Universitat
de les Illes Balears

DOCTORAL THESIS

2021

CONTRIBUTIONS ON USING EMBEDDED MEMORY CIRCUITS AS PHYSICALLY UNCLONABLE FUNCTIONS CONSIDERING RELIABILITY ISSUES

Abdel Rahman Naser Abed Alrazzaq Alheyasat

Universitat
de les Illes Balears

DOCTORAL THESIS

2021

Doctoral Program in Electronic Engineering

CONTRIBUTIONS ON USING EMBEDDED MEMORY CIRCUITS AS PHYSICALLY UNCLONABLE FUNCTIONS CONSIDERING RELIABILITY ISSUES

Abdel Rahman Naser Abed Alrazzaq Alheyasat

Supervisors:
Dr. Bartomeu Alorda Ladaria
Dr. Gabriel Torrens Caldentey

Tutor:
Dr. Bartomeu Alorda Ladaria

Doctor by the Universitat de les Illes Balears

ACKNOWLEDGMENTS

Writing fast is surely not my talent, but luckily now only this one page remains incomplete. So, let me take the opportunity to thank all of those who have been influential during the course of this thesis.

Before all others, I am grateful to my promotor *Dr. Bartomeu Alorda Ladaria* who graciously offered me a position as PhD student, and for welcoming me to the research group *Electronic System Group* of University of the Balearic Islands in May 2018. Without your advice none of this work can be done.

I am also much thankful to *Dr. Gabriel Torrens Caldentey* for introducing me to the related topics and simulations of this thesis and for guiding me from the beginning of this project until the last word of this thesis. His remarks have certainly improved the quality of this thesis.

Special thanks to *Dr. Sebastián Antonio Bota Ferragut* for his help in the beginning and his ideas and guidance during my PhD study. Finally, I express my gratefulness to the facility members of the *Doctorate School* for their Intellectual support throughout the course of this work.

Abdel Rahman Alheyasat

DEDICATION

﴿وآخر دعواهم أن الحمد لله رب العالمين﴾

To my parents (Naser, Ghada) and my beloved wife and daughter (Eng-Rand, Rand), who offered me unconditional love and support during the course of this thesis.

Un agradecimiento especial a Maria Guadalupe "Lupe", su amable hospitalidad durante mi estudio en Mallorca es algo que nunca olvidaremos en toda nuestra vida. Muchas gracias mamá mallorquina!

ABSTRACT

Moving towards Internet-of-Things (IoT) era, hardware security becomes a crucial research topic, because of the growing demand of electronic products that are remotely connected through networks. Novel hardware security primitives based on manufacturing process variability are proposed to enhance the security of the IoT systems. As a trusted root that provides physical randomness, a physically unclonable function is an essential base for hardware security.

SRAM devices are becoming one of the most promising alternatives for the implementation of embedded physical unclonable functions as the start-up value of each bit-cell depends largely on the variability related with the manufacturing process. Not all bit-cells experience the same degree of variability, so it is possible that some cells randomly modify their logical starting value, while others will start-up always at the same value. However, physically unclonable function applications, such as identification and key generation, require more constant logical starting value to assure high reliability in PUF response. For this reason, some kind of post-processing is needed to correct the errors in the PUF response.

Unfortunately, those cells that have more constant logic output are difficult to be detected in advance. This work characterizes by simulation the start-up value reproducibility proposing several metrics suitable for reliability estimation during design phases. The aim is to be able to predict by simulation the percentage of cells that will be suitable to be used as PUF generators. We evaluate the metrics results and analyze the start-up values reproducibility considering different external perturbation sources like

several power supply ramp up times, previous internal values in the bit-cell, and different temperature scenarios. The characterization metrics can be exploited to estimate the number of suitable SRAM cells for use in PUF implementations that can be expected from a specific SRAM design.

RESUM

En l'era de la Internet de les coses (IoT), garantir la seguretat del hardware ha esdevingut un tema de recerca crucial, en especial a causa de la creixent demanda de productes electrònics que es connecten remotament a través de xarxes. Per millorar la seguretat dels sistemes IoT, s'han proposat noves solucions hardware basades en la variabilitat dels processos de fabricació. Les funcions físicament inclonables (PUF) constitueixen una font fiable d'aleatorietat física i són una base essencial per a la seguretat hardware.

Les memòries SRAM s'estan convertint en una de les alternatives més prometedores per a la implementació de funcions físicament inclonables encastades. Això és així ja que el valor d'encesa de cada una de les cel·les que formen els bits de la memòria depèn en gran mesura de la variabilitat pròpia del procés de fabricació. No tots els bits tenen el mateix grau de variabilitat, així que algunes cel·les canvien el seu estat lògic d'encesa de forma aleatòria entre enceses, mentre que d'altres sempre assoleixen el mateix valor en totes les enceses. No obstant això, les funcions físicament inclonables, que s'utilitzen per generar claus d'identificació, requereixen un valor lògic d'encesa constant per tal d'assegurar una resposta fiable del PUF. Per aquest motiu, normalment es necessita algun tipus de postprocessament per corregir els possibles errors presents en la resposta del PUF. Malauradament, les cel·les que presenten una resposta més constant són difícils de detectar a priori.

Aquest treball caracteritza per simulació la reproductibilitat del valor d'encesa de cel·les SRAM, i proposa diverses mètriques per estimar la fiabilitat de les cel·les durant les fases

de disseny de la memòria. L'objectiu és ser capaç de predir per simulació el percentatge de cel·les que seran adequades per ser utilitzades com PUF. S’avaluen els resultats de diverses mètriques i s’analitza la reproductibilitat dels valors d’encesa de les cel·les considerant diverses fonts de pertorbacions externes, com diferents rampes de tensió per a l’encesa, els valors interns emmagatzemats prèviament en les cel·les, i diferents temperatures. Es proposa utilitzar aquestes mètriques per estimar el nombre de cel·les SRAM adients per ser implementades com a PUF en un disseny d‘SRAM específic.

RESUMEN

En la era de la Internet de las cosas (IoT), garantizar la seguridad del hardware se ha convertido en un tema de investigación crucial, en especial a causa de la creciente demanda de productos electrónicos que se conectan remotamente a través de redes. Para mejorar la seguridad de los sistemas IoT, se han propuesto nuevas soluciones hardware basadas en la variabilidad de los procesos de fabricación. Las funciones físicamente inclonables (PUF) constituyen una fuente fiable de aleatoriedad física y son una base esencial para la seguridad hardware.

Las memorias SRAM se están convirtiendo en una de las alternativas más prometedoras para la implementación de funciones físicamente inclonables empotradas. Esto es así, puesto que el valor de encendido de cada una de las celdas que forman los bits de la memoria depende en gran medida de la variabilidad propia del proceso de fabricación. No todos los bits tienen el mismo grado de variabilidad. Así pues, algunas celdas cambian su estado lógico de encendido de forma aleatoria entre encendidos, mientras que otras siempre adquieren el mismo valor en todos los encendidos. Sin embargo, las funciones físicamente inclonables, que se utilizan para generar claves de identificación, requieren un valor lógico de encendido constante para asegurar una respuesta fiable del PUF. Por este motivo, normalmente se necesita algún tipo de posprocesado para corregir los posibles errores presentes en la respuesta del PUF. Desafortunadamente, las celdas que presentan una respuesta más constante son difíciles de detectar a priori.

Este trabajo caracteriza por simulación la reproductibilidad del valor de encendido de celdas SRAM, y propone varias métricas para estimar la fiabilidad de las celdas durante

las fases de diseño de la memoria. El objetivo es ser capaz de predecir por simulación el porcentaje de celdas que serán adecuadas para ser utilizadas como PUF. Se evalúan los resultados de varias métricas y se analiza la reproductibilidad de los valores de encendido de las celdas considerando varias fuentes de perturbaciones externas, como diferentes rampas de tensión para el encendido, los valores internos almacenados previamente en las celdas, y diferentes temperaturas. Se propone utilizar estas métricas para estimar el número de celdas SRAM adecuadas para ser implementadas como PUF en un diseño de SRAM específico.

Table of Contents

Title Page

ACKNOWLEDGMENTS I

DEDICATION II

ABSTRACT III

TABLE OF CONTENTS IX

LIST OF FIGURES XIII

LIST OF TABLES XVII

LIST OF ABBREVIATIONS AND SYMBOLS XVIII

CHAPTER 1: INTRODUCTION........ 1

1.1 Motivation 2

1.2 Objectives and Major Contributions 5

1.3 Thesis Outline 7

CHAPTER 2: BACKGROUND........ 9

2.1 Physical Unclonable Function (PUF) 9

2.1.1 Definition of a PUF........ 10

2.1.2 PUF Application 11

2.1.2.1 Cryptographic Key Generation 11

2.1.2.2 Low-Cost Authentication 13

2.1.2.3 True Random Number Generation (TRNG)........ 14

2.1.3 PUF Classification 14

2.1.3.1 Extrinsic based PUFs 15

2.1.3.2 Intrinsic based PUFs........ 15

2.2 SRAM-PUF 16

2.2.1 SRAM Cell Architecture 16

2.2.2 SRAM SUV as Source for SRAM PUF 18

2.2.3 State of Art on SRAM-PUF Reliability Enhancement 20

2.2.3.1 Error Correcting Codes (ECCs) 21

2.2.3.2 Majority Voting Methods 22

2.2.3.3 Post-Fabrication Burn-in Enhancement 23

2.2.4 Direct and Indirect Preselection Approaches 24

2.2.4.1 Our Approach 25

CHAPTER 3: METRICS METHODOLOGY FOR IMPROVING SRAM-PUF RELIABILITY..27

3.1 Simulation Environment Setup 27

3.2 Reliability Metrics based on Inherent-Cell Mismatch 30

3.2.1 Threshold Voltage Distance Between the Individual Transistors.................. 30

3.2.2 Switching Voltage Point Distance between Inverters 36

3.2.3 Summary.. 40

3.3 Reliability Metrics Based on SNM Concept 41

3.3.1 SNM Concept 41

3.3.2 Proposed Metric 42

3.3.2.1 SNM Distance (SNM_d) Metric 43

3.3.2.2 VTCs Intersection Distance (INTd) Metric 46

3.3.3 Correlation with Previous Metric in literature (PSNM ratio) 48

3.3.4 Correlation with Inherent-Cell Mismatch.. 50

3.4 Voltage Noise Injection Methodology as Reliability Metrics 52

3.4.1 Metric Methodology.. 53

3.4.1.1 Noise Injection at Ground of the cell 53

3.4.1.2 Noise Injection at Storage Nodes of the cell...................................... 56

3.4.1.2 Noise Injection at Cell Power Supply Nodes 58

3.4.2 Correlations Between Noise Injection Locations 60

3.4.3 Correlation with Inherent-Cell Mismatch.. 63

3.5 Dynamic Start-up Behavior as Reliability Metrics 66

3.5.1 Graphical Representation of SUV 66

3.5.2 SRAM Separatrix as Reliability Metrics 73

CHAPTER 4: EXPLORING EXTERNAL AND INTERNAL PERTURBATIONS IMPCACT 81

4.1 Simulation Setup 82

4.2 Impact of Power Supply Ramp-Up Time (RUT) on Start-Up Behavior 83

4.2.1 RUT Test Methodology and Impact Explanation 84

4.2.2 Proposed Metrics Robustness Considering RUT Variations 89

4.3 Impact of Previously Stored Value (PSV) on Start-Up Behavior 93

4.3.1 PSV Test Methodology and Impact Explanation 94

4.3.2 Proposed Metrics Robustness Considering PSV Variations 97

4.4 Impact of Temperature on Start-Up Behavior and Robustness of Metrics 98

4.5 Impact of Internal Noise on Start-Up Behavior and Robustness of Metrics 101

CHAPTER 5: REPRODUCIBILITY CHARACTERIZATION AND PROPOSED METRICS DISCUSSION 107

5.1 Strong SRAM Cells to Improve PUF reliability 108

5.2 Selection of Strong Cells Using Mismatch Metrics 110

5.2.1 Parameter Distance-Based Metrics Discussion 110

5.2.2 SNM-Based Metrics Discussion 114

5.2.3 Voltage Noise Injection-Based Metrics Discussion...................................... 117

5.2.4 SRAM Separatrix-Based Metrics Discussion 119

5.2.5 Summary 120

5.3 Influence of Selected PUF Response Length 121

CHAPTER 6: CONCLUSION AND FUTURE WORK...123

6.1 Conclusions 123

6.2 Future Work 124

REFERENCES 126

LIST OF FIGURES

Figure Page

Figure 2.1: Diagram of embedded PUF-based key generator 12

Figure 2.2: Illustration of object authentication scheme based on PUF [10] 13

Figure 2.3 (a): 6T SRAM cell schematic in hold mode 17

Figure 2.3 (b): Latch voltage transfer characteristics 17

Figure 2.4: Start-up behavior for 50 cells when the power supply is ramped up 19

Figure 3.1: 6T SRAM Cell showing transistors contribution to SUV 31

Figure 3.2: Distribution of Mismatch Parameter between N-MOS and P-MOS 33

Figure 3.3: Histograms of Mismatch Metric (Pd_{Vtho}) values depending on the SUV 34

Figure 3.4: Histograms of Mismatch Metric (Pd_{Vth}) values including the Weighting Factors (w=0.76) 35

Figure 3.5: Graphical technique to obtain Mismatch Metric (Pd_{Vm}) values by using VTCs for the cell's inverters 38

Figure 3.6: Histograms of Novel Mismatch Metric (Pd_{Vm}) values depending on the SUV 39

Figure 3.7: The Relationship between Pd_{Vtho} and Pd_{Vm} 40

Figure 3.8: Static Noise Margin definition using VTCs curves 42

Figure 3.9: 45° rotated VTCs.. 44

Figure 3.10: Histogram of SNM_d values considering the reference SUV 45

Figure 3.11: Intersection Distance metric (*INTd*) definition using VTCs curves 46

Figure 3.12: Histogram of *INTd* values considering the reference SUV 48

Figure 3.13: Correlation between values of SNM-based metrics 49

Figure 3.14: Correlation between our proposed metrics and PSNM ratio 50

Figure 3.15: Correlation between values of SNM_d and the mismatch metric............... 51

Figure 3.16: Correlation between values of *INTd* and the mismatch metric................ 51

Figure 3.17: 6T SRAM cell with noise injection at the ground nodes 54

Figure 3.18: The histogram distribution of Vn_g metric.. 56

Figure 3.19: 6T SRAM cell with noise injection between storage nodes 57

Figure 3.20: The histogram distribution of Vn_i metric 57

Figure 3.21: 6T SRAM cell with noise injection at power supply nodes 59

Figure 3.22: The histogram distribution of Vn_{ps} metric... 59

Figure 3.23: The relation between Vn_g and Vn_{ps} metrics 60

Figure 3.24: The relation between Vn_i and Vn_g metrics... 61

Figure 3.25: The relation between Vn_i and Vn_{ps} metrics.. 62

Figure 3.26: The relation between Vn_g metric and inherent mismatch 64

Figure 3.27: The relation between Vn_i metric and inherent mismatch........................ 65

Figure 3.28: The relation between Vn_{ps} metric and inherent mismatch 65

Figure 3.29: Q versus QB voltages during start-up for several SRAM cells................... 67

Figure 3.30: The histogram distribution of Vmax indicator 69

Figure 3.31: The relation between the graphical indicator *Vmax* and Mismatch metric 69
Figure 3.32: The relation between the graphical indicator *ΔSlope* and Mismatch metric 71
Figure 3.33: The relation between *Area(Q-QB)* and *Vmax* indicators 72
Figure 3.34: The relation between *Area (Q-QB)* and Mismatch metric 72
Figure 3.35: The Phase-space of SRAM memory evolution at start-up stage........ 74
Figure 3.36: The Phase-space of memory cells evolution and the definition of *SID* metric 77
Figure 3.37: The histogram distribution of separatrix metric (*SID*) 79
Figure 3.38: The relation between the separatrix metric (*SID*) and mismatch 80
Figure 4.1: 6T SRAM cell including two random noise sources 83
Figure 4.2: percentage of cells that change the SUV at 5ns for several RUTs 84
Figure 4.3: percentage of stable cells and unstable cells that change their SUV at specific RUT value 85
Figure 4.4: Distribution of Vth variation of P-MOS and N-MOS transistors showing the SUVs at different RUTs 87
Figure 4.5: The relation between P-MOS and N-MOS contributions to SUV with respect to RUT 89
Figure 4.6: Proposed parameter distance-based metrics relation indicating the Stable-RUT cells 90
Figure 4.7: Proposed SNM-based metrics relation indicating the Stable-RUT cells 91
Figure 4.8: Proposed injected noise-based metrics relation indicating the Stable RUT cells........ 92
Figure 4.9: The histogram of Proposed *SID* metric indicating the percentage of Stable RUT cells........ 93

Figure 4.10: percentage of cells that change the SUV for PSVs variation compared to reference (0 V, 0 V) 95
Figure 4.11: Distribution of Vth variation of P-MOS and N-MOS transistors and their contributions to the SUVs at different PSVs 96
Figure 4.12: Studying the ability of the metrics against PSVs impact 98
Figure 4.13: Percentage of cells that change the SUV for temperature variations compared to nominal 27º 99
Figure 4.14: Studying the ability of the metrics against Temperature impact 101
Figure 4.15: Visualization of the SRAM array with the probability to start-up to a preferred SUV 103
Figure 4.16: The relation between inherent cell-mismatch and the probability of the cell to start-up to a preferred SUV 104
Figure 4.17: The performance of the proposed metrics in classifying cells repeatability considering the modeled internal noise 106
Figure 5.1: The histograms for literature (ΔN and ΔP) methodologies showing strong cells distribution 111
Figure 5.2: The histograms for the parameter distances showing strong cells distribution 112
Figure 5.3: The histogram distribution for literature PSNM ratio identifying the strong cells... 115
Figure 5.4: The histograms for the proposed SNM metrics showing strong cells distribution... 115
Figure 5.5: The histograms for the injected noise-based metrics showing strong cells distribution... 118
Figure 5.6: The histogram distribution for dynamic *SID* metric identifying the strong cells.. 119
Figure 5.7: Percentage of strong cells identified by each best metric approach considering different response lengths... 121

LIST OF TABLES

Table | Page

Table 5.1: Number of cells stable, repeatable and strong identified by Parameter distances metrics selecting 64 PUF-bits 113

Table 5.2: Number of cells stable, repeatable and strong identified by SNM-based metrics selecting 64 PUF-bits 116

Table 5.3: Number of cells stable, repeatable and strong identified by injected noise based metrics selecting 64 PUF bits 118

LIST OF ABBREVIATIONS AND SYMBOLS

IoT	Internet of Things.
PUF	Physically Unclonable Function.
SRAM	Static Random-Access Memory.
SUV	SRAM cell Start-Up Value.
SNM	Static Noise Margin.
DNMs	Dynamic Noise Margins.
VTCs	Voltage Transfer Characteristics.
ICs	Integrated Circuits.
CRPs	Challenge-Response Pairs.
NVM	Non-Volatile Memory.
POWFs	Physical One-Way Functions.
ECCs	Error Correction Codes.
TRNG	True Random Number Generation.
RUT	Ramp-Up Time.
PSVs	Previously Stored Values.
IBS	Index Based Syndrome.
BTI	Bias Temperature Instability.
FPGA	Field-Programmable Gate Array

ASIC	Application-Specific Integrated Circuit.
SA	sense-amplifier.
Vdd	SRAM Ramp-Up Voltage.
WL	Word-line.
BL	Bit-Line.
lpsvt	Low Power Standard Threshold Voltage CMOS.
lphvt	Low Power High Threshold Voltage CMOS.
lpsvt	Low Power Low Threshold Voltage CMOS.
Vpwl	Voltage Piecewise linear.
OCEAN	Open Command Environment for Analysis.
r_{SUV}	SUV reproducibility.
ΔP	Threshold voltage difference between P-MOS transistors.
ΔN	Threshold voltage difference between N-MOS transistors.
Pd_{Vtho}	Threshold Voltage Parameter Distance metric.
Pd_{Vm}	Inverter Switching Point Parameter Distance metric.
SNM_d	SNM Distance.
INT_d	VTCs Intersection Distance

Vn_g	Voltage Noise injection at Ground nodes of the cell
Vn_i	Voltage Noise injection between the internal storage nodes of the cell.
Vp_s	Voltage Noise injection at the power supply nodes of the cell.
SID	Separatrix Intersection Distance.

CHAPTER 1

INTRODUCTION

The Internet of Things (IoT) is an innovation with industrial, commercial and consumer applications. The IoT allows a massive number of devices to be connected remotely with an affordable price. This innovation can provide many benefits, like a more efficient and eco-friendly industry, and make daily life more convenient. Unfortunately, alongside with all the advantages the IoT comes with, the security of the communication systems is a serious challenge. In addition, this problem gets worse due to the fast growth of the number of IoT devices. As a consequence, a massive amount of data is collected and being transferred between the IoT devices. The data communicated through IoT networks can contain important information that might lead to threats to user's privacy if the security is compromised and data is leaked. Therefore, communication networks between devices must be highly secure, which is specifically challenging because of the large number of devices and the resource limitation of many IoT devices. It is difficult to reliably protect most IoT devices, as most of them cannot afford robust cryptographic systems within the restricted budget for both power consumption and manufacturing cost.

This chapter introduces the motivation of this work in Section 1.1. The achieved objectives and major contributions of the thesis are explained in Section 1.2. Finally, Section 1.3 presents the outlines of the thesis.

1.1 Motivation

The digital revolution that happened over the past decades allowed that transferring enormous amount of data across the world at high speed has become something normal. This uprising has demanded the usage of applications that need a large amount of online connections. From the essential online banking, to the entertainment of online gaming; from restraining the access to an administration facility, to supporting full access to a laboratory and research facility. Nearly each application utilized nowadays needs such interconnections. The data which is exchanged through these nets could be anything varying from user personal information to accounts with massive amount of money. Consequently, in order to protect the authenticity and confidentiality of these interconnections it is compulsory to consider some degree of security to be included in the systems. The information exchanged in these systems needs to be protected from theft while allowing them to be productive and accessible only by the intended users. However, the Information Security of IoT is the field where all techniques and tools are dedicated to protecting systems information.

The common practice is to utilize cryptographic methods such as signing algorithms, encryption, and decryption, to secure the transported and stored information. The cryptographic algorithms implemented for these methods are available for the public, nevertheless the secret key generated by them is securely saved into the device. Based on that, it is important to ensure the security of the generated key. The classical security systems that are available in the market rely on storing the secret key or the crucial information in a Non-Volatile Memory (NVM). Smart Cards, Credit Card and TV channels Card are examples for the application of these classical systems. However, the main

goals of these systems are to stop cloning and theft of service [1]. These methods, even though secure, cannot be protected against some of the adversary's attacks. As the secret key generated from these systems is stored permanently somewhere in the device, and hence, any person that can access the device and knows how the security is implemented can attack this system. Therefore, security solutions should improve to become more secure against these threats.

New hardware security solutions should be developed to compensate for the growing privacy and security risks in the IoT systems. Recent approach to overcome the drawbacks of the classical security system is the use of embedded physical unclonable functions (PUFs). The main idea of a PUF is to generate a secret key from unclonable and unpredictable physical features related to the variability related to the manufacturing process technology of Integrated Circuit (IC) of IoT devices. Taking profit form this process variation, PUF can generate a unique and unclonable fingerprint for each IC.

Therefore, to secure the communications between IoT devices, the secret key generation scheme based on PUF can be a more secure and lightweight method to generate a unique non-stored key for the devices. This key is generated whenever needed without requiring storing it in the device. These properties present the PUFs as an ideal candidate for hardware security of IoT [2]. However, the main issue in PUF implementation is to assure that the generated key is always the same under different operational and environmental conditions.

There are different approaches to introduce PUF solutions in IoT devices. However, SRAM devices are gaining attention because its presence is very common in many IoT devices and the start-up value (SUV) of each individual bit-cell depends to a greater or

lesser extent on the variability of the IC manufacturing process [3]. Remember that the SUV defines the logical value that the cell acquires by itself when it is polarized from 0V to its nominal value Vdd, before any data is written in it. Depending on how the variability affects each bit-cell transistor, we will have suitable-PUF SRAM cells whose SUV will always be the same, with a value that is unclonable and unpredictable a priori, on the other hand, other cells present an SUV that can be different in each star-up process. However, achieving acceptable reliability of SRAM-based PUF output usually requires some additional intervention, since not all cells in a memory can be suitable for that application [78].

In the literature, a huge amount of works has proposed and evaluated different SRAM methodologies using specific laboratory equipment to estimate the impact on reliability and robustness of bit-cells designs when process variation, temperature, and other factors are also involved [61, 79-80], but there are few works related to estimate the suitability of a given SRAM design to be used in PUF applications. One reason lies in the difficulty of simulating the different external processes involved in determining the SUV of a given symmetrical bit-cell affected by process variations with certain start-up conditions that are not fully known. Although it is completely unpredictable to know a priori the position of the suitable cells, or what their starting value will be, due to what has been previously commented, it will be very useful to have some indication in the design stage what percentage of suitable cells to expect, or if an optimization of the bit-cell design is possible to increase that value, or if it is advisable to use additional help modules.

1.2 Objectives and Major Contributions

SRAM-PUFs are becoming a popular solution. The reliability of this PUF by performing constant SUVs under internal noise and different environmental conditions is considered the main challenge [4]. Understanding several parameters that influence the SUV, and quantifying their influence is crucial for designing robust and reliable security systems based on SRAM-PUF. Therefore, this thesis aims to characterize the reliability of SRAM bit-cell in terms of its SUV reproducibility against external perturbations and internal noise. The impact of internal parameters (such as Threshold Voltage), and the external parameter (such as Temperature) are also investigated.

In this line, this work is focused in the characterization of the percentage of cells of a given SRAM cell design that are suitable to be used as PUFs. These cells should show more constant SUVs under different simulated perturbations. This is performed by electrical simulation and thus the percentage of suitable cells can be obtained at an early stage of the design phase. To achieve this, the SUVs of SRAM cells are simulated and a series of metrics based on Monte Carlo simulations involving process variation implemented using 65nm CMOS commercial technology, are proposed to contribute in quantifying the strength of the cell start-up behavior to the impact of the process variability, then, by correlating the distribution of each metric with the corresponding SUV of each SRAM cell. These metric based evaluations may be used at the design stages to adapt and change the SRAM implementation design to match the predicted number of PUF-suitable cells to the needs of the PUF. Of conversely, the PUF design can be modified to adapt to the expected SRAM performance.

Finally, some of these proposed metrics and other contents of this thesis have been published in a well-known conferences:

1. ***Alheyasat, A., Torrens, G., Bota, S. and Alorda, B., 2019, November. Weak and Strong SRAM cells analysis in embedded memories for PUF applications. In 2019 XXXIV Conference on Design of Circuits and Integrated Systems (DCIS) (pp. 1-6). IEEE.*** (The main contributions and results of this paper are included in Subsections *2.1.1*, *2.2.2* and *3.2.1* of this thesis).
2. ***Alheyasat, A., Torrens, G., Bota, S. and Alorda, B., 2020, October. Bit-Cell Selection Analysis for Embedded SRAM-Based PUF. In 2020 IEEE International Symposium on Circuits and Systems (ISCAS) (pp. 1-4). IEEE.*** (The main contributions and advanced results of this paper are presented in the following subsections of this thesis: *3.2.1*, *3.4.1.1*, *3.4.1.2*, *3.4.3*)
3. ***Alheyasat, A., Torrens, G., Bota, S. and Alorda, B., 2020, November. Selection of SRAM Cells to Improve Reliable PUF implementation using Cell Mismatch Metric. In 2020 XXXV Conference on Design of Circuits and Integrated Systems (DCIS) (pp. 1-6). IEEE.*** (Subsection *3.5.1* and Section *4.5* of this thesis are mainly based on the content of this conference paper)

Additionally, the last conference paper (Selection of SRAM Cells to Improve Reliable PUF implementation using Cell Mismatch Metric) is invited for publication in "*Special Issue on Innovation in Computing, Engineering Science & Technology organized by Advances in Science, Technology and Engineering Systems Journal (ASTESJ)*" and the Manuscript has been submitted with title "**Selection of SRAM Cells to improve Reliable PUF implementation using Separatrix and Mismatch metrics**". However, the main content

and construction of this thesis have been based on Article Manuscript that have been submitted to "*Integration - Journal - Elsevier*" with title of "**SRAM-cells Reproducibility Characterization for Physical Unclonable Function Applications**" and it is under review process.

1.3 Thesis Outline

Chapter 2 provides an overview of the literature and the important background related to the work in this thesis. The definition of PUFs and the common application, such as cheap identification, authentication, cryptographic key generation and true random number generator, are discussed. The PUF classifications based on the location of physical process variation is summarized to finish the first part of this chapter. The principle of SUV as Source for SRAM-PUF will be discussed in the second part, where the process variation and mismatch are presented as a reason behind the entropy of SUV. Then, we present a state of the art of some techniques to enhance the reliability of SRAM-PUF. Finally, we introduce the preselection methods where the reliable SRAM cells can be detected and selected to increase the PUF reproducibility.

Chapter 3 presents the main contribution of the thesis. We characterize the SUV reproducibility proposing several mismatch metrics suitable to estimate reliability during design phases. The evaluation of the proposed metrics is based on a Monte Carlo simulations methodology. These metrics can be categorized based on the parameter they are based on: Inherent Mismatch-based metrics, Static Noise Margin (SNM) based metrics, Voltage Noise injection-based metrics and some metrics based on the dynamic start-up behavior.

Chapter 4 describes the other main contribution, which is exploring the potential influences of external and internal perturbations on the start-up behavior of SRAM cells, and hence on the reliability of PUF cell under these perturbations. Firstly, we characterize the impact of power supply Ramp-Up Time (RUT) on the SRAM-PUF. Similarly, the influence of Previously Stored Values (PSVs) into SRAM cells will be studied. We also analyze how the ambient temperature affect the SUV of the cells. These three perturbations are used to distinguish between the stable and unstable cells. Additionally, we present the impact of internal noise on start-up behavior to achieve the probability of memory cells to have a repeatable SUV under the induced voltage noise. Finally, all the proposed metrics in chapter 3 will be studied under these perturbations.

Chapter 5 correlates both obtained results in Chapter 3 and 4 and their implications in terms of SRAM cell characterization. In other words, this chapter studies the robustness of the metrics in identifying the suitable PUF cells. Firstly, we define the most reliable SRAM cells, denoted as *Strong cells*, that tolerate all the perturbations described in Chapter 4. Secondly, we present and compares the ability of proposed metrics in identifying the *Strong cells*, also we describe the methodology of selecting those cells. Using the metrics, we explore the influence of selected cells length on the reliability of PUF operation in the last section. Where we finally present the methodology to estimate the percentage of suitable cells for PUF application.

Chapter 6 concludes the findings and contributions of the work in this thesis. It also suggests some recommendations for future work.

CHAPTER 2

BACKGROUND

The growing demand of IoT devices and applications increases the challenges in systems security. As a trusted root for these IoT devices, the embedded PUFs in the integrated circuits (ICs) are crucial for securing the devices and communication systems of IoT. By taking the advantage of random process variations, which are inherently produced during the manufacturing, PUFs can produce unique and unpredictable bits that can be used for various security applications.

This chapter presents an overview of the literature and subject background. In section 2.1 we discuss the definition of PUFs and its common application. Also, the PUF classifications to either extrinsic or intrinsic PUFs will be summarized. Section 2.2 explains the principle of SUV as Source for SRAM-PUF, discussing reasons behind the process variation and mismatch. Finally, we present a state of art on some techniques and preselection methods to enhance the reliability of SRAM-PUF, presenting the same goal where our approach is briefly introduced.

2.1 Physical Unclonable Function (PUF)

PUF is an emerging technology that provides a promising security solution with a relatively low complexity and cost. It operates by mapping a set of responses that corresponds to a set of challenges. Challenge-Response Pairs (CRPs) correlation are mainly determined by the inherent process variations in a silicon chip. These process variations are caused by uncontrollable manufacturing variations in the chip process.

These deviations are unpredictable and unique between different dies and wafers. Hence, a PUF can be implemented to provide a unique and unpredictable CRPs. Additionally, the randomness and complexity of inherent process variations makes a PUF physically and practically impossible to clone [5].

2.1.1 Definition of a PUF

PUF is basically used to provide a signature from a physical device, so the definitions of PUF may change based on the purpose it is utilized for. Generally, a PUF can be defined by the words it contains from as follows [4]: *Physical* means it is not purely a mathematical function, but a physical object, and its output generated from physical interaction. *Unclonable* indicates how hard to predict or replicate its output. And *Function* means that it has an input (Challenge) to produce output (Response) based on physical parameters.

Recently, PUFs are becoming more popular in the area of semiconductor security. As semiconductor manufacturing processes have intrinsic variations. A circuit fabricated in silicon shows slightly different electrical behavior from one sample to another even though the wafer mask and design are identical. This is the basis of PUF, which in turn allows to improve the security level of several IC-based systems. The operation of a PUF is based on applying a challenge and measure its response. For each challenge, there should be a valid response to have an authenticated operation. In 2001, the concept of Physical One-Way Functions (POWFs) was proposed by Pappu for the first time [2]. Using challenge-response setup, the physical object can be exposed to many challenges, all of which generate an unpredictable response that is unique between different objects.

2.1.2 PUF Application

Security applications can rely on silicon implemented PUFs due to the unique and inherent features of each IC that are derived from process variations. The first application is Cryptographic Key Generation [6-8], which draws out the keys from the unique bits generated by embedded PUF. The second application is the device Authentication [9-10], which saves the unique PUF feature of each chip and it uses them later on to check if a chip can be authenticated or not. Finally, an important base block for many cryptographic security systems is the True Random Number Generator (TRNG), that can be based on the noise properties of PUFs [11-14].

2.1.2.1 Cryptographic Key Generation

To improve the cryptographic key generation and storage applications, an alternative method based on PUFs can be exploited. Generally, Memory-based PUFs such as SRAM-PUF is typically applied for generating cryptographic keys [15 -16].

The principle of key generation based on PUF is summarized in Fig. 2.1. Firstly, a PUF array, that include entropy created from process variation, is readout using an interface circuit to convert the PUF output to digital bits, then these output bits are directed to the helper-data algorithm [6]. This algorithm usually operates off-chip to produce a sequent of helper data that will be used in the field to generate the key. The helper-data is then stored into an embedded NVM, where the data can be read without the ability to write, and this data is later-on utilized to reconstruct the required key for cryptographic application [7].

However, outputs from these PUF array are noisy because of environmental condition variations and ageing [17]. Therefore, direct PUF implementation for the cryptographic

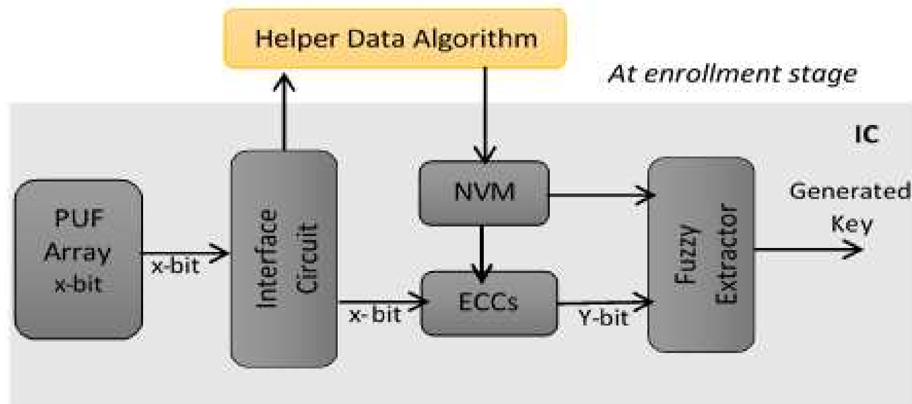


Fig.2.1: *Embedded PUF-based key generator.*

key generation is not feasible; as this security application necessitates that the generated key is highly stable [8]. Based on that, there are two core functions that should be added to the PUF-based key generation scheme; the Error Correction Codes (ECCs) and fuzzy extractor in Fig. 2.1. The ECCs are needed to correct the noisy PUF readouts. While the purpose of the fuzzy extractors is to counteract the impact of non-uniformity of PUF bits, ensuring that the generated key will be in line with the specification required by this application. Next section (section 2.2) provides more details about these techniques.

Using PUF for cryptographic key generation has many advantages compared to traditional key generation using a secure NVM. Mainly, the feature of physical security is changed from NVM to the PUF, that is more immunized against physical attacks because it is very hard to measure the inherent process variations.

Unfortunately, using ECCs requires to increase area overheads as the bit error rate rises from noisy PUF [18-19]. This increase in the area overhead can reduce the interest of PUFs as a low-cost security object. As a motivation, we propose a bit-selection metric methodology in Chapter 3 to identify a subset of most reliable cells that has very low bit

error rate. Selecting only these cells for key generation can reduce the need for ECCs and hence the needed area overhead for such security application.

2.1.2.2 Low-Cost Authentication

As another well-known application, the object authentication based on PUF is becoming more popular. The authors in [10] propose a lightweight object authentication structure implementing the so-called “strong PUF”, their method relies on the well-known challenge-response authentication technique in [20]. The object authentication in these works consists of two stages as shown in Fig.2.2, the enrollment stage and the authentication stage.

In the enrollment stage, the server will randomly produce many challenges for each device that have the same embedded PUF. The devices will receive these challenges and they will send the responses to the server matching to each challenge, these CRPs will be stored as authentication references. Note that the enrollment stage should be done in a secure environment.

In the authentication stage, to verify if a device is registered in the server database, the server will select one of the challenges from the database and direct it to the device that requests authentication. Corresponding to the arrived challenge, the device will

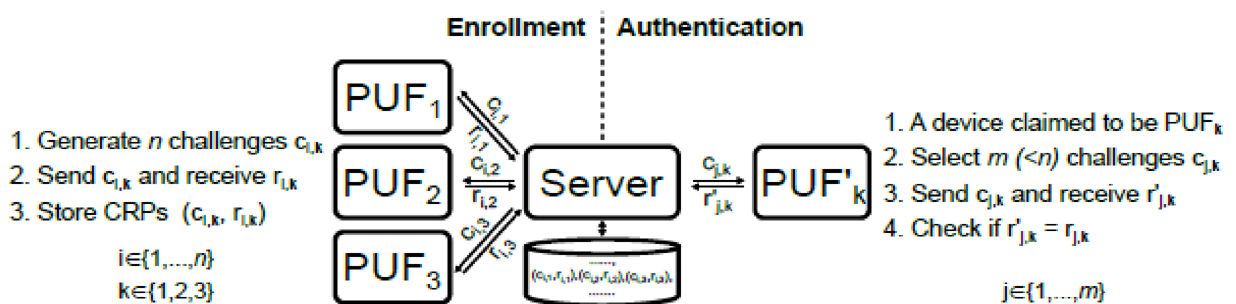


Fig.2.2: *Illustration of object authentication scheme based on PUF [10].*

generate a response and send it back to be verified. This process may be repeated many times. If the server received enough verified responses, the device will be authenticated; else the device will be rejected.

As the remote connection between the devices and the server may have a public communication channel, the server should delete the used CRPs immediately after the authentication process; avoiding replay attack.

2.1.2.3 True Random Number Generation (TRNG)

In this section, we will mention the possibility of implementing a PUF for random number generation [11-12,14]. Process variation can result in two kinds of PUF bit behavior. The *static* bit behavior, this behavior is related to the PUF circuits that are highly mismatched under the impact of process variations. Some designs may have well-matched elements, such as SRAM cell. These designs can work as a TRNG because they can be highly affected by internal circuit noise and very small variations in operating conditions. This is a *variable* bit behavior, that causes errors in the response of PUF.

Most often PUF designers assume that errors in the response are not useful. Alternatively, these errors can be used as a source of random numbers. However, the authors in [13] combined both behaviors to create a circuit that achieves both as a PUF and an RNG functions. This design provides the main two requirements for any cryptographic security systems: key and random number generation.

2.1.3 PUF Classification

The PUFs can be classified based on the location of the physical randomness to *Extrinsic* based PUFs or *Intrinsic* based PUFs [21].

2.1.3.1 Extrinsic based PUFs

Extrinsic based PUFs use the randomness that is explicitly shown in a physical system. To produce this randomness, the user or the manufacturer can have different choices of materials or size of particles; but the randomness distribution and their end location are out of control. Thus, these PUFs still have unique and random response. The main benefit of these types of PUFs that provide an extrinsic randomness, their parameters can be controlled and optimized in such a way to improve the uniqueness and reliability for these PUFs. The main examples of these types of PUF are the *Coating-PUFs* [22] and *Optical PUFs* [23]. The coating-PUF is based on the random sized dielectric particles that contained in the protective opaque coating at the top layer of an IC, while the optical-PUF is based on a transparent medium like glass, where light scattering particles are explicitly brought in. The position of light scattering particles in the medium is totally unpredictable providing a unique unpredictable property for this PUF.

2.1.3.2 Intrinsic based PUFs

Different from Extrinsic based PUFs, the *Intrinsic* based PUFs utilize the randomness that exist intrinsically in them to function. This intrinsic randomness is a result of process variation throughout the fabrication process that is uncontrollable and naturally random. Recently, such type of PUF has been more popular and attractive, as they can be implemented without needing any modifications to PUF circuit and manufacturing process. Another benefit of intrinsic based PUFs is that most of these PUFs provide a digital output ("1" or "0" response) reducing the need for quantization process before implementing them in security application. Mainly, there are three types of Intrinsic based PUFs [21]: *Silicon-PUFs* operates based on the random variations in transistor gate and wire delays in a circuit [24], *Buttery-PUFs* is mainly used to protect the Intellectual

Property (IP) in FPGAs that do not have integrated memories by creating structures within the FPGA matrix which behave similarly to an SRAM bit-cell during the power-up phase[25], and *SRAM-PUF* [26] that have utilized for this thesis work.

2.2 SRAM-PUF

A PUF instance can be contained within an IoT devices using two main approaches: adding a specialized primitive designed to offer the unclonable function, or reusing a non-specific and pre-existing circuit. Following the second approach, SRAM circuits were proposed for PUF implementation as its possible to take benefit of the positive feedback loop inherent of SRAM bit-cells to generate a unique, stable and unclonable binary response. As these memories are widely included in digital systems, power consumption and costs requirements can be minimized while creating embedded security resources for IoT systems.

2.2.1 SRAM Cell Architecture

A six transistors (6T) memory cell topology is very common to use among different SRAM cell topologies (such as 7T, 8T, 10T). Although, this work uses 6T memory cell, the same methodologies, that will be explained, can be extrapolated to rest cell topologies. The 6T cell consists of two access transistors controlled by the word-line (WL), and two cross-feedback inverters creating a latch circuit, see Fig 2.3 (a). The 6T-SRAM cell has three equilibrium points: two stable points corresponding to logic ‘0’ and logic ‘1’, and a third meta-stable point corresponding to the cross point of the voltage transfer characteristics (VTC) plot of the latch as in Fig 2.3 (b). The SRAMs are intended to work in three modes: Hold operation (as in Fig 2.3 (a) when WL=0 V, then the internal nodes are isolated), Read and Write operations. For stable operation, depending on which

combination, a logic “1” or “0” is stored. One of the internal nodes (Q and QB nodes, see Fig 2.3 (a)) should be at low voltage (0 V) and the other at high voltage (Vdd).

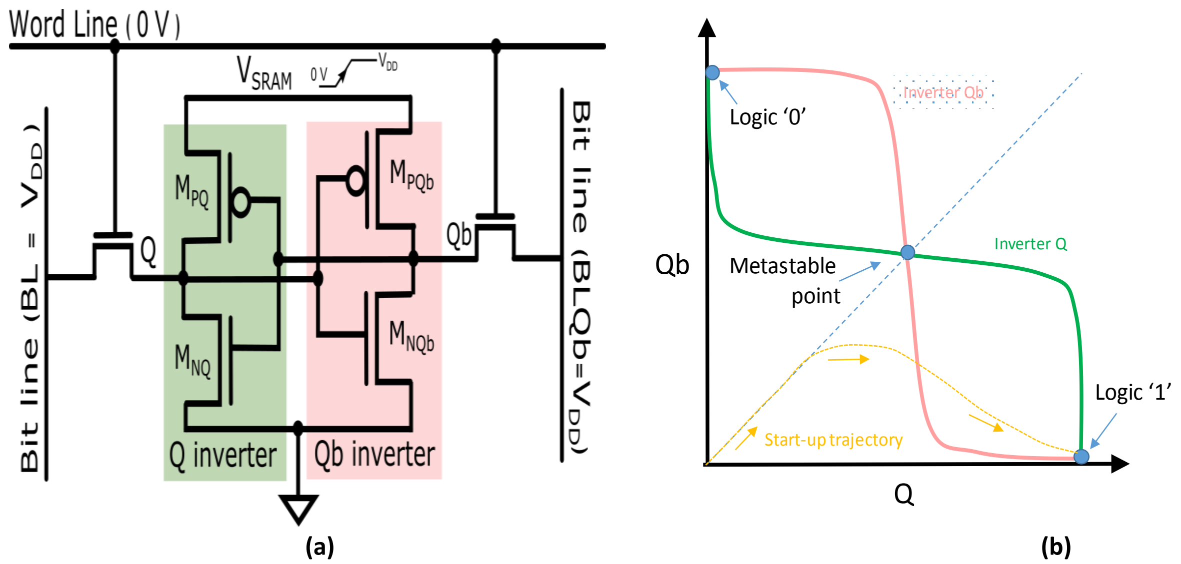


Fig.2.3: *(a) 6T SRAM cell schematic in hold mode. (b) Latch voltage transfer characteristics; the dashed line represents the transient start-up behavior (assuming the final output is logic ’1’).*

To read logic ‘0’ and logic ‘1’ from SRAM cell, the read operation is done by firstly pre-charging the bit-lines (BL_Q, BL_{Qb}). A pre-charge circuit is implemented to control the bit-lines ensuring the same voltage at both bit-lines. When both the bit-lines have the same voltage, the Word Line of the selected cell is activated. Depending on the stored value in the cell, one of the bit-lines will remain the voltage and the other one will be discharged. The voltage difference between bit-lines is detected and amplified by a sense amplifier to evaluate the read output.

The write operation is achieved by pulling one of the bit-lines (BL_Q, BL_{Qb}) to low level and the other to high level. The write driver is implemented to pull down either BL_Q or BL_{Qb} based on the value that is intended to be written into the cell. Then, the Word Line of the selected cell is activated. The feedback of the cross-coupled inverters will oppose to the write process. Despite of that, each of the internal nodes ends up having the same voltage logical level of the bit-line value it is connected to through pass transistors.

An ideal 6T-cell is designed to keep the data during read accesses and permit modifying the stored logic value during write operations, independently of the stored content (logic ‘0’ or logic ‘1’). Thus, the two inverters of the SRAM cell should be as identical as possible to produce symmetrical performance during both write and read operations.

In Hold-operation configuration (see Fig 2.3 (a)), the word line is set to a low voltage level and both the internal nodes are isolated from the bit-lines. Using this setup, in addition to controlling the cell voltage supply (V_{SRAM}), allows SRAM circuit to operate as a PUF; as it will be discussed in the next section.

2.2.2 SRAM SUV as Source for SRAM PUF

At power-up, the internal nodes of the cells are discharged (Q = 0V and Qb = 0V, see Fig 2.3 (a)), setting the Word Line to low voltage, these nodes are isolated from the bit-lines as the access transistors are in the cut-off region. Once the power is ramping up, at first, the voltages of the internal nodes increase equally. Through this period, the memory cell stays in its meta-stable point until the supply voltage exceeds a certain threshold value (different for each cell), from this moment, the effect of the feedback, together with the influence of the mismatch between inverters, will cause the cell to transition towards one of the two stable states as shown in Fig.2.4. Therefore, for a perfectly symmetrical SRAM cell, the final state will only be determined by the noise and environmental conditions, which are unknown. Fig.2.4 shows start-up behavior during ramping-up of the power supply voltage for 50 memory cells, these curves are obtained using Monte Carlo simulation to mimic the mismatch inside the cell and the process variation between the proposed cells.

The SRAM PUF primitive is based on the circumstance that, even though the cell design is symmetrical, the manufacturing process variations generates mismatch between cross-coupled inverters reducing the intrinsic symmetry of the cell, and leading it to one of the two stable states with more probability than the other [27]. The variation in the final SUV of the cells in Fig.2.4 is mainly caused by the non-controlled physical variations that are inherently produced in each cell during their manufacturing process. Hence, the response of a set of SRAM cells will be unpredictable, unique, and unclonable.

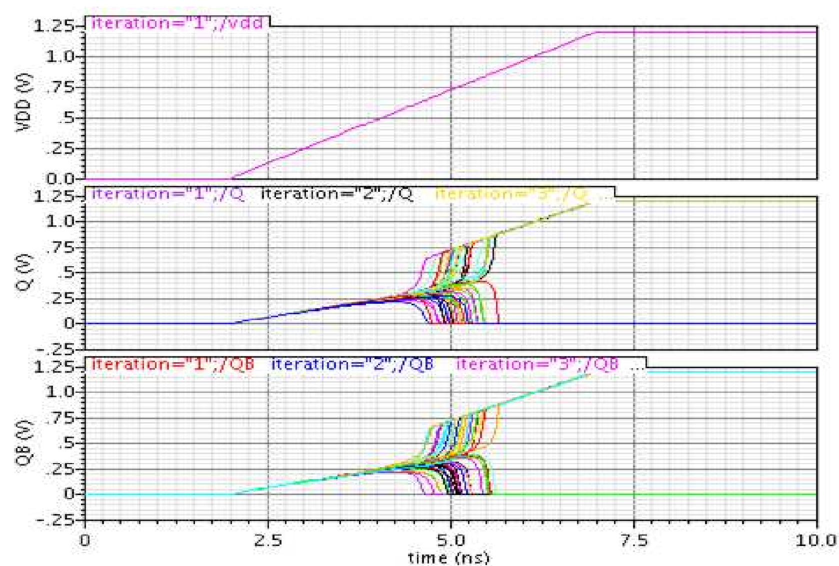


Fig.2.4: *Start-up behavior for 50 cells when the power supply is ramped up.*

Based on that, implementing an SRAM PUF in a circuit only needs to control the power supply of the SRAM core cells. So, the needs to modify the SRAM are minimal and thus it can be applied to any non-specific embedded memories.

Designing SRAM requires the cell to be as symmetric as possible aiming to decrease any bias to a preferred value. Highly symmetrical cells, which are less affected by process variability, will be characterized by varying their SUV between different start-ups. If a cell is started-up N times, it will choose n_1 times a logical '1' SUV, and n_0 times a logical '0' SUV. Then, the probability that the cell has a logical '1' SUV (or to logical '0') is

presented in the following equations:

$$p_1 = \frac{n_1}{N} \tag{2.1}$$

$$p_0 = \frac{n_0}{N} = 1 - p_1 \tag{2.2}$$

From here, we can define the SUV reproducibility of a cell as:

$$r_{SUV} = |p_1 - p_0| \tag{2.3}$$

For reliable PUF implementation, it is essential to select those cells that have significant inherent-mismatch to achieve r_{SUV} to be one or near one. These cells are typically denoted as strong cells from a PUF point of view. However, applying SRAM for TRNG applications requires that cells that are highly symmetrical with r_{SUV} around zero.

2.2.3 State of Art on SRAM-PUF Reliability Enhancement

Knowing the SUV reproducibility distribution at early design stages is essential to decide if a proposed memory scheme is suitable for specific application that requires PUF implementation. Additionally, the early knowledge of the percentage of strong cells will be useful to decide whether the SRAM-PUF can be implemented as it is without modification, or if it will be necessary to apply some reliability techniques. Among the available reliability techniques, we can find the inclusion of error correction codes [28-33], or Majority Voting techniques [4, 34-36]. Applying some of them implies the acceptance of some drawbacks like increasing the design complexity, or increasing power consumption and response time. Another method that can be used to improve SUV reproducibility is based on taking advantage of long period degradation mechanisms of the devices. These mechanisms, known as Burn-in Enhancement, aim to

increase the mismatch within cells and can, hence, producing more reproducible PUF data [38-40]. The next sections will discuss these methods and techniques and their feasibility.

2.2.3.1 Error Correcting Codes (ECCs)

As mentioned previously, PUF outputs are noisy because of environmental condition variations and ageing [17]. Therefore, direct PUF implementation for applications such key generation is normally not feasible. Several methodologies for ECC have been proposed, and most of them have a drawback of an extra cost overhead when being physically implemented.

Some recent designs of ECC for PUF-based key generation suggested the usage of 2-D Hamming Codes [10] [28]. The 2-D Hamming Code reformats the PUF response into a matrix having rows and columns. These codes are applied to generate redundant data for each column and row of the matrix. This data as well as parity bits is utilized to correct these errors in the noisy PUF responses. The main disadvantage of this code is that the generated matrix for PUF cannot contain more than two errors within one row to have a successful operation.

A more improved ECC technique for PUFs uses Bose-Chaudhuri-Hocquenghem (BCH) codes [29]. The implementation of this technique was able to correct 30 errors in a 255 PUF-bits response. Unfortunately, this implementation exposed 192 bits to be used as a Helper Data, so it limited the key size to 63 bits.

Some other approaches for improving PUFs error correction have been proposed. Index-Based-Syndrome (IBS) [28] is more immunized to data leak than conventional ECCs that utilize bitwise XOR-masks. Other enhancements involve soft decision data to

be included within Helper Data Algorithm to reduce the number of PUF bits needed to generate one bit for secure key generation [3]. To reduce the complexity of ECC, the authors in [30] introduce two stage coding that consist of using a syndrome generation based on XOR-mask and repetition coding, more details about these coding will be described in the next section.

As all the previously mentioned methods focus on ECC itself, other researches recommend reducing ECC overhead which is the main drawback this technique. The required Helper Data to produce a secret key increases with increasing the errors in the PUF response exposing more data about the PUF response. Therefore, to ensure a secure key generation, the number of response bits have to be increased to correct as much as possible the errors in the response and reducing the need to expose more data. The area overhead needed for ECC, associated with the increment in the number of PUF response bits, grows linearly with the error rate [31]. The area overhead can be reduced as the PUF reliability increases. Such as in PUF implementation based on FPGA, reducing the error rate from 10% to 3% can save up to 40% of the area overhead [32]. Similarly, ASIC applications using PUFs demonstrate that 60% of area is saved as the error rate decreases from 20% to 5% [33].

2.2.3.2 Majority Voting Methods

Majority voting has been proposed as a useful technique to enhance the reliability of a PUF cell [34-35]. This technique could effectively improve PUF reliability If the response is highly affected by environmental and transient noise.

Majority voting can be divided into two types: space majority voting or temporal majority voting. In temporal majority voting, each bit in the PUF response is challenged

several times and the final output of this bit is defined as the majority of the outputs [35]. Implementing this method, the error rate could be effectively reduced [34]. However, it is only useful for correcting the error rate up to 8% [36]. The main drawback of this type of majority voting is the extra runtime. In addition, its reliability cannot be assured, because some PUF bit-cells might either vary its output under environmental conditions, but others might be well-matched bit-cell, in such case, the majority is near to even [37].

On the other hand, the space Majority voting, also denoted as repetition coding, produces one reliable bit voting between a few unreliable PUF cells. The drawback is that it requires extra area, power, and more PUF cells [4].

2.2.3.3 Post-Fabrication Burn-in Enhancement

The post-fabrication burn-in technique is based on using a mechanism to degrade the devices for long-terms. The idea of this technique is to intentionally take advantage of specific aging effects, and induce them into the PUF circuit. This is done by providing additional run time while the circuit is subjected to temperature and voltage stress, with the goal to cause a time-based variability in addition to the pre-existing variability generated by process variations. Once these time-based variabilities are produced in the planned direction, they can increase mismatch inside the cells, and thus can produce more reliable PUF outputs.

As an example, exploring the Bias Temperature Instability (BTI) degradation in the SRAM-PUFs is explained in [38], where the process variation is only considered for P-MOS transistors. As the SRAM PUF keeps storing the SUVs in the cells, the transistors will display the stress condition. P-MOS transistor that have higher threshold voltage

(*Vth*) will be affected more by the stress, and its *Vth* will decrease as long as the same stress condition is applied. As the higher *Vth* is decreasing, the mismatch between the P-MOS pair will also decrease; which reduces the reproducibility of this PUF, and makes it less immunized against noise. To solve this issue, the authors of [38] also proposed a solution. The method is to store the opposite of the SUVs by re-programming the whole SRAM-PUF array aiming to increase the mismatch in the cells, and thus their reliability.

It is important to mention that It needs long time for these BTI stresses to be effective under nominal conditions, and hence, this method requires to be accelerated by increasing the temperature and voltage. This process still needs many hours up to days to produce a good result.

The burn-in enhancement methods are also proposed for specific PUF implementations. A method is presented in [39], implementing a sense-amplifier PUF (SA-PUF) including burn-in function to explore the HCI degradation mechanism. Another method is the hybrid-PUF that uses BTI mechanism for burn-in improvement [34, 40].

PUF implementations using these burn-in methods have resulted in improving the stability of the generated data. Besides their advantage, the main drawback is the required time and cost, as discussed in [38].

2.2.4 Direct and Indirect Preselection Approaches

To overcome some of the drawbacks of the previous methods, PUF-bit preselection approaches have been proposed [18,41-44]. In these approaches, the unreliable PUF-bits are identified in the PUF array and then masked out from the PUF response. However, these methods can be divided into direct and indirect PUF-bit preselection. In case of direct preselection, during the testing stage, the PUF-bit array is challenged

several times under wide range of operational and environmental conditions, such as different power supply and temperatures variations. The cells that display unstable behavior are directly marked and their pattern in the array is saved, to be excluded from PUF response. The disadvantage of this type of preselection that it requires massive tests and thus it will be expensive in test time, also it will be not totally reliable as its impossible to predict all the operational and environmental conditions. In the works [34, 40], results show that implementing this method can improve reliability of PUF response, especially when increasing the number of the PUF test preformed at different conditions, as it may detect more unstable cells.

The indirect PUF-bit preselection method is based on a test that indirectly detects the unstable PUF-bits; this test is done for each of the PUF bits [4]. If a PUF-bit passes the test, it will be classified as stable and could be implemented to generate the PUF response. The indirect preselection may detect all the unstable PUF-bit, but it may classify some of the stable PUF-bits as unstable, decreasing the number of suitable bits in the response [44]. The preselection pattern of those stable cells can be saved in NVM, as in [18] and [45-46].

2.2.4.1 Our Approach

Tacking profit of the indirect preselection approaches, we propose a metric-based methodology obtained by simulation to characterize the reproducibility of SUV (r_{SUV}) for SRAM-PUF implementation. The simulated metrics are evaluated based on internal SRAM parameters affected by process variation and mismatch. The proposed metrics can be simulated under any single operational and temperature conditions. The idea of our metric methodology is to assign a value for each SRAM cell in the proposed memory to represent how reliable the cell will be under the impact of internal and external

perturbations. Such that, the cells that have higher values of a metric are considered more reliable and its SUV will be more reproducible under these perturbations. Including only this type of cells in the PUF will improve the quality of its response. By contrast, lower metric values will expose the cells that have low reproducibility in their SUVs. Even though these cells should be ignored and masked out for PUF applications such key generation, some of them could be useful as a source of randomness for TRNG applications, however, this fact is out of the scope of this thesis and it has not been studied. The next chapter will provide the detailed methodology of these metrics.

CHAPTER 3

METRICS METHODOLOGY FOR IMPROVING SRAM-PUF RELIABILITY

SRAM PUFs use the start-up value (SUV) of an SRAM cell for PUF application such as cryptographic key generation as discussed in previous chapter. The reliability and the stability of this start-up behavior is a crucial issue that requires to be ensured when using SRAM cells as a PUF. This chapter uses several simulation-based metrics to investigate the parameters affecting the reliability of SUV based SRAM PUFs. Additionally, we characterize the SUV reproducibility proposing several mismatch metrics suitable for reliability estimation during design phases.

3.1 Simulation Environment Setup

The proposed metrics are applied to the common 6T SRAM design using Cadence Environment, see the schematic in chapter 2 in Fig.2.3 (a). The Virtuoso Schematic Editor is utilized to draw the circuit and Spectre Simulator is used to simulate the proposed SRAM design implemented on commercial 65nm CMOS technology. This technology provides three types of transistor based on threshold voltage parameter, such as Low Power Standard Threshold Voltage CMOS (lpsvt), Low Power High Threshold Voltage CMOS (lphvt) and Low Power Low Threshold Voltage CMOS (lpsvt); while in this work the lpsvt CMOS is used. However, the SRAM cell Schematic is powered utilizing a Voltage Piecewise linear (Vpwl) source, as obtaining the SUV of SRAM cell (see section

2.2) requires controlling the source ramp-up voltage and time. Similarly, both Bit-Lines (BL, BLB) and Word-Line (WL) are controlled by same type of voltage source. The Vpwl source allows us to adjust the magnitude and the speed of ramp-up voltage (Vdd) and to turn on and off the BL, BLB, WL.

AL the simulations in this chapter utilize Monte Carlo analysis provided by Spectre Simulator to mimic the inherent mismatch between the transistors inside SRAM cell, also we applied Process Variation feature to mimic different SRAM cells. In this work, we propose 1000 cells SRAM array represented by 1000 Monte Carlo iterations. To have a fixed Monte Carlo iteration order for all simulations in this thesis, the location of the simulated transistors on the Schematic is fixed through this work simulations, as we observed that changing only the transistor location can highly affect the distribution of Monte Carlo; such that a new random parameter values are generated for each iteration. In the following sections, we introduce the new metrics methodology where some of them are evaluated implementing DC-Monte Carlo Simulations while the rest of metrics by Transient-Monte Carlo Simulations. Based on that, the chapter simulations are divided into:

1-DC Monte Carlo Simulations

Section.3.2 presents two metrics to model the inherent-cell mismatch. A metric based on the threshold voltage differences is proposed. In this sense, the Spectre Simulator provides three types of threshold voltage to be calculated: Model *Vth*, Transient *Vth* and DC *Vth*. However, the Model *Vth* is utilized for this metric calculation, as it has a fixed value regardless the type of simulation. The other metric is based on the distance between VTCs for the individual inverters of the SRAM cell. The DC Simulation is mainly

utilized to draw the VTCs, as proposed in [47], for both inverters; where the voltage switching points are calculated. Similarly, the metrics in *Section.3.3* are based on the SNM concept and are evaluated using the same DC-VTCs methodology to draw the SNM-based butterfly diagram.

2- Transient Monte Carlo Simulations

The SUV for each cell is evaluated using transient simulation. At power-up, the cell remains in retention operation, that is, the access transistors are cut-off (WL = 0 V) and the memory cell is isolated from the bit-lines. In this sense, we found that starting-up the cell either with discharged bit-lines (*BL*=0 V, *BLB*=0 V) or with fully charged bit-lines (BL=Vdd, BLB=Vdd) will slightly affect the final SUV for the cells in the proposed memory; as only 0.24% of the cells have different SUV for both cases.

This chapter uses a reference SUVs set for the proposed SRAM array to validate the prediction ability of the proposed metrics. While the internal nodes of the memory cell are discharged by setting the initial nodes conditions to Q = 0 V and QB = 0 V, this reference SUV is obtained by ramping the cell power supply from 0 V to 1.2 V in 5ns under nominal temperature (27° C).

Using similar Transient simulations settings, we obtained the metrics in *Section.3.4* and *Section.3.5.* But, in *Section.3.4,* two Vpwl sources are utilized to inject voltage noise at different locations of SRAM cell aiming to estimate inherent-cell mismatch. While in *Section.3.5,* the *SID* metric methodology is based on varying the initial nodes conditions from the reference values (VQ=0 V, VQB=0V).

Finally, the resulting data from both type of simulations are collected and organized using Open Command Environment for Analysis (OCEAN) Script. This tool is a powerful

programming language that can automate the simulations within Cadence. It is a subset of SKILL language and uses this language to configure the design environment. Then all the resulting data are transferred to MATLAB environment, where we analyzed them to produce the results and figures in this chapter.

3.2 Reliability Metrics based on Inherent-Cell Mismatch

All SRAM cells have inherent mismatch caused by fabrication process variations. In this work, the mismatch in SRAM cell is calculated based on two different transistors and inverters parameters. Generally, the mismatch metric is based on Parameter distance (*Pd*) defined as the difference between some parameter values, S, that accounts for the trend that an inverter has in attaining one of the two possible logic states.

$$Pd = S(inv_Q) - S(inv_{QB}) \tag{3.1}$$

This section introduces two mismatch metrics based on selecting the S to be related to: firstly, the transistors *Threshold Voltage* parameter. Secondly, the inverters *Switching Voltage Point.*

3.2.1 Threshold Voltage Distance Between the Individual Transistors

There are several transistor parameters that may contribute to SUV behavior. When power is applied to the memory, the SRAM cells will reach a final SUV that depends primarily on threshold voltage (*Vth*) mismatch of the constituent transistors [48], [49], [50]. In [49], the authors apply *Vth* mismatch to define the adequate cells for PUF application. Their definition of the mismatch only considers the *Vth* mismatch of N-MOS transistors while the mismatch in P-MOS transistors is neglected. However, the *Vth*

mismatch of N-MOS transistors is controlled by the transistor area; a smaller area generates higher mismatch. On the other hand, the work in [50] only considers the process variation and mismatch in P-MOS transistors to select SRAM cells to be used as a TRNG. In this work, both P-MOS and N-MOS *Vth* mismatch are implemented to define the Threshold Voltage Parameter Distance (Pd_{Vtho}).

Firstly, Monte Carlo simulation is applied to mimic the process variation and mismatch and then a DC simulation in Spectre simulator is utilized to observe and save the Vth for the transistors (Mp3, Mp2, Mn0, Mn1), see Fig.3.1. Secondly, the SUV for each cell of the memory is obtained by transient simulation under the reference conditions as discussed in Section 3.1.

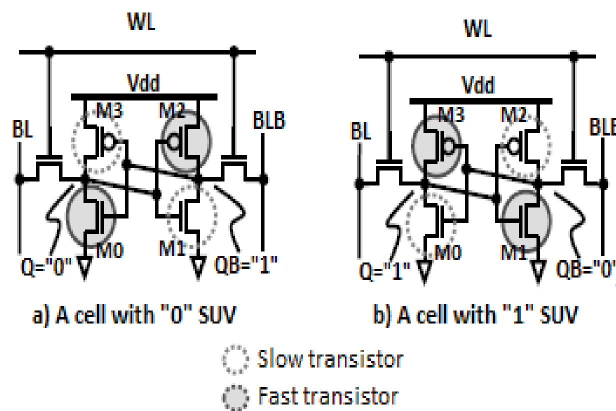


Fig.3.1: *6T SRAM Cell showing transistors contribution to SUV.*

After studying and comparing both simulations, the SUV could be determined by the random Vth variation of the four transistors in each SRAM memory cell as follows:

- A slow transistor means that it has high Vth so its current will be slow. Conversely, a fast transistor means that it has low Vth so its current will be fast.
- A cell will have “0” SUV (Q = 0 and QB = 1 in Fig.3.1 (a)) if Mp3 and Mn1 are slower transistors than Mp2 and Mn0.

- A cell will have "1" SUV (Q = 1 and QB = 0 in Fig.3.1 (b)) if Mp3 and Mn1 are faster transistors than Mp2 and Mn0.

In order to estimate this misalliance between transistor pairs, the Pd_{Vtho} could be defined as the subtraction of Vth attributes from both (Mp3, Mn1) and (Mp2, Mn0) pairs as follows:

$$Pd_{Vth\ o} = \left(Vth_{Mp3} + Vth_{Mn1}\right) - \left(Vth_{Mp2} + Vth_{Mn0}\right) \quad (3.2)$$

If Pd_{Vtho} is negative, the SUV is expected to be "1", while if Pd_{Vtho} is positive, the SUV will be "0". In addition, if N-MOS transistors are matched and equal to the matched P-MOS transistors in cell, the Pd_{Vtho} will be equal to 0 which indicates that cell is matched.

However, the Pd_{Vtho} can be described in terms of differences between the P-MOS and N-MOS transistors. Equation (3.5) rewrites equation (3.2) considering the differences between types of transistors in the cell:

$$\Delta P = Vth_{Mp3} - Vth_{Mp2} \quad (3.3)$$

$$\Delta N = Vth_{Mn0} - Vth_{Mn1} \quad (3.4)$$

$$Pd_{Vth\ o} = \Delta P - \Delta N \quad (3.5)$$

As we mentioned in the previous section, the Spectre simulator provides three types of *Vth* (Model *Vth*, Transient *Vth*, DC *Vth*). Even though they have different values for the same transistor, ΔP and ΔN are equal using all these types as the difference is cancelled. As a result, all of them can be used to calculate Pd_{Vtho}.

The new equations define the mismatch space where the differences between P-MOS transistors (ΔP) and the differences between N-MOS transistors (ΔN) could explain the SUV of the cell. Fig.3.2 represents this mismatch space where ΔP and ΔN results

obtained for each cell from the proposed memory. Cells starting at Q = “1” have been represented in red, while cells starting at Q = “0” are in black.

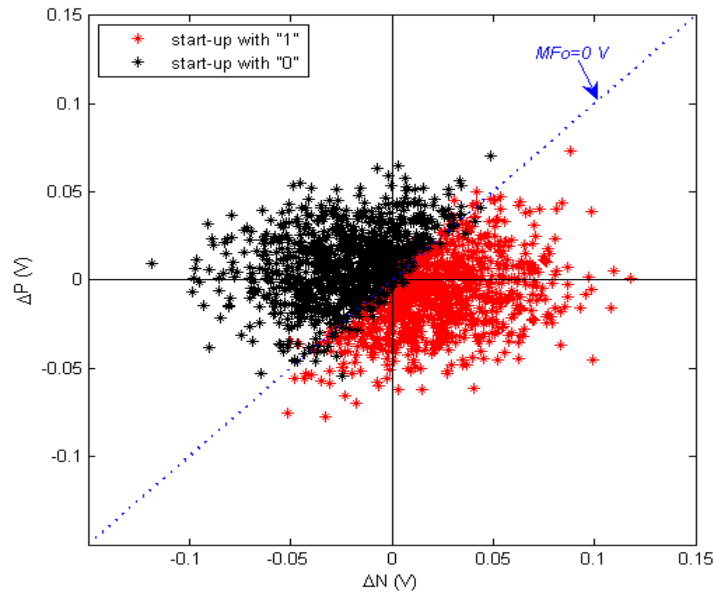


Fig.3.2: *Distribution of Mismatch Parameter between N-MOS and P-MOS.*

In Fig.3.2, most cells with Pd_{Vtho} >0 start at logic “0”, The diagonal line where the colors are separated , equivalent to Pd_{Vtho} = 0, divides the mismatch space in two planes: the plane where (*ΔP, ΔN*) pairs results in $Pd_{Vtho} > 0$, then the SUV is “0” , and the plane where $Pd_{Vtho} < 0$ and the SUV will be “1”. However, there are cells (19% of the memory cells) that show the opposite behavior close to the diagonal line. These cells have an Pd_{Vtho} value whose absolute value is low. In this sense, all cells having a large enough $|Pd_{Vtho}|$ are well classified considering its start-up value.

The histogram distribution of Pd_{Vtho} values has been shown in Fig.3.3, where the cells located at the leftmost (red bars) will start-up with “1” while the cells located at the rightmost (black bars) will start with “0”. The rest of the cells are found to have an unpredictable SUV. As a result, the farther the cells are from Pd_{Vtho} = 0, the more predictable SUV the cells will have. We can define the predictable SUV cells as reliable PUF cells and the unpredictable SUV cells as unreliable PUF cells. Actually, a Pd_{Vtho}

threshold range using Fig.3.3 can be established to differentiate between those reliable and unreliable cells based on a comparison between the SUVs using transient simulations and the sign of Pd_{Vtho} which obtained by DC simulation.

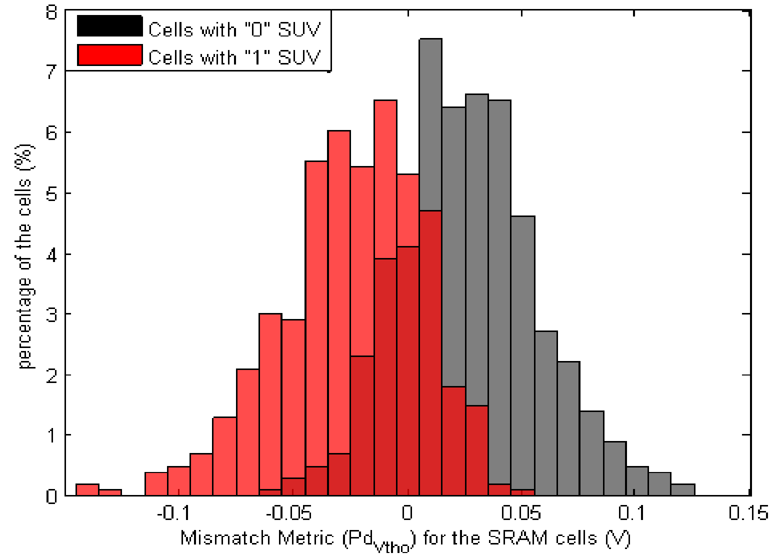


Fig.3.3: *Histograms of Mismatch Metric (Pd_{Vtho}) values depending on the SUV.*

The threshold can be defined from histogram overlaps as [-0.06v, +0.06v]. All cells located outside of this range can be predicted correctly and agree with SUV prediction using Pd_{Vtho} sign.

In general, cells that have low Pd_{Vtho} values produce a high unpredictability in their start-up outputs, unpredictability decreases as Pd_{Vtho} values increases. We observed that cells near $Pd_{Vtho} = 0$ can't be predicted which may indicate that inverters are highly matched, and it is expected to be the higher number of cells in a non-specific SRAM design. These cells may be useful in TRNG applications as it is expected that their outputs will be random.

On the other hand, P-MOS and N-MOS transistors are not equivalent due to structural differences, i.e electron mobility, current drain or internal capacitances. According to previous works [4], [51] and [52], it was reported that P-MOS and N-MOS transistors

have different contribution on the SUV of SRAM cell. Specifically, their results show that the P-MOS transistors dominate more in deciding the SRAM-PUF final output.

Based on that, we include a Weighting factor (*w*) comprised between 0 and 1, to account for this difference in contribution. So, equation (3.5) can be written as:

$$Pd_{Vth} = w * \Delta P - (1 - w) * \Delta N \tag{3.6}$$

where *w* is defined by implementing an optimization process to adjust the Pd_{Vtho} (obtained by DC Monte Carlo simulation) based on the SUV (which obtained by transient Monte Carlo simulation). This optimization process aims to maximize the number of cells that have SUV agree with the sign of Pd_{Vtho}. After the optimization fitting process, the *w* value is optimized equal to 0.76. As a result, This Pd_{Vth} will combine both DC and Transient simulation, which can be useful to study and compare the mismatch with any other transient simulations. The histogram of Pd_{Vth} values is presented in Fig.3.4**,** it can be noticed that the overlap range ([-0.005v, +0.005v]) decreases where the percentage of unpredictable cells decreases from 19% to only 5% of the memory cells. These results

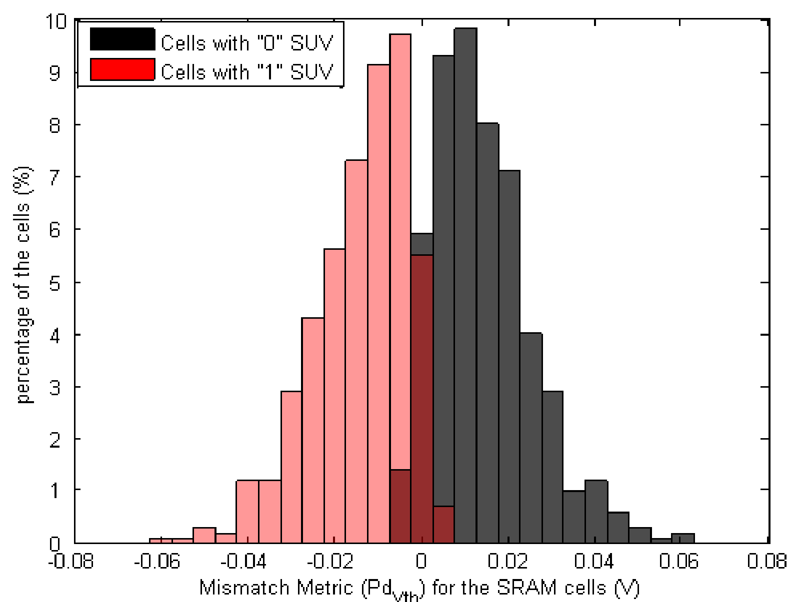


Fig.3.4: *Histograms of Mismatch Metric (Pd_{Vth}) values including the Weighting Factors (w=0.76).*

agree with previous mentioned works, as if the P-MOS transistors have higher contribution (w=0.76) than N-MOS contribution (1-w=0.24) the SUV predictability will be higher. And thus P-MOS will be more dominant on SUV for SRAM-PUF.

In this thesis, Pd_{Vtho} and Pd_{Vth} will be implemented as a key factor to explain different SRAM-PUF behavior under internal noise effects, such as thermal noise, and external conditions effects, such as ramp-up time and internal nodes initial conditions. Also, they will be used as a reference-metrics to be compared with other proposed metrics in the next sections.

3.2.2 Switching Voltage Point Distance between Inverters

The previous Pd_{Vtho} only considers the threshold voltage to evaluate the mismatch in SRAM cells. In order to include the contribution of all physical parameters to obtain the cell mismatch, a novel-method is proposed for the first time to calculate the cell mismatch. This method implements the Inverter Switching Point (V_M) for both cell's inverters.

V_M is defined as the point where $V_Q = V_{QB}$ ($V_{in} = V_{out}$). At this point, both P-MOS and N-MOS transistors are always saturated, because $V_{DS} = V_{GS}$. An analytical expression for V_M is calculated by equating the currents passing through the transistors in the inverter. The final expression for V_M is shown in the following equation [53]:

$$V_M = \frac{\sqrt{\frac{\beta_n}{\beta_p}}\, V_{th,N} + V_{dd} - V_{th,P}}{1 + \sqrt{\frac{\beta_n}{\beta_p}}} \tag{3.7}$$

Where $$\beta = \mu_o * C_{ox} * \frac{W}{L} \tag{3.8}$$

In the last equation, μ_0 is the average carrier mobility, C_{ox} is the gate oxide capacitance per unity area, W is the channel width and L is the channel length.

The proposed methodology will implement V_M for both cell's inverter, where V_{M1} is defined for the first inverter, which include Mp3 and Mn0 transistors (see Fig.3.1), while V_{M2} is defined for the second inverter that include Mp2 and Mn1 transistors (see Fig.3.1). The following equations show both V_M:

$$V_{m1} = \frac{\sqrt{\frac{\beta_n}{\beta_p}}\, Vth_{MN0} + V_{dd} - Vth_{MP3}}{1 + \sqrt{\frac{\beta_n}{\beta_p}}} \tag{3.9}$$

$$V_{m2} = \frac{\sqrt{\frac{\beta_n}{\beta_p}}\, Vth_{MN1} + V_{dd} - Vth_{MP2}}{1 + \sqrt{\frac{\beta_n}{\beta_p}}} \tag{3.10}$$

The novel mismatch metric (Pd_{Vm}) is defined for each cell as the difference between both inverters V_M as follows:

$$Pd_{Vm} = V_{m2} - V_{m1} \tag{3.11}$$

To observe the analytical relation between this factor and the previous Pd_{Vtho}, equation (3.11) is rewritten in terms of ΔP and ΔN (equations (3.3), (3.4)) as follows:

$$\mathrm{Pd_{Vm}} = \frac{1}{1 + \sqrt{\frac{\beta_n}{\beta_p}}}\,\Delta\mathbf{P} - \frac{\sqrt{\frac{\beta_n}{\beta_p}}}{1 + \sqrt{\frac{\beta_n}{\beta_p}}}\Delta\mathbf{N} \tag{3.12}$$

On one hand, we can notice that this equation is similar to Pd_{Vtho} (equation (3.5)) in terms of Vth mismatch. On the other hand, Pd_{Vm} includes the effect of other physical parameters mismatch, such as carrier mobility, gate oxide capacitance and area of the transistors.

A graphical technique to obtain V_M value for an inverter is presented in [54], where its value can be obtained graphically by finding the intersection of the inverter VTC with the line given by Vin = Vout. In this work, we have obtained V_{M1} and V_{M2} graphically for both inverters in the cell; while the feedback between the inverters is disconnected. Fig.3.5 shows two VTCs for one SRAM cell where the intersection points with V_Q=V_{QB} line is labeled with V_{M1} and V_{M2}. The distance between these two points is defined as our novel Pd_{Vm}.

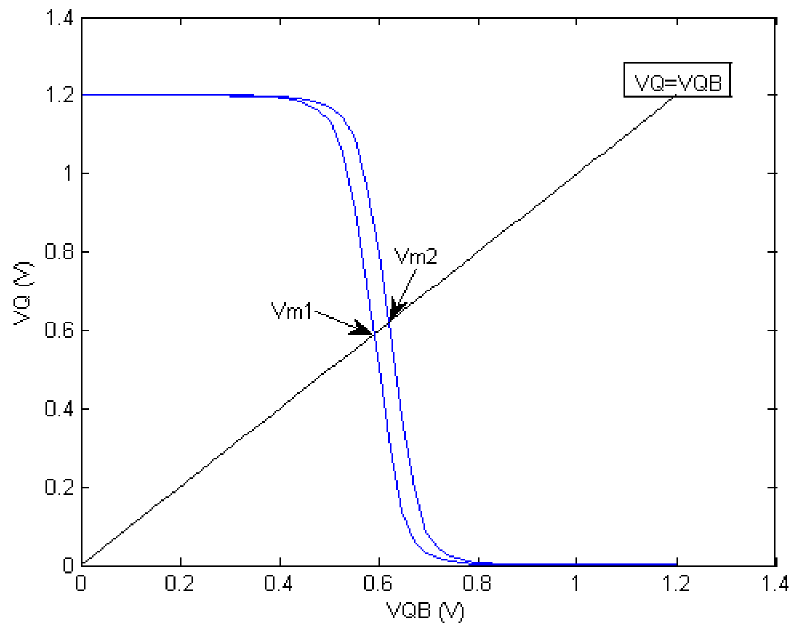


Fig.3.5: *Graphical technique to obtain Mismatch Metric (Pd_{Vm}) values by using VTCs for the cell's inverters.*

The Pd_{Vm} values for each cell of our proposed memory are achieved by DC Monte Carol simulation to draw the VTCs. Also, we have used the same Reference SUV for the memory to compare it with the sign of Pd_{Vm}. The comparison shows that most of the

cells that have negative Pd_{Vm} value start-up with logic "1" while the cells with positive Pd_{Vm} value will start with logic "0".

This SUV behavior agrees with the previous Pd_{Vtho} SUV-assumption, where the histogram of Pd_{Vm} values has been shown in Fig.3.6 , we can see that the cells placed at the leftmost (red bars) will start-up with '1', while the cells placed at rightmost (black bars) will start with '0'. The rest of the cells have an unpredictable SUV (represent 20.8% of whole proposed memory).

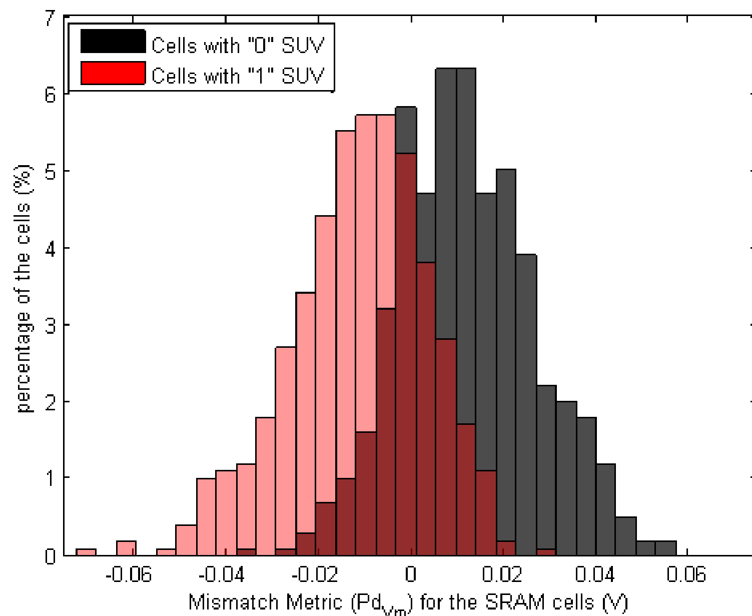


Fig.3.6: *Histograms of Novel Mismatch Metric (Pd_{Vm}) values depending on the SUV.*

As a result, the farther the cells are from Pd_{Vm} = 0, the SUV of the cells will be more predictable. Similar to Pd_{Vtho}, We will define the predictable SUV cells as reliable PUF cells and the unpredictable SUV cells as unreliable PUF cells while the threshold value that differentiate between those reliable and unreliable cells can be defined from histogram overlaps as [-0.03v, +0.03v]. All cells located outside of this range can be predicted correctly.

3.2.3 Summary

The inherent mismatch in SRAM cell can highly affect the SUV of SRAM-PUF. Implementing both Pd_{Vtho} and Pd_{Vm} to study the cell mismatch reflects in that both metrics are able to predict the PUF-SUV behavior. The highest value of these metrics means that the cell is highly mismatched and thus will be more reliable. On the other hand, Pd_{Vtho} has slightly better predicting ability (81%) than Pd_{Vm} (79.2%) which indicates that only the *Vth* of the cell's transistor can significantly control the cell SUV.

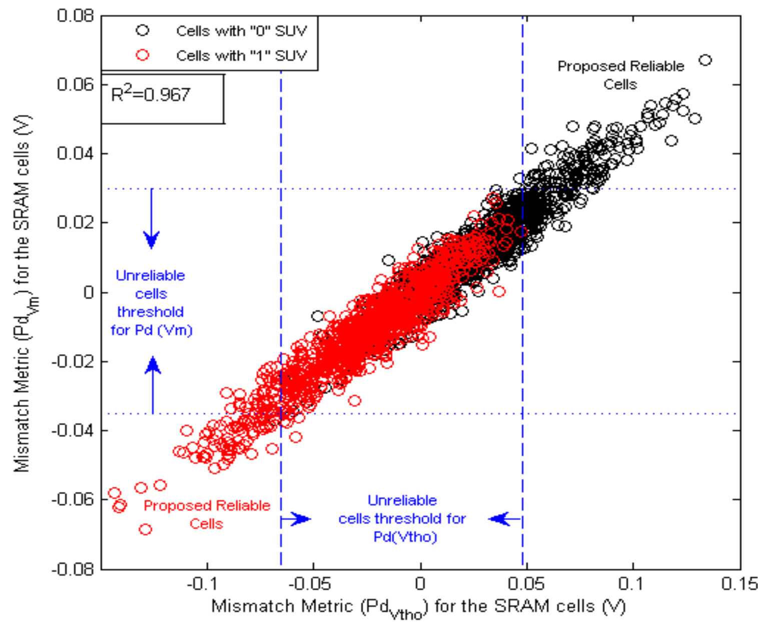


Fig.3.7: *The Relationship between* Pd_{Vtho} *and* Pd_{Vm}.

The relation between both metrics has been shown in Fig.3.7. Where each "o" represents one cell while the cells that colored black have "0" SUV and red cells have "1" SUV. Also, the threshold value between reliable and unreliable cells for each metric is shown in this figure. We can observe that both metrics are correlated together where the correlation factor between them equals to 0.967. Finally, both Pd_{Vtho} and Pd_{Vm} metrics can be obtained easily by DC simulation, while obtaining Pd_{Vth} (Pd_{Vtho} with weighting factors) requires implementing both DC and Transient simulations. This Pd_{Vth} will be very useful to study and analyze the relations between the inherent mismatch

and other external perturbations and internal noise that can be only achieved by Transient simulations.

3.3 Reliability Metrics Based on SNM Concept

This section describes the implementation of the Static Noise Margin (SNM) concept to provide a representation for the SUV behavior of SRAM-PUF. Firstly, the SNM concept is discussed. Secondly, two SNM-based metrics are proposed to estimate inherent-cell mismatch. Thirdly, the relation between the proposed metrics and the previous metric in the literature will be discussed. Finally, the mismatch relation with the SNM proposed metrics will be studied.

3.3.1 SNM Concept

The SNM of 6T SRAM cells is typically utilized to describe a cell internal node noise immunity. Specifically, this metric is used to quantify the maximum noise voltage that could be tolerated by SRAM cell without changing its logic state. SNM can be determined by measuring the side length of the largest square that can fit inside the butterfly curve of the VTCs for the SRAM-cell's back-to-back inverters. Fig.3.8. shows both inverters VTCs for two SRAM cells. Each cell is shown by a different color, and the intersection of those curves create the butterfly curve (eyes shape). The side length of the biggest square that could be located inside both eyes of this curve (see Fig.3.8) is the SNM value [55]. The size of each eye of the butterfly curve can vary between cells (see Fig.3.8), as this variation could be caused by mismatch and process variation in the SRAM cells [56].

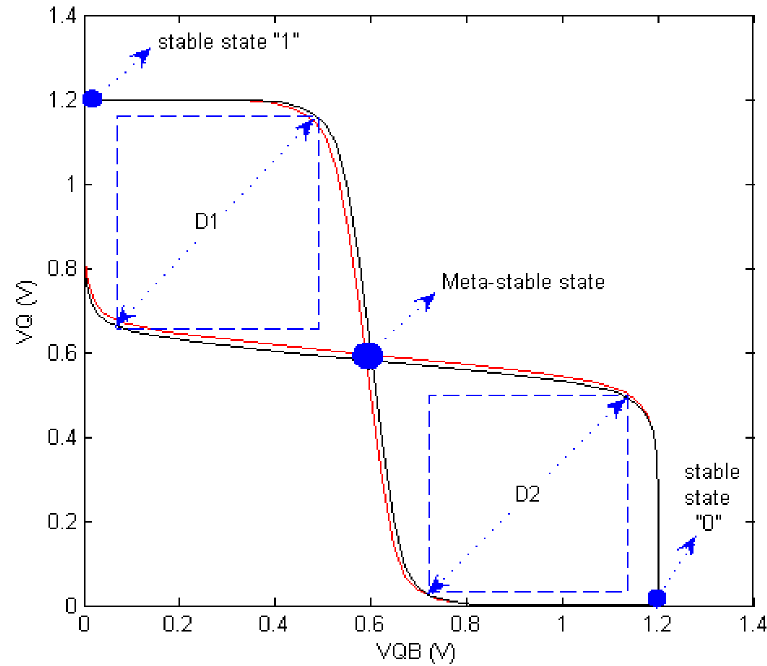


Fig.3.8: *Static Noise Margin definition using VTCs curves.*

For normal SRAM operation (Read, Write, Hold), the SNM model in [55] considers the effect of all cell's transistors on the stability of SRAM cell. The model in [55] calculates the SNM based on Read Mode because it's the worst mode scenario in normal SRAM applications.

3.3.2 Proposed Metric

Modeling the SUV behavior for SRAM-PUF using SNM was introduced as a reliable PUF metric in [43], also the same model is utilized and discussed in [57-58]. The authors in [43] claim that the previous SNM model cannot be directly implemented to analyze the SUVs behavior, as the SUV is generated when the cell is in hold mode and not in read mode. Thus, they defined two metrics based on obtaining the noise margins (NM and NM') of the VTCs. The NM and NM' metrics are calculated by the values of four critical points, located on VTCs, where the derivative of Q node voltage with respect to QB node voltage is equal to -1 [43]:

- PSNM ratio defined as the ratio between noise margins (NM / NM').
- PSNM noise defined as the minimum value of both noise margins (min (NM, NM')).

Where the preferred SUV of the cell is logic "1" if PSNM ratio is higher than 1, and logic "0" if PSNM ratio is lower than 1. As the cells have values of PSNM ratio are higher than 1, they will have higher asymmetry between their back-to-back inverters; and thus, their SUV will be more reliable.

Although the traditional SNM (as described in the previous section) in [55] is evaluated utilizing read mode conditions, the way how it is evaluated can also be applied in hold mode configuration to study the SUV for PUF; in other words, when Word Line (WL) = 0. Therefore, the noise margins can also be defined using both length sides of both squares that can fit inside the butterfly curve of the VTCs.

3.3.2.1 SNM Distance (SNM_d) Metric

Recently, the work in [47] implements DC simulation method to calculate SNM. This method is based on rotating the VTCs of the memory cells 45° to obtain the diagonals of each largest squares that fit inside the butterfly curve (D1 and D2 in Fig.3.8). Taking profit of this method, our work introduces a new metric, denoted as SNM Distance (SNM_d), to represent the start-up behavior of the SRAM cell. Therefore, the VTCs in Fig.3.8 is rotated 45° by multiplying each VTC by a rotating vector to become like in Fig.3.9. Hence, the SNM_d is defined as the subtraction of both diagonals D1 and D2 (see Fig.3.9) as follows:

$$SNM_d = D_1 - D_2 \tag{3.13}$$

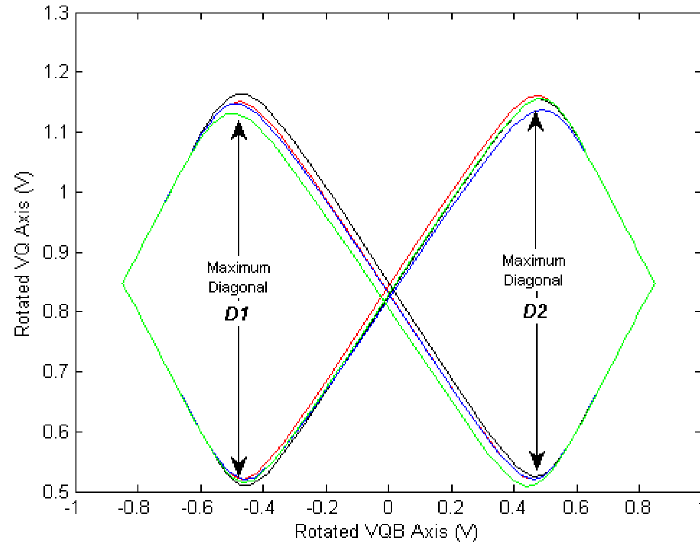


Fig.3.9: *45° rotated VTCs.*

The SNM_d values for each cell of our proposed memory are achieved by DC Monte Carlo simulation to draw the VTCs, while MATLAB platform is implemented to rotate these curves and evaluate the maximum diagonals as in Fig.3.9. Also, we have used the same reference SUV to compare it with the sign of SNM_d. The comparison shows that most of the cells that have negative SNM_d value start-up with logic "0" while most of the cells with positive SNM_d value will start with logic "1".

The assumption of SUV direction (logic "0" or "1") using this metric is reversed when it is compared with parameter distance metrics (Pd_{Vtho} and Pd_{Vm} in section.3.2); a positive value of parameter distance metrics defines the cell that start-up at logic "0", and a negative value for the cell that start-up at logic "1". The histogram of SNM_d values has been shown in Fig.3.10 , where we can see that the cells placed at the leftmost (black bars) will start-up with '0', while the cells placed at rightmost (red bars) will start with '1'. The rest of the cells have an unpredictable SUV (representing 22.4% of the proposed memory).

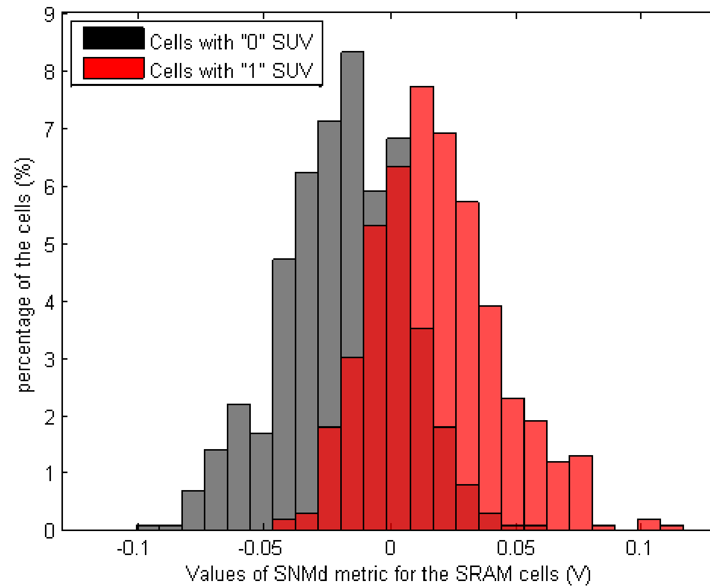


Fig.3.10: *Histogram of SNM_d values considering the reference SUV.*

It is noticeable that, the farther the cells are from $SNM_d = 0$, the SUV of the cells will be more predictable. Similar to the parameter distance metrics, the cells that have predictable SUV will be defined as reliable PUF cells, while the cells that have unpredictable SUV as unreliable PUF cells. Also, a threshold range from histogram overlaps can be established to differentiate between those reliable and unreliable cells as [-0.05v, +0.05v]. All cells located outside of this range can be predicted correctly.

Our results agree with the works in [21, 43], where they found that a totally symmetrical eyes of butterfly curve indicates that the cell inverters are symmetrical and thus the cell will not has a preferred logic state ("0" or "1"). While large butterfly curve eyes asymmetry indicates a high tendency towards one of the preferred logic states. Hence, the proposed SNM_d evaluates these asymmetries.

3.3.2.2 VTCs Intersection Distance (INTd) Metric

The Butterfly diagram of an SRAM cell can provide much information about the start-up behavior of SRAM. In this section anther novel metric to classify the SRAM cells strength is proposed based on the intersection points between the VTCs of the cell. Generally, both cell VTCs intersect in three points: stable state “1”, stable state “0” and meta stable point; as shown in Fig.3.8. On the one hand, for a perfectly symmetrical cell, the meta stable point is located on the line VQ=VQB, which represents the ideal value of meta-stable state where the cell inverters are fully matched. On the other hand, for non-symmetrical cells, the farther that the meta-stable point is from the VQ=VQB line, the more asymmetrical the cell will be, and thus the cell’s inverters are more mismatched. Based on that, our novel metric, denoted as INT_d (see Fig.3.11) is defined as the distance between the meta-stable point (VTCs intersection point) and the VQ=VQB line. Fig.3.11 represents the VTCs of two different SRAM cells including a detailed picture for their intersection points where our novel metric is shown. On one

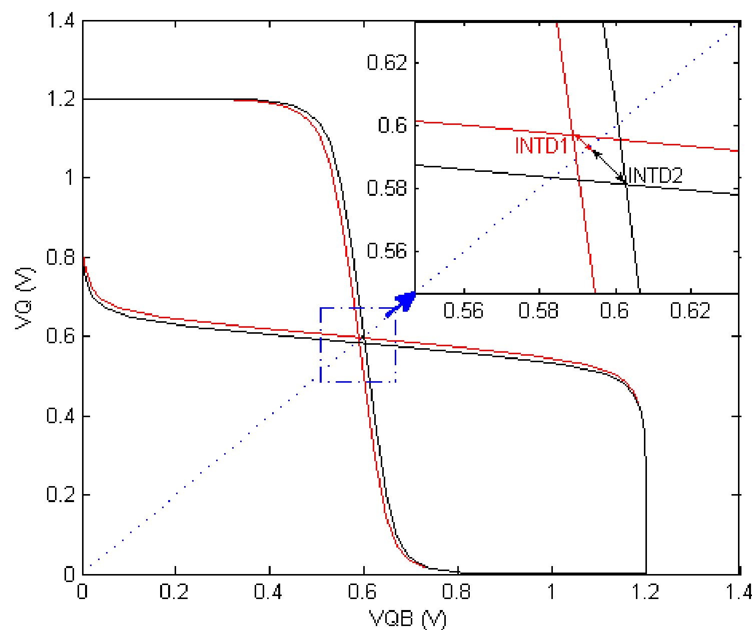


Fig.3.11: *Intersection Distance metric (INT_d) definition using VTCs curves.*

hand, the red curves in this figure represent a cell that has a logic "1" SUV, where the its intersection point is located above the line VQ=VQB. On the other hand, the black curves present a cell start-up at logic "0", where the intersection point of this cell is located under the line VQ=VQB.

Based on these observations, we define positive INT_d value for the cells with intersection point located above the VQ=VQB line (see Fig.3.11), while a negative INT_d value is be assigned for the cells with intersection point located below the VQ=VQB line (see Fig.3.11). In addition, from a reliability point of view, the INT_d magnitude is related to the cell's asymmetry. A higher INT_d magnitude means that the cell will be more mismatched and thus more reliable; where in Fig.3.11, the cell represented by the black curves (the cell with INT_{d2}) is more reliable than the red curves cell (the cell with INT_{d1}).

The INT_d values for each cell in the proposed memory are achieved by using similar procedure as in SNM_d metric to obtain the VTCs, while the intersection of these curves is calculated in MATLAB platform. Again, the same Reference SUV set for the proposed memory is utilized to compare the SUV for each cell with the sign of INT_d.

The histogram of INT_d values is shown in Fig.3.12, we can observe, similar start-up behavior to SNM_d metric. The cells located at the leftmost (red bars) will start-up with '1', while the cells placed at rightmost (black bars) will start with '0'. The rest of the cells that are located near to INT_d =0 have an unpredictable SUV (represent 20.9% of all cells). The results in this histogram are in line with our reliability assumption using INT_d metrics. The farther the cells are from INT_d = 0, the SUV of the cells will be more predictable, and they can be defined as reliable PUF cells. while the cells with low INT_d absolute value have unpredictable SUV and they can be defined as unreliable PUF cells. Also, a

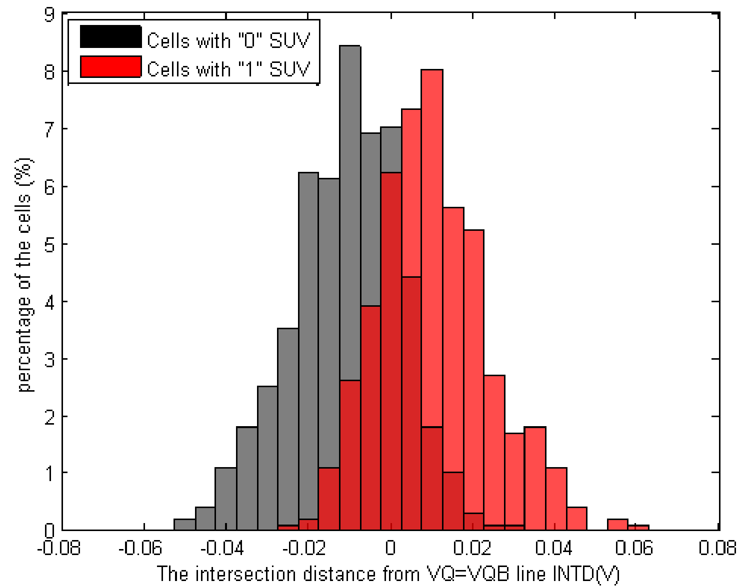


Fig.3.12: *Histogram of INT_d values considering the reference SUV.*

threshold range that differentiates between those reliable and unreliable cells can be defined as [-0.03v, +0.03v], where all cells located outside of this range can be predicted correctly.

3.3.3 Correlation with Previous Metric in literature (PSNM ratio)

SNM is typically applied to study a cell's internal node noise immunity. The variation of the size of the butterfly curve of a SRAM cell is utilized to evaluate the SNM. In this work, two proposed metrics (SNM_d and INT_d) based on SNM are implemented to study the reliability of SUV of SRAM-PUF. A high absolute value of these metrics means that the cell is highly asymmetrical and thus will has high tendency to one of preferred SUV, also it will be more reliable PUF cell.

Based on our simulations, we noticed that INT_d has better SUV predicting ability (79.1%) than SNM_d metric (77.6%). The relation between these two metrics has been shown in Fig.3.13. Where each "*" represents one cell, while the cells colored in black also have "0" SUV and red cells have "1" SUV. Also, in this figure, the threshold range between

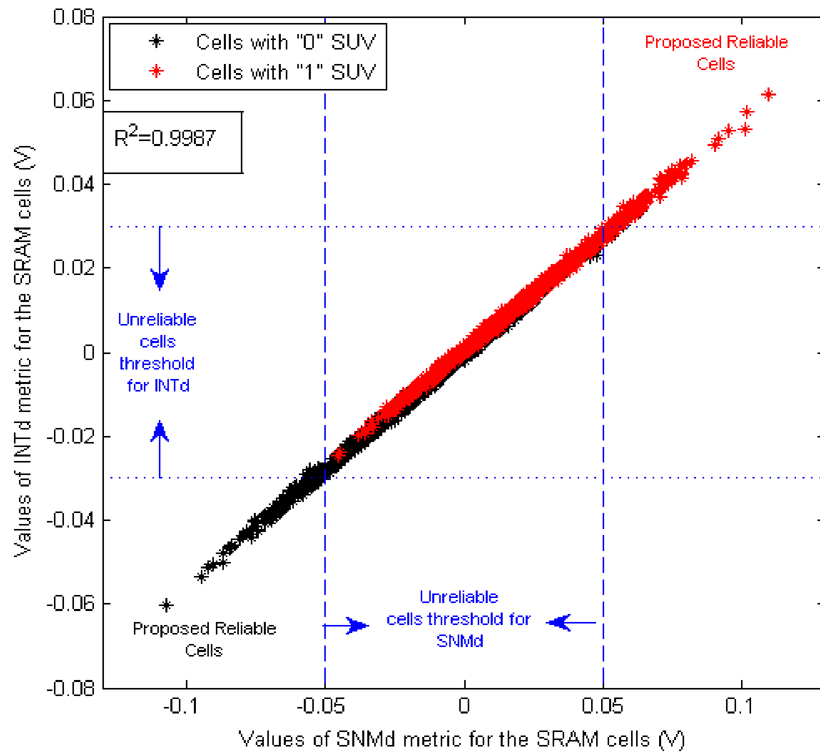


Fig.3.13: *Correlation between values of SNM-based metrics.*

reliable and unreliable cells is presented by the vertical lines for SNM_d metric, while the horizontal lines represent the threshold range for INT_d. We can observe that most of the reliable cells are common between both metrics. Additionally, the proposed metrics are highly correlated where the correlation factor between them reaches up to 0.9987.

Finally, the correlations between our proposed metrics SNM_d and INT_d with the PSNM ratio metric in [43,57-58] are presented in Fig.3.14(a) and Fig.3.14(b), respectively. Where the PSNM ratio metric was the only metric in literatures applied to model the reliability of SUV for SRAM-PUF. In these figures, it is remarkable that our proposed metrics demonstrate good correlation with PSNM ratio by achieving linear coefficients superior to 0.99. Therefore, the new proposed metrics appear to be suitable for SRAM-PUF cell classification.

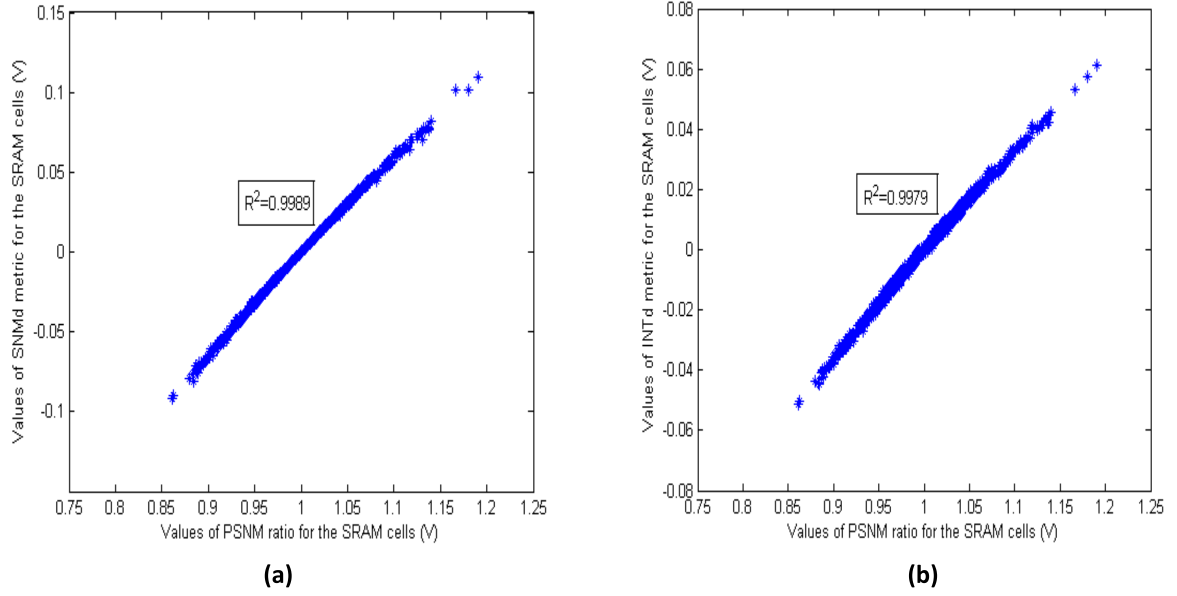


Fig.3.14: *Correlation between our proposed metrics and PSNM ratio.*

3.3.4 Correlation with Inherent-Cell Mismatch

SNM is typically utilized to describe a cell's internal node noise immunity, while the mismatch of the cells describes the effect of manufacturing variability on the cell's transistors and thus their relative strength. In this section, the relation between the mismatch and the SNM metrics will be studied. The Pd_{Vm} metric will be utilized to study this relation; as Pd_{Vm} metric is also evaluated by implementing VTCs of the cell.

The relation between SNM_d metric and the mismatch is presented in Fig.3.15, where the reversed correlation coefficients between them R^2 equals to -0.9987. Also, a slightly better relation is shown in Fig.3.16 between INT_d and the mismatch, where the reversed correlation coefficients between them is very high and equals to -0.9998. In Fig.3.15 and Fig.3.16, each "*" represents one cell, while the cells that colored black have "0" SUV and red cells have "1" SUV. Also, in these figures, the reliable and unreliable cells threshold ranges for the metrics are shown.

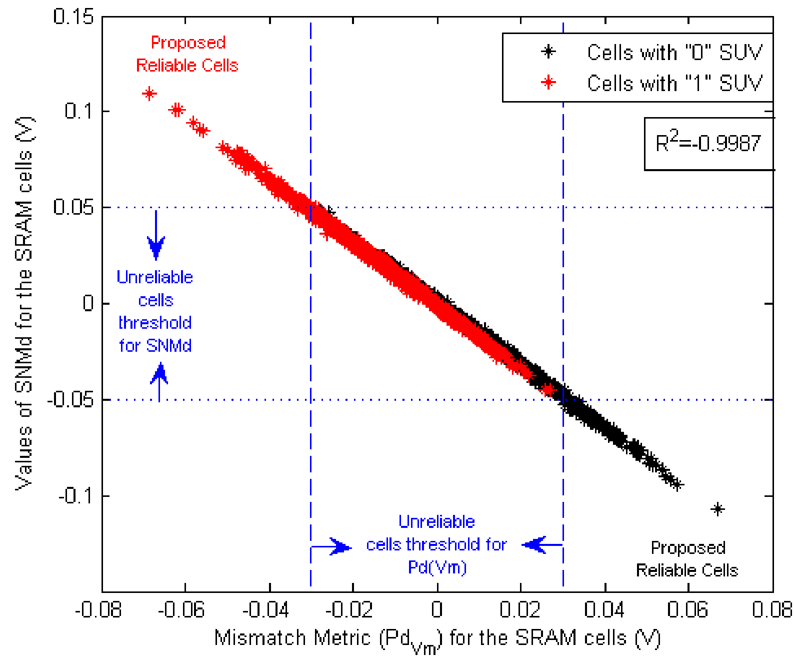


Fig.3.15: *Correlation between values of SNM_d and the mismatch metric.*

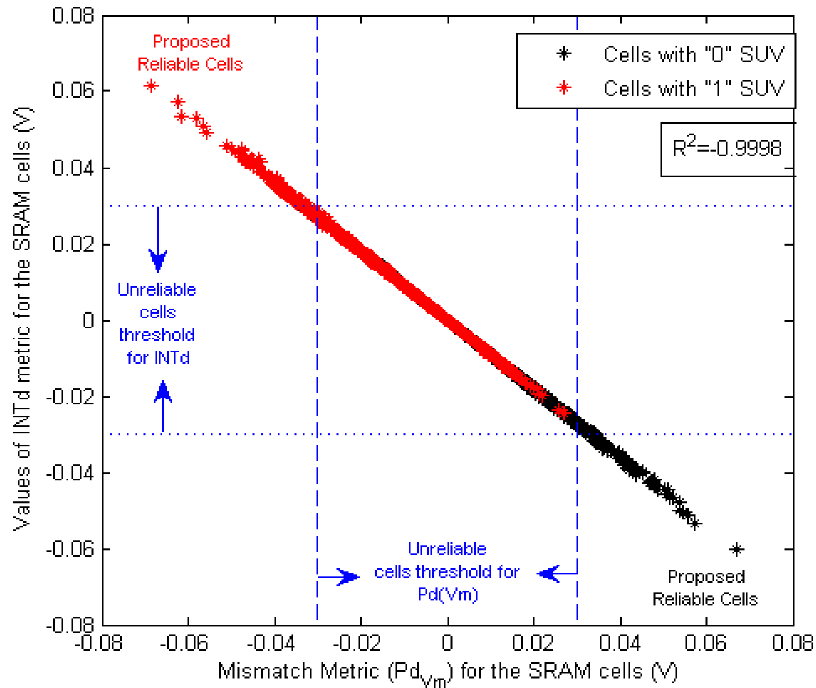


Fig.3.16: *Correlation between values of INT_d and the mismatch metric.*

Additionally, we have observed that the percentage of common reliable cells between INT_d metric and the mismatch is slightly higher than the common reliable cells between SNM_d and the mismatch. Finally, the reverse in the linearity relation between the SNM

metrics and the mismatch represents only the direction of the SUV (preferred “0” or “1”). However, the cells with high absolute value of SNM metrics have a high absolute value of mismatch metric. This indicates that a highly mismatched cell can tolerate high level of noise and thus it will be considered as a reliable cell.

3.4 Voltage Noise Injection Methodology as Reliability Metrics

There are many researches that focus on improving and implementing test strategies for SRAM-PUF in order to enhance the reliability of the response produced by the PUF cells. Those researches also try to minimize cost, time and hardware of the tests [42], [44] and [59].

In this work, the start-up behavior of SRAM-PUF has been widely studied and we have observed that threshold voltage variations have very high impact on the SUV. An important factor that causes inconstant behavior in SUV is the bit-induced noise voltage such as thermal noise [52]. The work in [59] proposes a preprocessing algorithm to overcome this inconstant start-up behavior that is caused by the noise-induced in the SRAM cells. Also, postprocessing fuzzy extractors can be implemented as in [60] to reduce the effect of that behavior. However, these solutions aren’t practical as the complexity and cost will be increased [42].

A proposed SRAM-PUF design that could be used to identify the unreliable cells that are highly affected by bit-induced noise is presented in [42]. In this design, low cost modifications for the SRAM cell are required. This work relies on injecting a certain level voltage noise at the ground of the proposed memory to identify the cells that show inconstant SUV. Those cells will be masked out from the PUF response while the rest of the cells in the memory will be considered as reliable cells for PUF application.

In this Section, a metric-based methodology, based on injecting a DC voltage at different location of the SRAM cell, is proposed to classify the immunity of the memory cells against the induced noise. This work proposes three metrics that aim to evaluate the maximum voltage noise that can be tolerated by each cell in the proposed memory. On one hand, the three proposed metrics are evaluated using the same methodology. On the other hand, the location of the injected voltage noise between those metrics is different.

Firstly, we will present the methodology of evaluating the maximum tolerated noise by the cells according to the following locations: (i) The injected noise at the ground node of the cell, (ii) The injected noise between the cell's storage nodes and (iii) The injected noise at the power supply node of the cell. Then, the relation between those proposed metrics will be discussed. Finally, the noise injection-based metrics will be studied with respect to the inherent cell-mismatch, where we will show that the proposed metrics can provide a significant indication on cell's mismatch.

3.4.1 Metric Methodology

3.4.1.1 Noise Injection at Ground of the cell

The injection of DC noise voltage into SRAM memory is proposed in [42], as a technique to characterize the reliability of the cells for PUF applications. In that method, two voltage noise sources are added to the ground terminals of the SRAM array as it can be seen in Fig.3.17. The goal of that work is to obtain a ratio for those cells that change its start-up value under certain level of the injected noise.

However, our approach also consists in adding two DC voltage noise sources, but they will be added to each cell ground node of the proposed memory. Therefore, we will be

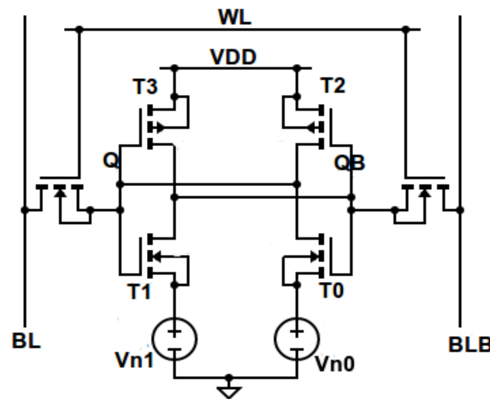


Fig.3.17: *6T SRAM cell with noise injection at the ground nodes [42].*

able to find the maximum DC noise voltage value that can be tolerated by each cell without changing its start-up value. Fig.3.17 shows the SRAM cell setup with DC noise voltage sources, Vn_0 and Vn_1, connected to the ground nodes of the cell. The first observations regarding to SUV are:

- The cells that originally start-up at logic “0” (VQ=0, VQB=1) require varying Vn_1 source to change their SUV to logic “1”; see Fig.3.17.

- The cells that originally start-up at logic “1” (VQ=1, VQB=0) require varying Vn_0 source to change their SUV to logic “0”; see Fig.3.17.

To achieve a specific value for each cell that determines the ability to tolerate the maximum injected voltage noise, we utilized transient-Monte Carlo simulations to mimic the process variation between cells and the mismatch inside each cell. The procedure that we have implemented is described as follows:

- On one hand, if the cell starts-up at logic “0”, the voltage of Vn_1 source will be increased starting from 0 V in steps of 5 mV until the value where the cell

flips its SUV. Meanwhile, the voltage of Vn_0 source will be kept to 0 V; as varying Vn_0 voltage in this case will not cause a flip in the SUV.

- On the other hand, if the cell start-up at logic “1”, the voltage of Vn_0 source will be increased similarly until the value where the cell flips its SUV. Meanwhile the voltage of Vn_1 source will be kept to 0 V; as varying Vn_1 voltage in this case will not cause a flip in the SUV.
- For each simulation set in each previous case, we have determined the lowest noise voltage level which is able to change the start-up value for each cell. Therefore, the maximum noise that can be tolerated by each cell, denoted as Vn_g, is defined as:

$$Vn_g = Vn_1 - Vn_0 \tag{3.14}$$

Note that, a positive Vn_g values are assigned to the cells that have an original SUV at logic “0” (minimum noise voltage that applied to Vn_1 source to flip the SUV to ”1”), and a negative Vn_g value to the cells that have an original SUV at logic ”1” (minimum noise voltage that applied to Vn_0 source to flip the SUV to ”0”).

Fig.3.18 shows the histogram of the Vn_g values, for the proposed memory, computed by Monte Carlo simulations. Implementing the Vn_g definition and observing the histogram, all cells represented by the bars at the positive x-axis have logic “0” SUV while the cells represented by the bars at the negative x-axis will start-up at logic ‘1’. However, cells located at the leftmost and rightmost bars of this histogram have high absolute Vn_g values, and thus they can tolerate higher amount of noise level without changing their SUV, so they will be defined as reliable cells.

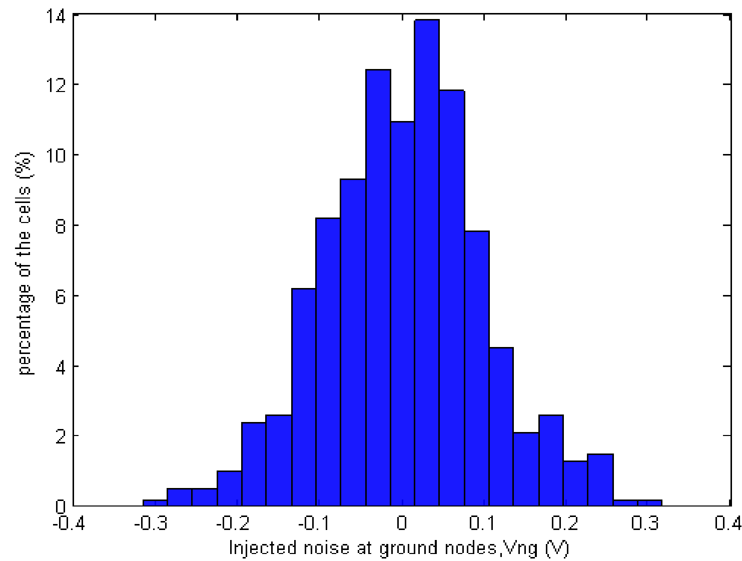


Fig.3.18: *The histogram distribution of Vn_g metric.*

3.4.1.2 Noise Injection at Storage Nodes of the cell

In this section, another metric based on injecting a DC voltage noise between the internal storage nodes of SRAM cell will be proposed for the first time to classify the immunity of the PUF cells against internal noise. The location of the injected noise is based on well-known SRAM stability metrics measurement, where the Read Noise Margin (RNM) in [55] and the Write Noise Margin (WNM) in [61] have utilized similar noise injection location to be evaluated. The goal of our proposed metric is to investigate if the location of the injected noise has any impact on the classification of the memory cells for PUF application. Fig.3.19 presents the SRAM cell setup, where the same DC noise voltage sources that have been used in the previous Vn_g metric, Vn_0 and Vn_1, are connected between the cell's internal nodes.

This metric also assigns a specific value to each cell in the memory by implementing a similar procedure as in the previous Vn_g metric. One of the noise sources is varied while the other is kept at 0 V. In this case, the maximum DC noise voltage that can be tolerated

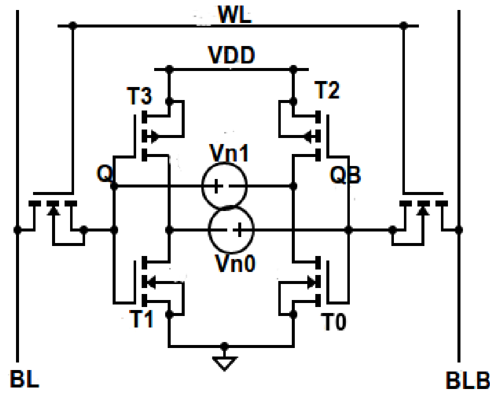


Fig.3.19: *6T SRAM cell with noise injection between storage nodes.*

by each cell, when the noise is injected between its internal nodes, denoted as Vn_i, is defined as follows:

$$Vn_i = Vn_1 - Vn_0 \tag{3.15}$$

An injected voltage higher than this Vn_i value, will change the cell SUV. Similar to Vn_g definition, positive Vn_i values are associated with the cells that originally have logic "0" SUV, and a negative Vn_i values with those cells that have logic "1" SUV.

The histogram for Vn_i values for the proposed memory is presented in Fig.3.20. Again, the reliable cells that are identified by Vn_i are the cells placed on the rightmost and

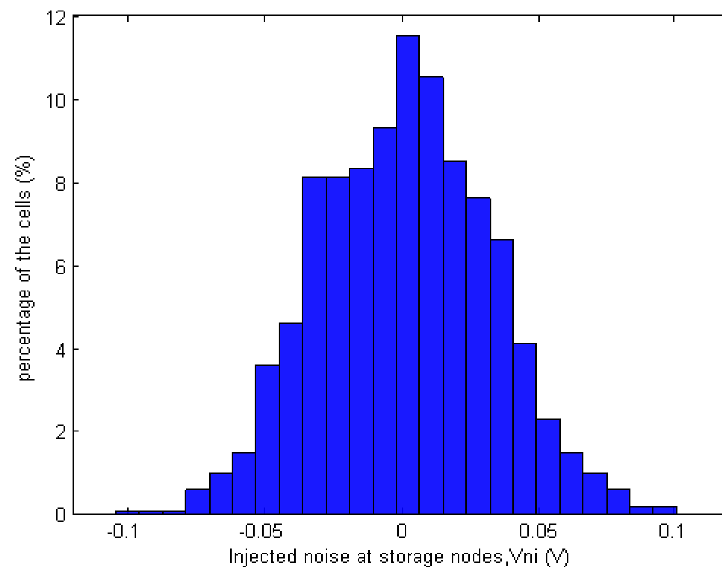


Fig.3.20: *The histogram distribution of* Vn_i *metric.*

leftmost bars on this histogram. Those cells have high absolute values of Vn_i and thus can tolerate high level of the injected noise at the storage nodes.

3.4.1.2 Noise Injection at Cell Power Supply Nodes

The variation in the power supply voltage can slightly affect the start-up behavior of an SRAM cell [62-63]. While in [44], the authors separate the power supply (Vdd) of the cells into two power supplies (VddL, VddR) for each inverter in the memory's cells. In their methodology, they introduce a small DC voltage difference (positive and negative) between the two power supplies, the cells that show unstable start-up behavior will be masked out from the PUF response. Their method can only define a ratio for the unreliable cells in the memory under specific value of voltage difference between the sources, but it can't classify the reliability of the individual cells.

In this section, the third metric, based on injecting a voltage noise at the power supply nodes, will be proposed to classify the reliability of the individual cells. Also, this metric will allow us to compare and study the effect of injected noise location with the previous two locations (at the ground and between the internal nodes of the cell). In this case, we have covered three possible locations where the voltage noise could be injected, with goal to compare between impact of P-MOS and N-MOS transistors on cells noise immunity; as we will discuss in the next section.

Similar voltage noise sources (Vn_0, Vn_1) are utilized to evaluate the third metric, where the SRAM cell setup with the two noise sources located at the power supply nodes is shown in Fig.3.21. Also, similar simulations are utilized to assign a value to each cell, where this value presents the maximum voltage noise that can be injected into the

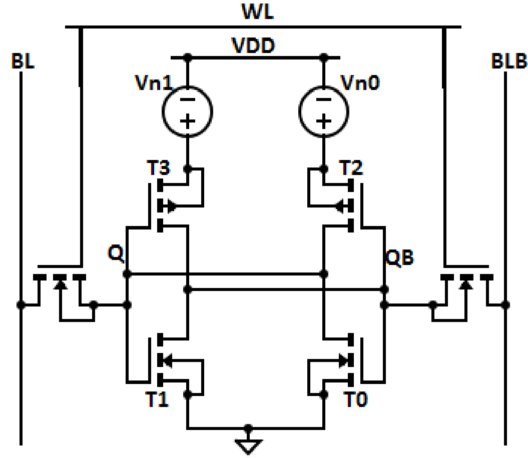


Fig.3.21: *6T SRAM cell with noise injection at power supply nodes.*

power supply nodes of a cell without changing its start-up behavior. This metric is denoted as Vn_{ps} and it is defined as follows:

$$Vn_{ps} = Vn_1 - Vn_0 \tag{3.16}$$

Similarly, the cells that have logic "0" SUV are assigned with positive Vn_{ps} values, and the logic "1" SUV with negative Vn_{ps} values. Fig.3.22 shows the histogram of Vn_{ps} values for also 1000 memory cells. The farther the Vn_{ps} values are from Vn_{ps}=0 V, the higher immunity against noise the cells will be; those cells that have high absolute Vn_{ps} values will be considered as reliable cells for PUF application.

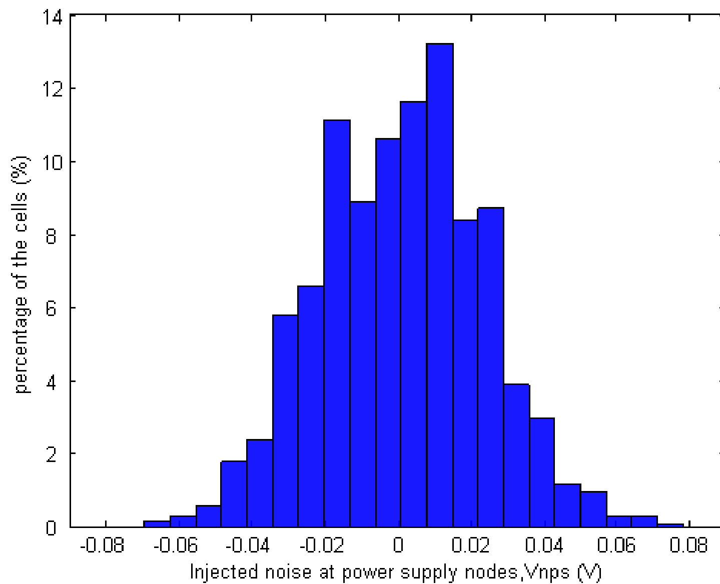


Fig.3.22: *The histogram distribution of Vn_{ps} metric.*

3.4.2 Correlations Between Noise Injection Locations

All the proposed noise injection metrics are presented to study the immunity of SRAM-PUF cells against the internal noise. For this reason, some correlation between them is expected. However, the degree of correlation could be different based on the location of the injected noise. Injecting the noise at the ground of the cell can focus on the contribution of N-MOS transistors in the immunity against noise; as the injected voltage noise will act as an added offset (bias) to those transistors. Conversely, injecting the noise at the power supply node focuses on P-MOS transistors contribution. On the other hand, the N-MOS transistors will have an equal noise immunity contribution to the P-MOS transistors when the noise is injected between the internal nodes of the cell; as the added offset (noise) will be divided between them.

Fig.3.23 presents the relation between the noise injected at the ground (Vn_g) and the noise injected at the power supply (Vn_{ps}), where each star in this figure represents one simulated SRAM cell. Those two metrics achieve a liner correlation factor of 0.92. While

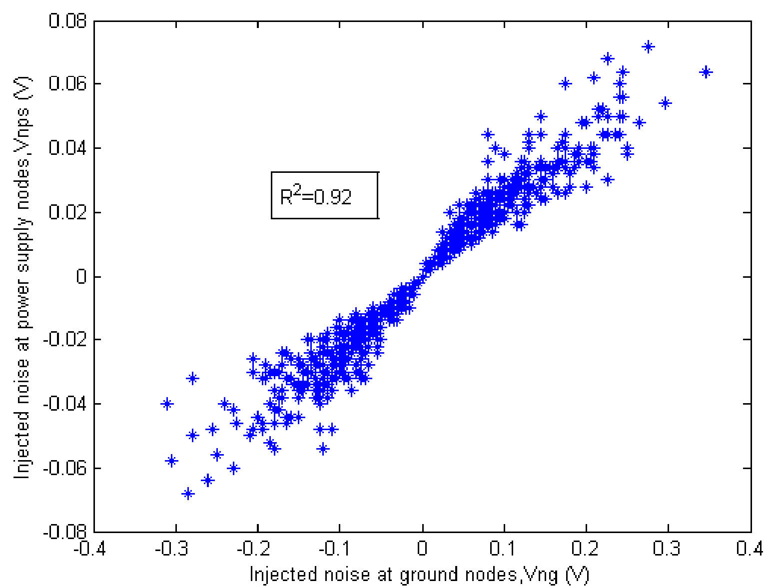


Fig.3.23: *The relation between Vn_g and Vn_{ps} metrics.*

the linearity in this relation means the cells that can tolerate high voltage noise injected at ground of the cell, can also tolerate high voltage noise injected at the power supply nodes.

Slightly better linearity is shown in Fig.3.24, where the relation between the noise injected at the ground (Vn_g) is compared with the noise injected between the storage nodes (Vn_i). The linear correlation factor can reach up 0.95. Again, this linear relation also justifies the definition of the injected noise metrics, where a cell with high absolute value of Vn_g metric will also has high absolute value of Vn_i metric and thus it will be

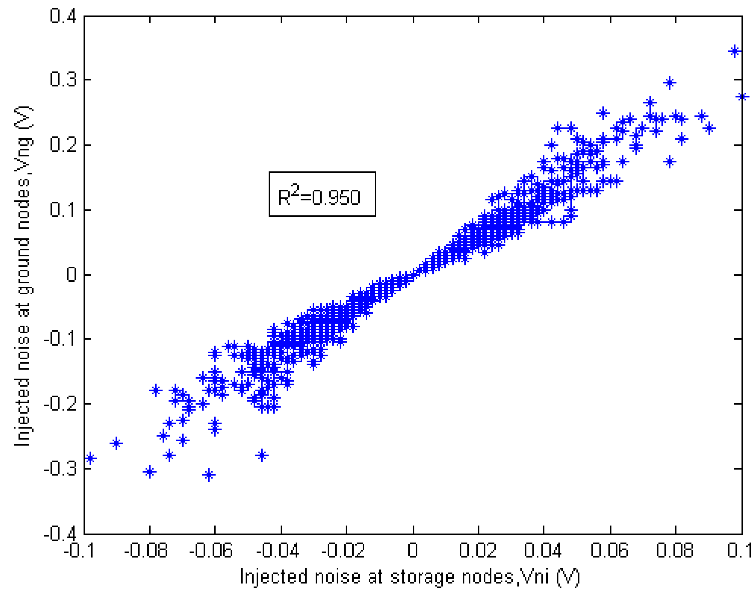


Fig.3.24: *The relation between Vn_i and Vn_g metrics.*

immunized against injected noise at both locations. Finally, the relation between Vn_i and Vn_{ps} metrics is presented in Fig.3.25, where a high linearity relation is shown between them. The linear correlation factor in this relation is very high and equals to 0.99. Based on that, to classify the noise immunity of the cells for PUF implementation, injecting the noise at power supply nodes is approximately similar to injecting the noise between the internal nodes.

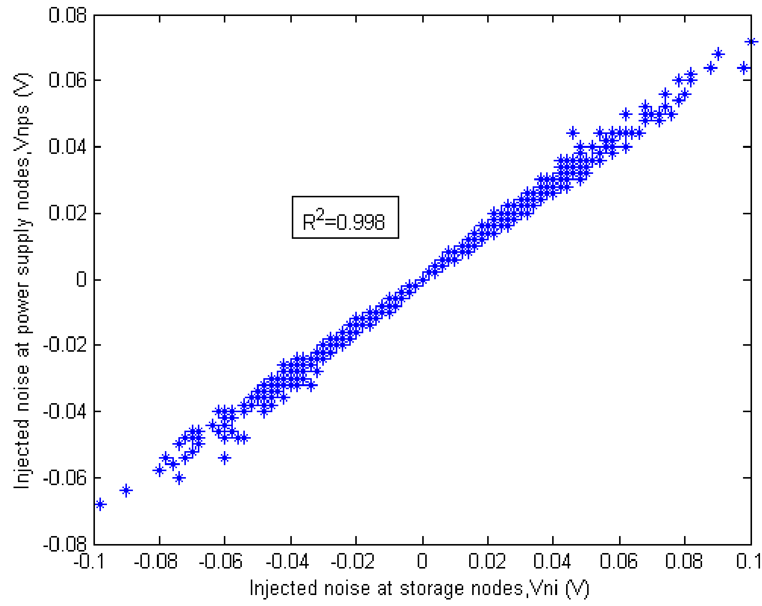


Fig.3.25: *The relation between Vn_i and Vn_{ps} metrics.*

In case of Vn_{ps} metric, the injected noise will add a full voltage offset (bias) to the P-MOS transistors. While in case of Vn_i metric the offset will be divided between P-MOS and N-MOS transistors. In this sense, adding a full offset or a divided offset to the P-MOS transistors will not highly affect the classification of the cells; as the relation between both cases has high linearity with R^2 equal to 0.998. On the other hand, adding a full offset to N-MOS transistors, in case of Vn_g metric, can slightly affect the classification; as the relation between adding a full offset and adding the same divided offset (in case of Vn_i metric) has R^2 equals to 0.950.

As a result, these observations can indicate a higher influence of the P-MOS transistors on the classification of cells in terms of noise immunity rather than N-MOS transistors. This result also agrees with mismatch metric (Pd_{Vth}) results (see Equation (3.6), Section 3.2.1) and the work in [52], where the P-MOS transistors are more dominant in deciding the SUV and thus the reliability of the cell.

3.4.3 Correlation with Inherent-Cell Mismatch

The implementation of SRAM circuits for PUF application requires that the selected cells should be immunized against transient electrical noise at cell's start-up stage. On the other hand, the cell's start-up behavior is mainly depended on the relative strength between the cell's transistors, that could be represented by inherent cell-mismatch. Therefore, a highly mismatched cell should tolerate high amount of injected noise at any location in the cell.

Even though the proposed noise injection-based metrics can assign a specific value to each cell which describes the maximum voltage noise that can be tolerated by the cell, they cannot provide a threshold range to distinguish between the reliable and unreliable cells. To solve this issue, we will utilize the mismatch metric (Pd_{Vth} in Equation (3.6), Section 3.2.1) to define a threshold range for noise injection-based metrics. Pd_{Vth} metric, that includes the weighting factors, is specifically used here as it is evaluated by DC and Transient simulations which allow us to compare it with the proposed metrics that obtained by Transient simulations; while the rest of the mismatch metrics (Pd_{Vtho}, Pd_{Vm}) are only evaluated by DC simulations.

To define a threshold range for noise injected-based metrics, a similar percentage of reliable cells that identified by Pd_{Vth} threshold range (see Fig.3.4, Section 3.2.1, Page) will be applied to define the threshold ranges for the three proposed metrics. In that case, we have selected 70% of the cells starting from the highest absolute values of the three metrics Vn_g, Vn_i and Vn_{ps}.

The relation between Vn_g and Pd_{Vth} is shown in Fig.3.26. Where each "o" represents one SRAM cell. The cells that are colored blue represent the reliable cells identified by Pd_{Vth},

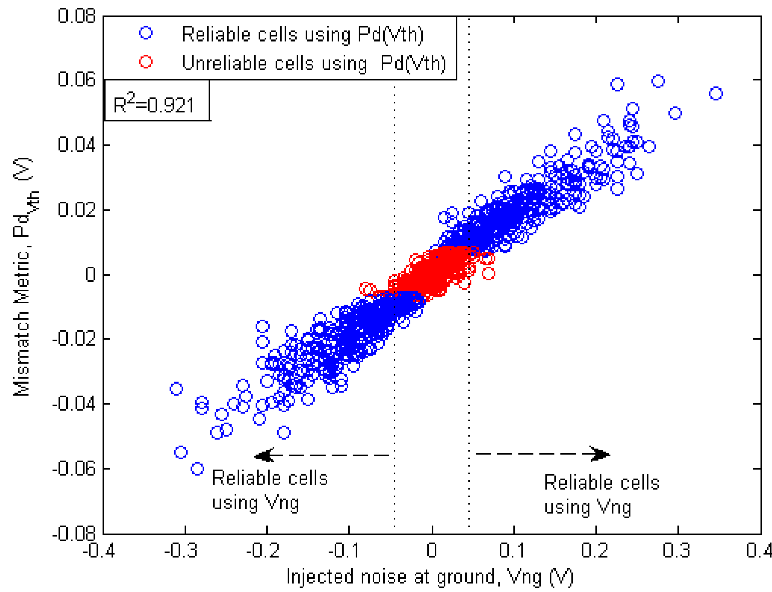


Fig.3.26: *The relation between Vn_g metric and inherent mismatch.*

while the red ones are the unreliable cells. The calculated threshold range for Vn_g metric is shown in this figure, where it is represented by the two vertical lines in Fig.3.26 to distinguish between the reliable and unreliable cells. The linear correlation factor for this relation equals 0.921, also we observed that around 91% of the selected reliable cells are common between Vn_g and mismatch metrics. So, a highly mismatched cell can tolerate high level of the noise injected at the ground of that cell.

A slightly better relation is shown in Fig.3.27, where the Vn_i metric is studied with respect to the inherent mismatch. The reliable cells that are identified by Pd_{Vth} are also represented by the blue "o" while the red "o" represents the unreliable cells. The vertical lines in this figure also show the calculated threshold range for Vn_i metric, where the cells located outside those lines are the reliable ones. In addition, the linear correlation factor between Vn_i and Pd_{Vth} equals to 0.965 while around 93% of the reliable cells are common between them.

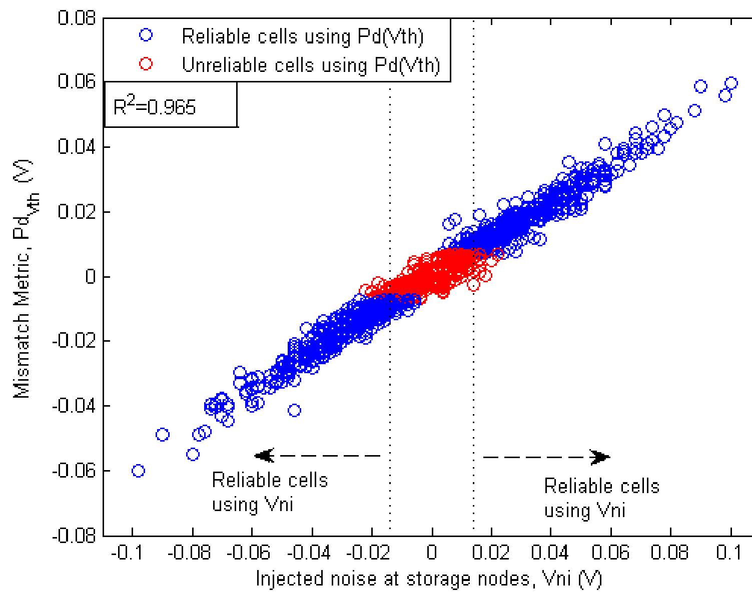


Fig.3.27: *The relation between Vn_i metric and inherent mismatch.*

Finally, Fig.3.28 shows the relation between Vn_{ps} and Pd_{Vth}, this relation is almost similar to Vn_i and Pd_{Vth} relation in Fig.3.27. Where the linear correlation factor between the metrics in Fig.3.28 equals to 0.963 and the common reliable cells between them equals to 93%.

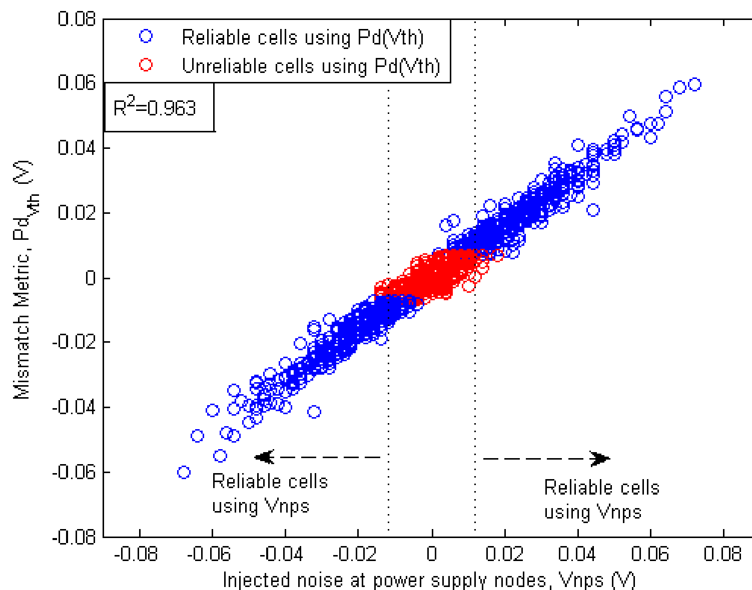


Fig.3.28: *The relation between Vn_{ps} metric and inherent mismatch.*

The similarity between the relations in Fig.3.27 and Fig.3.28 means that injecting the voltage noise either at the power supply nodes or between the storage nodes will not

affect to a great extent the classification of the reliable cells. However, this similarity also supports our conclusion in the previous section, where the P-MOS transistors are more dominant in the noise immunity and start-up behavior of the cell rather than the N-MOS transistors.

3.5 Dynamic Start-up Behavior as Reliability Metrics

Many approaches based on the dynamic behavior have been presented to improve and evaluate the SRAM-cell reliability [64-66]. In SRAM-PUF applications, the behavior of the cell at power-up stage is the main interest. However, the basic 6T SRAM-cell can be represented as a non-linear time variant system [64], thus the cell's start-up behavior is determined by dynamic (transient) behavior. Based on that, a dynamic analysis is required to study the reliability of SRAM-PUF.

In this Section, the reliability of the memory cells will be classified based on their dynamic behavior. Firstly, we will graphically utilize the transient start-up behavior of the memory cells to obtain a value for each cell which represents the cell reliability in PUF applications. Secondly, a state space representation for SRAM start-up behavior will be proposed. In the state space, the separatrix of the SRAM cells is utilized to define a new metric that can efficiently classify the cells for PUF implementations.

3.5.1 Graphical Representation of SUV

When cells power-up, some of those cells have tendency to stabilize at the stable logic state ”1” (VQ=Vdd, QB=0 V) and the others prefer to stabilize at “0” (Q=0 V, QB=Vdd). This different tendencies between the cells can be evaluated utilizing Transient Monte Carlo analysis to graphically represent the evolution of output voltages of Q and QB nodes with respect to time throughout the start-up stage.

Fig.3.29 presents the voltage evolution of the internal nodes for several SRAM cells. Those cells are powered-up with initial nodes values, VQ_o=0 V, VQB_o=0 V, and power supply ramp-up time equal to 5ns. The cells that converge to Q-axis, prefer to have SUV towards logic “1”. In contrary, the cells that prefer to start-up at logic “0” will converge to QB-axis. By comparing the resulting curves in both cases, the output nodes of memory cells experiment dissimilar curves evolution; while the reason is mainly due to the inherent mismatch between cell’s inverters. On the one hand, if the cell is mismatched. The stronger inverter in the cell, that has the highest current gain, will decide the final SUV. On the other hand, if the cell’s inverters are highly symmetrical (well matched), the final SUV will require higher time and voltage levels for Q and QB nodes to be decided. Therefore, we can notice in Fig.3.29, that SRAM cells that have higher peak (denoted as *M* in Fig.3.29) in their curves usually correspond to well-matched memory cells. Conversely, the curves have lower peaks, correspond to highly mismatched cells.

The peak point for each cell (*M*) represents the coordinate (Qmax, QBmax) of the curve. The M point defines the voltages of internal nodes that are required by the memory cell

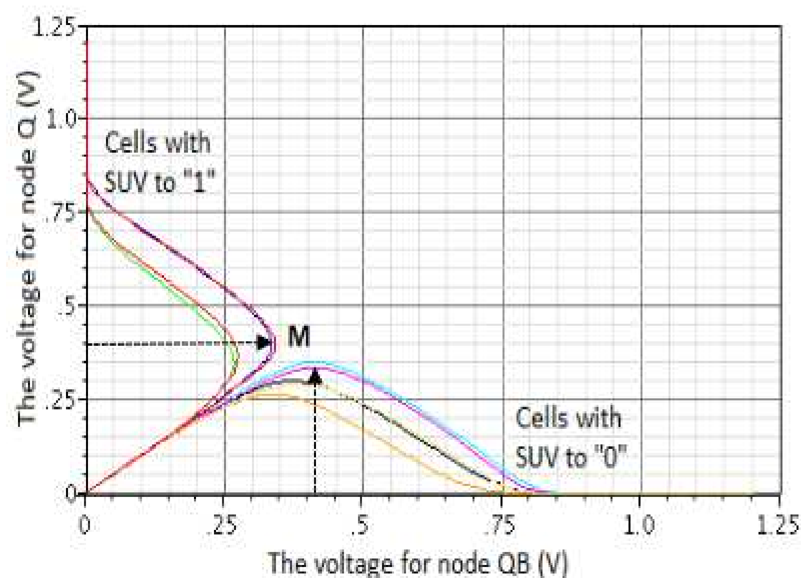


Fig.3.29: *Q versus QB voltages during start-up for several SRAM cells.*

feedback at moment of deciding the final SUV. In case of memory cell that starts-up at logic “0”, the maximum voltage at moment of deciding the final SUV is higher in value for Q-node (Qmax). While in case of memory cells that have SUV at logic “1” the higher node’s voltage is the value of QB-node (QBmax). These maximum voltage values for both cases are representative of the tendency to the final preferred SUV for the memory cell. Utilizing this assumption, the proposed maximum voltage value for each cell can be defined as follow:

$$V_{max} = Min(Q_{max}, QB_{max}) \tag{3.17}$$

The previous equation evaluates the minimum voltage required by in the internal SRAM cell nodes to decide the final preferred SUV.

As a result, high *Vmax* values correspond to highly matched memory cells, while low Vmax values correspond to highly mismatched memory cells. The *Vmax* values for the memory cell, are obtained by implementing Transient Monte Carlo simulation to draw the curves, while OCEAN is used to define the peak points as in Fig.3.29. However, the histogram distribution for Vmax indicator is shown in Fig.3.30, the lower X-axis values (*Vmax*) are the more mismatched cell and thus more reliable. So, the cells located at the leftmost side of this histogram are defined as reliable cells, and they will be selected to produce the PUF response. The proposed indicator is studied with respect to the inherent cell-mismatch. In this sense, we have used Pd_{Vth} metric to represent the inherent cell-mismatch. The relationship between *Vmax* and Pd_{Vth}, is shown in Fig.3.31. Where each star represents one SRAM cell. In addition, the coloring is based on the cell final SUV; the cells that have logic “0” are colored in black, and logic “1” cells are colored in red. The best fitting curve for this relation, added in Fig.3.31, indicates an inverse

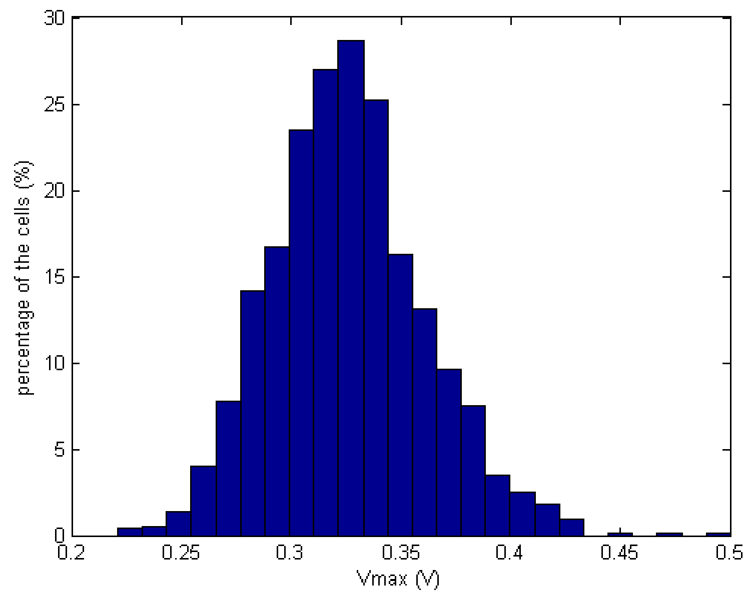


Fig.3.30: *The histogram distribution of Vmax indicator.*

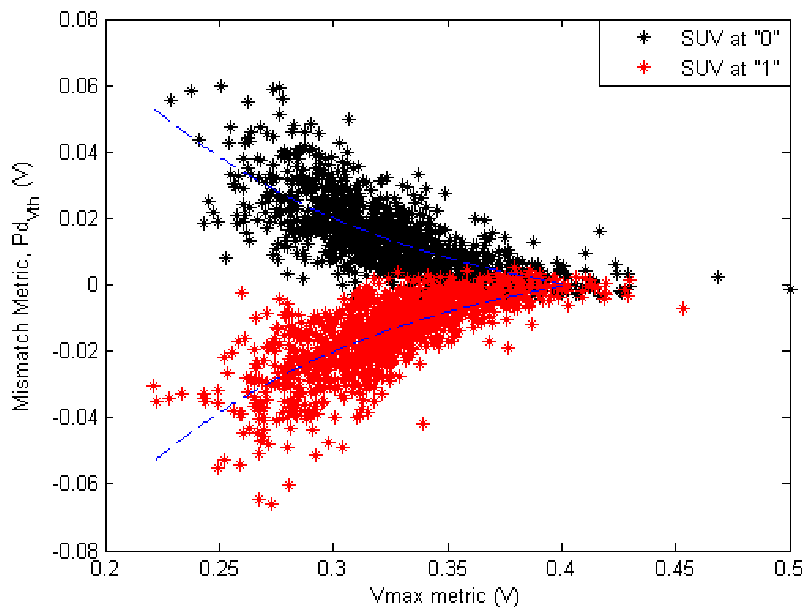


Fig.3.31: *The relation between the graphical Vmax and mismatch metric.*

non-linear relation, as it is expected by our assumption: the majority of the proposed memory cells which have high absolute Pd_{Vth} values also have low *Vmax* values and vice versa. On the other words, the cells that have low *Vmax* values don't necessitate to reach to a high voltage at their nodes to decide the final SUV, and this means that they are identified as highly mismatched. Conversely, the cells that have higher *Vmax* need

to reach high node's voltage to decide the final SUV, so they will be considered more matched and thus unreliable in PUF implementation.

Another indicator that can be defined, based on transient start-up behavior, by also utilizing Q and QB voltages evolution behavior that is presented in Fig.3.29. In this case, we have observed that the moment when the SRAM cell decides the final SUV, the (Q, QB) transient curves pulls away from the 45° line (VQ = VQB). The reason is that the voltage of one of cell nodes starts to increase, meanwhile the voltage of the other cell node falls-down to 0 V. Based on this observation, we define the new indicator to represent the difference between the slope of Q-QB curve at peak point (M) and the slope of the VQ=VQB line (equals to 1). In other words, the value of this indicator, denoted as *ΔSlope*, is considered as the non-return value toward the final SUV, as the feedback in the memory cell after this value will force the cell to the final preferred SUV. For each cell in the proposed memory, the *ΔSlope* indicator is defined as follows:

$$\Delta Slope = 1 - \frac{Q_{max}}{QB_{max}} \tag{3.18}$$

In this sense, positive values for *ΔSlope* represent the cells that start-up at logic "0", and negative values represent the cells with logic "1". However, *ΔSlope* describes how far the mismatch of cell's inverters is from the ideal perfectly matched (balanced) case, which is represented by the slope of VQ=VQB line. So, lower *ΔSlope* values refer to the cells that decide their final SUV due to small differences between the cell's inverters and thus are more matched cells. On the contrary, the higher values of *ΔSlope* refer to highly mismatched cells. Similarly, to support the assumption of *ΔSlope* indicator, we present the relation between Pd_{Vth} and *ΔSlope* in Fig.3.32. The cells that have higher absolute

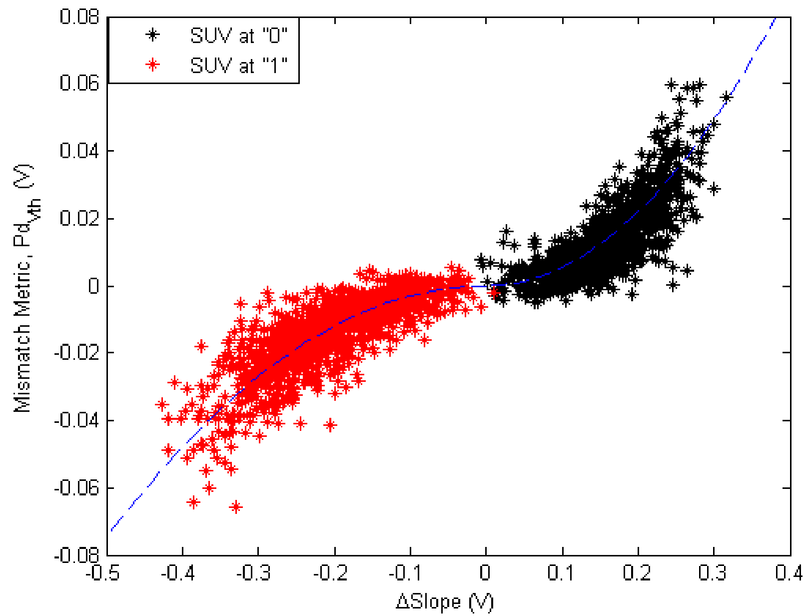


Fig.3.32: *The relation between the graphical indicator ΔSlope and Mismatch metric.*

ΔSlope values also have higher absolute Pd_{Vth} values and thus they are highly mismatched cells.

Finally, it can be also noticed from Q-QB curves in Fig.3.29, that the cells have different areas under their curves. The area under those curves, is related to the maximum required time and voltage at the internal nodes (Q, QB) of each cell to reach the final preferred SUV. Based on that, we define the last indicator in this section, denoted as *Area(Q-QB)*. The definition of *Area(Q-QB)* indicator is almost similar to the previous *Vmax* indicator.

In this sense, the cells that have larger area under their curves, require longer time and higher node voltage to decide their final SUV. Those cells are more balanced or matched cells. On the other hand, the cells that will be useful for PUF implementation are the ones with smaller area under the curve. However, Fig.3.33 shows the relation between

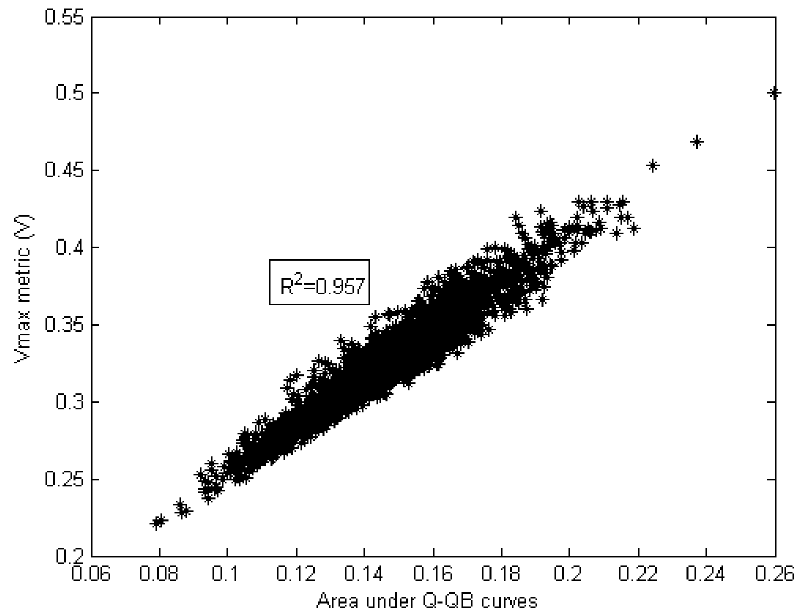


Fig.3.33: *The relation between Area(Q-QB) and Vmax indicators.*

the area under the curves and *Vmax* indicator with linear correlation factor equals to 0.957.

The relation between *Area(Q-QB)* and Pd_{Vth}, is approximately similar to the relation between *Vmax* and Pd_{Vth}, as shown in Fig.3.34. A cell with larger area under its curve is more matched cell while a cell with smaller area refers to highly mismatched cell.

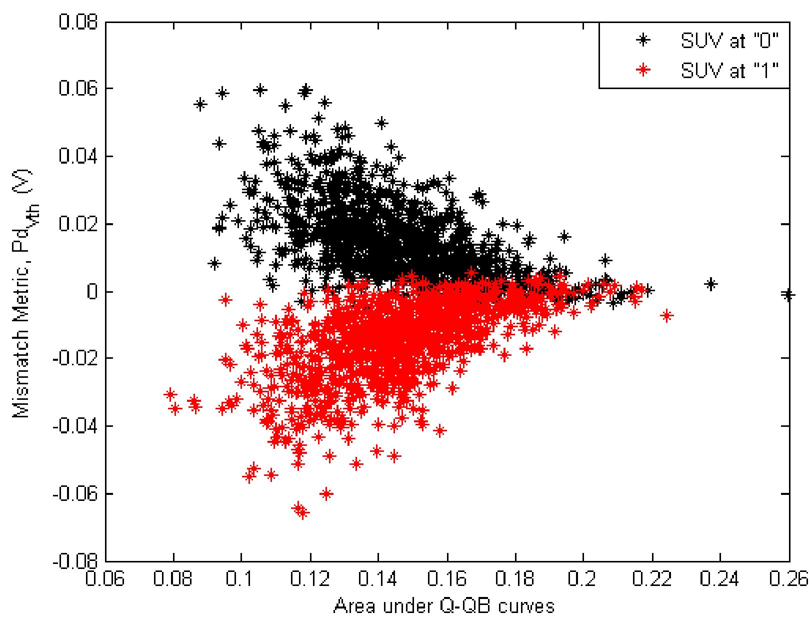


Fig.3.34: *The relation between the Area (Q-QB) and Mismatch metric.*

However, the *Vmax* indicator still show better relationship with inherent mismatch rather than the Area indicator, as we can notice when the both relations in Fig.3.34and Fig.3.31 are compared.

3.5.2 SRAM Separatrix as Reliability Metrics

The SRAM dynamic noise margins (DNMs) have been proposed in [67]. The concepts of stability boundary, state-space separatrix, are implemented to define and evaluate the write and read DNMs, with the goal to ensure successful write and read operations. However, in the similar dynamic aspects, the authors suggest that the state-space separatrix could also be implemented in hold operation.

The state-space analysis is utilized to describe the behavior of the memory cell under their dynamic evolution [68], while a second order nonlinear time invariant system is used in [69] to represent this state-space analysis. A state-space analysis is a representation of a physical system through a mathematical model that is based on inputs, outputs and state variables that are correlated by differential equations. The system state can be characterized as a vector inside the state-space. A phase-space is a space in where all potential states of the system are presented, with each potential state of the system refers to one of different points in the phase-space. The non-forced evolution of each potential state to the way to the equilibrium point is named as trajectory (black doted lines in Fig.3.35).

As we mentioned previously, SRAM cell has three equilibrium points, two of them are stable representing logic ' 0' (VQ=0 V, VQB=Vdd) and logic' 1' (VQ=Vdd V, VQB=0 V) states, S0 and S1 in Fig.3.35, respectively. The other one is a meta-stable state, the point M in Fig.3.35. Each one of the stable points has its own area of attraction in the

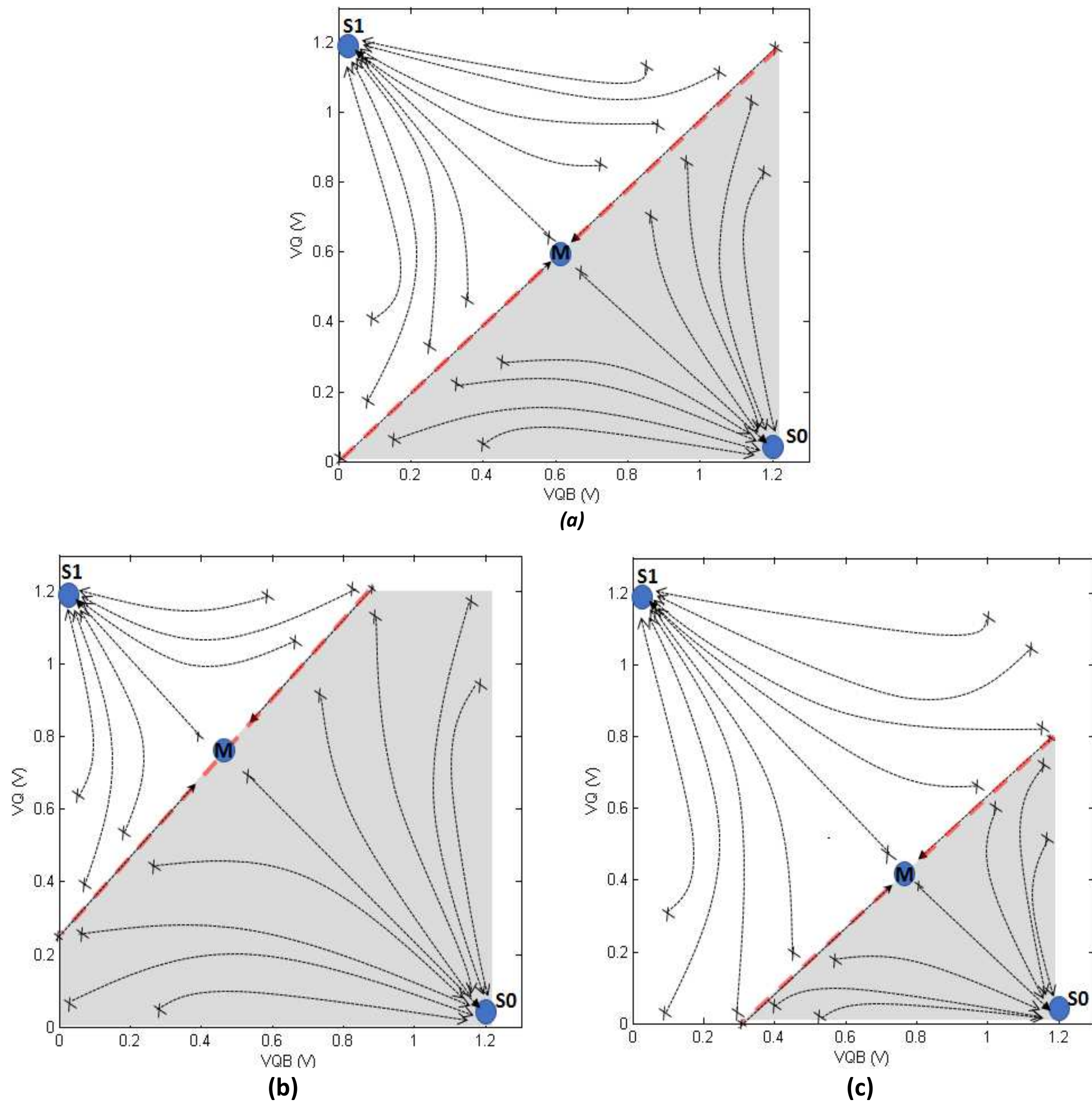


Fig.3.35: *The Phase-space of SRAM memory evolution at start-up stage: a) for an ideal symmetrical cell, b) for an asymmetrical cell with tendency towards logic "0", c) for an asymmetrical cell with tendency towards logic "1".*

state-space, the border between these areas is called SRAM separatrix; red dashed lines in Fig.3.35. If a memory cell is started-up from any initial node conditions (VQ_o, VQB_o) in the area of attraction, the state-trajectory of this cell will tend towards one of the stable-state points as the time grows.

On the other hand, in an ideal cell, the state-trajectory will go to the meta-stable state, if the cell is started-up from any initial condition located on the separatrix line. However, in real memory cell, the impact of process variation and mismatch will force its state-trajectory towards one of the stable states.

Based on that, the dynamic start-up behavior of memory cell, can be determined by exploring: the internal nodes initial conditions, the location of the separatrix boundary in the state-space, and the ramp-up time and voltage of the input power supply. However, the separatrix line of a perfectly symmetrical (matched) memory cell goes along with the diagonal line, VQ=VQB, as show in Fig.3.35 (a). In case of asymmetrical memory cell, the separatrix line location will be far from the ideal position, as shown in Fig.3.35 (b) and Fig.3.35 (c).

The presented phase-state in Fig.3.35 (b), refers to a mismatched SRAM cell, this cell has area of attraction towards S0 state (shadowed area) bigger than the area of attraction toward S1 state (logic “1”). Therefore, it will prefer to star-up at logic “0”. By contrast, Fig.3.35 (c) represents a cell that has a bigger area of attraction towards S1 state (unshadow area) and thus it will prefer to start-up at logic “1”. However, the size difference between the areas of attraction for each case, determines the strength of the inherent mismatch of the memory cell. In other words, the level of inherent cell mismatch can be described as how far the separatrix line from the ideal location, at VQ=VQB line.

Recently, a stability test has been proposed in [70], that relies on the dynamic evaluation of memory-cell stability for PUF applications. The authors in this work implement a 2-step test procedure described as follows:

- The memory is started-up with initial condition of the cells tilted towards the stable state S1. This is done by setting the initial value for node QB_o = 0 V and for node Q_o= *Vskew*.

- The memory is started-up another time but with initial condition of the cells tilted towards the stable state S0. This is done by setting the initial value for node QB_o= *Vskew* and for node Q_o= 0 V.

If the SUV of a cell in the first test is S1, while the SUV in the second test for this cell is S0, the cell is considered highly matched. Conversely, if the SUV of a cell is the same for both tests, the cell is considered as mismatched cell, and thus it will be very stable to be used in PUF applications. Even though this stability test is able to classify the highly symmetrical cells, if the selected value for *Vskew* is low, selecting the initial condition value (*Vskew*) is the main drawback of this method. A very low value of *Vskew* can cause that some of the unreliable cells are not detected by the test. While a high value of *Vskew* can lead to oversizing the required memory for PUF implementation, as more reliable cells will be eliminated and defined as unreliable cells.

In this section, we propose a new metric-based methodology to classify the SRAM-PUF cells based on their reliability. The position of the separatrix line will be utilized to assign a value for each cell in the proposed memory. In this sense, for a mismatched cell, if the separatrix line is located above the ideal line (symmetrical cell) as in Fig.3.35 (b), the cell will have tendency to power-up at logic “0”. In contrary, if the separatrix is below the ideal line the cell will prefer to power-up at logic “1” as in Fig.3.35 (c).

However, the proposed metric aims to calculate how far the separatrix is from the symmetric position. The farther the separatrix of a cell is from ideal position, the larger the area of attraction towards one of stable states the cell will have, and thus the more reliable it will be. Specifically, we aim to find the intersection of the separatrix with Q-axis or QB-axis; the separatrix of a cell with tendency to logic “0” state will only

intersect the Q-axis (see Fig.3.36 (a)) while for a cell with tendency to logic “1” the separatrix will only intersect the QB-axis (see Fig.3.36 (b)). The new metric, denoted as Separatrix Intersection Distance (*SID*), will be defined as the distance between those intersections and the intersection of the ideal separatrix at (VQ_o=VQB_o=0 V), this distance is shown in Fig.3.36.

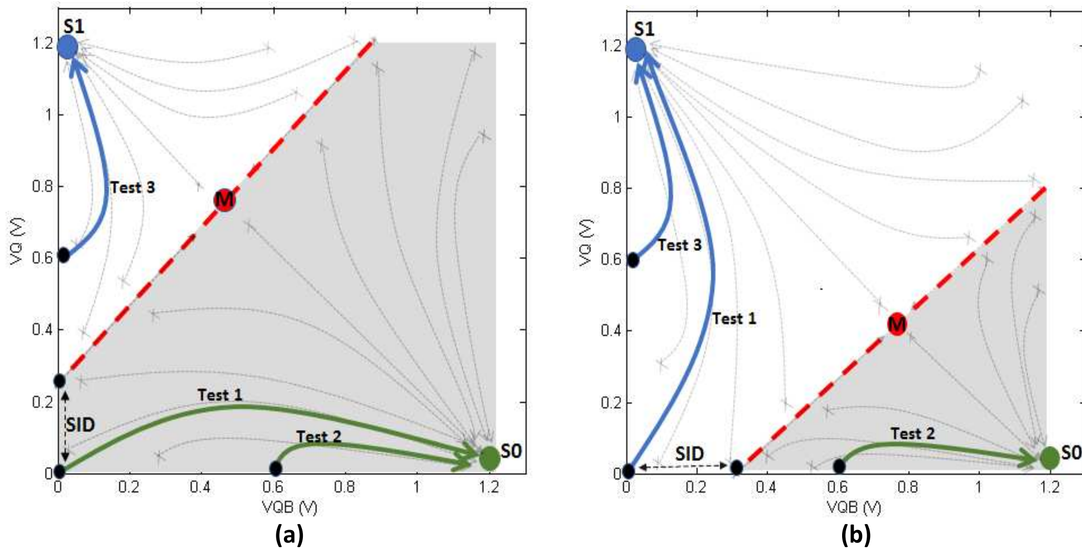


Fig.3.36: *The Phase-space of memory cells evolution where the proposed testes and SID metric is presented: a) for an asymmetrical cell with tendency towards logic “0”, b) for an asymmetrical cell with tendency towards logic “1”.*

Transient Mote Carlo analysis is utilized to evaluate the new metric *SID*. However, the methodology that we have used can be divided into two parts. The first part will be dedicated to observing the direction of the SUV tendency (either to logic “0” or logic “1”) and thus we will be able to decide where the separatrix intersects the phase-state axis (either at Q-axis or QB-axis). While the second part aims to find the intersection value. The procedure that we have used is summarized as follows:

- To find which axis the separatrix will intersect, we implement three tests for each cell in the proposed memory:
    1. Test 1: the cell is started-up with neutral initial conditions, where VQB_o=0 V and VQ_0=0 V, see Fig.3.36.

2. Test 2: the cell is started-up with initial condition highly tilted towards the stable state S0 (logic “0”), where VQB_o=0.6 V and VQ_0=0 V, see Fig.3.36. We have chosen the value 0.6 V as it is half of the supply voltage, which allows to detect the tendency towards S0 even for highly symmetrical cells; low value of initial condition doesn’t allow detecting the low mismatched cells [70].
3. Test 3: the cell is started-up with initial condition highly tilted towards the stable state S1 (logic “1”), where VQB_o=0 V and VQ_0=0.6 V, see Fig.3.36. Also, we have chosen the value 0.6 V which allows to detect the tendency towards S1.

After implementing those tests in the proposed memory, we found that if the final SUV of a cell in Test 1 is similar to the final SUV in Test 2 the cell will have a tendency toward S0, and thus its separatrix will intersect the Q-axis as shown in Fig.3.36 (a). On the other hand, if the final SUV of a cell in Test 1 is similar to the SUV in Test 3, the cell will tend to start-up at S1 and thus the separatrix intersection will be on QB-axis as presented in Fig.3.36 (b).

- After knowing the location of the separatrix intersection for each cell, we will find the value of intersection on the detected axis. This value, *SID* in both Fig.3.36 (a) and (b), represents when a cell starts changing its tendency towards one stable state to the other stable state. In this sense, we have implemented a search algorithm in Ocean platform to evaluate the value of separatrix intersection. This algorithm will control the initial condition (starting from the point VQ_o=VQB_o=0 V) for the detected node (Q-node or QB-node that is decided

in the previous step), in steps of 5 mV. The algorithm will keep changing this initial condition meanwhile checking the SUV until the start-up behavior changes. The value of the initial condition for the detected node where the start-up behavior changes is defined as *SID*.

Finally, we have assigned a positive *SID* values for the cells have tendency towards logic "0" state (the cells that have intersection at Q-axis as in Fig.3.36 (a)), while a negative sign is assigned to those cells with tendency towards logic "1" (the cells that have intersection at QB-axis as in Fig.3.36 (a)). The histogram distribution of the *SID* values for the memory is shown in Fig.3.37, the cells that represented by the right bars will have more tendency to start-up at logic "0" while the cells represented by the left bars will have more tendency to stabilize at logic "1". However, the most reliable cells are located at the rightmost and leftmost bars, because their separatrix are far from the ideal position; so, they have bigger area of attraction towards one of the stable states. Based on the assumption of *SID* metric, the cells that have high absolute value of *SID* should also be highly mismatched cells, while the matched cells should have low

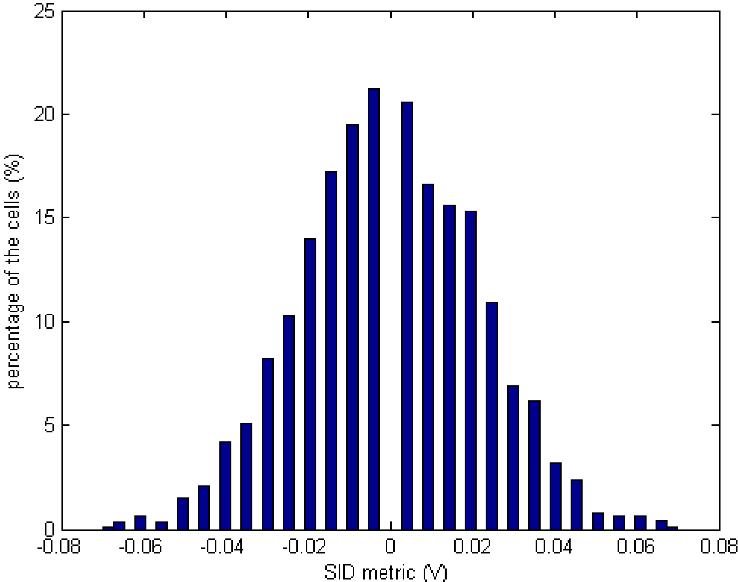


Fig.3.37: *The histogram distribution of separatrix metric (SID).*

absolute value of *SID*. Fig.3.38 presents the relation between the proposed *SID* metric and the inherent mismatch represented by Pd_{Vth} metric. Each star in the figure represents one cell of the memory, where the black color represents the tendency towards logic "0" state and the red one represents the tendency of the cells towards logic "1". We can notice that the cells that have high value of *SID* metric also have high value of Pd_{Vth}. Which means that the farther is the distance of the separatrix from ideal location, the cell will be more mismatched. This relationship has linear coefficient equal to 0.965 that shows the agreement of the both metrics on classifying the memory cells for PUF implementation.

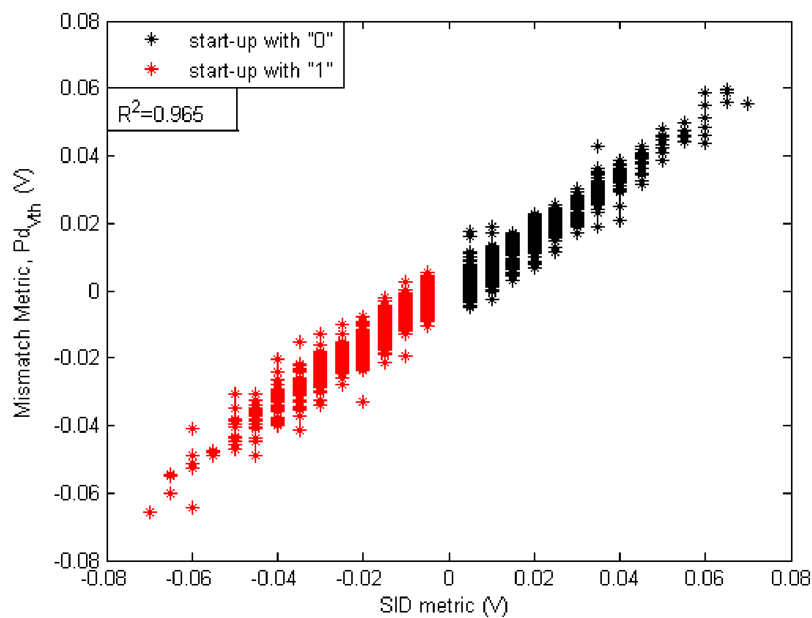


Fig.3.38: *The relation between the separatrix metric (SID) and Mismatch Metric*.

CHAPTER 4

EXPLORING EXTERNAL AND INTERNAL PERTURBATIONS IMPACT

A reliable PUF security scheme is the one that generates more constant response pattern, SUV pattern in SRAM-PUF, regardless of external and internal perturbations; such as environmental temperature variations. In other words, the reliability of an SRAM-PUF depends on how sensitive the SUV is to these perturbations.

SRAM cell uniformity and reliability for PUF implementations has been studied by applying comprehensive electrical experiments with suitable equipment in the research laboratory. Hence, many works like [43, 47] propose to classify memory cells by exploring the changes in the cell's reliability due to different external conditions: power supply ramp up voltage and time, and temperature. Also, the repeatability of SUV pattern of memory cells under the internal induced noise, such as thermal noise in [71], is used to study the cell's reliability in [72]. Using the methodologies in [43,47,72], the SRAM-PUF reliability is evaluated by exploring a wide range of operational conditions at test period during post-manufacturing process. Those exhaustive methods may classify and identify SRAM cells that generate repeatable SUVs under different external and internal conditions and propose to include these memory cells in PUF applications [40]. In [26], indirect preselection approaches are presented, where massive tests for memory cells under wide range of perturbations are performed. The cells that pass the

tests are identified as reliable and included to produce the PUF response. The main disadvantage is the needs for large number of tests which will increase the time and costs.

This chapter explores the potential influence of external and internal perturbations on the start-up behavior of SRAM cells, and hence on the stability of PUF cell. The unperturbed cells will be classified as stable cells due to their high tolerance. Firstly, we introduce the simulation setup designed to explore the impact of external perturbations in section 4.1, then we show the simulation results of the impact of power supply Ramp-Up Time (RUT) on the SRAM-PUF reliability in section 4.2. In section 4.3, the influence of previously stored values into SRAM cells will be studied, where the impact of these values will be used to classify cell stability. Section 4.4 analyzes how temperature affects SUV of the cells, and how it could be used to distinguish between stable and unstable cells. The probability of memory cells to have a repeatable SUV under the induced voltage noise will be presented in section 4.5. Finally, in each section, all the proposed metrics in the previous chapter will be studied under these perturbations.

4.1 Simulation Setup

The impact of external perturbations analysis is performed in this chapter to observe the SUV under several perturbations. The SUV is evaluated using a similar simulation setup and a similar 6-T SRAM cell schematic as described in Section 3.1. However, in section 4.2, the RUT is swept from 1ns to 1ms. In section 4.3 and 4.4, the initial node conditions and temperature of the cell are varied; respectively. A modified SRAM cell schematic is introduced for Section 4.5, modeling the internal cell noise to mimic a more

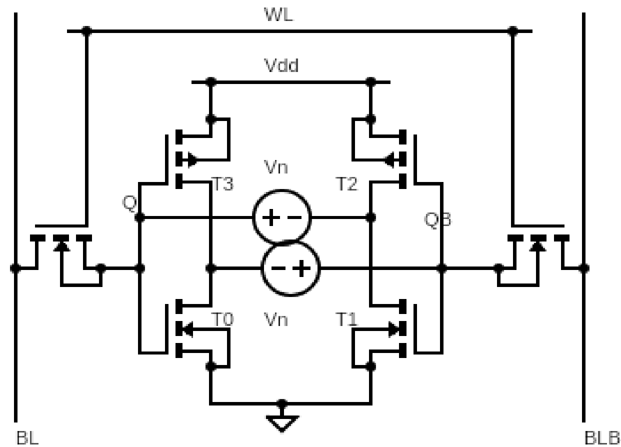


Fig.4.1: *6T SRAM cell including two random noise sources.*

realistic stat-up behavior. In this sense, two Transient Random Voltage Sources are used as shown in Fig.4.1. The values of these sources are generated randomly with respect to time using MATLAB.

4.2 Impact of Power Supply Ramp-Up Time (RUT) on Start-Up Behavior

An important issue that has been explored in a few of prior works on SRAM start-up behavior is the role of voltage supply RUT [73-74]. Where the SUVs of the memory cells has been proved to be affected by RUT variations. The RUT was proposed in [73] as an optimization parameter to improve the reliability and reproducibility of SRAM-PUF response under wide range of extreme temperatures, demonstrating the effect of RUT in varying the SUVs. The authors in [74] study the relationship between internal cell parameters, like transistors threshold voltages, and SUV at different RUTs. They claim that, as the RUT becomes faster, the threshold voltage mismatch of the P-MOS transistor pair will dominate the SUV of memory cells more than the N-MOS pair. This result is exploited to identify robustly strong SRAM cells for PUF application. By contrast, the authors in [52] consider RUT as a factor that affects the difference between both

P-MOS and N-MOS contributions to SUV. In this sense, they claim that slow RUT makes the P-MOS transistors more dominant in deciding the final SUV; while this result doesn't agree with the work in [74]. Next section will provide more details about the impact of RUT on the difference between P-MOS and N-MOS pairs contributions to SUV of memory cells, where the presented results agree more with the work in [74].

4.2.1 RUT Test Methodology and Impact Explanation

In this section, we perform a RUT approach to classify the memory cells with the goal of studying the ability of our proposed metrics in selecting the most reliable cells. In theory, the voltage supply of an SRAM cell can be powered-up very quickly, like raising the voltage from 0 V to Vdd in range of nanoseconds. Also, it can be powered-up very slowly in range of seconds [74]. Therefore, the RUT values are evaluated for 1000 transient start-up, using Monte Carlo simulations, to explore from 1ns to 1ms ramp-up time with steps of 10x. Fig.4.2 presents the percentage of memory cells that have changed their SUV with respect to the reference SUV obtained at 5ns RUT. Similar to the work in [75],

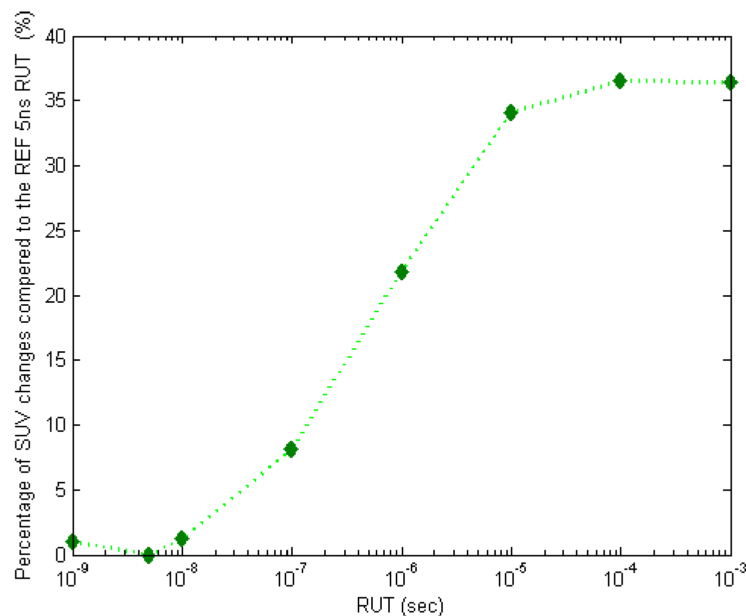


Fig.4.2: *percentage of cells that change the SUV at 5ns for several RUTs.*

where the author used this value of RUT as a reference to evaluate the SUV for PUF implementations.

When RUT value increases (becomes slower), the percentage of cells that change their SUV also increases, which indicates that memory cells are affected by RUT variation. Those cells can be identified as unstable cells; because the strength of inherent mismatch is not sufficient to tolerate the noise cause by the variation in RUT. On the other hand, the cells that have strong inherent-mismatch can reliably start-up at the same logic state, even in the existence of significant impacts of slow or fast transient RUT [74]. In this sense, we will define an SRAM-cell as stable, if the SUV of the cell is constant under all RUT range. Otherwise, if at any RUT, the cell changes the SUV, it will be defined as unstable cell due to the variation in its start-up states.

The percentages of the stable and unstable cells are shown in Fig.4.3. The stable cells represent around 60% of the proposed memory; as shown in green bar in Fig.4.3. However, the rest of the bars represent the unstable memory cells. When the RUT is

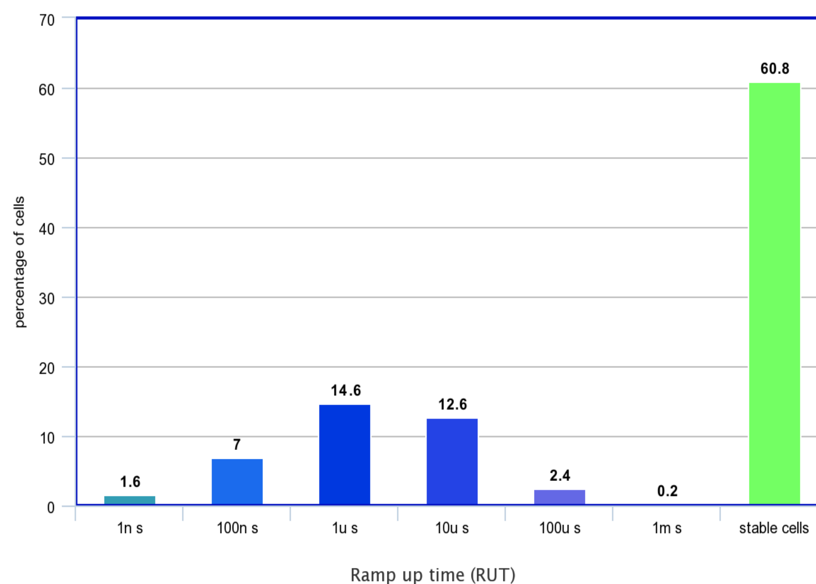
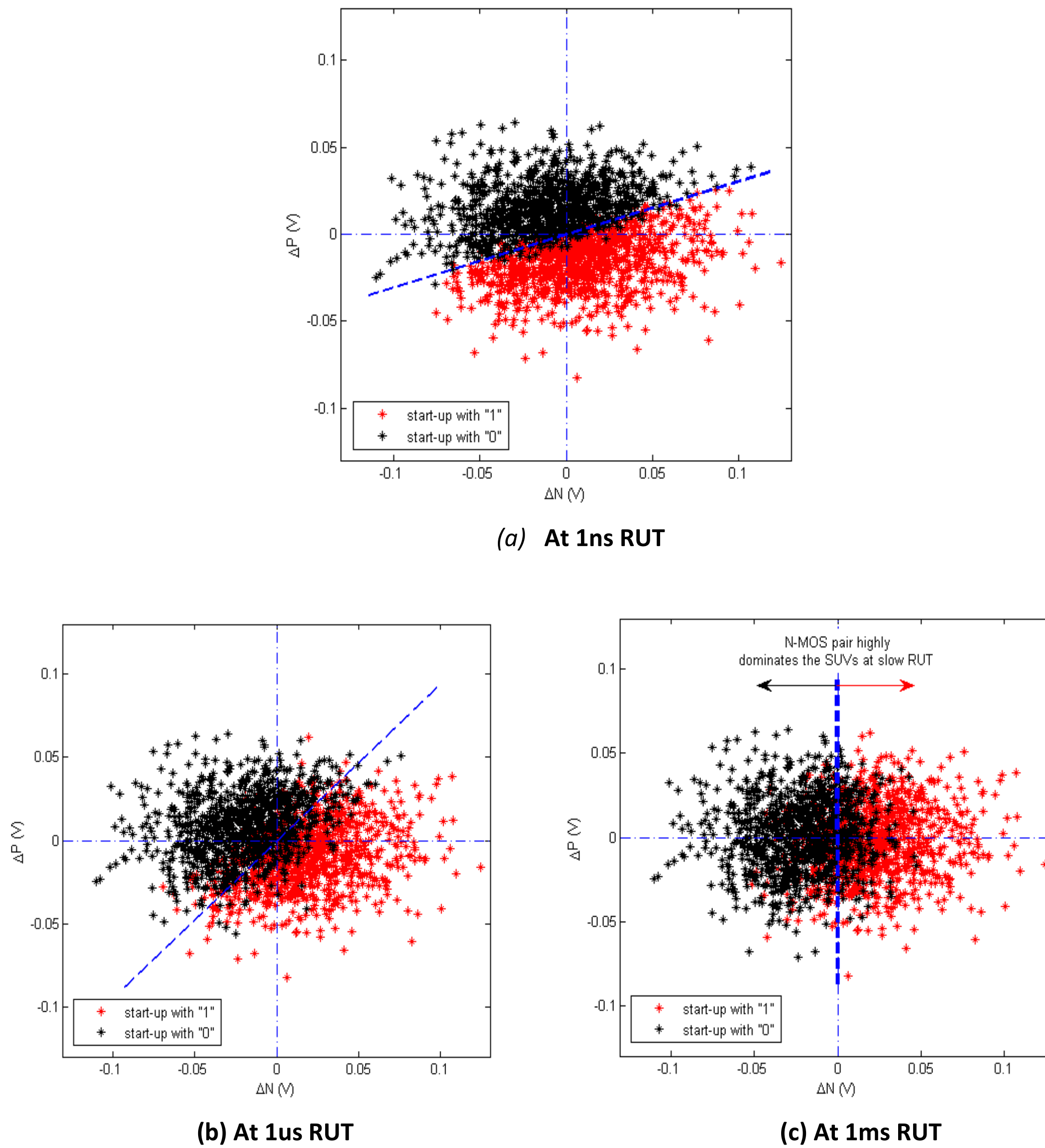


Fig.4.3: *percentage of stable cells and unstable cells that change their SUV at specific RUT value.*

increased (from 1ns to 1ms), most of cells start changing their SUV at specific RUT and keep the SUV constant for rest of the RUT range. In this sense, the low bars in Fig.4.3 represents the percentage of these cells when the SUV is started to change at specific RUT value. We can notice that, small percentages of the unstable cells (represented by the lowest bars) will start changing their output at slower or faster RUT values. However, most of the unstable cells change their SUV at 1us RUT and keep the same SUV until 1ms RUT. It means that, this value of RUT (1us) is critical as most of the unstable cells will change the final SUV. In the following discussion, we mention the reason behind this behavior and why this RUT value is critical.

According to previous works [4], [51] and mismatch metric (Pd_{Vth}, in section 3.2.1), it was reported that P-MOS transistor pair are more dominant than the N-MOS pair in deciding the SUVs of SRAM cells. The works [52], [74] consider RUT as a factor that changes the different P-MOS and N-MOS contributions to decide the SUV. However, both works have different results. In [74], if the voltage supply is ramped-up very fast (nanoseconds range), the raised voltage will initially all drop at the drain and source of the P-MOS transistors as the inherent capacitances at output nodes will take time to charge. Hence, with the gates of the P-MOS transistors initially held low because of this inertia of the inherent node capacitances, the P-MOS transistor pair will turn on strongly to dominate more the final SUV. Therefore, they claim that as the RUT becomes faster, the P-MOS transistors will dominate more the SUV of memory cells than the N-MOS pair. While in [52], using a slow RUT (250us), makes the inherent node capacitances less able to keep node voltage lower than VDD/2 at the first stage of evaluation. This will bring larger V_{GS} to P-MOS than the N-MOS pair, and makes P-MOS dominates more the SUV.

In order to understand this misalliance between both transistor type contributions under the existence of RUT effect, we computed the threshold voltage mismatch distribution between the same type of transistors (*ΔP* and *ΔN* introduced in section 3.2.1). Fig.4.4 shows the distribution of *ΔP* and *ΔN* results obtained for each cell in the



(a) **At 1ns RUT**

(b) At 1us RUT

(c) At 1ms RUT

Fig.4.4: *Distribution of threshold voltage variation of P-MOS and N-MOS transistors and the difference between their contributions on the SUVs at different RUTs (from fast to slow).*

proposed memory, while the coloring in those figures is based on SUVs of memory cells at different RUT values; Fig.4.4 (a) with SUVs at 1ns, Fig.4.4 (b) with SUVs at 1us and Fig.4.4 (c) with SUVs at 1ms. The red stars represent the cells with SUV at logic “1”, while

the cells that start-up at logic "0" are represented by black stars. We can notice from these figures that, as the RUT changes, the difference between P-MOS and N-MOS contributions to decide the SUV also changes.

In Fig.4.4 (a), the fast RUT reflects that both P-MOS and N-MOS pairs are involved in deciding the SUVs of the memory cells. Specifically, P-MOS pair has more contribution, as the line that divides most of SUVs (the dashed blue line) is located between ΔP and ΔN axis while this line is more near to ΔP=0V. As the RUT becomes slower as in Fig.4.4 (b), the slope of the line that divides cell SUVs increases toward ΔP axis (ΔN=0V); this means that N-MOS gains more domination on SUV. Reaching to 1us RUT (see Fig.4.4 (b)), both type of transistor have almost similar contributions, as the dashed blue line is diagonal (with slope around 45°). In this sense, P-MOS and N-MOS pairs will fight to decide the final SUV, and this makes most of the unstable cells start changing their SUV at 1us as presented in Fig.4.3.

However, N-MOS transistors play a major role in deciding cell's SUVs in Fig.4.4 (c) when RUT is very slow, as most of the cells with positive ΔN start-up at logic "1" while negative ΔN values start-up at logic "0"; regardless of ΔP values.

To summarize the relation between P-MOS and N-MOS contributions with respect to RUT, we have used Equation 3.6 ($Pd_{Vth}=W*\Delta P-(1-W)*\Delta N$). As we mentioned in section 3.2.1, the added weighting factor (*W*) can represent contribution of P-MOS pair to final SUV, while (*1-W*) can represent contribution of N-MOS pair. Using similar procedure, the W is optimized based on the SUVs at each RUT in the proposed range. The resulting *W* at each RUT value is shown in Fig.4.5, where the red line corresponds to *W* values and the blue line for (*1-W*) values. As expected, a faster RUT causes that P-MOS transistor

pair dominates the cell SUVs more than the N-MOS pair, agreeing with result in [74]. By contrast, as the RUT becomes slower, the N-MOS will become more dominant; disagreeing the result in [52].

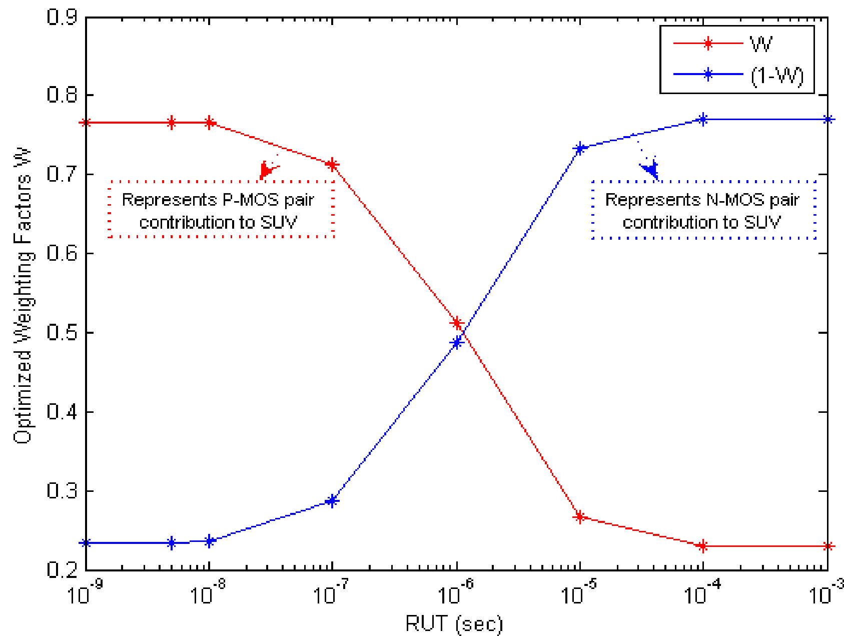


Fig.4.5: *The relation between P-MOS and N-MOS contributions with respect to RUT, using optimized Weighting Factor methodology as in equation (3.6).*

On the other hand, the intersection of these two lines (see Fig.4.5) represents the RUT where both type of transistors have an equal contribution to SUV. This RUT is very close to 1us, verifying our assumption that 1us is critical RUT for the proposed memory.

4.2.2 Proposed Metrics Robustness Considering RUT Variations

In the previous chapter, we defined several metrics to characterize the cell mismatch. The definition of the proposed mismatch metrics assumes that the cells which have high metric magnitude values are more reliable under all operational and noise conditions, while the cells with low metric magnitude values are less reliable, and thus their final SUV will change with small variations of these operational and noise conditions. In this sense, the extreme RUT variation can be considered as a noise that affects the reliability

of SRAM-PUF cells. So, in order to implement our metric based methodology for PUF applications, the proposed metrics should be able to identify those cells that are stable with respect to RUT variation (denoted as *Stable-RUT Cells*).

To study the ability of the parameter distance-based metrics in identifying the *Stable-RUT cells*, the relation between Pd_{Vm} metric (calculated by utilizing inverter's VTCs in section 3.2.2) and Pd_{Vtho} (calculated by obtaining the threshold voltages of the individual cell transistors in section 3.2.1) is presented in Fig.4.6. The *Stable-RUT cells* are colored in green while the cells in blue are unstable during RUT variation. We notice that most of the cells that have high absolute values of both mismatch metrics are stable under RUT variation; meaning that they have enough inherent mismatch to tolerate a significant noise caused by slow or fast RUT variation. On the other hand, low absolute values of Pd_{Vm} and Pd_{Vtho} metrics correspond to the cells that have sufficient mismatch to tolerate the variation in RUT, where most of these cells located closer to Pd_{Vm} =0 V and Pd_{Vtho} =0 V are identified as unstable RUT cells as shown in Fig.4.6.

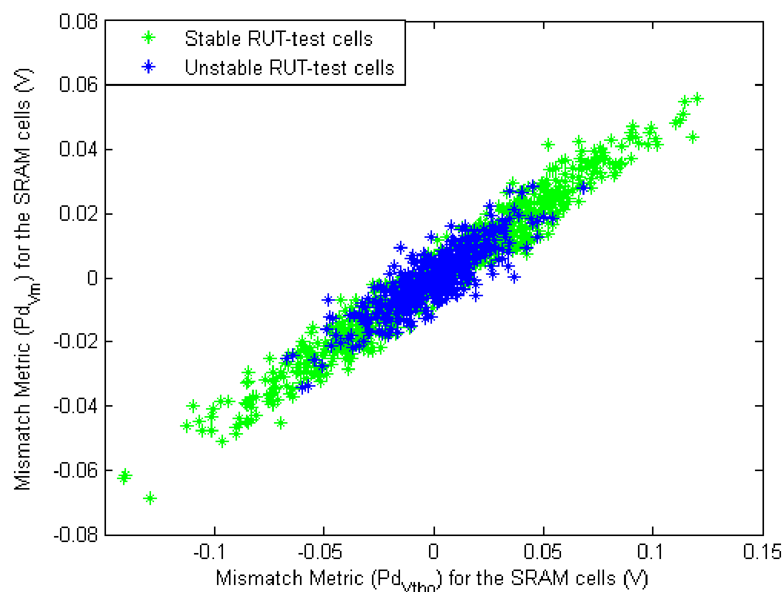


Fig.4.6: *Proposed parameter distance-based metrics relation indicating the Stable-RUT cells.*

In the same direction, the SNM-based metrics, SNM_d and INT_d in section 3.3.2, are proposed to characterize SRAM-PUF cell mismatch based on their static noise tolerance. High magnitude of these metrics indicates a more reliable. In this sense, the extreme variation of power supply RUT was considered as a noise, and the proposed SNM-based metrics should identify the cells that are immunized against this noise. Fig.4.7 shows the distribution of stable-RUT cells through the relation between SNM_d and INT_d metrics. Also, the stable-RUT cells (green colored) are located at both far ends of this plot, where both metrics have high absolute values. On the other hand, the cells with low absolute metric values (blue colored) cannot sustain the noise caused by RUT variations and are identified as unstable-RUT cells.

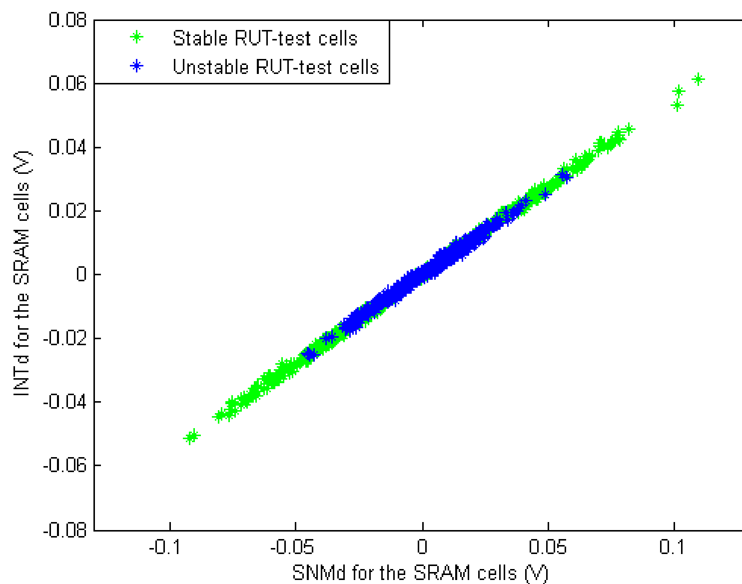


Fig.4.7: *Proposed SNM-based metrics relation indicating the Stable-RUT cells.*

Similarly, we study the ability of detecting the stable-RUT cells for injected noise-based metrics (the transient metrics Vn_g, V_{ni} and Vn_{ps} in section 3.4.1), where DC voltage noise is injected at different locations of the SRAM cell to evaluate the maximum noise that can be tolerated at these locations. Also, a high magnitude of those metric means that the cells can tolerate high amount of injected DC noise. To observe the metrics ability in

tolerating RUT variations, a 3D plot showing the distribution of stable-RUT cells among the relation between those metrics is presented in Fig.4.8. The highest absolute values of injected noise-based metrics can identify the stable cells during RUT changes, but the unstable cells are not well classified by lower metric magnitude; compared with the previous metrics.

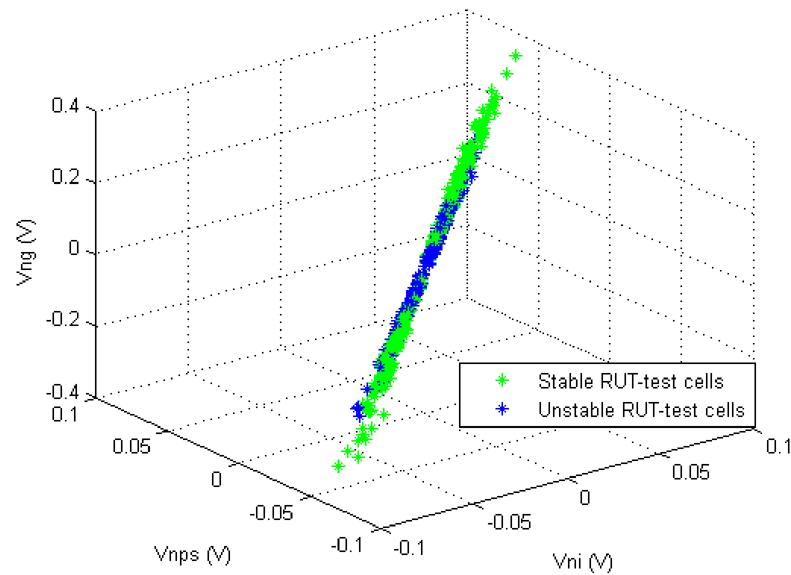


Fig.4.8: *Proposed injected noise-based metrics relation indicating the Stable-RUT cells.*

As we mentioned, the start-up of SRAM cells is a dynamic behavior, and we proposed several metrics to classify PUF cells reliability based on this behavior. The SRAM separatrix metric (*SID* in section 3.5.2) seems the most promising one, as it is highly correlated with inherent cell-mismatch. The *SID* metric represents the tendency of the cells towards the final SUV, and hence, the cell mismatch. A higher tendency towards the preferred SUV means a higher magnitude of *SID*. Fig.4.9 presents the histogram of *SID* values where the percentage of the cells that show stable-RUT behavior are highlighted in green. It can be seen that as the magnitude of *SID* metric increases the percentage of *Stable-RUT cells* also increases with respect to the unstable ones; the

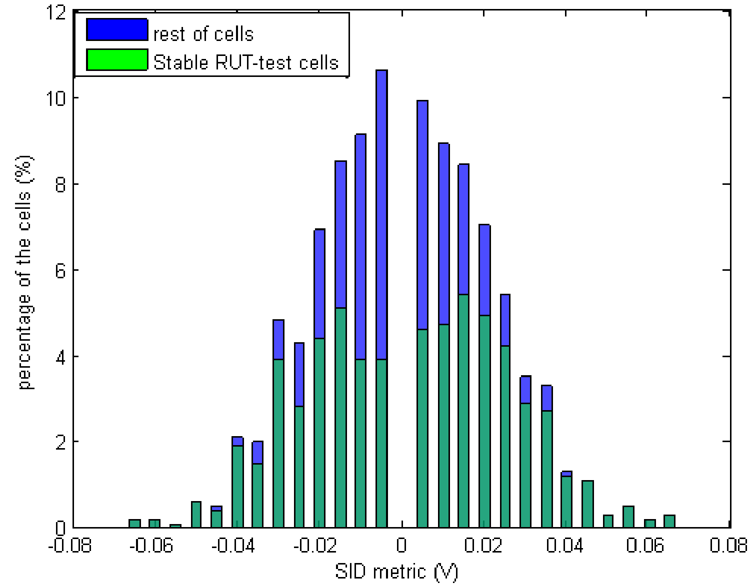


Fig.4.9: *The histogram of Proposed SID metric indicating the percentage of Stable-RUT cells.*

highest *SID* magnitude corresponds to the stable-RUT cells. The reason is that a cell with a high absolute value of *SID* has a high tendency towards the preferred SUV (highly mismatched), and thus the RUT variations will not affect its SUV, and it keeps stable. On the other hand, lower magnitudes of *SID* have lower percentage of *Stable-RUT cells*. Although a higher metric magnitude is able to identify most of the stable cells, some of these cells are identified by lower metric magnitude.

4.3 Impact of Previously Stored Value (PSV) on Start-Up Behavior

The data remanence effect is used as an approach in [76] to identify the highly stable bit-cells with minimum test time and hardware. This approach proposes to store a value into a SRAM cell and then reduce the time between two power-down cycles resulting that the SUV of this cell can be reverted to the previous stored values. On the one hand, if the memory cell is powered-down slowly enough to make the effect of data remanence comparable to the inherent cell-mismatch, then some cells will flip their SUV, while other cells will revert to the previous SUV [76]. On the other hand, if the

power-down cycle is too long, the data stored in the memory cells is entirely collapsed and their start-up behavior will remain unaffected by the Previous Stored Values (PSVs).

4.3.1 PSV Test Methodology and Impact Explanation

The impact of PSVs on start-up behavior has been evaluated by powering-up the memory cells with different initial voltages at the internal cell nodes. The Q and QB nodes (see the SRAM cell in Fig 4.1 in section 4.1) have been initiated with different voltage to mimic different power-down cycle times to observe if the cells will keep the PSVs. As the voltage of PSVs is higher, the power-down time is lower, and thus the data remembrance will have more impact on SUVs compared to inherent mismatch. The PSVs effect on SUVs has been explored by performing transient Monte Carlo simulations to obtain the SUVs using PSVs from initial values Q = 0V and QB = 0V, until to values of Q = 1.2V and QB = 1.2V. The range of PSVs is explored in equal steps of 0.2V for both nodes. However, we have used a DC voltage power supply to ignore the RUT impact and only focus on the PSVs effect on the start-up behavior.

Fig.4.10 reports the percentage of memory cells that change their SUVs with respect to the reference SUV generated with discharged initial conditions, where Q_o = 0V and QB_o = 0V. it can be noticed that, as the PSVs increase, the percentage of cells affected by shorter power-down times also increases. After performing the PSVs test, the percentage of memory cells that show unstable behavior in their SUVs is around 34% from the proposed memory.

However, the case when both internal nodes are initiated with Vdd may seem too extreme for a realistic exploration for PSVs range, nevertheless in [72] the authors suggest that the SUVs for SRAM-PUF could be evaluated by discharging the cells from

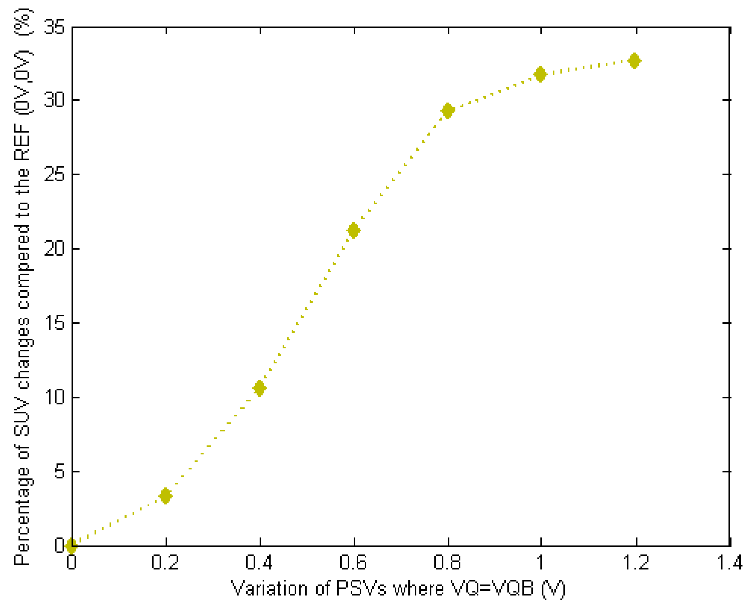


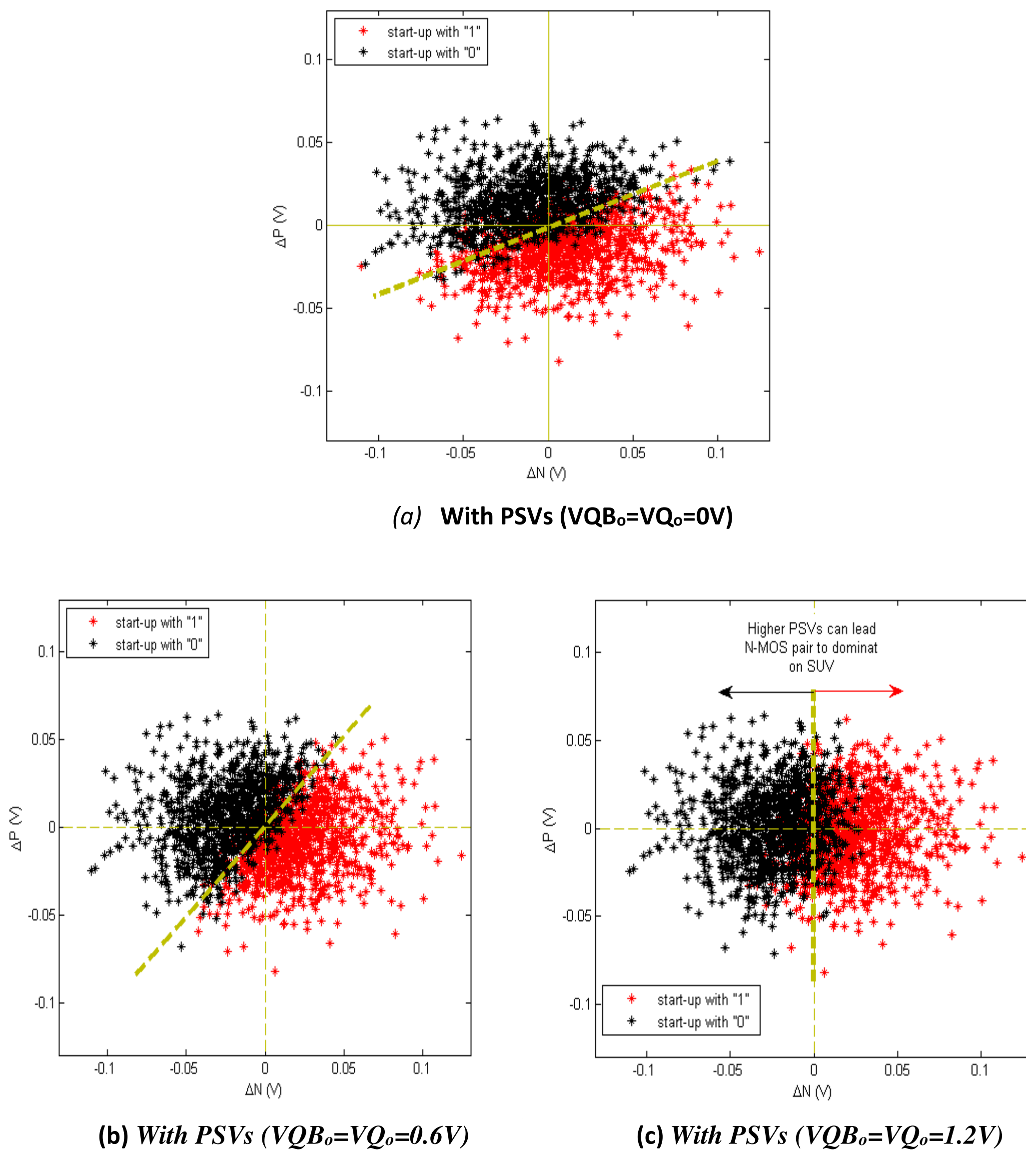
Fig.4.10: *percentage of cells that change the SUV for PSVs variation compared to reference (VQ_o=0V, VQB_o=0V).*

Vdd to 0V, and that is why we considered this case in the PSVs range. In this sense, the memory cells that can tolerate all PSVs in the proposed range will be defined as *Stable-PSVs cells*, while the memory cells that vary their SUVs at any value of the PSVs range will be identified as unstable-PSVs cell.

Similar to RUT variation, we found that a variation in PSVs can change the different contributions of P-MOS and N-MOS to decide the final SUV. A higher PSVs provide more domination to N-MOS transistor pair than the P-MOS pair; as in SRAM cell a higher PSVs mean that all the cross-coupled transistors will have a high voltage at their gates, and hence N-MOS pair will operate faster to decide the SUV, and thus more domination for N-MOS transistors.

Fig.4.11 shows the same *ΔP* and *ΔN* distribution as in Fig.4.4, but the coloring in Fig.4.11 is based on SUVs of memory cells at different PSVs; Fig.4.11 (a) with PSVs (VQB_o=VQ_o=0V), Fig.4.11 (b) with PSVs (VQB_o=VQ_o=0.6V) and Fig.4.11 (c) with PSVs (VQB_o=VQ_o=1.2V). It can be noticed that, low PSVs (as in Fig.4.11(a)) reflect that both

P-MOS and N-MOS pairs are involved in deciding the SUVs. Specifically, the P-MOS pair has more contribution than the N-MOS pair; as the line that divides most of SUVs (to logic “0” or “1”) is located near *ΔP*=0V. As we increase PSVs as in Fig.4.11 (b), the slope of the line that divides cell SUVs increases toward *ΔN*=0V; this indicates that N-MOS pair becomes more dominant on SUV. Finally, in Fig.4.11 (c), N-MOS transistors play a main role, regardless of *ΔP* values, in deciding cell’s SUVs when the PSVs is very high.



(a) **With PSVs (VQB_o=VQ_o=0V)**

(b) ***With PSVs (VQB_o=VQ_o=0.6V)***

(c) ***With PSVs (VQB_o=VQ_o=1.2V)***

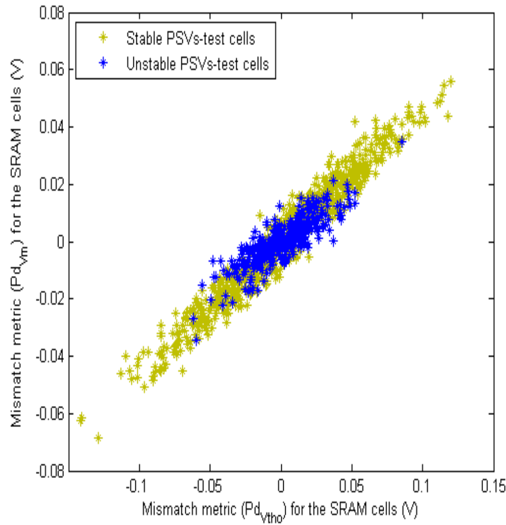
Fig.4.11: *Distribution of threshold voltage variation of P-MOS and N-MOS transistors and the difference between their contributions on the SUVs at different PSVs (from low to high).*

Finally, these results could be useful to understand the SUV at different node conditions, like starting-up the cells with fully charged nodes as in [72]. Additionally, by controlling these PSVs, the impact of RUT variation could be canceled; as both have similar effect on SRAM cell SUV.

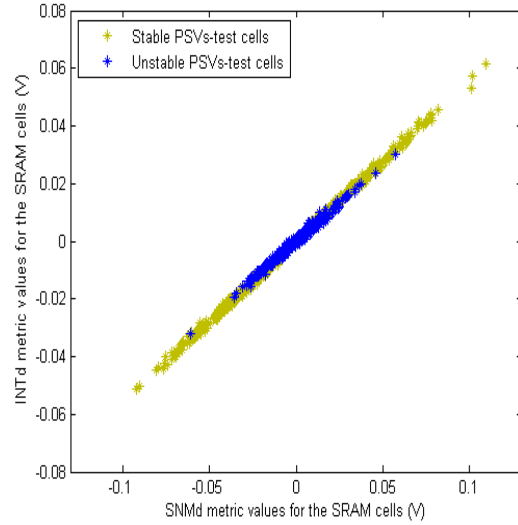
4.3.2 Proposed Metrics Robustness Considering PSV Variations

Again, the goal of this stress test is to support the ability of the proposed metrics in characterizing the mismatch of SRAM cells for PUF applications. Here, we present the same figures, as Figs.4.6-4.9 in the previous section, to study the reliability of our metrics in selecting the *Stable-PSVs cells*. Fig.4.12 shows the distribution of those *Stable-PSVs cells* (dark yellow colored). Fig.4.12 (a) for parameter distance-based metrics, Fig.4.12 (b) for SNM-based metrics, and Fig.4.12 (c) for injected noise-based metrics. The histogram in Fig.4.12 (d) shows the percentage of those cells with respect to SRAM separatrix metric. All of these sub-figures in Fig.4.12, the *Stable-PSVs cells* are identified by the higher metrics absolute values. While, the lower absolute values of DC-based metrics in Fig.4.12 (a) and (b) can identify the unstable-PSVs cells, better than the lower absolute values of Transient-based metrics in Fig.4.12 (c) and (d).

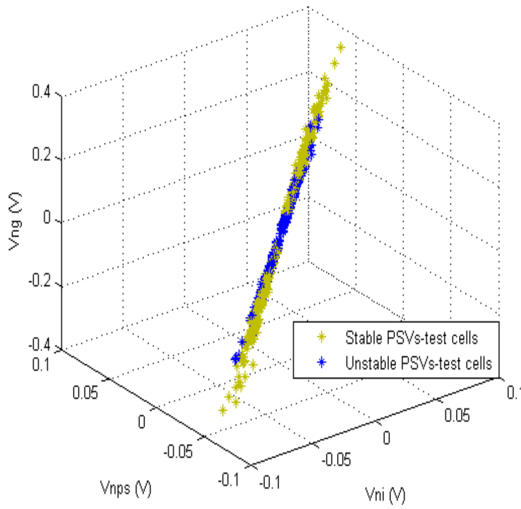
As a result, selecting the memory cells that have the highest metrics magnitudes for PUF implementations will improve PUF reliability. As those cells have high strength features to tolerate, with high repeatability of SUVs, the data remanence effect.



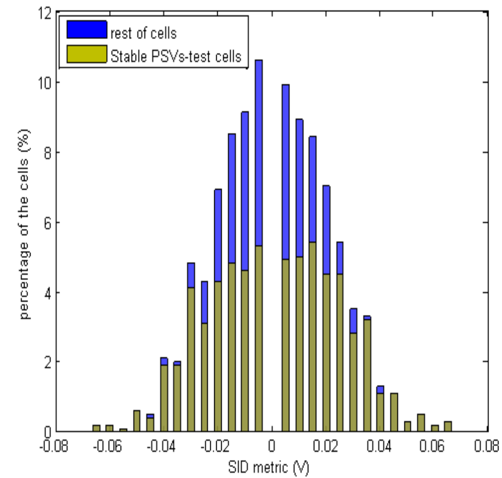
(a) Ability of parameter distance-based metrics



(b) Ability of SNM-based metrics



(C) Ability of injected noise-based metrics



(d) Ability of *SID* metric

Fig.4.12: *Studying the ability of the proposed metric against PSVs impact.*

4.4 Impact of Temperature on Start-Up Behavior and Robustness of Metrics

Another well-known external perturbation that affects the stability of SUVs is the temperature. The effect of this parameter on the reliability of SRAM-PUF has been widely explored in the literature [27], [73]. In these works, the authors reported that

temperature variations affect the start-up behavior of memory cells, where the selected PUF cells should show stable SUVs under those variations.

In this work, the temperature impact is explored to identify the stable and unstable cells with the main goal to support the ability of our proposed metrics in classifying those cells. We define the unstable-temperature cells as the cells that change their SUVs at any temperature in the simulated range, while the *Stable-Temperature cells* will be defined as the memory cells that tolerate the variations in all temperature range without changing their SUVs.

The range of temperature that we have explored starts from -40° C and reaching up to 120° C; in steps of 20° C. In Fig.4.13 the percentage of cells in the proposed memory that change their SUVs at any temperature in the range, is compared with the SUVs achieved at typical corner technology and nominal temperature (27° C). Note that the ratio of changed memory cells is higher where the temperature is higher than nominal temperature, with comparison to the ratio of cells at lower temperatures.

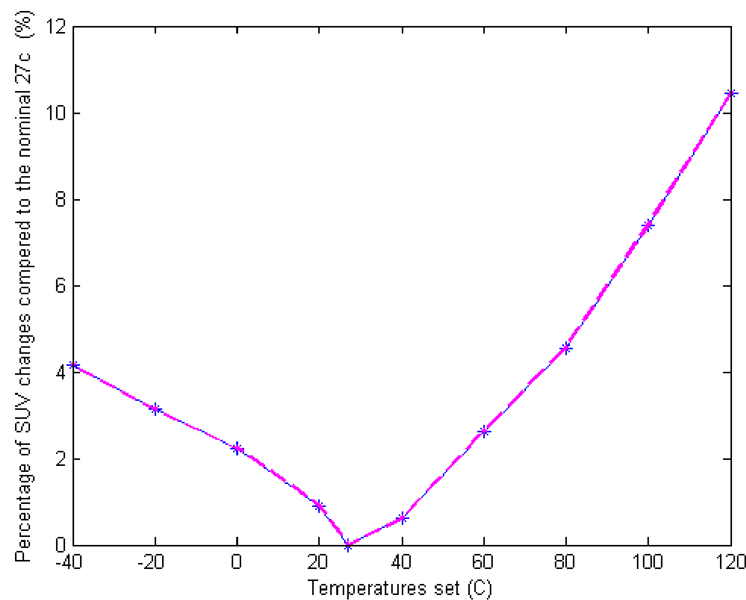
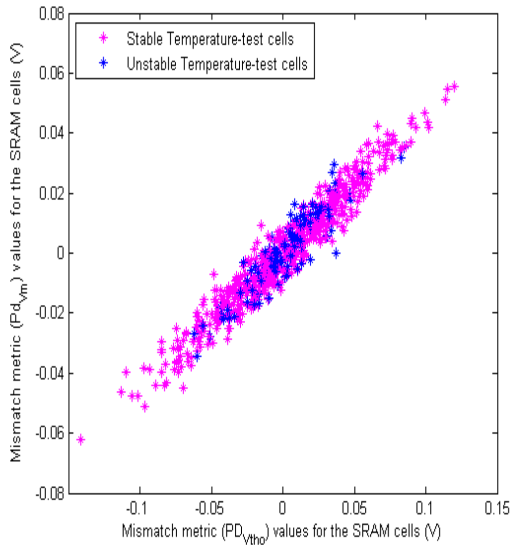


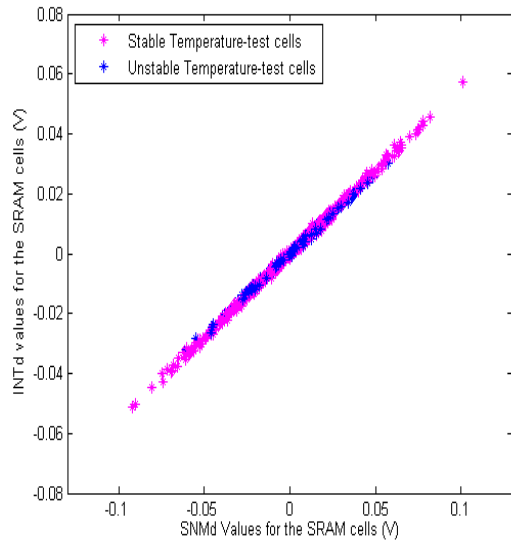
Fig.4.13: *Percentage of cells that change the SUV for temperature variations compared to nominal 27°C*

Note that the percentage of cells influenced by temperature is lower than the percentage of cells affected by the previous analyzed perturbations. This work pretends to analyze the implementation of non-specific SRAM design to achieve high reliability cells for PUF applications. For this reason, the memory can experiment an increase in temperature caused by long runtime periods, and the cells that will change their SUVs due to this increment in temperature should be identified as unstable and masked out from PUF response. Useful cells keep their SUVs unaltered, and they will be considered as stable.

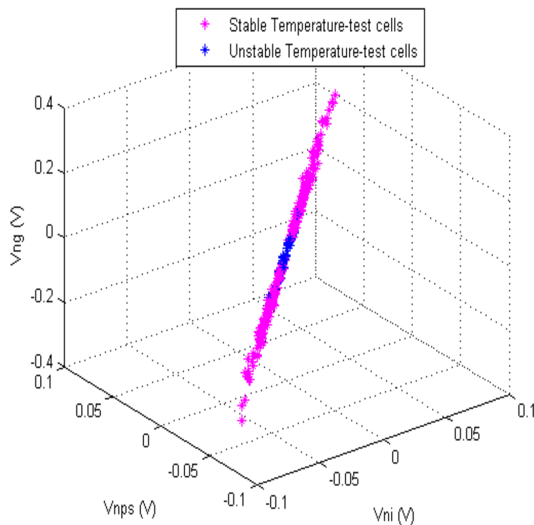
Fig.4.14 highlights the *Stable-Temperature* cells (pink colored) in the relations between our metrics similar as in the previous perturbations. Although, the number of memory cells affected by temperature is lower than with the perturbations considered in previous sections (only 11% are unstable-temperature cells), the range of DC metrics (in Fig.4.14 (a) and (b)) where the cells should be identified as unstable is longer than in previous cases. Thus, the parameter distance and SNM based metrics have some issues to correctly identify the *Stable-Temperature cells*. Despite of this, higher metric values continue corresponding to most *Stable-Temperature cells*. On the other hand, the Transient-based metrics (in Fig.4.14 (c) and (d)) show better performance in identifying both the stable and unstable-Temperature cells.



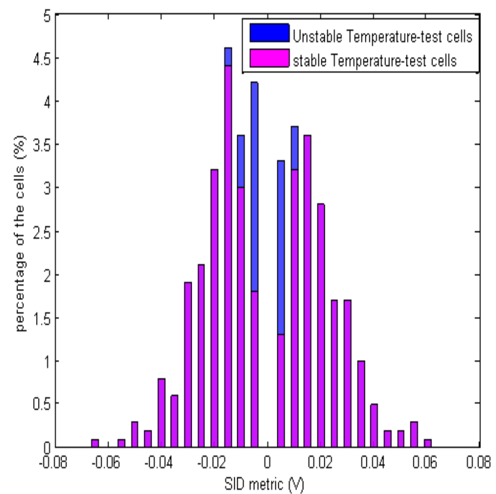
(a) Ability of parameter distance metrics



(b) Ability of SNM metrics



(C) Ability of injected noise metrics



(d) Ability of *SID* metrics

Fig.4.14: *Studying the ability of the proposed metric against Temperature impact.*

4.5 Impact of Internal Noise on Start-Up Behavior and Robustness of Metrics

PUF circuits must be implemented to offer as higher reliability as possible. Specifically, the start-up output should be, with high probability, the same each time the SRAM-PUF is challenged. One possible method to statistically evaluate the SRAM SUV performance

for PUF applications is to fabricate one or several instances of the circuit and experimentally measure the reliability [63]. However, in this section we will use another approach and model the internal thermal noise in the SRAM-PUF evaluating the statistical behavior of memory cells by performing transient simulations [72].

The internal thermal noise that exists in SRAM cells can significantly modify the SUV. To emulate a real SRAM start-up behavior, thermal noise can be modeled in memory cells by inserting transient random voltage sources between the cross-coupled inverter storage nodes of the cells with similar setup described in [72]. The schematic of the modeled noise was shown in Fig.4.1, also the setup was described in section 4.1. The magnitude of thermal noise at any node can be characterized in terms of a normal distribution with 0 mean. The standard deviation of this noise is mainly based on the node capacitance (*C*) and temperature (*T*), as the following equation [33, 72]:

$$\sigma_{noise} = \sqrt{\frac{KB\ T}{C}} \tag{4.1}$$

KB is Boltzmann constant. In [72], the standard deviation of internal thermal noise for each cell node in the memory was set to 4.5mV in 90 nm CMOS technology to produce enough SUVs variability level. Accordingly, we set σ_{noise} to 8.5mV for each cell node.

When noise is present, some cells change their SUVs each time they started-up while, others have more constant SUV. To observe the repeatability of SUVs, the SRAM cells have been sequentially powered-up 200 times by utilizing Monte Carlo simulations. Before each power-up, we have ensured that both Q and QB nodes are completely discharged. So, the impact of PSVs is not considered. The statistical SUV for each cell is calculated and represented as the probability to start-up at either logic “0” or logic “1”

states. Fig.4.15 displays a visualization of probability of SUVs achieved by implementing this methodology. Where the color variations describe the preferred SUV probability. The memory cells with greener color have higher probability to start-up towards logic "0" while the cells with darker blue have higher probability to start-up towards logic "1". On the other hand, the cells that have around 50% of SUV probability are represented by colors located at the middle of color legend in Fig.4.15. The start-up behavior of these cells is extremely affected by the internal noise and thus may be considered as random SUV.

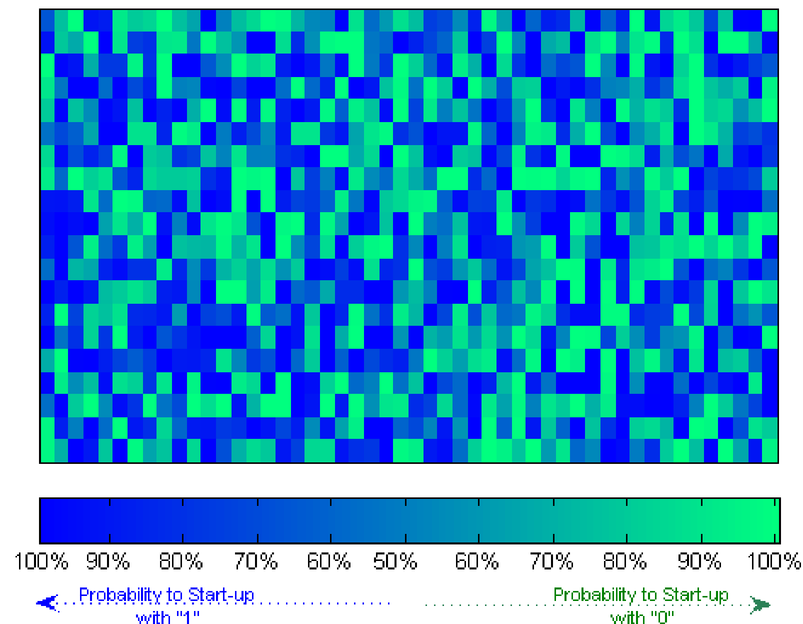


Fig.4.15: *Visualization of the used SRAM array showing the probability to start-up to a preferred SUV.*

The repeatability of SUVs may describe the immunity of SRAM against internal noise and reflect the strength of inherent cell-mismatch. A highly mismatched cell is able to repeat its SUV at every power-up. To support that, we have studied that relation between the probability to repeat the SUV (probability to start-up at logic "0" or logic "1") and inherent cell-mismatch (see Pd_{Vth} at section 3.2.1). We have found that well-matched

cells, low absolute Pd_{Vth} values, have very low repeatability, and their probability to start-up at "0" or "1" is around 50% (random SUV). While the strongly mismatched cells can have full repeatability considering the existence of the modeled noise. This result can be seen in Fig.4.16, where the correlation between the SUV probability and the MF is calculated for each memory cell. Each cell is represented by two stars: the blue star for probability of SUV towards logic "1", P("1"), and the green one for probability of SUV towards logic "0", P("0"). Where P("1") = 1-P("0"). So, both probability sets are complementary, and they are assigned to the same memory cells and highly related to Pd_{Vth}.

We also notice from Fig.4.16, that the cells with higher P("1") have the more negative Pd_{Vth}, while the cells with higher P(0) have more positive Pd_{Vth}. This correlation between the probability of SUV and the sign of the Pd_{Vth} is in line with the proposed metric assumption in section 3.2.1, where higher absolute value of mismatch-based metrics indicates more repeatable SUV and thus more reliable cells.

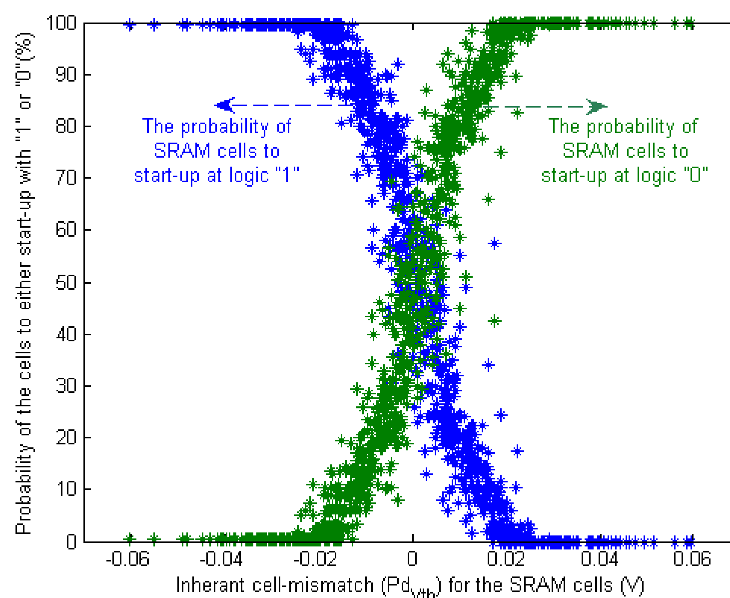


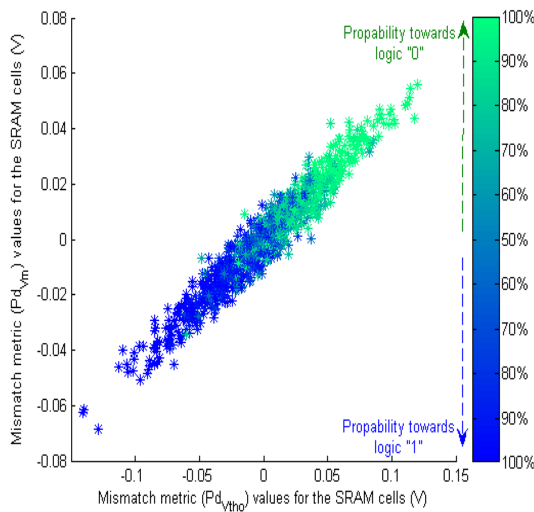
Fig.4.16: *The relation between inherent cell-mismatch and the probability of the cell to start-up to a preferred SUV.*

The most repeatable cells will be selected when they present a 100% of probability to repeat the SUV. A percentage of 31,5% cells from the proposed memory achieve this probability with the existence of the modeled noise. In the next chapter, we will use this set of cells (denoted as *Repeatable Cells*) in defining the *strong cell set* with the goal to correlate the characterization set using our proposed mismatch metrics.

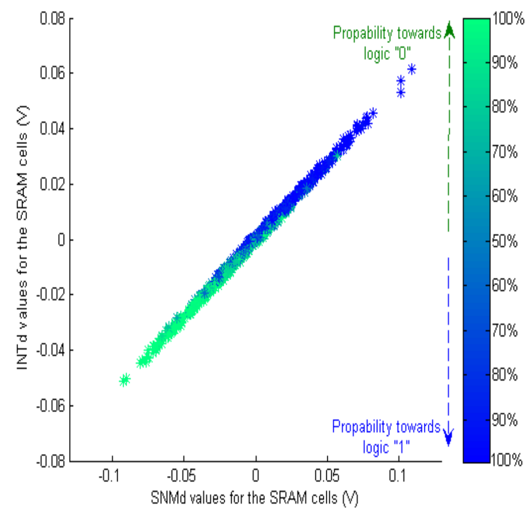
Finally, to observe the ability of all the proposed metrics in classifying cells repeatability against internal noise, we preset Fig.4.17 with relations similar to Figs.4.12 and 4.14 in the previous external perturbations. Where, Fig.4.17 (a) is for parameter distance-based metrics, Fig.4.17 (b) is for SNM-based metrics, Fig.4.17 (c) is for injected noise-based metrics. While in Fig.4.17 (d), we present the relation of SRAM separatrix metric (*SID*) with respect to Pd_{Vth} instead of *SID* histogram distribution; to clearly see the cells with colored probability distributed in *SID* relation.

Generally, the metrics achieved by transient simulations (see Fig.4.17 (c) and Fig.4.17 (d)) show better classification for cells repeatability than the metrics obtained by DC simulations (see Fig.4.17 (a) and Fig.4.17 (b)), this means that Transient-based metrics are more efficient to study the impact of internal noise. However, for all the metrics in Fig.4.17, the memory cells which have high SUVs repeatability (darker blue and green colored) correspond to the metrics high absolute values. While the memory cells with lower probability to repeat the same SUV correspond to lower metric magnitudes.

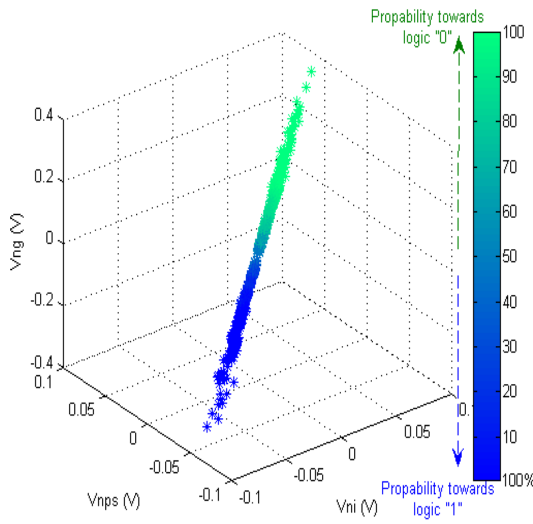
This shows the ability of the metrics in identifying most of cells that have repeatable SUVs to be implemented in PUF response. Additionally, the cells that have very low SUV probability (around 50%) may be useful for application such TRNG; as their SUVs are physically random.



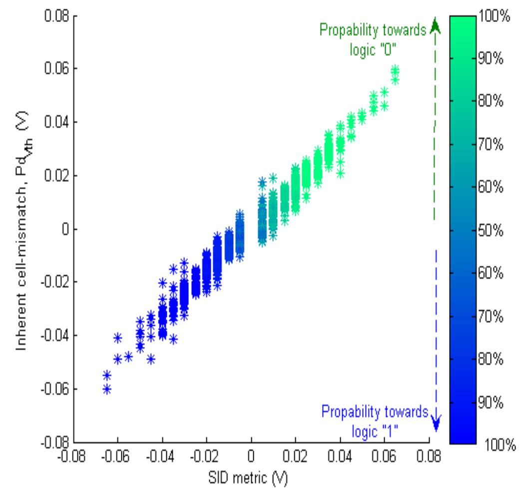
(a) **Ability of parameter distance metrics.**



(b) Ability of SNM metrics.



(C) Ability of injected noise metrics.



(d) Ability of *SID* metrics.

Fig.4.17: *The performance of the proposed metrics in classifying cells repeatability considering the modeled internal noise.*

CHAPTER 5

REPRODUCIBILITY CHARACTERIZATION AND PROPOSED METRICS DISCUSSION

Physically unclonable function assists to store and generate a secret key using SRAM circuit by taking profit of manufacturing process variation that occurs in its semiconductor devices. The secret key property of PUF is a unique regeneratable key, that makes it hard to predict or characterize the uncontrollable manufacturing variations. However, the main problem of SRAM-PUF is to assure its reliability under the external and internal conditions [4].

We have defined new metrics, in Chapter 3, to characterize the impact of different parameters on SUV repeatability of SRAM-PUF cells. The proposed metrics are obtained either using DC electrical simulations for both parameter distance and SNM based metrics, or using transient electrical simulations for both injected noise-based metrics and SRAM separatrix metric. Monte Carlo analysis is used to evaluate the metrics values for each memory cell with the main goal of predicting the percentage of suitable cells for PUF implementation at design phases. The obtained results are analyzed to characterize the strength of 65nm SRAM CMOS technology, when comparing SRAM-PUF behavior with noisy environment and considering different perturbation scenarios; as described in Chapter 4.

Based on the assumptions that cells with high metric values present a more stable and repeatable SUV, it is possible to identify the most suitable cells for PUF applications of a given set of cells in presence of process variations. As it is expected that these suitable cells can generate a more reliable PUF operation with wide range of operational conditions and more tolerance to noisy environments.

This chapter introduces a methodology that aims to predict by simulation the percentage of cells that will be suitable to be used as PUF generators. Firstly, we define the most suitable SRAM cells, denoted as Strong cells, that tolerate all the perturbations described in Chapter 4. Second section presents and compares the validity of proposed metrics in identifying the Strong cells, where the methodology of selecting those cells is also presented. Finally, using the metrics, we discuss the influence of selected response length on the reliability of PUF operation, where we estimate the percentage of cells that will be suitable for PUF applications.

5.1 Strong SRAM Cells to Improve PUF reliability

In the previous chapter, the results of several external perturbations and the modeled internal-noise indicate how the strength of SRAM-cell can be characterized by implementing the proposed reliability metrics that were defined in Chapter.3. In fact, the metrics showed good classification for the memory cells at each individual external perturbation. In this sense, if the SUV of a cell remains unchanged withstanding for all range of an individual perturbation, the cell was defined as stable respecting to this perturbation (*Stable-RUT cell*, *Stable-PSVs cell* and *Stable-Temperature cell*). By contrast, a cell was defined as unstable for an individual perturbation (*unstable-RUT cell*,

unstable-PSVs cell and *unstable-Temperature cell*), if its SUV varies due to this external perturbation within the considered range.

In this section, we define a cell as *Stable cell* if and only if it shows a constant SUV with all ranges of all external perturbations (RUT, PSVs and temperature), this type of cells should be selected for PUF applications. While a cell is defined as *Unstable cell* if it changes its SUV at any of the external perturbations.

In addition to the *Stable cell* , the most repeatable memory cells that have high SUV probabilities when internal noise was introduced (*Repeatable Cells*), that were also defined in the previous chapter (section 4.5), will complete the identification of best candidate for PUF applications. Hence, based on the results, the cells can be classified as follows:

- ***Strong SRAM cells***: a memory cell is defined as strong cell if and only if it is considered as *Stable cell* (has stable SUV against RUT, PSVs and temperature), and if it is included in *Repeatable cells* set (has repeatability probability equal to 100% against internal noise variability).

- ***Partial skewed SRAM cells***: The rest of the memory cells which are not classified as strong, will be defined as partial skewed cells. Therefore, all cells that have unstable SUV at any external perturbation (*Unstable cells*) and all cells that are not included in *Repeatable cell* set.

In order to improve PUF response reliability, only strong cells should be included to insure the reproducibility of the response under all conditions. Which will reduce the

need for Post-fabrication burn-in enhancement (see section 2.2.3.3) and for post process coding (see sections 2.2.3.1 and 2.2.3.2) to correct the reproducibility errors.

5.2 Selection of Strong Cells Using Mismatch Metrics

In this section, we present a methodology to implement the proposed metrics to select suitable memory cells for PUF and predict the percentage of cells available. Therefore, we can observe and compare the ability of proposed metrics in identifying the strong cells. Also, the results from previous work methodologies in the literature will be compared with this work metrics methodology to support the strength of our metrics in improving the reliability of SRAM-PUF.

5.2.1 Parameter Distance-Based Metrics Discussion

The start-up behavior of SRAM-PUF is mainly controlled by threshold voltage (*Vth*) mismatch of the transistors in cross-coupled inverters of SRAM cell, a highly mismatched cell is stronger and more reliable for PUF implementation [48-50]. In this work, both P-MOS and N-MOS voltage threshold mismatches are considered to study the cell-mismatch. We have implemented two parameter distance to evaluate the mismatch inside SRAM cell. Either by implementing a DC simulation to obtain either the threshold voltages for each individual transistor in cross coupled inverters to calculate parameter distance metric (Pd_{Vtho} in section 3.2.1), or utilizing the VTCs of cross coupled inverters to define the second novel parameter distance where both P-MOS and N-MOS threshold voltage mismatch are also included in this model (Pd_{Vm}) as shown in equation (3.12), section 3.2.2.

On the other hand, in [49], the authors use voltage threshold mismatch to define the suitable cells for PUF application. Their definition of the mismatch only considers the

voltage threshold mismatch of N-MOS transistors (*ΔN* as in equation (3.4) in section 3.2.1) while the mismatch in P-MOS transistors is neglected. By contrast, the work in [50] only considers the process variation and mismatch in P-MOS transistors ((*ΔP* as in equation (3.3), section 3.2.1) to classify memory cells.

The histogram distributions for *ΔN* and *ΔP* as proposed by [49] and [50] are shown in Fig.5.1 (a) and Fig.5.1 (b), respectively. Where the strong and the partial skewed cells are classified on *ΔN* and *ΔP* distributions. The red bars represent the percentage of the strong cells while the blue bars show the percentage of partial skewed cells. The methodology of using only *ΔN* results, see Fig.5.1 (a), in this is not able to classify the strong cells among its values; high or low absolute *ΔN* values will not indicate a high

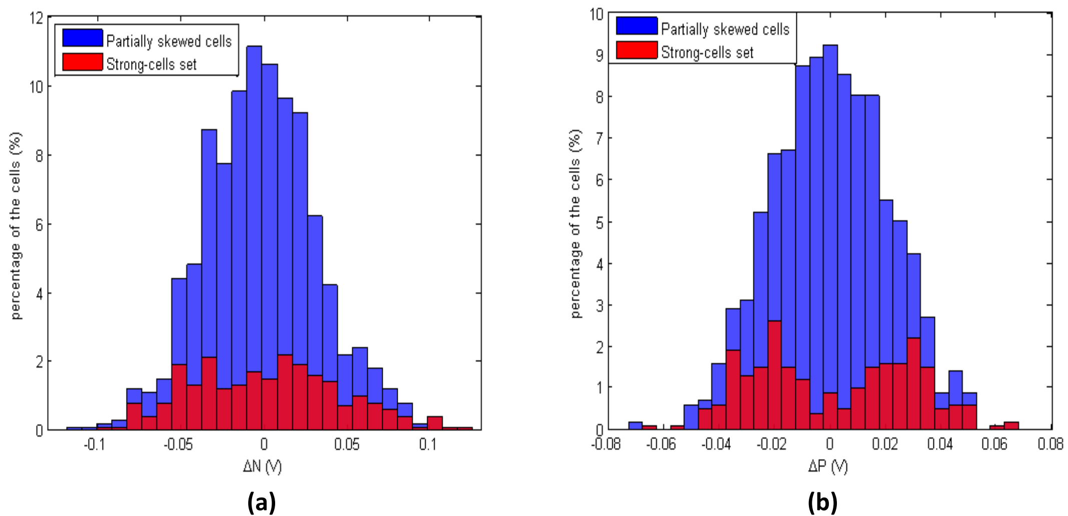


Fig.5.1: *The histograms for literature methodologies showing strong cells distribution: a) for ΔN methodology, b) for ΔP methodology.*

percentage of strong cells. In case of using only *ΔP* to classify the strength of memory cells, slightly better distribution of strong cells is shown in Fig.5.1 (b). As Lower absolute *ΔP* values (closer to *ΔP*=0 V) may indicate lower percentages of strong cell compared with higher absolute values. However, it can be noticed that some of the highest *ΔP* magnitudes correspond to partial skewed cells, disagreeing with the goal of this methodology and reducing the characterization efficiency.

Similarly, the histogram for the two proposed parameter distance-based metrics is presented in Fig.5.2, where both ΔP and ΔN are considered by the metrics. Fig.5.2 (a) shows the distribution of Pd_{Vtho} values, while Fig.5.2 (b) shows the distribution of Pd_{Vm} values. Also, the percentage of strong cells is highlighted in red color in these figures. It can be noticed that partial skewed cells are concentrated near the low absolute values of the metrics, while the strong cells correspond to higher metrics absolute values. These observations support that the parameter distance-based metrics can characterize and estimate the cells strength. In addition, the highest magnitude values of these metrics can identify robustly the strong cells that have high tolerance to external and internal perturbations.

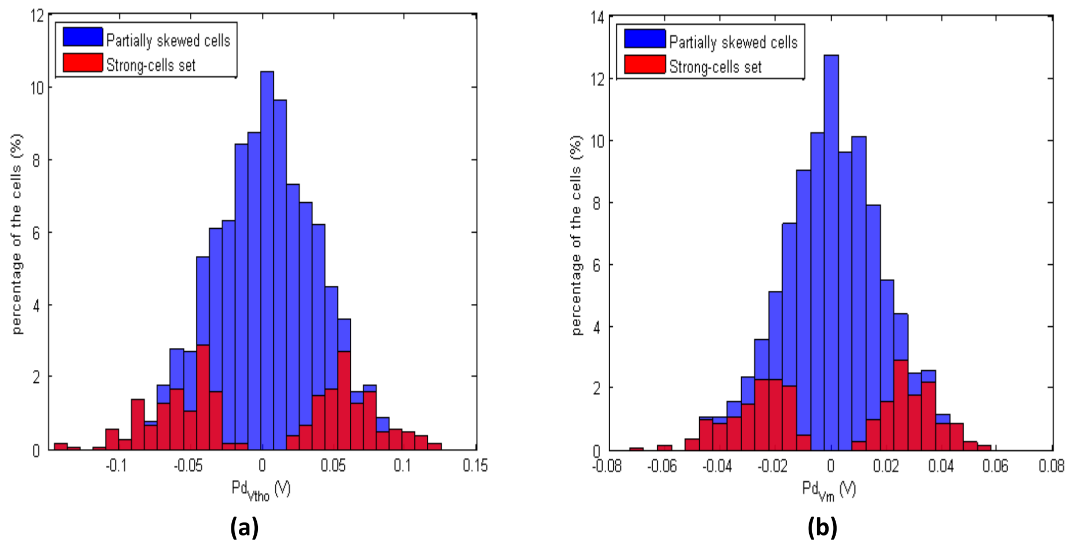


Fig.5.2: *The histograms for the parameter distances showing strong cells distribution: a) for Pd_{Vtho} metric, b) for Pd_{Vm} metric.*

The proposed strength characterization can efficiently detect the most suitable PUF cells based on the assumption that higher absolute values of parameter distance-based metrics can significantly improve the reliability of SRAM-PUF. In this sense, we select ***64*** *bits* to create PUF challenge-response pairs from the whole proposed memory. The selection of those *64 PUF-bits* is done by choosing the cells that have the highest metric

magnitudes. Table.5.1 summarizes the selection results for proposed metrics compared to the methodologies in [49] (*ΔN*) and [50] (*ΔP*), we also have introduced a random cells selection to support the effectiveness of the metrics. Each column of this table shows the number and percentage of cells that are selected as *Stable* (stable against external perturbations), *Repeatable* (100% repeatable SUV considering internal noise), and last column as *Strong cells* (*Stable* and *Repeatable*).

Table 5.1 Number of cells stable, repeatable and strong identified by Parameter distances metrics selecting 64 PUF-bits

Selection Method	*STABLE CELLS*	*REPEATABLE CELLS*	*STRONG CELLS*
ΔN as in [49]	49 (76.6%)	36 (56.3%)	34 (53.1%)
ΔP as in [50]	30 (46.9%)	56 (87.5%)	30 (46.9%)
Pd_{VM}	64 (100%)	56 (87.5%)	**56 (87.5%)**
Pd_{Vtho}	62 (96.9%)	60 (93.8%)	**60 (93.8%)**
Random Cells Selection	23 (35.9%)	20 (31.3%)	13 (20.3%)

It can be noticed that using only *ΔP* or only *ΔN* to classify cell strength is not efficient, as the selected PUF response has low percentage of strong cells (around 53% for *ΔN* and 47% for *ΔP*). However, the parameter distance-metrics show a good classification with high strong cells percentage in the selected PUF cells. The best identification results are achieved by Pd_{Vtho} metric, although the Pd_{Vm} metric is quite close. Comparing the achieved results with random cell selection, we observe how the metric methodology improves the reliability of SRAM-PUF when the cells that show better metric are the ones selected.

5.2.2 SNM-Based Metrics Discussion

To classify the reliability of PUF cells, the SNM concept is implemented to explain the SUV behavior of SRAM-PUF (section 3.3). In this sense, we have proposed two metric methodologies. These metrics are evaluated using DC simulations to obtain inverters VTC, where the intersection of those curves create the butterfly curve. The difference between the diagonal lengths of both biggest squares that could be fitted inside both eyes of the butterfly curve defines the first metric (SNM_d in section 3.3.2.1), while the distance between the intersection point of VTCs and the diagonal line (VQ=VQB line) defines the second novel metric (INT_d in section 3.3.2.2). From a reliability point of view, a high absolute value of these metrics defines the most reliable PUF cells. However, modeling the SUV behavior for SRAM-PUF using SNM was also introduced as a reliable PUF metric (PSNM ratio) in [43, 57-58]. This metric is defined as the ratio between both noise margin diagonals used to calculate the first metric (SNM_d). The cells that have higher or lower ratio than 1 are considered more reliable.

The histogram distribution for PSNM ratios is shown in Fig.5.3. Where the strong and the partial skewed cells are highlighted on PSNM ratio values. Similarly, red bars represent the percentage of the strong cells, while the blue bars show the percentage of partial skewed cells. Highest and lowest PSNM ratios correspond to the strong cells, indicating a good ability of this metric in classifying PUF cells.

To see the effectiveness of the SNM-based metric, we present similar histograms in Fig.5.4. Where Fig.5.4 (a) shows the strong cells classification ability for SNM_d metric and Fig.5.4 (b) present the ability for INT_d metric. In both figures, the percentage of strong

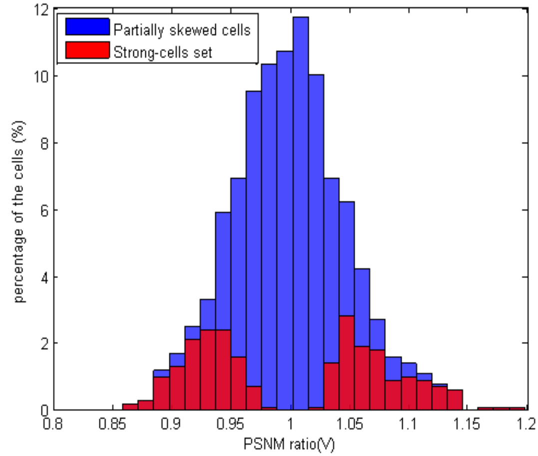


Fig.5.3: *The histogram distribution for literature PSNM ratio identifying the strong cells.*

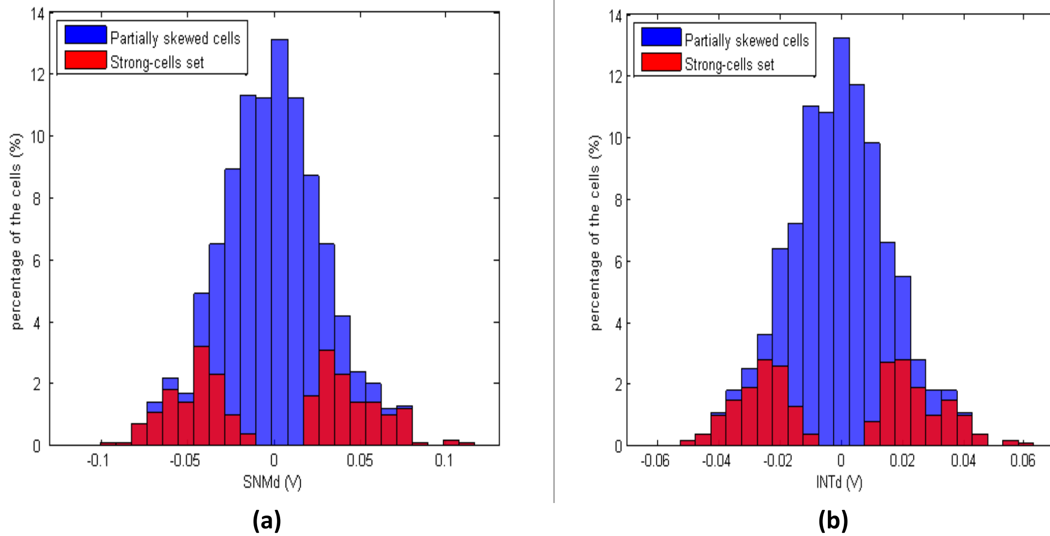


Fig.5.4: *The histograms for the proposed SNM metrics showing strong cells distribution: a) for SNM_d metric, b) for INT_d metric.*

cells is highlighted in red color. We can see that the strong cells are identified by higher metrics absolute values, while the partial skewed cells are concentrated near to the low absolute values (slightly better concentration than PSNM ratio in Fig.5.3). Based on these observations, the proposed SNM-based metrics can efficiently estimate the cells strength for PUF implementations.

Similarly, we selected *64 bits* creating PUF challenge-response pairs from the proposed memory, to characterize the strength of the proposed SNM-based metrics and compare

it with PSNM ratio [43]. These *64 PUF-bits* are selected by choosing also the cells that have highest absolute values of the proposed metrics, while choosing the lowest 32 ratios and the highest 32 ratios to create the 64 PUF-bits using PSNM ratio. The summary of the selection results for proposed metrics compared to PSNM ratio is presented in Table.5.2, where we also compare the results with 64 random cells selection. In the same way, the columns of this table represent the number of cells that are *Stable*, *Repeatable*, and *Strong cells* (*Stable* and *Repeatable*); respectively.

Table5. 2 Number of cells stable, repeatable and strong identified by SNM-based metrics selecting 64 PUF-bits

Selection Method	*STABLE CELLS*	*REPEATABLE CELLS*	*STRONG CELLS*
PSNM ratio as in [43]	63 (98.4%)	54 (84.4%)	54 (84.4%)
SNM_d	63 (98.4%)	54 (84.4%)	54 (84.4%)
INT_d	64 (100%)	57 (89.1%)	**57 (89.1%)**
Random Cells Selection	35 (54.7%)	23 (35.9%)	19 (29.7%)

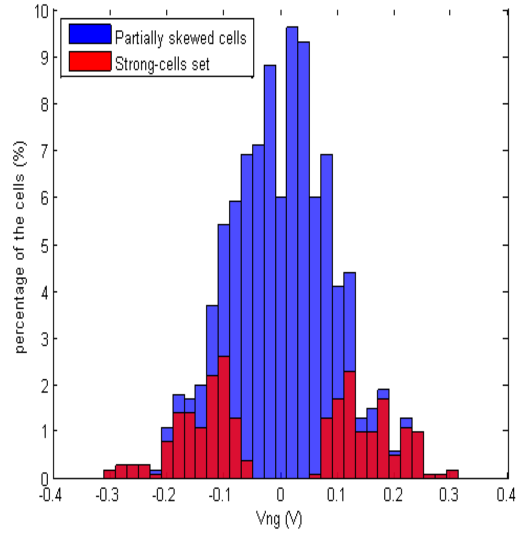
Here, both PSNM ratio and SNM_d have similar identification percentage results. Additionally, the novel INT_d metric achieves the best strong cell selection, despite the results of INT_d and SNM_d metrics are quite close and good; specially if the results are compared with random cell selection. Therefore, the proposed SNM-based metrics shows a good classification for the strength of SRAM cells and implementing these metrics in SRAM-PUF can also improve its the reliability.

5.2.3 Voltage Noise Injection-Based Metrics Discussion

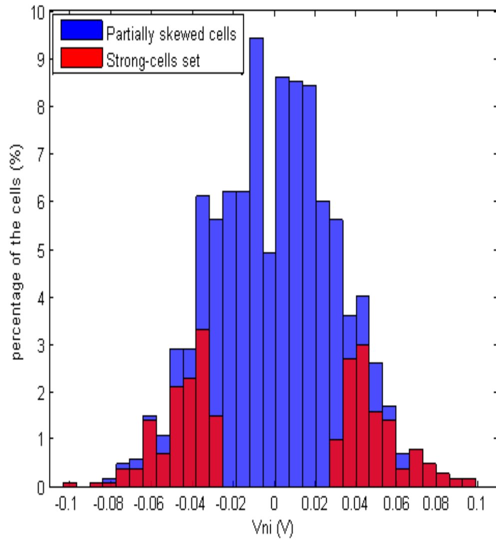
The metrics in this section are based on injecting a DC voltage noise at the SRAM cells to classify the immunity of the cells against this injected noise. The proposed metrics are obtained using transient simulations to evaluate the maximum voltage noise that can be tolerated by each cell in the memory. The difference between these metrics rely on the location of injected noise. Vn_g metric injects the noise at the ground node, Vn_i metric injects the noise between the cell's storage nodes, and Vn_{ps} injects the noise at the power supply nodes. Fig.5.5 shows the distribution of strong cells on the values of these metrics: Fig.5.5 (a) is for Vn_g metric, Fig.5.5 (b) is for Vn_i metric and Fig.5.5 (c) is for Vn_{ps} metric. In all of these figures, the partial skewed cells are highly concentrated at low absolute values of the metrics, with longer range of concentration is provided by Vn_i and Vn_{ps} metrics. Additionally, most of the strong cells are associated with higher metrics absolute values. So, the noise injection as a metric methodology is useful to classify the strength of SRAM cells for PUF applications.

Similar to the previous sections, Table 5.3 summarizes the selection results for proposed metrics, where the three metrics are implemented to select 64 bits for PUF challenge-response pairs.

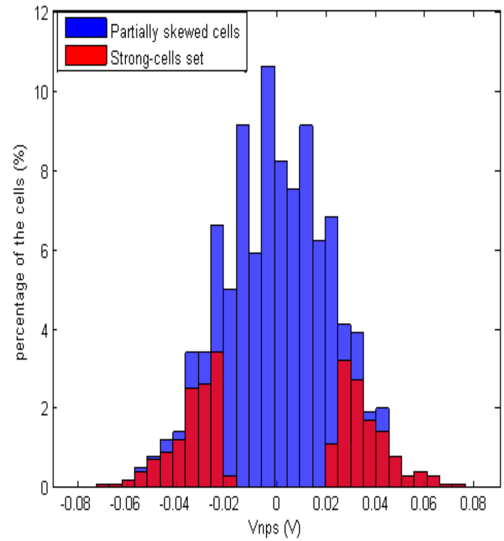
All the metrics in this table are able to select 100% repeatable cells, this agrees with definition of these metrics; as they describe the immunity of cells against internal noise. Therefore, higher metric magnitude corresponds to highly repeatable cell. Finally, all metrics show a decent classification with high percentages of strong cells in the selected PUF response. The best identification result is achieved by Vn_g metric, where the noise is injected at the ground of SRAM cell.



(a)



(b)



(C)

Fig.5.5: *The histograms for the proposed injected noise-based metrics showing strong cells distribution: a) for* Vn_g *metric, b) for* Vn_i *metric and c) for* Vn_g *metric*

Table 5.3 Number of cells stable, repeatable and strong identified by injected noise-based metrics selecting 64 PUF-bits

Selection method	STABLE CELLS	REPEATABLE CELLS	STRONG CELLS
Vn_g	57 (89.1%)	64 (100%)	**57 (89.1%)**
Vn_i	54 (84.4%)	64 (100%)	54 (84.4%)
Vn_{ps}	55 (85.9%)	64 (100%)	55 (85.9%)
Random Cells Selection	28 (43.8%)	21 (32.8%)	17 (26.6%)

5.2.4 SRAM Separatrix-Based Metrics Discussion

We proposed several indicators to classify the SRAM cells reliability based on their dynamic start-up process (see section 3.5). However, the SRAM separatrix metric (*SID*) seems the most promising one (subsection 3.5.2), as it highly correlates with inherent cell-mismatch compared to rest dynamic-based indicators. The *SID* metric describes cell mismatch in term of the cell tendency towards its final SUV; higher magnitudes of *SID* correspond to highly mismatched cells that have higher tendency towards the preferred SUV.

The distribution of the strong cells on the *SID* metric histogram is shown in Fig.5.6. Similar to the previous metrics, the percentage of the strong cells is represented in red bars and those cells are associated with highest magnitudes of *SID*. This reflects the strength of this metric in selecting the best PUF cells. The partially skewed cells (blue bars) are significantly identified by low magnitude of the metric, especially if we

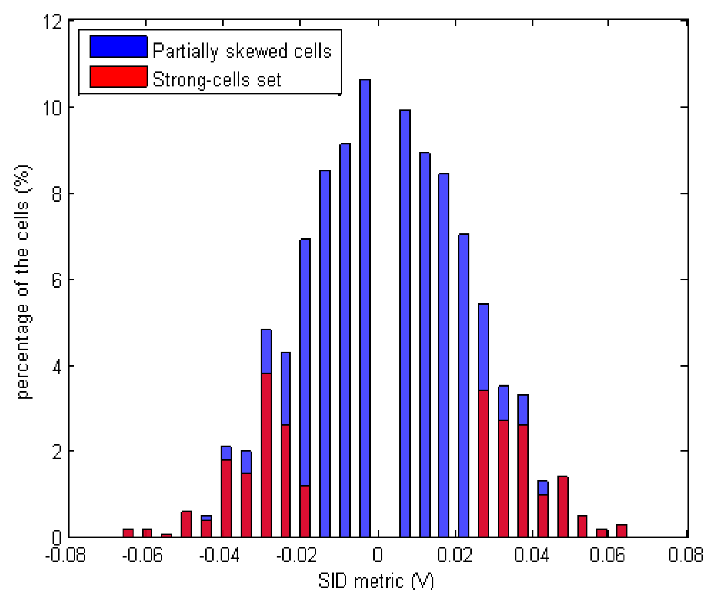


Fig.5.6: *The histogram distribution for dynamic SID metric identifying the strong cells.*

compare it with the partially skewed cells identification using the previous DC-based metrics.

We have applied *SID* metric to detect the most suitable PUF cells based on the definition that higher absolute values of *SID* metric can improve the reliability of SRAM-PUF. In this sense, we select *64 bits* to create PUF challenge-response pairs similar to the previous metrics. The selection of these *64 PUF-bits* is done starting from the cells that have the highest metric magnitudes.

The results show that this metric is able to select 59 *Stable cells* (92.2%), while all the selected PUF cells are identified as *Repeatable* (100%). However, the *Strong cells* represent **92.2%** of the selected cells. Similarly, this result shows the classification strength of *SID* metric, as it can identify the cells that highly protected against internal noise with high tolerance to extreme external perturbations. Therefore, the transient *SID* metric is one of the best metrics in this work to estimate the reliability of SRAM-PUF.

5.2.5 Summary

The strength characterization using the proposed metrics presents high performance in identifying the suitable memory cells for PUF applications. The results that achieved by the metrics are quite close between them, but there are some differences. Among the proposed parameter distance metrics, $\boldsymbol{Pd_{Vtho}}$ is the best metric in classifying the strong cells (**93.8%**). The $\boldsymbol{INT_d}$ is the best metric to represent the SNM metrics, this metric is able to identify **89.1%** of the strong cells in the selected response. While $\boldsymbol{Vn_g}$ metric identifies the highest number of strong cells (**89.1%**) among the three injected noise locations. Finally, the dynamic ***SID*** metric can select up to (**92.2%**) of the strong cells.

Based on that, the best metric obtained by DC simulations is the parameter distance (***Pd_{Vtho}***), while the SRAM Separatrix (***SID***) is the best metric obtained by transient simulations. However, the DC simulations are much faster than transient ones. Additionally, the ***Pd_{Vtho}*** achieves the best selection results between all the proposed metrics.

5.3 Influence of Selected PUF Response Length

Reducing the length of the selected PUF-response, by selecting minimum required number of PUF-cells, can improve the overall PUF operation time [59]. Additionally, the memory addresses used for PUF operation must remain uninitialized until the PUF-response is generated, and cannot be used for other purposes, as claimed in [77].

However, the reliability of selected PUF-response using the proposed metrics is highly affected by the response length. As we mentioned in the previous section, the strong cells are more concentrated at highest metric values and the concentration of these cells

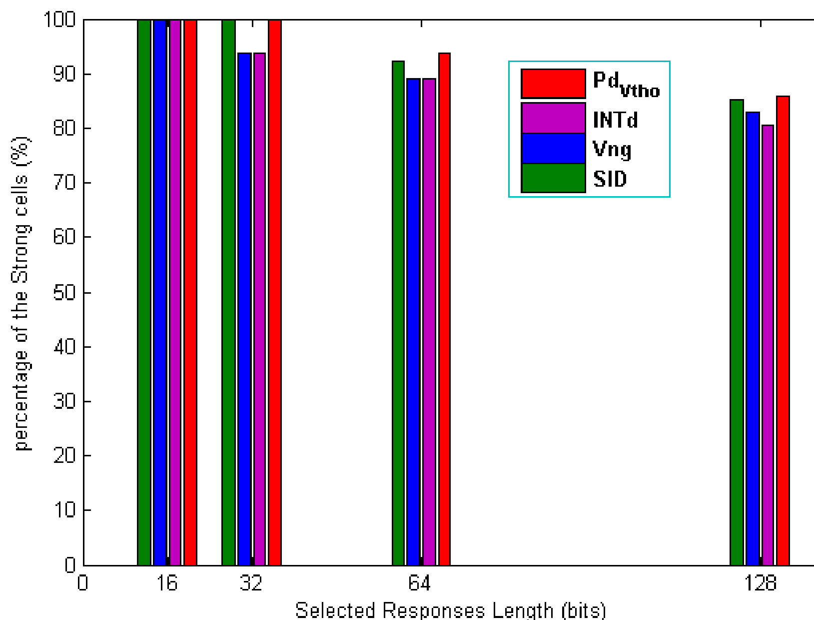


Fig.5.7: *Percentage of strong cells identified by each best metric approach considering different response lengths.*

decreases with metric values reduction. The PUF-response is selected starting from highest metric absolute values. Therefore, if the response length decreases, higher percentage of strong cells will be included in the response, and thus the reliability of this PUF-response increases. Fig.5.7 represents the percentages of strong cells identified by each best metric approach with respect to several length of bits (16, 32, 64 and 128 bits) creating the response of the SRAM-PUF. It is clear that the percentage of identified strong cells increases when the response length is reduced. In all cases, the Pd_{Vtho} metric achieves the maximum percentage, despite the *SID* metric is quite close. Generally, all the proposed metrics in Fig.5.7 that based on characterize the cell strength are the author recommendation. Finally, a percentage of strong cells over 90% is achieved if only a subset of 32 cells is selected. In this case, those cells represent roughly a 3% of the total 1000 cells. If only 16 cells are selected, all of them are strong cells, which is a 1,5% of the 1000 cells.

The main benefit of this selection technique is allowing the designer to make this kind of predictions by simulation, I.E. the overall percentage of PUF suitable cells that will be available in a certain memory design. In addition, this methodology allows determining, the minimum size of the memory array needed if a fixed number of strong cells is necessary for given PUF application.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusions

SRAM memories are becoming one of the most attractive alternatives for the implementation of PUFs. To ensure their SUV reliability it is necessary to quantify the impact produced by temperature, previously stored values, thermal noise or ramp-up characteristics. In this sense, some cells are more affected by these perturbations than the others. The SUV of well-matched (more symmetrical) cell is more affected, while the highly mismatched cell has more constant SUV under the disturbances. To avoid the effects of these disturbances, it has been proposed as a solution to implement PUFs using only a subset composed of the most reliable memory bit-cells; those that have the most reproducible SUV.DC and transient simulated metric methodologies to characterize SRAM cells for PUF applications are implemented on 65nm CMOS technology node. These methodologies are used to study the reliability of cell SUV based on either the dynamic cell behavior or the static cell parameters, like transistor threshold voltage. The metrics implementation is further extended for the percentage estimation of strong cells. Additionally, external perturbations and internal noise techniques are implemented to support the strong cell identification capability of the proposed metrics. The threshold voltage distance (Pd_{Vth}) metric shows the best results, indicating that the mismatch in transistors threshold voltage parameter is crucial in operation and reliability of SRAM as a PUF. Also, based on DC simulation, the SNM-based metrics show

excellent correlation between them and between previous published metric (PSNM ratio). However, the characterization result of *INTd* metric is slightly better. On the other hand, the SRAM separatrix implementation (*SID* metric) shows the best result between all the transient metrics.

In general, all achieved results show good agreement between metrics-based percentage estimation methodology and strong cells. The mismatch characterization of SRAM cells using either the DC simulated metrics or the transient simulated metrics can reduce the need for massive test simulations to achieve substantial statistics for SRAM reliability analysis. Furthermore, by only implementing the strongest cells as PUF, it will generate more repeatable and consistent output each time it is challenged, reducing the need for postprocessing ECCs. In contrary, the weakest skewed cells are highly affected by environmental and operational conditions and their SUVs are expected to be more random. Therefore, selecting only these cells can contribute to applications like true random number generators.

The characterization of bit-cells using the proposed metrics can be exploited to explore the tolerance margins and evaluate the benefits and drawbacks of different SRAM PUF implementations against different scenarios, such as temperature.

6.2 Future Work

Based on the findings offered in this thesis, some directions for future research are proposed and described in this section:

1. Some of the metrics, Such as Pd_{Vth}, consider only the variation in one parameter in one of the cell transistors at a time. But, in real scenario, there are many

parameters can be affected by process variation. Even though, the threshold voltage variation will dominate the SUV as shown by the metrics, these metrics can be extended to study the variation in more than one parameter. However, these parameters can have conflicted impacts on the SUV reliability; the impact of some of them may increase the reliability while others may decrease it. To achieve the optimum reliability of SRAM-PUF, an optimization algorithm could be used to optimize the values of these parameters at design phase.

2. Different technology nodes will have different impacts on SRAM-PUFs, as reducing the technology scaling will increase the influence of process variation. The metrics developed can be implemented to explore the effect of technology scaling on SUV of SRAM cell, also to explore the percentage estimation of strong cells among these technology nodes. Additionally, different memory structures, such as 8-T and 10-T SRAM cell, could be studied using similar metric methodology.
3. The characterization of SRAM cell reproducibility for PUF application using the developed metrics has several benefits for pre-design and design stages. Although, measuring the metric parameters experimentally is quite difficult. This work can be extended by experimentally validating the estimated percentage of strong cells for an SRAM design.

REFERENCES

[1] Skorobogatov, S.P., 2005. Semi-invasive attacks: a new approach to hardware security analysis.

[2] Pappu, R.S., 2001. Physical one-way functions [Ph. D. thesis]. *Massachusetts Institute of Technology, Cambridge, Mass, USA*.

[3] Maes, R., Tuyls, P. and Verbauwhede, I., 2009, June. A soft decision helper data algorithm for SRAM PUFs. In *2009 IEEE international symposium on information theory,* pp. 2101-2105.

[4] Böhm, C. and Hofer, M., 2012. *Physical unclonable functions in theory and practice*. Springer Science & Business Media.

[5] Gao, Y., Ranasinghe, D.C., Al-Sarawi, S.F., Kavehei, O. and Abbott, D., 2016. Emerging physical unclonable functions with nanotechnology. *IEEE access*, 4, pp.61-80.

[6] Delvaux, J., Gu, D., Schellekens, D. and Verbauwhede, I., 2014. Helper data algorithms for PUF-based key generation: Overview and analysis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *34*(6), pp.889-902.

[7] Maes, R., Van Herrewege, A. and Verbauwhede, I., 2012, September. PUFKY: A fully functional PUF-based cryptographic key generator. In *International Workshop on Cryptographic Hardware and Embedded Systems* (pp. 302-319). Springer, Berlin, Heidelberg.

[8] Suh, G.E. and Devadas, S., 2007, June. Physical unclonable functions for device authentication and secret key generation. In *2007 44th ACM/IEEE Design Automation Conference* (pp. 9-14). IEEE.

[9] Delvaux, J., Peeters, R., Gu, D. and Verbauwhede, I., 2015. A survey on lightweight entity authentication with strong PUFs. *ACM Computing Surveys (CSUR)*, *48*(2), pp.1-42.

[10] Gassend, B., Van Dijk, M., Clarke, D. and Devadas, S., 2007. Controlled physical random functions. In *Security with Noisy Data* (pp. 235-253). Springer, London.

[11] Holcomb, D.E., Burleson, W.P. and Fu, K., 2008. Power-up SRAM state as an identifying fingerprint and source of true random numbers. *IEEE Transactions on Computers*, *58*(9), pp.1198-1210.

[12] Van der Leest, V., Van der Sluis, E., Schrijen, G.J., Tuyls, P. and Handschuh, H., 2012. Efficient implementation of true random number generator based on sram pufs. In *Cryptography and security: from theory to applications* (pp. 300-318). Springer, Berlin, Heidelberg.

[13] Varchola, M., Drutarovsky, M. and Fischer, V., 2013, December. New universal element with integrated PUF and TRNG capability. In *2013 International Conference on Reconfigurable Computing and FPGAs (ReConFig)* (pp. 1-6). IEEE.

[14] Van Der Sluis, E., Schrijen, G.J. and Handschuh, H., Intrinsic ID BV, 2016. *Random number generating system based on memory start-up noise*. U.S. Patent 9,383,969.

[15] Rührmair, U., Sehnke, F., Sölter, J., Dror, G., Devadas, S. and Schmidhuber, J., 2010, October. Modeling attacks on physical unclonable functions. In *Proceedings of the 17th ACM conference on Computer and communications security* (pp. 237-249).

[16] Handschuh, H., 2012. Hardware-anchored security based on SRAM PUFs, Part 1. *IEEE Security & Privacy*, *10*(3), pp.80-83.

[17] Maiti, A. and Schaumont, P., 2013. The impact of aging on a physical unclonable function. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *22*(9), pp.1854-1864.

[18] Bhargava, M. and Mai, K., 2014, March. An efficient reliable PUF-based cryptographic key generator in 65nm CMOS. In *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE)* (pp. 1-6). IEEE.

[19] Lao, Y., Yuan, B., Kim, C.H. and Parhi, K.K., 2016. Reliable PUF-based local authentication with self-correction. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *36*(2), pp.201-213.

[20] Rhee, K., Kwak, J., Kim, S. and Won, D., 2005, April. Challenge-response based RFID authentication protocol for distributed database environment. In *International Conference on Security in Pervasive Computing* (pp. 70-84). Springer, Berlin, Heidelberg.

[21] Dargar, A., 2011. Modeling SRAM Start-up Characteristics for Physical Unclonable Functions (Master's thesis, Delft University of Technology).

[22] Skoric, B., Schrijen, G.J., Ophey, W., Wolters, R., Verhaegh, N. and van Geloven, J., 2007. Experimental hardware for coating PUFs and optical PUFs. In *Security with Noisy Data* (pp. 255-268). Springer, London.

[23] Lu, X., Hong, L. and Sengupta, K., 2018. CMOS optical PUFs using noise-immune process-sensitive photonic crystals incorporating passive variations for robustness. *IEEE Journal of Solid-State Circuits*, *53*(9), pp.2709-2721.

[24] Zhang, J.L., Qu, G., Lv, Y.Q. and Zhou, Q., 2014. A survey on silicon PUFs and recent advances in ring oscillator PUFs. *Journal of computer science and technology*, *29*(4), pp.664-678.

[25] Kumar, S.S., Guajardo, J., Maes, R., Schrijen, G.J. and Tuyls, P., 2008, June. The butterfly PUF protecting IP on every FPGA. In *2008 IEEE International Workshop on Hardware-Oriented Security and Trust* (pp. 67-70). IEEE.

[26] Böhm, C., Hofer, M. and Pribyl, W., 2011, September. A microcontroller sram-puf. In *2011 5th International Conference on Network and System Security* (pp. 269-273). IEEE.

[27] Holcomb, D.E., Burleson, W.P. and Fu, K., 2007, July. Initial SRAM state as a fingerprint and source of true random numbers for RFID tags. In *Proceedings of the Conference on RFID Security* (Vol. 7, No. 2, p. 01).

[28] Yu, M.D. and Devadas, S., 2010. Secure and robust error correction for physical unclonable functions. *IEEE Design & Test of Computers*, *27*(1), pp.48-65.

[29] Suh, G.E., O'Donnell, C.W. and Devadas, S., 2007. Aegis: A single-chip secure processor. *IEEE Design & Test of Computers*, *24*(6), pp.570-580.

[30] Bösch, C., Guajardo, J., Sadeghi, A.R., Shokrollahi, J. and Tuyls, P., 2008, August. Efficient helper data key extractor on FPGAs. In *International workshop on cryptographic hardware and embedded systems* (pp. 181-197). Springer, Berlin, Heidelberg.

[31] Islam, M.N., Patil, V.C. and Kundu, S., 2017. On enhancing reliability of weak PUFs via intelligent post-silicon accelerated aging. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *65*(3), pp.960-969.

[32] Usmani, M.A., Keshavarz, S., Matthews, E., Shannon, L., Tessier, R. and Holcomb, D.E., 2018. Efficient PUF-based key generation in FPGAs using per-device configuration. *IEEE Transactions on very large scale integration (VLSI) systems*, *27*(2), pp.364-375.

[33] Patil, V.C., Vijayakumar, A., Holcomb, D.E. and Kundu, S., 2017, May. Improving reliability of weak PUFs via circuit techniques to enhance mismatch. In *2017 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)* (pp. 146-150). IEEE.

[34] Mathew, S.K., Satpathy, S.K., Anders, M.A., Kaul, H., Hsu, S.K., Agarwal, A., Chen, G.K., Parker, R.J., Krishnamurthy, R.K. and De, V., 2014, February. 16.2 A 0.19 pJ/b PVT-variation-tolerant hybrid physically unclonable function circuit for 100% stable secure key generation in 22nm CMOS. In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)* (pp. 278-279). IEEE.

[35] Guajardo, J., Kumar, S.S., Schrijen, G.J. and Tuyls, P., 2007, August. Physical unclonable functions and public-key crypto for FPGA IP protection. In *2007 International Conference on Field Programmable Logic and Applications* (pp. 189-195). IEEE.

[36] Sklavos, N., Chaves, R., Di Natale, G. and Regazzoni, F., 2017. Hardware security and trust. *Cham, Switzerland: Springer*.

[37] Li, J., Yang, T. and Seok, M., 2017, May. A technique to transform 6T-SRAM arrays into robust analog PUF with minimal overhead. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 1-4). IEEE.

[38] Maes, R. and Van Der Leest, V., 2014, May. Countering the effects of silicon aging on SRAM PUFs. In *2014 IEEE International symposium on hardware-oriented security and trust (HOST)* (pp. 148-153). IEEE.

[39] Bhargava, M. and Mai, K., 2013, August. A high reliability PUF using hot carrier injection based response reinforcement. In *International Conference on Cryptographic Hardware and Embedded Systems* (pp. 90-106). Springer, Berlin, Heidelberg.

[40] Satpathy, S., Mathew, S.K., Suresh, V., Anders, M.A., Kaul, H., Agarwal, A., Hsu, S.K., Chen, G., Krishnamurthy, R.K. and De, V.K., 2017. A 4-fJ/b delay-hardened physically

unclonable function circuit with selective bit destabilization in 14-nm trigate CMOS. *IEEE Journal of Solid-State Circuits*, *52*(4), pp.940-949.

[41] Baturone, I., Prada-Delgado, M.A. and Eiroa, S., 2015. Improved generation of identifiers, secret keys, and random numbers From SRAMs. *IEEE Transactions on Information Forensics and Security*, *10*(12), pp.2653-2668.

[42] Pandey, S., Deyati, S., Singh, A. and Chatterjee, A., 2016, November. Noise-resilient SRAM physically unclonable function design for security. In *2016 IEEE 25th Asian Test Symposium (ATS)* (pp. 55-60). IEEE.

[43] Cortez, M., Dargar, A., Hamdioui, S. and Schrijen, G.J., 2012, October. Modeling SRAM start-up behavior for physical unclonable functions. In *2012 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)* (pp. 1-6). IEEE.

[44] Shifman, Y., Miller, A., Keren, O., Weizmann, Y. and Shor, J., 2018. A Method to Improve Reliability in a 65-nm SRAM PUF Array. *IEEE Solid-State Circuits Letters*, *1*(6), pp.138-141.

[45] Karpinskyy, B., Lee, Y., Choi, Y., Kim, Y., Noh, M. and Lee, S., 2016, January. 8.7 Physically unclonable function for secure key generation with a key error rate of 2E-38 in 45nm smart-card chips. In *2016 IEEE International Solid-State Circuits Conference (ISSCC)* (pp. 158-160). IEEE.

[46] Böhm, C., Bucci, M., Hofer, M. and Luzzi, R., 2016. A reliable low-area low-power PUF-based key generator.

[47] Rajput, A.S., Pattanaik, M. and Tiwari, R.K., 2018. Estimation of static noise margin by butterfly method using curve-fitting technique. *Journal of Active and Passive Electronic Devices*, *13*(1), pp.1-9.

[48] Chellappa, S., 2011, September. Improved circuits for microchip identification using SRAM mismatch. In *2011 IEEE Custom Integrated Circuits Conference (CICC)* (pp. 1-4). IEEE.

[49] Hofer, M. and Boehm, C., 2010, August. An alternative to error correction for SRAM-like PUFs. In *International Workshop on Cryptographic Hardware and Embedded Systems* (pp. 335-350). Springer, Berlin, Heidelberg.

[50] Kiamehr, S., Golanbari, M.S. and Tahoori, M.B., 2017, March. Leveraging aging effect to improve SRAM-based true random number generators. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017* (pp. 882-885). IEEE.

[51] Okumura, S., Yoshimoto, S., Kawaguchi, H. and Yoshimoto, M., 2012. A 128-bit Chip Identification Generating Scheme Exploiting Load Transistors' Variation in SRAM Bitcells. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, *95*(12), pp.2226-2233.

[52] Shinohara, H., Zheng, B., Piao, Y., Liu, B. and Liu, S., 2017, April. Analysis and reduction of SRAM PUF bit error rate. In *2017 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)* (pp. 1-4). IEEE.

[53] Baker, R.J., 2019. *CMOS: circuit design, layout, and simulation*. John Wiley & Sons.

[54] Rabaey, J.M., Chandrakasan, A.P. and Nikolić, B., 2003. *Digital integrated circuits: a design perspective* (Vol. 7). Upper Saddle River, NJ: Pearson education.

[55] Seevinck, E., List, F.J. and Lohstroh, J., 1987. Static-noise margin analysis of MOS SRAM cells. *IEEE Journal of solid-state circuits*, *22*(5), pp.748-754.

[56] Pavlov, A. and Sachdev, M., 2008. *CMOS SRAM circuit design and parametric test in nano-scaled technologies: process-aware SRAM design and test* (Vol. 40). Springer Science & Business Media.

[57] Roelke, A. and Stan, M.R., 2018. Controlling the reliability of SRAM PUFs with directed NBTI aging and recovery. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *26*(10), pp.2016-2026.

[58] Lee, J., Jee, D.W. and Jeon, D., 2019. Power-up control techniques for reliable SRAM PUF. *IEICE Electronics Express*, *16*(13), pp.20190296-20190296.

[59] Eiroa, S., Castro, J., Martínez-Rodríguez, M.C., Tena, E., Brox, P. and Baturone, I., 2012, December. Reducing bit flipping problems in SRAM physical unclonable functions for chip identification. In *2012 19th IEEE International Conference on Electronics, Circuits, and Systems (ICECS 2012)* (pp. 392-395). IEEE.

[60] Yin, C.E. and Qu, G., 2009, July. Temperature-aware cooperative ring oscillator PUF. In *2009 IEEE International Workshop on Hardware-Oriented Security and Trust* (pp. 36-42). IEEE.

[61] Guo, Z., Carlson, A., Pang, L.T., Duong, K.T., Liu, T.J.K. and Nikolic, B., 2009. Large-scale SRAM variability characterization in 45 nm CMOS. *IEEE Journal of Solid-State Circuits*, *44*(11), pp.3174-3192.

[62] Schrijen, G.J. and Van Der Leest, V., 2012, March. Comparative analysis of SRAM memories used as PUF primitives. In *2012 Design, Automation & Test in Europe Conference & Exhibition (DATE)* (pp. 1319-1324). IEEE.

[63] Claes, M., van der Leest, V. and Braeken, A., 2011, October. Comparison of SRAM and FF PUF in 65nm technology. In *Nordic Conference on Secure IT Systems* (pp. 47-64). Springer, Berlin, Heidelberg.

[64] Zhang, B., Arapostathis, A., Nassif, S. and Orshansky, M., 2006, November. Analytical modeling of SRAM dynamic stability. In *Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design* (pp. 315-322).

[65] Vatajelu, E.I., Panagopoulos, G., Roy, K. and Figueras, J., 2010, May. Parametric failure analysis of embedded SRAMs using fast & accurate dynamic analysis. In *2010 15th IEEE European Test Symposium* (pp. 69-74). IEEE.

[66] Vătăjelu, E.I., Gómez-Pau, Á., Renovell, M. and Figueras, J., 2014. Sram cell stability metric under transient voltage noise. *microelectronics Journal*, *45*(10), pp.1348-1353.

[67] Dong, W., Li, P. and Huang, G.M., 2008, November. SRAM dynamic stability: Theory, variability and analysis. In *2008 IEEE/ACM International Conference on Computer-Aided Design* (pp. 378-385). IEEE.

[68] Zhang, Y., Li, P. and Huang, G.M., 2010, June. Separatrices in high-dimensional state space: System-theoretical tangent computation and application to SRAM dynamic stability analysis. In *Proceedings of the 47th Design Automation Conference* (pp. 567-572).

[69] Sharifkhani, M. and Sachdev, M., 2009. SRAM cell stability: A dynamic perspective. *IEEE Journal of Solid-State Circuits*, *44*(2), pp.609-619.

[70] Vatajelu, E.I., Di Natale, G. and Prinetto, P., 2016, March. Towards a highly reliable SRAM-based PUFs. In *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)* (pp. 273-276). IEEE.

[71] Lundberg, K.H., 2002. Noise sources in bulk CMOS. *Unpublished paper*, *3*, p.28.

[72] Holcomb, D.E. and Fu, K., 2014, September. Bitline PUF: building native challenge-response PUF capability into any SRAM. In *International Workshop on Cryptographic Hardware and Embedded Systems* (pp. 510-526). Springer, Berlin, Heidelberg.

[73] Cortez, M., Hamdioui, S., Kaichouhi, A., van der Leest, V., Maes, R. and Schrijen, G.J., 2015. Intelligent voltage ramp-up time adaptation for temperature noise reduction on memory-based PUF systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *34*(7), pp.1162-1175.

[74] Wang, W., Singh, A., Guin, U. and Chatterjee, A., 2018, March. Exploiting power supply ramp rate for calibrating cell strength in SRAM PUFs. In *2018 IEEE 19th Latin-American Test Symposium (LATS)* (pp. 1-6). IEEE.

[75] Ho, A., 2017. *Circuit Design of SRAM Physically Unclonable Functions* (Master's thesis, University of Waterloo).

[76] Liu, M., Zhou, C., Tang, Q., Parhi, K.K. and Kim, C.H., 2017, July. A data remanence based approach to generate 100% stable keys from an sram physical unclonable function. In *2017 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)* (pp. 1-6). IEEE.

[77] Saxena, N. and Voris, J., 2009, July. We can remember it for you wholesale: Implications of data remanence on the use of RAM for true random number generation on RFID tags. In *Proceedings of the Conference on RFID Security*.

[78] Yang, K., Blaauw, D. and Sylvester, D., 2017. Hardware designs for security in ultra-low-power IoT systems: An overview and survey. *IEEE Micro*, *37*(6), pp.72-89.

[79] Toh, S.O., Guo, Z., Liu, T.J.K. and Nikolic, B., 2011. Characterization of dynamic SRAM stability in 45 nm CMOS. *IEEE journal of solid-state circuits*, *46*(11), pp.2702-2712.

[80] Alorda, B., Carmona, C., Torrens, G. and Bota, S., 2016. An affordable experimental technique for SRAM write margin characterization for nanometer CMOS technologies. *Microelectronics Reliability*, *65*, pp.280-288.