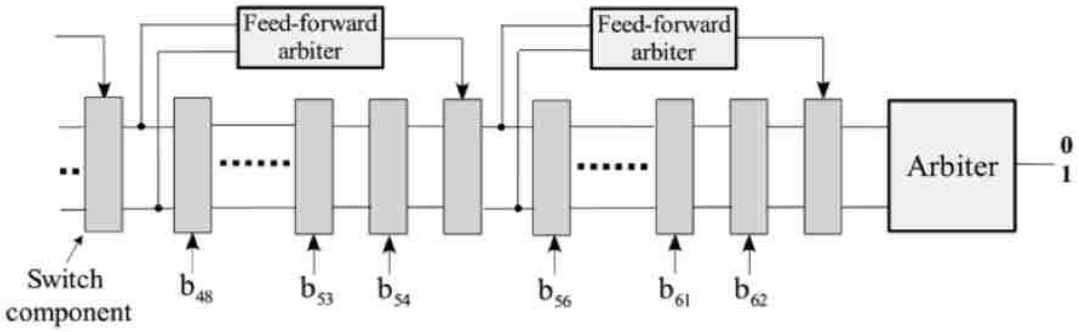


Fig.2.4: Feed-forward arbiter scheme for improved security [19].

However, experimental results from literatures [20] show that with more advanced modeling techniques it is still possible to build an accurate model for the feed-forward arbiter PUF. More advanced architectures were proposed in [11][21]. In [21], the authors developed a new set of techniques for FPGA-based PUF design and implementation, which use reconfigurable structure to eliminate some of the classical arbiter-based PUF limitations and make them more resistant to modeling.

Also, note that arbiters in practice are implemented by D flip-flops. When the absolute delay difference of the arriving signals is smaller than its setup and hold time, which results in metastability and high sensitivity to environmental variations. In order to solve these problems, Devadas et al. in [22] introduced a new arbiter-based PUF structure that exploits programmable delay lines (PDL) to tune and cancel out the delay skews caused by asymmetries in routing on FPGAs. The authors in this literature also stated that there is still other direction of improving the security of arbiter-based PUFs, which is by reducing noise probability in PUF responses. Further, a PDL-based symmetric switch

structure was also introduced to resolve the routing issues.

Another type of delay-based PUFs is called Ring-Oscillator PUF (RO-PUF). The basic structure of a RO-PUF is proposed in [7] as shown in Fig. 2.5. An AND gate in the loop allows to enable/disable the oscillation. As introduced in [7], the RO-PUF uses a different approach towards measuring small random delay deviations caused by manufacturing variability. The source of this randomness is again the uncontrollable effect of process variations on the delay of digital components. The basic idea of RO-PUF is to connect the output of the delay chain (that for the Arbiter PUFs) to the input and measure the frequency (instead of delay).

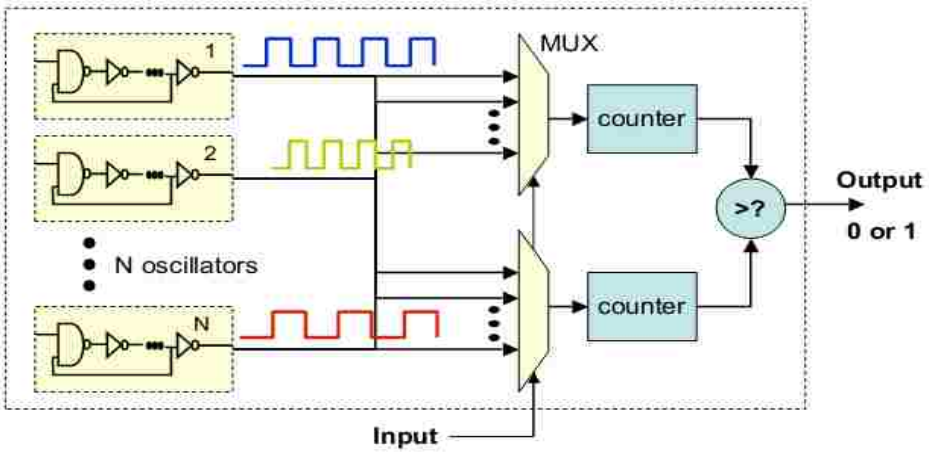


Fig.2.5: Basic Ring-Oscillator PUF circuit proposes in [7].

However, the RO-PUF suffers from the systematic or correlated process variation and the environmental noise caused by the voltage and temperature variations, which degrade the uniqueness and the stability of PUF responses [12]. There have been many effective solutions trying to improve the performance of RO-PUFs [23][24][25]. In [23],

the authors presented a temperature-ware cooperative RO-PUF implementation to reduce the hardware cost. In [24], Maiti et al. proposed a compensation method to mitigate the negative influence of systematic process variations. The method is as follows: firstly place the group of ring oscillators as close as possible to each other, and secondly pick the physically adjacent pair of ROs while evaluating a responses bit. Also, they introduced a novel configurable ring oscillator technique, which can drastically reduce the effect of noise on PUF responses. In [26], Mansouri et al. introduced a new type of RO-PUF. In this structure, the inverters composing the ring oscillators can be supplied by independent voltages. The authors stated that this basic concept can be used to design temperature-aware RO-PUFs, whose output bits do not depend on the operating temperature. Therefore, the reliability of the PUF is improved to temperature variations.

Also, an improved RO-PUF called reconfigurable PUF with a mechanism to update its challenge-Response pairs was proposed in [10]. This PUF preserves the properties of original PUF but has unpredictably different challenge-Response behaviors after every reconfiguration.

For all the delay-based PUFs, they are based on the delay variations and propagation time comparison or rings oscillator frequency comparisons. There exist contest that these delays or frequencies are too similar to decide the response. Based on this contest, the Grenoble INP team in [27] developed a PUF based on asynchronous communication in order to generate violation depending of flip-flop set-up time. When a synchronous design does not respect the setup time, this can induce metastability in the circuit (a “quasi-stable” state which will be resolved either by '0' state or by '1' state in

an undefined time).

Besides these two popular types of delay-based PUFs, in [28], a new delay-based PUF architecture called glitch PUF was proposed, which exploits glitches that behave non-linearly from delay variation between gates and the characteristic of pulse propagation of each gate. The authors presented a method to accurately acquire the waveforms and to convert them into response bits and also evaluate the characteristics of glitch PUF. This structure is expected to solve the current problem of delay-based PUF that it is easy to predict the relation between delay information and generated information.

2.3.3 Memory-Based PUFs

In this section, another popular type of PUFs that leverage the bi-stability characteristics of memory primitives will be listed. It is implemented by putting a memory cell into an metastable state and then releasing it. In most cases, digital memory cells heavily prefer one of the two stable states. However, in some cases, meta-stability results, which results in oscillations between stable states and non-deterministic settling times.There have shown several kinds of memory-based PUFs such as Static Random-Access Memory (SRAM) PUFs [29][30][31], Butterfly PUFs [32], Flip-flop PUFs [33], Latch PUFs [34].

In [29], the authors provided the first construction of an intrinsic PUF based on SRAM memory randomness. The typical SRAM PUF is implemented with six transistors (MOSFETs) as shown in Fig. 2.6, and formed of two cross-coupled inverters and two access transistors connecting to the data bit-lines based on the word-line signal. In the

same year, a very similar concept was also presented in [30].

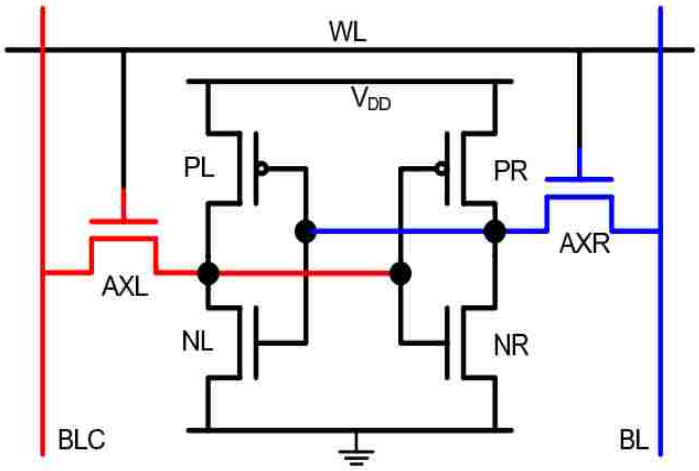


Fig.2.6: Six transistor SRAM cell [29].

Later in the year of 2008, the authors of [32] proposed a new PUF structure called Butterfly PUF that can be used on all types of FPGAs. The structure of the butterfly PUF is as shown in Fig.2.7, constructed as symmetrically as possible by manual routing of the signal wires. It consists of two latches, each with a preset signal and a clear signal. The data signal is transferred to the output when the CLK is high. The stable state of the butterfly PUF depends on the slight differences in the delays of the connecting wires which are designed using symmetrical paths on the FPGA matrix.

The authors in [35] introduced an SRAM PUF which they implemented on a microcontroller using the internal SRAM block. In [33], the authors proposed a so-called flip-flops PUF, which is based on the power-up behavior of regular flip-flops. The experimental results in this literature show that this flip-flop PUFs is equivalent to the previous proposed SRAM PUFs. The authors stated that the flip-flops PUFs have some

advantages with regard to the use in reconfigurable devices.

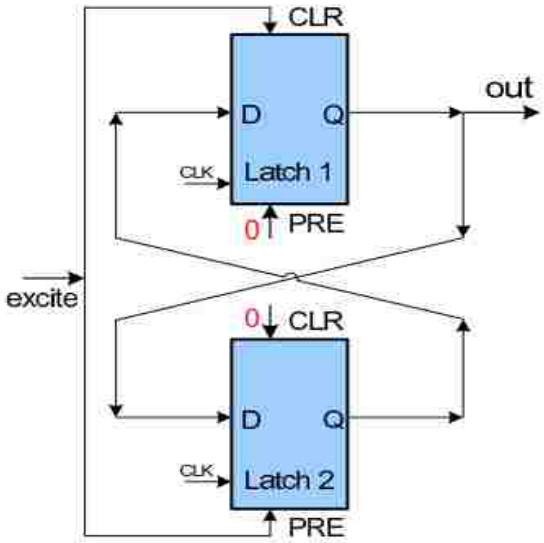


Fig.2.7: Butterfly PUF: cross-coupled latches[32].

In [36], the authors presented detailed evaluations of 90 nm 6T-SRAM as PUF for secure key generation in wireless sensor nodes. In order to put SRAM PUF in low-power security applications, a design for using a 65 nm 10T sub-threshold SRAM as a PUF was proposed in [37]. In this design, significant improvements are achieved in power consumption and security over existing designs. Experimental results show that this design is suitable for ultra-low power security applications since it have more than 50% of uniqueness and near 100% reliability with a 60% of power saving.

These memory-based PUFs are reasonably resistant to model-attacks compared to the delay-Based PUFs as their responses are based on independent random elements. But the memory-based PUFs grow with overheads (size) exponentially related to the length of their challenges.

2.3.4 Power Grid-Based PUFs

The power grid-based PUF (PG-PUF) was firstly proposed in [38], which is based on the resistance variations in the power grid of a chip. There are two strategies for the PG-PUFs in this paper: one is based on voltage drops, and another is based on equivalent resistance. Voltage drops and equivalent resistances in the power distribution system are measured using external instruments and it is again observed that these electrical parameters are affected by random manufacturing variability.

The authors in [38] also stated that there are many advantages using process variations of power grid as PUFs. Firstly, the PG-PUFs is based on the variations in only passively components of the IC, specifically the power grid, and is therefore less susceptible to environmental variations. Secondly, because the power grid is an existing distributed resource in every design, the overhead of the PG-PUFs is lower than other kind of PUFs.

2.3.5 Misc Category

In the year of 2011, many novel PUFs showed up [39][40][41][42][43]. In [39], the authors presented PUF circuits designed to exploit inherent fluctuations in physical layout, called litho-PUFs. Variations arising from proximity effects, density effects, etch effects, and non-rectangularity of transistors are leveraged to implement lithography-based PUFs. And the authors showed that the uniqueness level of these PUFs are adjustable and are typically much higher than traditional ring-oscillator or tri-state buffer

based approaches. In [40], the authors proposed a PUF called PE-PUF. This PUF takes into account process variations, temperature, power supply noise and crosstalk, which are major sources of variations and noise in ICs. In [41], the authors introduced a new class of dynamic PUF (DPUF) that employs dynamic system alteration with respect to individual gate speeds, enabling each user to create his own PUF as needed. The authors stated that this DPUF has very high unpredictability and can easily be stabilized. The authors in [42] presented a PUF called Bi-stable Ring PUF. The basic idea of it is based on the fact that an inverter ring consisting of an even number of inverters has two possible stable states. The PUF proposed in [43] is built using leakage sensors with each measuring the leakage current of a transistor. When compared with a popular ring oscillator PUF architecture of the same entropy, the authors stated that this proposed PUF consumes about 80% less power, occupies about 85% less area, and has a high level of stability across a wide range of temperature. Very recently, the authors in [44] used the Buskeeper as a PUF, and gave experimental results showing that the Buskeeper PUF can be considered as a good alternative to D flip-flop PUFs since the Buskeeper PUF are much more efficient than D flip-flops in regard to the amount of hardware resources required while they can maintain the performance as well as the already and generally accepted ones. In [45], J. Aarestad et al. introduced a new PUF called the Hardware-Embedded Delay PUF, or HELP, which leverages the natural variations that occur in the path delays of a core micro on a chip to create an unique, stable, and random bistring of virtually any length.

2.4 Applications of PUF

PUFs have shown great promise in many security applications such as IC identification [7][46][47][48], generating unique keys for encryption [49][50], authentication [8][47] [51][52], IP protection on FPGAs [32], RFID for anti-counterfeiting [53].

The unclonability property makes PUFs attractive for supporting anti-counterfeiting technologies, in the biometric sense. Therefore, PUFs can be used to identify and authenticate a chip. The process works by the following steps: at time '0' (after manufacturing), a trusted entity applies a process called enrollment phase. In this phase, a subset of challenges are applied from the much larger number of available challenges and the challenge-Response pairs (CPRs) are stored in a secure database. This database need to be securely managed by the trusted party, only who has knowledge of which challenges are applied; later in the field, during the authentication process, the trusted party apply a challenge and then check whether the response is close enough to that in the database. After the use of the CPRs, it needs to be removed from the database in order to avoid information release to attackers.

Secret keys are essential and important to many security applications such as IP protection and software licensing as well as conventional ID cards and smart cards. The response bitstrings of PUFs are proposed to integrate secret keys for the use in cryptography. When compared to the conventional methods of storing secret keys in NVM, PUFs have some additional security against probing and side-channel attacks: 1) PUFs do not require a key-programming step, as is true of conventional methods; 2) the

secret key is not generated and stored in volatile memory until it's needed. Usually, the response bitstrings of PUFs cannot be directly used as secret key due to the environmental noise such as temperature and supply voltage. Therefore, an intermediate processing step is required to extract a crypto key from the responses of PUFs. I will describe several of the popular methods used to decrease or remove the noise in the output bitstrings of PUFs in Section 2.7.

In recent years, Intellectual Property (IP) protection of FPGA hardware designs has gaining more attention and becoming a requirement for many IP vendors. In [54], the authors explained why PUFs can be a very valuable technology to protect a company's IP and present several cryptographic algorithms and protocols to use them in IP protection applications. In [29][49], new protocols are introduced for the IP protection problem on FPGAs based on public-key cryptography. In [32], the authors introduced the Butterfly PUF and stated that it promises to be a significantly secure way to protect IP with no additional costs in manufacturing.

In [55][53] , the authors stated that in order to make Radio Frequency Identification (RFID) tags resistant against many attacks, they can be linked to PUFs inseparably. The authors demonstrated that PUFs can securely authenticate an RFID with minimal overheads. The authors also stated the PUF-based unclonable RFIDs can be used in anti-counterfeiting and security applications.

2.5 Performance Metrics of PUFs

In order to apply PUFs in security applications such as encryption key storage, unique

device identifiers, they should possess several properties that determine the quality of bitstrings generated by these PUFs. Mainly, there are three characteristics to evaluate their performance. They are uniqueness, stability and randomness.

2.5.1 Uniqueness

This characteristic is used to evaluate how much one chip is different from one another. It is the basis for PUFs to serve as identifiers. The inter-chip HD will be applied to evaluate the uniqueness and decide how many PUF output bits are different from other PUFs. The histogram of the inter-chip HD can be characterized as a Gaussian distribution and summarized using an average value of μ_{inter}. The ideal inter-chip HD is 50%. Usually, the evaluation results of proposed PUFs cannot achieve the ideal uniqueness. Some publications have been focused on how to improve the uniqueness, e.g. in [56], the authors proposed a novel arbitration scheme called Response Generation according to Delay Time Measurement (RG-DTM), which divide the delay-time differences at regular interval and decide the response by the time domains of the interleaved response 0 and 1. In [57], the authors proposed an arbiter-based PUF circuit built on current starved inverters, whose drain currents are set by local current mirrors. This circuit amplifies process variations that result in great uniqueness when compared against a simple inverter chain. The experimental results in this literature show superior performance in uniqueness and reliability.

2.5.2 Stability (Reproducibility)

When repeatedly applying the same challenge to a PUF, ideally, the responses should

keep the same. Also, when PUFs are used in practical implementations they can be subjected to all kinds of environmental variations such as temperature, varying supply voltages and different voltage ramp-up curves. To my knowledge, there is no PUF that can avoid the 'bit flip' problem, and it must be dealt with before PUFs can be used for encryption applications. The intra-chip HD will be used to evaluate the stability of output bitstrings generated from PUFs.

In most literatures, the authors applied error correction or helper data schemes to gain good reliability [2][3]. However, the overhead associated with these blocks increase very quickly with increasing error correction capability and can become prohibitively large for many applications. Also, these schemes require additional helper or syndrome bits to be publicly stored, which may leak information. Recently, several new reliability improvement methods have been presented. In [58], the authors proposed an effective method of regenerating a finite and exact sequence of bits, which exposes response patterns and keeps secret the particular challenge that generate response patters. The experimental results show that this method can efficiently and reliably generate bitstrings under extreme environmental variations.

Later in 2012, Bhargava et al. demonstrated 3 reliability enhancing techniques for bi-stable PUF designs: directed accelerated aging (DAA), multiple evaluations (MA) and activation control (AC) [59]. From the measured results in this literature, these three techniques are able to reduce the percentage of unreliable bits by up to 40%, 83%, and 71% respectively.

2.5.3 Randomness

Besides uniqueness and stability described above, it is also important to analyze the randomness of output bitstrings generated from PUFs. Here, the NIST tests suite are applied to test the randomness of them. In [60], the authors used the NIST tests suite to evaluate the randomness of output strings for PUFs, resulting in excellent raw material for key generation and for authentication. In [61], the authors stated that the amount of randomness in the PUF output could be a significant limitation. They introduced a method of passing the PUF response to a shift-register, and discover that the randomness of the PUF output could be greatly increased while maintaining reliability. Also, the authors showed experimental results that authentication with the shifted response data is superior to that with non-shifted data.

2.6 Evaluation Methods of PUFs

2.6.1 Hamming Distance

In order to evaluate the uniqueness and stability for a PUF implementation, HD has been introduced to measure these metrics. The test analyzes the relationship between CRPs' HD for each PUF. The inter-chip HD and intra-chip HD are introduced to evaluate these two metrics of PUF realizations. In order to study the statistical behavior of PUFs, all inter-chip and intra-chip HD measurements are often combined and summarized as histograms. In many cases, the HD histograms can be characterized as a Gaussian distribution and summarized using a μ_{inter} and μ_{intra} and a σ_{inter} and σ_{intra} .

For a particular challenge, the inter-chip HD between two different PUF instantiations is the distance between the two responses, resulting from applying this challenge simultaneously to both different PUFs. The inter-chip HD, when expressed as a fraction, the ideal value is 50%, i.e., half of the bits are different on average.

For a particular challenge, the intra-chip HD between two measurements on the same PUF instantiation is the distance between the two responses, resulting from applying this challenge twice to the same PUF. Intra-chip HD is strongly tied to random noise and systematic environment variations, such as those introduced by temperature and supply voltage. The intra-chip HD expresses the average noise in the response, and reflects reproducibility. When expressed as a fraction, the ideal value is 0%, i.e., the bit string can be reliably regenerated across different environmental conditions.

2.6.2 NIST SUITE

The National Institution of Standards and Technology (NIST) test suite is regarded as an industrial standard to test cryptographic random number generator (RNGs) and recently has been widely used to evaluate the randomness of bitstrings generated by PUFs [62] [63]. The NIST Test Suite has totally 15 tests [1], including Frequency Test, Frequency Test within a Block, Runs Test, Longest Run of ones in a Block, Binary Matrix Rank Test, Discrete Fourier Transform Test, Non-overlapping Template Matching Test, Overlapping Template Matching Test, Maurer's “Universal Statistical” Test, Linear Complexity Test, Serial Test, Approximate Entropy Test, Cumulative Sums Test, Random Excursions Test. For many of them, it is assumed the bit sequence is large, on order of

10^3 to 10^7. Therefore, parameters are chosen according to NIST recommendations, e.g., the length of bit sequence n>100 for Frequency Test, n>38,912 for Binary Matrix Rank Test, block length M=32 for Block Frequency Test, block length m=2 for approximate Entropy test and block length m=5 for Serial Test.

2.7 Techniques of Improving the Stability of PUFs

2.7.1 Threshold Mechanism

For some PUFs, the response bit is generated by comparing two analog values such as delay and voltage. When the compared two values are very similar, there exists the possibility that they maybe cause bit flipping due to the environmental variations. For example, in the first measurement, value1>value2; however, due to the small difference of these two compared values and the environmental variations, in the second measurement, it is possible that value1<value2. In this case, the bit generated by the PUFs is considered as an unstable bit. The bistrings produced from PUFs can not be used in the application of encryption if they contains unstable bits. As we know, people usually use the error correction and Helper Data to deal with the bit-flipping problems of PUFs. However, due to the additional overhead and possibility of leaking information of these schemes, a 'threshold' mechanism can be setup and used as an alternative to deal with the 'bit-flips' to improve the stability of output bitstrings. According to the idea of the 'threshold' mechanism, when applying each challenge to PUFs and comparing two analog values, the bits generated by those that are less than the threshold values will be discarded.

Therefore, the threshold mechanism should serve several primary goals: 1) it needs to avoid bit-flipping under different environmental noise; 2) it also needs to preserve as many strong bits as possible for each chip; 3) it should make the number of strong bits consistent across chip.

2.7.2 Majority Voting and TMR

Majority voting is a mechanism that the final result is decided according to the majority, that is , more than half the votes. For our PUF experiments, majority voting is the process of collecting multiple measurements through repeatedly sampling, and determining one output bit based on the individual value of the single measurement.

I also use a technique called Triple Modular Redundancy (TMR) [64] to improve the stability of the output bitstrings. The basic principle of the TMR is fault-tolerant form of N-modular redundancy, in which three system perform process and that result is processed by a majority voting system to produce a single output. If any one of the three systems fails, the other two systems can correct and mask the fault. In this dissertation, I will apply the basic idea of TMR into the stability improvement of PG-PUFs. The process is as follows: I investigate this technique by using fixed-length bitstrings, e.g., 256 bits. A TMR-based bitstring is created during enrollment by copying the first 256 strong bits into the first copy of the fixed-length bitstring. The second two copies are created by parsing the remaining strong bits and searching for matches to the first copy. The positions of the matching bits are indicated by writing a '1' in the public storage bitstring, while the positions of the skipped bits are indicated by writing a '0'. Later during regeneration, the

public storage bitstring is consulted to determine which challenges are to be used to re-construct the 3 copies of the bitstring. Once created, the final bitstring is obtained by majority voting on each bit.

Chapter 3

Two Primitives of a PUF Based on Resistance Variations in Metals

This chapter will follow the structure of our paper [65], and mainly focus on the following several aspects. Firstly, two implementations of PUFs for both 65 nm and 90 nm technologies are described in Section 3.1. Secondly, the experimental techniques and procedure are given in Section 3.2. Finally, the experimental results are analyzed by statistical tests in Section 3.3.

3.1 Experimental Setup

3.1.1 Test Chip Architecture: 90 nm chips

Fig. 3.1 gives a block diagram of the 90 nm test chip architecture. The chip padframe consists of 56 I/Os, and surrounds a chip area of approx. 1.5 mm x 1.5 mm. Four PADs labeled PS_1, PS_2, NS_1, NS_2 along the top of the figure refer to voltage sense connections, the 'P' version for sensing voltages near V_{DD} and the 'N' version for voltages near GND.

These four terminals wire onto the chip and connect to 85 copies of a Stimulus/Measure circuit (SMC). The SMCs are distributed across the entire chip (see small rectangles) as two arrays, a 7x7 outer array and a 6x6 inner array. Although not shown, a scan chain connects serially to each of the SMCs to allow each of them to be controlled.

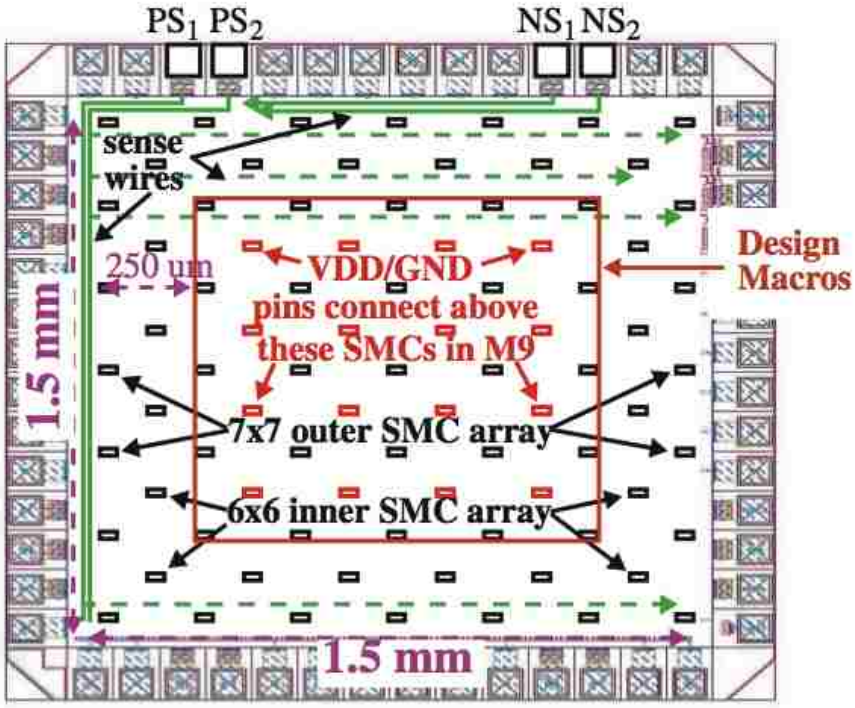


Fig. 3.1: Block diagram of 90 nm chips, with voltage sense pins along top and two arrays of SMCs, a 7x7 outer array and a 6x6 inner array.

The schematic diagram of the SMC is shown in Fig. 3.2. A pair of large 'shorting transistor', capable of sinking approx. 10 mA of current through the power grid when enabled, are shown along the bottom of the figure. A set of 16 'pseudo' pass gates (hereafter referred to as transmissions gates or TGs) serve as voltage sense devices. Eight

of the TGs connect to 8 (of the 9) metal layers that define the V_{DD} stack-up of the power grid, as shown on the left side of Fig. 3.2, while the other 8 connect to the GND stack-up. Scan FFs and 3-to-8 decoders allow exactly one of the pass gates to be enabled in each of the stack-ups.

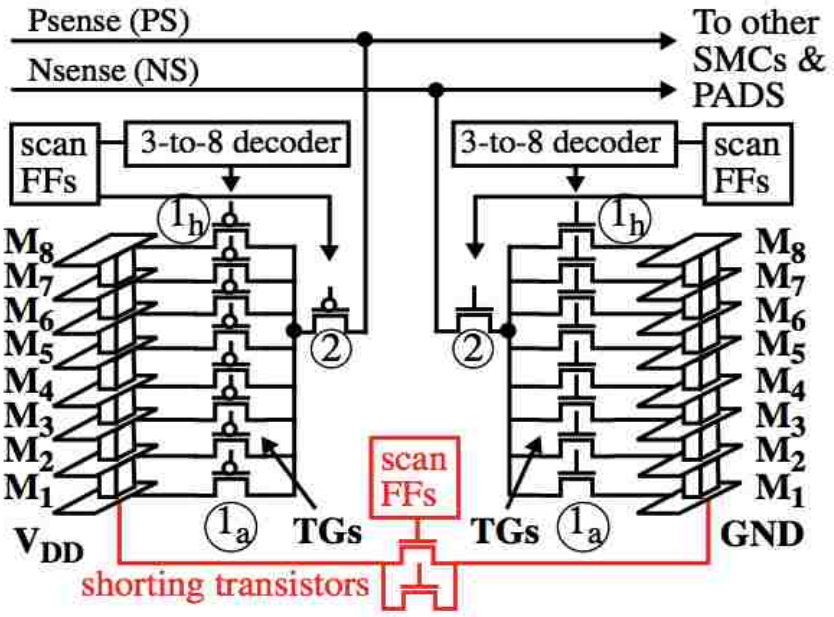


Fig. 3.2: SMC schematic in 90 nm chips.

An additional TG connects to the drains of the 8 stack-up TGs, labeled as '2' in Fig. 3.2, one for V_{DD} and one for GND. Separate scan FFs control the connection to the chip-wide wires that route to the P/NS_X pins of Fig. 3.1. The configuration and control mechanism allows any V_{DD} and GND voltages to be measured using off-chip voltmeters. We refer to these experiments using PG90, for power grid 90 nm chips.

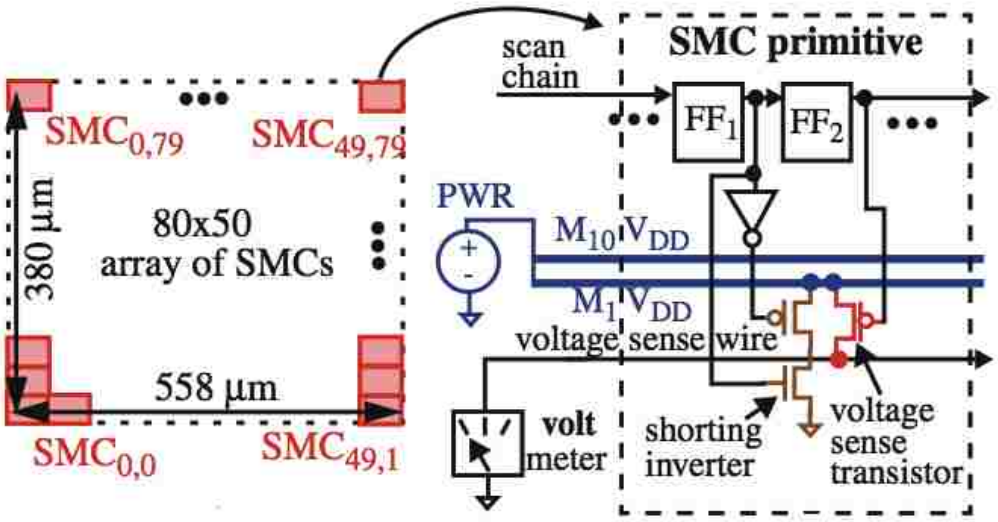
A 'challenge' in our experiments is applied by configuring the scan chain to 1) enable the shorting transistors within an SMC, and 2) enable two TGs in the same SMC,

in particular, the TG labeled 2 in Fig. 3.2 and one from the group 1_a through 1_h. Once enabled, the voltage drop/rise is measured on the NS and PS pads using voltmeters.

Although the details of the power grid are not shown, approx. 20% of the routing resources are used to define it, as would be typical of a high-performance commercial power grid. Moreover, the metal wires defining the power grid are at least 3 times wider than the minimum width, and via arrays (as supposed to single via) are used to connect one metal layer to the next. The low resistance associated with these wires and via arrays produces voltage drops in some experiments of 500 μV or less. Although our off-chip instrumentation can measure voltage drops at high resolution (approx. 5 μV), these levels will challenge the capabilities of on-chip 'instruments'. We discuss solutions to these issues and others as lessons learned in the following sections.

3.1.2 Test Chip Architecture: 65 nm chips

The 65 nm test chip architecture is shown as a block diagram in Fig. 3.3(a). A 50x80 array of SMCs are distributed, in close proximity, over a region that spans 560 m by 380 m. The power grid is wired in a mesh configuration over 10 metal layers using wide metal wires and via arrays. The details of the SMC are shown in Fig. 3.3(b). Each consists of a shorting inverter, a sense transistor and two scan FFs. The shorting inverter draws approx. 1 mA and introduces a 5-10 mV drop on the V_{DD} grid. The single sense transistor allows only M1 voltages to be sensed. We refer to these experiments as PG65.



(a) (b)

Fig. 3.3: a) Block diagram of 65 nm chip and (b) details of the SMC.

3.1.3 Challenge Scenarios

There are several 'challenge' scenarios possible in the PG90 and PG65 experiments. The basic approach we take in our experiments is to enable the shorting devices within each SMC, as well as the corresponding pass gates, one at a time, and then measure the voltage drop/rise produced at each location. A bit in the digital response of our PG-PUF is obtained by comparing two voltages obtained from a pairing of SMCs on a single bit.

In order to increase the size of the bitstring generated from the PG-PUFs, each of these voltages can be compared with other voltages in various combination to produce a bit string. We focus our analysis on bit string generated by using each voltage in (n-1) comparisons, where n is the total number of voltages measured from one chip. Bit strings

constructed in this manner are referred to as all combinations or AC.

As is customary, we randomize the order in which the comparisons are made. On chip, this can be accomplished using an LFSR and a seed. The process is modeled in our experiments using the functions srand (seed) and rand ( ) from the C programming library. In order to show that the characteristics of the bit strings are insensitive to the value of the seed, we report statistical results using 10 different seeds.

In addition to randomizing the order of the pairings, we found that periodically inverting the comparison order within pairings produced better results. In particular, inverting every group of three comparisons worked well for both the PG90 and PG65 experiments. For example, for a randomized set of pairing numbered 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, etc., pairings numbered 3, 4, 5, 9, 10, 11, etc. are inverted. Note that the inversion operation occurs after the randomization, which makes this process trivial to implement on chip.

3.2 Experiment Techniques

3.2.1 Bias Issues

The voltage measured from the chips in our experiments consists of three basic components: bias, regional variation and noise component. From the PUF perspective, only the regional variation component is important, and the other two components actually work to reduce randomness and stability, resp. Bias is a systematic voltage trend that is introduced by, e.g., a non-uniform distribution of power port connections to the power grid (the case for the PG90 chips) or non-uniform power grid mesh (the case for

the PG65 chips). Any type of systematic voltage trend will produce bits that are biased to 0 or 1 across chips, and needs to be reduced or eliminated in practice. We first discuss the biases that exist on our data sets and then methods of dealing with them.

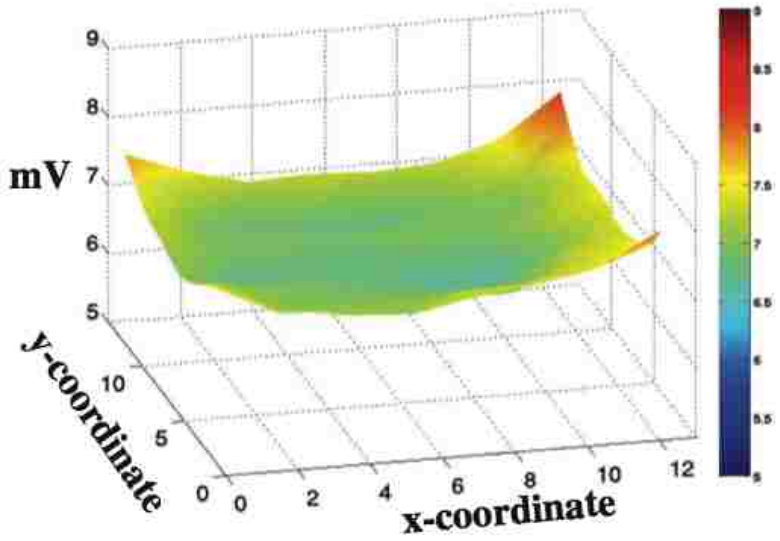


Fig. 3.4: V_{DD} profile using SMCs of a 90 nm chip.

Fig.3.4 show the voltage drop profile obtained from PG experiments on one of the 90 nm chips as each SMC is enabled. The voltage drop profile is derived from the M1 sense transistors on the V_{DD} grid. The (x,y)-plane in the figure represents the position in the array of the SMC in the 2-D array. From the bowl-like surface, it is clear that the voltage drops are larger along the edges of the power grid than in the center. This systematic voltage trend is caused by the non-uniform distribution of the power port connections, which from Fig. 3.1 are located over the 'Design Macros' in the center region of the chip, i.e., there are no power ports along the edges.

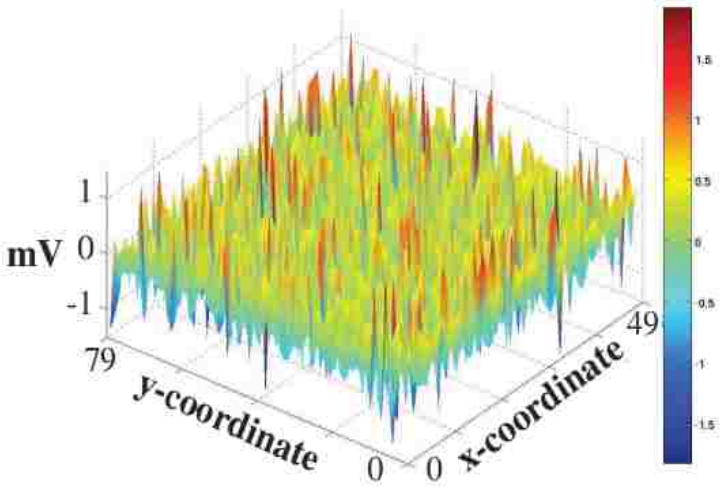


Fig. 3.5: Voltage drops from 4,000 SMCs on a PG65 chip.

Although much less apparent, a small bias also exists in the voltage drop profile shown in Fig. 3.5, which is derived from one of the 65 nm chips. Here, the voltage differences computed for the SMCs are plotted for each of the 4,000 elements of the array shown in Fig. 3.3. The majority of the voltage variations (approx. +/- 1 mV from Fig. 3.5) is introduced by regional resistance variations in the metal defining the power grid. However, a small 'saw-tooth-shaped' bias exists between adjacent rows of approx. 200 μV, which is introduced by a pattern in power grid metal mesh that is different for the even and odd rows of the array. This subtle pattern is revealed by computing an average voltage using the voltages measured across each row, for each of the 80 rows in the array. Averaging reduces the random variations allowing the bias effect to be more easily observed.

Fig. 3.6 plots the average values on the y-axis for each row identified along the x-axis. It is clear that the average voltages are smaller for even-numbered rows, than for odd-numbered rows. Moreover, the points labeled 'edge effects' reveal a second source of bias that is introduced by the power grid architecture. In this case, the increase in the voltage drops/rises is caused by the close proximity of these SMCs to the edge of the power grid.

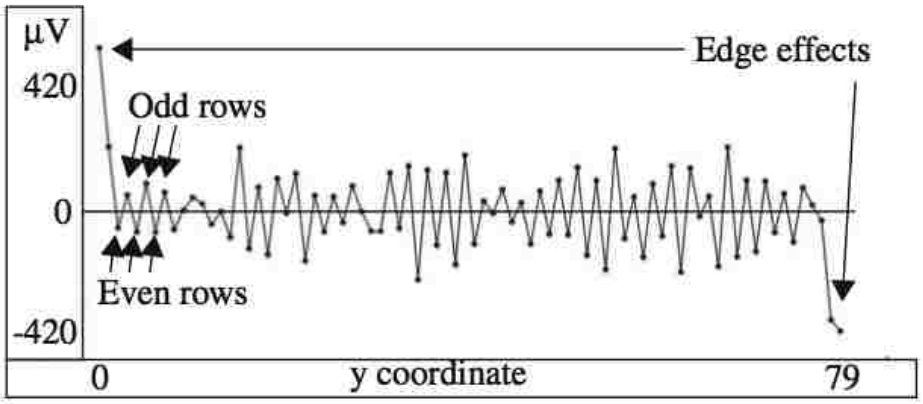


Fig. 3.6: Average voltage drops across 80 rows of PG65.

3.2.2 Dealing with Bias

There are several ways of eliminating the bias from the voltage measurements in these chips. The multi-layered architecture of the SMCs used in the PG90 chips allow voltage drops/rises to be computed across each of the metal layers. In order to reduce bias effects and correlations that exist in the V_{DD} and GND stack-ups, we compute inter-layer voltage drops/rises, which can be computed by subtracting pair-wise, the voltages measured from the consecutive metal layers, i.e., V_{M1}-V_{M2}, V_{M2}-V_{M3}, etc. These Voltage differences, called power grid voltage difference (PGVD), also allow the PUF to leverage the

independent resistance variations that occur in each of the metal layers of the power grid, and given their differential nature, significantly reduces bias effects such as those discussed in reference to Fig. 3.4. The bit string is then produced by comparing these voltage differences, e.g., if V_{M1}-V_{M2} in the V_{DD} stack-up at SMC_1 is greater than the voltage V_{M1}-V_{M2} in the V_{DD} stack up at SMC_2, then a '1' is generated, otherwise a '0'.

In the PG65 chips, each SMC has only one 'voltage sense' transistor connected to the V_{DD} grid, and therefore, computing inter-layer voltage drops is not possible. In this case, we avoid the bias by restricting voltage comparisons between SMCs positioned on every other row (to deal with the “saw-tooth” effect) and eliminate those positioned along the edges of the array (to deal with “edge effects”). In particular, we eliminate the SMCs along the left and right columns, the bottom row and the top two row. This reduces the size of the array to 77x48, and restricts comparisons between SMCs on the remaining 39 odd numbered rows and the 38 even numbered rows. We refer to this scheme as bias avoidance.

We also propose a more general technique to eliminate bias which is applicable to any type of PUF measurement, including, for example, RO frequencies and delays. For PUFs based on voltage measurements, the method first divides the voltage drops by a value that upper bounds the noise level. The general idea here is to eliminate the smaller noise variations, effectively reducing the range of values observed in a set of repeated samples for a given challenge to 1 or 2 distinct values. Once the voltage drops are re-digitized, the remainder from applying a modulus is used in the comparison operation to produce the bit. The modulus is chosen to preserve the regional (high frequency)

variation while simultaneously 'trimming off' the bias effects present in the high order bits. This approach is also applicable for PUFs that compare digital values, e.g., counter values in RO-based PUFs. In this case, the modulus operation eliminate systematic, across chip shift in frequency.

3.2.3 Bit Stability

In our experiments, we found that unstable bits, defined as bits that are susceptible to 'flipping' because their voltages are very similar, actually reduce several quality metrics associated with the overall bit string, including inter-chip HD and NIST statistical test results. Moreover, including unstable in the bit string requires the inclusion of error correction and Helper data schemes, that weaken security and increase overhead.

An alternative scheme that is able to identify and discard unstable bits works as follows. First, process is applied to determine the approx. noise level, either by using measurements carried out after manufacture or by using an on-chip noise evaluation process. In either case, noise levels can be determined empirically by applying a set of challenges repeatedly and examining the stability of the bit strings produced. In cases where bit flips occur, the magnitude of the difference between the voltages being compared is used to determine a threshold, that defines an upper bound on the voltage difference required to ensure bit stability. A safety margin is added to the threshold to account for environmental variations that can occur later in the field.

The usage scenarios that enables this process to be applied where exact regeneration of a bit string is required works as follows. During the initial bit string

generation, the threshold method is used to identify the unstable bits. For each unstable bit, its numbered position in the sequence challenges applied to generate the bit string (for a given seed) is recorded in non-volatile memory. Later, during regeneration, thresholding is disabled and the non-volatile memory is consulted to determine which challenges to skip during bit generation. To further enhance protection against one-time events, such as voltage spikes or scan configuration errors, repeated sampling and majority voting can be used again in the regeneration process to arrive at a final bit string. We investigate this process in the experimental results section below.

3.2.4 Statistical Characterization of the Bit Strings

Several statistical techniques are applied to evaluate the 'quality' of the bit strings produced by the PG PUFs. Hamming Distance (HD) is defined as the number of bits that are different when two bit strings are compares. An average inter-chip HD is compared using all combinations of bit string from the chip population. The best result occurs when exactly half of the bits from any two bit strings are different, i.e, when the HD, expressed as a percentage, is 50%. An intra-chip HD is computed using all combinations of bit strings obtained from one chip in the population, i.e., using the bit strings produced when the challenges are repeatedly applied. An average intra-chip HD is computed by averaging all of the individual intra-chip HDs. The ideal value in this case is 0%, i.e., each chip is able to reproduce the same bit string given the same set of challenges.

The statistical tests developed at NIST are also applied at significance level of 0.01 (the default) [1]. In general, the NIST tests look for 'patterns' in the bit strings that are not

likely to be found at all or above a give frequency in a 'truly random' bit string. For example, long or short strings of 0's and 1's, or specific patterns repeated in many places in the bit string work against randomness. The output of the NIST statistical evaluation engine is the number of chips that pass the null hypothesis for a given test. The null hypothesis is specified as the condition in which the bit-string-under-test is random. Therefore, a good result is obtained when the number of chips that pass the null hypothesis is large.

3.3 Experimental Results

3.3.1 PG90 Experiments

We collected data from 60 copies of the 90 nm chips and generated bit strings using the AC (all combinations) scenarios described in Section 3.1.3 . Although the SMC is designed to sense voltages in any of the first 8 metal layers, the voltage drops above M_4 are smaller than those on the lower metal layers. This occurs because the metal wires in the upper metal layers are much wider (and thicker) than wires in the lower portion of the grids. Given the concern we expressed in Section 3.1.1 regarding the capabilities of on-chip instrumentation, we restrict our analysis to voltages measured in the lower 4 metal layers.

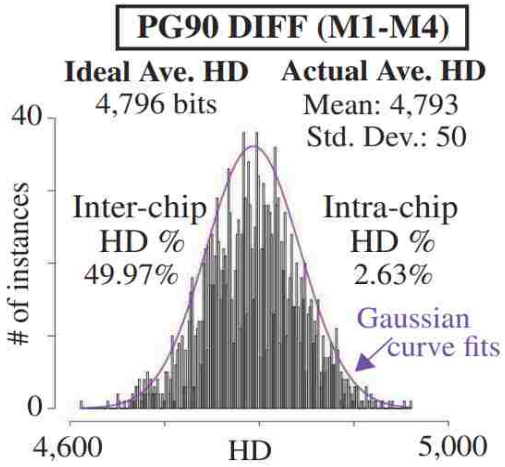
The focus of the PG90 experiments is on inter-layer resistance variations. Inter-metal layer resistance variations can be captured by computing voltage differences between consecutive metal layers. Since there are 8 TGs in the V_{DD} and GND stacks as

shown in Fig. 3.2, which indicating that 7 PGVD can be computed per stack. Given these constraints, each chip generates 85 SMCs*3 metal layer pairings=255 PGVDs for each of the V_{DD} and GND stacks. The bit string is then produced by comparing these voltage differences, e.g., if V_{M1}-V_{M2} in the V_{DD} stack-up at SMC_1 is greater than the voltage V_{M1}-V_{M2} in the V_{DD} stack-up at SMC_2, then a '1' is generated, otherwise a '0'. We focus our analysis on bitstrings generated by comparing each PGVD with all others generated using the same metal layer pairing.Therefore, the total number of bits per chip is 85*84/2 per metal layer pairing *3 metal layer pairing*2 grids=3,570*6=21,420 bits.These experiments are referred to as PGVD or DIFF. For comparison purposes, we also carry out a similar analysis using the absolute voltages (referred to as ABS). In this case, the number of bits increases to 28,560 bits because there are 4 metal layers.

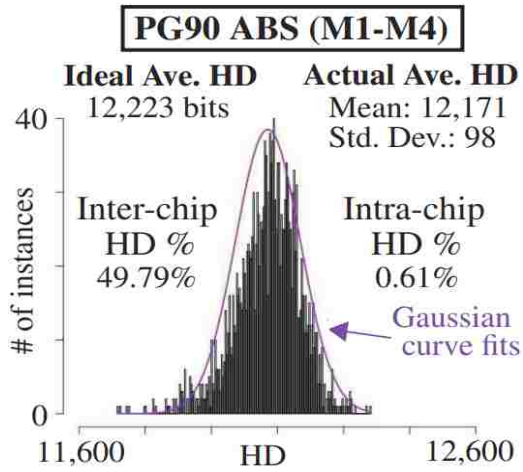
As indicated in Section 3.2.3 , our methodology eliminates unstable bits by comparing the voltage difference of the pairing with a threshold. To increase confidence, repeated sampling is also employed during bit stability evaluation. Therefore, for a bit to be considered stable, the voltage difference must exceed the threshold in all samples. For the DIFF experiments, a 25 μV threshold was sufficient to ensure that no bit flips occurred for voltage pairings where the absolute value of the analog difference across all samples (5 in this case) was larger than this value. The threshold increased to 50 μV for the ABS analysis. Bear in mind that the threshold actually guards against voltage variations as large as the threshold in each voltage of the pairing. For example, a voltage pairing defined as (6.000 mV, 6.025 mV) in the first sample would need to vary to values larger than (6.050 mV, 6.000 mV) in all subsequent samples to violate a 50 μV threshold

model.

The distribution of the HDs for the ABS and DIFF analyses are shown in Fig. 3.9(a) and (b), resp. HD is plotted along the x-axis against the number of instances on the y-axis.



(a)



(b)

Fig. 3.7: Distribution of HDs using (a) stable DIFF (M1-M4) (b) stable ABS (M1-M4) bit strings from PG90 exps.

The total number of the instances is given by all combinations of the chips' bit strings, i.e., 60*59/2=1,770. After removing unstable bits using to the algorithm described above, the bit string sizes reduce by approx. 46% on average to 11,600 bits for DIFF and by approx. 12% to 25,121 bits for ABS. The results are also shown in Table. 3.1. Bear in mind that these percentages do NOT represent the number of bit flips. The actual average intra-chip HDs are 2.63% and 0.61%, respectively. The size of bit strings used in the HD analysis are slightly smaller at 9,592 and 24,446 respectively. This adjustment is necessary because the HD analysis must be carried out on bit strings of equal length. To accomplish this, the chip with the shortest stable bit string is used to define the length of all bit strings.

	# of chips	**Total raw bitstring**	**Threshold value**	**Average survived rate with thresholding**	**Average stable bitstring with thresholding scheme**
DIFF	60	21420	25 μV	54.00%	11600
ABS	60	28560	50 μV	88.00%	25121

Table 3.1: Results for DIFF and ABS analyses with the thresholding scheme for PG90 exps.

Although the number of bits discarded as unstable using the threshold method is relatively large, the benefits of constructing a stable bit string in this fashion are significant. First, the average inter-chip HD for DIFF improves from 38.2% (not shown but with unstable bits included) to 49.97% as shown in the figure. For ABS, the improvement is even more dramatic, from 8.4% to 49.79%. The results can be seen more clearly in Table. 3.2. The superimposed Gaussian curve on the DIFF distribution of Fig.

3.7(a) illustrates the distribution to close to ideal, and reflects the power of differential analysis. The ABS distribution, on the other hand, is skewed somewhat to the left, which indicates that a component of the bias discussed earlier in reference to Fig. 3.4 is still present.

	# of chips	Average inter-chip HD without thresholding	Average inter-chip HD with thresholding	Ideal inter-chip HD
DIFF	60	38.20%	49.97%	50.00%
ABS	60	8.40%	49.79%	50.00%

Table 3.2: Improvement of inter-chip HDs for DIFF and ABS analyses with the thresholding scheme for PG90 exps.

Fig. 3.8 shows a bargraph of the number of passing chips (z-axis) for a subset of the NIST tests (x-axis) using 10 different seeds (y-axis). Only those tests applicable to bit strings of length 9,952 are applied. As indicated earlier, the best result is a 'pass' for all 60 chips. Although difficult to see, an ideal score of 60 was achieved in 40 (of the 90) cases. The worst result occurs for the NonOverlapping Template test. The height of the bars for this test represent the average number of passing chips across 148 templates. Although the chips performed very well on this test overall, there are 37 cases (of a total of 1,480) where the minimum number of passing chips was not met (57 is required according to NIST). The smallest number of passing chips among these fails is 54, with the majority (20) failing by only 1 chip. In a second set of seed trials (not shown) a couple of other tests failed, e.g., Cumm. Sums test, but never by more than 2 chips. Moreover, all of Pvalue-of-the-Pvalues tests passed, indicating the P-values are uniformly

distributed between 0.0 and 1.0.

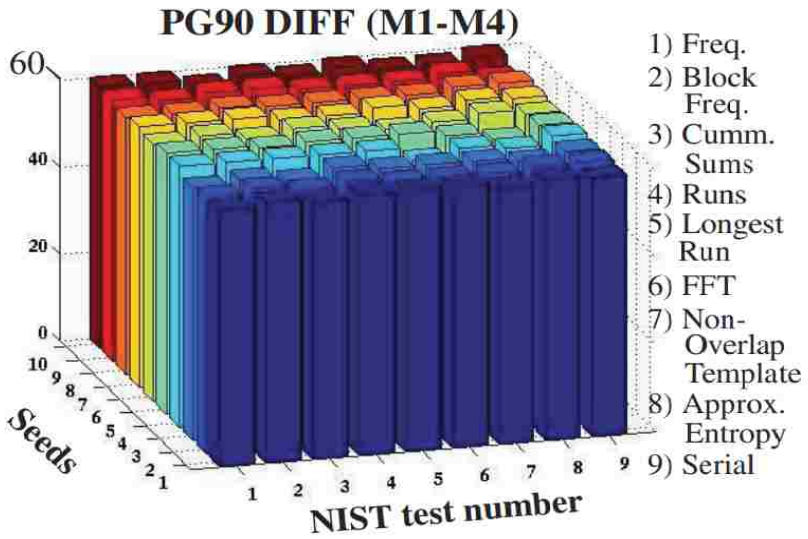


Fig. 3.8: NIST test suite statistics using DIFF (M1- M4) stable bit strings in PG90 exps.

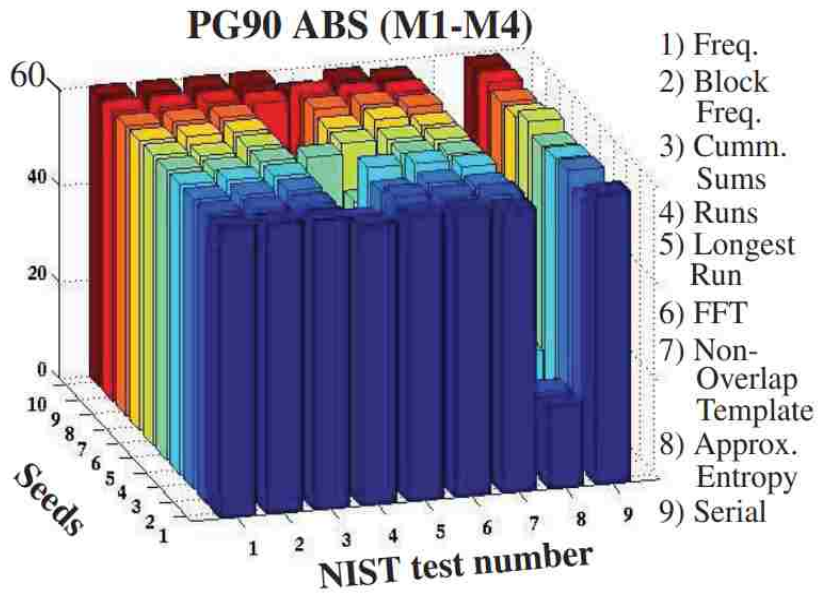
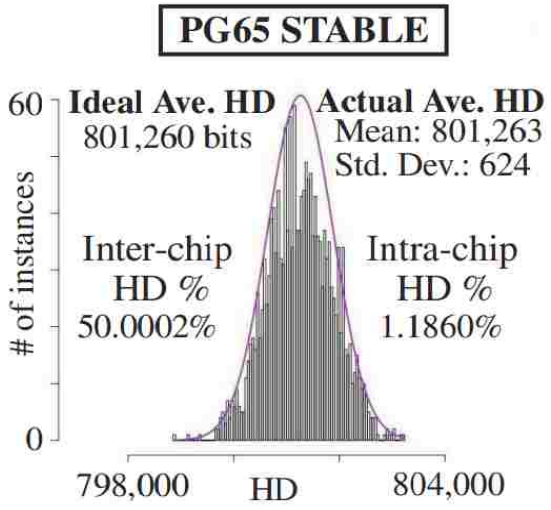


Fig. 3.9: NIST test suite statistics using ABS (M1-M4) stable-bit strings in PG90 exps.

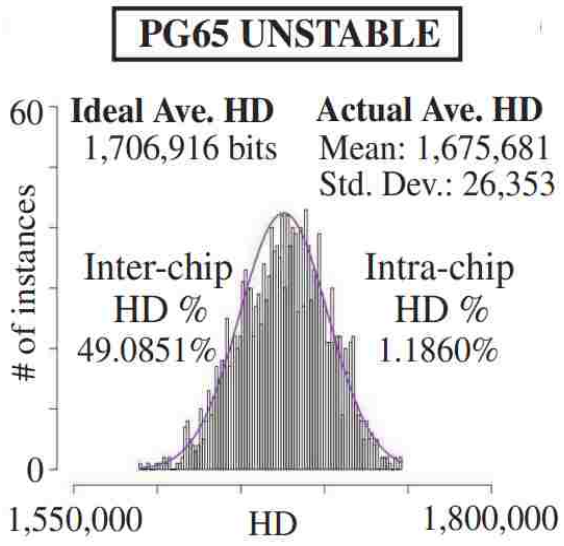
In contrast, the results for ABS data shown in Fig. 3.9 reveal that the bit strings fail the Runs and Approx. Entropy tests by as many as 50 chips in some instances. Overall, these results indicate the bit strings generated from the DIFF (M_1-M_4) version of the PG PUF are of cryptographic quality.

3.3.2 PG65 Experiments

As described in Section 3.2.2 , the bias avoidance scheme pairs voltages from 39 odd-numbered rows + 38 even-numbered rows in all combinations to produce bit strings of length 3,413,832 in the PG65 experiments. We repeatedly applied the challenges 10 times to each of the 58 chips at 25°C, and used a voltage threshold of 250 μV to eliminate all bits in the repeated samples. As indicated earlier, only V_{DD} voltages in M_1 can be sensed in the 65 nm chips, so the analysis is classified as ABS.



(a)



(b)

Fig. 3.10: Distribution of HDs using (a) stable ABS (b) unstable ABS bit strings from PG65 exps.

The distribution of the inter-chip HDs is shown in Fig. 3.10(a). The length of the stable bit strings are 2,052,003 bits on average, which represents approx. a 40% decrease in the original length. The results are summarized in Table 3.3.

	# of chips	**Total raw bitstring**	**Threshold value**	**Average survived rate with thresholding**	**Average stable bitstring with thresholding scheme**
ABS	58	3413832	250 μV	60.00%	2052003

Table 3.3: Results with the thresholding scheme for PG65 exps.

The intra-chip HD is approx. 1.2%, which indicates that a significant number of bits that are discarded as unstable do not flip in any of the 10 samples. The benefits of

discarding these bits are, once again, the improvements obtained in bit string quality and the elimination of on-chip error correction hardware.

In order to illustrate the improvement in quality, Fig. 3.10(b) shows the distribution of the inter-chip HDs using the full length 3,413,832-bit strings. The average inter-chip HD drops from 50.0002% to 49.0851% when the unstable bits are included. Moreover, the standard deviation of the distribution is significantly larger, e.g., 624 vs. 26,353.

The bargraphs in Figs. 3.11 and 3.12 give the NIST statistical test results for the stable and unstable bit strings, resp. The results from all 15 NIST tests are displayed along the x-axis, for 10 different seeds plotted along the y-axis. The ideal value is 58 for all tests except NIST tests 12 and 132. A numerical analysis of the data indicates that 58 of the bars reach the ideal value in Fig. 3.11 while only 35 achieve this status in Fig. 3.12.

The stable bit strings pass all tests except for 7 (of a total of 1,480) NonOverLapping Template tests. 6 of the 7 fails are by only 1 chip (54 chips passed instead of the required 55) and 1 failed by 2 chips. In addition, 3 PValue- of-the-Pvalue tests failed in this group of 1,480 tests. Moreover, three of the seeds passed every test. Overall, these results indicate that the stable bit strings are high quality and can be used in cryptographic applications.

The larger number of fails in the unstable bit string results of Fig. 3.12 indicate that these bit strings are lower in quality. For example, only 1 seed passed the NIST Frequency test, which measures the balance of ‘0’s and ‘1’s in the bit strings. Overall, 50 of the 150 (10 seeds * 15 NIST tests) fail. However, the fewest number of passing chips

is 41, which is better than the worst case result obtained in the PG90 ABS M1-M4 analysis described in Section 3.3.1 .

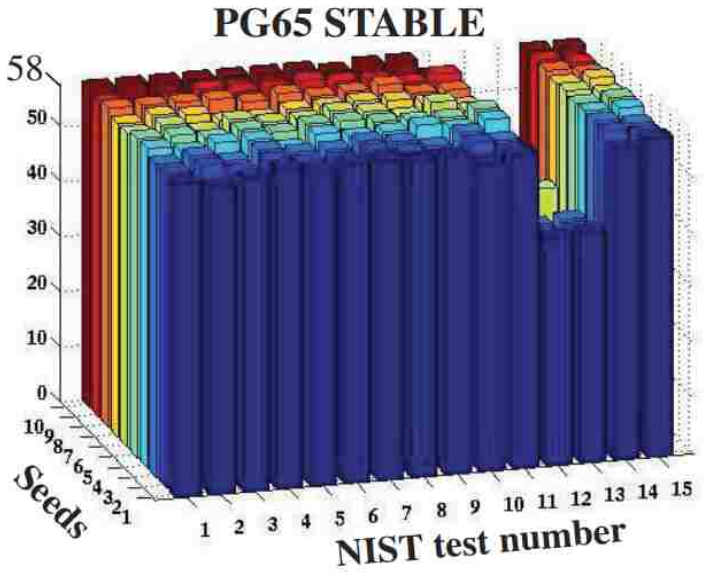


Fig. 3.11: NIST test suite statistics using stable bit strings for PG65 exps.

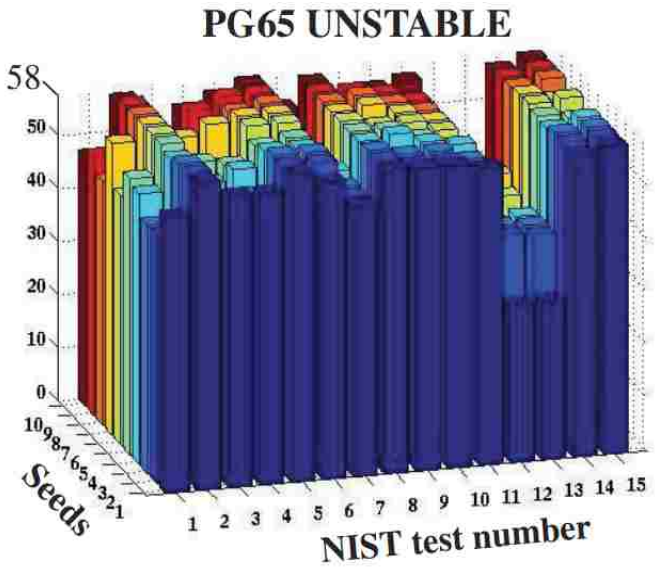


Fig. 3.12: NIST test suite statistics using unstable bit strings for PG65 exps.

3.3.3 Modulus Technique for Bias in PG65 Experiments

We investigate a second strategy for dealing with bias using a subset of the data from the PG65 experiments where bias is particularly evident. In these experiments, we do not use the bias avoidance scheme. The inter-chip distribution of HDs shown in Fig. 3.13(a) is derived from the 4,000 SMCs paired with their 'vertical' neighbor. Bias is reflected here in a lower-than-ideal average HD of 46.5% and in the bi-modal shape characteristic of the distribution.

The distribution labeled 'Modulus Op." shown in Fig. 3.13(b) is derived using the same pairings. The voltages in this case are first manipulated (as discussed in Section 3.2.2) by dividing by a noise threshold of 350 μV and then using the remainder from a modulus operation (11 is used in this analysis) in the bit generation process. The removal of the bias improves the average HD to 50.1% and re-shapes the distribution to conform better to a Gaussian. Although this technique is effective, it is no substitute for a well-designed architecture that minimizes bias. Fig. 3.13(c) shows the distribution obtained when the bias avoidance scheme is used to illustrate this concept.

Note that the length of the bit strings is smaller (3,600) because of the pairing constraints. Although the average HD is slightly less than ideal at 49.7%, the shape and standard deviation of the distribution are both superior to those of Fig. 3.13(b).

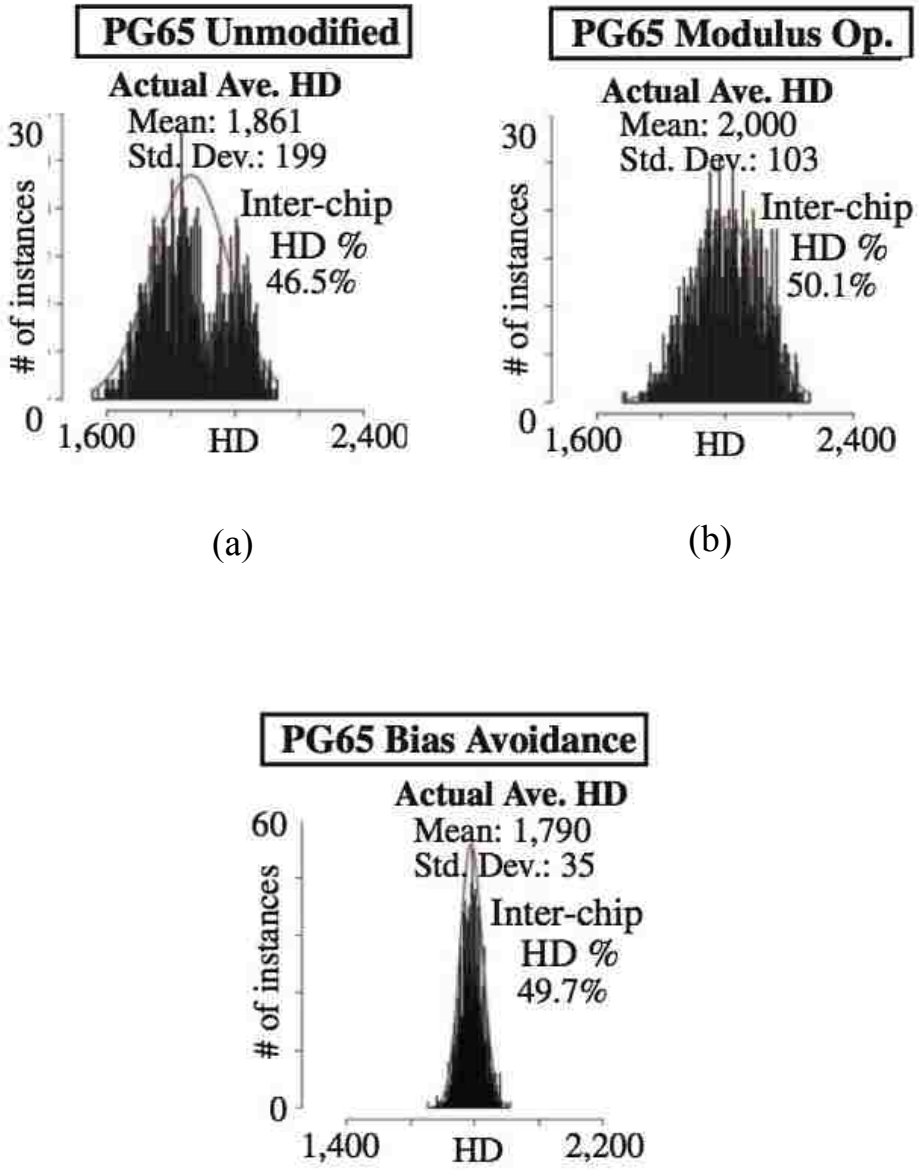


Fig. 3.13: HD analysis of bit strings using unmodified voltages (a) with modulus op. (b) and using bias avoidance (c) for PG65 exps.

3.4 Conclusion

Two PUF implementations, which leverages resistance variations of the power grid, are discussed and analyzed. Experimental results are reported for chips fabricated in 90 nm and 65 nm technologies. A voltage threshold technique is investigated that eliminates unstable bits and is shown to significantly improve inter-chip HD and the results from NIST statistical tests. Both of these two PUF primitives are shown to generate cryptographic quality bis strings of length upto 1.6M bits.

Chapter 4

Stability Analysis of PG-PUFs with Environmental Variations

In Chapter 3, all the experiments for the PG-PUFs were carried out under room temperature (25^{o}C). Actually, PUFs are sensitive to environmental variations and will detract from their ability to generate the same signature (reproducibility or stability). Since PG-PUFs leverage the passive component of metal resistance to generate bit strings, theoretically, they should be less sensitive than other types of PUFs, i.e., transistor-based PUFs. In this section, we will carry out a series of experiments against different temperatures and voltages to evaluate the stability of PG-PUFs. Therefore, the main contribution of this chapter is focusing on determining the temperature and voltage (TV) stability of PG-PUFs. A significant benefit of using metal structure is that "noise-related" variations, such as those introduced by TV variations, result in linear changes to the measured voltages. This linear scaling characteristic allows the relative magnitude of two voltages to remain constant across changes in temperature and voltage, which, in turn, improves the stability of the PUF to bit-flips, when compared, for example to PUFs which leverage transistor-based PUFs.

In our experiments, we evaluate the PG-PUFs at 9 TV corners, i.e., over all combinations of 3 temperatures; -40^{O}C, 25^{O}C and 85^{O}C, and 3 voltages; normal and +/- 10% of nominal. The evaluation is carried out on a set of chips fabricated in IBM's 90 nm, 9 metal layer bulk silicon process. The stability of the bitstrings is measured using intra-chip HD and “probability of failure' techniques. Randomness and uniqueness are also evaluated using the NIST test suite and inter-chip HD methods. A bit-flip avoidance scheme is proposed and evaluated that reduces the probability of a failure to reproduce the bitstring to less that 1E-9. We also investigate an on-chip voltage-to-digital converter (VDC) and its stability across the 9 TV corners. This chapter is following the structure of our paper [67].

4.1 PGV Experiments and Challenge Scenarios

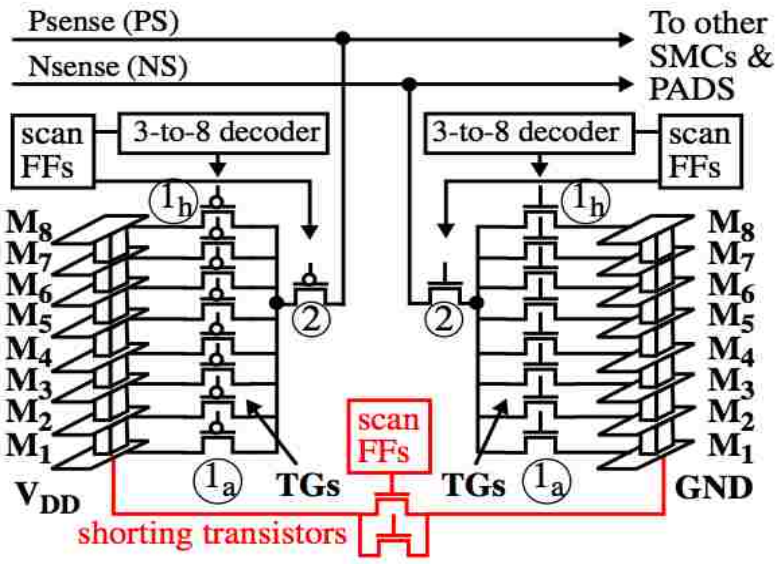


Fig. 4.1: SMC schematic in 90 nm chips.

The experimental setup in this chapter is still the same as described in 3.1.1 . Here, I will give the SMC schematic of 90 nm chip again as shown in Fig. 4.1. For the challenge scenarios, we still use the method of PGVD (or DIFF) presented in section 3.1.1 and 3.3.1 . As described in chapter 3.1.3 , each of the PGVDs can be compared with other PGVDs in various combinations to produce a bitstring and can get 21,420 bits. In the TV experiments, we also randomize the order in which the comparisons are made. In an on-chip implementation, this can be accomplished using an LFSR and a seed.

4.2 Overhead

On the 90 nm chip, each SMC occupies an area of 684 μm^2(38 μm x 18 μm), so the total area occupies by the array of 85 SMCs is 684x85=58,140 μm^2. The VDC occupies an area of 145 μm x 63 μm. The size of on-chip memory to store the helper data is approx. 2,680 bytes. Generally, the equation 4.1 can decide how many bits for the helper data are needed in order to remove the unstable bits with thresholding scheme.

# of bits of the helper data=# of bits (required bitstring size)*

(1/survived rate of the raw bitstring with thresholding scheme) (4.1)

For the helper data needed for both thresholding and TMR schemes, the size becomes 5 times of that for thresholding only, since the TMR-based bitstring of length n requires approx. 5*n strong bits to construct. For example, with the thresholding scheme, the survived rate of the raw bitstring is 35%; if a fixed-length bitstring of 256 bits is desired, the helper data should be 256*(1/0.35)= 731 bits. For both thresholding and TMR schemes, 731*5=3,655 bits are needed to determine the desired bitstring.

4.3 Experimental Results

4.3.1 Bit Stability

In our TV experiments, there also exists the problem of bit flipping or unstable bits. We still use the thresholding scheme to identify and discard unstable bits as described earlier in 3.2.3 . As we know, thresholding is carried out by first computing a threshold from the distribution characteristics of the PGVDs. The only difference is that in Chapter 3, the threshold is used with a fixed value for both 90 nm and 64 nm technology, while the threshold here is decided by using the percentage range of PGVD distribution. This is illustrated using the GND and V_{DD} PGVD distributions for a sample chip, $CHIP_1$, in Fig. 4.2. Each distribution contains 255 PGVD values, derived as described in Section 3.3.1 . The distance between the 10% and 90% points in the distribution is used to derive the thresholds for the thresholding algorithm, which is approx. 0.3 mV for GND PGVDs and 0.15 mV for the V_{DD} PGVDs for this chip. The limits at 10% and 90% are used to avoid distortions caused by potential outliers in the PGVD values for each chip. This method is more effective since the width of PGVD distribution of each chip is different. Using the percentage range to derive threshold can make sure every chip can produce the close number of strong bits.

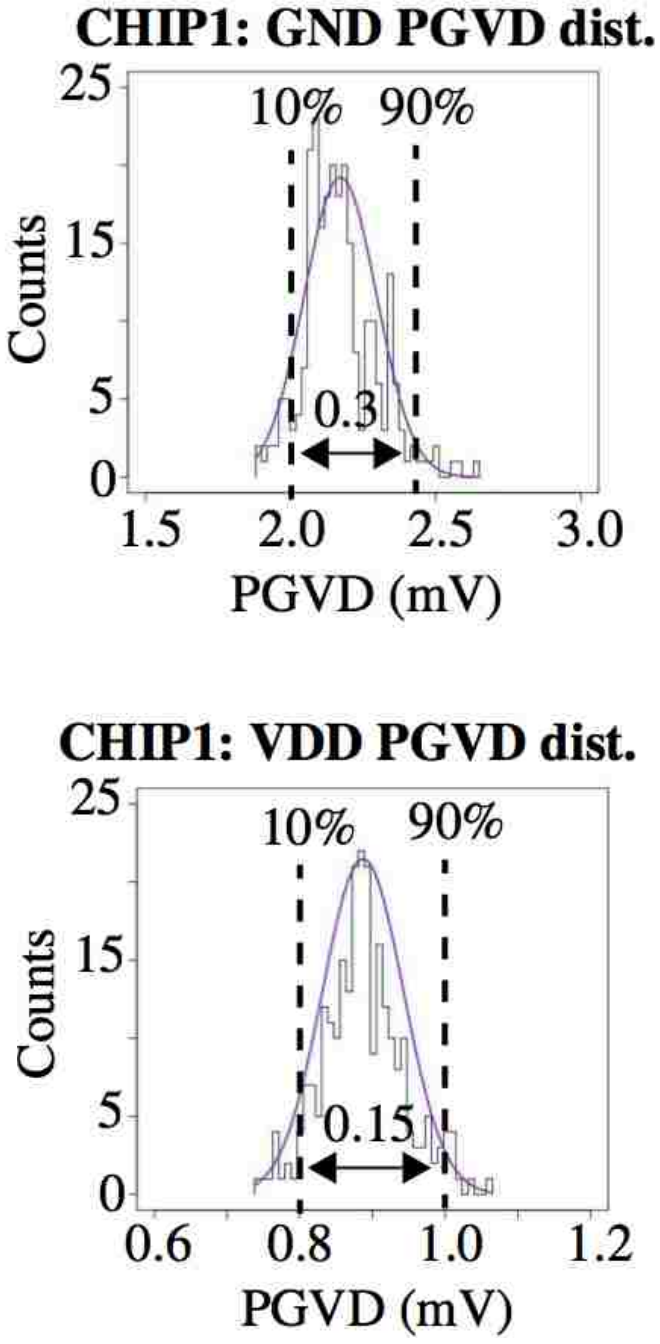


Fig. 4.2: CHIP$_1$ GND and V_{DD} PGVD distribution with Gaussian curve fits and 10% and 90% thresholds.

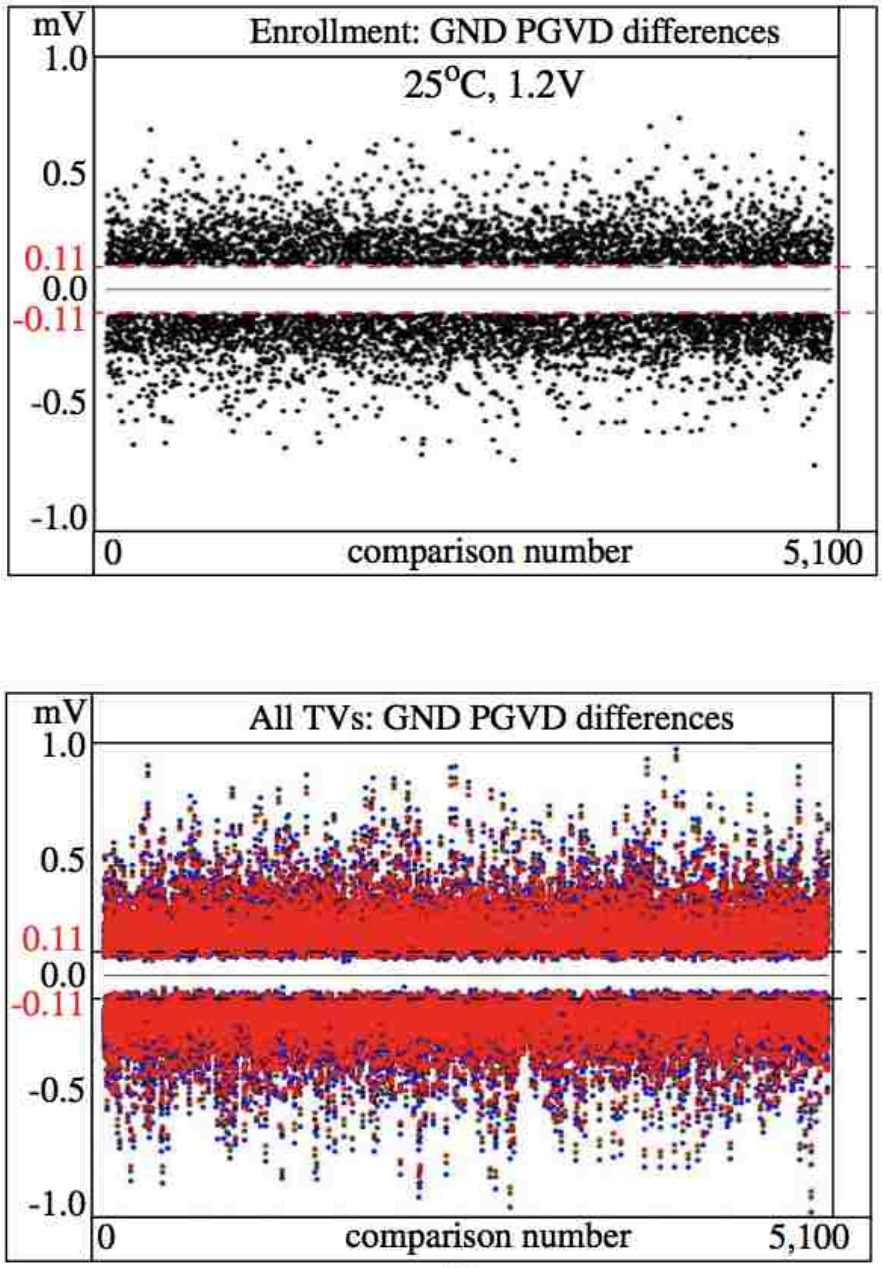


Fig. 4.3: CHIP$_1$ GND PGVD differences computed for bit generation during enrollment (a) and regeneration (b) at 9 TVs. Points in upper portion of plots generate '1's, points in lower portion generate '0's.

The thresholds are then scaled by a constant to produce the actual threshold used during bit generation. Fig. 4.3 provides an illustration of the bit generation process using the GND PGVDs for CHIP1. Both graphs plot the bit comparison number along the x-axis against the value of the difference between the two PGVDs being compared. Only the bits that survive the thresholding, called strong bits, are included in the plots, i.e., the x-axis shows only about half of the 10,710 comparisons. Points that appear in the upper portion of the figure generate a '1' bit while points in the lower portion generate a '0' bit.

Fig.4.3(a) shows only the points obtained from enrollment, which is carried out at 25^{o}C, 1.2V. The thresholds are depicted using two horizontal lines at 0.11 and -0.11 mV obtained from the distribution by a constant 0.37. In contract, Fig. 4.3(b) adds in the data points from the remaining 8 TV (regeneration) corners, colored-coded to indicate the temperature; green for 25 ^{O}C, blue for -40^{O}C and red for 85^{O}C. Close inspection reveals that some of the data points from regeneration appear within the threshold band of width 0.22 mV, centered around 0.0. Noise that occurs during regeneration causes points to move vertically, but as long as none move across the 0.0 line, no bit-flips occur.

The usage scenario that enables this process to be applied in situations where exact regeneration of a bitstring is required works as follows. During the initial bitstring generation, thresholding is used to identify the unstable bits. For each unstable bit, its numbered position in the sequence of challenges applied to generate the bitstring is recorded in public storage. Later during regeneration, during regeneration, thresholding is disabled and the public memory is consulted to determine which challenges to apply during bit generation.

4.3.2 Statistical Characterization of the Bitstrings

The results of applying the thresholding techniques to 63 chips tested under 9 TV corners are described in this section. An important concern regarding the thresholding technique deals with the fraction of bits that survive it. In our experiments, we found this fraction to be different for the GND and V_{DD} stacks. On average, approx. 50% of the comparisons using the GND PGVDs survive the thresholding, while only 20% of the comparisons survive using the V_{DD} PGVDs. The lower value for the V_{DD} PGVDs analysis occurs because of the increased noise levels on the V_{DD} grid relative to the GND grid. As a consequence, the average bitstring length reduces to approximate 7,765 (36.25%) bits from the original size of 21,420 bits. These bitstrings are, however, reproducible at all of the 9 TV corners.

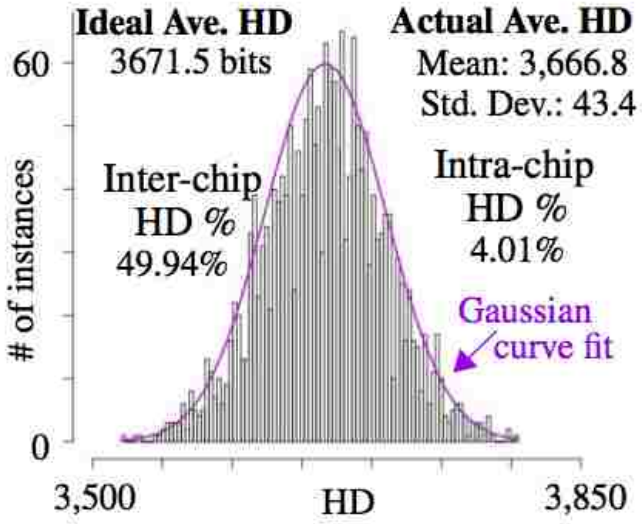


Fig. 4.4: Distribution of HDs using stable bitstrings from 63 chips. Number of HD is 1,953 using bitstrings of length 7,343 bits.

The true average intra-chip HD, which is a measure of the underlying bit stability across the TV corners, is computed as 4.01%. This value is obtained by analyzing the full length, i.e., 21,420-bit, bitstrings with thresholding disabled and counting the number of times a bit-flip occurs in each bit position across all pairings (9*8/2=36) of the bitstrings produced under each of the 9 TV corners for each chip. The average intra-chip HD, expressed as a percentage, is obtained by dividing the number of bit flips by 36*21,420, which is the total number of bit pairings inspected for each chip, and multiplying by 100. The value reported is the average of these percentages across all chips. Any value less than approx. 5% is onsidered high quality according to the published literature on PUFs.

Inter-chip HD, as indicated earlier, measures the uniqueness of the bitstrings, where the best possible result is 5%, i.e., on average, half of the bits in the bitstrings of any two arbitrary chips are different. Fig. 4.4 plots the distribution of inter-chip HDs. The 1,953 HDs included in the distribution are obtained by pairing the stable bitstrings from all chips under all combinations. The chip with the shortest stable bitstring is used to set the size of the bitstrings used in each HD calculation, requiring all bitstrings to be truncated to 7,343 bits. The average HD is 3,666.8 (49.94%), which is very close to the ideal HD of 3,671.5 (50.00%).

We also evaluate randomness using the NIST statistical tests at the default significance level of 0.01. Given the relatively short length of the stable bitstrings, only 11 of the tests are applicable. The bar graph shown in Fig. 4.5 gives the number of passing chips on the z-axis for each of the 11 tests on the x-axis, and for each of 10 different seeds on the y-axis. The number of passing chips is in reference to passing the null

hypothesis. The null hypothesis is specified as the condition in which the bitstring-under-test is random. Therefore, a good result is obtained when the number of chips that pass the null hypothesis is large.

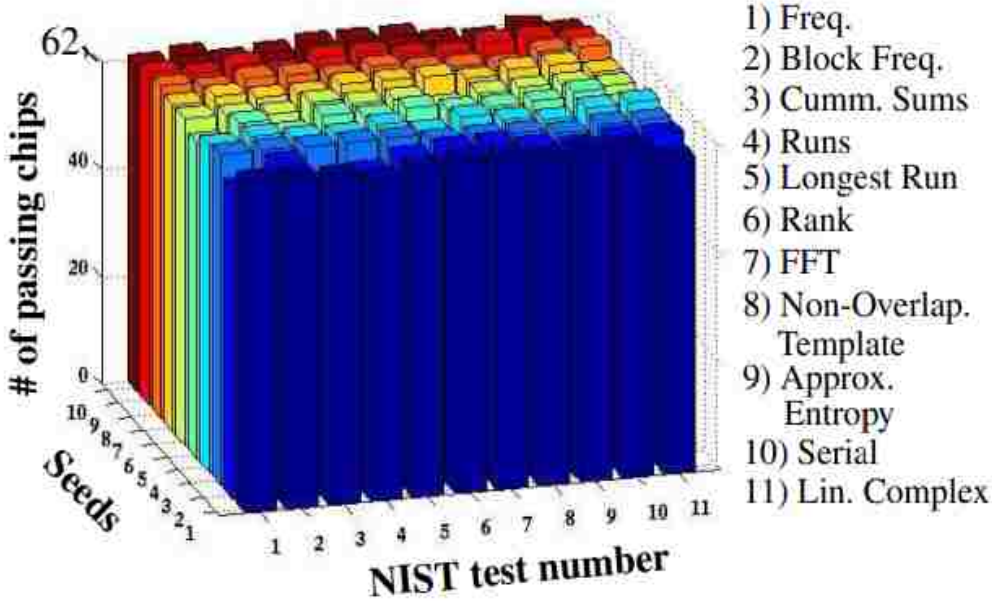


Fig. 4.5: Number of passing chips from NIST tests using 11 of the 15 applicable tests.

With 63 chips, NIST requirs that at least 60 chips produce a p value that is larger than the significance level (α=0.01), otherwise the whole test is considered 'failed'. Overall, of the 11*10=110 bars, 41 are full height indicating that all 63 chips passed the test, 39 bars have height 61 have 2 have height 60. Therefore, 109 bars of the 110 are equal to or larger than the required value to pass the test, and only 1 bars are below the threshold at 58. Moreover, all but 8 of Pvalue-of-the-Pvalues tests passed, indicating the P-values are uniformly distributed between 0.0 and 1.0. The fails in this category

occurred in the Rank and Non-overlapping Template tests, both which NIST recommends testing with much larger bitstrings than those used here. Overall, these all very good results and indicate the bitstrings are cryptographic quality.

4.3.3 Bit-Flip Probability Analysis and Triple-Module-Redundancy (TMR)

The large size of the bitstrings produced by the PUF can be used to further enhance their reliability over that provided by thresholding alone. This can be accomplished creating 3 copies of a fixed-length bitstrings from the sequence of strong bits produced by the PUF. The 3 copies can then compared as a means of detecting and correcting bit flips, in the spirit of a popular scheme used in fault tolerance called triple-module-redundancy or TMR. TMR is based on a majority voting' scheme in which the final bit for a given bit position is obtained by taking the majority across all 3 copies of the bitstrings.

We investigate this technique using fixed-length bitstrings of 256-bits. A TMR-based bitstrings is created during enrollment by coping the first 256 strong bits into the '1st copy' of the fixed-length bitstrings as shown in Fig. 4.6. The second two copies are created by parsing the remaining strong bits, searching for matches to the 1st copy. As described above for thresholding, the positions of the matching bits are indicated by writing a '1' in the public storage bitstring, while the positions of the skipped bits (and the weak bits encountered under thresholding) are indicated by writing a '0'. Later, during regeneration, the public storage bitstring is consulted to determine which challenges are to be used to re-construct the 3 copies of the bitstring. Once created, the final bitstring is obtained by

majority vote on each column as shown in Fig. 4.6. This allows the correct bitstring to be generated despite any single bit-flips that may occur in a column, such as the one shown in the last column of 'Redundant BS_1'.

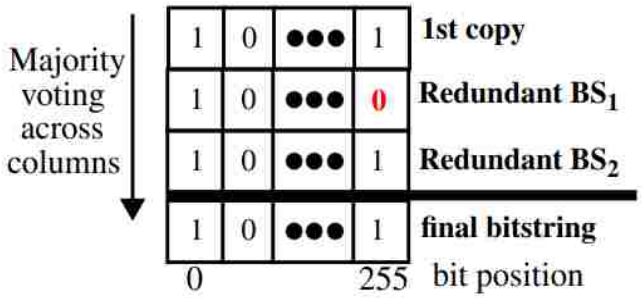


Fig. 4.6: TMR process for bitstring regeneration.

In order to illustrate the improvement provided by TMR over voltage thresholding alone, we iteratively decreased the GND threshold scalar given above as 0.37, in 0.01 steps down to 0.0. As the threshold is decreased, bit flips begin to occur in the thresholding-only bitstrings. A Thresholding-only 'probability of failure' curve can be constructed by counting the number of bit flips that occur in the bitstrings from all 63 chips and dividing it by the total number of bits. A similar curve can be constructed using TMR, but in this case, a bit flip is not counted unless it occurs in 2 or more of the 3 bits of a column as shown in Fig. 4.6. Moreover, the total number of bits used in the denominator for the TMR-based curve is reduced by a factor of 3 to account for the actual number used in the final TMR-based bitstring.

Fig. 4.7 plots to the data points for these two curves as well as two 'exponential-curve' fits to them. The GND threshold scaling constant is plotted along the x-axis against the probability of failure on the y-axis. The exponential curve fits allow the probability of

failure to be predicted for thresholds beyond (to the right) of the last recorded bit flip in our small population of chips. For example, the probability of failure using voltage thresholding alone at the 0.37 threshold is 6.5E-7. This improves by over three orders of magnitude to 2.4E-10 using the TMR-based scheme. Of course, the TMR-based scheme can be expanded to further improve bit-flip resilience by generating 5 (or more copies) of the bitstring, at the expense of increased usage of bits and public storage size.

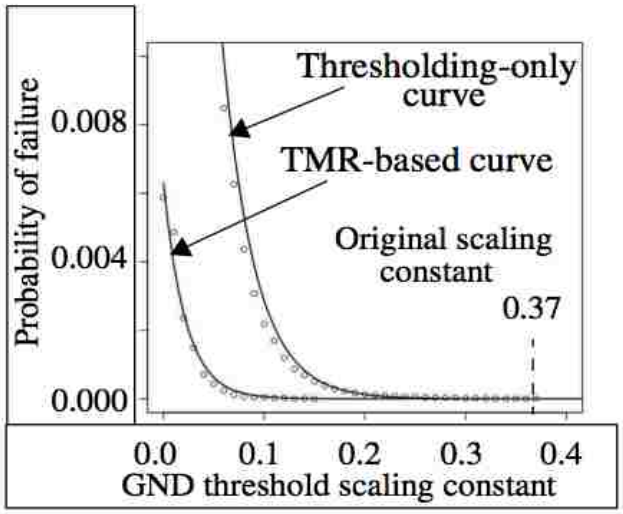


Fig. 4.7: GND threshold scaling constant vs. probability of failure (y-axis).

4.4 VDC experiments

The analysis presented in the previous sections is carried out on digitized voltages obtained from an off-chip voltmeter. In this section, we analyze the bitstrings generated by the PUF after digitizing the voltages using an on-chip voltage-to-digital converter (VDC) that is subjected to the same TV corners as the PUF itself. The VDC is similar in design to that described in [66] but is used in our application in a unique way.

The architecture of the VDC is shown in Fig.4.8. The VDC is designed to 'pulse shrink' a negative input pulse as it propagates down an inverter chain. As the pulse moves down the inverter chain, it activates a corresponding set of latches to record the passing of the pulse, where activation is defined as storing a '1'. A thermometer code (TC), i.e., a sequence of '1's followed by a sequence of '0's, represented the digitized voltages.

The VDC works by introducing a fixed-width (constant) input pulse, which is generated by the pulse generator shown on the left side of the Fig. 4.8. Two analog voltages, labeled Cal0 and Cal1 connect to a set of series-inserted NFET transistors in the inverter chain, with Cal0 connecting to NFETs in even-numbered inverters and Cal1 to the NFETs in odd numbered inverters (see call-out on right side of Fig. 4.8). The propagation speed of the two edges associated with the pulse are controlled separately by these voltages. The pulse will eventually die out at some point along the invert chain when the trailing edge of the pulse 'catches up' the leading edge. This is ensured by fixing Cal0 at a voltage higher than Cal1. The digital representation of the applied Cal0/Cal1 voltages can then be obtained by counting the number of '1's in the latches.

As described earlier, PGVDs are created by subtracting the voltages measured on consecutive metal layers in the power grid. Instead of digitizing these PGVs once-at-a-time with the VDC and then subtracting them, we carry out the difference operation in the analog domain by applying the two voltages from consecutive metal layers to the Cal0 and Cal1 inputs. The larger PGV from the lower metal layer, M_n, of the pair is applied to Cal0 while the PGV from the adjacent, higher metal layer, M_{n+1}, is applied to Cal1 (voltage drops are used for the V_{DD} grid voltages, e.g., V_{DD}-V_{Mn}).

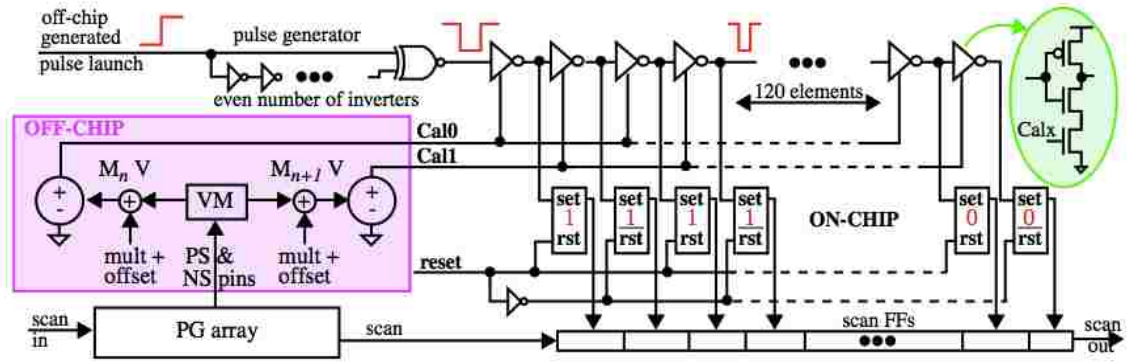


Fig. 4.8: Voltage-to-Digital Converter (VDC). On the left side is off-chip instrumentation that measures two voltages from the PG array, adds and offset and programs the Cal0/Cal1 inputs of the VDC on the right.

Fig. 4.8 shows how this is accomplished. The PG array is configured to enable the PGV on Mn to drive the NS (or PS) pin and an off-chip voltmeter (VM) is then used to digitize the value (some process as described for the original experiments). The PGV is then multiplied by 15 and added to an offset, and the voltage sum is used to program an off-chip power supply which drive Cal0. The exact same process is carried out for the PGV produced on metal layer M_{n+1} except the value is used to program a second off-chip power supply which drive Cal1. The multiplication and offset operations are necessary because the VDC requires the Cal0/Cal1 to be set between 500 mV and V_{DD} for proper operation.

Note that unlike the PGV experiments in 4.3, the on-chip VDC is subjected to the same TV variations as the PUF (as would be the case in actual implementation), and therefore its characteristics will vary as well. We developed a calibration process that 'tune' the offset voltage to compensate for some of the changes in VDC behavior but

since the measurements are differential, the VDC is able to self-calibrate and cancel out most of the adverse effects of TV variations by itself.

We carried out the same set of experiments and followed the same process as described in Section 3 and 4 on the 63 chips. The results are as follows. The average bitstring length after thresholding is 8,388 bits (39.16%) and the shortest one (used to truncate the bitstrings from the other chips for the statistical tests) is 7, 506 bits. Both of these numbers are slightly larger than the numbers obtained using the PGVs, as described in Section 4.3.2 , and indicates that the VDC compensates for some of the TV variations that occur in the measured PGVs.

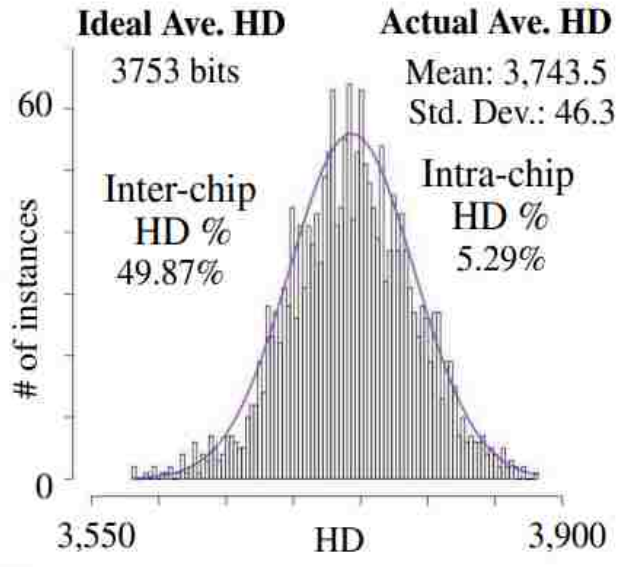


Fig. 4.9: Distribution of HDs using stable bitstrings from 63 chips. Number of HDs is 1,953 using bitstrings of length 7,506 bits.

On the other hand, the statistical test results for the VDC-based bitstrings are slightly worse than those presented for the PGVs. Fig. 4.9 shows the HD distribution of

the bitstrings and several statistical results, in the same format as Fig. 4.4 for the PGVs. Although the inter-chip HD is close to the ideal at 49.87%, the value obtained for the PGVs is slightly better (49.94%). Moreover, the standard deviation of 46.3 bits given in Fig. 4.9 is slightly larger than that given in Fig. 4.4, primarily due to the long tail on the left side of Fig. 4.9. The VDC-based bitstrings were tested using the same 11 NIST statistical tests listed in Fig. 4.5, but using only the first seed. Although most tests were passed, the Runs and Approx. Entropy tests failed with 57 and 49 chips passing, resp., and 20 of the 148 Non-Overlapping Template tests failed. From table 4.1, the difference of experimental results for both PGV and VDC experiments are listed.

	# of chips	**Total raw bitstring**	**Average survived rate with thresholding scheme**	**Average stable bitstring**	**Shortest stable bitstring**	**Average inter-chip HD for stable bitstrings**
PGV	63	21420	36.25%	7765	7343	49.94%
VDC	63	21420	39.16%	8388	7506	49.87%

Table 4.1: Results with the thresholding scheme for both PGV and VDC experiments across 9 TV corners for PG90 exps.

In summary, the digitization process carried out by the VDC works well but is not as efficient as the off-chip voltmeters at removing the bias that exist in the PGVs. In [67], we show a 'bowl-shaped' pattern exists in the M1 voltages across the 2-D array of SMCs and indicated that computing inter-metal layer voltage differences (as we do there) effectively eliminates it. The basic problem with using the VDC to compute the analog difference directly deals with the different sensitivities that exist for Cal0 and Cal1. In

particular, Cal1 has higher sensitivities than Cal0, and therefore, the amplification factors for voltages applied to Cal0 and Cal1 need to be different (we used 15 for both factors in our experiments. The asymmetry in the sensitivities behaves as follows. Assume that the M_n voltage from Fig. 4.8 increases by a fixed constant ΔV and the M_{n+1} voltage remains constant. Under these conditions, assume the TC for these two measurements is equal to x. In contrast, a similar scenario where the voltage M_n remains fixed and the M_{n+1} voltage increases by the same fixed constant ΔV does not result in the same ΔTC. Instead, the ΔTC is equal to y, where y>x. In other words, a delta change in the upper metal layer (M_{n+1}) voltage has a larger impact on the change in the TC than it does for an equivalent lower metal layer (M_n) voltage change. Therefore, the TCs weigh the voltage change in the lower metal layer less than a change in the upper metal layer, which distorts their relationship to the actual voltage difference.

4.5 Temperature-Voltage Stability Analysis

As stated in the earlier part of the thesis, leveraging metal resistance variations as the source of entropy for the PUF should be inherently more stable across environmental (TV) variations than leveraging transistor-based variations because metal resistance scales linearly with temperature and voltage. The PGVs used in the analysis presented in Section 4.1 actually include variation for both sources. Although the shorting transistors from Fig. 3.2 are very large (57x minimum size) and therefore exhibit smaller variations in comparison to minimum-sized transistors, they do introduce a component of entropy in the PGV analysis. The entropy works to improve the results, but the gain is reduced, as we show here, because of the increased sensitivity of transistor-based variations to TV

variations (hereafter TV noise).

In this section, we eliminate transistor variations by dividing the PGV voltages by the shorting current, and use the term PGERs, for power grid equivalent resistances, to refer to them. In order to get as 'pure' a form as possible of the PGERs, we also subtract the leakage voltage and leakage current from the values measured with the shorting transistors enables. The expression for PGER is given by the following equation 4.2.

$$\text{PGER} = \frac{V_{short} - V_{leak}}{I_{short} - I_{leak}} \tag{4.2}$$

The 4 measurements used to define the PGER each add measurement noise, which we separate and distinguish in this analysis from TV noise through sample averaging. We create PGER differences (PGERDs) by subtracting pairings of PGERs, as we did for PGVs in Section 4.1.

The objective of our analysis is to show that the PGERDs are more resistant to TV variations than are the PGVDs. In order to determine the magnitude of the TV variations (or 'noise') we calibrate the PGVD and PGERD data. Calibration removes the DC offsets introduced by TV noise in the data but preserves the variations. Calibration is carried out by computing the mean PGERD and PGVD over the entire set of SMCs for a given metal layer pairing and TV corner. Correction factors are then computed by subtracting the mean value at each of the TV corners from a reference TV corner. The reference is the data collected at 25^{o}C, 1.2V. The correction factors are then added to the corresponding data from the TV corners.

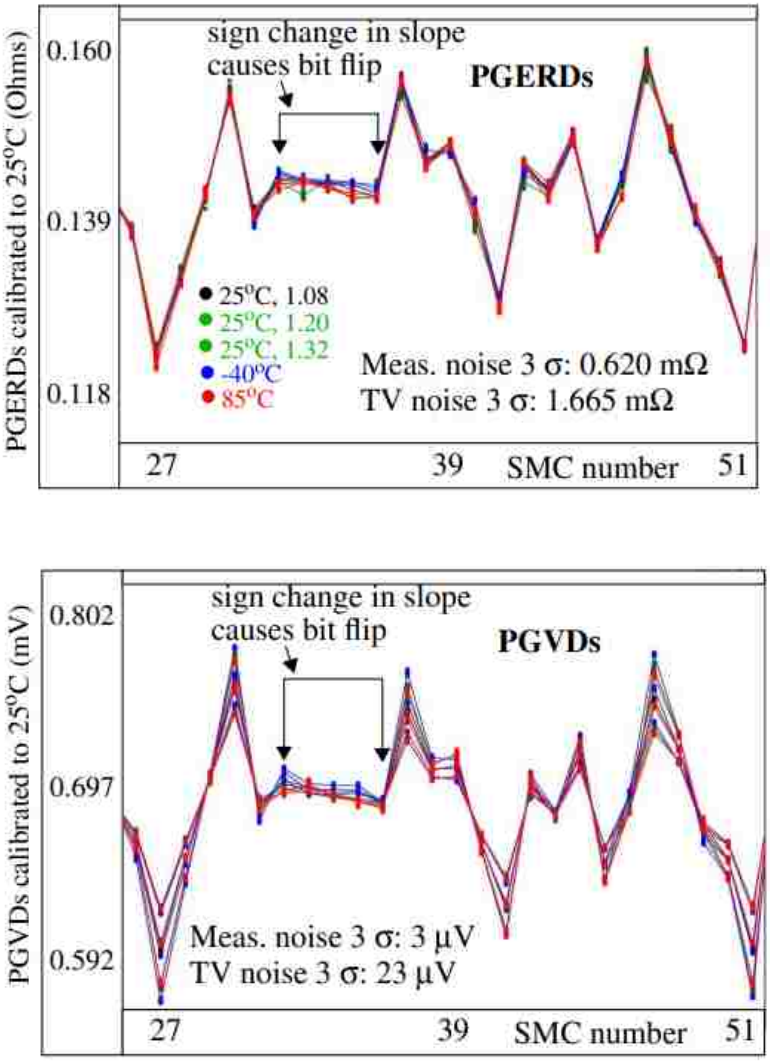


Fig. 4.10: Subset of PGERD (top) and PGVDs (bottom) from M2-M3 metal layer pairing for GND grid. Data from 9 TV corners is calibrated to 25°C, 1.2V. All 11 samples from 9 TV corners are also shown for each SMC.

A subset of the calibrated M2-M3 PGERDs and PGVDs computed using data from one of our chips as shown in the top and bottom plots of Fig.4.10 resp. The SMC number is given along the x-axis and the PGERD/PGVD calibrated to 25°C is plotted along the y-axis. Each plot shows 9 line-connected curves, one for each TV corner. Each point in the

curves is the average of 11 samples (the samples are also plotted as unconnected points to illustrate measurement noise). The averaging eliminates most of the measurement noise. Therefore,variations in the line-connected curves are introduced primarily by TV noise.

The 3σ values listed in the PGERD plot indicate that TV noise is approx. 2.7 times the measurement noise (1.665/0.620). Bit flips occur when the slopes of the lines between any two adjacent pairing of points reverses sign (see plots for examples). In contrast, the ratio increases to 7.7 (23/3) for the PGVD analysis. Therefore, TV noise is nearly 3 times more likely to introduce a bit flip in the PGVD analysis than in the PGERD analysis.

4.6 Conclusion

In this chapter, we analyze the statistical quality of bitstrings produced by a PUF that leverages resistance variations in the power grid wire of an IC. Experimental results are reported for chips fabricated in a 90 nm technology, and which are tested under 9 different temperature-voltage corners given by the industrial standard specifications. Voltage thresholding and TMR-based techniques are investigated as a means of improving the bit-flip resilience of the regenerated bitstrings. An on-chip voltage-to-digital converter is also investigated. The statistical results indicate that the power grid PUF is able to generate cryptographic quality bitstrings of significant length.

Chapter 5

Future Work

The experimental results given in Chapter 3 show that the bitstrings derived from metal resistance variations perform very well on statistical tests, and can be used in both identification and authentication. In Chapter 4, I apply different temperatures and voltages in order to evaluate the stability of the PG-PUFs against environmental noise. The experimental results also show that the proposed PG-PUFs are still feasible and can be used for real applications. However, in order to make the PG-PUFs more attractive, there is still plenty of work to do in the future. In this chapter, I will list several points that we will focus on in the following research work. First, a more effective VDC and bit generation engine will be developed on the new chip. Since the VDC circuit in Fig. 4.8 has the problem of susceptibility to differential power attacks (DPA), a new VDC circuit is proposed as shown in Fig.5.1, which is more resistant to DPA and also avoid the problem of different sensitivity described in section 4.4. Second, to make the PG-PUFs get larger manufacturing variations and more entropy, the power grid of the next-generation chip will be designed using the minimum size of metal wires. Third, since in our present experiments, we randomize the voltage comparisons using the software of C programming. In the next chip design, the LFSR and seed will be integrated on the chip.

5.1 Differential-Power-Analysis Resistant VDC

Besides the problem of different sensitivities for the VDC in Fig.4.8, there still exists another problem, which is its susceptibility to differential power attacks (DPA). DPA is a statistical technique that is used to steal secrets embedded within ICs. It works by deducing internal states (and secrets) of the IC by analyzing power supply transistors that are generated from operating a functional unit, such as the Advanced Encryption Engine. The pulse-shrinking behavior of the VDC makes it relatively easy to determine the TC code for a given voltage difference measurement. The power transient generated by the VDC simply stops when the pulse shrinks and disappears and therefore, the length of the power transient is proportional to the TC.

An architecture that addresses this issue is shown in Fig. 5.1. Here, the two GND PGVs from the M_n and M_{n+1} metal layers drive the even-numbered current starved inverters within two identical delay chains, one shown along the top of the figure and one along the top delay chain ahead of the rising transition introduced into the lower delay chain. Given that the M_{n+1} voltage is lower than the Mn voltage, the top delay chain propagates the edge more slowly, and eventually, the edge propagating along the bottom delay chain passes the top edges. Similar to the VDC in Fig. 4.8, as the edges propagate, each record a '1' in a latch as long as it precedes in time the edge on the other delay chain. Otherwise a '0' is stored. The duality of the delay chains causes complementary Tcs to be stored in the latches, which are subsequently transferred to the scan chain. An example test result is given in the center right of the figure which shows the complementary Tcs that are produced when the bottom edge passes the top edge at the 3rd latch.

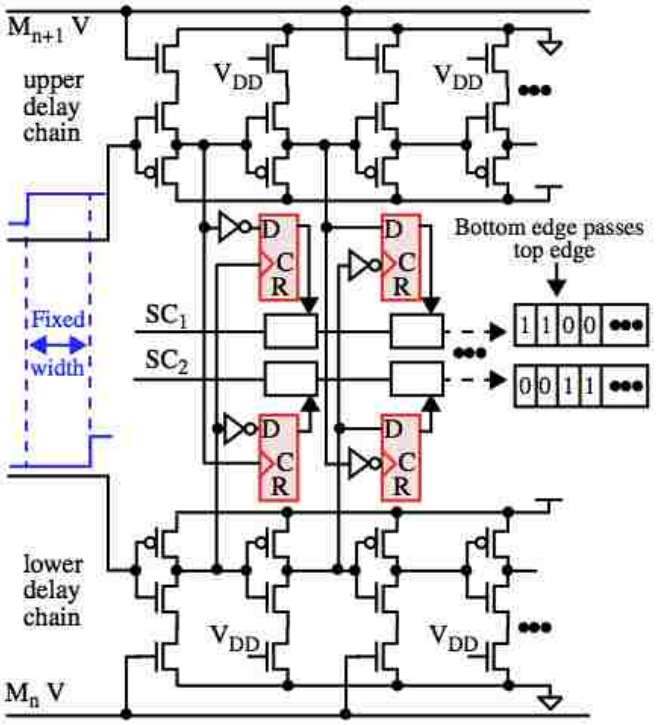


Fig. 5.1: Proposed differential-Power-Analysis (DPA) resistance VDC.

Although the VDC architecture is almost 2 times larger in area than the VDC shown in Fig. 4.8, it provides a significant advantage. The power transient signature remains constant when the bottom edge passes the top edge, so it is difficult or impossible for an adversary to determine the precise time at which this occurred. Although some information is revealed at the end of the power transient that reflects the additional time taken by the top edge to propagate to the end of its delay chain relative to the bottom edge, it requires higher signal-to-noise ratios when analyzing the power transient to correlate it to the actual PGV voltages. Moreover, it is possible in this architecture to introduce a 'stop condition' where the signal propagation is halted in the

top delay chain, effectively eliminating this type of information leakage. For example, by gating the V_{DD} inputs on the top delay chain, it is possible to turn these inputs off at the instant the bottom edge propagates off the end of its delay chain. This action halts the propagation along the top scan chain (and the corresponding power transient) and therefore 'hides' the difference in their delays.

5.2 An Improved Version of PG-PUF

An improved version of PG-PUF for a new chip will be designed, which will be submitted for fabrication at MOSIS in 130 nm technology. The power-grid in the new chip will be designed using minimum size metal and single via as a means of increasing the magnitude of the resistance variations and voltage drops and rises.

In the PG-PUF design for the new chip, we will still use the same SMC structure as in 90 nm technology. The main improvement of the new version PG-PUF is that we will integrate some analog and digital components required in an IP-block implementing a complete PUF system into the chip. These components will be added to next to SMC arrays as shown on the right side of Fig. 5.2.

The Digital Challenge Generator generates the scan chain configuration bits to enable SMC shorting inverters and voltage sense connections in one or more SMCs. The voltage sense wires are routed to the Voltage Comparator and Perturbation unit, which incorporates the analog-to-digital bit converter and bit stability control. The PUF Engine serves to coordinate the activities of the PUF IP-block components and acts as the interface to high level applications which request PUF signature bits.

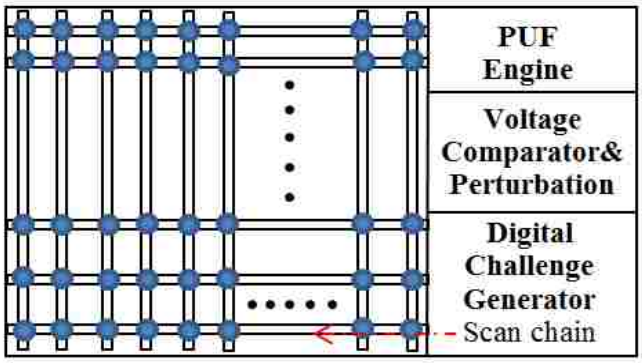


Fig. 5.2: PUF IP-block (blue circles represent SMC arrays).

The proposed Voltage Comparator and Perturbation unit is shown in Fig.5.3. It consist of an OpAmp array that compares two sense wire voltages on PS_1 and PS_2 as shown or on NS_1 and NS_2. The outputs of OpAmps are a a digital '1' if $PS_1>PS_2$, or '0' otherwise. The individual bits from each of the OpAmps in the array are collected in scan FFs and compared using a 'voting scheme' (not shown). The voter tallies up the responses and decides if the comparison generates a '1' or '0' PUF signature bit. The array of OpAmps and the voting scheme are designed to reduce the adverse effect of biasing offsets in the OpAmps themselves.

A critical component of (any) PUF system is a scheme to distinguish between stable and unstable bits. The voting scheme described above can be used to evaluate bit stability. For example, the voter may decide that the bit is unstable of the number of '1's produced by the OpAmp array is not larger by margin than the number of '0's, or vice versa. A second, complimentary scheme is shown in Fig. 5.3. A set of perturbation transistors are added to each SMC, that function to introduce a small dynamic voltage perturbation, either on the power grid as shown or on the voltage sense wires

themselves. The voltage variations introduced will cause the output of selected OpAmp to flip between '0' and '1' in cases where the voltages being compared are very similar. The glitch detector shown in the lower right of the figure is designed to detect the bit flip using an XOR gate and an SR-latch. The 'stability-indicator' output of the SR-latch will remain a '0' if the bit is stable and '1' otherwise.

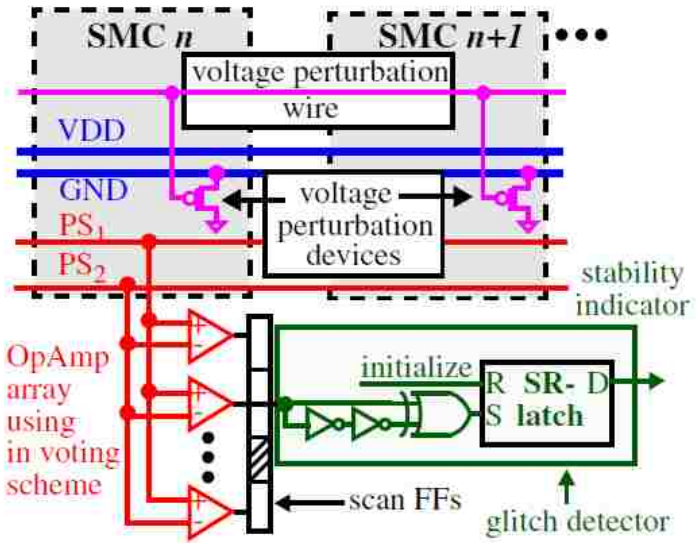


Fig. 5.3: Voltage Comparator and Perturbation circuit for on-chip bit stability evaluation.

The PUF engine collects the sequence of bits generated, partitioning the bit string into stable and unstable bits based on the stability evaluator(s). The stable bits represent the 'repeatedly random' group and can be used for authentication and encryption keys, while the second group is not repeatable and can be used for one-time pads and other applications that require a true random number. We plan to build the entire PUF system on a custom ASIC for experimental evaluation.

Chapter 6

Conclusion

Two implementations of a PUF based on the resistance variations of metal wires are presented. Experiments are carried out on two sets of chips, one set fabricated in IBM's 65 nm SOI technology and another fabricated in IBM's 90 nm bulk technology. The results show that metal resistance variations that occur in the power grid are able to produce bit strings that perform well under HD and NIST statistical tests. Both of these two implementations are shown to generate cryptographic quality bit strings of length up to 1.6M bits. Experiments are also carried out across different temperatures and voltages in order to evaluate the stability of PG-PUFs in 90 nm technology.

In Chapter 1, I described the importance of exploiting PUFs for security applications and introduced the advantages of PG-PUFs which leverage the resistance variations of the power grid of chips.

In Chapter 2, I briefly introduced the development of concepts of PUFs, and introduced the extensions of PUFs; Also, I listed several popular PUFs published in recent years and described the applications of PUFs. Finally, I gave some methods of evaluating the quality of PUFs and ways of improving the stability of PUFs.

In Chapter 3, I proposed two primitives of PG-PUFs. One is based on the technology of 65 nm. The PG-PUF for this technology can only measure the voltage from one metal layers. In order to increase the entropy and size of the bitstrings generated from our PG-PUFs, I proposed another primitive based on the 90 nm technology, which can measure the voltage from 8 metal layers of the power grid. By this way, the voltage difference between two consecutive metal layer, which can eliminate the bias of the power grid, can also be compared. The statistical results indicate both of these two primitives can produce cryptographic quality bit strings of significant length.

In Chapter 4, I applied different environmental parameters to the PG-PUFs of 90 nm technology in order to evaluate the stability of them. In the experiments, I used the temperature of -40^{o}C, 25^{o}C, 85^{o}C and the +/-10% of the nominal supply voltage and combine them into 9 TV corners. And then I carried out a series of measurements across these 9 TV corners. The statistical results show that the bitstrings generated from the PG-PUFs can still be used for cryptographic applications during different environment.

In chapter 5, I described the many facets of future work. I found some problems existing in the present VDC circuit structure: one is its susceptibility to DPA; another one is the different sensitivities that exist for Cal0 and Cal1. Therefore, I proposed a complementary VDC structure which can fix both of these two problems. I also proposed an improved PG-PUF structure for the next version chip design. In the new power grid structure, the minimum size of metals will be used, which will be expected to have larger manufacture variations and increase entropy. Finally, I presented my idea of integrating the LFSR and seed on to the next chip design.

References

[1]NIST: Computer Security Division, http://cs-rc.nist.gov/groups/ST/toolkit/rng/stats_tests.html.

[2] B. Gassend, D. Clarke, M. van Dijk, S. Devadas, "Controlled Physical Random Functions", in *Conference on Computer Security Applications*, pp. 149-160, 2002.

[3] Y. Dodis, L. Reyzin, A. Smith, "Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data", *SIAM Journal on Computing*, vol. 38, no. 1, pp. 97-139, 2008.

[4] R. Pappu, B. Recht, J. Taylor, N. Gershenfeld, "Physical One-Way Functions", *Science*, vol. 297, no. 5589, pp. 2026 - 2030, 2002.

[5] S. Nassif, "Modeling and Analysis of Manufacturing Variations", in *Conference on Custom Integrated Circuits,* pp. 223-228, 2001.

[6] K. Agarwal and S. Nassif, "Characterizing Process Variations in Nanometer CMOS", in *Conference on Design Automation*, pp. 396-399, 2007.

[7] B. Gassend, D. Clarke, M. van Dijk, S. Devadas, "Silicon Physical Random Functions", in *Conference on Computer and Communication Security*, pp. 148-160, 2002.

[8] B. Gassend, D. Lim, D. Clarke, M. van Dijk, S. Devadas, "Identification and Authentication of Integrated Circuits", *Concurrency and Computation:Practice & Experience*, vol. 16, no. 11, pp.1077-1098, 2004.

[9] J. Guajardo, S. S. Kumar, G. -J. Schrijen, P. Tuyls, "FPGA Intrinsic PUFs and Their Use for IP Protection", in *Cryptographic Hardware and Embedded Systems (CHES)*, pp.63-80, 2007.

[10] K. Kursawe, A. Sadeghi, D. Schellekens, B. Skoric, "Reconfigurable Physical Unclonable Functions", in *International Workshop on Hardware Oriented Security and Trust*, pp.22-29, 2009.

[11] Yingjie Lao and K. K. Parhi, "Reconfigurable Architecture for Silicon Physical Unclonable Functions", in *International Conference on Electro/Information*

Technology (EIT), pp.1-7, 2011.

[12] R. Maes and I. Verbauwhede, "Physically Unclonable Functions: A Study on the State of the Art and Future Research Directions", Information Security and Cryptography, pp.3-37, 2010

[13] J. D. R. Buchanan, R. P. Cowburn, A. V. Jausovec, D. Petit, P. Seem, G. Xiong, D. Atkinson, K. Fenton, D. A. Allwood, M. T. Bryan, "Forgery: Fingerprinting' Documents and Packaging", *Nature*, vol. 436, no. 7050, pp.475, 2005.

[14] G. Hammouri, A. Dana, B. Sunar, "CDs Have Fingerprints Too", in *Conference on Cryptographic Hardware and Embedded Systems* , pp.348-362, 2009.

[15] R. S. Indeck and M. W. Muller, Method and Apparatus for Fingerprinting Magnetic Media, U.S. Patent No. 5365586, 1994.

[16] G. DeJean and D. Kirovski, "RF-DNA: Radio-Frequency Certificates of Authenticity", in *Conference on Cryptographic Hardware and Embedded Systems*, pp. 346-363, 2007.

[17] S. Vrijadenhoven, "Acoustical Physical Unclonable Functions", Master's thesis, Technische Universitert Eindhoven, the Netherlands, 2004

[18] J. Lee, D. Lim, B. Gassend, G. E. Suh, "A Technique to Build a Secret Key in Integrated Circuits for Identification and Authentication", in *Symposium On VLSl Circuits Digest of Technical Papers*, pp.176-179, 2004.

[19] D. Lim, J. W. Lee, B. Gassend, G. E. Suh, "Extracting Secret Keys From Integrated Circuits", *IEEE Transactions on VLSI Systems*, vol. 13, no. 10, pp.1200-1205, 2005.

[20] U. Ruhrmair, F. Sehnke, J. Solter, G. Dror, "Modeling Attacks on Physical Unclonable Functions", in *Conference on Computer and Communications Security*, pp. 237-249, 2010.

[21] M. Majzoobi, F. Koushanfar, M. Potkonjak, "Techniques for Design and Implementation of Secure Reconfigurable PUFs", in *Transaction on Reconfigurable Technology and Systems (TRETS)*, vol. 2, no. 1, pp. 232, 2009.

[22] M. Majzoobi, F. Koushanfar, S. Devadas, "FPGA PUF Using Programmable Delay Lines", in *IEEE International Workshop on Information Forensics and Security*, pp.1-6, 2010.

[23] C. Yin and G. Qu, "Temperature-Aware Cooperative Ring Oscillator PUF", in *International Workshop on Hardware-Oriented Security and Trust*, pp.36-42, 2009.

[24] A. Maiti and P. Schaumont, "Improved Ring Oscillator PUF: An FPGA-friendly SecurePrimitive", *Journal of Cryptology*, vol. 24, no. 2, pp.375-397, 2011.

[25] D. Merli, F. Stumpf, C. Eckert, "Improving the Quality of Ring Oscillator PUFs on FPGAs", In *Workshop on Embedded Systems Security* , 2010.

[26] S. S. Mansouri and E. Dubrova, "Ring Oscillator Physical Unclonable Function with Multi Level Supply Voltages ", in *International Conference on Computer Design*, pp.520-521, 2012.

[27] D. Hely, M. Augauneur, Y. Clauzel, J. Dubeuf, "A Physical Unclonable Function Based on Setup Time Violation", in *International Conference on Computer Design*, pp.135-138, 2012.

[28] D. Suzuki and K. Shimizu, "The Glitch PUF: A New Delay-PUF Architecture Exploiting Glitch Shapes", in *Cyptographic Hardware and Embedded Systems*, pp. 366-382, 2010.

[29] J. Guajardo, S. S. Kumar, G. -J. Schrijen, P. Tuyls, "FPGA Intrinsic PUFs and Their Use for IP Protection", in *International workshop on Cryptographic Hardware and Embedded Systems* , pp.63-80, 2007.

[30] D. E. Holcomb, W. P. Burleson, K. Fu, "Initial SRAM State as a Fingerprint and Source of True Random Numbers for RFID Tags", in *Conference on RFID Security*, pp.11-13, 2007.

[31] D. E. Holcomb, W. P. Burleson, K. Fu, "Power-Up SRAM State as an Identifying Fingerprint and Source of True Random Numbers", *IEEE Transactions on Computers*, pp.1198-1210, 2009.

[32] S. Kumar, J. Guajardo, R. Maes, G. -J. Schrijen, "Extended Abstract: The Butterfly PUF Protecting IP on every FPGA", in *International Workshop on Hardware-Oriented Security and Trust*, pp.67-70, 2008.

[33] R. Maes, P. Tuyls, I. Verbauwhede, "Intrinsic PUFs from Flip-flops on Reconfigurable Devices", in *Workshop on Information and System Security*, 2008.

[34] Y. Su, J. Holleman, B. Otis, "A 1.6pJ/bit 96% Stable Chip-ID Generating Circuit using Process Variations", in *International Conference on Solid-State Circuits*, pp.406-611, 2007.

[35] C. Bohm, M. Hofer, W. Pribyl, "A Microcontroller SRAM-PUF", in *International Conference on Network and System Security (NSS)*, pp. 269-273, 2011.

[36] G. Selimis, M. Konijnenburg, M. Ashouel, J. Huisken, "Evaluation of 90nm 6T-

SRAM as Physical Unclonable Function for Secure Key Generation in Wireless Sensor Nodes", in International Symposium on Circuits and Systems (ISCAS), pp.567-570, 2011

[37] M. Kassem, M. Mansour, A. Chehab, A. Kayssi, "A Sub-threshold SRAM Based PUF", in *International Conference on Energy Aware Computing*, pp.1-4, 2010.

[38] R. Helinski, D. Acharyya, J. Plusquellic, "A Physical Unclonable Function Defined Using Power Distribution System Equivalent Resistance Variations", in *Conference on Design Automation*, pp.676-681, 2009.

[39] A. Sreedhar and S. Kundu, "Physically Unclonable Functions for Embeded Security based on Lithographic Variation", in *Europe Conference & Exhibition on Design, Automation & Test* , pp.1-6, 2011.

[40] X. Wang and M. Tehranipoor, "Novel Physical Unclonable Function with Process and Environmental Variations", in *Conference & Exhibition on Design, Automation & Test* , pp.1065-1070, 2010.

[41] S. Meguerdichian and M. Potkonjak, "Device Aging-Based Physically Unclonable Function", in *Conference on Design Automation*, pp.288-289, 2011.

[42] Q. Chen, G. Csaba, P. Lugli, U. Schlichtmann, U. Ruhrmair, "The Bistable Ring PUF: A New Architecture for Strong Physical Unclonable Functions", in *International Symposium on Hardware-Oriented Security and Trust (HOST)*, pp.134-141, 2011.

[43] D. Ganta, V. Vivekraja, K. Priya, L. Nazhandali, "A Highly Stable Leakage-Based Silicon Physical Unclonable Functions", in *International Conference on VLSI Design*, pp.135-140, 2011.

[44] P. Simons, E. van der Sluis, V. van der Leest, "Buskeeper PUFs, a Promising Alternative to D Flip-Flop PUFs", in *Symposium on Hardware-Oriented Security and Trust*, pp.7-12, 2012.

[45] J. Aarestad, P. Ortiz, D. Acharyya, J. Plusquellic, "HELP: A Hardware-Embedded Delay PUF", *Journal of Design and Test*, vol. 30, no. 2, pp.17-25, 2013.

[46] H.Yu, P. H. W. Leong, Q. Xu, "An FPGA Chip Identification Generator Using Configurable Ring Oscillator", in *International Conference on Field-Programmable Technology (FPT)*, pp.312-315, 2010.

[47] G. E. Suh, and S. Devadas, "Physical Unclonable Functions for Device Authentication and Secret Key Generation", in Conference on *Design Automation*, pp.9-14, 2007.

[48] T. Holotyak, S. Voloshynovskiy, O. Koval, F. Beekhof, "Fast Physical Object Identification Based on Unclonable Features and Soft Fingerprinting", in *International Conference on Acoustics, Speech and Signal Processing*, pp.1713-1716, 2011.

[49] J. Guajardo, S. S. Kumar, G. -J. Schrijen, P. Tuyls, "Physical Unclonable Functions and Public-Key Crypto for FPGA IP Protection", in *International Conference on Field Programmable Logic and Applications*, pp.189-195, 2007.

[50] U. Rührmair, C. Jaeger, M. Bator, M. Stutzmann, P. Lugli, G. Csaba, "Applications of High-Capacity Crossbar Memories in Cryptography", *Transactions on Nanotechnology*, vol. 10, no. 3, pp. 489-498, 2011.

[51] A. Maiti and P. Schaumont , "Improving the quality of a Physical Unclonable Function using configurable Ring Oscillators", in *Conference on Field Programmable Logic and Applications*, pp.703-707, 2009.

[52] K. B. Frikken, M. Blanton, M. J. Atallah, "Robust Authentication Using Physically Unclonable Functions", in *International Conference on Information Security*, pp.262-277, 2009.

[53] S. Devadas, E. Suh, S. Paral, R. Sowell, "Design and Implementation of PUF-Based "Unclonable" RFID ICs for Anti-Counterfeiting and Security Applications", in *International Conference on RFID*, pp.58-64, 2008.

[54] J. Guajardo, S. S. Kumar, G. -J Schrijen, P. Tuyls, "Brand and IP Protection with Physical Unclonable Functions", in *Symposium on Circuits and Systems, IEEE International*, pp. 3186-3189, 2008.

[55] P. Tuyls and L. Batina, "RFID-Tags for Anti-counterfeiting", Topics in Cryptology, vol. 3860, 2006.

[56] K. Fruhashi, M. Shiozaki, A. Fukushima, T. Murayama, "The Arbiter-PUF with High Uniqueness Utilizing Novel Arbiter Circuit with Delay-Time Measurement", in *International Symposium on Circuits and Systems*, pp.2325-2328, 2011.

[57] R. Kumar, V. C. Patil, S. Kundu, "Design of Unique and Reliable Physically Unclonable Functions Based on Current Starved Inverter Chain", in *Symposium on Computer Society*, pp.224-229, 2011.

[58] Z. Paral and S. Devadas, "Reliable and Efficient PUF-based Key generation Using Pattern Matching", in *Symposium on Hardware-Oriented Security and Trust*, pp.5-6, 2011.

[59] M. Bhargava, C. Cakir, Ken Mai, "Reliability Enhancement of Bi-stable PUFs in

65nm Bulk CMOS", in *Symposium on Hardware-Oriented Security and Trust*, pp.25-30, 2012.

[60] M. D. Yu and S. Devadas, "Recombination of Physical Unclonable Functions", in *Conference on GOMACTech*, 2010.

[61] H. Kang, Y. Hori, T. Katashita, A. Satoh, "Performance of Physical Unclonable Functions with Shift-Register-Based Post-processing", Journal of *Communications in Computer and Information Science*, vol. 339, pp.14-21, 2012.

[62] M. D. Yu and S. Devadas, "Secure and Robust Error Correction for Physical Unclonable Functions", *Journal of Design & Test of Computers*, vol. 27, no. 1, pp.48-65, 2010.

[63] Chi-En Yin and Gang Qu, A Regression-Based Entropy Distiller for RO PUFs, report, 2011

[64] Wikipedia, "Triple Modular Redundancy", http://en.wikipedia.org/wiki/Triple_modu

-lar_redundancy.

[65] J, Ju, R, Chakraborty, R. Chakraborty, R. Rad, "Bit String Analysis of Physical Unclonable Functions based on Resistance Variations in Metals and Transistors", in *Symposium on Hardware-Oriented Security and Trust*, pp.13-20, 2012.

[66] Guansheng Li, Y. M. Tousi, A. Hassibi, E. Afshari, "Delay-Line-Based Analog-to-Digital Converters", *Transaction on CASII*, vol. 56, no. 6, pp.464-468, 2009.

[67] J. Ju, R. Chakraborty, C. Lamech, J. Plusquellic, "Stability Analysis of a Physical Unclonable Function based on Metal Resistance Variations", in *Symposium on Hardware-Oriented Security and Trust*, accepted, 2013.