

2-8-2011

A Physical Unclonable Function derived from the power distribution system of an integrated circuit

Ryan Lee Helinski

Follow this and additional works at: https://digitalrepository.unm.edu/ece_etds

Recommended Citation

Helinski, Ryan Lee. "A Physical Unclonable Function derived from the power distribution system of an integrated circuit." (2011).
https://digitalrepository.unm.edu/ece_etds/115

This Dissertation is brought to you for free and open access by the Engineering ETDs at UNM Digital Repository. It has been accepted for inclusion in Electrical and Computer Engineering ETDs by an authorized administrator of UNM Digital Repository. For more information, please contact disc@unm.edu.

Ryan Lee Helinski

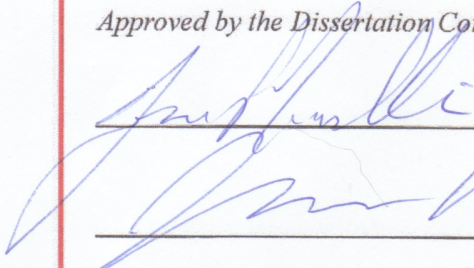
Candidate

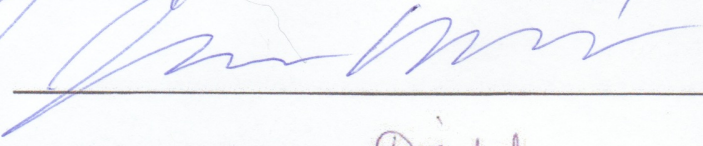
Electrical and Computer Engineering

Department

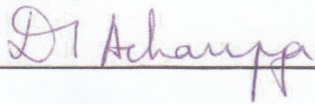
This dissertation is approved, and it is acceptable in quality and form for publication:

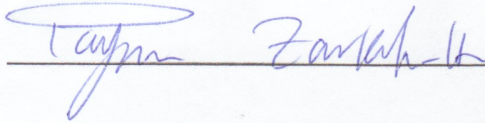
Approved by the Dissertation Committee:

 11/8/2010, Chairperson

 8 Nov 2010

Dr. Dhruva J. Acharyya

 11/8/2010

 11, 8, 2010

A Physical Unclonable Function Derived from the Power Distribution System of an Integrated Circuit

by

Ryan L. Helinski

B.S., Computer Engineering, Univ. of Maryland, Baltimore County, 2006

M.S., Computer Engineering, Univ. of Maryland, Baltimore County, 2008

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, 2010

©2010, Ryan L. Helinski

Dedication

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

Acknowledgments

I thank Dr. Jim Plusquellic for inviting me to join this research and for his support and contributions on my master's thesis and further on this dissertation. I would also like to thank Dr. Sani Nassif of IBM Austin Research Laboratory for his support of this research.

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

To my colleague Jim Aarestad and my sister Lauren who both helped to edit this document, thank you.

This research would not have been possible without R (<http://www.r-project.org/>), LabVIEW, Perl (<http://www.perl.org/>), GNU Octave (<http://www.gnu.org/software/octave/>), and Subversion (<http://subversion.tigris.org/>).

This document was typeset in L^AT_EX¹ using VIM². The L^AT_EX styles are based on the UMBC template by Eric Eaton³, and the UNM template by James Howse and Neall Doren⁴.

¹<http://www.latex-project.org/>

²<http://www.vim.org/>

³<http://www.umbc.edu/gradschool/etd/latex.html>

⁴<http://www.math.unm.edu/~nedoren/latex/>

A Physical Unclonable Function Derived from the Power Distribution System of an Integrated Circuit

by

Ryan L. Helinski

ABSTRACT OF 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, 2010

A Physical Unclonable Function Derived from the Power Distribution System of an Integrated Circuit

by

Ryan L. Helinski

B.S., Computer Engineering, Univ. of Maryland, Baltimore County, 2006

M.S., Computer Engineering, Univ. of Maryland, Baltimore County, 2008

Ph.D., Engineering, University of New Mexico, 2010

Abstract

Hardware support for security mechanisms such as authentication, cryptographic protocols, digital rights management and hardware metering depend heavily on the security of embedded secret keys. The current practice of embedding this key as digital data in the Integrated Circuit (IC) weakens security because the keys can be learned through attacks. Physical Unclonable Functions (PUFs) are a recently-proposed alternative to storing digital keys on the IC. A PUF leverages the inherent manufacturing variations of an IC to define a random function. However, poor performance under PUF quality criteria such as the level of randomness and reproducibility in the responses have detracted from their adoption and widespread use. In this dissertation, I propose several ways to define a novel PUF using the Power Distribution System (PDS) of an IC. First, I describe the hardware primitive and test setup that is required to obtain the PUF responses. Then, I evaluate the analog

PUF responses from silicon against standard PUF quality metrics in order to qualify the strengths and weaknesses of the proposed PUF. I show that the analog PUFs exhibit very high levels of randomness and reproducibility, but are sensitive to changes in temperature. Next, I propose extensions to our PUF that enable an exponential number of Challenge/Response Pairs (CRPs) with respect to the number of hardware resources, as well as yielding a marginal increase in the level of randomness. I also use these same analog measurements from silicon to simulate an integrated implementation of the PUF that takes a digital challenge and returns a digital response. I show that the integrated architecture also exhibits high levels of randomness and reproducibility, and is also resistant to changes in temperature. Future work includes designing and building a new IC that implements a more-powerful hardware primitive that will improve both the number and accuracy of the measurements, as well as additional hardware that will allow the challenge and response generation to be performed on-chip.

Contents

List of Figures	xiii
List of Tables	xix
List of Acronyms	xxi
1 Introduction	1
2 Background	6
2.1 PDS Variation Characterization	6
2.2 Physical Unclonable Functions	7
2.2.1 Applications	8
2.2.2 Techniques	9
2.3 PUF Metrics	11
3 PDS Characterization	14
3.1 PDS Architecture	15

Contents

3.2	Power Grid Equivalent Circuit Model	18
3.3	PDS Resistance Measurement Procedure	22
3.3.1	Branch Current Calculation	24
3.4	PDS Resistance Equations	26
3.4.1	Horizontal Resistance Analysis	26
3.5	Noise Analysis	30
3.6	Analysis of Power Grid Resistance Variations	33
3.6.1	Statistical Analysis	33
3.6.2	Alternative RMC Analysis	35
3.7	Conclusion	37
4	Using PDS Variations as a PUF	38
4.1	Analog PUF Definition	39
4.2	Experimental Results	44
5	Extension of the PUF and Evaluation of Metrics	49
5.1	Multiple-Shorting Scenarios	50
5.2	Single-Bit Probability Analysis	54
5.3	Collision Probability Analysis	56
5.4	Entropy Analysis	59
5.5	Discretized Signature Evaluation	64

Contents

6	Temperature Effects	67
6.1	Modifying the Experiment Setup	68
6.1.1	On-Chip Thermistor	70
6.1.2	GE Thermistor Characterization	74
6.1.3	Controlling On-Chip Temperature	77
6.2	Noise Analysis	83
6.3	Effects on Analog R_{eq} and V_{drop} PUFs	93
6.3.1	Temperature Effects on Analog PUFs	93
6.3.2	Probability Analysis	94
6.3.3	Vector Angles	102
6.4	Effects on Digital R_{eq} and V_{drop} PUFs	107
6.4.1	Function of Temperature	107
6.4.2	Aliasing Probability	111
6.4.3	Hamming Distances	113
6.4.4	Bit Probabilities	117
6.5	Observe Net Leakage Current	118
7	Future Work	123
7.1	Extensions to Hardware Primitive	123
7.2	Integrated Architecture	125
7.3	Integrated Architecture with an ADC	126

Contents

7.4 Short-term Goals	128
8 Conclusions	129
A Signature Tables	133
B Code Statistics	136
References	137

List of Figures

3.1	Power grid architecture	15
3.2	Instrumentation setup	16
3.3	Block diagram of the test structure (a) and details of the Resistance Measurement Circuit (RMC) (b)	17
3.4	1-port and 2-port power-up schemes to determine appropriate resistance model. Since the stimulus transistors can be modeled as a current source with a known voltage, the non-linearity and process variations inherent to the transistors do not affect the test.	19
3.5	Top and bottom voltage profile of 2-port simulation experiment of the test chip grid.	21
3.6	Top and bottom voltage profile of 6-port simulation experiment of a commercial grid	21
3.7	Complete model: 1 st test.	22
3.8	Complete model: 2 nd test.	22
3.9	Complete model: 3 rd test.	23
3.10	Power schemes investigated	27

List of Figures

3.11	R_{pv} and R_v results under different power-up configurations	28
3.12	R_h values from Eq.'s 3.12 and 3.13	29
3.13	R_h values from Eq.'s 3.9 and 3.10	29
3.14	Resistance network on the test chips.	31
3.15	Noise analysis of R_{pvs}	32
3.16	Noise analysis of R_vs	32
3.17	R_{pv} analysis for CS_1	34
3.18	R_{pv} analysis for CS_2	34
3.19	R_v analysis for CS_1	35
3.20	R_v analysis for CS_2	35
3.21	Alternative RMCs used in special experiments	35
3.22	Alternative RMC R_v analysis for CS_1	36
3.23	Alternative RMC R_v analysis for CS_2	36
3.24	R_h analysis for CS_1	37
3.25	R_h analysis for CS_2	37
4.1	Review of PUF circuit operation theory	39
4.2	(a) Connections of a modified version of the SMC and (b) details of the modified SMC.	41
4.3	On-chip instrumentation for signature generation.	43

List of Figures

4.4	(a) Voltage drop signatures for 12 chips and 12 control samples. (b) Equivalent resistance signatures for the same 12 chips and 12 control samples.	45
4.5	Illustration of the Euclidean distance D between two chips C_1 and C_2 and the uncertainty δ introduced by noise.	46
4.6	Histogram of chip (a) and noise (b) equivalent resistance distances and Gamma function fit.	47
5.1	Box plots of 1-on through 6-on (x -axis) ER values (y -axis) measured from 36 chips.	52
5.2	Box plots of 1-on through 6-on (x -axis) VDrop values (y -axis) measured from 36 chips.	53
5.3	Trends and fits of means and variances of ERs and VDrops for the multiple-on scenarios.	54
5.4	Scaled box plots of 1-on through 6-on (x -axis) ER values (y -axis) measured from 36 chips.	55
5.5	Histogram and Gaussian fit of standardized ERs from 192 responses and 36 chips.	56
5.6	Single-bit probability analysis of the ER PUF.	57
5.7	Gamma function fit of noise (left) and chip (right) ER Euclidean distance (ED) histograms.	58
5.8	ER ED cumulative PDFs of noise for groups 1-on through 6-on. . . .	60
5.9	ER ED cumulative PDFs of chips for groups 1-on through 6-on. . .	61
5.10	Collision probability using ER response vector sizes from 6 to 192. .	62

List of Figures

5.11	Pairing and analysis illustration.	62
5.12	Entropy analysis of VDrops and ERs.	64
6.1	Clamshell apparatus cross-sectional diagram	69
6.2	On-chip resistor for characterizing resistance	70
6.3	On-chip resistance versus temperature	71
6.4	Residuals for linear regression of on-chip resistance versus temperature	72
6.5	GE RL1007-624 thermistor characterization	75
6.6	Example transition from 52 to 53 degrees Celsius	78
6.7	Trends of leakage current versus temperature, y -axis is logarithmic .	79
6.8	LabVIEW front panel	82
6.9	Raw measurement noise: leakage voltage and current without and with active temperature control	84
6.10	Voltage and current measured under shorting condition without tem- perature control (result similar when temperature is controlled) . . .	89
6.11	V_{drop} and R_{eq} measurement noise with and without temperature control	90
6.12	Mean and standard deviations of various measurements versus tem- perature	91
6.13	$3\sigma/\mu$ relative noise floor for various measurements versus temperature	92
6.14	72-point mean R_{eq} versus temperature for the 192-values of the noise sample	95
6.15	Histograms of R_{eq} vector inter- and intra-chip Euclidean distances .	98

List of Figures

6.16	Histograms of V_{drop} vector inter- and intra-chip Euclidean distances	99
6.17	Histograms of aggregate (0 - 75°C) V_{drop} and R_{eq} vector inter- and intra-chip Euclidean distances	100
6.18	Histograms of nominal (25°C) V_{drop} and R_{eq} vector inter- and intra-chip vector angles	105
6.19	Histograms of aggregate (0 - 75°C) V_{drop} and R_{eq} vector inter- and intra-chip vector angles.	106
6.20	Histograms of Percent Differences ($100 \times (x - y)/x$) for <i>core</i> bits that flipped (left) and bits that were stable (right) for R_{eq} (top) and V_{drop} (bottom). Each plot includes the four temperature points between 0 and 75°C. The vertical lines indicate the corresponding measurement noise floor.	109
6.21	Histograms of Percent Differences ($100 \times (x - y)/x$) for <i>all</i> bits that flipped (left) and bits that were stable (right) for R_{eq} (top) and V_{drop} (bottom). Each plot includes the four temperature points between 0 and 75°C. The vertical lines indicate the corresponding measurement noise floor.	110
6.22	Histograms of Hamming distances for chip and noise samples for the V_{drop} and R_{eq} PUFs, using the “all” construction	115
6.23	Histograms of Hamming distances for chip and noise samples for the V_{drop} and R_{eq} PUFs, using the “core” construction	116
6.24	The observe transistor physical circuit view	119
6.25	Original and corrected equivalent resistance versus temperature . . .	122
7.1	SMC with multiple sense transistors for different metal layers	124

List of Figures

7.2	Extension of SMC primitive shown in Figure 7.1 to support two voltage sense wires.	127
-----	--	-----

List of Tables

3.1	Numerical analysis of 1-port and 2-port simulation and hardware experiments.	20
3.2	Numerical analysis of 6-port simulations of a low-resistance PDS. . .	22
4.1	Probability that the Euclidean distance between chips is less than 99.7% of all noise Euclidean distances.	48
5.1	Chip configurations and number of response	50
5.2	Collision Analysis	60
5.3	Discrete Signature Metrics	65
6.1	Fitted model parameters for leakage current versus temperature using the model $I_{leak} = e^{mT+b} = e^b e^{mT}$	80
6.2	Noise Levels without Temperature Control	88
6.3	Noise Levels with Temperature Control (25°C)	88
6.4	Review of previous estimates of probability. Note: the DAC2010 V_{drop} results were previously unpublished.	96

List of Tables

6.5	Results of probability analysis for various combinations of analog R_{eq} and V_{drop} and different temperature points.	97
6.6	Results of probability analysis for analog R_{eq} and V_{drop} , in the nominal and aggregate case, using vector angles	103
6.7	Results of probability analysis of digital PUF signatures for various combinations of R_{eq} and V_{drop} , core and all, and nominal temperature (25°C) and aggregate (worst-case) over 0°C and 75°C	112
6.8	Mean inter-chip and noise Hamming distances, reported in percent bits	114
6.9	Single-bit probabilities for binary signatures (ideally 50%)	118
6.10	Bit flip probabilities for binary signatures (ideally 0%)	118
A.1	Chip ER signature for chip 1	134
A.2	Chip ER signature for chip 2	135

List of Acronyms

PUF Physical Unclonable Function

IC Integrated Circuit

RMC Resistance Measurement Circuit

SMC Stimulus/Measurement Circuit

CRP Challenge/Response Pair

PDS Power Distribution System

HD Hamming Distance

ED Euclidean Distance

FEOL Front End-of-Line

BEOL Back End-of-Line

PP Power Port

GCSM Global Current Source Meter

LCA Local Current Ammeter

VSW Voltage-Sense Wire

List of Tables

PG Power Grid

KVL Kirchhoff's Voltage Law

KCL Kirchhoff's Current Law

PDF Probability Density Function

CDF Cumulative Distribution Function

ER Equivalent Resistance

VDrop Voltage Drop

NTC Negative Temperature Coefficient

PTC Positive Temperature Coefficient

TCR Temperature Coefficient of Resistance

TEC Thermo-Electric Cooler

SNR Signal to Noise Ratio

ADC Analog-to-Digital Converter

Chapter 1

Introduction

It is widely accepted that the level of systematic and random process-induced variations in devices and interconnects is increasing as technologies are aggressively scaled [1, 2], and the sources of lithographic and non-lithographic process variations continue to grow [3, 4, 5]. *Process variations* impact key electrical parameters, including threshold voltage, resistance and capacitance, and have a significant impact on power and delay. For advanced technologies, it becomes increasingly important to understand and track process variations in order to model the process and avoid delays in time-to-market. In particular, new *methods and test structures* are needed to reduce the manufacturing development and yield learning cycle times, and to support rapid product and process debug. One technique is to include so-called “process monitors” directly into production designs. I propose such a structure and show that it can be used to understand interconnect resistance variations.

Many hardware security and trust mechanisms depend on the availability of *secret keys* that serve as a unique identity of each Integrated Circuit (IC). These keys serve as the basis for many higher-level hardware security mechanisms such as identification, authentication, remote activation, hardware metering and/or encryption.

Chapter 1. Introduction

Conventionally, secret keys are stored using fuses, flash or EPROM on the chip immediately after the IC is manufactured. For all mechanisms except identification, it is critical that access to this key remains restricted to the hardware circuits on the chip (i.e., remains secret). Unfortunately, since the keys are non-volatile, they are subject to both invasive and non-invasive physical attacks by adversaries who may be able to extract the key and thereby defeat the security mechanisms on which the key is built. Also, once a digital key is known, it becomes possible to produce *clone chips* that have the same key, which is as simple as programming the compromised key into a new chip, in most cases.

Methods of utilizing the same process variations I mentioned earlier, which are undesirable in the context of product quality, are sought to provide new hardware primitives for applications to *hardware security*. A trend in the literature is to leverage existing methods of characterizing process variations and design new circuits to serve these needs. The vulnerability of embedded digital keys to attacks can be mitigated if the keys are derived from the inherent, statistically-random manufacturing variations of the IC instead of being stored in a ROM. Physical Unclonable Functions (PUFs) embody structures that are sensitive to these silicon process variations can be used to generate keys which are a function of the specific random process variations of the device [6]. The process variations in sub-micron technologies are extremely difficult to control, and therefore creating two ICs that have the same random function is extremely difficult. In other words, a PUF is easy to fabricate, but practically impossible to duplicate. This is referred to as the “unclonable” property of PUFs, and is the hardware analog of a mathematical one-way function [7]. Typically, a PUF consists of a complex arrayed structure, each part of the structure producing what I will call “physical property”. Whether it is the speed of a ring oscillator or the delay of a path, the physical properties are in turn used to produce a 1 or a 0, depending on the process variations specific to the device. Unlike the ROM methods that I mentioned earlier, PUF keys are also “volatile”, which means

Chapter 1. Introduction

that the key is not present without the circuit in a fully-functional state (i.e., intact and powered on). Physical intrusions that involve de-processing the IC are considered to alter the function that the PUF originally had, and are therefore destructive to the PUF. For these two reasons, the most-attractive way to attack a PUF is to study its Challenge/Response Pairs (CRPs) and create a system that responds the same way. This is known as the “spoofing attack”, where the attacker does not produce a counterfeit chip, but rather masquerades as a known chip over a network or other communication medium. The best defense against spoofing is to increase both the number and unpredictability of the CRPs. Since PUFs are “unclonable” and “volatile”, they have the potential to revolutionize next-generation security and trust infrastructures in ICs.

However, since real PUFs are not ideal, other properties such as randomness and reproducibility also need to be considered. *Randomness* relates to the uniqueness of the function between ICs and specifies the probability that the function will have the same mapping on different ICs. Randomness is a function of the number of properties, the size of the CRPs and the statistical independence of the responses. Ideally, each bit in each response is like a fair coin and is a 0 exactly half of the time and a 1 the other half. In practice, the bits tend to be a biased toward 0 or 1, and the responses are somewhat dependent. *Reproducibility* relates to the integrity of the function under different environmental variations. Ideally, the response is always the same, but in practice the bits can flip and this has to be taken into account. More quality metrics are explained later, in Section 2.3. PUFs can be classified by the type of components that affect their function; e.g., MOSFETs, metal wires, insulator dielectric, etc. Each of these components is affected by process variations, which detract from the reproducibility of the PUF. However, some components are more sensitive than others. For example, the effect of ambient temperature on FET saturation current is quadratic, RO frequency (or inversely, path delay) is linear, and leakage current is exponential.

Chapter 1. Introduction

Typically, a PUF circuit includes an *interface* for retrieving a unique set of response vectors $\{R_1, R_2, \dots, R_n\}$ from a variety of different challenge vectors $\{C_1, C_2, \dots, C_m\}$. This interface serves two functions: (1) the PUF functions more like a RAM rather than a register, and (2) the underlying physical properties that define the function can be hidden. More formally, a PUF can be defined as a function

$$R = f(C), \quad f : \mathcal{B}^n \rightarrow \mathcal{B}^m, \quad (1.1)$$

where the challenge C is n bits and the response R is m bits. Typically, $n = m$, and the number of hardware resources p is at least $n \times m$. Since there is a response for each challenge, there are 2^n responses, and therefore the number of responses is *exponential* to the number of hardware resources. The total number of response bits is therefore $m2^n$. Having an exponential number of challenges is ideal since cloning or spoofing then requires matching a large number of CRPs, which is intractable for large n .

A common method of using these physical properties to produce a bit is to pair two identical structures together and use the difference between them. I refer to this type of implementation as “differential”. It is effective at balancing the process variations in order to produce an approximately-equal number of 1’s and 0’s. Using a “differential” method also means that the resulting bits are resistant to environmental variations that have a common-mode effect on the properties. For example, if two ring oscillator frequencies are 43MHz and 45MHz at one temperature, but increase to 44MHz and 47MHz at another temperature, then the relative difference is preserved and hence is resistant to changes in temperature. I will revisit this concept later in the discussion of the integrated PUF architecture, which is differential.

In this dissertation, I present the theory of a new PUF that is defined using resistance variations in Power Distribution System (PDS) (or power grid) of an IC. In Chapter 3, I present techniques for measuring resistance variations in the PDS that is enabled by an embedded primitive and external instrumentation. Then in

Chapter 1. Introduction

Chapter 4, I show that these same resistances which are useful for understanding process variations can be used to define a PUF and I evaluate the potency of the PUF to distinguish one IC from one another using the metric of the probability of aliasing. In Chapter 5, I describe extensions of this PUF that increase the number of CRPs from linear to exponential and I apply several other quality metrics that characterize the performance and security of the PUF. However, in order for a PUF to be used as a low-level primitive in hardware security applications, it needs to be integrated on-chip. To solve this problem, Chapter 4 also presents an architecture that implements the PUF on-chip and has digital input and output. In Chapters 5 and 6, I show that this architecture is also a viable PUF implementation. In Chapter 6, the previous analysis is repeated with temperature control and the temperature sensitivity of the various PUF implementations is evaluated. In Chapter 7, I present plans for implementing the integrated architecture, as well as extensions to the hardware primitive that should make it both more powerful and more accurate. Chapter 8 reviews the theory and presents reflections on what I learned.

Chapter 2

Background

This chapter provides a discussion of previous work in the areas of studying process variation in Section 2.1, building hardware security with PUFs in Section 2.2 and quality metrics for PUFs in Section 2.3. This builds upon the overview and definitions set forth in Chapter 1.

2.1 PDS Variation Characterization

There is a wide spectrum of published works on measuring and analyzing process variations. The techniques proposed in [8, 9] make use of ring oscillators and other types of test structures to track variations in Front End-of-Line (FEOL) parameters or single wire/via variations in Back End-of-Line (BEOL) parameters. For example, the authors of [8] propose a logic characterization vehicle to investigate the yield and performance impact of process variations. The authors of [10] proposed digitally-configurable ring oscillators to measure the effects of process variations on performance. A framework for the statistical design of experiments to measure the variance in critical dimensions of gate poly-silicon is proposed in [11]. A test struc-

Chapter 2. Background

ture to measure cell-to-cell delay mismatch due to process variations is proposed in [12], and another for the statistical characterization of local device mismatches is proposed in [13]. The authors of [14] propose a test structure that enables the extraction of spatial- and layout-dependent variations in both transistor and interconnect structures.

The techniques proposed in [9, 15, 16, 17, 18] focus on the measurement and analysis of resistance variations, but again, the work is limited to isolated test structures (wires, vias, etc.). For example, the authors of [15] and [16] propose test structures for characterizing wire resistance mismatch. Resistance measurement and analysis techniques for line width and step variation are described in [17] and [18]. In [9], dishing and erosion in non-ideal copper Chemical-Mechanical Planarization (CMP) is described, and dummy feature insertion techniques are proposed to reduce its impact on resistance variations. To my knowledge, [19] was the first time a technique has been proposed for measuring resistance variations in the PDS.

In [19], we proposed a test infrastructure that supports measurement of the PDS resistance characteristics for tracking BEOL *process variations*. The method enables a fast, first-order analysis of metal resistivity, and facilitates the identification of process problems. Since it is designed as a minimal augmentation to an existing design, it also serves to enable the resistance characteristics of the PDS to be evaluated for validation purposes, and provides meaningful data in the context of *an actual circuit design*, as opposed to the use of isolated test structures. This method is explained in Chapter 3.

2.2 Physical Unclonable Functions

Although the topic is relatively new, there is also a broad spectrum of work on PUFs (sometimes called Physical Unknown Functions[20], or Physical Random Functions

[21]), that can be classified by their technique (Section 2.2.2) and its application (Section 2.2.1).

2.2.1 Applications

Many applications of PUFs have been proposed including IC identification [22], labeling RFID tags, addressing wireless sensor nodes, IC process quality control [23], providing unique keys for encryption [24], IP protection on FPGA's [25, 26], authentication via challenge-response protocols [21, 27], and remote service and feature activation [28].

The authors of [29, 21, 22, 30] explain that, if an IC was able to provide its own unique physical identifier, then this signature could be used in the same way that human fingerprints are. That is, the signatures of known (and authorized) ICs could be collected, and the IC could output its signature later in the field. This could be used for tracking purposes, Return Materials Authorization (RMA), and detection of hardware piracy (e.g., counterfeit ICs, over-manufacturing, etc.).

Authentication is a mechanism by which the IC is identified via a challenge-response protocol. The term authentication means that the identity of the response is also verified. As the authors of [21, 31, 27] explain, a chip ID alone is not sufficient for authentication, since the response is always the same and can be reproduced. Instead, a secret key is embedded that enables the IC to generate a unique response to a challenge, which is generated each time. That way, so long as the key is secret, the authentication mechanism is not vulnerable to spoofing.

The authors of [24, 32] propose that PUFs be used to integrate secret keys for the use in cryptography. The author of [20] explains that, just as algorithmic one-way functions are critical to cryptography in software, PUFs are useful to cryptography in hardware. Specifically, an ideal PUF has the property that it is easy to generate a

response, but difficult to predict one, and PUFs are therefore inherently asymmetric.

Intellectual Property (IP) protection in FPGAs is presented as a major problem in [33, 24, 26, 34]. FPGA IP are bit streams that describe large functional blocks (like a microprocessor) and are licensed to be used to build larger designs or are complete designs themselves. These bit streams are generally stored in SRAMs, which makes them vulnerable to copying since the SRAM can be read directly. Therefore, methods are sought to stop that bit stream from working on FPGAs other than those authorized. The authors of [24] propose new protocols for the IP protection problem on FPGAs, that are based on public-key cryptography, and exploit PUFs derived from SRAM start-up conditions.

The authors of [28, 35] describe remote activation schemes that enable IC designers to lock each IC either once or at every start-up and then to enable it remotely. In [28], their objectives were realized by adding a few states to the finite state machine (FSM) of a design and by adding control signals that are a function of the unique IDs. In effect, the hardware “locks up” waiting for an activation code specific to that IC. This enables the designer, who knows the unique internal control signals, to issue a unique activation code that unlocks only that IC. This mechanism offers protection against unauthorized use of Intellectual Property (IP) and hardware piracy (the illegal manufacturing of ICs).

2.2.2 Techniques

Various PUF *techniques* have been proposed, including mismatched delay-lines [36, 37, 21, 31, 27] and Ring Oscillators (ROs) [38, 34], exploiting inherent SRAM power-on patterns [26, 24], MOS device mismatch [29, 22, 38, 23, 34] and input-dependent leakage patterns [39].

The authors of [26] proposed that the start-up values of embedded SRAM mem-

Chapter 2. Background

ories be used to create a PUF, especially in FPGAs where the start-up values can be accessed. When first powered on, SRAM memories are in an unstable state. The mis-match of the transistors composing the SRAM cells dictate the final start-up value of a 0 or a 1. Hence, a PUF can be constructed by dedicating some of the intrinsic SRAM memories (they used 64 bits) to be read only.

The authors of [36, 37, 21, 27] propose using delay properties of ICs for identification. Delay-based approaches involve mismatch in both MOS devices *and* interconnects. In [36], the authors apply the relative mismatch of delay lines, which are dependent on random process parameters, to generating unique signatures. The authors of [37] propose *integrating* the delay fingerprint hardware into the functional design, which enables identification *and* Trojan circuit detection.

The authors of [22] propose that the relative current-driving capabilities of transistors, which are a function of random polysilicon crystal formations, be exploited. To utilize these random formations, they proposed that MOSFET device mismatches be detected by comparing their current against a reference current. The resulting ones and zeros form the fingerprint (PUF). The authors of [29] propose that the MOSFET threshold voltages, which are a function of the random placement of impurity dopant atoms, be exploited. Frequently, a ROM-like structure is used that allows the PUF to be compact and easily read.

In [40], we proposed two PUFs which are described in Chapter 4. One PUF utilizes the voltage drop across the power grid due to a current being drawn through the grid. We call this the voltage drop PUF and it is a function of both metal resistance and transistors. The second PUF utilizes the global current in order to measure the equivalent resistance of the power grid. We call this the equivalent resistance PUF and it is a function of only the power grid, which is a function of specific metal resistances. This is an attractive property because the resistance of the power grid is marginally affected by environmental variations. In fact, resistance is a

linear function of temperature, and temperature is the only environmental parameter that directly affects metal resistance. Moreover, the distributed nature of the power grid makes it more prone to both random and systematic process variation effects, thereby decreasing its collision probability with other chips. Another significant advantage of using the power grid as a PUF is that it is an existing, distributed resource in every design. Therefore, the overhead of the power grid PUF is limited to the challenge/response circuitry which is well below 1% of the chip area. To our knowledge, this was the first time that a PDS measurement architecture has been proposed as a PUF.

2.3 PUF Metrics

Many metrics for PUFs have been established to assess their quality and security [41]. The major categories of metrics are *predictability*, *reverse-engineering*, *collision* and *sensitivity*, and are defined as follows.

The first type of vulnerability is *predictability*. In the context of pseudo-random number generators, if the generator being used is known, and the last output is known, then an attacker can accurately compute subsequent outputs. A similar attack is considered in the context of PUFs whereby an attacker learns several CRPs and tries to model unknown responses. The susceptibility of the PUF to this type of attack is related to how unique the responses are from one another. Three metrics for qualifying this are the single bit probability $P(R_i = 0, 1)$, the conditional probability $P(R_i = 0, 1 | R_j = 0, 1)$ and the Hamming distances between responses. Following is a description of these three metrics.

The single bit probability metric is the probability of response bit i , under all possible challenges, being a zero or a one. In the degenerate case, the output bits are zeros and ones, but are invariant under different challenges. In the ideal case,

Chapter 2. Background

$P(R_i = 0) = P(R_i = 1) = \frac{1}{2}$ and each output bit is like a flip of a coin for a given IC and a given challenge. The conditional probability metric addresses the independence between response bits. In the degenerate case, all of the output bits are the exactly the same (a zero or a one), but they can still all change under different challenges. In the ideal case, the result of flipping any bit in the challenge is a random set of half of the bits flipping in the response. As shown in [41], the effect of different challenge bits is not always the same; some challenge bits can affect the output more than others, and this is an artifact of the architecture. The Hamming distance between responses is another, more systematic method of addressing the same issue as the conditional probability, and is more amenable to computational analysis.

The second type of vulnerability is susceptibility to *reverse-engineering*. This vulnerability is a matter of how accurately the PUF can be modelled given a set of CRPs. In the ideal case, each m -bit response is independent from the rest. In that case, knowing all but one response does not increase the accuracy of guessing that last response; it still contains new information. For example, a plot of the modeling accuracy versus the number of CRPs tends to start at near zero (a guess) and then approach 100% with an exponential decay. To resist this weakness, the PUF circuit should obfuscate the physical properties of the system well so that the responses are highly “non-linear” or uncorrelated to the challenge. An example PUF interface that is highly “linear” would be simply XORing the n -bit input with n physical properties to produce the output. This would score very poorly against this metric. As mentioned earlier, other forms of reverse-engineering are considered destructive and therefore the original function is destroyed during the attack.

The third metric for PUFs is collision vulnerability. The previous metrics have only considered the set of responses from a single IC. Collision vulnerability considers how differentiable or distinguishable ICs are to one another. This is the essence of the power of the PUF to separate ICs, and is easily quantified by a collision

Chapter 2. Background

probability—there is one collision in every x ICs. In the ideal case, each IC is as different as possible from one another; in this case, we use the upper bound on the number of m -bit binary strings and we say that a PUF is resistant to collision attacks if the probability of collision approaches one in every 2^m ICs. In the degenerate case, each IC has 2^n possibly-distinct responses, but those responses are identical from IC to IC. Realities that detract from are systematic similarities that manifest in the layout-level of the PUF circuitry. These effects cause each IC to tend to have a similar response, detracting from the PUFs utility.

The last metric used to evaluate PUFs is the *sensitivity of the PUF to environmental variations*. Ideally, the response of a PUF to a given challenge is invariant over time and under different operating conditions such as temperature and supply voltage level. Unfortunately, these environmental variations tend to affect the PUF and cause the response to change. Depending on the components involved, some implementations are more sensitive than others. For example, the effect of ambient temperature on FET saturation current is quadratic, RO frequency (or, inversely, path delay) is linear, and leakage current is exponential.

A common approach to overcoming environmental variations in the physical quantities being measured is to pair them together so that they vary in the same way, and their *differential* persists even under different conditions. Another way to solve this problem is to use error-correcting codes. A designer would choose the complexity of the PUF (the number of physical properties) in order to achieve the desired level of collision, and then add an extra, say 15% in order to account for environmental variations. Then, in practice, a response would be recorded and then later the IC would give the same response, and one could use the Hamming distance between those responses to see if they are within some distance from one another.

Chapter 3

PDS Characterization

This chapter is organized into several sections, following the structure of our paper [19]. First, the architecture of the power grid, the on-chip support circuitry and the experiment setup are described in Section 3.1. Next, a model for the power grid is developed and validated with experimental data in Section 3.2. The experimental procedure for conducting PDS resistance measurements is given in Section 3.3. Finally, the simultaneous equations that need to be solved to obtain the resistance components of the PDS are described in Section 3.4. Finally, the resolution limits of our measurements are discussed in Section 3.5.

The results reported in this paper are derived from chips fabricated in a 65nm technology from IBM, and are therefore meaningful to state-of-the-art practices. However, the PDS measured in the hardware experiments was not designed to minimize ohmic IR and inductive $L\frac{di}{dt}$ voltage drops, and from this perspective it does not conform to a typical PDS of a commercial product. In particular, the resistances of many of the PDS components of our test chips are larger—some by more than an order of magnitude—than those found in commercial chips. In order to validate our technique for commercial applications, we supplement our test chip results with

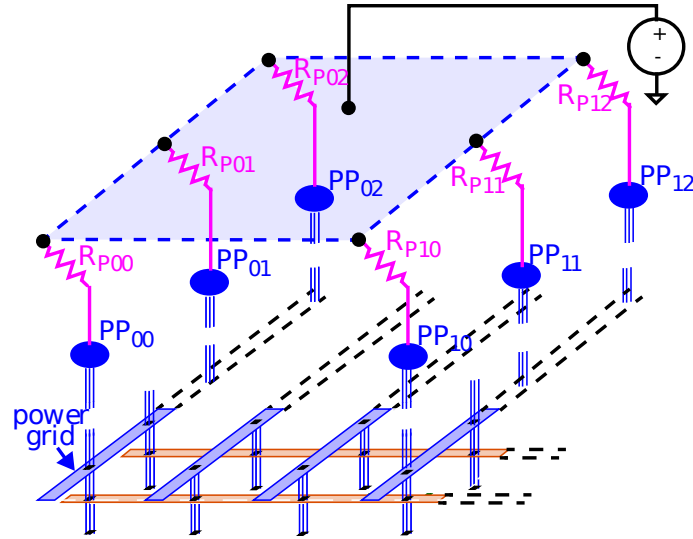


Figure 3.1: Power grid architecture

data from a simulation model that is representative of commercial designs.

3.1 PDS Architecture

A high-level representation of the power grid architecture used in the simulation and hardware experiments is shown in Figure 3.1. The bottom portion shows that adjacent metal layers are routed at right angles to each other in a mesh configuration with vias between the intersections. The ground (GND) grid, which is not shown, is interleaved with the power grid and routed in a similar fashion. Both grids are routed across the ten metal layers available in the 65 nm process. The width of the wires and the granularity of the mesh vary across the metal layers. In particular, the widths of the lower metal tracks are smaller and the granularity is finer than the widths and granularity of the metal wires in the upper layers. This feature of the power grid is typical of commercial designs [42].

The power grid is connected to a set of six C4s or PP in the top metal layer. The

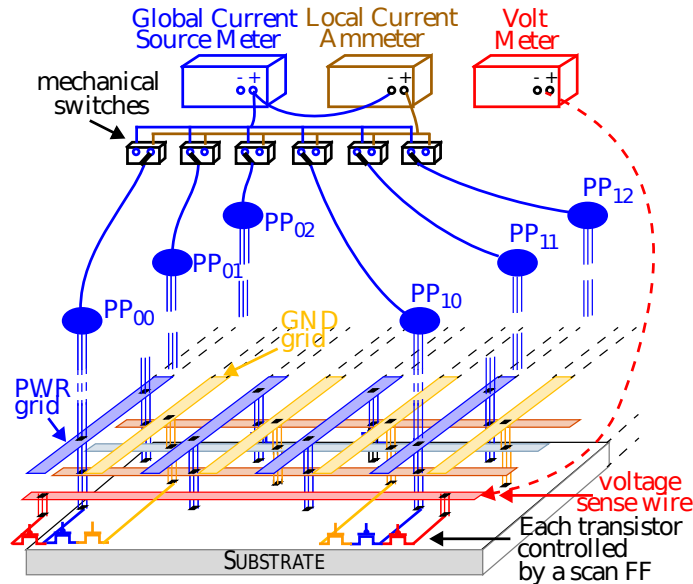


Figure 3.2: Instrumentation setup

Power Ports (PPs) are shown as ovals in the figure and are labeled PP₀₀ through PP₁₂. Commercial power grids can have hundreds of such PPs. The C4s enable the power grid to be connected to the power supply, either through a membrane-style probe card (during wafer probe) or through the package wiring. The finite resistances of PP connections are represented as series resistances $R_{P_{xy}}$ (where x, y are indices), in Figure 3.1. The measurement technique proposed in this work requires the measurement of branch currents through each of the PPs. For packaged chips, the PPs are *typically* wired into a power plane(s) within the package before being routed off-package through the power pins. Therefore, it is *not possible* to apply our technique directly to packaged parts without additional on-chip support circuits (beyond those described herein). We assume in the remainder of the paper that our technique is applied at wafer probe, where it is possible to access the PPs directly.

In our test setup, we emulate a wafer probe environment in our packaged chips by dedicating a separate package pin for each of the six PPs. The details of the test setup are shown in Figure 3.2. The package pins that are connected to the

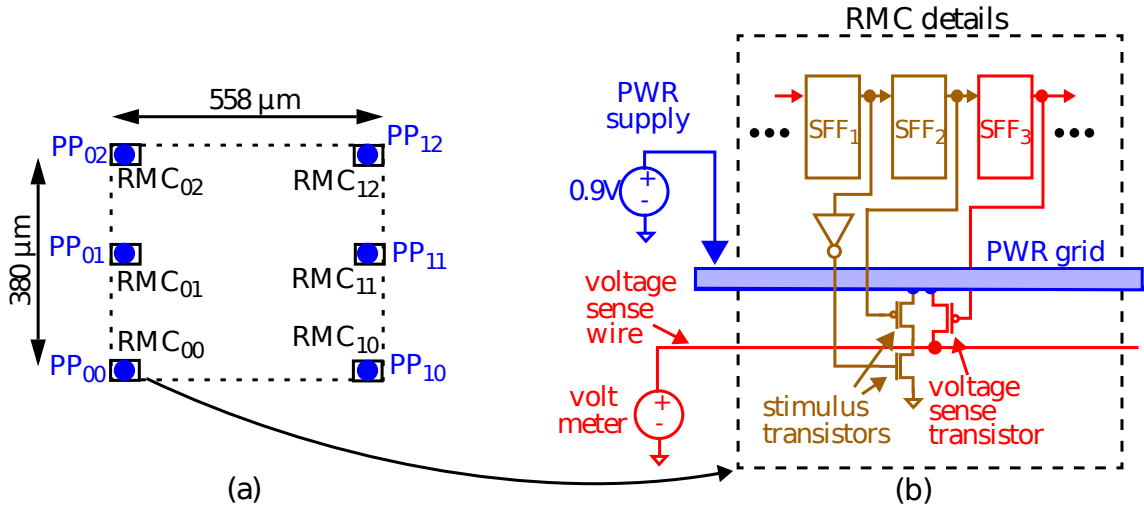


Figure 3.3: Block diagram of the test structure (a) and details of the RMC (b)

PPs are routed onto a PCB to a set of six mechanical, low-resistance switches. The switches can be configured in a left or right position. The left and right outputs of the switches each connect to a common wire that is routed to the Global Current Source Meter (GCSM) and Local Current Ammeter (LCA), respectively, as shown in the figure.

The GCSM provides 0.9 V to the PDS and can measure current with a precision of approximately 300 nA. The LCA is wired in series between the switches and the GCSM and allows measurement of the individual PP (local) currents at the same level of precision. For example, the switch configuration in Figure 3.2 allows measurement of the local PP₀₀ current I_{00} , as well as the global current.

In addition to branch currents, our technique to measure resistance also requires on-chip voltage measurements. The voltage is measured in our experiments using an additional (test-only) pin that is connected internally to a globally-routed Voltage-Sense Wire (VSW). A voltmeter is connected to this pin off-chip, as shown in Figure 3.2.

The last element of the test infrastructure is shown along the bottom of Figure 3.2 and in more detail in Figure 3.3. A RMC is inserted under each of the six C4s. The RMC consists of stimulus transistors, a voltage-sense transistor and a set of three scan flip-flops (SFFs). The outputs of the SFFs connect to the gates of the three transistors¹ as shown in Figure 3.3(b). The stimulus transistors provide a controlled stimulus—that is, a short between the power and ground grid—when the states of the SFF₁ and SFF₂ are set to 0. The voltage on the Metal 1 (bottom) layer of the power grid is measured using the voltage-sense transistor, which is enabled when a 0 is placed in SFF₃.

3.2 Power Grid Equivalent Circuit Model

The equivalent resistance models shown for the power grid in Figure 3.4(a), 3.4(b) and 3.4(c) were deduced from SPICE DC simulation data collected from a resistance model of the test chip’s power grid. The power grid resistances, given as R_x , R_y and R_z , represent the equivalent resistance of an entire mesh of resistors in the simulation model. The resistances R_{p1} and R_{p2} represent the external connection or probe resistances to the power grid.

The models shown in Figures 3.4(a) and 3.4(b) are referred to as 1-port experiments because only one PP is connected to the power supply—the others are left floating. Similarly, the configuration in 3.4(c) is called a 2-port experiment. The stimulus in each configuration is provided by RMC₁, which is depicted as a current source. The currents and voltage drops are labeled symbolically for each of the three experiments, e.g. I_1 and V_1 .

¹The stimulus as shown in our test structures was designed to serve other purposes beyond those described in this paper. A more efficient implementation would use only the p-channel transistor portion of the series transistor pair.

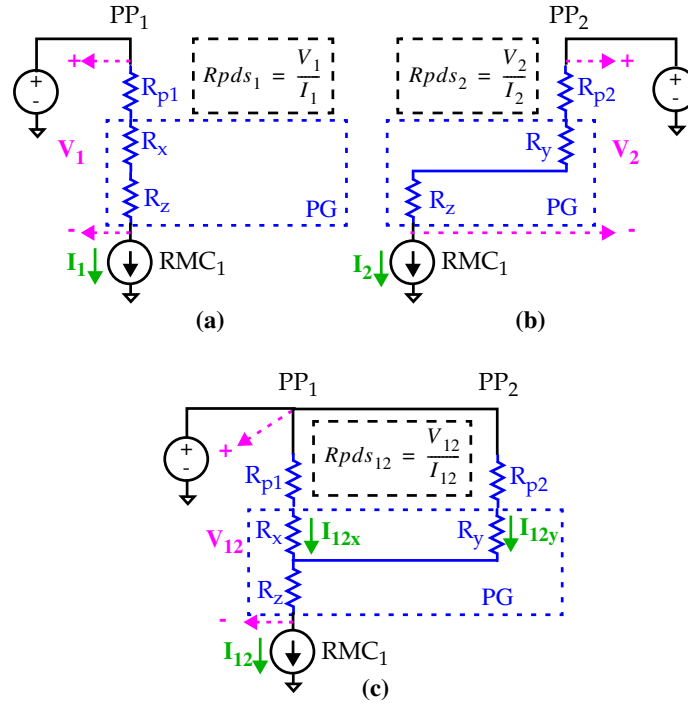


Figure 3.4: 1-port and 2-port power-up schemes to determine appropriate resistance model. Since the stimulus transistors can be modeled as a current source with a known voltage, the non-linearity and process variations inherent to the transistors do not affect the test.

Our objective is to verify that the equivalent resistance models of Figure 3.4 are valid representations of the actual PDS. Assuming they are valid, then

$$R_{pds,12} = \frac{1}{\frac{1}{R_{p1}+R_x} + \frac{1}{R_{p2}+R_y}} + R_z = \frac{1}{\frac{1}{R_{pds,1}-R_z} + \frac{1}{R_{pds,2}-R_z}} + R_z \quad (3.1)$$

expresses the relationship between the equivalent resistances in the three models $R_{pds,1}$, $R_{pds,2}$ and $R_{pds,12}$. Each of these is defined as V_i/I_i where V_i is the voltage drop and I_i is the total current, for $i = 1, 2$ or 12 (see dashed boxes in Figure 3.4). For example, the first element on the right side of the equation gives the parallel resistance of the upper network in Figure 3.4(c), expressed using the equivalent resistances in Figures 3.4(a) and 3.4(b). The second element on the right side of the equation accounts for the shared resistance R_z that is in series with the parallel

Chapter 3. PDS Characterization

network.

As mentioned earlier, we confirmed these models using a numerical analysis of data collected from a simulation model and from one of the 65 nm test chips. The values of the equivalent resistances that were computed are presented in Table 3.1, and compared with the values measured from hardware. Columns two, three and four give the equivalent resistances computed using data from the configurations shown in Figures 3.4(a), 3.4(b) and 3.4(c), respectively. The measured value of $R_{\text{pds},12}$ agrees with the value predicted by Eq. 3.1. The values of R_z in the fifth column are derived by solving Eq. 3.1 for R_z .

	$R_{\text{pds},1}$	$R_{\text{pds},2}$	$R_{\text{pds},12}$	R_z	R_x	$R_z/(R_x + R_z)$
Simulation	14.05 Ω	20.04 Ω	12.10 Ω	8.18 Ω	0.63 Ω	92.9%
Hardware	14.24 Ω	20.02 Ω	12.27 Ω	8.38 Ω	0.62 Ω	93.1%

Table 3.1: Numerical analysis of 1-port and 2-port simulation and hardware experiments.

The series resistance combinations $R_{\text{p1}} + R_x$ and $R_{\text{p2}} + R_y$ are represented by the terms in the denominator of Eq. 3.1, as indicated before, but the three tests as shown in Figure 3.4 are not sufficient to determine the individual values (e.g., R_{p1} and R_x). We were able to derive the individual values by creating a simulation model that closely approximates one of our test chips² The estimated values for R_{p1} and R_{p2} derived in this fashion are 5.24 Ω and 11.71 Ω , respectively.

The estimated values of R_x and R_y are easily obtained once R_{p1} and R_{p2} are known. The R_x values are given in the sixth column of Table 3.1 (the R_y values are similar). When compared with the R_z values in the fifth column, it is clear that R_x is smaller by more than an order of magnitude. Given that R_x and R_z are both grid equivalent resistances, this data indicates that the paths followed by the branch

²The actual values can be measured by adding voltage observe points in the PDS's top metal layer directly beneath the C4s, as discussed later.

Chapter 3. PDS Characterization

currents I_{12x} and I_{12y} from Figure 3.4(c) are common over a large fraction of the vertical resistance of the power grid. The last column in the table gives the fraction of common resistance at nearly 93%. The wire characteristics described for the power grid in Section 3.1 support this result. There, we disclosed that the resistance of the wires in the upper layers of the power grid is smaller than that of the wires in the lower layers.

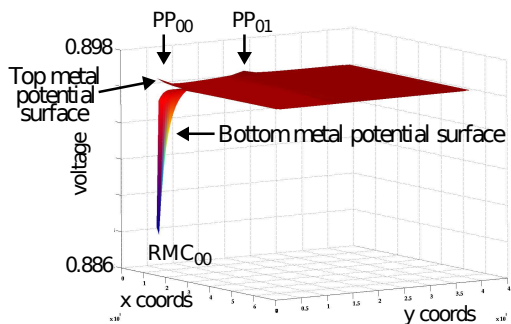


Figure 3.5: Top and bottom voltage profile of 2-port simulation experiment of the test chip grid.

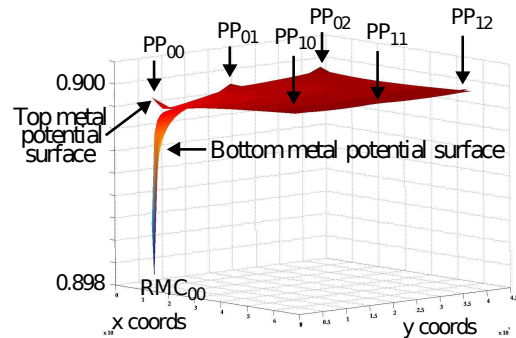


Figure 3.6: Top and bottom voltage profile of 6-port simulation experiment of a commercial grid

Our simulation model enabled a more detailed investigation of the spatial distribution of currents through the power grid. Figure 3.5 shows a 3-D voltage profile for the 2-port simulation model with RMC_{00} enabled (see Figure 3.4(a)). The voltage potential surfaces of both the top-most metal layer and bottom-most metal layer are superimposed. For most of the x - y dimension of the grid, the top and bottom surface potentials are nearly identical, indicating that current from remote PPs, e.g. PP_{01} , remains in the top portion of the grid until reaching the potential well near PP_{00} . At this point, the branch currents from other PPs combine and traverse the majority of the vertical dimension together. This type of current behavior will tend to amplify the magnitude of local IR drops.

We performed another simulation on our power grid with much smaller via and

wire resistance-per-square resistances to determine how the values in Table 3.1 would change for a PDS that better represents a commercial design. The R_p s were also reduced by a factor of twenty to model the contact resistance of a typical probe card. The voltage profile of this grid is shown in Figure 3.6 and its resistance characteristics are given in Table 3.2. $R_{pds,00}$ is the equivalent resistance measured with RMC_{00} enabled. It is a factor of eight times smaller than the value in the second column of Table 3.1. The lower resistances of the metal wires in this model are also reflected in columns three and four. However, the fraction in column five is still significant at 80.6%, and therefore, the lower resistance of this grid only partially explains the current distribution characteristics. We determined using other grid configurations that the most significant factor affecting this fraction is the overall architecture of the Power Grid (PG). For example, PGs configured such that each layer has the same resistance produce a fraction of 50%.

	$R_{pds,00}$	R_z	R_x	$R_z/(R_x + R_z)$
Simulation	1.74 Ω	1.47 Ω	0.35 Ω	80.6%

Table 3.2: Numerical analysis of 6-port simulations of a low-resistance PDS.

3.3 PDS Resistance Measurement Procedure

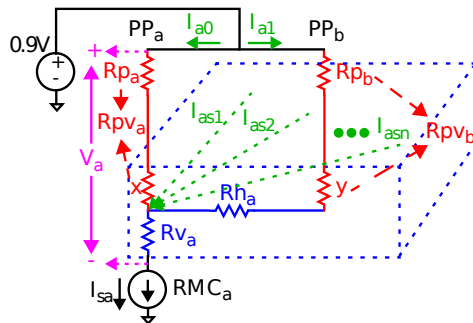


Figure 3.7: Complete model: 1st test.

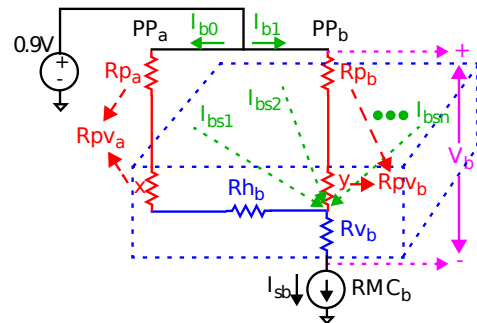


Figure 3.8: Complete model: 2nd test.

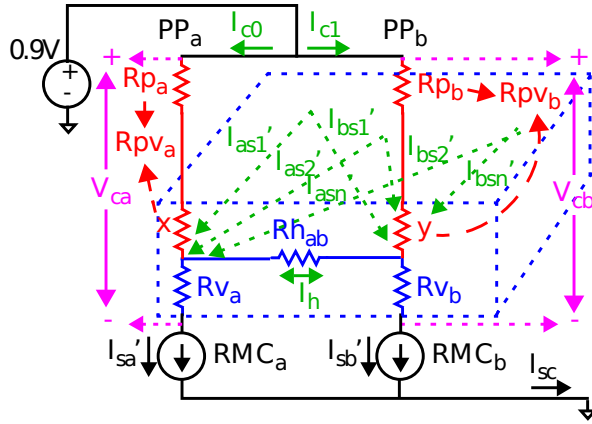


Figure 3.9: Complete model: 3rd test.

One of our goals is to define a set of tests that provide data to solve for six unknown resistances in the PDS. The three tests and corresponding equivalent circuit models are shown in Figures 3.7, 3.8 and 3.9. The six resistances, two of which are the sum of two series resistances, are labeled $R_{pv,a} = R_{p,a} + x$, $R_{v,a}$, $R_{pv,b} = R_{p,b} + y$, $R_{v,b}$, $R_{h,a}$ and $R_{h,b}$, where 'p' indicates *probe*, 'v' denotes *vertical* and 'h' denotes *horizontal*. As noted in the previous section, it is not possible to separate the series resistances, e.g. $R_{p,a} + x$, unless capability is added to the infrastructure to allow the voltage to be sensed at the point where the C4 attaches to the power grid.

According to the models, R_{ha} , R_{hb} and R_{hab} identify the same resistance and therefore represent only a single unknown. From simulation experiments, we find there are actually small differences in these resistances. The equations that we present later treat R_{ha} and R_{hb} in a special way *and* as separate variables. The values derived from our equations represent a good estimate of R_{ha} , R_{hb} and R_{hab} .

Each test provides two independent equations, enabling values to be derived for the six resistances from the solution to a system of simultaneous equations (to be described). The third test shown in Figure 3.9 requires enabling both RMC_a and RMC_b and measuring two voltages, V_{ca} and V_{cb} . Under the proposed infrastructure,

it is necessary to measure each of these sequentially by enabling the appropriate voltage sense transistor.

The current and voltages shown in Figures 3.7, 3.8 and 3.9 are calibrated to remove the impact of leakage currents. This is an important step to obtaining a meaningful result in modern technologies, given the trend of increasing background leakage currents. Calibration is carried out by measuring the currents and voltages, as given in Figures 3.7 and 3.8, and under a fourth configuration in which both RMC_a and RMC_b are disabled. These leakage currents are subtracted from the values measured under the three tests.

3.3.1 Branch Current Calculation

Unlike the 1-port and 2-port experiments shown earlier, the multi-port scheme introduces a set of additional currents, such as those labeled I_{as1} , I_{as2} through I_{asn} in Figure 3.7. These currents originate from the PPs distributed across the PG. The total current, e.g. I_{sa} in Figure 3.7, includes their contribution. Although it is straightforward to compute these supplementary currents, only the *total* current is needed in the equations given in the next section³.

The only currents that cannot be measured individually are the stimulus currents, I'_{sa} and I'_{sb} , shown in Figure 3.9. They are labeled using the prime symbol because they are related to the ‘unprimed’ values measured under the first and second tests. Under ideal conditions, the sum of current, $I_{sa} + I_{sb}$, measured under the first and second tests, is equivalent to $I'_{sa} + I'_{sb}$ (or I_{sc}). However, the p-channel stimulus transistors are not ideal current sources, and the small change in V_{DS} introduced by having both RMC_a and RMC_b enabled reduces their magnitudes.

³In our experiments, we compute the total current as the sum of the calibrated power port currents.

Chapter 3. PDS Characterization

In our experiments, the difference is small—at most a couple μAs —and can be derived using

$$\Delta I_s = I_{sa} + I_{sb} - I_{sc} \quad (3.2)$$

where the constituent currents (right-hand side of the equation) are the total currents measured under each of the three tests. From simulation experiments, we determined that the reduction in current given by ΔI_s splits nearly equally across both RMC_a and RMC_b in the third test. This holds under the condition that the resistance characteristics of the PDS as measured from either stimulus location are similar—a reasonable assumption given the uniform architecture of the power grid. We examined a variety of resistance configurations and found that the magnitudes of I'_{sa} and I'_{sb} are well approximated using

$$I'_{sa} = I_{sa} - \frac{\Delta I_s}{2} \quad (3.3)$$

$$I'_{sb} = I_{sb} - \frac{\Delta I_s}{2}. \quad (3.4)$$

The supplemental currents, e.g. I'_{as1} , as well as the current across R_{hab} , e.g. I_h , as shown in Figure 3.9, can also be derived but are not needed to solve the set of equations given in the next section.

3.4 PDS Resistance Equations

The first four equations are derived from the models shown in Figures 3.7, 3.8 and 3.9 using Kirchhoff's Voltage Law (KVL).

$$V_a = I_{a0} \cdot R_{pva} + I_{sa} \cdot R_{va} \quad (3.5)$$

$$V_{ca} = I_{c0} \cdot R_{pva} + I'_{sa} \cdot R_{va} \quad (3.6)$$

$$V_b = I_{b1} \cdot R_{pvb} + I_{sb} \cdot R_{vb} \quad (3.7)$$

$$V_{cb} = I_{c1} \cdot R_{pvb} + I'_{sb} \cdot R_{vb} \quad (3.8)$$

Equations 3.5 through 3.8 yield values for R_{pva} , R_{va} , R_{pvb} and R_{vb} directly if solved as a set of simultaneous equations.

3.4.1 Horizontal Resistance Analysis

The equations that we use to compute values for R_{ha} and R_{hb} , Equations 3.9 and 3.10, are *not* consistent with KVL applied to the models in Figures 3.7 and 3.8. In particular, R_{ha} is multiplied by the total current I_{sa} , in contrast to the model, which indicates the multiplier should be the branch current I_{a1} .

$$V_a = I_{a1} \cdot R_{pva} + I_{sa}(R_{ha} + R_{va}) \quad (3.9)$$

$$V_b = I_{b0} \cdot R_{pva} + I_{sb}(R_{hb} + R_{vb}) \quad (3.10)$$

$$R_h = R_{ha} + R_{hb} \quad (3.11)$$

Given that I_{a1} is strictly less than I_{sa} , the values obtained for R_{ha} and R_{hb} using Equations 3.9 and 3.10 underestimate the actual values. Interestingly, the sum of R_{ha} and R_{hb} using these equations produces a good estimate of their actual value, under the assumption that R_{ha} is nearly equal to R_{hb} , as we noted above, is reasonable. We use R_h to represent the sum as given by Equation 3.11

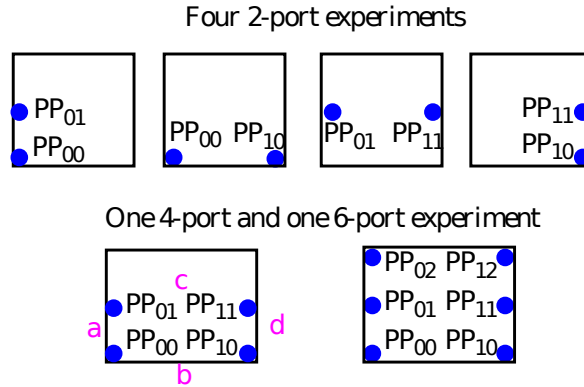


Figure 3.10: Power schemes investigated

To demonstrate that these equations provide a better estimate of the R_h resistances over those derived using KVL, which are

$$V_a = I_{a1}(R_{pvb} + R_{ha}) + I_{sa} \cdot R_{va} \quad (3.12)$$

$$V_b = I_{b0}(R_{pva} + R_{hb}) + I_{sb} \cdot R_{vb}, \quad (3.13)$$

we conducted a sequence of experiments using a variety of PP configurations. The criteria that we used to determine the best analytical form is based on the consistency of the results across the different PP configurations. Intuitively, the values computed for the six horizontal resistances should remain consistent, independent of the power-up scheme. However, this is not the case for R_{ha} and R_{hb} if Equations 3.12 and 3.13 are used.

We computed the values of the six resistances using hardware data from each of the PP configurations shown in Figure 3.10. The upper portion of the figure shows four 2-port experiments while the bottom portion shows a 4-port and a 6-port experiment. For each of the four 2-port experiments, the three tests described in Section 3.3 were applied using a pair of RMCs located underneath the labeled PPs. These twelve tests were also applied to the 4-port and 6-port configurations.

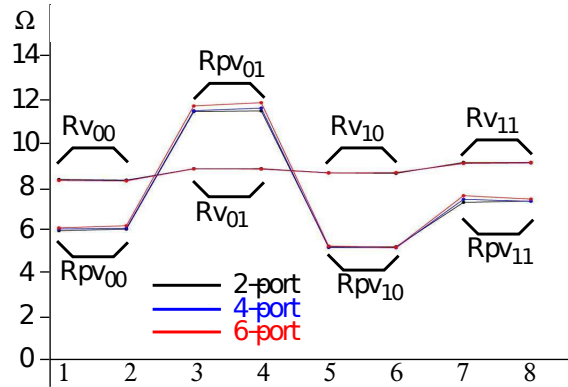


Figure 3.11: R_{pv} and R_v results under different power-up configurations

We first applied Equations 3.5 through 3.8 to derive values for each pair of R_{pv} and R_v under each of these configurations. For example, the resistances computed under the left-most 2-port configuration are R_{pv00} , R_{pv01} , R_{v00} and R_{v01} , labeled according to the PP coordinate space shown in Figure 3.10. The overlap of the PPs across the 2-port configurations allowed each of the four distinct R_{pv} and R_v pairs to be computed twice, yielding a total of eight values. The same held true for the 4-port and 6-port experiments. The results are shown in Figure 3.11 as a set of curves. The two values computed for each variable are adjacent in the curves to illustrate that they are similar, as expected. The three curves for the 2-port, 4-port and 6-port experiments are superimposed to illustrate that there exists strong agreement among the computed values, independent of the PP configuration scheme. We conclude that Equations 3.5 through 3.8 give the appropriate analytical form for these resistances.

We then carried out this analysis on R_{ha} and R_{hb} using Equations 3.12 and 3.13. The results are shown in Figure 3.12, but in a different format; the R_{ha} and R_{hb} values computed under each port configuration are offset in the x -dimension (not superimposed as in Figure 3.11, and they are labeled 2-port, 4-port and 6-port. The curves on the far right-hand side are simply the average of the two curves for each port configuration. The individual pairs of data points are labeled with letters ‘a’

through ‘d’, to associate them with the position given in the 4-port graphic shown in Figure 3.10.

The differences in the curves illustrate that Equations 3.12 and 3.13 are *not* of the appropriate form, particularly for the 4-port and 6-port configurations. We suspect that the supplementary currents, e.g. I_{as1} in Figure 3.7 and 3.8, are not properly represented by Equations 3.12 and 3.13. This is supported by the results obtained from the 2-port model, where the supplementary currents are zero. Here, the computed values for R_h are, in fact, good approximations of the actual values.

In contrast, the R_h values computed using Equations 3.9 and 3.10 across the various PP models are very similar, as shown in Figure 3.13. The curves are arranged in a similar fashion to those in Figure 3.12, except the computed values are scaled up by a factor of two, to better illustrate their variation around the ‘average R_h ’ values displayed in the curves on the far right. The similarity of the 4-port and 6-port curves to the 2-port curves suggests that Equations 3.9 and 3.10 are better able to represent the resistance characteristics of the PDS. A major portion of the difference that remains in these curves is due to the measurement noise, as is described in the next section.

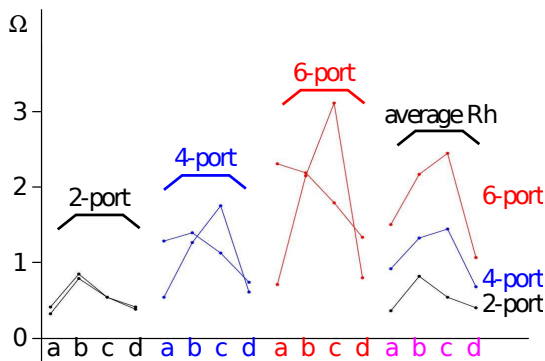


Figure 3.12: R_h values from Eq.'s 3.12 and 3.13

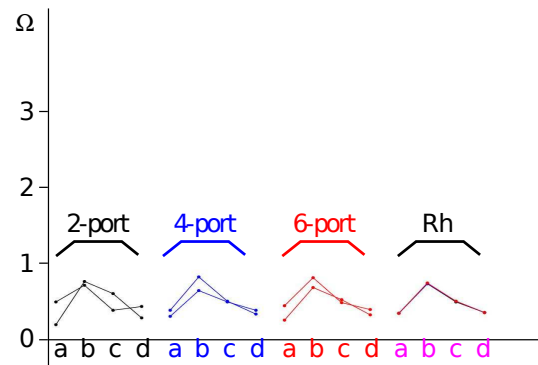


Figure 3.13: R_h values from Eq.'s 3.9 and 3.10

3.5 Noise Analysis

In this section, we describe the uncertainty associated with each measurement, how it can be overcome, and how it affects repeatability. The large differences in the magnitudes of the resistances of the various PDS components and supporting infrastructure make it imperative to evaluate the resolution limits of the method. For example, the RMCs use transistors as the stimulus with DC resistances up to approximately 1000 Ω s, and the resistances in the PDS of our chips vary over two orders of magnitude from a few hundred m Ω s to approximately 10 Ω s. In other words, does the method yield accurate results for both the large and small quantities, within a few repetitions of the procedure?

The limits are defined by the level of precision available in the instrumentation as well as the noise floor⁴. We used Keithley 2400 precision source meters to collect all the data. In our experiments, the noise floor is approximately 300 nA when the Keithley is configured as an ammeter, and approximately 500 nV when configured as a voltmeter. The range of currents varied from a few hundred μ As to a few mAs, yielding approximately five (5) digits of precision in the measurements. With the power supply voltage range set to 1.0 V, it was possible to get approximately 6.5 digits of voltage precision from the instrumentation. Given these measurement limits and resistance characteristics, the resistance resolution is estimated to be approximately 100 m Ω s.

This approach to calculating the resistance resolution, however, ignores the other detractors such as temperature effects—temperature fluctuations that occur while the data is collected. The most straightforward way of accounting for all sources of error is to repeat the data collection process on the same chip several times and then use statistics to characterize the resistance variations. We collected *twelve* sets

⁴Noise floor (n.): the sum of all the noise sources and unwanted signals within a measurement system.

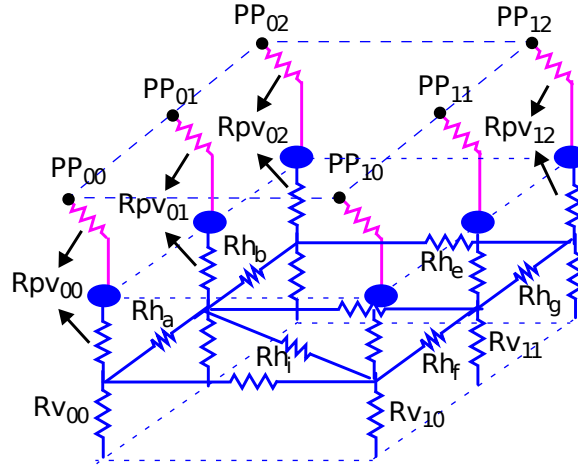


Figure 3.14: Resistance network on the test chips.

of data from *one* of the chips, and then computed the mean and standard deviation statistics on the resistance values derived from the equations.

The experiments were performed with all six PPs connected to the power supply as shown in Figure 3.14. The set of experiments consisted of applying the three-test procedure as described in Section 3.3 to eleven pairings of the power supply ports. Seven of the pairings involved adjacent orthogonally-positioned PPs. They include, in reference to Figure 3.14, PP₀₀-PP₀₁, PP₀₁-PP₀₂, PP₀₀-PP₁₀, PP₀₁-PP₁₁, PP₀₂-PP₁₂, PP₁₀-PP₁₁ and PP₁₁-PP₁₂. The remaining experiments involved diagonally-oriented PP pairings PP₀₀-PP₁₁, PP₀₁-PP₁₀, PP₀₁-PP₁₂ and PP₀₂-PP₁₁. For example, R_{pv00} , R_{pv02} , R_{pv10} and R_{pv12} are measured three times each, while R_{pv01} and R_{pv11} are measured five times each. The same is true for R_{pv} . Each of the eleven R_h values are computed only once, using Equation 3.11. The labels ‘a’ through ‘k’ are used to identify the R_h resistances (see Figure 3.14 for the labeling scheme).

A statistical plot illustrating the variations in R_{pv} is shown in Figure 3.15. The six groups of R_{pv} are distributed along the x -axis as a sequence of twenty-two vertical line plots. Each line plot contains twelve samples—one for each time the experiment

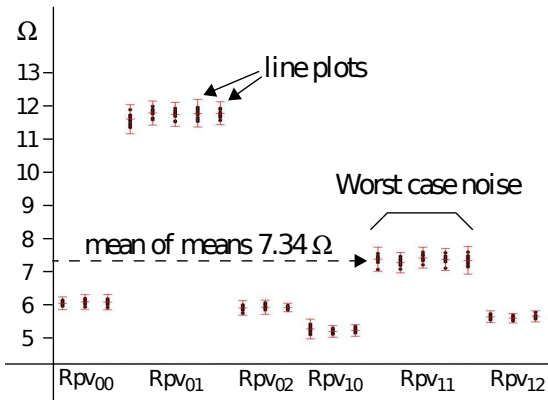


Figure 3.15: Noise analysis of R_{pv} s

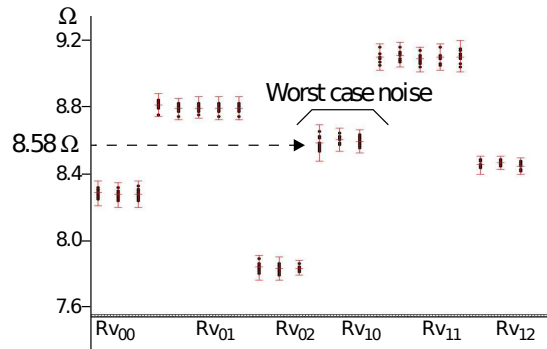


Figure 3.16: Noise analysis of R_v s

was repeated. The variation in the values is illustrated as a dispersion along the y -axis of the graph, and is quantified by the standard deviation σ . The mean and three- σ limits are displayed as horizontal lines within each line plot.

The variation among the line plots within each of the groups as well as the variation within each line plot itself reflect the measurement resolution. This is true because, ideally, all of these points should have the same value. The worst-case fractional error is given for R_{pv11} in which the largest 3σ limit is $420 \text{ m}\Omega$. The mean value is $7.34 \text{ }\Omega$, which yields a 6% error.

The noise floor is smaller for R_v and R_h . Figure 3.16 shows the mean and standard deviation for R_v where the worst-case deviation is $120 \text{ m}\Omega$ for R_{v10} . With a mean value of $8.58 \text{ }\Omega$, the fractional error in this case is 1.4%. Similar results were obtained for the R_h analysis.

As we indicated earlier, the magnitudes of the resistance elements in the PDS of our chips are larger than those of a commercial product. The smaller resistances in a commercial grid impact the resistance resolution analysis reported here. For example, the voltage drop with RMC_{00} enabled is approximately 7 mV in our chips. In contrast, Figure 3.6 presents simulation data for a model that better represents

a commercial chip, and shows the voltage drop is approximately 2 mV (3.5 times smaller). Since the precision of the voltmeter is unchanged, and the noise level is expected to be about the same, this suggests that our 6% maximum error could increase to 21% for a commercial grid. However, the reduction in voltage drop is compensated for—in part—by the increase in current resolution. For example, the fraction of the total current drawn in our chips from PP_{00} with RMC_{00} enabled is 24% and the fraction increases to 44% in the model of the commercial grid. This factor of 1.8 partially compensates for the loss in voltage resolution. Based on this analysis, we expect the worst-case error to be approximately 10% for a commercial grid⁵.

3.6 Analysis of Power Grid Resistance Variations

In this section, I describe how our measurement technique was applied to two sets of twelve chips and report the PDS resistances as modeled by Figure 3.14. The first chip set, denoted by CS_1 , was fabricated early in the development of the 65 nm process. The second set CS_2 was fabricated in the same process at a later time, after improvements were made. Our analysis demonstrates that the proposed methodology can be used to measure and identify the major sources of process variations in the BEOL process steps.

3.6.1 Statistical Analysis

An illustration of the dispersion of the various observations that were made of the various PDS components is shown in Figures 3.17 and 3.18, for chip sets CS_1 and CS_2 , respectively. Although the mean of the values are similar, the variance of each

⁵We expect the error level can be reduced to less than 5% if more sophisticated instrumentation and noise reduction techniques are employed.

Chapter 3. PDS Characterization

resistance is larger for CS₁ than it is for CS₂ in four of the six cases. For example, the variation in R_{pv01} for CS₁ is more than twice that for CS₂ and is well above the noise floor of 420 mΩ as shown in Figure 3.15. The extreme values in the line plots of this group suggest that resistance varies by almost 4 Ω. The reverse trend occurs for R_{pv02} and R_{pv12} , however—the variation is larger for CS₂ than for CS₁. These are the only instances where this occurred in the entire analysis, and the root cause is difficult to determine without intrusive physical inspection. One possible explanation is that the resistance variations in the package has changed, since R_{pv} includes an off-chip R_p component.

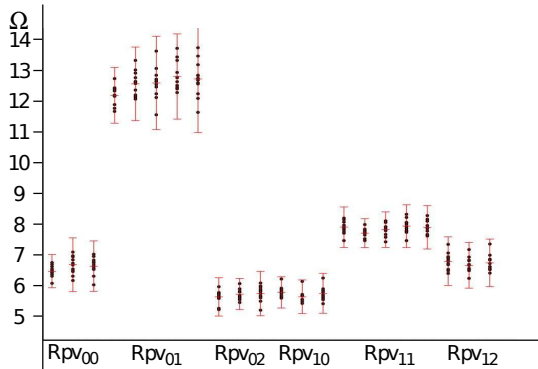


Figure 3.17: R_{pv} analysis for CS₁

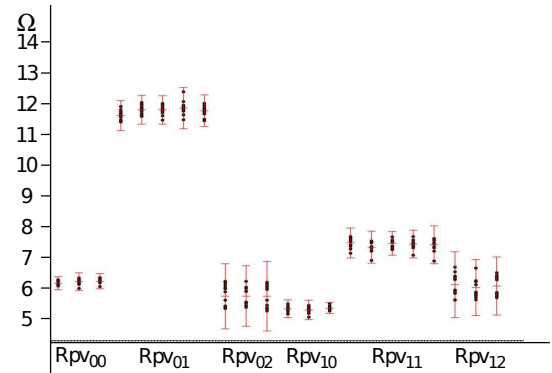


Figure 3.18: R_{pv} analysis for CS₂

The most significant differences in variation between the two sets of chips occur in R_v . The line plots in Figures 3.19 and 3.20 display the results in a 3-D format. The mean values of R_v for CS₁ vary from 9.0 to 12.0 Ω, while those for CS₂ vary from 7.8 to 8.0 Ω. The variance for R_v in CS₁ is nearly three times that of the R_v in CS₂. As noted before, the noise floor (three- σ limit thereof) is 120 mΩ, which is well below the inter-chip variations observed in both plots (Figures 3.19 and 3.20). The worst-case 3σ variance for R_v in CS₁ is 14.3 Ω, in contrast to 2.01 Ω for R_v in CS₂.

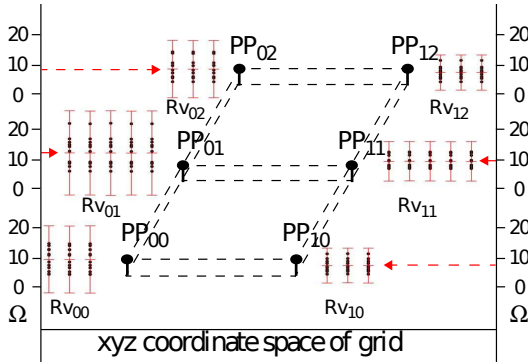


Figure 3.19: R_v analysis for CS_1

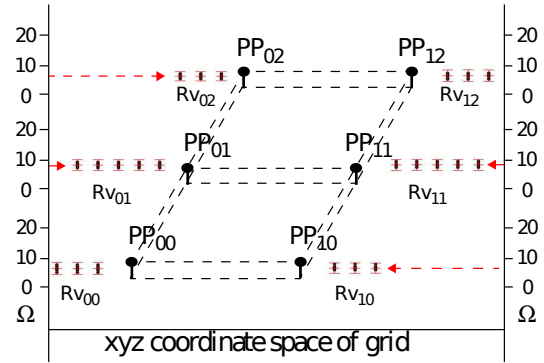


Figure 3.20: R_v analysis for CS_2

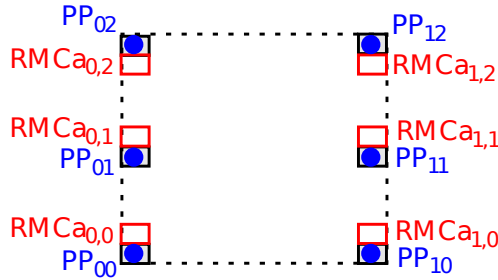


Figure 3.21: Alternative RMCs used in special experiments

3.6.2 Alternative RMC Analysis

In order to investigate the source of R_v variation further, we ran a special set of experiments involving a set of ‘alternative’ RMCs as shown in Figure 3.21, labeled as $RMCA_{x,y}$. These alternative RMCs are within $5 \mu\text{m}$ of the primary set of RMCs, which are shown as shaded boxes in Figure 3.21. The stimulus transistors in the original set of RMCs were used in these tests, however, the voltages were measured using the alternative $RMCA$ voltage sense transistors. The voltage profile shown in Figure 3.5 suggests that the resistances measured using the $RMCA$ reflect the characteristics of only the upper layers of the power grid⁶. In other words, the

⁶The infrastructure can be designed to enable all metal layers to be characterized in the fashion, by routing a set of voltage sense wires to each of the metal layers.

Chapter 3. PDS Characterization

RMCa provides an alternative measurement from the same place under the lower power grid layers. If that second measurement is not the same, then there must be a large amount of variation *in the lower layers*. If it is the same, then the inter-chip variation must be primarily *in the upper layers*. If the variation measured from these tests is smaller than that shown in Figures 3.19 and 3.20, then it can be inferred that the main source of variation is in the lower layers of the power grid.

This is indeed the case, as shown by the line plots in Figures 3.22 and 3.23, which illustrate that the magnitude of the variation is much smaller than that inferred from Figures 3.19 and 3.20. Note that the magnitude of variation in Figures 3.22 and 3.23 is in the 100's of $m\Omega$ range. In contrast to Figures 3.19 and 3.20, only a small increase in variation is observable in the upper layers of CS_1 over CS_2 . From this, we can infer that the main source of variation in R_v shown in Figure 3.19 is in the lower vias and wires.

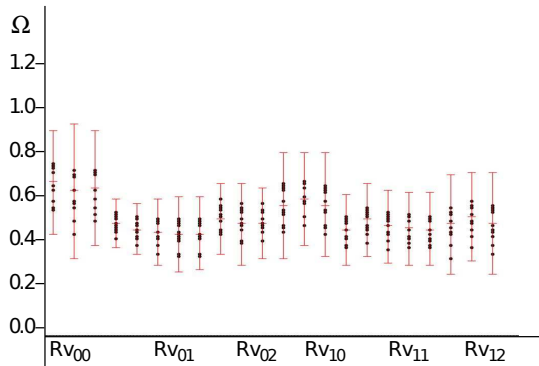


Figure 3.22: Alternative RMC R_v analysis for CS_1

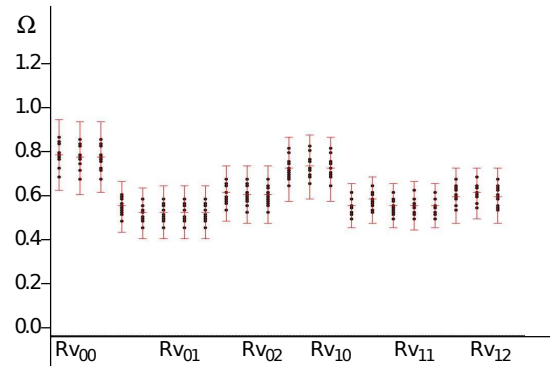


Figure 3.23: Alternative RMC R_v analysis for CS_2

The variation in R_h is given by the line plots in Figure 3.24 and 3.25 for CS_1 and CS_2 , respectively. The magnitude of the variation in CS_2 is only slightly smaller than that for CS_1 , which suggests the resistance per square remained fairly uniform in the two chip sets. Bear in mind that the R_h primarily reflects the characteristics of the top metal layers. Consequently, the lateral resistance of the lower metal layers

cannot be measured directly using this approach⁷.

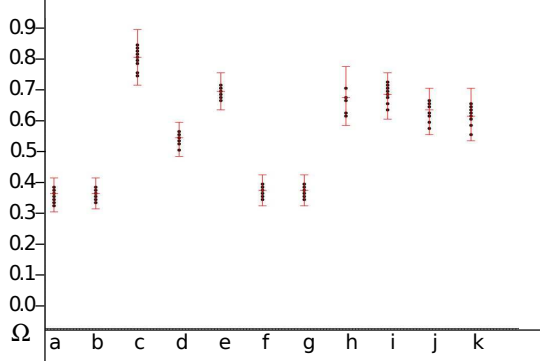


Figure 3.24: R_h analysis for CS_1

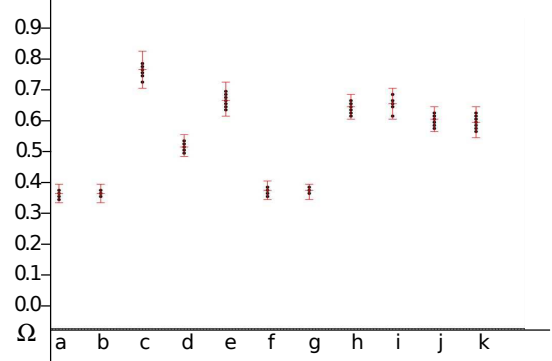


Figure 3.25: R_h analysis for CS_2

3.7 Conclusion

In this chapter, we presented a method to model and measure PDS resistances. This infrastructure can be used to characterize BEOL resistance variations during process bring-up and debug. It can also serve as a process monitor to track variations over time. The embedding of the infrastructure in the context of the actual circuit increases the relevance of the resistance analysis that it provides. The results of the analysis of resistance variations on two sets of chips fabricated in a 65nm technology illustrates that BEOL variations can be significant (Section 3.6). The analysis showed the proposed infrastructure can help reduce delays in manufacturing development and yield learning cycle times caused by BEOL resistance variations.

⁷Although the analysis given for R_v reflects variations in the lower metal layers, it also includes variations in the lower via resistances.

Chapter 4

Using PDS Variations as a PUF

In this chapter, I describe PUFs that are based on the measured voltage variations and equivalent resistance variations in the power distribution system (PDS) of an IC, using the infrastructure presented in Chapter 3. The effectiveness of the PUF is evaluated on thirty-six ICs fabricated in a 65 nm technology. This chapter is organized into two sections, following the structure of our paper [40]. In Section 4.1, I describe the way in which we collect and derive an identity for each IC and present an integrated architecture for the voltage drop PUF. In Section 4.2, I perform a quantitative analysis to predict the performance of these PUFs against signature aliasing.

In Chapter 3, we referred to the added hardware primitive as a Resistance Measurement Circuit (RMC), but in the context of this and the following chapters, we will refer to it as a Stimulus/Measurement Circuit (SMC).

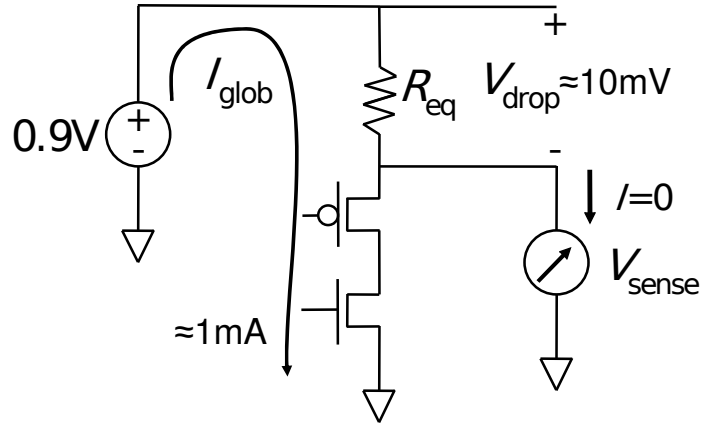


Figure 4.1: Review of PUF circuit operation theory

4.1 Analog PUF Definition

In [40], we proposed a PUF derived using two strategies: one that is based on voltage drops and one based on equivalent resistance. In either case, the signature associated with the chip is composed of six quantities, each corresponding to one of the six SMCs.

An equivalent circuit for the PUF with one of the SMCs switched on is given in Figure 4.1. To obtain the PUF response for both the Voltage Drop and Equivalent Resistance PUF definitions, we perform the following procedure. Clocks to the core logic are disabled to prevent transient events on the power supply system. The leakage current of the chip is measured with no SMC enabled and later subtracted from the measurements. The shorting inverter and the voltage sense transistor are both switched on. This creates a short on the bottom layer of the power and ground grids, and allows the voltage drop to be measured on the globally-routed voltage sense wire. The voltage drop $V_{drop} = V_{PWR} - V_{sense}$, which is the difference between sense voltage V_{sense} and the power supply voltage $V_{PWR} = 0.9\text{V}$. The shorting inverter draws approximately 1 mA (I_{glob}), which produces a voltage drop of approximately 10 mV at this point on the power grid. In the ER PUF, R_{EQ} is derived from voltage

division and is given by Equation 4.1.

$$R_{\text{EQ}} = \frac{(V_{\text{PWR}} - V_{\text{sense}})}{I_{\text{glob}}} \quad (4.1)$$

By enabling each of the SMCs individually, we obtain a set of six ER and VDrop responses, respectively, for an IC. We refer to all six quantities as the ER or VDrop signature for that IC.

The values in the voltage drop signature are affected by the magnitude of the current through the shorting inverter. The variations in the current magnitude among the shorting inverters actually adds to the ‘randomness’ of the PUF. However, the PUF is also more sensitive to environmental conditions, which detracts from its ability to generate the same signature (reproducibility). The Equivalent Resistance (ER) strategy eliminates this dependency by dividing voltage drops by the global currents. The elimination of the current dependency makes the ER-based PUF less sensitive to environmental variations.

Bear in mind that hundreds of SMCs can be inserted into commercial power grids, and there is no need to limit the number of SMCs to the number of power ports. Inserting many more SMCs would greatly expand the complexity of the signature over that shown in these proof-of-concept experiments. Doing so is practical because the overhead of the SMC is small, e.g., assuming a total of 100 SMCs, each with an area of $50 \mu\text{m}^2$ yields $5,000 \mu\text{m}^2$. This is only 0.02% of the $25,000,000 \mu\text{m}^2$.

The PUF as described has several drawbacks. First, it is only able to produce a single signature. i.e., n is linear to p . Second, signature generation requires the use of external instrumentation to measure the voltages and currents. Although this serves some applications, it poses problems for others that need to apply a challenge and obtain a response while operating in mission mode.

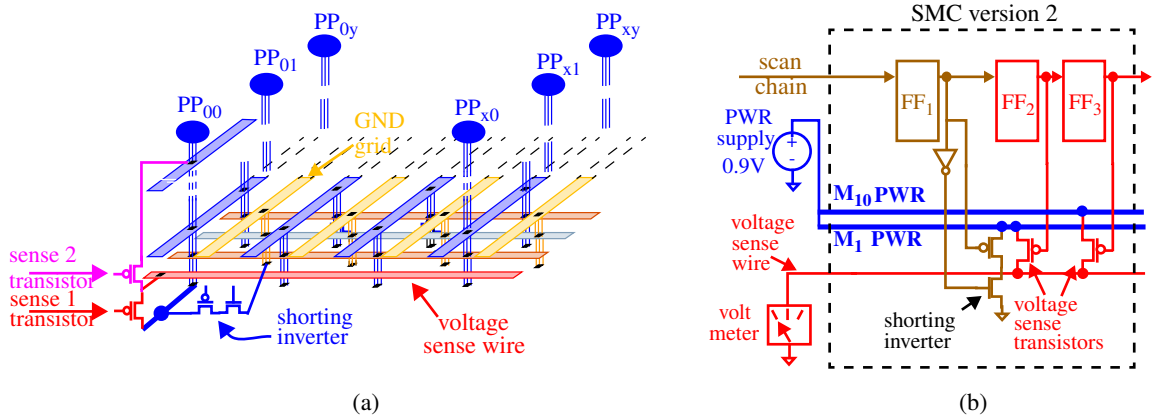


Figure 4.2: (a) Connections of a modified version of the SMC and (b) details of the modified SMC.

Simple modifications of the PUF can address these issues. For example, the SMC shown in Figure 3.3 can be modified to incorporate more than one ‘voltage sense’ transistor. Figure 4.2(a) shows a modification in which a second sense transistor, ‘sense 2 transistor’ is added to enable the voltage to be measured in metal 10 underneath the power port. With the second sense transistor, the *voltage drops* between M_1 and M_{10} can be measured at different places on the power grid (wherever there is an SMC). This increases the number of possible stimulus/response pairs of the PUF (w.r.t. the number of SMCs) from linear to quadratic because voltage drops can now be computed between any pairing of ‘sense 1’ and ‘sense 2’ transistors across the array of SMCs. Figure 4.2(a) shows a schematic in which an additional flip-flop, labeled FF_3 , is used to control the second sense transistor¹.

Another strategy to increase the number of challenge/response pairs is to allow the stimulus to be applied from more than one SMC. In this scenario, multiple shorting inverters are enabled simultaneously at different locations and the voltage drops

¹Although the ‘shorting inverter’ can be replaced with a single PFET, it is more robust to defects since it uses stacked devices and is proposed as a fault-tolerant strategy to prevent yield loss that would result if a defect caused the stimulus transistor to remain in the on state.

are measured using different combinations of sense 1 and sense 2 transistor pairs. We refer to these scenarios as multiple-on and the former as single-on. However, since the power grid is a linear system, superposition applies. Therefore, to make this more resilient to attack, whereby the attacker systematically deduces the voltage drops that would occur under a multiple-on scenario by combining the single-on measurements, this scheme can be combined with an obfuscation of the scan chain control bits. An alternative would be to disable the single-on scenarios by design. Under obfuscation, the number and position of the enabled shorting inverters are deterministically (or randomly) scrambled for a given scan chain control sequence, making it difficult or impossible to systematically apply single-on tests at known locations on the chip. We have investigated scan-chain obfuscation techniques in previous work where the objective was to prevent an adversary from using the scan chain to reverse engineer a design [43, 44]. These techniques are applicable here as well. For chip-specific random scrambling, a subset of the SMCs can be used during initialization to define the state of a *selector* that controls the scan chain scrambling configuration.

The PUF as proposed requires the use of external instrumentation to measure the voltages and global currents needed to compute the IC’s signature. Although this approach serves chip authentication well, e.g., where the objective is to periodically check the authenticity of a chip with counterfeits, it is not amenable to cryptography applications that use the signature as the secret key in hardware-implemented encryption/decryption algorithms. In order to serve this latter need, the signature generation process needs to occur using on-chip instrumentation.

The simplest approach to accomplishing this is shown in Figure 4.3. The *key generator control unit* drives the scan-in, scan-out and scan-clock signals of the SMCs with a specific pattern to enable one or more of the shorting inverters in the array

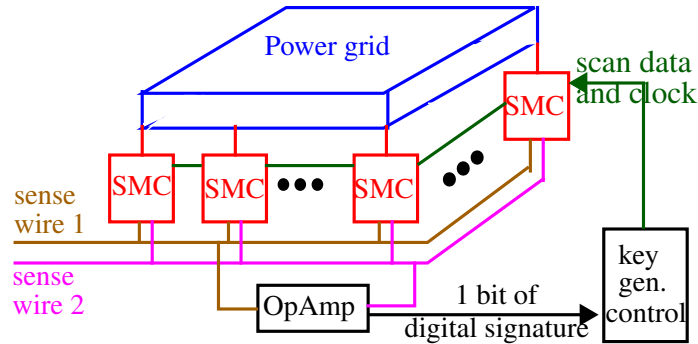


Figure 4.3: On-chip instrumentation for signature generation.

of SMCs². The scan pattern also enables two voltage sense transistors, one for each of the two voltage sense wires, labeled *sense wire 1* and *sense wire 2*. The two voltage sense wires are routed to the inputs of a simple differential Operational Amplifier (OpAmp). The OpAmp outputs a '0' or a '1' depending on whether the voltage on 'sense wire 1' is larger or smaller than 'sense wire 2', respectively. The 1-bit output is sent to the *response generation control unit* and the process is repeated until a sufficient number of bits are generated to realize the key. Note that this implementation is more sensitive to environmental variations because it makes use of voltages instead of equivalent resistances, as described earlier, and depends on the performance of the OpAmp to loyal comparison. Therefore, the response for a given chip under a given sequence of scan patterns may differ over time unless temperature and power supply noise are monitored and tightly controlled. However, this method may be relatively robust to environmental variations since it is comparing the analog voltage values, and is therefore a “differential” PUF. Other more noise-tolerant architectures are possible but they will increase the area overhead associated with the key generation infrastructure. The performance of this architecture is evaluated in Chapter 5.

²This scheme refers to the original SMC (Figure 3.3) modified to include a second sense transistor connected between M1 and a new voltage 'sense wire 2' (Figure 4.2).

4.2 Experimental Results

We carried out a set of experiments to evaluate the diversity in the voltage drops and equivalent resistances in a set of thirty-six chips. The chips are from two sets which were fabricated at different times. The first set (Chip Set 1) was fabricated earlier during technology development cycle and therefore have larger resistance variations. The second set (Chip Set 2) was fabricated after the process matured and better represent typical levels of process variations.

We also carried out an additional set of experiments to evaluate the stability of the PUF. These experiments were performed on one of the chips in the set. To evaluate stability, we repeated the signature generation/measurement process seventy-two times for two of the chips, one from each chip set³. The variation across the set of signatures from these experiments is due entirely to environmental noise and temperature variations. These experiments are important for determining the probability of signature aliasing, i.e., the probability that two chips from the population generate the same signature. We will refer later to data from these stability experiments as *control data*.

The experimental results for twelve of the chips from the set of thirty-six are shown in Figures 4.4a and 4.4b, using the voltage drops and equivalent resistances, respectively. The left half of the figure lists the chip number along the x -axis. The right half gives the PUF stability results for one of the chips. The six data points defining the chip signature are displayed vertically above the chip identifier. The y -axis gives the voltage drop and equivalent resistance, respectively, in each of the figures.

The diversity among the signatures within the twelve chips shown on the left side of the figures is evident in both plots. In addition to the different patterns of

³No temperature control or specialized low-noise test apparatus was used

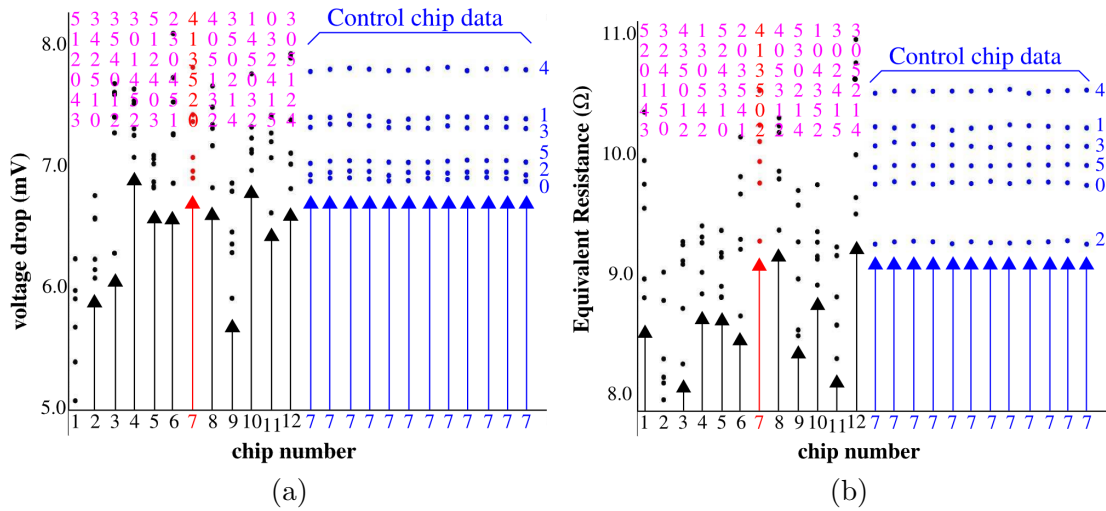


Figure 4.4: (a) Voltage drop signatures for 12 chips and 12 control samples. (b) Equivalent resistance signatures for the same 12 chips and 12 control samples.

dispersion in the signatures, the order of the data points from top to bottom is also distinct across all chips. The ordering is in reference to the SMCs that each data point corresponds to. For example, SMC_{00} in Figure 3.3 is assigned ‘0’, SMC_{01} is assigned ‘1’, ..., SMC_{12} is assigned 5. In Figure 6, the ordering for chip 1 is 5, 1, 2, 0, 4, 3, while the ordering for chip twelve is 3, 0, 5, 1, 2, 4. Therefore, the diversity among the signatures due to dispersion is actually larger than what is apparent because of the differences in the orderings. It is also clear from the PUF stability experiments that environmental variations have an impact on the signature and therefore, they must be taken into account.

In many cases, there are differences in the dispersion and ordering of the data points for the same chip across the voltage drop and equivalent resistance analyses. However, the voltage control points seem to have more uncertainty than the resistance control points. This is expected because the equivalent resistance eliminates an element of the diversity introduced by variations in the magnitude of the shorting currents.

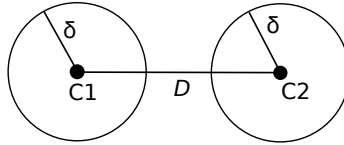


Figure 4.5: Illustration of the Euclidean distance D between two chips C_1 and C_2 and the uncertainty δ introduced by noise.

In order to quantify the dispersion among the chip signatures, we compute the Euclidean distance between the data points and analyze their variance. The six data points in each signature can be interpreted as a single point in six-dimensional space. The Euclidean distance between two signatures for chips x and y is given by Equation 4.2.

$$D = \|x - y\| = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_6 - y_6)^2} \quad (4.2)$$

Figure 4.5 illustrates our nomenclature of the response vectors of two chips, C_1 and C_2 , in n -dimensional space that is separated by a distance D . The Euclidean distance is computed between all possible pairings of chips, i.e. $(36 \times 35)/2 = 630$ combinations. Noise adds uncertainty in the exact distances between the chip response vectors, which is represented by a circle of radius δ . We define δ by computing the Euclidean distances between all possible pairings of the 72 vectors, i.e., $(72 \times 71)/2 = 2,556$ combinations, from the control data set. The distance δ is derived as *one-half* of the 3-sigma upper bound that characterizes the distribution of noise distances, i.e. it is defined as $1/2$ of the worst-case noise distance. With these definitions, we can define the probability of a collision. If a response vector from one chip is within 2δ of another chip's response vector, then the response vectors are considered identical and a collision occurs.

In order to compute the probability of two chips producing the same signature given the uncertainty associated with the measurements, we first compute a histogram that tabulates the number of Euclidean distances partitioned into a set of

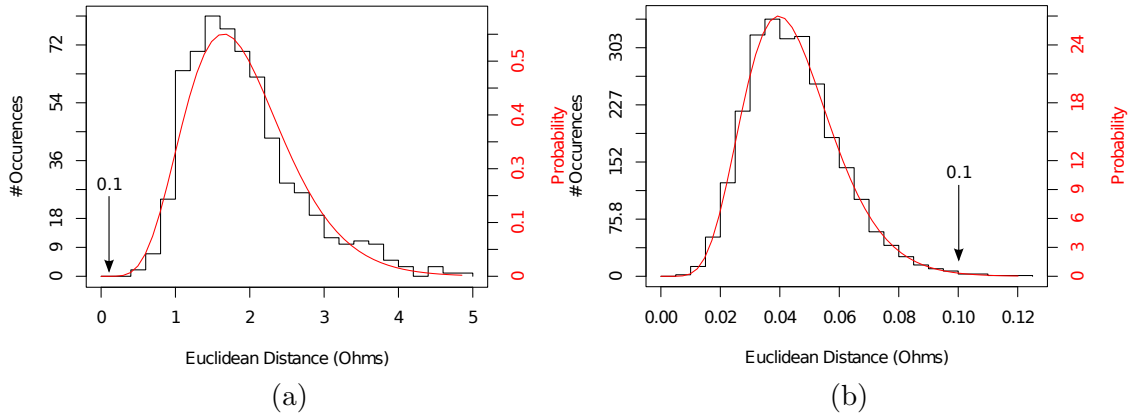


Figure 4.6: Histogram of chip (a) and noise (b) equivalent resistance distances and Gamma function fit.

bins for the chip and noise data sets, separately. The bins in each histogram are equal in width, with each equal to $1/25^{\text{th}}$ of the total span that defines the range of Euclidean distances among the 630 and 2556 combinations of chip and noise data pairings, respectively. We then fit these histograms to gamma probability density functions (PDFs). The histograms and the gamma PDFs are shown superimposed in Figures 4.6a and 4.6b, showing the chip and noise data, respectively, for the equivalent resistance analysis. In both cases, the gamma functions are a good fit to the histograms, based on evaluations of other types of PDF. The range of values found among the 630 chip pairings is between 0.45 and 5.0, as indicated by the x -axis, while the range for the noise analysis is between 0.01 and 0.12. Therefore, the largest value in the noise data is approximately four times smaller than the smallest value in the chip data.

We compute the probability of aliasing by first determining the Euclidean distance in the noise data that bounds 99.7% (3 sigma) of the area under the PDF. This particular Euclidean distance upper-bounds the worst case noise and is equal to 0.099 Ohms for the data shown in Figure 4.6. We then compute the cumulative distribution function (CDF) of the chip data and use this worst-case noise value (an

Chapter 4. Using PDS Variations as a PUF

	Volt.	Eq. Res.
P(alias)	3.5×10^{-11}	6.9×10^{-8}

Table 4.1: Probability that the Euclidean distance between chips is less than 99.7% of all noise Euclidean distances.

x value) to determine the probability of aliasing by looking up the y value on the chip CDF associated with this x value. This gives us an estimate of the probability that the Euclidean distance between any pairing of two chips is less than or equal to the worst-case Euclidean distance among the control data.

The results for the equivalent resistance and voltage analyses are given in Table 4.1. Using equivalent resistance, the probability of aliasing was found to be 6.9×10^{-8} or approximately 1 chance in 15 million. For the voltage analysis, the probability increases to approximately 1 chance in 28 billion, however this method may be more susceptible to environmental variations. Evaluating the sensitivity to temperature (for instance) is the subject of future research. Given that the number of SMCs used to define the signature in these experiments is only six, we can expect, based on these results, that the probability would improve in a commercial design that included a larger number of SMCs.

Chapter 5

Extension of the PUF and Evaluation of Metrics

In this chapter, I present extensions to the PUF that make better use of the hardware primitives available and an analysis of the PUF under many further quality metrics. The discussion is broken into several sections, following the structure of our paper in [45]. In Section 5.1, I discuss scenarios where more than one shorting-inverter can be enabled and how these scenarios can be used. In Section 5.2, I present a new method of creating PUF signatures that is “differential” using the same voltage data from the previous chapter. I assess the probability of the response bits being 0s or 1s using this differential PUF definition. In Section 5.3, I analyze the probability of aliasing as I did in the previous chapter with the addition of the new multiple-on scenarios. In Section 5.4, I perform a similar analysis on the differential PUF definition, where I quantify the statistical dependence of the multiple-on vectors using the metric of bit “entropy”.

Table 5.1: Chip configurations and number of response

# SMCs enabled	# Configurations	# Responses
1-on	6	6
2-on	15	30
3-on	20	60
4-on	15	60
5-on	6	30
6-on	1	6

5.1 Multiple-Shorting Scenarios

In this section, we introduce an extension to our PUF where additional challenges are introduced by allowing more than one of the shorting inverters to be enabled at a time. For example, if the shorting inverters from two SMCs are enabled simultaneously, then two responses can be obtained by measuring the VDrop at each SMC location separately. We refer to these configurations as x -on scenarios, to distinguish them from the 1-on scenario described in Chapter 4. A corresponding set of ERs can be computed by dividing each of the VDrops by I_{short} , the sum of the shorting currents from the set of enabled SMCs. With a total of six SMCs in our test chips, it is possible to obtain a total of 192 response bits by enabling different combinations of SMCs. For example, there are a total of 15 configurations in which two SMCs are enabled (2-on scenario), with each configuration generating 2 responses, for a total of 30 responses. For the 3-on scenario, there are 20 configurations and 60 response bits. The number of configurations and response bits for each x -on scenario for our test chips are tabulated in Table 5.1. The general closed-form expression for the number of possible response values for n SMCs is given by Equation 5.1.

$$\sum_{i=1}^n i \binom{n}{i} = n2^{n-1} \quad (5.1)$$

Figure 5.1 gives a box plot analysis of the ERs computed from our 36 chips split into 6 groups along the x -axis, one group for each x -on scenario (1-on, 2-on, etc.). The groups are labeled on the x -axis, with the label indicating the number of SMCs simultaneously enabled. The distribution is summarized by 5 values in each box plot; the medium, the upper and lower fence limits (for largest and smallest observations, resp.), upper and lower quartiles, and outliers (figure labels the 1-on case). The range and variation of the ERs decrease by a factor proportional to the number of enabled SMCs, n , because of the increasing magnitude of the accumulating stimulus currents. Figure 5.2 is the same box plot for the VDrop analysis. According to the figure, as the number of shorting inverters is increased, the effective voltage drop measured increases linearly, since additional stimulus current is drawn through the PDS.

By using the means and variances of the data in each column of these box plots, the following observations were made. The trend of the ER magnitude versus the number of stimulus currents n can be estimated by the exponential function $R = 16.2e^{-0.86n} + 2.5 \Omega$, with a variance that is highly non-linear for $n = 1, 2$ but follows $0.62e^{-0.70n} + 0.0023$ for $n > 2$. For the VDrop magnitude versus the number of stimulus currents, the trend can be estimated by an line, $V = (7.67E-4)n + 0.0065$ V, with a variance that follows $(2.35E-8)n + 2E-7$. These trends and fits are shown in Figure 5.3 Therefore, we can make the following generalizations. The magnitude of the ERs decreases exponentially and has a non-linearly-decreasing variation. The magnitude of the VDrops increase linearly and have a variation that is approximately constant.

Figure 5.4 shows the box-plots obtained after standardizing the data from Figure 5.1. Standardization makes it possible to compare the ERs from different groups and is accomplished using Equation 5.2, the standard Z -score equation. The μ_{group} term is the mean ER computed using all ER responses from the same group and the

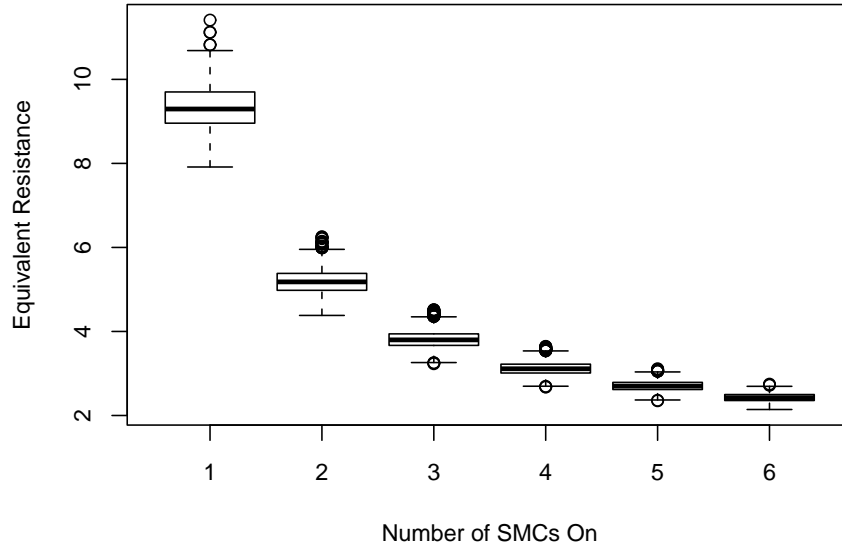


Figure 5.1: Box plots of 1-on through 6-on (x -axis) ER values (y -axis) measured from 36 chips.

σ_{group} value is the standard deviation for this group. Separate means and standard deviations are computed for each of the other x -on groups and used to create the Z -scores (as shown on the y -axis). Standardization effectively shifts and scales the data so that the mean is zero and the standard deviation is one.

$$ER_{\text{std}} = \left(\frac{ER_{\text{orig}} - \mu_{\text{group}}}{\sigma_{\text{group}}} \right) \quad (5.2)$$

Although standardization eliminates the effects of increasing global currents for the multiple-on scenarios, signal-to-noise will eventually limit how many SMCs can be enabled in practice. As more SMCs are enabled, the magnitude of the global current will continue to increase but the resolution of the instrumentation will remain fixed (in our setup, current resolution is approximately $1 \mu\text{A}$). Assuming the instrumentation provides five digits of resolution and each SMC introduces approximately 1 mA , this sets the limit to approximately 100 or fewer simultaneously-enabled SMCs. This

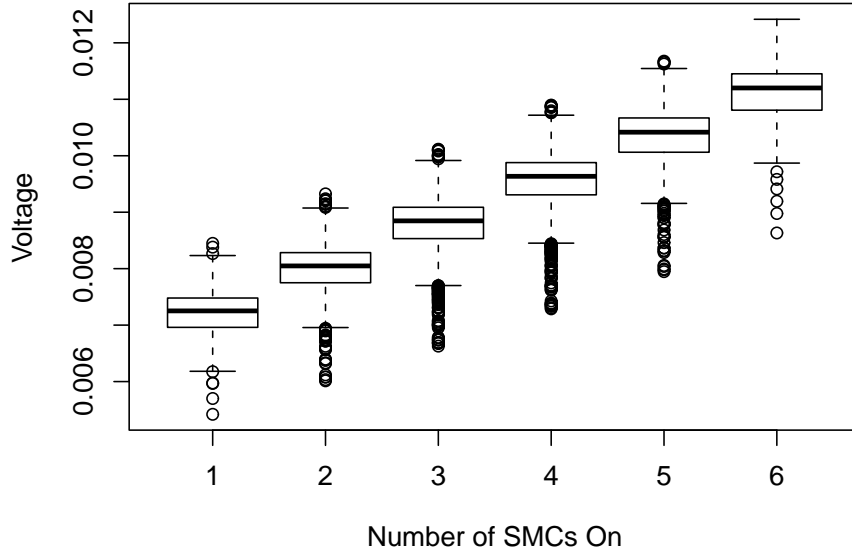


Figure 5.2: Box plots of 1-on through 6-on (x -axis) VDrop values (y -axis) measured from 36 chips.

limit restricts the number of ER responses available below that given by Equation 5.1 for scenarios in which more than 100 SMCs are embedded in an IC. We address this topic further in Section 5.4.

Standardization also allows an analysis of the entire data set. We indicated earlier that an important quality metric of a PUF is its degree of randomness. A first order measure of randomness can be obtained by constructing a histogram that bins the Z -score representation of the responses. The ideal distribution with respect to randomness is a uniform distribution. Non-uniform behavior, e.g. clustering, in the responses makes the PUF susceptible to certain attacks such as the prediction attack [41]. Figure 5.5 gives the histogram of all 192 ER Z -scores from the 36 chips. The distribution is best fit with a Gaussian PDF, shown superimposed on the histogram in the figure. Although not ideal, the symmetric nature of a Gaussian is desirable and more robust to attacks in comparison to skewed distributions. A

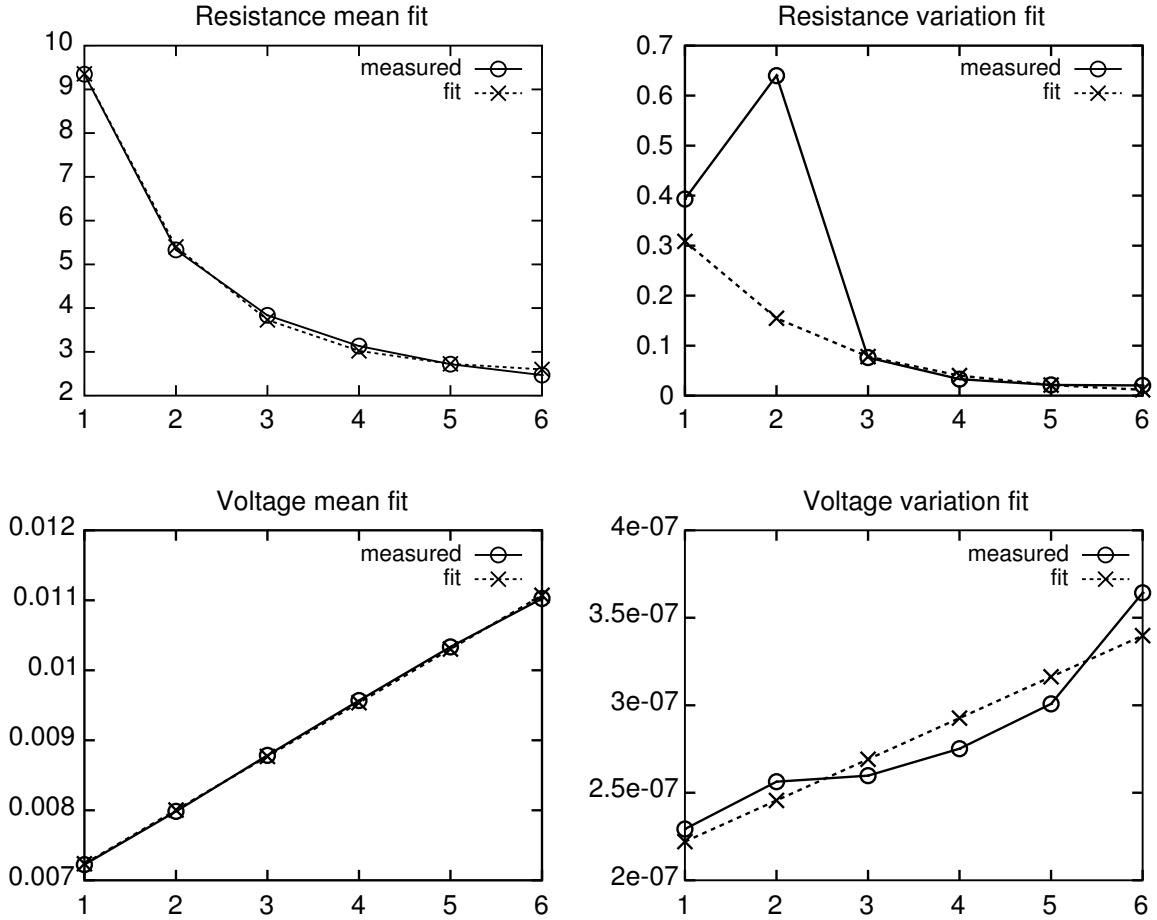


Figure 5.3: Trends and fits of means and variances of ERs and VDrop for the multiple-on scenarios.

similar distribution and conclusion holds for the VDrop analysis (not shown).

5.2 Single-Bit Probability Analysis

A more quantitative evaluation of PUF randomness is presented in this section. The single-bit probability evaluates the symmetry in the statistical distribution of each ER response, as opposed to the entire population as shown in Figure 5.5. In this analysis, we first *discretize* the ER responses by computing the mean of each of the

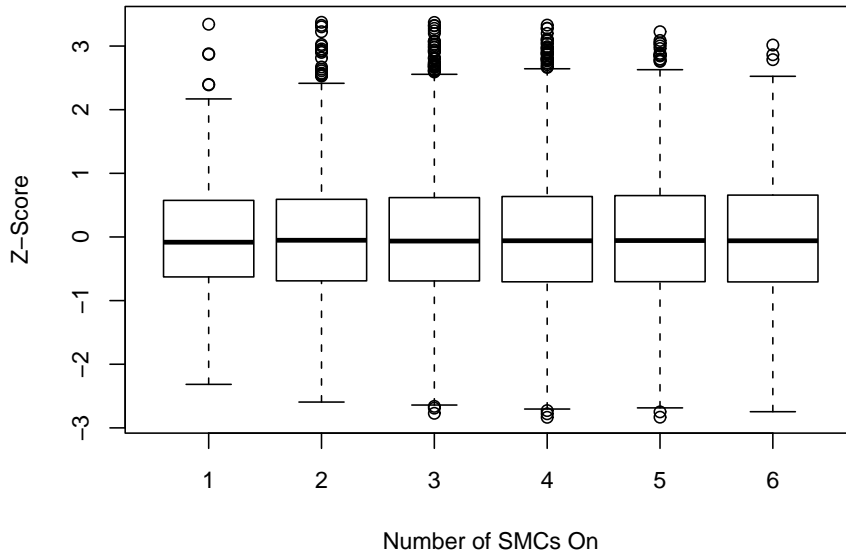


Figure 5.4: Scaled box plots of 1-on through 6-on (x -axis) ER values (y -axis) measured from 36 chips.

192 response values. Each of the 192 means are used to threshold the 36 individual responses from the chips. Response values larger than the mean are assigned 1 while those below the mean are assigned 0.

It is not clear how to perform this operation (comparing a voltage with the mean voltage) in hardware, however there are several possible solutions. For example, the voltage at one SMC could be designated as a representative of the mean, or the voltage could be compared with all other voltages and the number of 1's (from being greater than) could be counted.

The level of randomness can then be easily measured by counting the number of '1's and '0's in each set. Sets that have equal numbers of '1's and '0's, i.e. 50% of each, are perfectly random. Each bit is then like a flip of a fair coin. Figure 5.6 gives the results of the analysis using ERs. The x -axis numbers the response bit groups from 1 to 192 and the y -axis gives the probability of a '1' across the 36

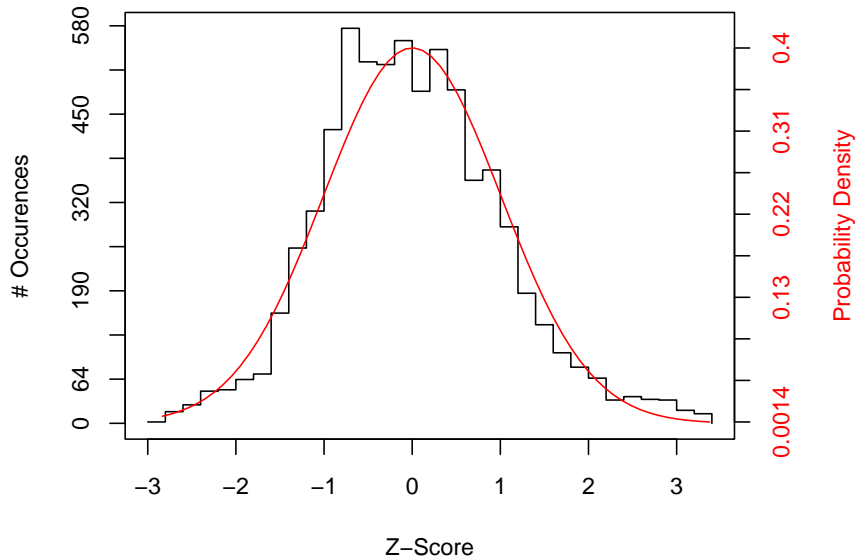


Figure 5.5: Histogram and Gaussian fit of standardized ERs from 192 responses and 36 chips.

chips analyzed. It is clear that the individual distributions cluster around the ideal behavior of 50%, with deviations ranging from 40% to 60%. The average probability across all 192 groups is 47.5% for the ER analysis, and 54.5% for the VDrop analysis (not shown).

5.3 Collision Probability Analysis

In this section, we analyze the probability that two chips produce the same response, as we did in Chapter 4, but we will now consider the multiple-on scenarios. Although this analysis can be performed using the binary versions of the ERs, as described in Section 5.2, the analog ERs more accurately portray the true variations in the data and allow noise to be more easily factored into the analysis. The analysis is carried out on pairings of the chip response vectors. With 36 chips, there are 630

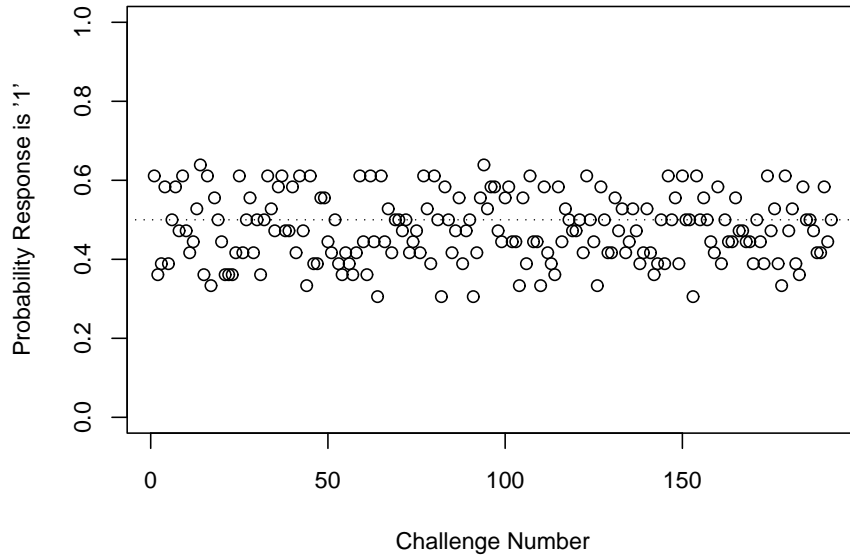


Figure 5.6: Single-bit probability analysis of the ER PUF.

such pairings (36 choose 2). The ERs (or Voltage Drops (VDrops)) for a specific IC are arranged into a 192-dimensional vector, $R = [r_1, r_2, \dots, r_n]$, and the Euclidean Distance (ED) between each pairing of vectors is computed using Equation 4.2. The probability of a collision is computed by creating two histograms: one constructed using all 630 ER EDs from the 36 chips and one constructed from a set of 72 noise samples, obtained by repeating the entire SMC measurement process 72 times using one of the chips. The number of pairings and resulting EDs for the noise samples is 2556 (72 choose 2). We then fit each histogram using a gamma Probability Density Function (PDF). The probability of a collision is computed by first determining an ED value that bounds 99.73% (3 sigma) of the area under the noise PDF. The area to the left of this value in the chip PDF expresses the probability of collision [41].

We compute the probability of a collision by creating two histograms that estimate the measured probability density function, one for the chips using the 630 ER EDs and one for the noise data using the 2,556 ER EDs. We then fit each histogram

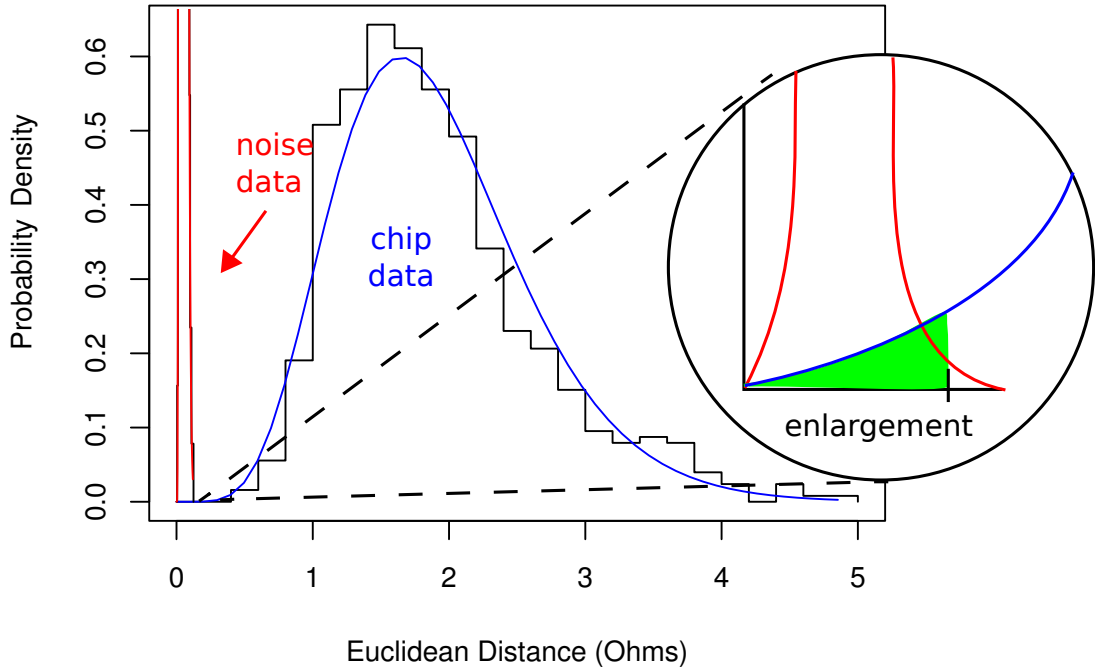


Figure 5.7: Gamma function fit of noise (left) and chip (right) ER Euclidean distance (ED) histograms.

using a gamma PDF, as shown in Figure 5.7. The probability of collision is determined by first determining the Euclidean distance that bounds 99.7% (3-sigma) of the area under the PDF from the noise data set. This value is shown as 0.1 in the blow-up on the right side of Figure 5.7. We then compute the area under the chip PDF to the left of this value, as shown by the shaded region in the blow-up. Finding this area is the same as looking up the noise bound (an x -value) on the corresponding chip Cumulative Distribution Function (CDF). This area is a fraction of the total area under the chip PDF, which is one (1), and expresses the probability of collision.

To understand how the additional responses are adding to the diversity in presence of additional measurement noise, we compute the probability of a collision as a function of the response vector size by considering incrementally-larger sets of re-

response bits. The PDFs for the noise and chip ER analysis are shown in Figures 5.8 and 5.9, respectively, with ED on the x -axis and frequency on the y -axis. The histograms are shown as curves in both figures, with the left-most curves corresponding to the analysis of response vectors using only the 6 1-on values from the noise/chips. The sequence of curves to the right correspond to analyses of increasingly-larger response vectors with values added from the 2-on, 3-on, etc. (x -on) tests. By comparing the noise and chip analysis, it is apparent that the ED of the response vectors increases as values are added for both the noise and chip data, so the merit of including the multiple-on scenarios depends on how these two track. Ideally, the rate of increase in the noise is smaller than the rate of increase in the chip diversity, which is the case for this data.

Figure 5.10 plots the inverse probability of collision (y -axis) as the response vector size is increased from 6 to 192 (x -axis). The increasing trend associated with the curve illustrates that by adding the responses from the higher-order x -on tests, the inverse probability of collision increases by a factor of 36. Table 5.2 summarizes the important characteristics of the analysis: the maximum noise and minimum chip ED measured, the threshold chosen on the noise, and the corresponding probability of aliasing $P(\text{alias})$. This analysis indicates that these additional responses add to the diversity in the response vectors. It is also clear, however, that the diversity increase begins to saturate with the addition of the 5-on and 6-on responses. Therefore, increasing the number of simultaneously SMCs beyond 6 is of limited value.

5.4 Entropy Analysis

The primary objective of this analysis is to determine the level of entropy that exists in various subsets of the ER and VDrop response vectors, including the new multiple-on scenarios. The analysis is performed on the digital values, as discussed in Section

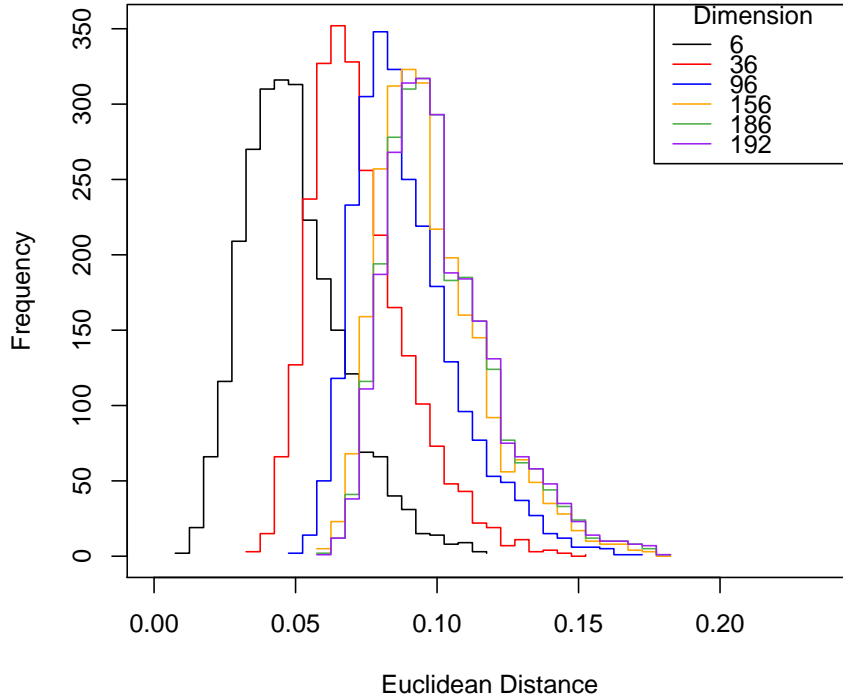


Figure 5.8: ER ED cumulative PDFs of noise for groups 1-on through 6-on.

5.2, and is computed by comparing pairs of ER and VDrop response values on the same chip. This models an actual use scenario in which a response bit is determined by the relative differences in the analog response from two SMCs in the circuit, which is emulated using two configurations of the PUF circuitry.

n	Max Noise	Min Chip	Threshold	P(alias)
6	0.1172	0.4740	0.1092	4.27e-07
36	0.1507	0.8061	0.1247	3.29e-08
96	0.1735	1.0055	0.1431	1.55e-08
156	0.1811	1.0728	0.1539	1.22e-08
186	0.1841	1.0858	0.1583	1.18e-08
192	0.1849	1.0875	0.1591	1.19e-08

Table 5.2: Collision Analysis

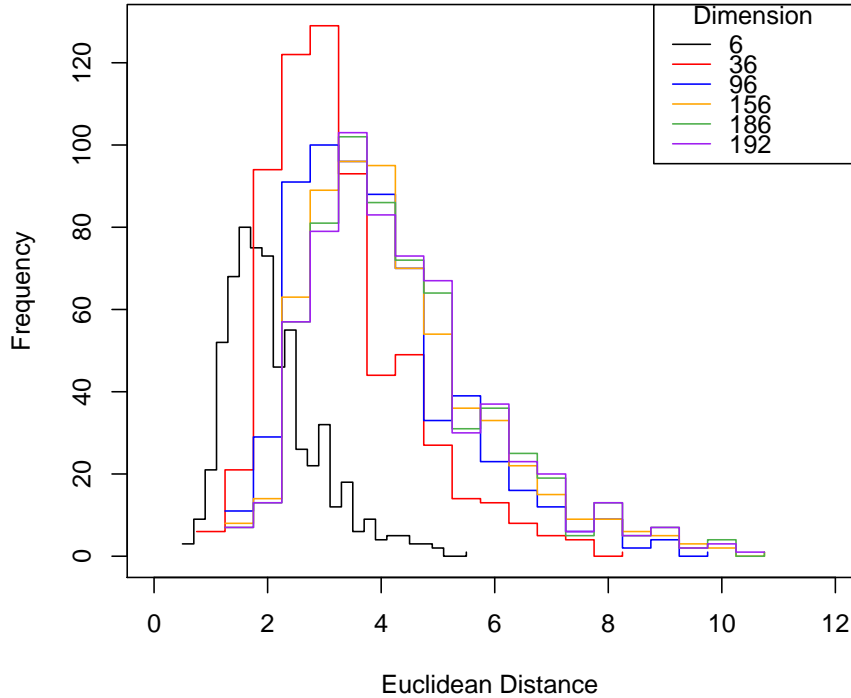


Figure 5.9: ER ED cumulative PDFs of chips for groups 1-on through 6-on.

A response bit in our analysis is ‘1’ if the first ER or VDrop response of a SMC pairing is larger than the second, and ‘0’ otherwise. To determine upper and lower bounds on entropy, we consider two ways of selecting the pairs. In the first analysis, called **Core**, only 5 pairings of the 6 SMCs are considered, as a means of avoiding correlation (see [38]). We treat the results of this analysis as a lower bound on the available entropy. The Core analysis pairings are illustrated in Figure 5.11(a) as P_0 through P_4 . The second, called **All**, includes all possible pairings of the 6 SMCs, which generates $6 \times 5/2 = 15$ bits.

As indicated earlier, it is possible to enable more than one SMC at a time. The ER response bits under the x -on scenarios can be different from the response bits from the 1-on scenario because they are affected by the total current, which is a

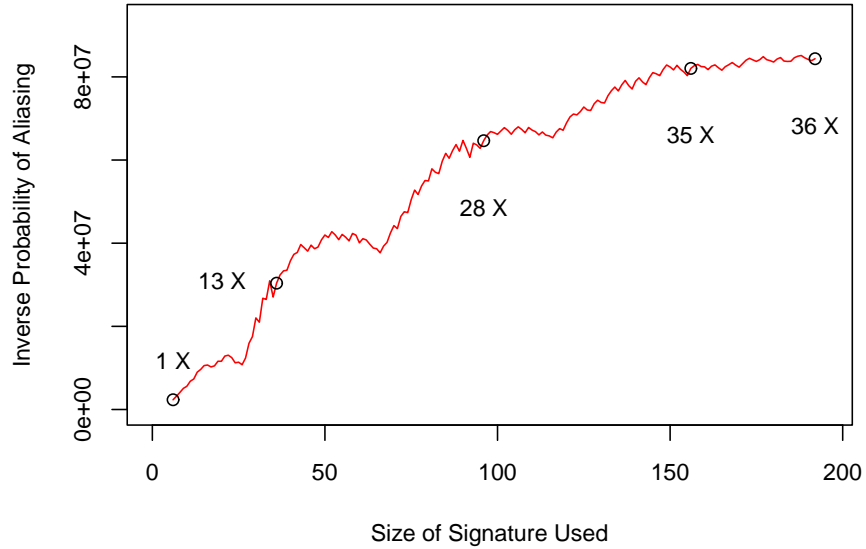


Figure 5.10: Collision probability using ER response vector sizes from 6 to 192.

function of multiple independent shorting currents. Changes in the relative values of the ERs on the same chip will reflect as bit-flips as shown by the example in Figure 5.11 (b) and (c). The response vectors under (b) portray the response bits across the 5 pairings in the 1-on Core analysis. The response vectors for each of the chips, C_x , are given as rows. In contrast, (c) shows the response vectors under the 2-on scenario for the same chips and pairings. The values in parenthesis on the far right are the Hamming Distance (HD) between the two vectors. For example, C_1 under

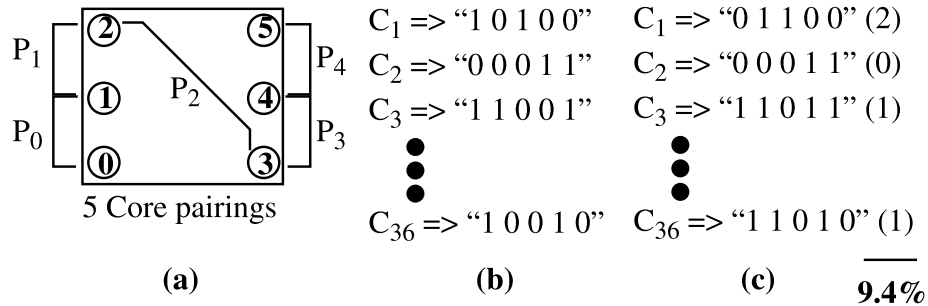


Figure 5.11: Pairing and analysis illustration.

(c) has two bit-flips (and an HD of 2) when compared with the vector under (b).

We use the HD to measure how much entropy is added over the 1-on base case for each of the x -on scenarios. The value of 9.4% given in the bottom-right of Figure 5.11(c) is computed by summing the HDs of the individual chips and dividing by the total number of bits that are compared. For the Core analysis shown in the example, the sum of the 36 chip HDs is 17. The entropy measure of 9.4% is computed as $17/180$, where the denominator is computed as $36 \text{ chips} \times 5 \text{ bits}$. The curves in Figure 5.12 show the average increase in entropy across the 6 x -on analyses as 4 curves, one each for the Core and All scenarios using the VDrop and ER data sets. The 1-on base case shown as the left-most data point on each of the curves is the probability of an arbitrary response bit being ‘1’, decided by comparing it with the mean (see Section 5.2. For the ER data curves, the probability is precisely 50%. Under the Core analysis, the response vector size is 5 bits for each of the 36 chips. Of the 180 bits (5×36), we observed exactly 90 ‘1’s. The result under the All analysis is 270 ‘1’s, exactly half of the 540 bits (15×36).

The remaining points on the graph each represent the average HD between the previous response vector and the vector generated using the x -on data identified on the x -axis. We refer to this change in entropy as ‘delta entropy’, and it represents the additional diversity obtained by adding those scenarios. For example, the ER Core analysis value for the 2-on scenario is given as 9.4% (we described this case earlier in reference to Figure 5.11). From the graph, the All analysis produced a similar value. Both of these values represent a relatively-small increase in entropy over the 1-on base case. This indicates that the 1-on and the 2-on scenarios are correlated. The VDrop values indicate very little delta entropy. This is intuitive because the VDrop responses under the multiple-on scenarios cannot leverage the cross-coupling interaction of the SMC shorting currents used in the ER response calculation.

For the 3-on through 6-on scenarios, the delta entropies, although small, are

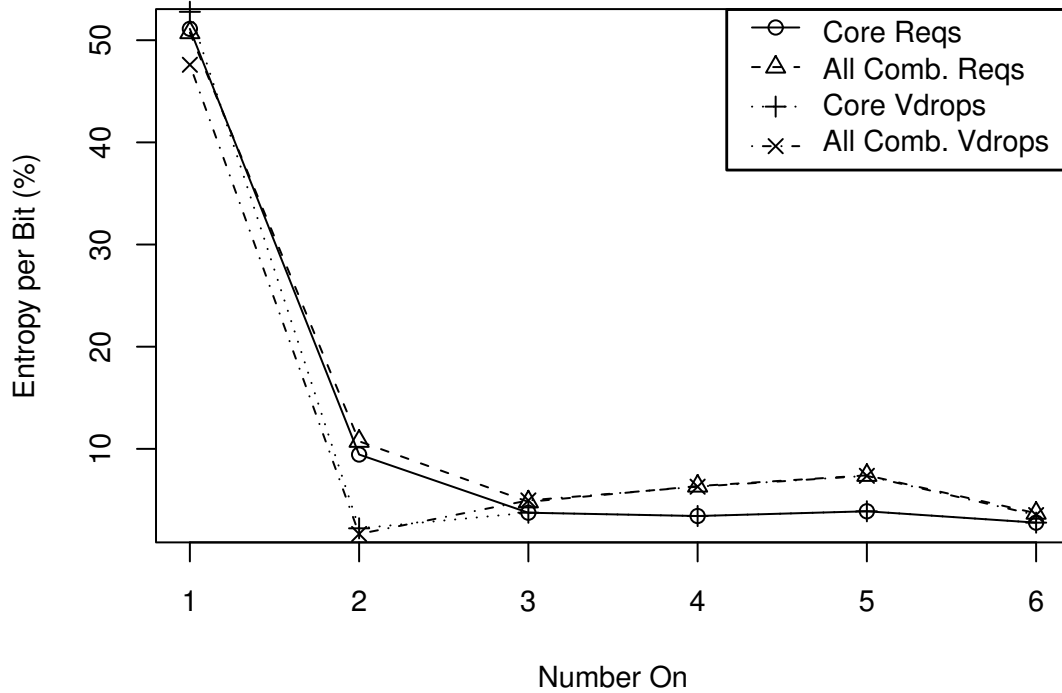


Figure 5.12: Entropy analysis of VDrops and ERs.

not zero and therefore represent a positive increase in the cumulative entropy. For the 3-on through 5-on scenarios, we arbitrarily chose the locations of the additional enabled SMCs, e.g., 1 additional SMC for 3-on, 2 for 4-on, etc., beyond the two used in the pairing. The trends in delta entropy in Figure 5.12 support the behavior of the curve shown in Figure 5.10, which tends to saturate, particularly for the right-most data points representing the 5-on and 6-on scenarios.

5.5 Discretized Signature Evaluation

Given these results, we can approximate the number of response bits that are truly random. As indicated earlier, the Core analysis represents a conservative bound

Table 5.3: Discrete Signature Metrics

Analysis	Inter-chip		Intra-chip	
	ER	VDrop	ER	VDrop
Core	48.0%	48.0%	0%	0%
All	48.5%	48.5%	0.64%	0.59%

where the number of pairing is restricted to $(n - 1)$ per x -on scenario. Therefore, a chip with n SMCs can produce $6 \times (n - 1)$ unique response bits, assuming the delta entropy goes to zero for more than 6 enabled SMCs. For the optimistic All analysis, the number of meaningful response bits is given by Equation 5.3. For our chips, these expressions produce 30 and 255 bits, respectively, with 6 SMCs.

$$N_{\text{bits}} = \binom{n}{2} + \sum_{i=0}^{n-2} \binom{n-2}{i} \binom{n}{2} = \frac{n(n-1)}{2} + (n-1)n2^{n-3} \quad (5.3)$$

We also performed a pairwise HD analysis using the entire 30-bit and 255-bit response vectors from the Core and All analyses, respectively. We compute the average HD per bit by computing the HDs between all possible chip pairs, taking the average HD, and then dividing by the number of bits in the response. Ideally, each comparison should produce an HD that is exactly half of the number of bits in the response vector. The evaluation of our PUF under this metric is summarized in Table 5.3. The inter-chip HDs are computed between chips, and the intra-chip HDs are computed using the noise sample. These values compare favorably to 46.15%, as reported in [38].

We also evaluate reproducibility by carrying out a second pairwise HD analysis using the 72 sets of ‘noise’ samples described earlier. The average HD is computed, as described above, using the 30-bit and 255-bit response vectors. These results are also listed in Table 5.3. These results also compare favorably with 0.48% obtained in [38] and provides evidence that our PUF is robust to environmental noise and

Chapter 5. Extension of the PUF and Evaluation of Metrics

ambient temperature variations.

Chapter 6

Temperature Effects

In this chapter, I introduce active temperature control to the PUF analysis. The changes to the setup and the characteristics of parameters such as leakage current are presented in Section 6.1. Adding temperature control has two benefits. First, it helps us to improve the stability of the chip temperature, which is affected by changing room temperature conditions that we had no control of previously. I make a comparison of how stable the on-chip temperature and other parameters are both with and without control in Section 6.2. Second, temperature control allows us to subject our PUF to different temperature points between 0°C and 75°C and characterize the effects on performance. By aggregating samples from these extremes, we can comment on the effect that temperature has on the PUFs in the worst-case. The metrics on the analog and digital PUFs are revisited with temperature control in Sections 6.3 and 6.4, respectively. In Section 6.3.3, vector angles are also considered as a measure between vectors that is more resistant to temperature-induced changes in the analog vectors than the Euclidean distance. Finally, in Section 6.5, I discuss a reality of our chips that affects the accuracy of our measurements, and also explains some of the peculiar relationships of our parameters with respect to temperature.

Chapter 6. Temperature Effects

As a brief review of the techniques presented in Chapters 4 and 5, the LabVIEW VI that collects the PUF data yields a table for each run of the experiment. As is discussed in Section 5.1, if we consider the 6 single-shortening scenarios and include the multiple-on (2, 3, . . . , 6) scenarios, we have a total of 192 configurations. Each table then has 192 rows, one for each configuration. The columns of the table include four raw measurements that we make: the leakage current and voltage, and the current and voltage with the shorting transistors on. We refer to the nominal condition as the “leak” condition, and the condition with the shorting transistor on as the “short” condition. I compute the voltage drop V_{drop} as the difference of the leakage voltage and the short voltage, $V_{\text{leak}} - V_{\text{short}}$. The shorting current I_{on} , for the shorting inverter is computed using the difference of the total current under the shorting condition and the leakage current, $I_{\text{short}} - I_{\text{leak}}$. This represents the additional current drawn by the shorting transistors. The equivalent resistance R_{eq} is given by the ratio of the voltage drop to the shorting current,

$$R_{\text{eq}} = \frac{V_{\text{leak}} - V_{\text{short}}}{I_{\text{short}} - I_{\text{leak}}}.$$

6.1 Modifying the Experiment Setup

Our test PCB was designed at IBM to enable the test chips to be inserted in a clamshell apparatus. This allows them to be changed quickly, and also supports temperature control hardware. The apparatus consists of four posts attached to a heat sink and a tightening mechanism that retains a stack of parts, and aligns the pads on the chip to the pads on the PCB. See Figure 6.1. At the bottom of the stack is an aluminum part into which the ends of the four posts lock. Through this piece is a screw that allows pressure to be applied to the next metal piece in the stack. That metal piece firmly applies pressure, pushing the PCB up against the heat sink, and compressing the whole stack. Between that piece of metal and the

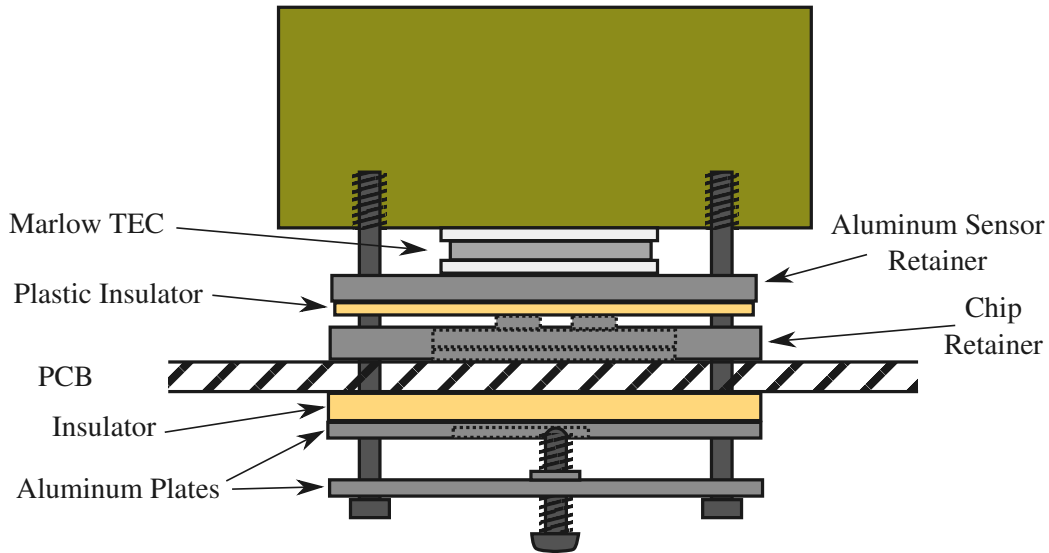


Figure 6.1: Clamshell apparatus cross-sectional diagram

PCB is an insulating plastic piece which protects the pads on the PCB from the aluminum plate. The chip sits in a retainer which has an array of gold spring-like pins that connect the ball grid pads on the chip package to the pads on the board when under pressure. The chip package has two chips on it, which rise slightly out of the package. Above this, we have another plastic insulator, which is thin enough that heat insulation is minimal, and a reasonable amount of heat transfers from the plate into the chips. This insulator serves to protect the chips from the aluminum sensor retainer, both from mechanical damage and condensation.

I built the aluminum sensor retainer in the department machine shop using a milling machine and a drill press. It accepts two disc thermistors with their lead wires. The lead-wires were insulated with heat-shrink wrap and the thermistors are held in place with some thermal paste and pressure from the clamshell apparatus. The aluminum sensor retainer is designed to conduct heat from the Marlow TEC to the plastic chip insulator and to the chip, while retaining temperature sensors. One

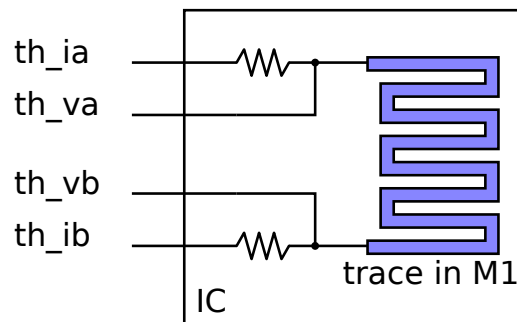


Figure 6.2: On-chip resistor for characterizing resistance

thermistor is connected to a multimeter to allow the temperature to be displayed, providing a redundant measure of temperature. The other is connected to the temperature controller for feedback. The thermoelectric cooler is a Marlow Industries DT6-4L. It is capable of handling up to 3.7 amps (I_{\max}) at 8.2 volts DC (V_{\max}) and is capable of moving up to 22 Watts of heat. The temperature controller is a Newport Model 325B, and can drive up to 2.5 amps. It can work with several types of temperature sensors. The first is any Negative Temperature Coefficient (NTC) thermistor that can be modeled by the Steinhart-Hart equation, which is defined later in Equation 6.6. The second type are IC sensors, which accept a constant current and control their voltage to make the temperature-voltage relationship linear. This type of sensor would have been easier to interface with, but they were too large to be easily adapted to our system.

6.1.1 On-Chip Thermistor

A long track in the lowest metal layer (M1) is embedded in the chip for the purposes of measuring the resistance per square in that layer. A four-wire configuration is provided so that the resistance of the track can be accurately measured without being affected by the resistance of the probe wiring. There are four connections,

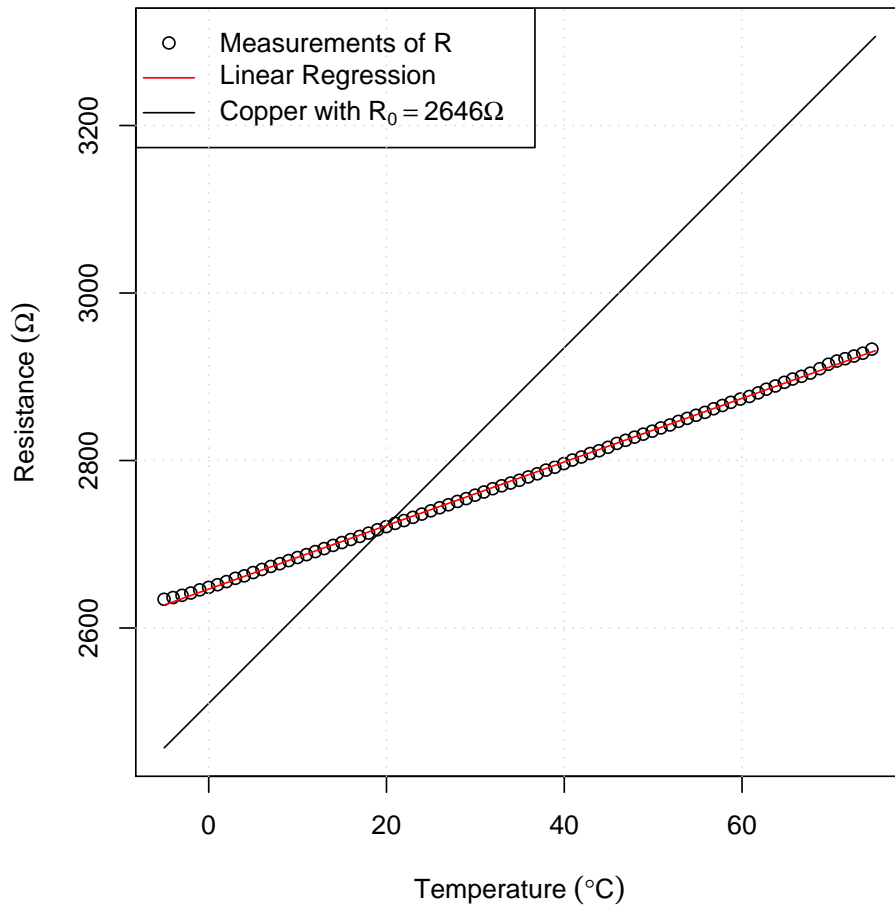


Figure 6.3: On-chip resistance versus temperature

+current, -current, +voltage and -voltage. The four-wire method works by connecting a constant-current source between the +current and -current connections, and a voltage meter between the +voltage and -voltage connections. The current is then controlled, for example $100\mu\text{A}$. The voltage lines measure the voltage accurately at either end of the snaking metal trace since no current is drawn through these lines. Then, it is straightforward to find the resistance by applying Ohm's law, $V = IR$, since I is known and V is measured.

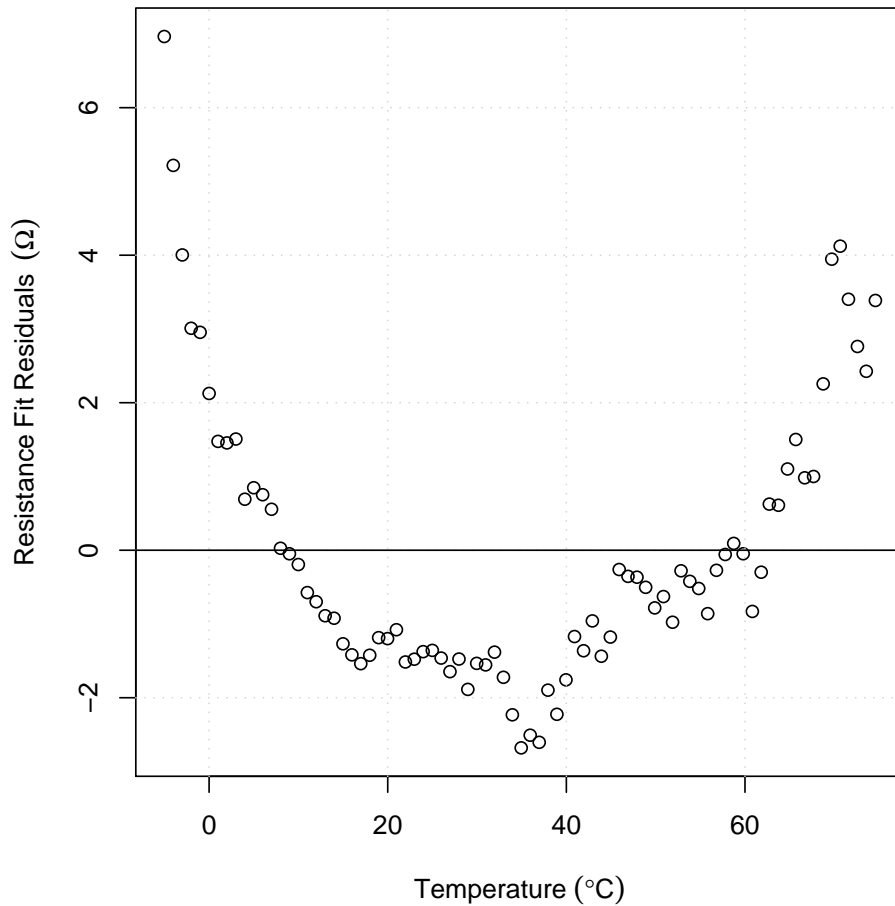


Figure 6.4: Residuals for linear regression of on-chip resistance versus temperature

The resistance as a function of temperature for many metals can be approximated by a linear model involving the temperature coefficient of resistance α . The model for a generalized resistor can be written as

$$R = R_{\text{ref}}[1 + \alpha(T - T_{\text{ref}})], \quad (6.1)$$

where R_{ref} is the resistance at the reference temperature T_{ref} in $^{\circ}\text{C}$, α is the temperature coefficient of resistance and T is the specific temperature (the independent variable). For copper, with $T_{\text{ref}} = 20^{\circ}\text{C}$, $\alpha = 0.0039$ [46].

Chapter 6. Temperature Effects

The metal track on the chip can also be used as a thermistor since its resistance is a function of temperature. We were unable to use this thermistor in our experiments because we would have to characterize the resistance-temperature function for each chip and this would not have been feasible. I performed a temperature sweep from -5°C to 75°C in one-degree increments. At each step, the resistance was measured with the instrument in 4-wire mode after thermal equilibrium (as described later in Section 6.1.3) was reached. Figure 6.3 shows these 80 measurements as circles \circ . Then, I performed a linear regression using $y = mx + b$, which yields the slope $m = 3.796$ and the y -intercept $b = 2646$. This line is drawn in red in the figure. From this linear model, $R_0 = 2722\Omega$. Then, from Equation (6.1), we can draw a line that pivots about the point $T = 20$, $R = 2722$ and represents the behavior the resistor would have if it were pure copper.

Although it is clear from the figure that a linear fit is reasonable, there is some minor error in the fit. I have plotted the residuals, which represent the difference between the measurement and the fit, in Figure 6.4. This error probably represents heat being lost and gained to the room, at the hot and cold temperature extremes, respectively. Since the resistance of the copper metallization layer that we're measuring should certainly be a linear function of temperature, this error must represent the limits of our temperature control apparatus. In other words, although it is very close, the chip is not at the exact temperature target at the hot and cold extremes, and this will affect our further measurements.

Next, I wish to relate the coefficients m, b of the linear regression with the parameters of the physical resistance model. From Equation (6.1), when $T = T_{\text{ref}}$, it is clear that $R = R_{\text{ref}}$. The linear model $y = mx + b$ also yields R_{ref} when $T = T_{\text{ref}}$, so $R_{\text{ref}} = 2722\Omega$ using m and b from above for the experimental data. I can also solve

Chapter 6. Temperature Effects

for m and b in terms of R_{ref} and α by rearranging terms in Equation (6.1),

$$R = R_{\text{ref}}[1 + \alpha(T - T_{\text{ref}})] \quad (6.2)$$

$$= R_{\text{ref}} + R_{\text{ref}}\alpha(T - T_{\text{ref}}) \quad (6.3)$$

$$= R_{\text{ref}} + R_{\text{ref}}\alpha T - R_{\text{ref}}\alpha T_{\text{ref}} \quad (6.4)$$

$$= \underbrace{R_{\text{ref}}\alpha T}_m + \underbrace{R_{\text{ref}}(1 - \alpha T_{\text{ref}})}_b. \quad (6.5)$$

From this, it is clear that $m = R_{\text{ref}}\alpha$ and $b = R_{\text{ref}}(1 - \alpha T_{\text{ref}})$. From $m = R_{\text{ref}}\alpha$, $\alpha = m/R_{\text{ref}}$ and therefore $\alpha = 0.001394$. This experimental α is 64% smaller than the known coefficient for pure copper, which is an indication of the quality of the metallization [47].

6.1.2 GE Thermistor Characterization

In this section, I present the characterization I did of the thermistor that we used to measure the temperature as close as possible to the chip and provide feedback to the temperature controller. In an ideal setup, the thermistor would be on-chip, but as mentioned in Section 6.1.1, it was not possible to use our on-chip thermistor with our temperature controller.

Most NTC thermistors can be modeled by the Steinhart-Hart Equation, which is a third-order polynomial that relates the temperature T to the natural logarithm of the thermistor resistance R [48]. The equation is

$$\frac{1}{T} = a + b \ln(R) + c \ln^3(R) \quad (6.6)$$

The parameters a , b and c are specific to each device and published by the manufacturer. The error in the Steinhart-Hart equation is generally less than 0.02 °C in the measurement of temperature [48].

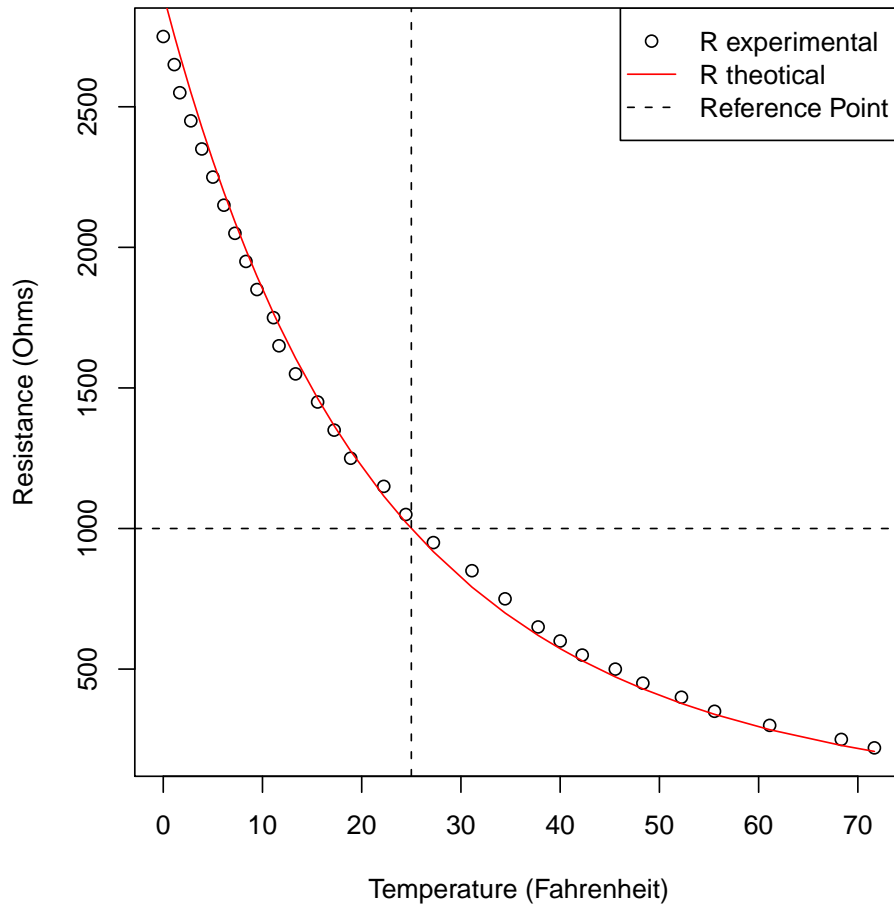


Figure 6.5: GE RL1007-624 thermistor characterization

However, the thermistor we used has its specifications given using the B -parameter equation, which is an alternative representation of the Steinhart-Hart equation for NTC thermistors. Instead of a , b and c , the temperature-resistance relationship is specified by R_0 and B . The B -parameter equation,

$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{B} \ln \left(\frac{R}{R_0} \right)$$

Chapter 6. Temperature Effects

is obtained by substituting the following for the Steinhart-Hart parameters,

$$a = (1/T_0) - (1/B)\ln(R_0) \quad (6.7)$$

$$b = 1/B \quad (6.8)$$

$$c = 0. \quad (6.9)$$

The specifications given for the General Electric (GE) RL1007-624-73-D1 are as follows. It is a NTC thermistor that has $1\text{k}\Omega$ of resistance at 25°C with $B = 3468$. So, $R_0 = 1\text{k}\Omega$ and $T_0 = 25^\circ\text{C} = 298.15\text{ K}$. However, the NewPort 325B Temperature Controller will only accept Steinhart-Hart thermistor parameters for the purposes of converting internally between resistance and temperature. Therefore, I computed the Steinhart-Hart parameters using Equations (6.7) through (6.9), which yield:

$$a = 1.3622 \times 10^{-3} \quad (6.10)$$

$$b = 2.8835 \times 10^{-4} \quad (6.11)$$

$$c = 0. \quad (6.12)$$

The temperature controller would not accept $c = 0$, due to some internal limitation, so I was forced to use 0.001×10^{-7} . Using these parameters with the Steinhart-Hart equation, and in conjunction with a temperature sweep I did using an auxiliary thermometer, I was able to create the plot shown in Figure 6.5. From the figure, it is clear that the parameters R_0 and B fit the device well, so we used them directly with the assumption that the manufacturer's specifications are more accurate than our auxiliary thermometer.

6.1.3 Controlling On-Chip Temperature

Now that we have a way to control the temperature of a plate that contacts the chips through a thin film of plastic (see Figure 6.1 in Section 6.1), we can drive the chips very close to a target temperature within a reasonable amount of time. The next challenge is to ensure that the temperature of the chip is both within range of the target temperature and in thermal equilibrium (i.e., the temperature is stable). A good indication of on-chip temperature is leakage current. I observed that a change in temperature has an immediate effect on the chip leakage current, however the leakage current continues to change after the temperature controller has reached its set-point. Therefore, the chip has some “thermal mass” and does not reach the temperature of the plate immediately. However, the leakage current does saturate within a few minutes, indicating thermal equilibrium. Figure 6.6 shows the TEC temperature and chip leakage as the temperature goes through a transition of one degree, from 52°C to 53°C. On the first row in the figure, the temperature is plotted over time. The controller first overshoots the target temperature before settling back down to 53°C. The next row is the controller’s output, the TEC current, which is positive when it is cooling and negative when it is heating. The last row is the leakage current, which tracks the temperature, but requires some time to do so.

After the temperature and the leakage current have stabilized, the first experiment was to characterize the relationship between temperature and leakage current. This helped us verify that our setup is correct and helped us understand more about the leakage current behavior of our chips. The chip was brought to 0°C and then warmed up by one degree Celsius at a time to 75°C. As I performed this sweep of temperature, I made measurements of the corresponding leakage currents after waiting for the system to reach thermal equilibrium at each degree. From semiconductor theory, we know that temperature should have an exponential effect on the leakage current of MOSFETs, since it appears in the exponential of the sub-threshold

Chapter 6. Temperature Effects

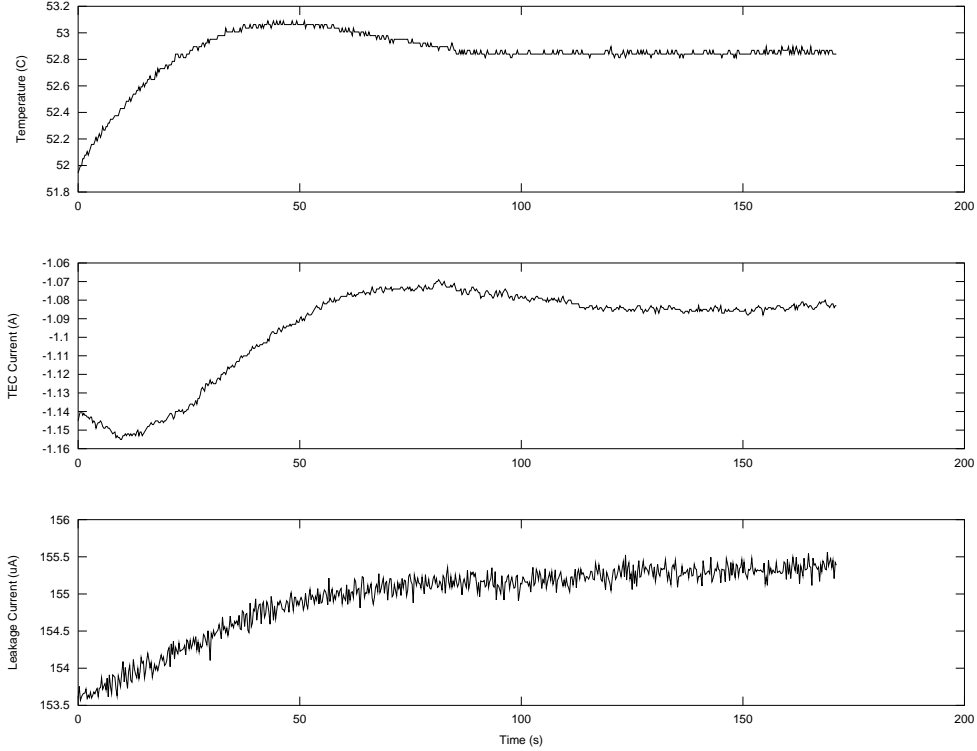


Figure 6.6: Example transition from 52 to 53 degrees Celsius

conduction model [49] for I_D ,

$$I_D = I_S e^{\frac{V_{GS}}{n k T / q}} \left(1 - e^{-\frac{V_{DS}}{k T / q}} \right) (1 + \lambda V_{DS}). \quad (6.13)$$

We are only measuring the leakage current of the entire chip, and most of the circuits involve a PMOS and an NMOS in series. Fitting experimental data to the model in Equation 6.13 is not possible without more specific information, for example, a characterization of single transistors on the same chip. Understanding all of the transistor characteristics was not the goal of this work. Instead, we use the form of a general exponential relationship, $f(x) = a e^{bx}$, where a and b are constants, to characterize the leakage current.

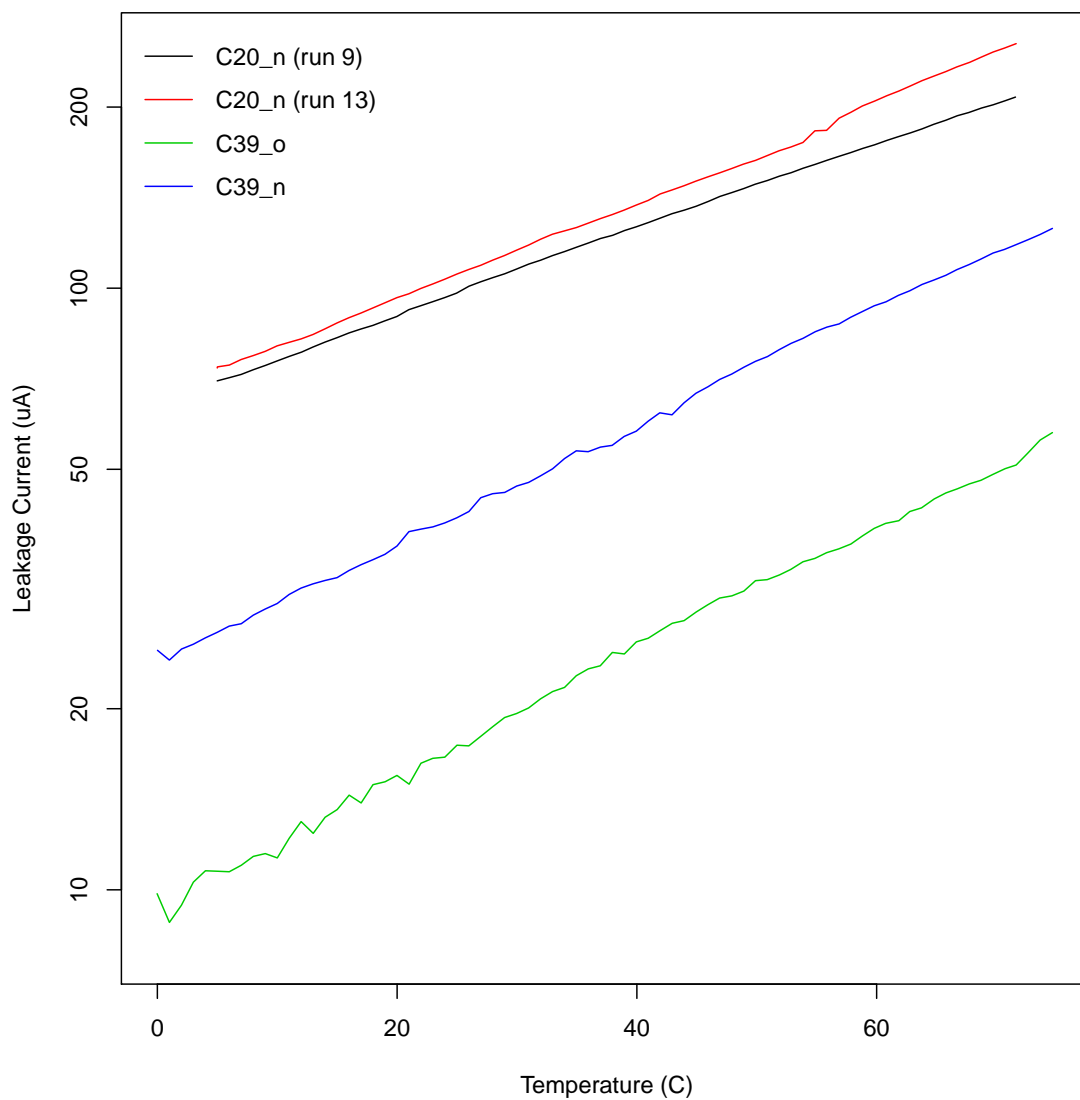


Figure 6.7: Trends of leakage current versus temperature, y -axis is logarithmic

Chapter 6. Temperature Effects

Chip	e^b	m
C20 n	6.605×10^{-5}	0.01865
C39 o	9.476×10^{-6}	0.02410
C39 n	2.400×10^{-5}	0.02256

Table 6.1: Fitted model parameters for leakage current versus temperature using the model $I_{leak} = e^{mT+b} = e^b e^{mT}$

Figure 6.7 shows the trends of leakage current as a function of temperature for measurements made on several chips. Note that the y -axis is logarithmic. From the figure, the trends appear to be straight lines when a logarithmic y -axis is used, and therefore an exponential fit is appropriate for the data. I take pairs of temperature T and the natural logarithm of leakage current I_{leak} , and form (x, y) pairs. Then, I can apply a linear regression $y = mx + b$ to the data using the model,

$$\underbrace{\ln(I_{leak})}_y = m \underbrace{T}_x + b.$$

By using both sides of this equation as the exponent of e , I obtain

$$I_{leak} = e^{mT+b} = e^b e^{mT} = b' e^{mT},$$

where m and $b' = e^b$ are constants. This is then the appropriate generalized form for leakage current versus temperature, as described above. Table 6.1 presents the constants found for various chips using this model.

I created a LabVIEW virtual instrument (VI) in order to manage controlling temperature to different points and collecting the PUF data. A screen shot of the front panel is shown in Figure 6.8. The LabVIEW VI first turns on the temperature controller, then begins to read the temperature feedback, thermoelectric current and chip current. The chip is configured so that all the transistors are off during this process and therefore the chip current is the leakage current. The VI then goes into a loop that waits until equilibrium is reached. It qualifies equilibrium using

Chapter 6. Temperature Effects

the temperature and leakage current and checks for four simultaneous conditions before collecting the PUF data. It waits for (1) the temperature read-back to be within a tolerance of the target (e.g., 10%) using the Kelvin scale, (2) the change in temperature per change in time (dT/dt) to be small enough to indicate that the temperature is stable, (3) the change in leakage current per change in time (dI/dt) to be small enough to indicate the leakage current is stable and (4) enough time to pass to make a stable measurements of dT/dt and dI/dt . It uses the slope of a linear regression to compute dT/dt and dI/dt . The thresholds on these quantities were determined empirically. Since the leakage current is a strong (exponential) function of temperature, we can use the leakage temperature behavior to understand on-chip temperature more accurately than the temperature read-back, which is an off-chip thermistor. Specifically, if the leakage current is not changing ($dI/dt = 0$), then we assume that the change in on-chip temperature $dT/dt = 0$ and it is practically equal to the controlled temperature.

Chapter 6. Temperature Effects

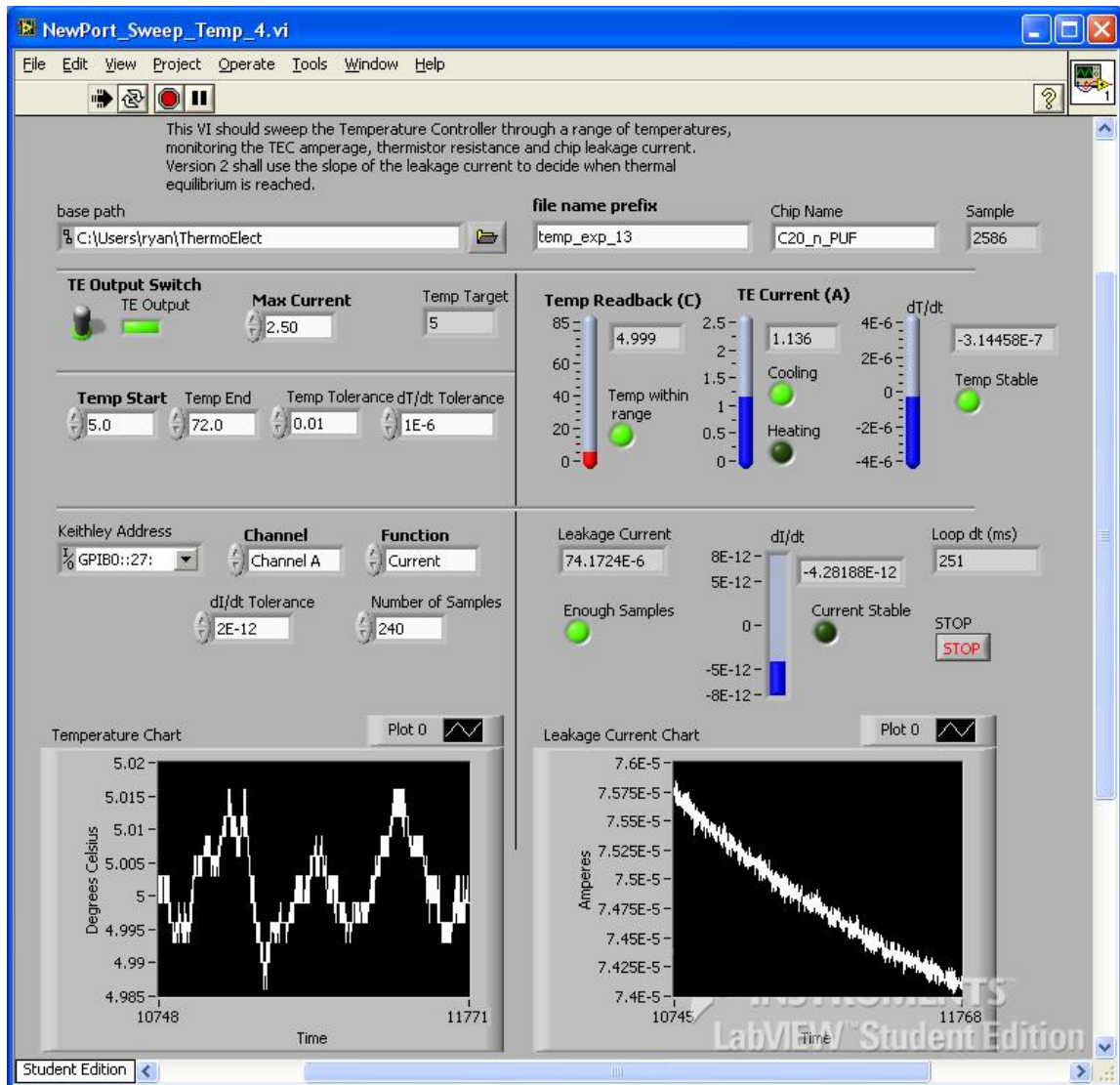


Figure 6.8: LabVIEW front panel

6.2 Noise Analysis

In order to better understand the uncertainty in our measurements, I performed a thorough characterization of the measurement noise. This gives us confidence in the results that we derive using these measurements to understand the behavior of the performance of the various PUFs at different temperature points. In this section, I explain how this was done and present the noise characteristics.

Revisiting our experimental data that we used for the DAC 2010 publication [45], I was able to characterize the variation of the system without temperature control for comparison. The left-hand side of Figure 6.9 shows the $12 \times 6 \times 192 = 13824$ samples of leakage voltage and leakage current and the $\pm 3\sigma$ limits for the sample. We define the noise floor as 3σ , three times the standard deviation, for the noise samples. From the figure, it is clear that there is a high level of variation in the leakage current, presumably due to temperature drift. Although additional leakage current should increase the voltage drop across the power grid, there is not a clear relationship between the drift in the leakage current and leakage voltage.

The right-hand side of Figure 6.9 shows the same two measurements, leakage voltage and current, with the new temperature-controlled experimental setup. It is clear that the $\pm 3\sigma$ range is smaller: $2\mu\text{A}$ rather than $6\mu\text{A}$. The discontinuities that are visible are due to the system cycling through the temperatures, to prevent staying at cold temperatures for too long. Fluctuations in room temperature still introduce some variability in the on-chip temperature after reaching equilibrium. We could eliminate this uncertainty by using an on-chip thermistor in the TEC feedback loop.

Chapter 6. Temperature Effects

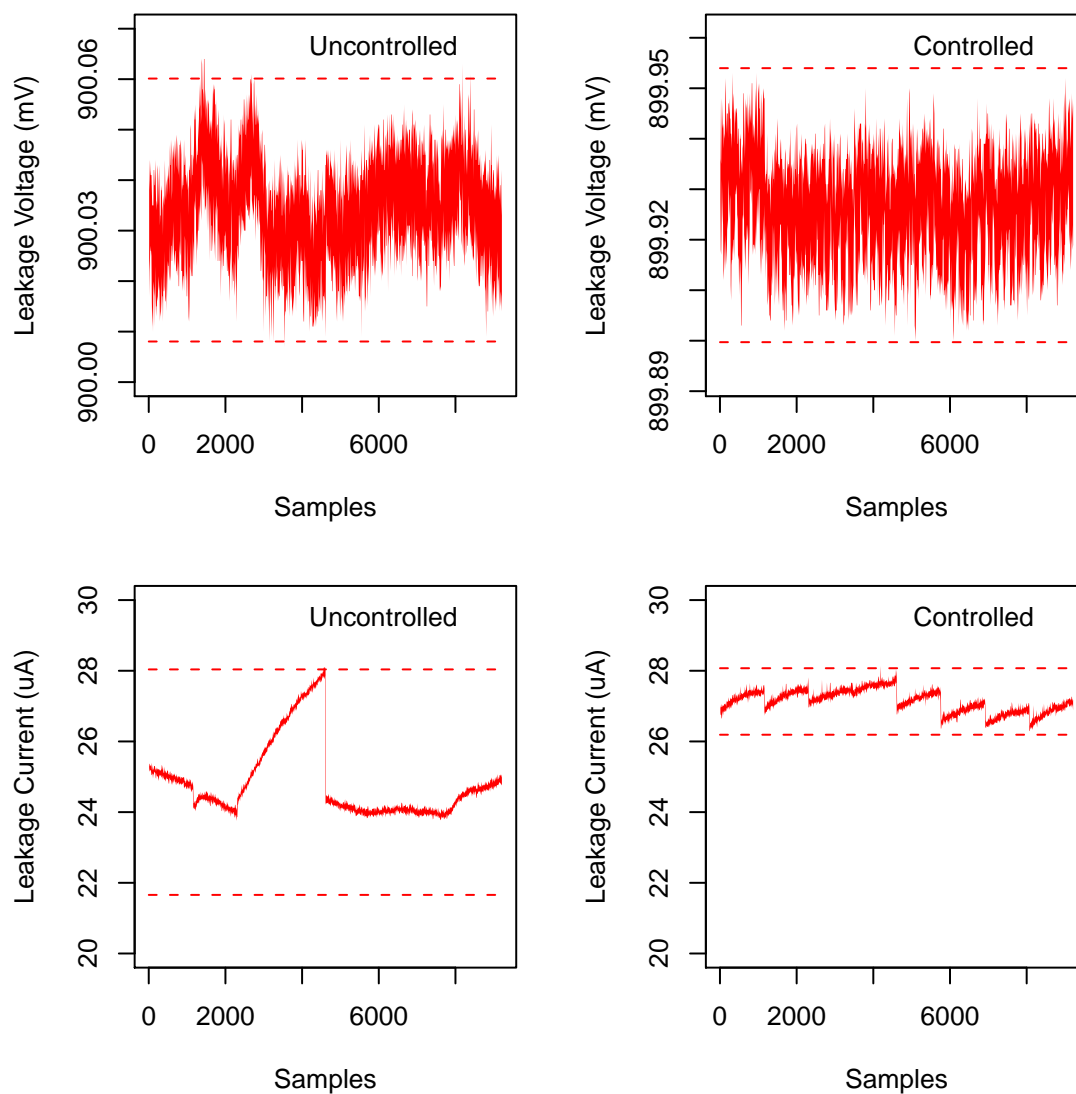


Figure 6.9: Raw measurement noise: leakage voltage and current without and with active temperature control

Chapter 6. Temperature Effects

Figure 6.10 shows the voltage and current measurements that we made with the shorting conditions present. The top row is a plot of all the samples we made, and it is clear from the periodicity that there is a “signal” present. In the bottom row, I have plotted the same variables but with fewer samples so that the periodicity of the signal is clear. The detail view shows the first $2.5 \times 192 = 480$ samples so that two and a half cycles can be seen. This is compelling because it illustrates our “PUF signal”. To understand the noise floor of each measurement, I pick the first one from each period, yielding 12×6 samples of that part of the signal. These 72 samples are plotted in Figure 6.11 on the left-hand side. Both V_{drop} and R_{eq} are relatively stable. They are not highly correlated with the fluctuations in temperature which are seen in the left-hand side of Figure 6.9. The results from the experiment with active temperature control is shown in Figure 6.11 on the right-hand side. The result is similar to that without temperature control, but the noise level is somewhat lower, as I will show later.

In Figure 6.12, I show the mean value of the parameter versus temperature. The $\pm 3\sigma$ limits are drawn as dashed lines in the figure. From the figure, the following trends can be observed. The leakage voltage (top-left), which is affected by both the leakage current (bottom-left) and changes in resistance due to temperature, tends to decrease as temperature increases. For example, approximately four times the leakage current is being drawn through the grid at 75°C than at 0°C , and this additional current pulls the grid voltage down by approximately 150nV. The leakage current (bottom-left) increases exponentially with increasing temperature, as is expected. The short voltage (top-middle) is affected by both the short current (bottom-middle) and changes in the grid resistance. The short voltage and current shown here are the direct measurements; the leakage voltage and current have not been subtracted out. Unlike the leakage voltage, the short voltage (top-middle) is not pulled down lower when there is greater short current (bottom-middle), but rather they do not seem to be directly related. The short current (bottom-middle), which is mostly the

Chapter 6. Temperature Effects

“on” (saturation) current of the transistor, increases as temperature increases. The voltage drop (top-right) is the difference between the leakage voltage (top-left) and the short voltage (top-middle). The voltage drop tends to decrease as temperature increases. The equivalent resistance is the voltage drop divided by the on current (short current minus the leakage current). The equivalent resistance also tends to decrease as temperature increases. The effect of temperature on the V_{drop} and R_{eq} is revisited later in Section 6.3.1. Finally, there is a parasitic leakage current that affects the observe voltage and distorts this analysis. I analyze this effect later in Section 6.5

Figure 6.13 shows the value 3σ used to draw the dashed lines in Figure 6.12 divided by the mean u . The quantity $3\sigma/\mu$, which we refer to as the relative noise floor, represents the measurement noise with respect to the magnitude of the measurement. From this detailed view, we can make the following observations about how the relative measurement noise is affected by temperature. The leakage voltage (top-left) and leakage current measurements are both more stable at higher temperatures. The short voltage (top-middle) is more stable at lower temperatures, while the short current (bottom-middle) shows a minimum noise level at 50°C. The voltage drop and equivalent resistance also have a minimum at 25°C and are worst at higher temperatures.

Table 6.2 reports the 3σ noise floors for the various samples recorded without temperature control (for DAC 2010), from which the following observations can be made. Note that the noise floors reported in Table 6.2 are absolute and not relative as in Figure 6.13. The first three columns of the table represent progressively-larger sample sizes. A run is a single table of the 192 measurements as discussed previously, where we make a leakage measurement each time just before we apply one of the 192 shorting and observing configurations. The variation in a run represents the absolute best-case measurement noise, which we refer to as the instrument noise. A

Chapter 6. Temperature Effects

set is 6 runs, and contains $6 \times 192 = 1152$ leakage measurements. A superset is 12 sets, contains $12 \times 6 \times 192 = 13824$ leakage measurements and is what we use to characterize the measurement noise.

With these definitions in place, we can now make the following observations from Table 6.2. Without temperature control, the temperature tends to drift over time, and therefore as we add more measurements to our sample (run \rightarrow set \rightarrow superset), the noise floor tends to increase. The leakage voltage and current measurements have a noise floor of about $22\mu\text{V}$ and 244nA , respectively. As more samples are taken, the noise floor of the leakage voltage increases slightly to $26\mu\text{V}$, but the leakage current increases by an order of magnitude to $3.2\mu\text{A}$. This indicates that the leakage current is more difficult to control. For the short voltage and current, shown in the third and fourth rows of the table, the run statistics are not available because we single out a single measurement from the 192 (as discussed previously in reference to Figure 6.10). Furthermore, since there are only 6 numbers on which to do statistics for the set, the set statistics for the short voltage and current are not very accurate, but are presented for completeness. However, the superset has 12×6 measurements and is an accurate measure of the noise level for the short voltage and short current. The signal column represents the variation in the measurement across the 192 configurations, and we therefore refer to this as the “signal ceiling”. It is computed exactly like the superset noise floor for the leakage current and voltage, but it encompasses our “PUF signal” and therefore represents the maximum signal dispersion. We can compare the “signal ceiling” with the “noise floor” and produce the final column, labeled signal-to-noise ratio (SNR). For the voltage and current, the PUF signal is approximately 75 times and 1,500 times greater than the measurement noise, respectively. Finally, the last two rows of the table are the V_{drop} and R_{eq} statistics, which are computed in a fashion similar to that of the short voltage and current. Similar to the short voltage and current statistics, the “set” noise floor is based on 6 samples and is therefore not very accurate. As with the short voltage and current statistics, we have a signal to

Chapter 6. Temperature Effects

	Run	Set	Superset	Signal	SNR
Leak V	2.2261e-05	2.2379e-05	2.6038e-05		
Leak I	2.444e-07	4.8069e-07	3.1901e-06		
Short V		1.7740e-05	3.545e-05	0.0026441	74.587
Short I		2.0456e-06	1.598e-06	0.0024963	1562.1
V_{drop}		2.3755e-05	4.1385e-05	0.002647	63.97
R_{eq}		0.04353	0.03565	4.085	114.6

Table 6.2: Noise Levels without Temperature Control

	Run	Set	Superset	Signal	SNR
Leak V	2.5512e-05	2.6348e-05	2.7136e-05		
Leak I	2.4164e-07	5.4763e-07	9.4091e-07		
Short V		3.2390e-05	2.7876e-05	0.0021098	75.686
Short I		7.0103e-07	1.0735e-06	0.0024968	2325.9
V_{drop}		2.988e-05	2.2478e-05	0.002114	94.03
R_{eq}		0.034959	0.029169	4.058	139.1

Table 6.3: Noise Levels with Temperature Control (25°C)

noise ratio which is approximately 64 and 115 for the V_{drop} and R_{eq} .

Table 6.3 reports the same metrics for the experiment with active temperature control. It is clear from the first column that the instrument noise floors are approximately the same as without temperature control, as we expect. Specifically, the leakage voltage and current noise floors are 15% larger and 1% smaller, respectively. The leakage voltage and current superset noise floors, which represent how well we are controlling temperature, are 4% larger and 70% smaller, respectively. The short voltage and current superset noise floors are 21% and 33% smaller, respectively, again indicating reduced temperature variations. Finally, the signal to noise ratios are all higher under temperature control. Specifically, the short voltage, short current, voltage drop and equivalent resistance noise floors are 1.5%, 49%, 47% and 21% greater, respectively.

Chapter 6. Temperature Effects

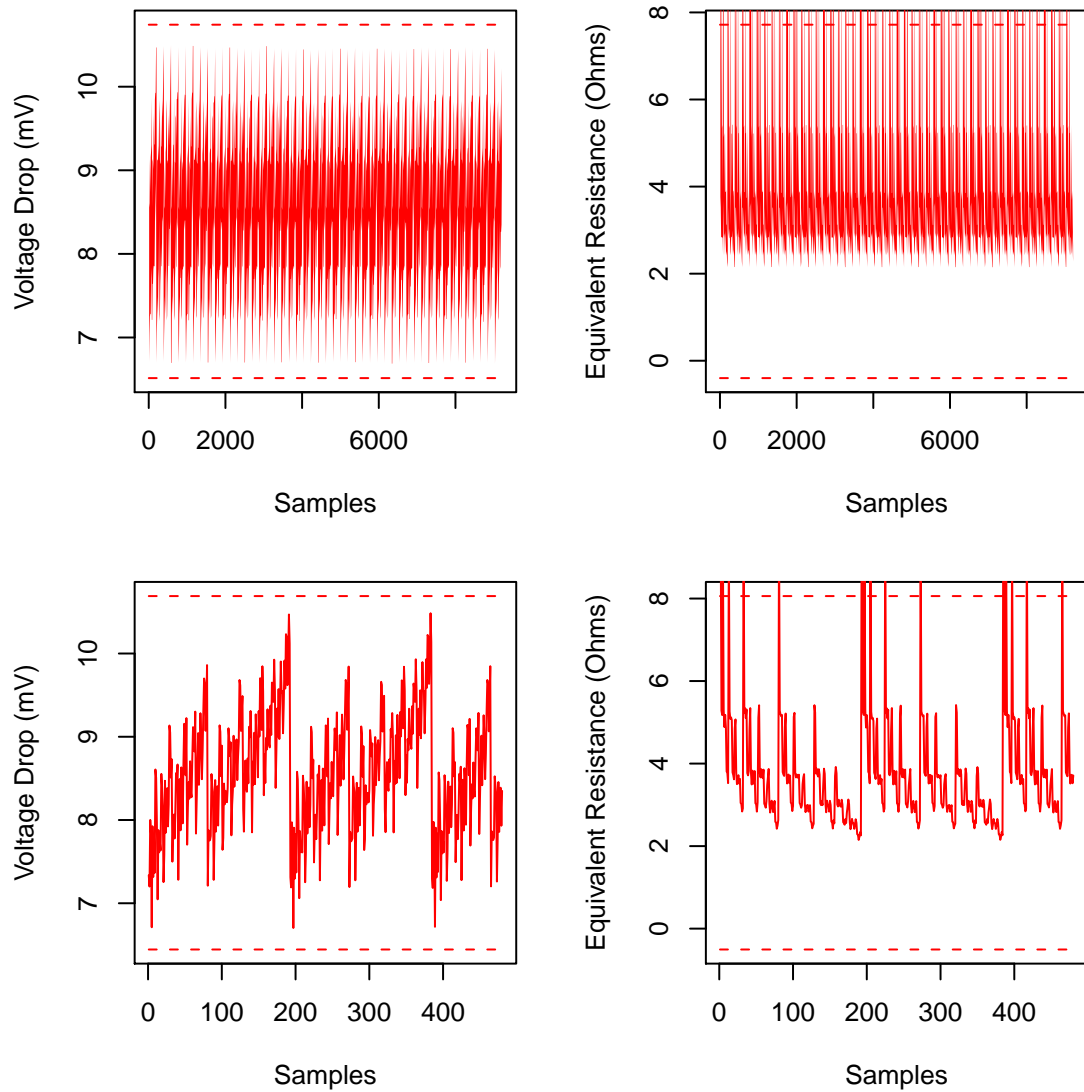


Figure 6.10: Voltage and current measured under shorting condition without temperature control (result similar when temperature is controlled)

Chapter 6. Temperature Effects

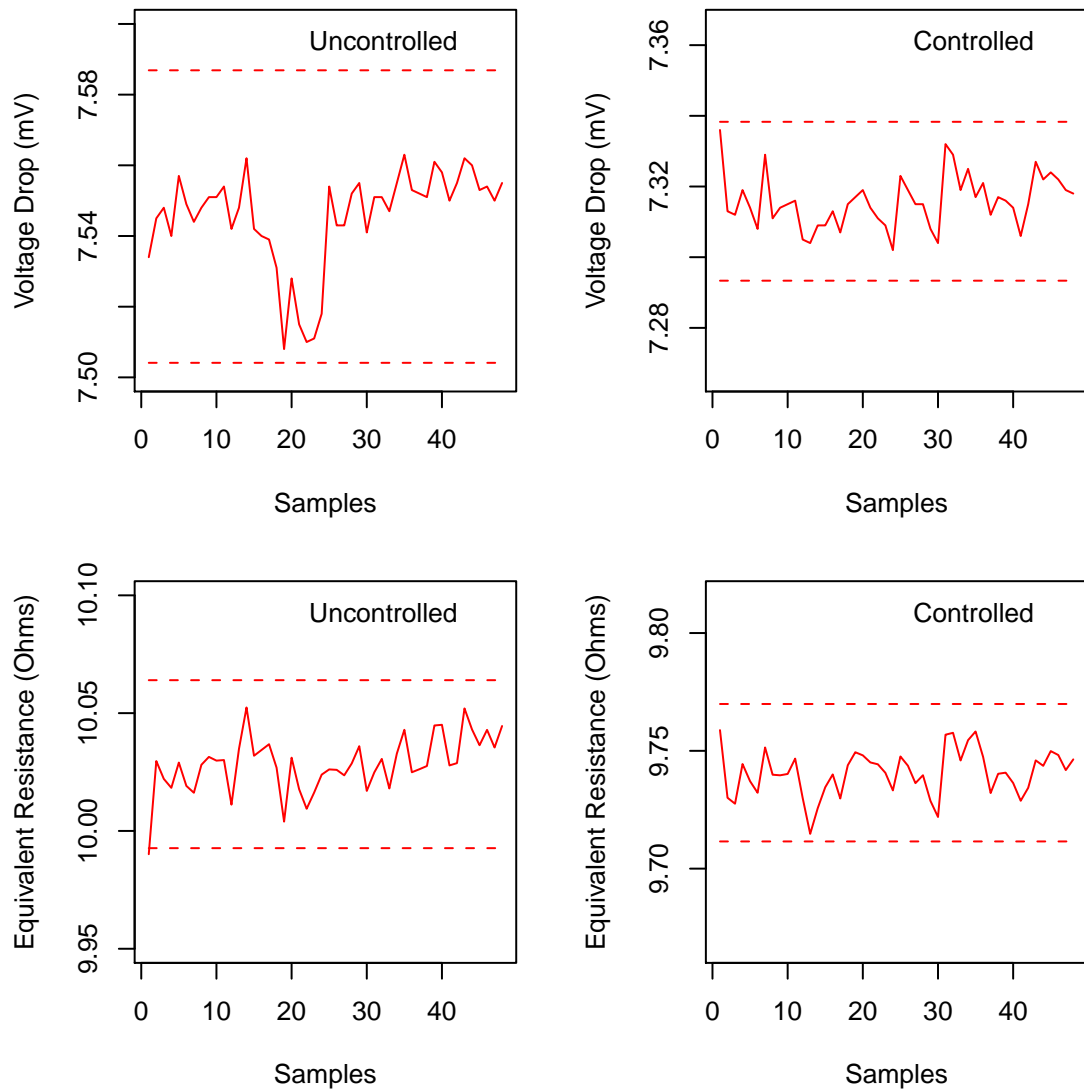


Figure 6.11: V_{drop} and R_{eq} measurement noise with and without temperature control

Chapter 6. Temperature Effects

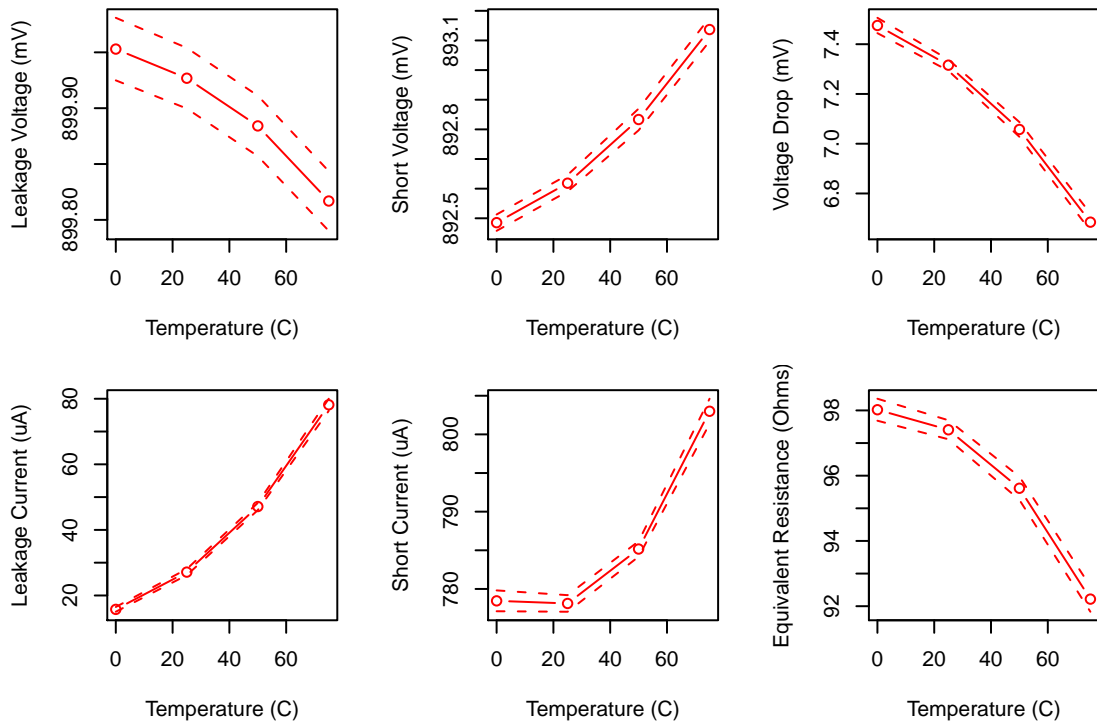


Figure 6.12: Mean and standard deviations of various measurements versus temperature

Chapter 6. Temperature Effects

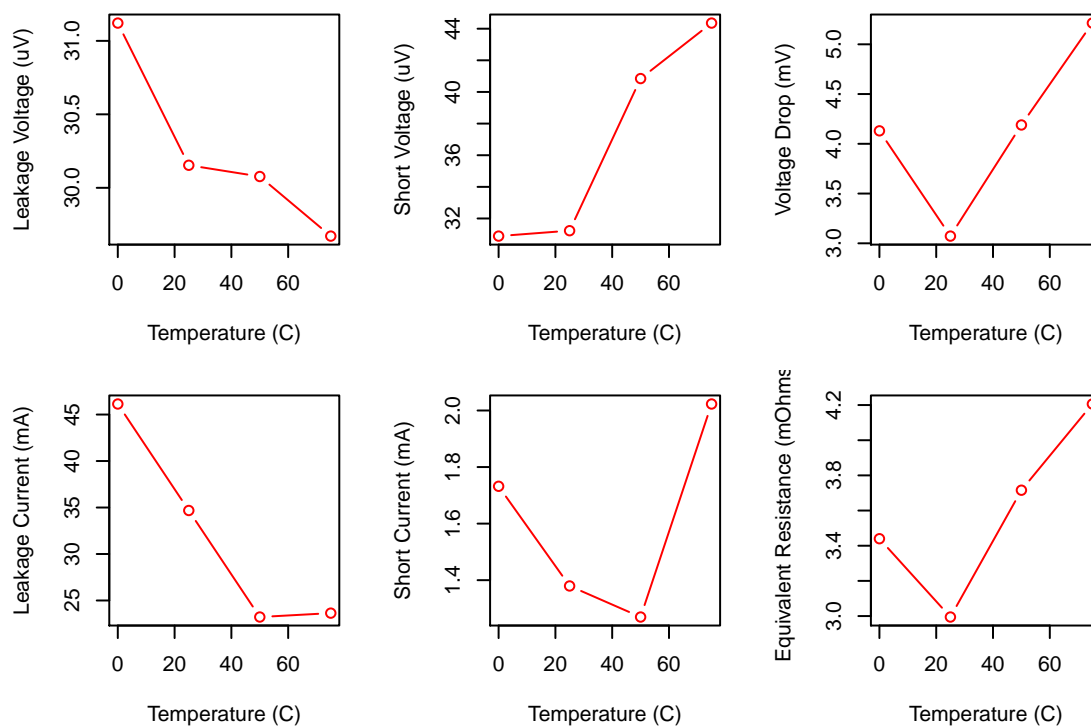


Figure 6.13: $3\sigma/\mu$ relative noise floor for various measurements versus temperature

6.3 Effects on Analog R_{eq} and V_{drop} PUFs

We expect that the analog PUF values will be affected by temperature, but the exact relationships were not well understood. As shown in Section 6.1.3, the leakage current, supply voltage, short voltage and short current are all affected temperature. In this section, I will explain how the analog PUF parameters, equivalent resistance R_{eq} and voltage drop V_{drop} , are affected by temperature, based on new experimental data using the Thermo-Electric Cooler (TEC) setup.

With the experiment run at four temperature points, 0C, 25C, 50C and 75C, the resistances and voltages for the various chips at each point can be compiled. I also collected noise samples, which are repeated runs on a single chip, for two chips in order to characterize the stability of the measurement. I was forced to only make 6 runs at each temperature point due to the temperature extremes involved and the risk of condensation. For this noise experiment, the LabVIEW code was modified to repeat the sweep with 6 runs at the 4 temperature points, and repeat the whole process 12 times to yield a total of $6 \times 12 = 72$ runs at each temperature point. After all the data has been compiled for the various combinations of R_{eq} or V_{drop} , chips, temperatures, I continue the analysis in the statistical computing package called R (<http://r-project.org/>).

6.3.1 Temperature Effects on Analog PUFs

In this section, I describe how the R_{eq} and V_{drop} are affected by temperature, using the 72-point noise sample for C47_o, and how that impacts the performance of the PUFs. This is a further treatment of the relationship of these parameters with temperature shown in Figures 6.12 and 6.13 in Section 6.2. The metrics of Chapter 5 are presented for the new temperature-controlled data.

Chapter 6. Temperature Effects

To characterize this relationship, the first thing I do is average the 72 noise samples into one value for each resistance or voltage (of 192), and repeat for each of 4 temperatures. We can then track each of these physical parameters through the different temperature points. The first issue that presented itself was that the dispersion between the various analog values on the same chip was much larger than the variations that I saw due to the change in temperature. This can be readily seen in the plots of the average values in Figures 6.14. Recall from Section 5.1 that the magnitude of the equivalent resistance is a function of the number of shorts that are on. In the figure, the equivalent resistances are scaled into the same range by multiplying the R_{eq} by the number of shorts that are on, which are marked by different colors. This puts them all around 10Ω for comparison.

6.3.2 Probability Analysis

In this subsection, I will repeat the probability analysis of Section 5.3 with temperature control. In that section, we showed that the addition of the multiple-on test conditions to the 6 single-on scenarios provide the highest-possible level of information from the device, as well as making the number of CRPs exponential with respect to the number of hardware resources. Unless specified otherwise, I will use the 192-dimension vectors from here on.

Recall that in the previous analysis, we computed 192 resistances or voltage drops and considered this vector comprised of all the CRPs as the signature for that chip. Then, in order to measure the dispersion, we compute all $\binom{n}{2}$ Euclidean distances of pairs of these vectors, which is equivalent to an upper-triangular matrix of distances $D_{i,j}$. The same distances are computed for the noise sample, and a histogram is generated for both sets of distances. The histogram serves to illustrate the distribution, but the original sets of chip and noise distances used directly fit to a Gamma distribution. Next, we choose a practical upper bound on the noise

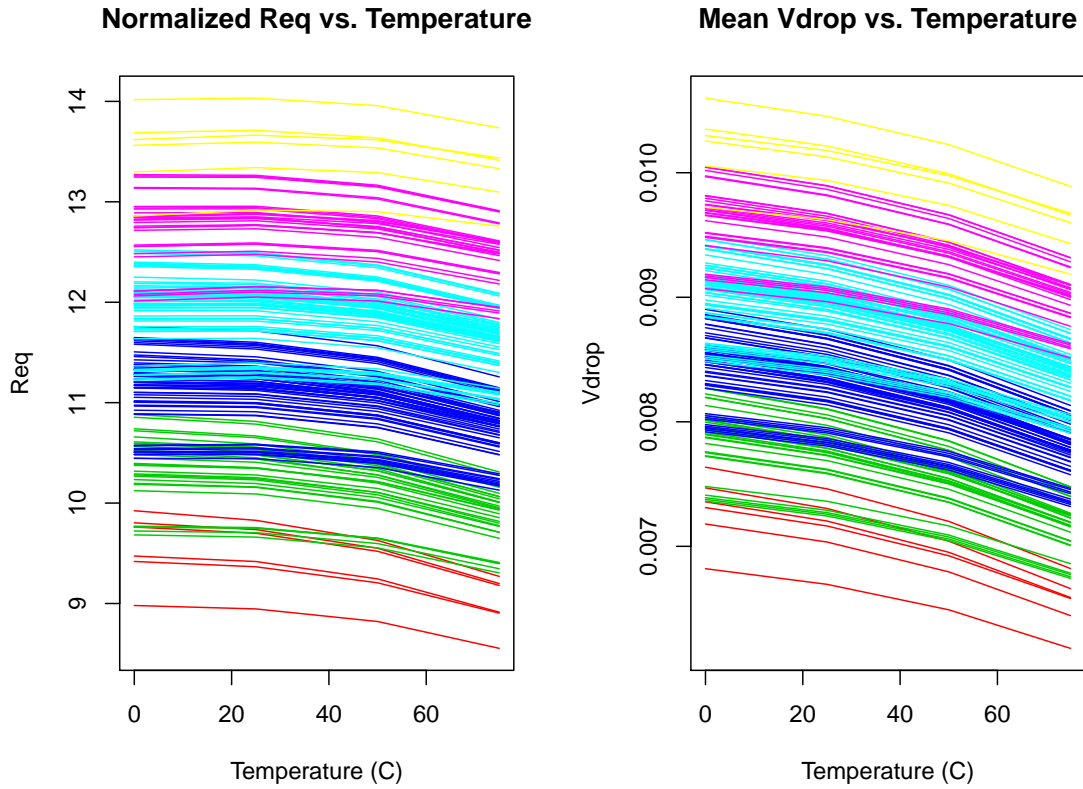


Figure 6.14: 72-point mean R_{eq} versus temperature for the 192-values of the noise sample

distances by finding where the noise CDF is equal to 99.73%, the same area for the $\pm 3\sigma$ interval of the Normal distribution. Then, we apply this threshold to the CDF of the Gamma fit for the chip distances to decide estimated the probability of aliasing. For convenience, Table 6.4 reports all of the estimates of the probability of aliasing we have made previously.

In Figure 6.15, the histograms for all pairwise Euclidean distances of chip (top) and noise (bottom) vectors at 25°C are plotted. I have also plotted the result from the three other temperature points that we used, 0°, 50° and 75°C. The legend

Chapter 6. Temperature Effects

		Aliasing Probability			
		R_{eq}		V_{drop}	
DAC2009	Single-on	6.9×10^{-8}	1 in 15 M	8.8×10^{-9}	1 in 113 M
DAC2010	Single-on	4.27×10^{-7}	1 in 2.3 M	2.96×10^{-5}	1 in 33 k
	Multiple-on	1.18×10^{-8}	1 in 84 M	1.13×10^{-5}	1 in 88 k

Table 6.4: Review of previous estimates of probability. Note: the DAC2010 V_{drop} results were previously unpublished.

indicates the colors that are used for each histogram. It is clear from the figure that temperature does not have a profound affect on the inter-chip Euclidean distances. Unlike the inter-chip distances, the noise distances exhibit a noticeable effect of temperature, as we expect. The black “stair” line shows the 0°C case, and the noise distances are centered about 0.9Ω . However, at the nominal case of 25°C, drawn with a red stair line, we observe a smaller tail to the right-hand side, and the histogram has more of a Gaussian bell curve. Moving up to 50°C, drawn with a green line, the mean is greater and the standard deviation looks smaller. At 75°C, drawn with a blue stair line, the mean is less but the noise distance are essentially the same as they are at 50°C. Similar observations can be made for the V_{drop} analysis, shown in Figure 6.16.

The thresholds on Euclidean distance for each temperature are also shown with circles on Figures 6.15 and 6.16, and are drawn with colors corresponding to the stair line colors. The thresholds are also reported in Table 6.5¹. For the R_{eq} analysis, the threshold has a minimum at 75°C and appears to be monotonically-decreasing with increasing temperature. For the V_{drop} analysis, the thresholds are again a monotonically-decreasing series with increasing temperature. Table 6.5 also reports the probability of aliasing for the various analyses performed. For the analog R_{eq} PUF, room temperature represents the minimum probability of aliasing, at 3.7×10^{-9} .

¹These estimates are slightly more pessimistic than our previous estimates because 99.7300204% was used rather than 99.7% to find the noise threshold.

Chapter 6. Temperature Effects

	Temp.	Threshold	Probability of Aliasing
R_{eq}	0 C	0.16762	5.6018×10^{-9}
	25 C	0.14119	3.7317×10^{-9}
	50 C	0.13100	1.6358×10^{-8}
	75 C	0.12825	1.4607×10^{-7}
V_{drop}	0 C	4.4313×10^{-4}	3.4923×10^{-9}
	25 C	3.5254×10^{-4}	8.5433×10^{-10}
	50 C	3.0688×10^{-4}	3.9388×10^{-10}
	75 C	2.8414×10^{-4}	9.3278×10^{-10}

Table 6.5: Results of probability analysis for various combinations of analog R_{eq} and V_{drop} and different temperature points.

For the analog V_{drop} PUF, the minimum probability of aliasing is found at 50°C to be 3.91×10^{-10} . This analysis addresses the performance of the PUFs in thermal equilibrium at different temperatures. However, this does not address the issue of the temperature changing between measurements of the same chip or comparing two chips at different temperatures.

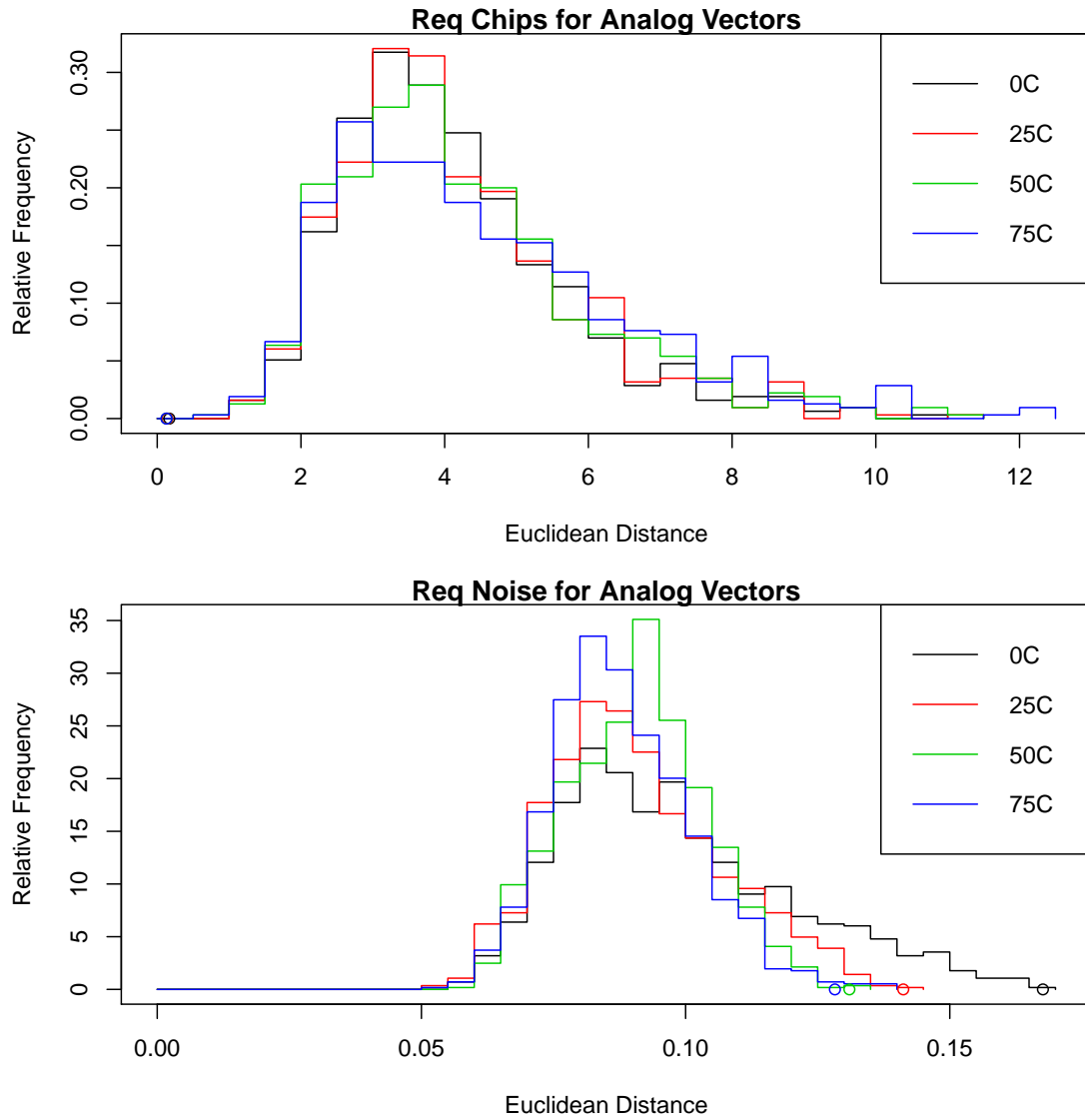


Figure 6.15: Histograms of R_{eq} vector inter- and intra-chip Euclidean distances

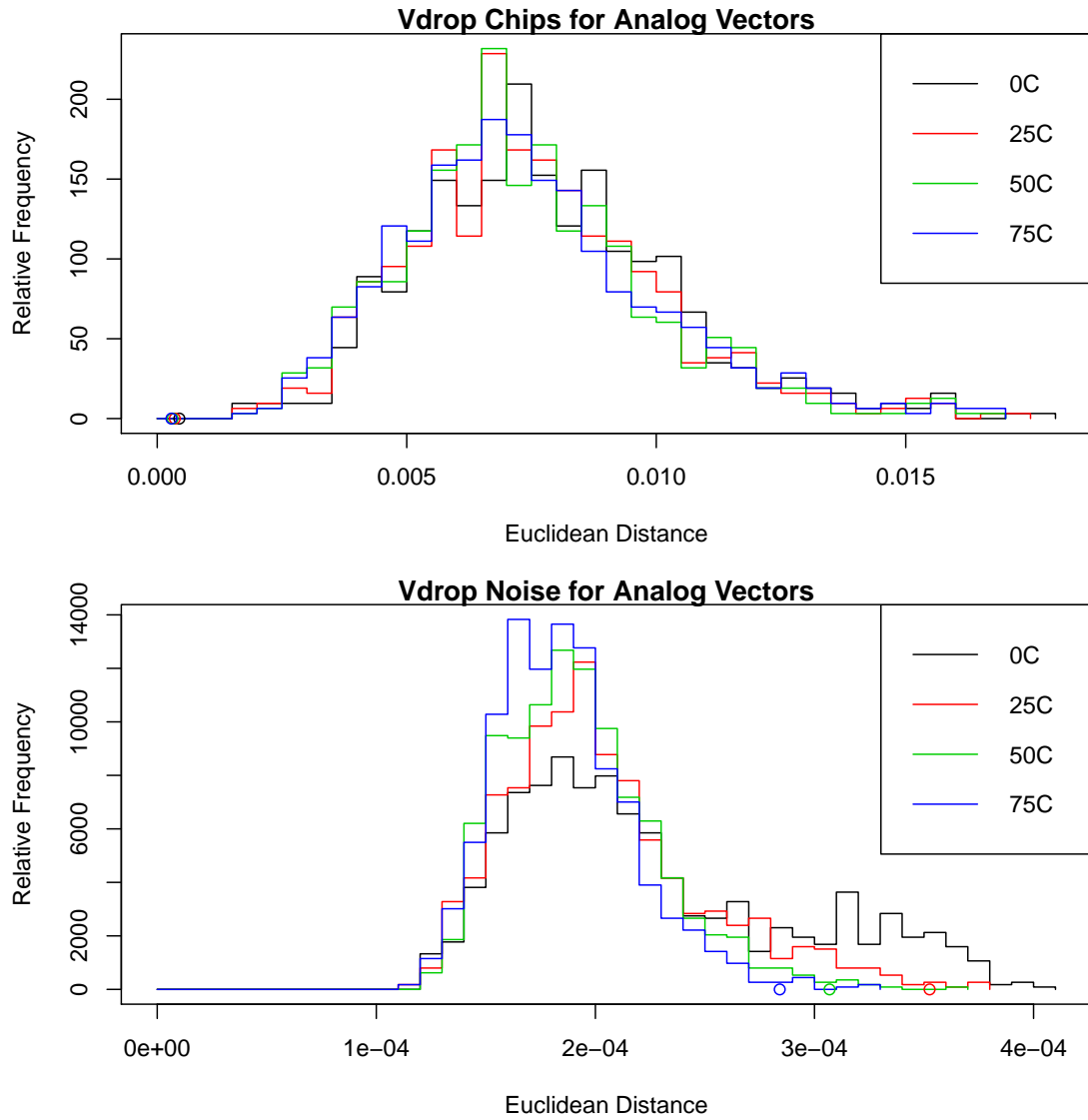


Figure 6.16: Histograms of V_{drop} vector inter- and intra-chip Euclidean distances

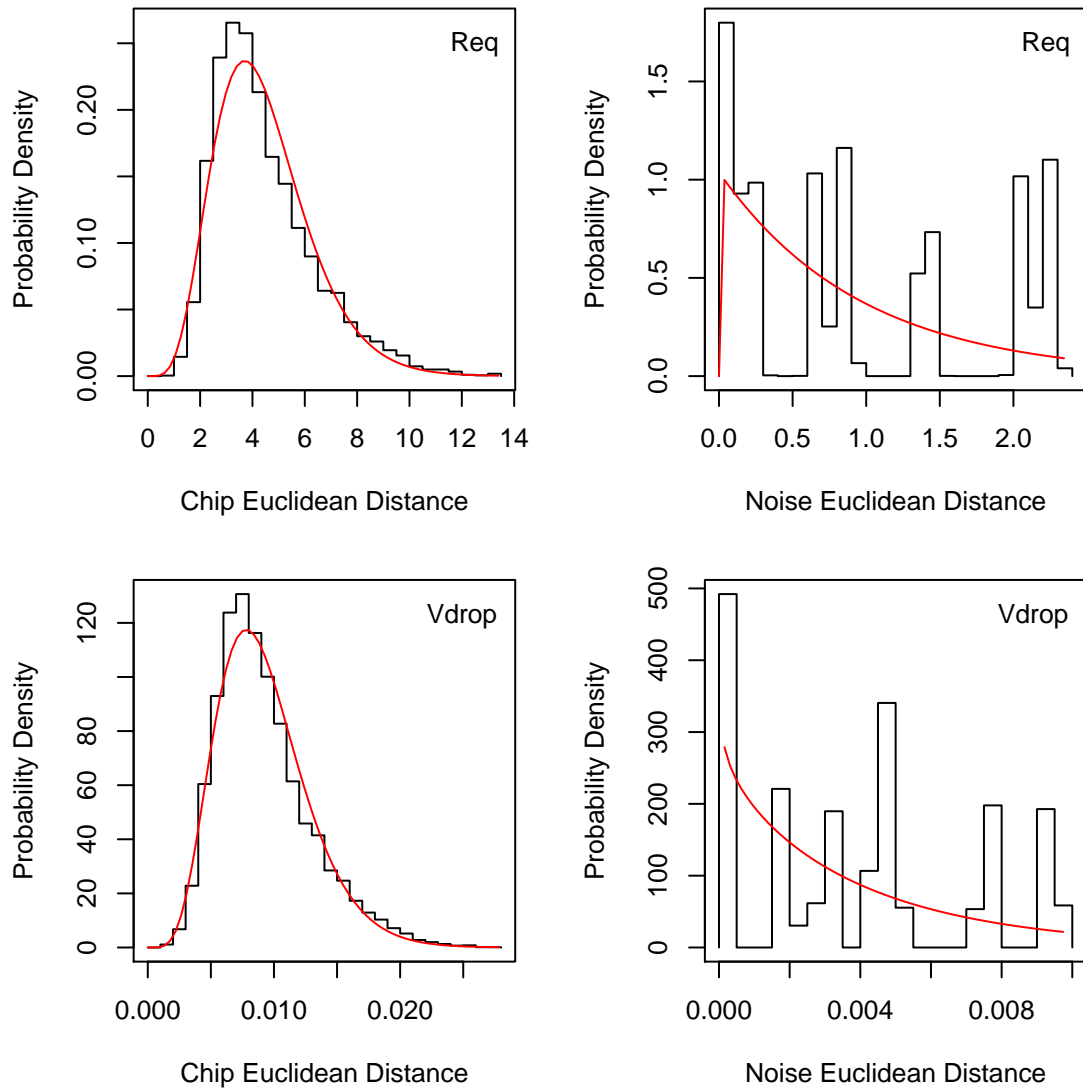


Figure 6.17: Histograms of aggregate (0 - 75°C) V_{drop} and R_{eq} vector inter- and intra-chip Euclidean distances

Chapter 6. Temperature Effects

To evaluate the performance of the PUF when the temperature cannot be controlled, we can “aggregate” the chip and noise samples at nominal temperature (25°C) with the other temperature points. This represents a worst-case scenario for thermally-induced noise. We do this by simply adding the measurements made at the other temperature points (0°C, 50°C and 75°C) to the nominal sample. The effect that aggregating the different temperature points should have is to make noise distances larger on average, and (possibly) make chip distances smaller, which will effectively decrease the Signal to Noise Ratio (SNR). In the case of inter-chip distances, we cannot simply aggregate the chips at the other temperature points. We have to take special care when choosing the pairwise combinations. The problem is that if we just use all $\binom{n}{2}$ combinations, we will end up with a few comparisons of a chip against itself at another temperature point, which is unreasonable. Therefore, the pairs of inter-chip distances are chosen such that this case is excluded. Then, we found that the chip distances in the temperature-aggregated sample were approximately the same as the nominal sample. The first row of Figure 6.17 show the aggregate chip distance histograms for the R_{eq} and V_{drop} .

Another problem arises when we aggregate the noise sample for the analog signatures. When we compute the pair-wise noise distances from the aggregated sample, we observe four distinct modes in the corresponding histogram. These histograms are shown in the second row of Figure 6.17. These are a result of comparing measurements of the chip at varying temperature *differences*. Measurements taken at the same temperature tend to yield one noise distance (which is the same as observed previously), measurements made at two temperatures that are 25°C apart yield a higher distance, measurements made at two temperatures 50°C apart yield an even higher distance and so on. The four distinct modes are readily observed: same temperature, 25°C difference, 50°C difference and 75°C difference. It is unreasonable to use the fits shown on the histogram from this analysis in order to measure the probability of aliasing. In order to estimate a worst-case scenario, we could choose

Chapter 6. Temperature Effects

the noise distances so that we only compare measurements made between the two temperature extremes. This would effectively single out the largest mode in the noise histogram, which is centered at 2.25Ω or 9mV , for the R_{eq} and V_{drop} , respectively. The noise distances might then have a Gamma distribution, however it is clear from the figures that the noise (upper-limit) threshold would be well into the range of most of the inter-chip distances, and therefore would not yield a reasonable probability of aliasing (e.g., 50% or 80% could be aliases). Therefore, when temperature cannot be controlled, it is not feasible to use a threshold on the Euclidean distance between two chips to determine if they are identical.

6.3.3 Vector Angles

Instead of computing the Euclidean distance between two vectors of analog quantities that were taken at different temperatures as we did in the previous section, one possible solution is to use a different metric to compare how different two vectors are. The effect of temperature on these analog quantities is a scaling effect that changes the magnitude of the voltage or resistance that we measure proportional to its magnitude at room temperature. It also tends to affect each of the quantities in a similar manner. For example, we have seen that an increase in temperature leads to a decrease in the voltage drop. In other words, changes in temperature tend to affect the vector's length, and perhaps not the vector's angle. Therefore, if we use angles between the vectors as a means to measure the distance, then the angles between vectors should be preserved over temperatures. For example, consider a three-dimensional vector $(1, 0.5, 1)$. If temperature drift were to add 10% to each term of this vector, then it would be $(1.1, 0.55, 1.1)$. If we compute the angle between this vector at both temperature points, we would get exactly zero.

Chapter 6. Temperature Effects

		Threshold	Probability of Aliasing
Nominal	R _{eq}	0.002476	4.5541×10^{-13}
	V _{drop}	0.002341	1.7156×10^{-11}
Aggregate	R _{eq}	0.02193	4.0587×10^{-3}
	V _{drop}	0.01776	2.3333×10^{-3}

Table 6.6: Results of probability analysis for analog R_{eq} and V_{drop}, in the nominal and aggregate case, using vector angles

The formula I use to compute the angle between two n -dimensional vectors is

$$\theta = \arccos \left(\frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|} \right), \quad (6.14)$$

which is given by

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos \theta. \quad (6.15)$$

Essentially, I compute the dot product and then divide by both vector norms. I pass the result to the inverse cosine function (\arccos) to obtain the angle between the vectors in radians. This applies to vectors of any dimension. Therefore, we can use this as an alternative to using Euclidean distances to measure the difference between chips when temperature cannot be controlled.

From Table 6.6, it is apparent that when we use the vector angles, the probability of aliasing actually decreases (improves) over the metrics using the Euclidean distance shown in Table 6.5. This is because there is a larger dichotomy between the chip and noise distances. Figure 6.18 shows the histograms of both chip and noise pairwise distances when we use the vector angles of the 192-term analog vectors, for the nominal case of 25°C. Figure 6.19 show the corresponding histograms when we *aggregate* the angles of the vectors for temperatures between 0°C and 75°C. The corresponding thresholds and probabilities of aliasing are reported in Table 6.6. From the table, it is clear that aggregating these vector angles results in a probability of

Chapter 6. Temperature Effects

aliasing that is several orders of magnitude smaller than the nominal case. This indicates that the effect of temperature on each of the terms in the vector is not exactly the same within a given chip. Otherwise, the probabilities would be approximately the same, as we will see later when we aggregate the digital signatures. Nevertheless, these probabilities of aliasing are 1 in 246 and 1 in 428, for the R_{eq} and V_{drop} PUFs, respectively. In summary, the method provides a worst-case analysis that is reasonable for the analog vectors.

Another way of assessing this measure of difference between two vectors is to study the histograms in Figure 6.19. Although the probability of aliasing estimates are reasonable for both the nominal and the aggregate vectors, there is still a multimodal distribution of the noise distances. This can be seen in the second row of Figure 6.19. The different modes are not as distinct as they are in Figure 6.17, where we used the Euclidean distance as the measure of difference between the vectors. Instead, they are more bunched together, indicating that this metric is resisting the changes between different temperature points, but not eliminating them.

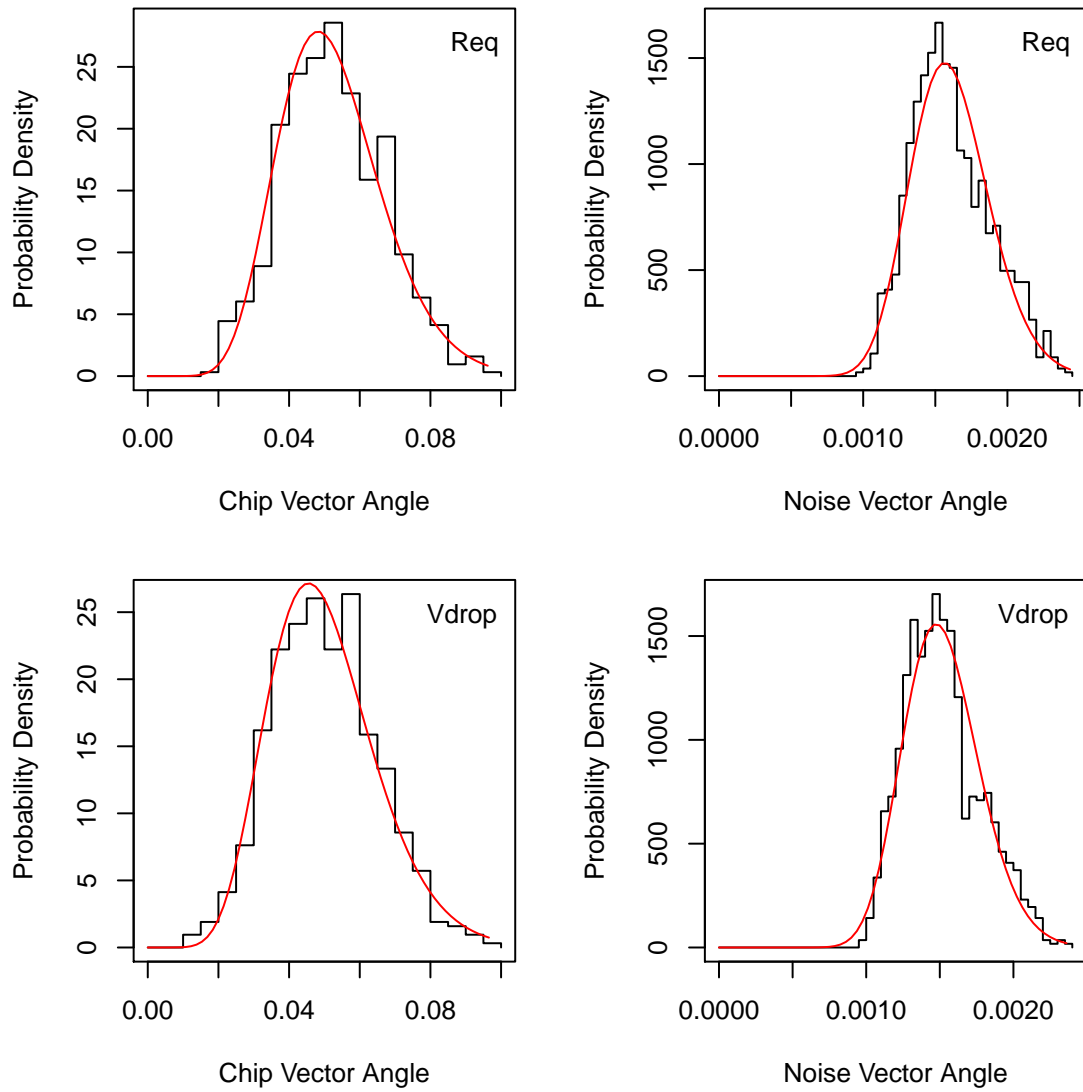


Figure 6.18: Histograms of nominal (25°C) V_{drop} and R_{eq} vector inter- and intra-chip vector angles

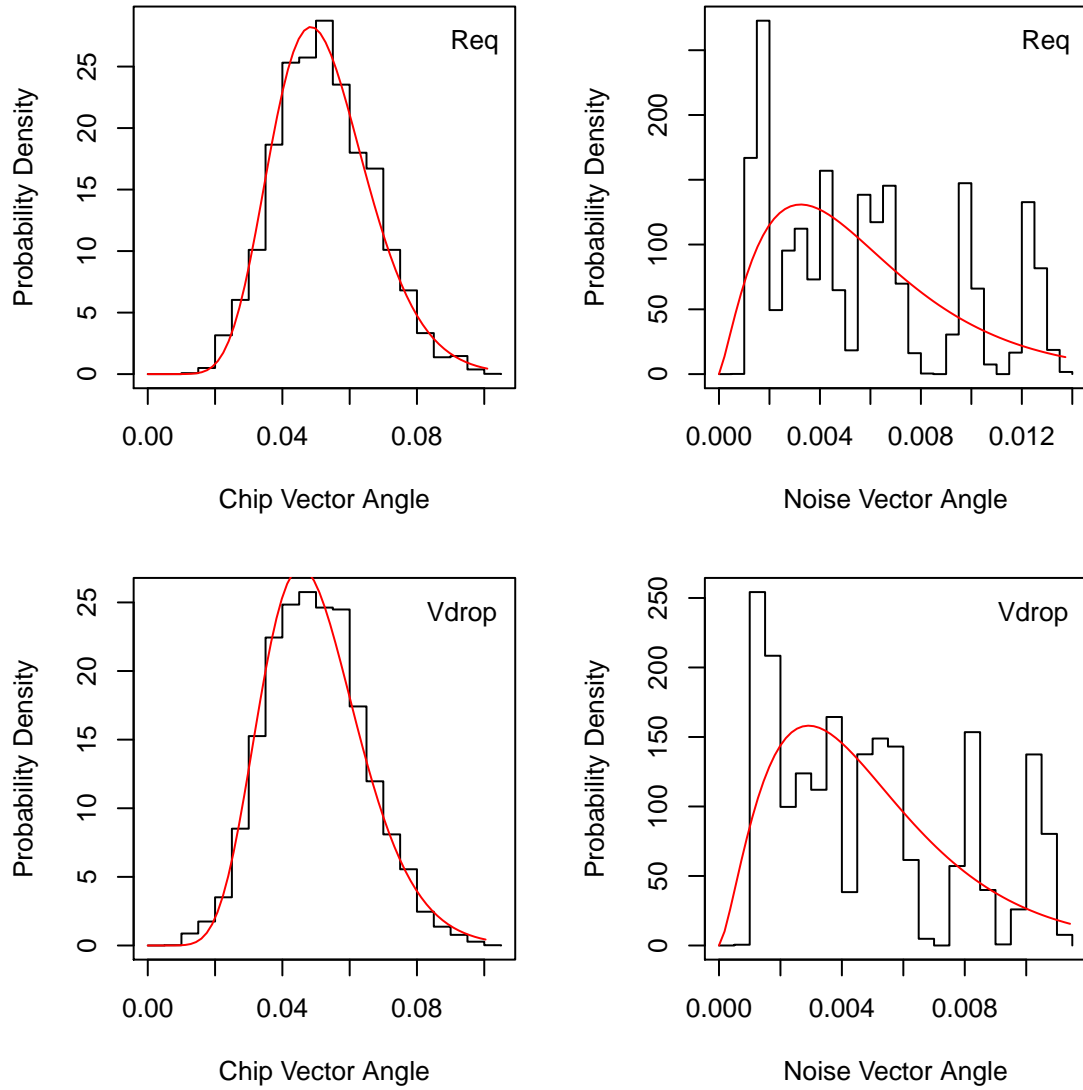


Figure 6.19: Histograms of aggregate (0 - 75°C) V_{drop} and R_{eq} vector inter- and intra-chip vector angles.

6.4 Effects on Digital R_{eq} and V_{drop} PUFs

In Chapter 5, we explained how to generate 30- and 255-bit binary response vectors from the analog data that we measure with our instrumentation. Essentially, we construct pairs of the analog values and generate a 1 or a 0 depending on their relative magnitude. As mentioned, this is an operation which can be implemented on-chip using an operational amplifier, and it therefor models a realistic use scenario. We consider two methods for constructing a set of pairs of observe locations which we compare to produce a 1 or a 0. We call these methods “core” and “all”, and represent the pessimistic and optimistic choices, respectively. In this section, I revisit those results with the new data that was taken under temperature-controlled conditions.

6.4.1 Function of Temperature

The first analysis we performed helped us understand how the dispersion of the analog values affected the stability of the bits. In other words, we wanted to understand the relationship between the proximity of the analog values that are compared to produce a bit and the likelihood of that bit to change. The bit would change if one analog value (a resistance or voltage) became larger than the other when it was previously smaller. We expect that a bit flip is most likely when the analog values are close to one another.

The bits that we use to construct the 30- and 255-bit binary vectors are self-relative. If temperature affects the analog values in a common way, then we do not expect the bits to change between different temperatures. For example, if resistances R_a and R_b are such that $R_a > R_b$ at 0°C , then we expect that the rate of change of both resistances as a function of temperature is approximately the same and should not cause $R_b \leq R_a$ at 25°C or even 75°C . We observed that the *stability* of bits was affected by temperature. In other words, the bits we generate are more likely to flip

Chapter 6. Temperature Effects

at higher temperatures. Recall that previously, at room temperature, we observed no bit flips in the case of the *core* analysis, and we observed bit flips under only the *all* analysis. The case was the same for the *core* analysis with the new data for both 0°C and 25°C. However, there were some bits that flipped at 50 and 75°C. These metrics are also reported in Table 6.7.

We can also visualize the dispersion of the analog values and how their proximity affects the stability of corresponding bits with histograms. In Figure 6.20, four histograms are plotted for the *core* analysis. On the left-hand side are the histograms for bits that flipped, and on the right-hand side are the histograms for bits that were stable. The histograms on the top row correspond to the R_{eq} PUF and those on the bottom row correspond to the V_{drop} PUF. In each histogram, four trends are plotted corresponding to the four different temperature points. The number of samples used to create the histogram is also indicated in parenthesis in the legend. It is clear that the number of unstable bits for the *core* analysis grows as temperature increases, but for the large majority of cases, the distances are large enough to prevent any bit flipping (50 out of 2000 cases). Furthermore, the largest distance that tends to cause a bit to flip is around 0.4% for both the R_{eq} and V_{drop} PUFs, where most of the distances are at 1, 3, or 5%. For each temperature point, I have also plotted a dotted vertical line which represents the measurement noise floor. This shows that the majority of analog values that cause bits to flip are closer to each other than the measurement noise floor. In Figure 6.21, the same analysis as above is presented for the *all* analysis. The trends are more well-defined since there are many more bit comparisons involved (approximately 18,100), but the results are similar. Bits start to flip when the relative difference in the analog quantities gets smaller than the measurement noise. It can also be seen that as the percent difference gets smaller, the bit is more and more likely to flip. Finally, any integrated architecture will have a similar problem with comparing values that are too close to one another, and therefore this models the performance of an integrated architecture.

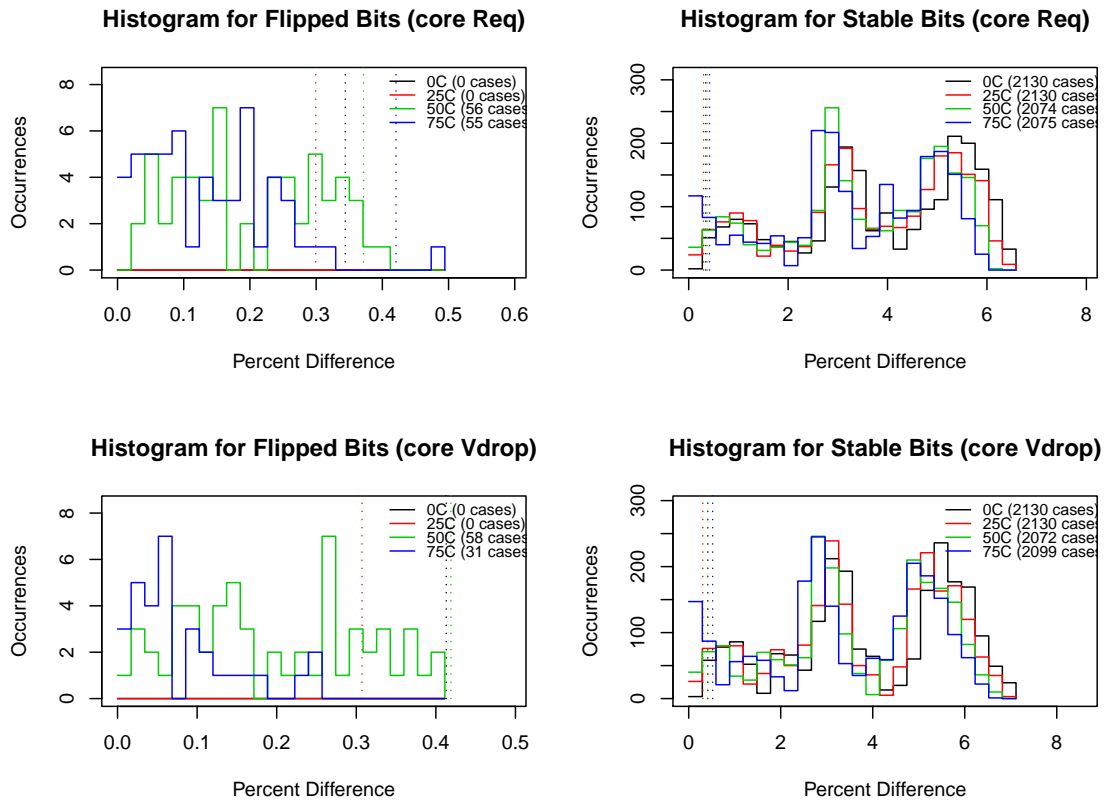


Figure 6.20: Histograms of Percent Differences ($100 \times (x - y)/x$) for *core* bits that flipped (left) and bits that were stable (right) for R_{eq} (top) and V_{drop} (bottom). Each plot includes the four temperature points between 0 and 75°C. The vertical lines indicate the corresponding measurement noise floor.

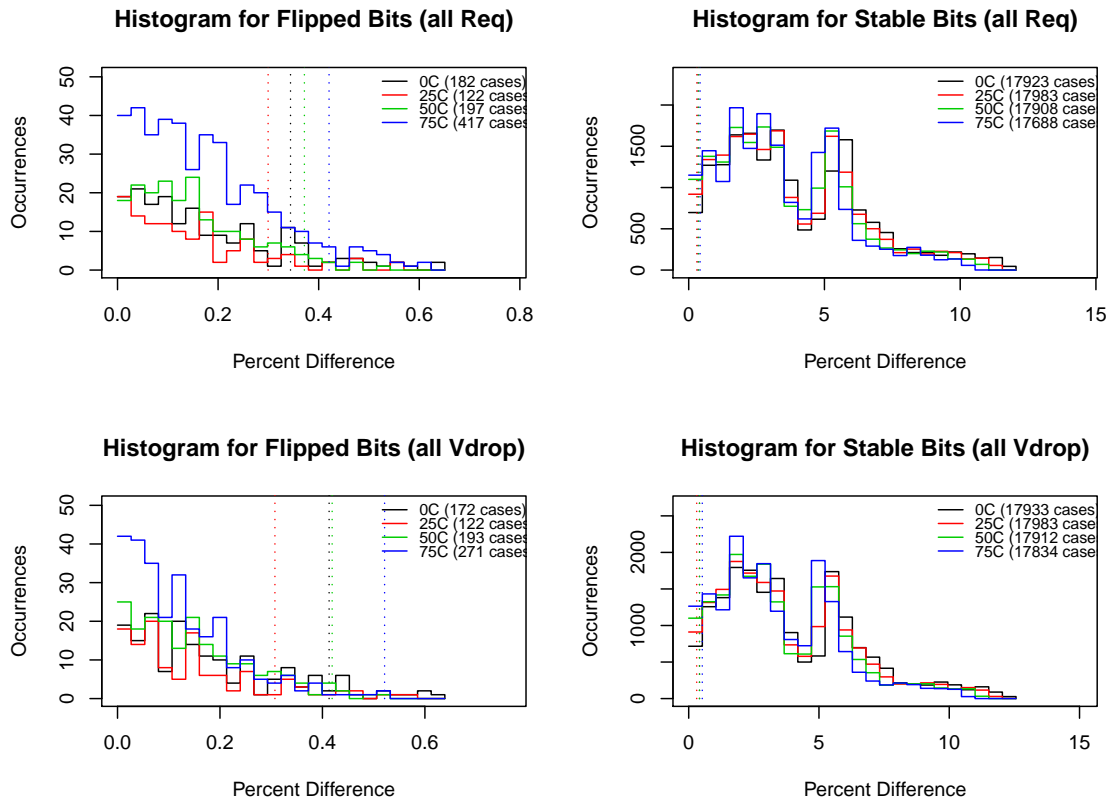


Figure 6.21: Histograms of Percent Differences ($100 \times (x - y)/x$) for *all* bits that flipped (left) and bits that were stable (right) for R_{eq} (top) and V_{drop} (bottom). Each plot includes the four temperature points between 0 and 75°C. The vertical lines indicate the corresponding measurement noise floor.

6.4.2 Aliasing Probability

As we learned in Section 6.4.1, the digital signatures are self-relative and therefore are resistant to changes in temperature that affect the parameters we measure in a common mode. However, we also saw that the uncertainty of the bits increases as temperature rises. In this section, I apply a probability analysis similar to that of Section 6.3, but with a few key differences. First, since the elements in the vectors are now bits, we use the Hamming distance rather than the Euclidean distance. Second, because the Hamming distances are now discrete, I will apply the Negative Binomial Distribution in order to fit the histograms of distances. Recall that the Euclidean distances were continuous and that we used the Gamma distribution to fit them.

In Figure 6.22, there are two histograms of Hamming distances taken from the chip and noise samples for the R_{eq} PUF, using the “all” construction. The black stair line is for the nominal temperature of 25°C, and the red line is for the aggregate (0 - 75°C) analysis. The top histograms are for inter-chip distances and the bottom histograms are for the noise (intra-chip) distances. As we reported earlier, the average chip distance is approximately 121, half the code length of 255 bits. As we expect, nearly 40% of the noise distances are zero in the nominal case. About 30% are 1-bit differences, and the remaining 30% are two bits or greater. Next, we “aggregate” the nominal sample with other temperature points as we did before in Section 6.3.2. The aggregate trends are drawn with red stair lines in Figure 6.22. In the case of the noise aggregate sample, adding the vectors from these other temperature points just exacerbates the noise that we see. This is reflected in the histogram, which shows that the average noise distance has increased from about 2.0 to about 3.8 bits. Finally, Figure 6.22 shows the corresponding histograms for the V_{drop} PUF. The results are almost identical.

As with the analog analysis, the threshold chosen for each noise histogram is

Chapter 6. Temperature Effects

			Threshold	Probability of Aliasing	Mean Chip	Mean Noise
Core (30-bit)	Nominal (25°C)	R _{eq}	1	2.64×10^{-3}	14.3	0
		V _{drop}	1	3.40×10^{-3}	14.3	0
	Aggregate (0-75°C)	R _{eq}	3.2	1.87×10^{-2}	14.3	0.554
		V _{drop}	3.2	2.24×10^{-2}	14.3	0.554
All (255-bit)	Nominal (25°C)	R _{eq}	7.5	2.07×10^{-7}	121	2.00
		V _{drop}	7.5	2.98×10^{-7}	121	2.00
	Aggregate (0-75°C)	R _{eq}	13	6.18×10^{-6}	120.7	3.78
		V _{drop}	13	8.50×10^{-6}	120.7	3.78

Table 6.7: Results of probability analysis of digital PUF signatures for various combinations of R_{eq} and V_{drop}, core and all, and nominal temperature (25°C) and aggregate (worst-case) over 0°C and 75°C

indicated with circles \circ on both plots, in the appropriate color. The thresholds are also reported in Table 6.7. The probability of aliasing for each analysis is the next column in the table. It is clear from Table 6.7 that the R_{eq} PUF fairs slightly better than the V_{drop} PUF, as was the case with the analysis of the analog values. The worst-case analysis degrades the probability of aliasing by an order of magnitude (e.g., 2×10^{-7} down to 6×10^{-6} for R_{eq} all). This is an excellent result that indicates that although temperature does have an effect, the digital PUFs are still viable when temperature cannot be controlled.

Another point of reference that can be used in conjunction with the probability of aliasing is the theoretical upper bound on the number of distinct signatures, which is 2^{-N} , where N is the number of bits in the signature. For our 30-bit core and 255-bit all signatures, these bounds are 9.31×10^{-10} and 1.73×10^{-77} , respectively. However, the number of independent bits and the number of independent responses is not ideal in practice. From Table 6.7, we can compare the probability of aliasing with these upper bounds. For the core analysis, we are seeing approximately 1×10^{-3} out of 1×10^{-10} . For the all analysis, we are seeing approximately 1×10^{-7} out of 1×10^{-77} .

6.4.3 Hamming Distances

Many publications comment on the average of the inter-chip and noise (intra-chip) Hamming distances, and using the separation between the two to qualify the power of the PUF to distinguish chips. We have given estimates of the probability of aliasing whenever possible, which is more quantitative than the mean distances alone, but it is important to use the same metrics that have become standard. In this section, I comment further on the distances computed for Figure 6.22.

The average inter-chip Hamming distance, which is simply the average number of bits between two chips, is ideally 50%. You can easily convince yourself of this by considering an average distance that is greater than 50%, such as 100%. In that case, all the bits would be different between two chips and you would quickly find that there are only two signatures with one the logical complement of the other. In Chapter 5, we reported that the average inter-chip Hamming distances was 48.0% and 48.5% for the core and all constructions.

The mean chip and noise distances are reported in the last two columns of Table 6.7, since these are metrics used by others [50]. The mean can also be reported as a percentage of bits, which can be found by dividing the mean by the number of bits in the signature—30 bits for core and 255 bits for all. These metrics are reported in percent bits for convenience in Table 6.8. Recall that the ideal inter-chip Hamming distance is 50% and the ideal noise (intra-chip) distance is 0%. For example, the authors of [50] report that these metrics are 46.15% and 0.48%, respectively, for their PUF. The “core” and “all” analyses were essentially identical under these metrics both at nominal and aggregate temperature. The mean chip distances in percent are 47.6% and 47.5%, respectively, and the mean noise distances in percent are 0% and 0.78%, respectively. When we aggregate temperature, these metrics decay somewhat. In the core analysis, the chip mean remains essentially the same, but the

Chapter 6. Temperature Effects

		Mean Chip	Mean Noise
Core	Nominal	47.6%	0%
	Aggregate	47.6%	1.85%
All	Nominal	47.5%	0.784%
	Aggregate	47.3%	1.48%

Table 6.8: Mean inter-chip and noise Hamming distances, reported in percent bits

noise distance increases from 0% to 1.8%. In the all analysis, the mean chip distance decreases slightly to 47.3% and the noise distance increases to 1.5%.

For completeness, Figure 6.23 are the histograms corresponding to Figure 6.22, for the “core” binary signature construction. The trends are not as readily seen, however, since there are very few bits in the signature. Recall there are only 30 bits per chip, and up to 25 of these bits are dependent upon the first 5. In the first row of Figure 6.23, it can be seen that the number of chip distances that have a zero Hamming distance (far-left bin) is non-zero. In other words, some chip signatures under the “core” construction were exactly the same. Specifically, 1.7% and 2.1% of the R_{eq} and V_{drop} distances, respectively, have a zero distance. Upon inspection, 5 and 8 of our chips out of 36 are aliases under the 30-bit core R_{eq} and V_{drop} construction, respectively. This is expected, however, since we are trying to distinguish 36 chips, and in general we have at least 5 bits in the signature that are truly independent. Therefore, the lower bound on the number of distinct signatures is $2^5 = 32$, which is 4 fewer than the number of chips we considered. To resolve this issue, we could increase N from 6 to something much greater. For example, if we have 100 SMCs, then the number of bits would then be $100 \times 99 = 9900$ (see Section 5.5), and presumably at least 100 independent bits.

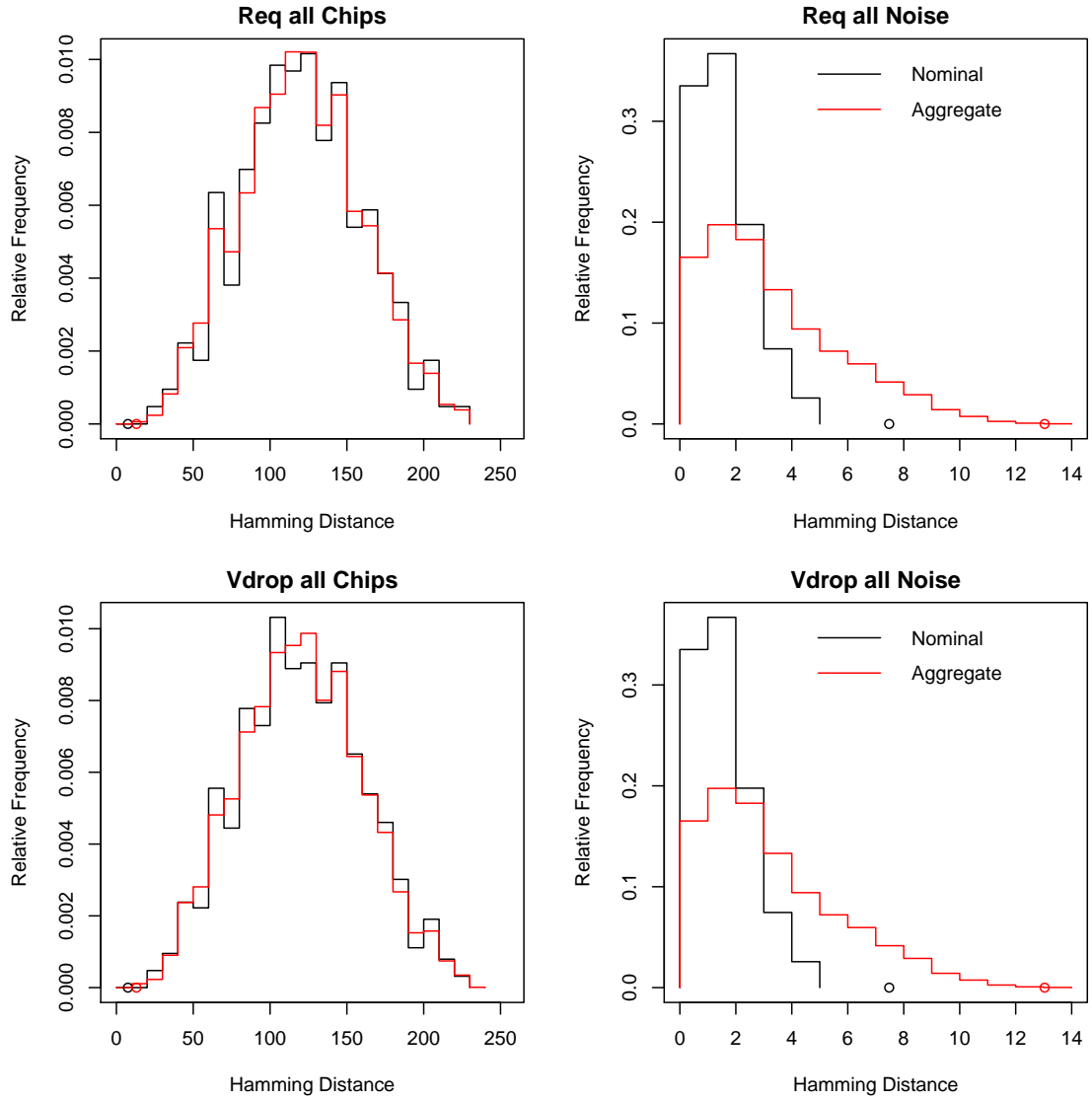


Figure 6.22: Histograms of Hamming distances for chip and noise samples for the V_{drop} and R_{eq} PUFs, using the “all” construction

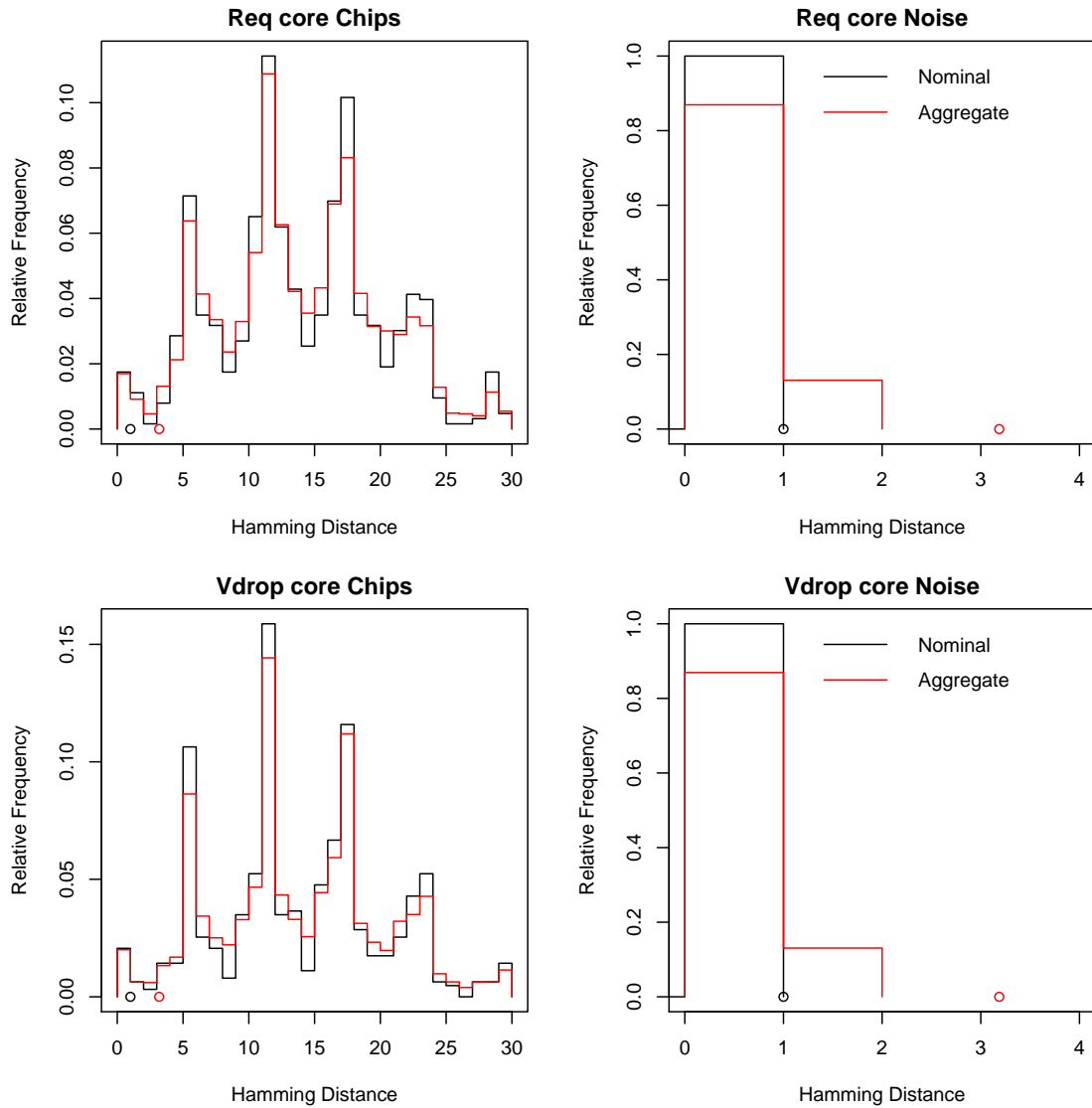


Figure 6.23: Histograms of Hamming distances for chip and noise samples for the V_{drop} and R_{eq} PUFs, using the “core” construction

6.4.4 Bit Probabilities

In this section, I report the single-bit probability and bit flip probability metrics, as defined by the authors of [41]. We commented on these metrics in Chapter 5 (also DAC2010) for the R_{eq} PUF. The single-bit probability is a metric of the quality of each bit, akin to the fairness of a coin, and is ideally $1/2 = 50\%$. If each bit is like a fair coin, then the maximum randomness is achieved in general. We reported 50% for both the core and all analyses in Chapter 5. The bit flip probability, the probability of a bit to flip between measurements, is ideally 0%. This characterizes the reliability or repeatability of that bit of the PUF. We reported 0%² and 0.644% for the core and all analyses, respectively.

The single-bit and bit-flip probabilities are reported in Tables 6.9 and 6.10. From Table 6.9, all of the single-bit probabilities are less than 4% away from the ideal 50%. There is no definite trend of the single-bit probability with temperature, so it is robust to changes in temperature. Finally, the “all” signatures all have a single-bit probability less than 50% and the “core” signatures all have a single-bit probability that is greater than 50%. Moving on to the bit flip probabilities (Table 6.10), it can easily be seen that they are all small—less than 3%. As was the case with our experiments without temperature control (for DAC2010), at 0°C and at 25°C, the “core” signatures had no bit flips from which to take statistics. However, for 50°C and 75°C, we see that there are up to 2.7% bits that flip on average. As we have seen before, the core analysis doesn’t give us enough samples to observe definite trends, but there is a trend from the all analysis. For both the R_{eq} and V_{drop} “all” signatures, there is a minimum bit flip probability at room temperature (25°C), and show an increasing trend at both colder and hotter temperatures.

²There was not enough data to find a single bit flip in this case.

Temp (C)	All		Core	
	R_{eq}	V_{drop}	R_{eq}	V_{drop}
0	47.57081	47.29847	52.87037	52.87037
25	46.97168	46.73203	52.31481	52.50000
50	46.75381	46.48148	51.75926	51.94444
75	46.42702	46.22004	52.03704	52.12963

Table 6.9: Single-bit probabilities for binary signatures (ideally 50%)

Temp (C)	All		Core	
	R_{eq}	V_{drop}	R_{eq}	V_{drop}
0	0.9912854	0.9368192	0.000000	0.000000
25	0.6644880	0.6644880	0.000000	0.000000
50	1.0729847	1.0511983	2.592593	2.685185
75	2.2712418	1.4760349	2.546296	1.435185

Table 6.10: Bit flip probabilities for binary signatures (ideally 0%)

6.5 Observe Net Leakage Current

One of the realities that we have abstracted away in the discussions before this point is that the voltage observe transistors suffer from sub-threshold leakage. For example, this means that if we want to enable one out of six observe transistors, that five are leaking current into the grid that constitutes the voltage sense wire. In this section, I describe how this leakage current affects the accuracy of our measurements of the voltage drop, which in turn impacts the equivalent resistant that we compute.

In fact, we have 4,000 observe transistors in our chips. We found that this source-to-drain leakage current significantly affects the observe voltage that we record. Figure 6.24 shows a circuit diagram that represents a voltage observe measurement. The 900mV power supply is represented as a voltage source V_{supply} and the shorting transistors are represented by a current source I_{short} . R_{eq} in the figure is the true

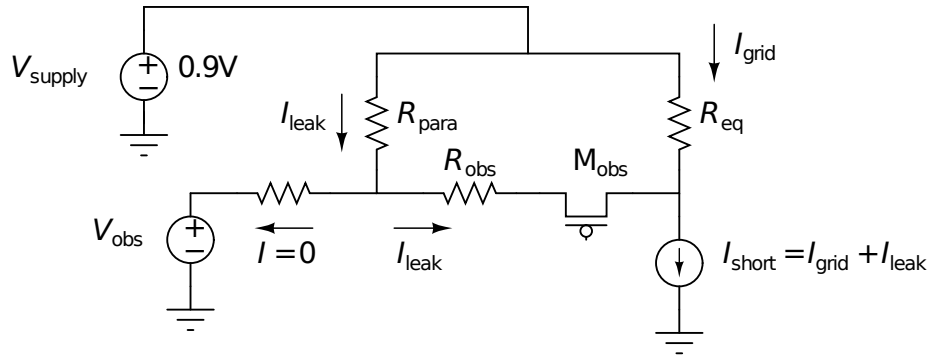


Figure 6.24: The observe transistor physical circuit view

equivalent resistance of the power grid, which we cannot measure exactly due to this problem. The current I_{short} , represented by a current source, is drawn through this resistance. At the same point that the shorting current is drawn, the voltage observe transistor, labeled M_{obs} is attached. On the other end of the transistor, the voltage sense wire net is represented. The voltage sense wire net is a grid that connects all 4,000 observe transistors to the voltage observe pin. To model this with an equivalent circuit, we insert a resistance both before and after the point where the leakage current I_{leak} enters the net. The voltmeter instrument draws practically no current, so the resistance on the left-hand side has practically no voltage drop, and therefore no effect on the observe voltage. However, the leakage current injected by the other observe transistors, represented in Figure 6.24 by R_{para} , does flow over the resistor on the right-hand side, labelled R_{obs} and through the voltage sense transistor. Since the current is flowing from left to right through the resistor R_{obs} and the observe transistor, the voltage decreases along the same path. This means that the voltage we measure with the voltmeter represented as V_{obs} is elevated above the ideal level. This current I_{leak} finally becomes a component of I_{short} , which is the global current that we measure.

From Figure 6.24, we can write the following equations using KVL and Kirchhoff's

Chapter 6. Temperature Effects

Current Law (KCL).

$$V_{\text{supply}} = I_{\text{grid}}R_{\text{eq}} - R_{\text{obs}}I_{\text{leak}} + V_{\text{obs}} \quad (6.16)$$

$$I_{\text{short}} = I_{\text{grid}} + I_{\text{leak}} \quad (6.17)$$

The terms I_{grid} , R_{eq} and R_{obs} are all unknown and M_{obs} is assumed to be an ideal short for now. In a separate experiment, we can drive the V_{obs} voltage instead of using it as a volt meter. We can then simulate the voltage drop that the shorting current and observe transistor will create across the observe net and measure the leakage current I_{leak} that is pulled across the 4,000 transistors. However, this is only a first-order approximation because the voltage drop we use to stimulate the observe grid is based on the V_{drop} measurement as presented previously, which we know is not accurate. From Equation (6.17), $I_{\text{grid}} = I_{\text{short}} - I_{\text{leak}}$. Then I can eliminate the I_{grid} term from Equation (6.16), which we cannot measure, and I have

$$V_{\text{supply}} = (I_{\text{short}} - I_{\text{leak}})R_{\text{eq}} - R_{\text{obs}}I_{\text{leak}} + V_{\text{obs}}. \quad (6.18)$$

Now, I have one equation and two unknowns, R_{eq} and R_{obs} . To overcome this, I can write Equation (6.18) for multiple temperature points, and use the appropriate known values for each temperature. I model unknown resistances as a function of temperature using Equation (6.1), repeated here for convenience,

$$R = R_{\text{ref}}[1 + \alpha(T - T_{\text{ref}})]. \quad (6.19)$$

I use the $\alpha = 0.001394$, from the metal characterization experiment in Section 6.1.1, and $T_{\text{ref}} = 25^\circ\text{C}$. Then each unknown resistance is a function of the reference resistance R_{ref} and the independent variable temperature T . After substituting Equation (6.19) into Equation (6.18) for R_{eq} and R_{obs} , I can write one equations at 25°C and another at 50°C . I then solve for $R_{\text{eq, ref}}$ and $R_{\text{obs, ref}}$.

Based on preliminary measurements, $R_{\text{eq, ref}} = 10.62\Omega$ and $R_{\text{obs, ref}} = 1640\Omega$. I can then compute the voltage drop across R_{obs} which I call $V_{\text{obs, para}}$ and adds to our

Chapter 6. Temperature Effects

measurement of V_{obs} . From the circuit diagram, $V_{\text{obs, para}} = I_{\text{leak}}R_{\text{obs}}$. Therefore, at 25°C and 50°C, $V_{\text{obs, para}} = 825\mu\text{V}$ and 1.407mV, respectively. The true voltage drop across the equivalent resistance of the power grid is then

$$V_{\text{drop, corrected}} = V_{\text{supply}} - (V_{\text{obs}} - V_{\text{obs, para}}) \quad (6.20)$$

$$= V_{\text{supply}} - V_{\text{obs}} + V_{\text{para}}. \quad (6.21)$$

The corrected equivalent resistance is then

$$R_{\text{eq, corrected}} = \frac{V_{\text{supply}} - V_{\text{obs}} + V_{\text{para}}}{I_{\text{short}} - I_{\text{leak}}}. \quad (6.22)$$

Using this information, I created the trends in Figure 6.25. The first curve in the figure represents one of the equivalent resistances that we have shown so far. Then, the corrected equivalent resistance is plotted in red. It is clear from this graph that our first-approximation of this non-ideal leakage current into the observe net is substantial enough to cause the NTC to become a Positive Temperature Coefficient (PTC). This effect is an unfortunate reality and it inversely affects our measurement of the voltage drop and equivalent resistance. However, it fortunately explains why the equivalent resistances are not behaving like copper, a metal with a PTC.

There are several solutions to this problem. For example, the observe transistors could be sized to reduce the amount of leakage current. It is known that scaling the length of a transistor by a factor of 2 can reduce the leakage current by a factor of 10 [49]. Alternatively, we're only proposing using 6 SMCs, not 4,000. If fewer observe transistors are present, then the total leakage current and corresponding impact on the measured voltage would be more ideal.

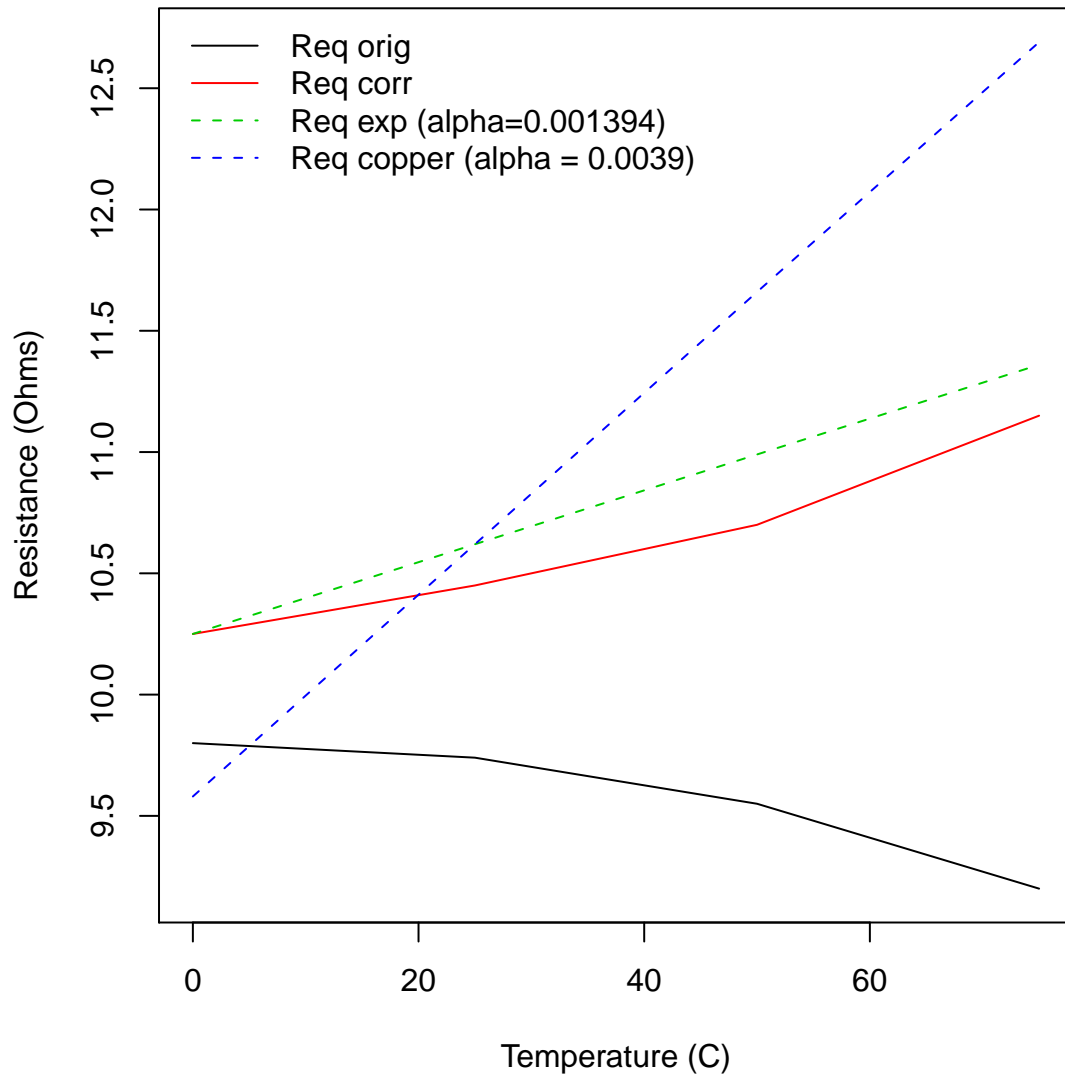


Figure 6.25: Original and corrected equivalent resistance versus temperature

Chapter 7

Future Work

The experiments and results presented in Chapters 4, 5 and 6 provide affirmation of the feasibility of the proposed PUFs. However, there is still plenty of work to be done. In this chapter, I discuss goals for work that still needs to be done and I describe plans to implement those goals. I first present extensions to the hardware primitive in Section 7.1. This primitive will support measuring process variations in the power distribution network, the analog PUFs and also the integrated architecture. The integrated architecture is reviewed and detailed again in Section 7.2. Then, in Section 7.3, I also present an alternative integrated architecture which may also be fruitful in a design. Finally, in Section 7.4, I discuss some short-term goals.

7.1 Extensions to Hardware Primitive

We plan to investigate one extension to the PUF architecture that will increase the number of single-on scenarios, as shown in Figure 7.1. The new SMC includes additional “voltage sense transistors” that connect to the upper metal layers so that the voltage at any layer, e.g. M1 through M10, can be sensed. This will improve

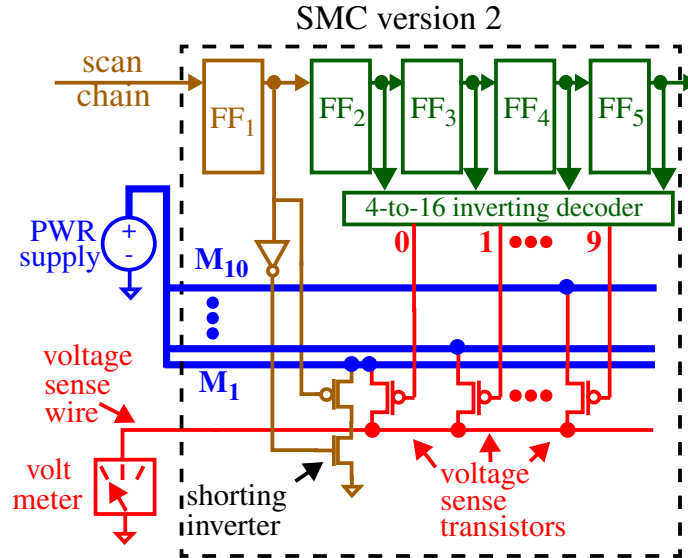


Figure 7.1: SMC with multiple sense transistors for different metal layers

the accuracy of the measurement of the supply voltage at the chip, and remove any variation introduced by the power supply wiring and the chip package. This also provides n VDrops and ERs (n as shown in the figure) to be measured from each SMC. FF₂ through FF₅ drive the inputs to a *4-to-16 inverting decoder* which functions to produce a single ‘0’ on one of the voltage sense transistors when driven with a specific binary pattern. The decoder is connected such that an input bit pattern of all ‘1’s disables all voltage transistors.

We have added this decoding logic to minimize the additional hardware overhead of this SMC architecture. Assuming these modifications triple the size of the PUF circuitry to $150\mu\text{m}^2$, this still only represents 0.06% of a $5\text{mm} \times 5\text{mm}$ chip that includes 100 copies of the SMC. Also, the SMC **leakage current** is negligible since the stacked transistors in the shunting inverter are both off, and there is no voltage drop across the voltage sense transistors, when the SMCs are not being used. Last, each SMC may be able to provide up to 10 times the number of response bits compared to the original scheme and therefore, fewer copies will be needed to achieve a specific

size for the response bit space.

This extended SMC architecture allows any pairing of VDrops or ERs from two different PUFs to be compared. However, in an actual use scenario we must constrain the use of quantities from **only the same layer** in order to maintain an even single-bit probability. This restriction is necessary because the VDrops and ERs should increase monotonically across the vertical dimension of the power grid.

As we did in Section 5.2, we will again explore comparing VDrop and ER *differences* instead of recording the individual absolute values. As with our current hardware, we can measure the absolute values and compute the differences by simply by subtracting the two. For example, as shown in Figure 7.1, we can measure 10 absolute VDrops/ERs but only 9 differences in these values. We believe that the differences will capture the random variations better and tend to reject environmental variations. We will evaluate the results of the per-layer analysis against the standard PUF metrics as discussed in Chapter 5.

7.2 Integrated Architecture

The PUF as proposed requires the use of external instrumentation to measure the voltages and global currents needed to compute the PUF responses. Although this approach serves the chip authentication application well, e.g. where the objective is to periodically check the authenticity of chips to circumvent attempts to replace the chips with counterfeits, it is not amenable to remote authentication or cryptography applications that use the PUF responses as secret keys in encryption/decryption algorithms. In order to serve this latter need, the PUF responses need to be computed using entirely on-chip instrumentation.

The simplest approach to accomplishing this goal was discussed in Section 4.1

and shown in Figure 4.3. In the integrated architecture, the objective will be to design a well-balanced operational amplifier and a “scan chain”¹ controller. In Figure 7.2, the diagram shown in Figure 7.1 has been extended to support the integrated architecture. This circuit is essentially the same as that shown in Figure 7.1, but uses a larger decoder and has the voltage observe transistors replicated for use with the second voltage sense net. The OpAmp connects to the two globally-routed voltage sense wires so that we can decide which voltage is greater. In our prototype, we will also connect these two voltage sense wires to pins so they can be interfaced with two off-chip voltmeters, as shown in the figure. Recall that the purpose of the scan chain controller will be to accept requests in some binary format, drive the scan chain through a sequence of configurations and produce a binary response. Encryption or obfuscation of the CRPs can also be implemented with the microcontroller. The primary constraint on the controller will be to occupy the least amount of area possible, since response time is not really critical.

7.3 Integrated Architecture with an ADC

We have also considered an alternative implementation of the integrated architecture mentioned so far. Instead of having two voltage sense wires and an operational amplifier that compares the two, we can use a single voltage sense net that connects to an Analog-to-Digital Converter (ADC). The calibration of this ADC will be critical to the function of the PUF, but it can be used to convert the voltage into, for example, 3 bits that represent voltages from 800mV to 900mV. Then we can cycle the system through all of the single-shortening configurations and produce N 3-bit numbers. Using these 3-bit numbers, we can perform a simple binary-integer subtraction and decide

¹The scan chain implemented won't be as complete as a standard scan chain definition such as JTAG. The scan chain in this case is essentially a shift register used for configuration.

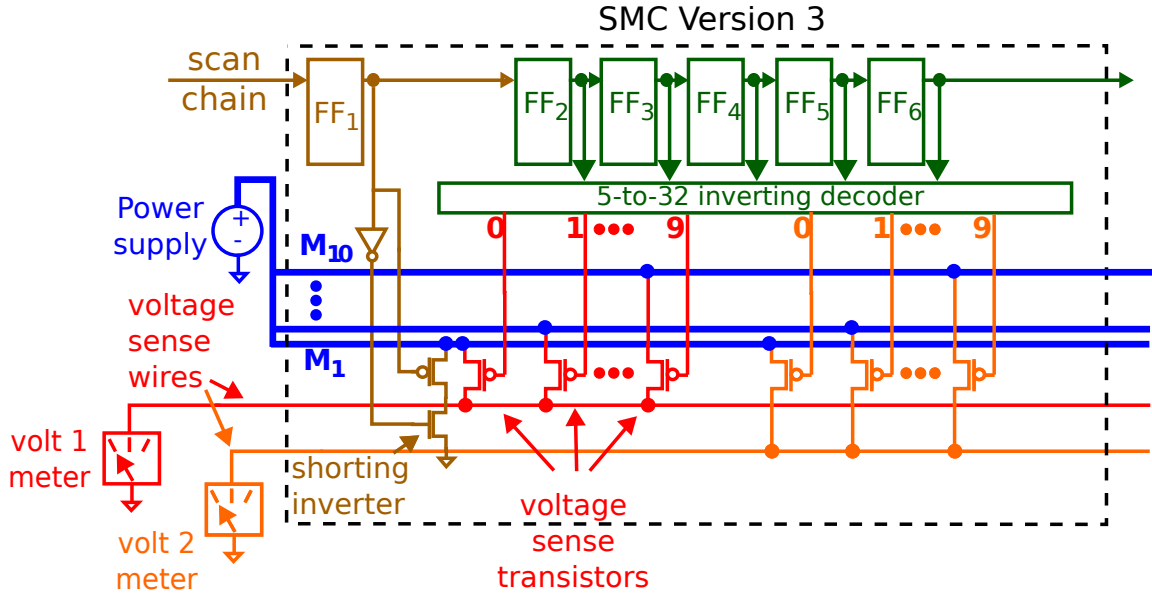


Figure 7.2: Extension of SMC primitive shown in Figure 7.1 to support two voltage sense wires.

which one is larger. This is exactly the process we are performing off-chip with more precision. However, this is by no means the only scheme that could be used. For example, the $3N$ bits could be randomly permuted (rearranged) and then XORed (exclusive or) with $3N$ challenge bits. Then, the result could be the response.

As well as supporting implementing the PUF on-chip, this architecture also supports the use of absolute difference between the two integers as a *metric* to decide if the bit will be “reliable”. For example, if two voltages were less than 2 steps apart, we could label them as “unreliable”. The micro-controller could issue an error code, indicate “equal to” rather than “less than” or “greater than” in the response, or pursue an alternative method of producing the bit that would be more reliable. There are other proposed methods for improving the reliability of PUFs over different operational points such as temperature. The authors of [38] proposed one way to do this is to allow two groups of values (such as ring oscillator frequencies) instead of just two values and pick the pair from the two groups that maximizes

the absolute difference. For example, assume we can measure two sets of voltages (865, 809, 843, 889)mV and (836, 899, 880, 849)mV. The pair consisting of one voltage from each set that maximizes the absolute difference is (809, 899)mV, and $809 < 899$ produces a 0. Then, we can use the second term of each group whenever comparing the groups, and the probability of a bit flip is greatly reduced. However, the cost of this scheme is that only 1/4 of the hardware is being utilized. A scheme such as this one could easily be incorporated into our integrated architecture.

7.4 Short-term Goals

We have been approved for a $3\text{mm} \times 3\text{mm}$ chip which we plan to tape out January, 2011. We plan to target the following objectives using a Process Development Kit (PDK) from IBM for a 90nm bulk technology. The two components I will build will be integrated into an IC which will be a larger group effort supported in part by the National Science Foundation (NSF).

The **SMC primitive** shown in Figure 7.1 **will be designed and layed out**. The **OpAmp and PUF Controller** shown in Figure 4.3 will be designed and layed out. We can then take the IC through the **parasitic extraction phase** using the IBM Monte Carlo process models for the purposes of functional and timing verification. This can also be used to simulate a population of ICs. From this virtual population, we can simulate the performance of the integrated architecture without getting silicon back. Optionally, we plan to **test up to 40 chips**. Although the simulated statistical analysis should track well with results from actual ICs, this is the ultimate proof-of-concept. Completion of this objective will depend on the arrival time of the chips. A **statistical analysis** similar to that performed in previous chapters should follow.

Chapter 8

Conclusions

In Chapter 1, I identified the need to measure process variations in sub-micron technologies and introduced the problems that Physical Unclonable Functions (PUFs) address in hardware security. I described some of the facets of PUF design such as weaknesses and how they are interfaced. I then outlined the rest of the document.

In Chapter 2, discussed other work that studies process variations and physical unclonable functions. I described the applications and implementation techniques that have been proposed for PUFs. I also give a summary of various PUF metrics that are used to evaluate the quality of a PUF.

In Chapter 3, I presented a method to model and measure Power Distribution System (PDS) resistances. This infrastructure can be used to characterize BEOL resistance variations during process bring-up and debug. It can also serve as a process monitor to track variations over time. The embedding of the infrastructure in the context of the actual circuit rather than test structures in isolation increases the relevance of the resistance analysis that it provides. The results of the analysis of resistance variations on two sets of chips fabricated in a 65nm technology illustrates that BEOL variations can be significant (Section 3.6). The proposed infrastructure

Chapter 8. Conclusions

can help reduce delays in manufacturing development and yield learning cycle times caused by BEOL resistance variations.

In Chapter 4, I proposed a PUF that leverages the inherent resistance variations in the metal layers that constitute the power grid. Data from a set of thirty-six chips fabricated in a 65nm technology is used to confirm the feasibility of this strategy. The results show that the responses of the analog PUF, using both equivalent resistance (ER) and voltage drops (VDrops), possess a high degree of randomness and stability. I also described a way to integrate the voltage drop PUF on-chip while requiring minimal hardware overhead, which I refer to as the integrated architecture.

In Chapter 5, I proposed the addition of “multiple-on” scenarios and investigated several quality metrics of the same PUFs that were defined in Chapter 4. The multiple-on scenarios yielded an exponential number of Challenge/Response Pairs (CRPs) with respect to the number of Stimulus/Masurement Circuits (SMCs) and also improved the probability of aliasing for the analog PUFs to 36 times the probability considering only the single-on scenarios. I also emulated the integrated architecture proposed in Section 4.1, by comparing two ERs or VDrops on the same chip to produce bits. Since it uses differences in values, it is therefore a “differential” PUF. The results show that the responses of this differential PUF possess a high degree of randomness and stability. The analysis also revealed that the single-bit probability of the response bits, and the inter-chip and noise Hamming distances are near ideal. However, the increase in entropy is small for response vectors that include data from the 2-on, 3-on, etc. tests, and the usefulness of enabling more than approximately 6 SMCs simultaneously is therefore limited.

In Chapter 6, I added temperature control to the system in order to more deeply understand the measurements I was making. I described the configuration of the Thermo-Electric Cooler (TEC) and the Negative Temperature Coefficient (NTC) thermistor I used for feedback. I characterized a special resistor on the chip as a

Chapter 8. Conclusions

function of temperature, which served later as a reference point for the Temperature Coefficient of Resistance (TCR) of other resistors. I described the theory beyond my temperature control mechanism which I used to bring the system to thermal equilibrium at different temperatures. In Section 6.2, I presented a thorough characterization of the measurement noise. The noise was treated in progressively larger samples, which allowed me to characterize everything from the baseline instrumentation noise to the worst-case noise where temperature drift became an issue. It was shown that temperature control, although not perfect, improved the stability of the temperature of the devices at nominal temperature. The effect of temperature on measurements such as leakage current and transistor “on” current were presented, along with the associated relative noise floors. In Section 6.3, I presented a new analysis of the analog PUFs. I described how the voltage drops and equivalent resistances were affected by temperature and presented a new estimate of the probability of aliasing at each of the temperature points I used. The aliasing probability tends to vary with temperature, but the analog PUFs can still be used at any temperature point. The concept of the “aggregate” temperature analysis was presented, but did not provide a meaningful worst case for the analog PUFs. In order to remedy this problem, I suggested the use of vector angles as an alternative measure of the difference between analog vectors in Section 6.3.3. I showed that this measure is more resistant to temperature-induced changes than the Euclidean distance. In Section 6.4, a new analysis of the digital PUFs, which simulate the performance of an integrated implementation, was presented. It was shown that bits that are unstable at room temperature tend to become more unstable when temperature varies, but stable bits are resistant to variations in temperature. I also observed that the bits that tend to flip are already near or below the noise floor of the corresponding measurement. The probability of aliasing was estimated, and it was shown that when temperature cannot be controlled, the probability of aliasing gets worse only by an order of magnitude. The digital PUF is therefore resistant to temperature

Chapter 8. Conclusions

variations. Other standard metrics such as inter-chip and noise Hamming distances, single-bit probability and bit flip probability were estimated again, and were shown to be near-ideal, regardless of the temperature. Finally, in Section 6.5, I presented an analysis of a reality of the test chips that adversely affects the accuracy of the measurement of on-chip voltage. This inaccuracy accounted for the NTC that I was seeing associated with the “equivalent resistances” that I compute and use as a PUF.

In Chapter 7, I described the many facets of future work on this subject. I presented my ideas for extending the hardware primitive to allow more-accurate characterization of Back End-of-Line (BEOL) process variations by providing more measurements. Extensions of the hardware primitive will also support an advanced PDS PUF that is both more accurate and will have many more CRPs. I also proposed an alternative PUF implementation that uses an Analog-to-Digital Converter (ADC) and lends nicely to integer arithmetic to decide things such as the “reliability” of a bit in the response. Finally, I set forth a list of my short-term goals.

In summary, the theory of this PUF that leverages variations in the power grid that is already present on every IC was presented. The concepts of measuring resistance alone and leveraging variations in the power grid are both novel. I am grateful for the opportunity to present this work two years in a row at the Design Automation Conference (DAC), and receive constructive criticism in the form of both peer reviews and questions at the end of the presentation. There have also been more than one commercial interests. The parametric study of the PUF at different temperature points was much more interesting than I expected. I was able to discover things that are invisible from a single temperature point. Overall, the analysis shows that the PUF exhibits near-ideal performance under the standard PUF metrics, making it an attractive PUF implementation.

Appendix A

Signature Tables

Appendix A. Signature Tables

Following are tables that represent the 192-valued signatures from a few of the chips we measured. The values are listed from left to right and from top to bottom. The first 6 numbers are from the single-on scenario, and the next 30 are from the 2-on scenario, and so on. See Table 5.1 for a complete enumeration. Only the first two chips are included for brevity.

Table A.1: Chip ER signature for chip 1

9.68	9.55	9.86	8.85	8.95	10.22	5.27	5.49	5.34	5.51	5.46	5.39
5.35	4.90	5.46	4.78	5.49	4.88	5.27	5.05	5.38	4.93	5.41	5.03
4.91	5.12	5.27	5.68	5.38	5.54	5.41	5.66	4.89	5.72	5.05	5.68
3.88	4.06	3.98	3.88	4.05	3.55	3.91	4.01	3.59	4.03	3.96	3.54
3.84	4.02	3.67	3.87	3.98	3.72	4.00	3.94	3.66	3.87	3.63	3.77
3.98	3.57	3.71	3.96	3.61	3.76	3.84	4.02	4.08	3.88	3.98	4.13
4.00	3.94	4.07	3.87	3.60	4.17	3.98	3.55	4.11	3.97	3.59	4.16
3.85	3.74	4.15	3.95	3.68	4.09	3.94	3.72	4.15	3.62	3.78	4.18
3.18	3.33	3.25	2.92	3.16	3.31	3.24	3.02	3.16	3.29	2.94	3.06
3.16	3.25	2.97	3.08	3.28	3.23	2.92	3.05	3.16	3.31	3.25	3.33
3.16	3.29	2.92	3.34	3.17	3.25	2.95	3.38	3.28	3.24	2.92	3.34
3.14	3.28	3.04	3.34	3.15	3.24	3.06	3.37	3.27	3.22	3.03	3.35
3.15	2.97	3.10	3.39	3.25	2.94	3.07	3.36	3.23	2.96	3.09	3.39
2.74	2.87	2.81	2.55	2.65	2.74	2.86	2.81	2.54	2.89	2.73	2.86
2.80	2.64	2.89	2.73	2.84	2.57	2.67	2.90	2.73	2.80	2.58	2.68
2.92	2.83	2.79	2.56	2.66	2.90	2.46	2.56	2.51	2.31	2.40	2.60

Appendix A. Signature Tables

Table A.2: Chip ER signature for chip 2

7.94	8.12	8.73	8.97	8.26	8.10	4.57	4.62	4.55	4.86	4.65	4.92
4.50	4.94	4.58	4.97	4.85	4.98	4.42	4.71	4.50	4.75	4.75	4.75
4.94	4.80	4.47	4.57	4.54	4.61	4.82	4.62	4.95	4.63	4.75	4.56
3.41	3.47	3.62	3.39	3.43	3.64	3.36	3.57	3.64	3.44	3.61	3.66
3.35	3.39	3.51	3.33	3.54	3.51	3.40	3.57	3.53	3.30	3.63	3.54
3.36	3.65	3.56	3.53	3.65	3.57	3.38	3.41	3.40	3.36	3.56	3.41
3.43	3.60	3.43	3.33	3.63	3.41	3.38	3.64	3.43	3.56	3.65	3.43
3.30	3.52	3.39	3.35	3.54	3.41	3.52	3.55	3.42	3.65	3.57	3.44
2.81	2.86	2.96	2.98	2.79	2.84	2.95	2.90	2.77	2.81	2.99	2.93
2.76	2.91	2.99	2.92	2.81	2.94	3.00	2.94	2.81	2.85	2.96	2.82
2.78	2.82	2.98	2.81	2.77	2.93	2.98	2.82	2.83	2.95	2.99	2.83
2.77	2.80	2.91	2.81	2.75	2.91	2.91	2.82	2.81	2.94	2.93	2.83
2.74	2.99	2.94	2.82	2.78	3.00	2.95	2.83	2.90	3.00	2.95	2.84
2.44	2.48	2.56	2.60	2.55	2.45	2.49	2.57	2.59	2.46	2.43	2.48
2.56	2.54	2.47	2.42	2.45	2.60	2.56	2.46	2.40	2.54	2.60	2.56
2.47	2.46	2.56	2.60	2.57	2.48	2.20	2.24	2.30	2.34	2.31	2.23

Appendix B

Code Statistics

A large part of the time that was invested into this work went into things outside of this document. To name a few, I spent a lot of time in meetings where I received constructive criticism, many hours working in the laboratory and managing the experiments. However, the largest fraction of time was spent working on code and computing statistics. Below I list some statistics on the codes that were developed to answer all the questions we had about the PUF.

Language	Files	Lines
R	17	1,575
Perl	7	1,286

I used Subversion to manage the files for the analysis. There were over 40 revisions to the code that was developed for the chapter on temperature effects alone. I ran over 38 different experiments in the lab, since we had a sample of 36 chips and we collected several noise samples. This presented a great opportunity to learn R, which was foreign to me when I got started. Without the results from this infrastructure, this research would not have been possible.

References

- [1] S. R. Nassif, “Modeling and Analysis of Manufacturing Variations,” in *Conference on Custom Integrated Circuits*, 2001, pp. 223–228.
- [2] K. Agarwal and S. Nassif, “Characterizing Process Variation in Nanometer CMOS,” in *Design Automation Conference*, 2007, pp. 396–399.
- [3] I. Ahsan, N. Zamdmer, O. Glushchenkov, R. Logan, E. Nowak, H. Kimura, J. Zimmerman, G. Berg, J. Herman, E. Maciejewski, A. Chan, A. Azuma, S. Deshpande, B. Dirahoui, G. Freeman, A. Gabor, M. Gribelyuk, S. Huang, M. Kumar, K. Miyamoto, D. Mocuta, and Mahoro, “RTA-Driven Intra-Die Variations in Stage Delay and Parametric Sensitivities for 65 nm Technology,” in *Symposium on VLSI Technology*, 2006, pp. 170–171.
- [4] T. B. Hook, “Lateral Ion Implant Streggle and Mask Proximity Effect,” *IEEE Trans. Electron Devices*, vol. 50, no. 9, pp. 1946–1951, September 2003.
- [5] C. Hedlunk, H. Blom, and S. Berg, “Microloading Effect in Reactive Ion Ethching,” *Journal of Vacuum Science and Technology*, vol. 12, pp. 1962–1965, 1994.
- [6] R. S. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, “Physical one way functions,” pp. 2026–2030, 2002. [Online]. Available: <http://web.media.mit.edu/~brecht/papers/02.PapEA.powf.pdf>
- [7] Wikipedia, “Physical Unclonable Function — Wikipedia, The Free Encyclopedia,” 2010, [Online; accessed 3-November-2010]. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Physical_Unclonable_Function&oldid=394631874
- [8] C. Hess, B. E. Stine, L. H. Weiland, and Kazuhiro, “Logic Characterization Vehicle to Determine Process Variation Impact on Yield and Performance of Digital Circuits,” in *International Conference on Microelectronic Test Structures*, no. 15, 2002, pp. 189–196.

References

- [9] K. Y. Y. Dong, “Electrical Characterization of Model-Based Dummy Feature Insertion in Cu Interconnects,” in *International Conference on Microelectronic Test Structures*, 2004, pp. 87–92.
- [10] A. Bassi, A. Veggetti, L. Croce, and A. Bogliolo, “Measuring the Effects of Process Variations on Circuit Performance by Means of Digitally-Controllable Ring Oscillators,” in *International Conference on Microelectronic Test Structures*, March 2003, pp. 214–217.
- [11] S. Park, “Statistical Design of Experiments and Analysis on Gate Poly-Silicon Critical Dimension,” *IEEE Trans. on Semiconductor Manufacturing*, vol. 17, no. 3, pp. 362–374, Aug 2004.
- [12] B. Zhou and A. Khouas, “Measurement of Delay Mismatch due to Process Variations by Means of Modified Ring Oscillators,” in *International Symposium on Circuits and Systems*, vol. 5, 2005, pp. 5246 – 5249.
- [13] K. Agarwal, F. Liu, C. McDowell, S. Nassif, K. Nowka, M. Palmer, D. Acharyya, and J. Plusquellic, “A Test Structure for Characterizing Local Device Mismatches,” in *Symposium on VLSI Circuits*, June 2006, pp. 67–68.
- [14] K. M. G. V. Gettings, D. Boning, and S. Duane, “Test Circuit for Study of CMOS Process Variation by Measurement of Analog Characteristics,” in *International Conference on Microelectronic Test Structures*, 2007, pp. 37–41.
- [15] F. Larsen, M. Ismail, and C. Abel, “A Versatile Structure for On-Chip Extraction of Resistance Matching Properties,” *IEEE Trans. on Semiconductor Manufacturing*, vol. 9, pp. 281–285, 1996.
- [16] J. Deveugele, Y. Libin, M. Steyaert, and W. Sansen, “Mismatch Characterisation of Chip Interconnect Resistance,” in *International Conference on Microelectronic Test Structures*, 2005, pp. 183–186.
- [17] H. Sayah and M. Buehler, “Linewidth and Step Resistance Distribution Measurements using an Addressable Array,” in *International Conference on Microelectronic Test Structures*, 1990, pp. 87–92.
- [18] P. J. Wright, E. Burke, and A. T. Appel, “VLSI Interconnect Linewidth Variation: a Method to Characterize Depth of Focus and Proximity Effects,” in *International Conference on Microelectronic Test Structures*, 1992, pp. 185–189.
- [19] R. Helinski and J. Plusquellic, “Measuring Power Distribution System Resistance Variations,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 21, no. 3, pp. 444 – 453, August 2008.

References

- [20] R. S. Pappu, “Physical One-Way Functions,” PhD in Media Arts and Sciences, Massachusetts Institute of Technology, 2001.
- [21] B. Gassend, D. E. Clarke, M. van Dijk, and S. Devada, “Silicon physical unknown functions,” in *Conference on Computer and Communications Security*, 2002, pp. 148–160.
- [22] S. Maeda, H. Kuriyama, T. Ipposhi, S. Maegawa, Y. Inoue, M. Inuishi, N. Kotani, and T. Nishimura, “An artificial fingerprint device (AFD): a study of identification number applications utilizing characteristics variation of polycrystalline silicon TFTs,” *Electron Devices, IEEE Transactions on*, no. 50, pp. 1451–1458, June 2003.
- [23] Y. Su, J. Holleman, and B. Otis, “A 1.6pJ/bit 96% stable chip ID generating circuit using process variations,” in *International Solid State Circuits Conference (ISSCC)*, 2007, pp. 406–407.
- [24] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, “Physical Unclonable Functions and Public Key Crypto for FPGA IP Protection,” in *Conference on Field Programmable Logic and Applications*, 2007, pp. 189–195.
- [25] E. Simpson and P. Schaumont, “Offline Hardware/Software Authentication for Reconfigurable Platforms,” vol. 4249, pp. 311–323, 2006.
- [26] S. S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls, “Extended Abstract: The Butterfly PUF Protecting IP on every FPGA,” in *Hardware-Oriented Security and Trust*, 2008, pp. 70–73.
- [27] B. Gassend, D. Lim, D. Clarke, M. van Dijk, and S. Devadas, “Identification and Authentication of Integrated Circuits,” *Concurrency and Computation: Practice and Experience*, 2003.
- [28] Y. Alkabani, F. Koushanfar, and M. Potkonjak, “Remote Activation of ICs for Piracy Prevention and Digital Right Management,” in *ICCAD '07: Proceedings of the 2007 IEEE/ACM international conference on Computer-aided design*. Piscataway, NJ, USA: IEEE Press, 2007, pp. 674–677.
- [29] K. Lofstrom, W. R. Daasch, and D. Taylor, “IC Identification Circuits using Device Mismatch,” in *International Solid State Circuits Conference (ISSCC)*, 2000, pp. 372–373.
- [30] W. Adi and B. Soudan, “Bio-Inspired Electronic-Mutation with genetic properties for Secured Identification,” in *ECSIS Symposium on Bio-inspired, Learning, and Intelligent Systems for Security*, 2007, pp. 133–136.

References

- [31] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, “Controlled Physical Random Functions,” in *18th Annual Computer Security Applications Conference*, December 2002.
- [32] B. Gassend, M. V. Dijk, D. Clarke, E. Torlak, S. Devadas, and P. Tuyls, “Controlled Physical Random Functions and Applications,” *ACM Transactions on Information and System Security*, vol. 10, no. 4, January 2008.
- [33] T. Gneysu, B. Mller, and C. Paar, “Dynamic Intellectual Property Protection for Reconfigurable Devices,” in *International Conference on Field-Programmable Technology*, December 2007, pp. 169–176.
- [34] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, “Brand and IP protection with physical unclonable functions,” in *IEEE Symposium on Circuits and Systems*, May 2008, pp. 3186–3189.
- [35] J. Huang and J. Lach, “IC Activation and User Authentication for Security-Sensitive Systems,” in *Hardware-Oriented Security and Trust*, 2008, pp. 79–83.
- [36] J. Li and J. Lach, “At-Speed Delay Characterization for IC Authentication and Trojan Horse Detection,” in *IEEE International Workshop on Hardware-Oriented Security and Trust*, 2008, pp. 8–14.
- [37] Y. Jin and Y. Makris, “Hardware Trojan Detection Using Path Delay Fingerprint,” in *IEEE International Workshop on Hardware-Oriented Security and Trust*, 2008, pp. 54–60.
- [38] G. E. Suh and S. Devadas, “Physical Unclonable Functions for Device Authentication and Secret Key Generation,” in *Design Automation Conference*, 2007, pp. 9 – 14.
- [39] Y. Alkabani, T. Massey, F. Koushanfar, and M. Potkonjak, “Input vector control for post-silicon leakage current minimization in the presence of manufacturing variability,” in *Design Automation Conference, 2008, 45th ACM/IEEE*, June 2008, pp. 606 – 609.
- [40] R. Helinski, D. Acharyya, and J. Plusquellic, “A Physical Unclonable Function Defined Using Power Distribution System Equivalent Resistance Variations,” in *Design Automation Conference*, 2009, pp. 676–681.
- [41] M. Majzoobi, F. Koushanfar, and M. Potkonjak, “Testing Techniques for Hardware Security,” in *International Test Conference*, 2008, pp. 185–189.
- [42] M. Popovich, A. V. Mezhiba, and E. G. Friedman, *Power Distribution Networks with On-Chip Decoupling Capacitors*. Springer, 2008.

References

- [43] J. Lee, M. Tehranipoor, and J. Plusquellic, “A Low-Cost Solution for Protecting IPs Against Scan-Based Side-Channel Attacks,” in *VLSI Test Symposium*, May 2006, pp. 42 – 47.
- [44] J. Lee, M. Tehranipoor, C. Patel, and J. Plusquellic, “Securing Designs Against Scan-Based Side-Channel Attacks,” in *Transactions on Dependable and Secure Computing*, vol. 4, no. 4, December 2007, pp. 325–336.
- [45] R. Helinski, D. Acharyya, and J. Plusquellic, “Quality Metric Evaluation of a Physical Unclonable Function Derived from an ICs Power Distribution System,” in *Design Automation Conference*, 2010, p. to appear.
- [46] Wikipedia, “Resistivity — Wikipedia, The Free Encyclopedia,” 2010, [Online; accessed 29-October-2010]. [Online]. Available: <http://en.wikipedia.org/w/index.php?title=Resistivity&oldid=393210689>
- [47] A. von Glasow, A. Fischer, and G. Steinlesberger, “Using the temperature coefficient of the resistance (TCR) as early reliability indicator for stressvoiding risks in Cu interconnects,” in *Reliability Physics Symposium Proceedings, 2003. 41st Annual. 2003 IEEE International*, May 2003, pp. 126 – 131.
- [48] Wikipedia, “Thermistor,” 2010, [Online; accessed 27-August-2010]. [Online]. Available: <http://en.wikipedia.org/w/index.php?title=Thermistor&oldid=381098142>
- [49] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits*, second edition ed., S. E. Charles G Sodini, Ed. Prentice Hall Electronics and VLSI Series, 2003.
- [50] S. Devadas, E. Suh, S. Paral, R. Sowell, T. Ziola, and V. Khandelwal, “Design and Implementation of PUF-Based ”Unclonable” RFID ICs for Anti-Counterfeiting and Security Applications,” April 2008.