Design for security in mixed analog-digital integrated circuits

Mohamed Elshamy

THÈSE DE DOCTORAT
DE SORBONNE UNIVERSITÉ

DESIGN FOR SECURITY IN MIXED ANALOG-DIGITAL INTEGRATED CIRCUITS

présentée par

MOHAMED ELSHAMY

École Doctorale Informatique, Télécommunications et Électronique

réalisée au

Laboratoire d'Informatique Paris 6

soutenue le 7 Juillet 2021

devant le jury composé de :

M.	Yiorgos MAKRIS	Prof., UT Dallas, USA	Rapporteur
M.	Bruno ROUZEYRE	Prof., LIRMM, Montpellier, France	Rapporteur
Mme.	Sule OZEV	Prof., Arizona State University, USA	Examinatrice
M.	Giorgio DI NATALE	DR CNRS, TIMA, Grenoble, France	Examinateur
M.	Vincent BEROULLE	Prof., LCIS, Valence, France	Examinateur
M.	Michele PORTOLAN	MCF, TIMA, Grenoble, France	Examinateur
Mme.	Marie-Minerve LOUËRAT	CR CNRS, LIP6, Paris, France	Co-directrice de Thèse
M.	Haralampos STRATIGOPOULOS	DR CNRS, LIP6, Paris, France	Directeur de Thèse
M.	Hassan ABOUSHADY	MCF, LIP6, Paris, France	Invité

To my family, without your support and encouragement I would never reach this point.

ABSTRACT

Recently, the enormous cost of owning and maintaining a modern semiconductor manufacturing plant has coerced many companies to go fabless. By outsourcing the manufacturing of integrated circuit/intellectual property (IC/IP) to third-party and often off-shore companies, the process has been extended to potentially untrustworthy companies. This has resulted in several security threats to the semiconductor industry such as counterfeiting, reverse engineering, and hardware Trojans (HTs) insertion. In this thesis, we propose an anti-piracy countermeasure to protect analog and mixed-signal (AMS) ICs/IPs, a novel HT attack for AMS ICs/IPs, and a novel physical unclonable function (PUF).

More specifically, we propose an anti-piracy technique based on locking for programmable analog circuits. The proposed technique leverages the programmability fabric to implement a natural lock-less locking. We discuss its implementation and its resilience capabilities against foreseen attacks, and we demonstrate it with simulation analysis and hardware measurements of programmable $\Sigma\Delta$ Analog-to-Digital Converters (ADCs) intended for use in highly-digitized, multi-standard RF receiver applications. The proposed HT attack for analog circuits leverages the test infrastructure. The HT is hidden effectively in a digital core and transfers its payload to the analog circuit via the test bus and the interface of the analog circuit to the test bus. Its key characteristic is that it is invisible in the analog domain. The proposed HT is demonstrated on two case studies. The first case study shown with simulation is a low-dropout (LDO) regulator and the second case study shown with hardware measurements is a programmable RF receiver front-end. This thesis sheds light on the importance of developing new security and trust countermeasures tailored for analog circuits. The proposed locking technique illustrates the feasibility of securing a large number of analog circuits while keeping their design intact, which is a crucial requirement for analog designers, in order for the locking technique to be widely adopted. For the proposed HT, it reveals the possibility of digital-to-analog HT attacks. The proposed PUF, called *neuron-PUF*, uses a single spiking neuron as the source of entropy. Its key characteristic is that it uses a single PUF cell and temporal redundancy to generate an arbitrarily long key, which results in significant low area and power overheads compared to mainstream PUFs, such as delay-based and memory-based PUFs.

RÉSUMÉ

Récemment, les coûts faramineux pour construire et entretenir une usine de fabrication de semi-conducteurs ont contraint de nombreuses entreprises à renoncer à avoir une usine de fabrication en propre. En externalisant la fabrication de circuits intégrés/propriété intellectuelle (CI/PI) à des sociétés tierces et souvent extraterritoriales, le procédé de fabrication a été confié à des sociétés potentiellement peu fiables. Il en résulte plusieurs menaces de sécurité pour l'industrie des semi-conducteurs, telles que la contrefaçon, la rétro-ingénierie et l'insertion de chevaux de Troie matériels (HT). Dans cette thèse, nous proposons une contre-mesure anti-piratage pour protéger les CI/PI analogiques et à signaux mixtes (AMS), une nouvelle attaque HT pour les CI/PI AMS et une nouvelle fonction physique non clonable (PUF).

La technique anti-piratage que nous proposons est basée sur le verrouillage des circuits analogiques configurables. Notre technique exploite le mécanisme de configuration intrinsèque du circuit pour y introduire une fonction verrouillage. Nous discutons de son implémentation et de ses capacités de résilience contre les attaques prévues, et nous la démontrons par une analyse en simulation et des mesures physiques d'un convertisseur analogique-numérique $\Sigma\Delta$ destiné à être utilisé dans des applications de récepteurs RF multistandard. L'attaque HT proposée pour les circuits analogiques exploite l'infrastructure de test. Le HT est introduit dans le sous-système numérique du système AMS et transfère sa charge utile au circuit analogique via le bus de test jusqu'à l'interface du circuit analogique à ce bus. Sa caractéristique principale est d'être invisible dans le domaine analogique. Le HT est démontré par deux études de cas. La première, en simulation, concerne un régulateur de tension (LDO) et la seconde, illustrée par des mesures en laboratoire, met en œuvre un récepteur RF configurable. Cette thèse montre l'importance de nouvelles contre-mesures de sécurité et de confiance adaptées aux CI analogiques. La technique de verrouillage proposée prouve la faisabilité de sécurisation d'un grand nombre de CI analogiques sans affecter leur conception, ce qui est une exigence majeure des concepteurs pour adopter cette technique. L'attaque par HT proposée révèle la possibilité d'attaques HT de système sur puce, ou SoC, AMS. La fonction PUF proposée utilise un neurone à impulsions comme source d'entropie. Sa caractéristique principale est de n'utiliser qu'une seule cellule PUF et une redondance temporelle pour générer une clé arbitrairement longue, ce qui permet de réduire considérablement les coûts additionnels en surface et en énergie par rapport aux fonctions PUF traditionnelles, telles que les PUF à retard et à mémoire.

PUBLICATIONS

As a result of this thesis the following publications have appeared:

[1] M. Elshamy, A. Sayed, M.-M. Louërat, A. Rhouni, H. Aboushady, and H.-G. Stratigopoulos, "Securing Programmable Analog ICs Against Piracy," in *Proc. Design, Automation and Test in Europe Conference*, 2020.

[2] M. Elshamy, G. Di Natale, A. Pavlidis, M.-M. Louërat, and H.-G. Stratigopoulos, "Hardware Trojan Attacks in Analog/Mixed-Signal ICs via the Test Access Mechanism," in *Proc. IEEE European Test Symposium*, 2020.

[3] J. Leonhard, M. Elshamy, M.-M. Louërat, and H.-G. Stratigopoulos, "Breaking analog biasing locking techniques via re-synthesis," in *26th Asia and South Pacific Design Automation Conference*, 2021.

[4] M. Elshamy and H.-G. Stratigopoulos, "Neuron-PUF: Physical Unclonable Function Basedon a Single Spiking Neuron," in *The 27th IEEE International Symposium on On-Line Testing and Robust System Design*, 2021.

ACKNOWLEDGMENTS

With one step far from finishing this chapter of my academic life, I have mixed feelings. While I am happy to finish my doctoral degree, I will miss these wonderful days, my amazing team, and my supportive supervisors. It was really a great privilege to work side by side with such brilliant minds.

First of all, I would like to express my sincere gratitude to my supervisor, my mentor, and my academic father Haralampos. Thanks, Haralampos for strengthening my academic personality. Our meetings and conversations were vital in inspiring me to think from multiple perspectives to form comprehensive and objective thoughts. Thanks for helping me with the apartment renting and understanding my personal circumstances. Thanks for letting me be part of this incredible scientific environment. I am also thankful to Marie-Minerve for her support and help. Thanks, Marie-Minerve for your precise review and your fast responses. Thanks for helping to get my "convention d'accueil".

I would like to say a special thank you to Hassan Aboushady. Thanks for letting us use your precious chips to prove our hypothesis. Thanks for your explanations and your constructive discussions. Further, I would like to thank Alhassan. Thank you for your time, support, and translations. It is truly a pleasure to know you, not just as a work mate but also a great friend.

I want to extend my thanks to my team, my academic brothers, and my lab-mates Julian, Antonios, Theofilos, Gabriel, Ilias, Alán, and Sara. It was my pleasure to spend those joyful days with you guys.

And I'd like to express my heartfelt gratitude to my family for their unwavering support and encouragement throughout this research, which is the culmination of three years of study abroad. For my parents, thank you is not enough to express how grateful I am to you. For my wife, thanks for all your support and for proof-reading my thesis, you have been amazing. For my daughter, sorry for being nervous than normal whilst I wrote this thesis! and I will now have some time to play with you as I promised!

From the bottom of my heart I would like to say to all of you guys

شكراً جزيلاً

Ευχαριστώ πολύ

Merci beaucoup

Tausend Dank

Thanks a lot

CONTENTS

I MAIN PART

1 INTRODUCTION 3
- 1.1 Vulnerabilities in the Semiconductor Supply Chain 3
  - 1.1.1 Expansion of the Supply Chain 3
  - 1.1.2 Problems Caused by Malice Actors in the Supply Chain 4
    - 1.1.2.1 Side-Channel Attacks 4
    - 1.1.2.2 Hardware Trojans Attacks 5
    - 1.1.2.3 Reverse Engineering Attacks 6
    - 1.1.2.4 Counterfeiting 7
  - 1.1.3 Vulnerabilities' Impact 7
- 1.2 Hardware Security and Trust 8
- 1.3 Countermeasure against Hardware Attacks 9
  - 1.3.1 Obfuscation 9
    - 1.3.1.1 Locking 10
    - 1.3.1.2 Camouflaging 10
    - 1.3.1.3 Split Manufacturing 10
  - 1.3.2 HTs Prevention and Detection 11
  - 1.3.3 Physical Unclonable Functions (PUFs) 11
- 1.4 The Growing Demand for Analog DfTr Techniques 12
- 1.5 Thesis Contributions 14
- 1.6 Thesis Structure 15

2 PRIOR ART 17
- 2.1 Prior Art on Analog Obfuscation 17
  - 2.1.1 Analog Locking 17
    - 2.1.1.1 Locking the Biasing Circuit 17
    - 2.1.1.2 Locking the Calibration Module 19
    - 2.1.1.3 Locking the Digital Section within a Mixed-Signal Circuit 20
    - 2.1.1.4 Locking via Emerging Technologies 21
    - 2.1.1.5 Compound Locking 22
  - 2.1.2 Analog Camouflaging 22
  - 2.1.3 Analog Split Manufacturing 23
- 2.2 Prior Art on Analog HTs 24
  - 2.2.1 Analog Circuits for Triggering HTs 24
  - 2.2.2 HT Attacks in the Analog Domain 24
  - 2.2.3 HT Defenses in the Analog Domain 25
    - 2.2.3.1 HT Prevention in the Analog Domain . . . 25
    - 2.2.3.2 HT Detection in the Analog Domain . . . 25

2.3 Prior Art on PUF . . . 26
2.4 Conclusion . . . 28
3 LOCKING BY UNTUNING: A LOCK-LESS APPROACH FOR ANALOG AND MIXED-SIGNAL ICS 29
3.1 Introduction . . . 29
3.2 Programming of Analog ICs . . . 30
3.3 Lock-Less Locking of AMS ICs . . . 33
3.3.1 Locking Principle . . . 33
3.3.2 Security Analysis . . . 37
3.3.2.1 Attacks in Digital Domain . . . 37
3.3.2.2 Removal Attacks . . . 37
3.3.2.3 Attacks on Biasing Locking . . . 37
3.3.2.4 Brute-Force and Multi-Objective Optimization Attacks . . . 37
3.3.2.5 Revealing the Calibration Algorithm . . . 38
3.4 Simulation Results . . . 39
3.4.1 RF Receiver Architecture and Programmability . . . 39
3.4.2 Calibration Algorithm . . . 40
3.4.3 Locking Results . . . 42
3.5 Experimental Results . . . 44
3.5.1 ΣΔ Modulator Architecture and Programmability . . . 44
3.5.2 Calibration Algorithm . . . 46
3.5.3 Locking Results . . . 47
3.6 Discussion on Resilience Against Foreseen Attacks . . . 52
3.6.1 Brute-Force and Multi-Objective Optimization Attacks 52
3.6.2 Revealing the Calibration Algorithm . . . 53
3.7 Conclusion . . . 54
4 DIGITAL-TO-ANALOG HARDWARE TROJAN ATTACKS 57
4.1 Introduction . . . 57
4.2 DfT for AMS and RF ICs . . . 58
4.3 Calibration of AMS and RF ICs . . . 59
4.4 Test Access and Control Mechanism in SoCs . . . 60
4.5 Proposed HT Attack . . . 62
4.5.1 Threat Model . . . 62
4.5.2 Attack Scenario . . . 63
4.5.3 HT Design . . . 64
4.5.4 Discussion on Countermeasures . . . 66
4.6 Case Study: LDO . . . 68
4.6.1 LDO Regulator Design . . . 68
4.6.2 DfT . . . 70
4.6.3 HT Payload Design . . . 73
4.7 Case Study: RF Receiver . . . 74
4.7.1 RF Receiver Programmable Architecture . . . 74
4.7.2 HT Payload Design . . . 76

4.8 Conclusion 78
5 NEURON-PUF: PHYSICAL UNCLONABLE FUNCTION BASED ON A SINGLE SPIKING NEURON 81
5.1 Introduction 81
5.2 Spiking Neurons 81
5.3 Neuron-PUF Architecture 83
5.4 PUF Quality Metrics 85
5.5 Results 87
5.6 Conclusion 89
6 CONCLUSION AND PERSPECTIVE 91
6.1 Conclusion 91
6.2 Contributions of the Thesis 91
6.3 Future Work and Perspective 93

II APPENDIX

BIBLIOGRAPHY 97

LIST OF FIGURES

Figure 2.1 Existing biasing locking methodologies for analog ICs. . . . 18
Figure 2.2 Existing locking via the calibration module for analog ICs. . . . 20
Figure 2.3 Locking the digital section within a mixed-signal circuits [55]. . . . 21
Figure 2.4 Biasing locking based on memristor crossbar [96]. 21
Figure 3.1 Configuration word loading and storage. . . . 32
Figure 3.2 Programmable current mirror. . . . 33
Figure 3.3 Programmable capacitor array. . . . 33
Figure 3.4 Lock-less locking via the programmability interface. 34
Figure 3.5 Key management schemes. . . . 35
Figure 3.6 Architecture of programmable multi-standard RF receiver. . . . 39
Figure 3.7 Architecture of tunable variable gain LNA. . . . 40
Figure 3.8 Architecture of tunable 2nd order BP RF ΣΔ modulator. . . . 40
Figure 3.9 The 2nd order ΣΔ modulator configured for calibrating the LC loop filter. . . . 41
Figure 3.10 SNR for correct key (green cross) and invalid keys (gray dots and red dot with index 7) computed at the output of the BP RF ΣΔ modulator. . . . 42
Figure 3.11 Transient output of BP RF ΣΔ modulator for the correct key (top) and the invalid key with index 7 in Fig. 3.10 (bottom). . . . 43
Figure 3.12 SNR for correct key (green cross) and invalid keys (gray dots and red dot with index 7) computed at the output of the RF receiver. . . . 44
Figure 3.13 PSD at the output of the BP RF ΣΔ modulator for the correct key (green) and the invalid key with index 7 in Figs. 3.10 and 3.12 (red). . . . 45
Figure 3.14 SNR versus input power with different LNA gain settings for the correct key (green) and the invalid key with index 7 in figures 3.10 and 3.12 (red). . . . 46
Figure 3.15 SFDR for the correct key (green) and the invalid key with index 7 in Figs. 3.10 and 3.12 (red). . . . 47
Figure 3.16 Architecture of ΣΔ modulator. . . . 48
Figure 3.17 The ΣΔ modulator configured for calibrating the LC loop filter. . . . 49

Figure 3.18 Test bench for measuring the performance of the modulator under the unlocking and the different locking scenarios. 50
Figure 3.19 SNR for correct (green cross) and invalid (blue dots) keys. Invalid key (orange star) has a relatively high SNR, and invalid key (red circle) has the min SNR. 51
Figure 3.20 PSD at the output of the BP RF ΣΔ modulator for the correct key (green) and the invalid key with max SNR (orange) and the invalid key with min SNR (red) in Fig. 3.19. 52
Figure 3.21 SNR versus input power for the correct key (green) and the invalid key with max SNR (orange) and the invalid key with min SNR (red) in Fig. 3.19. . 53
Figure 3.22 SFDR versus input power for the correct key (green) and the invalid key with max SNR (orange) and the invalid key with min SNR (red) in Fig. 3.19. . 54
Figure 3.23 IIP3 under unlock and lock operation. 55
Figure 4.1 Scan access including analog IPs (adapted from [174]). 60
Figure 4.2 HT scenario exploiting the SoC test infrastructure. 63
Figure 4.3 Payload mechanism based on transporting the malicious bit pattern to the victim analog IP. 65
Figure 4.4 Payload mechanism based on updating the TDR of the victim analog IP. 65
Figure 4.5 Payload mechanism based on requesting on-line testing or re-configuration and subsequently corrupting the transported data. 66
Figure 4.6 Block-level schematic of the LDO. 68
Figure 4.7 Schematic of the error amplifier within the LDO implemented with an OTA. 69
Figure 4.8 Schematic of SBGR generator. 69
Figure 4.9 Schematic of SOTA. 70
Figure 4.10 LDO output variation as a function of power supply variation. 70
Figure 4.11 LDO output variation as a function of temperature variation. 71
Figure 4.12 Transient response of the LDO for a variation of load current. 71
Figure 4.13 LDO with DfT. The added PD and PU transistors to enable topology modifications are shown in red color. 72
Figure 4.14 PSD under HT-free and HT-infected operation. . . 76

Figure 4.15 Dynamic range under HT-free and HT-infected operation. 77
Figure 4.16 SFDR under HT-free and HT-infected operation. . 78
Figure 4.17 IIP3 under HT-free and HT-infected operation. . . 79
Figure 5.1 The axon hillock circuit: (a) schematic; (b) transient response. 82
Figure 5.2 Neuron-PUF architecture. 83
Figure 5.3 Transient simulation of neuron-PUF showing relevant signals. 87

LIST OF TABLES

Table 5.1 Neuron-PUF quality metrics. 88

ACRONYMS

3PIP Third party IP
ACE adaptive channel estimation
ADC Analog-to-Digital Converter
AES Advanced Encryption Standard
AFGT Analog Floating-Gate Transistor
AMS Analog, Mixed-Signal
ATE Automatic Test Equipment
BEOL back-end-of-line
BIST Built-in Self-test
BP band-pass
CAD Computer-Aided Design
CE Consumer Electronics
CRP challenge-response pair
DAC Digital-to-Analog Converter
DfT Design-for-Test
DfTr design-for-trust
DNL Differential Non-Linearity
DR dynamic range
ECC error correcting code
EM Electromagnetic
EVM Error Vector Magnitude
FEOL front-end-of-line
FFT Fast Fourier Transform
FSM finite state machine

GA Genetic Algorithm
HD hamming distance
HDL Hardware Descriptive Language
HT hardware Trojan
IC Integrated Circuit
IDM integrated device manufacturer
IIP_3 input third-order intercept point
IM_3 third-order intermodulation
INL Integral Non-Linearity
IP Intellectual Property
LDO low-dropout
LFSR linear-feedback shift register
LNA low noise amplifier
LUT look-up table
MC Monte Carlo
NVM non-volatile memory
OFDM orthogonal frequency division multiplexing
OSR Oversampling Ratio
OTA operational transconductance amplifier
PA power amplifier
PD Pull-Down
PDK Process Design Kit
PLL Phase-Locked Loop
PRBS Pseudo-Random Bit Sequence
PSD power spectral density
PU Pull-Up
PUF Physical Unclonable Function
RE reverse engineering
RF Radio Frequency
RO Ring Oscillator
RSN reconfigurable scan network
RTL Register Transfer Level
SAT Satisfiability
SBGR sub-band gap reference voltage generator

SFDR spurious-free dynamic range
SFLL Stripped Functionality Logic Locking
SI scan in
SIB segment insertion bit
SMT Satisfiability Modulo Theory
SNN Spiking Neural Network
SNR Signal-to-Noise Ratio
SO scan out
SoC System-on-Chip
SOTA self-biased operational transconductance amplifier
SRAM Static Random-Access Memory
TAP test access port
TDR test data register
TPM Tamper-Proof Memory
TSMC Taiwan semiconductor manufacturing company
VGLNA Variable Gain Low Noise Amplifier
Vth various threshold voltage
WLAN wireless local area network

Part I

MAIN PART

1 INTRODUCTION

1.1 VULNERABILITIES IN THE SEMICONDUCTOR SUPPLY CHAIN

1.1.1 *Expansion of the Supply Chain*

Over the course of the past years, the business model of Integrated Circuits (ICs), which is a key component in Consumer Electronics (CE), has evolved. It shifts from vertical model to horizontal model motivated by the increasing cost of ICs fabrication. In the vertical model, all of the capacity required to construct a working IC and delivering it to the market, including knowledge, design software, manufacturing instruments, integrating capabilities, and testing equipment, is incorporated into the same entity. Firms that utilize the vertical model are known as integrated device manufacturers (IDMs). These IDMs such as Intel, Samsung, and SK Hynix have held the largest market share since the beginning, although this situation steadily changes over time [1], [2].

In the late 1980s and particularly in 1987, the foundation of Taiwan semiconductor manufacturing company (TSMC) paved the way for a new business model, namely the foundry model. The raising of such a model has contributed to the emergence of fabless companies, which are corporations or small startups focused on IC design but lack the ability to fabricate their own ICs. By outsourcing the IC fabrication, fabless companies have been able to reduce their capital requirements by at least tenfold [3]. The cooperation between foundries and fabless companies is referred to as the horizontal model. Utilizing the horizontal model not only reduces risks by eliminating the capital required to construct, maintain, and upgrade the fabrication facility, but it also provides other benefits such as reducing the time span from IC design to mass production. This operational efficiency is achieved by incorporating pre-approved modules into the IC design. All of these advantages have assisted the expansion of fabless companies, e.g., Qualcomm, Broadcom, Nvidia, and Apple, in terms of market share. Over the past ten years, the revenue of fabless companies has more than doubled [1].

Back at the procedure of reusing pre-approved modules, this sparked the creation of the System-on-Chip (SoC). SoC is the integration of conventional functions and special functions required for implementing a complex electronic system on a single silicon substrate. This, in turn, broadens the business model even further to include new models that are suitable for the new companies currently involved in the production of

SoCs. These new models can be categorized into different categories such as: (a) Computer-Aided Design (CAD) tool providers: firms that develop specialized software for designing and simulating electronic circuits; (b) Intellectual Property (IP) block providers: firms concerned with designing ICs that perform specific functions and assembling them into IP blocks; (c) Third party IP (3PIP) vendors: platforms used to identify and sell those IP blocks; (d) SoC integrators: specialized firms in constructing complex systems by incorporating several IP blocks into a single chip.

The continuous development of the business model has led to the expansion of the supply chain by introducing new companies. In more detail, the first step in the supply chain is the design stage, in which the chip is designed to perform a specific task. The chip design can be done internally, integrated from outsourced IPs, or a hybrid of the two. In terms of the business model, this stage combines the IP providers and the SoC integrators. Moving to the next stage, the chip is fabricated based on the design layout created in the previous step. This stage is called the fabrication stage and is performed in the foundry. The next actor in the supply chain is the packaging firm, which packages the chip then sends it to the testing facility. At the test facility, a production test bench is used to test the chips, then the chips that pass the test are sold in the market. Finally, the life cycle of a chip will reach an end at some point in time and it will be discarded.

1.1.2 *Problems Caused by Malice Actors in the Supply Chain*

The horizontal semiconductor business model has added fragility into the semiconductor supply chain. Despite its benefits, the outsourcing of IC fabrication and the distributed design flow involves multiple entities placed around the world and makes the semiconductor industry face several challenging security threats such as hardware Trojan (HT) insertion, reverse engineering (RE), and counterfeiting. Moreover, several side-channel attacks have been demonstrated with the aim of stealing sensitive data. The said threats' applicability and effect strongly depend on the supply chain phase where they are inserted. These threats will be explained in more detail the next subsections.

1.1.2.1 *Side-Channel Attacks*

Side-channel attacks aim at deducing sensitive data, i.e., secret key, from a chip or a system, through precise measurement and analysis of physical parameters, e.g., power consumption, processing time, or electromagnetic emission. The leaked information from indirect sources or channels relies on the intermediate values generated during the execution of a crypto-algorithm and are correlated with the applied inputs and the secret key of

the encryption [4]. An adversary can easily extract the encryption key by monitoring and analyzing the leaked information with substantially low-cost tools and in a short period of time. For example, an adversary, i.e., a malicious end-user, can apply an input while monitoring the processing time to get an output and then use the collected data, i.e., the applied inputs and the time taken for the computation, to reveal the secret key [5].

Another category of side-channel attack is based on fault injection (aka fault injection attacks). It aims to mitigate or weaken the implemented cryptographic systems in the chips by injecting malicious faults into the encryption module, which facilitates the leak of sensitive data. Unlike the other types of side-channel attacks that are considered passive attacks, fault injection is an active attack. For instance, an attacker can deploy an attack by applying a shorter clock pulse than normal to induce a clock glitch, making a rapid transient in the supply voltage to induce a power glitch, or exposing the under attack chip to Electromagnetic (EM) signals to disturb the operation of the chip.

These attacks pose a serious threat, especially to chips that integrate cryptographic modules and provide easy access to their physical parameters, e.g., smart cards. Side-channel attacks have proven effective in breaking powerful encryption techniques such as Advanced Encryption Standard (AES) and extracting the secret cipher [4], [6]–[9].

1.1.2.2 *Hardware Trojans Attacks*

A HT is a malicious modification of the design performed by an attacker within the IC supply chain that is intent to stay hidden and evade detection by the end-user who is the defender in this case. The HT is an undocumented functionality for the end-user and is designed in such a way that once activated it is capable of performing an undesired effect for the end-user. Any HT is in general composed of a trigger, i.e., activation mechanism, and a payload mechanism [10]. The HT may be always-on, in which case strictly speaking there is no trigger mechanism, it may be activated under rare conditions leaving a time bomb into the design, or it may have a well-timed activation controlled by the attacker. The payload mechanism refers to the HT effect on the chip's functionality.

The motivation for inserting a HT includes leaking sensitive information out of the chip, e.g., cipher keys, degrading the performance of the chip, or leading to complete malfunction, e.g. denial-of-service [11], [12]. A HT may be inserted by the CAD tool provider, i.e., by compromising the synthesis or verification scripts, by an IP design team, by a SoC integrator that can manipulate both the 3PIP cores and the test infrastructure comprising the test access and control mechanism and several embedded test instruments, and by a foundry that receives the GDSII file [13]–[17].

There is a multitude of HT designs proposed in the literature that range from simple to very complex attack modes. The simplest HTs are combinational circuits that monitor a set of nodes to generate a trigger on the simultaneous occurrence of rare node conditions and, subsequently, once the trigger is activated, the payload is simply flipping the value of another node. Another category of simple HTs is the sequential HTs which also have a condition-based activation, but they are triggered with a sequence of conditions and not with a specific state or condition like the combinational HTs. More complex HTs include silicon wearout mechanisms [18], hidden side-channels [19], changing dopant polarity in active areas of transistors [14], siphoning charge from victim wires [20], etc.

From the attacker's perspective, the goal is to achieve the desired effect via the use of a stealthy and minimum footprint HT such that it evades pre-silicon prevention and post-silicon detection methods applied by the defender.

1.1.2.3 *Reverse Engineering Attacks*

The term reverse engineering refers to the derivation of IC/IP proprietary information, i.e., architecture, netlist, layout, etc. It aims at reducing the attacker's technological disadvantage against the "author" of the IC/IP, gathering the necessary information for producing a similar or an identical IC/IP, e.g., a counterfeit, or locating the root-of-trust part of the IC/IP to steal secret information such as cipher keys. The attacker's target is to successfully reverse engineer the under attack IC to a specific abstraction level. This level can vary depending on the objective of the attacker, i.e., pirate the design, insert HTs, or assist counterfeiting attacks. For instance, the physical design level, the gate level, or the Register Transfer Level (RTL) could be the targeted abstraction level if the attacker wants to pirate the design. On the other hand, it is enough to abstract the gate level or the RTL if the attacker's target is to insert HT.

Nowadays, there exist equipment and software tools to successfully reverse engineer any unprotected IC/IP [21]. Reverse engineering involves the following steps: (a) de-packaging of the IC; (b) de-layering the individual layers of the IC using corrosive chemicals; (c) imaging the top-view of each layer using, for example, Scanning Electron Microscopy (SEM); (d) aligning and stitching the images of the different layers; and (e) extracting the netlist from the annotated images using dedicated software tools [22].

This attack can be initiated by any malicious actor in the supply chain, i.e., a SoC integrator, a test facility, a foundry, an end-user, or a recycling facility.

1.1.2.4 *Counterfeiting*

A counterfeit chip is a forgery or an unauthorized reproduction of an original chip. It includes cloning, recycling, overproducing, remarking, and out-of-specs [23]. A cloned counterfeit is an IC/IP that is being illegally cloned and sold as original. It may aim at copying the entire IC/IP or part from it. Cloning can be performed by an untrusted SoC integrator, foundry, or an adversary, e.g., end-user, via reverse-engineering. A recycled counterfeit is a used and possibly aged IC that is illegally recycled and resold as new. Overproduced ICs are ICs that are produced by an untrusted foundry beyond the number agreed in the contract and are illegitimately sold after in the market. Remarked ICs are ICs whose performances have been changed by a rogue test facility so that these ICs appear as higher-performing ICs. Out-of-spec ICs are ICs that have been proven through testing to be unreliable or have inadequate performance, but a rogue test facility can sell them in the market with forged documentation.

1.1.3 *Vulnerabilities' Impact*

The threats discussed in Section 1.1.2 have serious implications on the technological value chain (e.g. CAD tool providers, IC/IP providers, original equipment manufacturers, and users), on governments, and on the society as a whole [24]. This thesis focuses on the piracy and HT insertion threats, therefore herein we focus on the posed impacts of these threats.

Piracy has piqued great concern of the community, industry, and government because of its wide impact. It can emerge at any stage in the supply chain, resulting in a loss of company revenue or reputation, unreliability in the chip functionality, or catastrophic consequences, especially when used in critical applications such as healthcare and military applications. The broad spread of IC/IP piracy can be illustrated as an adversary's need to break down knowledge barriers with the IC/IP author to compete effectively in the market or get financial gain. For example, annual losses in the semiconductor industry due to counterfeit ICs/IPs are estimated to be $169 billion [25]. Despite being widespread, the practice of hardware piracy is highly dependent on the phase at which the threat is deployed in terms of complexity and cost. For example, while it is easy for a rogue foundry to clone a chip, cloning is quite a challenge for a rogue test facility. More specifically, the rogue foundry can benefit from the original blueprint of the chip to clone it, whereas the rogue test facility has to RE the chip to expose its design before cloning it, which dramatically increases the cost and the complexity of cloning. To that end, some threats may be considered less acute than others. But, it is

worth noting that the attack's complexity, and thus the cost, is gradually reduced due to the rapid development in the piracy capacities [21], [22], [26].

HT attacks are also a main preoccupation for society, industry, governments, and military since they pose severe risks with possibly disastrous outcomes. For this reason, HTs have received major attention in the scientific community throughout the last two decades [10], [11]. In the real world, HT hides effectively in the original design, evades testing, activates on a rare condition, as well as has various implications, hence it is hard to accurately gauge its impact. Though, its implications are obvious in different sectors. For example, the Syrian military sector exposed a HT being inserted in the radar system that caused the defense system to fail [27]. In 2012, a HT that allows an adversary to control a Boeing 787's navigation system was detected [28]. In 2018, HTs insertion into data servers resulted in leaking sensitive trade secrets and national security data from more than 30 companies in the United States as reported in [29]. However, the number of HTs reported thus far is significantly lower than the true number, and it is expected that more sophisticated HTs will be exposed in the near future [30].

1.2 HARDWARE SECURITY AND TRUST

Hardware security and trust is a topic that has attracted a lot of interest in recent years. It refers to protecting physical systems, i.e., ICs/IPs, by understanding the threats imposed by different actors through their lifecycle as a first step, then developing effective countermeasures against such threats.

One level of IP protection is the granting of patents. Patent guarantees the author's legal ownership of intellectual property rights and plays an important role in protecting intellectual property rights and copyrights. However, obtaining a patent takes a long time and seems to be useless in countries where intellectual property rights are not strictly applied. It is also not easy to determine and prosecute patent infringement. Usually, patents are important for intellectual property owners, but they can preferably be supplemented by additional levels of security.

Another level of protection is software-based, which can serve the hardware security through one of the following methods:

- *Language-based design security assessment* is an automated process aiming at checking the trustworthiness of an IC/IP design at the RTL. It is conducted while compiling the Hardware Descriptive Language (HDL). At the RTL level, it is critical to inspect the IC/IP design for potential security issues such as vulnerable logic and design flaws. These potential issues, if not detected, will then lead

to a multitude of threats, e.g., information leakage and HT insertion [31]–[35].

- *Cryptography* aims at developing suitable algorithmic protocols or techniques for preventing adversaries from accessing sensitive information. It is vital for practicing *confidentiality*, i.e., defining allowed parties from accessing sensitive information, *authenticity*, i.e., specifying trustworthy individuals, and *integrity*, i.e., checking message's contents from being altered during the transmission phase. To that end, different cryptographic algorithms such as symmetric cryptography, asymmetric cryptography, AES, Blowfish cipher, etc., can be used [36]–[38].

However, software-based protections suffer from scalability issues and their defense against different threats is questionable. Furthermore, cryptographic techniques depend on a hardware root of trust, which increases the required area overhead [24], [39].

Finally, hardware-based security includes hardware design-for-security and design-for-trust (DfTr). Hardware design-for-security aims at designing hardware techniques to secure sensitive data in hardware, while hardware DfTr aims at dealing with hardware threats, i.e., counterfeiting and HTs [24], [40], [41]. Hardware-based security offers protection by leveraging fabrication non-idealities to produce unique chip's signature, injecting noise to make all the performed operations even in terms of power consumption and processing time, modifying the design to obfuscate it, or inserting additional components to protect the design, e.g., locking. It provides effective tailored countermeasures for each threat while taking into account the IC/IP design and its targeted application. These countermeasures are discussed next.

1.3 COUNTERMEASURE AGAINST HARDWARE ATTACKS

To be consistent with the objective of this thesis, below we focus on anti-piracy countermeasures, i.e., locking, camouflaging, and split-manufacturing, Physical Unclonable Functions (PUFs), and HTs detection and prevention.

1.3.1 *Obfuscation*

Obfuscation is the process of concealing the functionality or layout of a design by incorporating a locking mechanism, camouflaging the design geometry, or splitting the layout fabrication process. Obfuscation techniques that aim at hiding the design's functionality are known as *Locking*, while they are referred to as *Camouflaging* if they are used to hide the design's layout, and finally *Split manufacturing* aims at dividing

the fabrication process between trusted and untrusted foundries. These techniques are discussed in the following subsections.

1.3.1.1 *Locking*

Locking is an end-to-end protection mechanism. It consists of inserting a lock into the design such that unless the valid key is used the functionality breaks [25], [42]–[44]. The key is kept as the IC/IP design house secret. Locking thwarts cloning by the SoC integrator or foundry and overbuilding by the foundry as the blueprint of the IP/IC is useless without knowing the key. It also thwarts cloning via reverse-engineering as the key is stored in a Tamper-Proof Memory (TPM) that cannot be read. It thwarts remarking for digital ICs since structural testing can be equivalently performed on a locked chip using any invalid key and, thereby, chips can be unlocked after testing [45]. Protection against remarking can be achieved for any IC type by remotely activating the chips during testing using asymmetric cryptography [25]. Finally, it thwarts recycling as long as the key is reloaded every time the IC is powered-up. This requires a different key management scheme that makes use of a public user-key and a chip identification-key generated, for example, by an on-chip PUF [46]. The user-key and chip identification-key are XORed to produce the secret key.

1.3.1.2 *Camouflaging*

Camouflaging addresses the RE threat, and hence counterfeiting threats that rely on RE, by making stealthy alterations in the design using mechanisms at the device and interconnect level, resulting in an extracted netlist that is deceiving for the attacker. These alterations have to defend against RE without affecting the chip's performance or corrupting its outputs. For example, the designer camouflages the developed design by using standard cells that can perform one of several possible functions, whereas he controls the positions of dummy contacts and real contacts to set the cell's actual function. When an adversary attempts to delayer the chip in order to extract its netlist, it is nearly impossible for him to distinguish between dummy contacts and real contacts. As a result of the inability to detect cell functionality, the adversary extracts a non-functional netlist [47], [48].

1.3.1.3 *Split Manufacturing*

Split manufacturing protects only against an untrusted foundry by manufacturing only the lowest layer of the target design at the untrusted foundry and the remaining part at a trusted low-end foundry, i.e., the upper layer. The untrusted foundry is refereed to as front-end-of-

line (FEOL) foundry, while the trusted foundry is refereed to as back-end-of-line (BEOL) foundry. As the FEOL foundry receives just a portion from the layout, that usually contains nothing more than the individual transistors and resistors, then it lacks the required data to successfully reverse engineer the layout.

1.3.2 *HTs Prevention and Detection*

Countermeasures against HT attacks include pre-silicon and post-silicon methods. Pre-silicon methods are used to prevent HTs insertion or make their detection easier, whereas post-silicon methods are used to expose HTs that have already been inserted.

Pre-silicon prevention methods include: (a) functional verification of 3PIP cores [49]; (b) structural analysis of HDL codes [49]; (c) targeted automatic test pattern generation algorithms [50] or simulating the circuit using specific test benches, i.e., performing aging simulations along with over-clocking [51]; (d) searching for unused components during design-time verification and removing them as potentially suspicious [52]; (e) filling in all unused spaces on the layout, which are most likely insertion areas for HTs, with functional filler cells and checking if those have changed [53]; and (f) design obfuscation, for example using locking [54]–[56], camouflaging [47], [57], or split manufacturing [58], aiming at obscuring the IC functionality so as to make it difficult for the attacker to insert the HT.

Post-silicon detection methods include: (a) destructive reverse engineering, which involves de-packaging and de-layering the chip, imaging the chip's layers, and using software to stitch together the prepared images, thereby recovering the layout and netlist, which thereafter can be carefully examined to detect the presence of HTs [22], [59]; (b) optical circuit analysis aiming at measuring optical emissions of the IC and comparing them with a trusted emission image of a "golden" IC [60]; (c) functional testing aiming at exposing the HT by applying test vectors similar to pre-silicon prevention methods [50]; (d) statistical side-channel fingerprinting aiming at exposing the HT by its effect on parametric measurements, i.e., delay, power, temperature, etc. [17], [61]; and (e) using run-time monitors, i.e., current sensors [62] and thermal sensors [63].

1.3.3 *Physical Unclonable Functions (PUFs)*

A PUF is a circuit that leverages statistical manufacturing variations of circuit parameters to generate a chip-unique signature. When queried with an input, referred to as *challenge*, it generates an output, referred to as *response*, that typically is a bitstring composing a digital key. PUF applications include among others device authentication, secret key gen-

eration, hardware anti-piracy, security in Internet-of-Things (IoT) devices, etc. In device authentication, the PUF is used as a silicon biometric to generate a unique fingerprint or ID per chip [64], [65]. In secret key generation, a PUF is used to generate the key on-the-fly at power-on, thus avoiding explicit key storage [64]. In hardware anti-piracy, the PUF can be used to provide each chip an ID such that it can be traced along its lifetime for anti-counterfeiting purposes [23], [66]. In addition, a PUF can be used in the key management scheme of chip locking techniques [25], [55], [56]. In a resource-constrained smart IoT edge device, a PUF can be used for lightweight low-cost authentication protocols [67], [68].

The PUF concept was originally introduced in [69]–[71] and several silicon PUF implementations have been proposed since then. The most popular PUFs are delay-based PUFs and memory-based PUFs. Delay-based PUFs exploit some race condition that is built-up inside the circuit, while memory-based PUFs exploit the natural tendency of a memory cell towards one of its two states as it will be discussed in more detail in Chapter 2.

No PUF is inherently robust and a percentage of PUF cells may generate unstable bits that should be handled accordingly. Stability boosting techniques include temporal majority voting to stabilize noisy bits, burn-in hardening to accelerate aging, masking of “dark" bits that are unstable across varying operating conditions, and error correcting codes (ECCs) circuits [72]–[76]. For this reason, PUFs typically generate an excess number of bits which can be thereafter down-sampled to a fully stable key.

Another categorization of PUFs takes into consideration the number of challenge-response pairs (CRPs) that the PUF can support. A PUF that can support only a small number of CRPs is called a *weak* PUF, while a PUF that can support a very large number of CRPs that cannot be tried out in a reasonable time frame is called a *strong* PUF.

1.4 THE GROWING DEMAND FOR ANALOG DFTR TECHNIQUES

While hardware security and trust aspects have been extensively studied for digital circuits, the space of vulnerabilities and solutions for analog circuits is largely unexplored and little understood as of today [77], [78]. The great dearth of techniques for Analog, Mixed-Signal (AMS) and Radio Frequency (RF) IC/IP countermeasures, the reasons for their lagging behind digital countermeasures, and the importance of broadening the development of such countermeasures can be summarized as follows:

- *Analog applications are everywhere*: Since all elements in nature are analog, e.g., light, sound, and EM signals, every chip must at least have an analog interface for interacting with its environment. Analog circuits act as a link between the analog and the digital regimes. For instance, an AMS interface is needed for receiving the signals

and preparing them for any additional digital operations in each communication standard receiver. From a system-level perspective, analog circuits are perhaps nowadays the weakest link in achieving the global security and trust requirements. As a result, the demand for securing analog circuits has been increased.

- *Design difficulties*: Analog IC design is complex because it has to deal with process variation effects on the chip performance, as well as be robust against fluctuations in temperature, noise, etc. On the other hand, achieving such reliability makes analog ICs very sensitive to design changes or the addition of new components. From another aspect, and in contrast to digital design, there are no mature and reliable enough automated means for designing analog circuits from a high-level design specification, nor automated synthesis software to synthesize these circuits until today. Due to the design complexity and sensitivity besides the lack of automated aid, analog designers are rather reluctant to change the design flows for adding features into the design unrelated to functionality, i.e., design-for-test, design-for-security, etc.

- *Analog chips are valuable*: The value of an analog chip is extremely high due to the complexity and challenges that its design presents. Even for a highly experienced and knowledgeable designer, designing a functionally reliable IC is a challenge because he normally needs to go through many design cycles and make a lot of modifications in the design to achieve his goal. As their values rise, so does the need to protect them.

- *Lag of technology developments*: In general, developments around analog circuits (i.e. CAD tools, migration of designs into more advance technology nodes, methodologies for design-for-test, design-for-yield, design-for-reliability, etc.) lag behind those for their digital counterparts. Hardware security is not an exception. Although solutions in the digital domain have been explored for over a decade now, the solution space for analog has only started being explored in the last 4 years or so, that is, with a delay of almost a decade.

- *Vulnerability to piracy*: According to collected data on counterfeit incidents, among all component types (i.e. analog, microprocessor, memory, programmable logic, etc.), about 25% of reported incidents concern analog ICs [23]. This is understandable considering the high value of analog chips and their vulnerability to IC/IP piracy. This vulnerability is due to the relatively low number of transistors in analog ICs and their association with distinct layout patterns, which facilitates rogue adversary to RE and counterfeit these valuable designs.

1.5 THESIS CONTRIBUTIONS

The thesis makes the following contributions:

- We propose an anti-piracy security approach for the class of highly-programmable analog ICs [79]. The security approach relies on functionality locking by leveraging the inherent programmability and utilizing the configuration settings as secret keys or, equivalently, the programming bits as key-bits. When invalid keys are applied, the circuit is untuned and, as a result, its functionality breaks, i.e., at least one of the performances violates its specification. As long as the calibration algorithm that produces the configuration settings can be concealed, the proposed approach can serve as a countermeasure against all types of counterfeiting, i.e., cloning, overbuilding, remarking, and recycling. An important advantage of the proposed approach is that it is lock-less. It leaves the design intact, there is no change to the design flow, and there are no performance penalties and no area or power overheads due to the lock operation. We demonstrate it on a ΣΔ Analog-to-Digital Converter (ADC) with high programmability and complex calibration algorithm used in the context of highly-digitized, multi-standard RF receivers.

- We propose a HT attack for analog circuits with its key characteristic being that it cannot be prevented or detected in the analog domain [80]. The HT attack works in the context of Systems-on-Chip comprising both digital and analog IP blocks. The attacker could be either the SoC integrator or the foundry. More specifically, the HT trigger is placed inside a dense digital IP block where it can be effectively hidden, whereas the HT payload is in the form of a digital pattern transported via the test bus or generated within the test bus, reaching the Design-for-Test (DfT) or programmability interface of the victim analog IP with the test bus. The HT payload unexpectedly activates the DfT and sets the victim analog IP into some possibly partial and undocumented test mode or changes the nominal programmability. The HT payload can be designed to result in performance degradation or complete malfunction, i.e., denial of service. We demonstrate this HT attack scenario on two analog IPs, namely a low-dropout (LDO) regulator using simulation and an RF receiver using hardware measurements.

- We propose a novel PUF concept based on a single spiking neuron [81]. The inherent variability of the neuron results in a chip-unique analog spiking pattern that is digitized to produce a chip-unique digital key. A stability booster is also employed based on self-masking to obtain a fully stable digital key. The PUF is area and

power effective since a single PUF cell is used to produce an arbitrarily sized digital key. We demonstrate PUF quality metrics close to ideal values and argue that the PUF is resilient against various physical attacks. Our results from this preliminary work provide a starting point to address this new concept.

1.6 THESIS STRUCTURE

The thesis is structured as follows: in Chapter 2, we review the prior art in analog hardware security related to piracy and HTs, as well as the prior art on PUFs. In Chapter 3, we present the proposed locking technique for the class of highly-programmable analog ICs. In Chapter 4, we present the HT attack for analog IPs in the context of SoCs. In Chapter 5, we present the design of the neuron-PUF. In Chapter 6, we conclude pointing to future work and perspective in the field of analog hardware security.

2

PRIOR ART

2.1 PRIOR ART ON ANALOG OBFUSCATION

2.1.1 *Analog Locking*

As discussed in Chapter 1, locking has piqued the attention of the research community since it can defend against any rogue adversary positioned anywhere in the IC supply chain. The locked, i.e., functionally obfuscated, netlist moves through different stages in the supply chain, that potentially include malicious parties, without the secret key. This guarantees (a) the uselessness of reverse-engineering the design's netlist, and (b) the functionality failing of the counterfeited ICs, e.g., cloned ICs, overproduced ICs, etc. The activation of locked IC is done by loading the activation key into a TPM within the IC [82]–[84].

In contrast to digital designs, it is not favorable to modify analog designs by inserting locking mechanisms such as logic gates and look-up tables (LUTs), as this usually affects the IC's performance. Consequently, this increases the challenges to develop locking techniques for analog ICs.

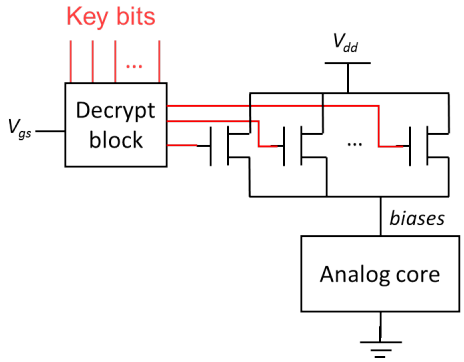
The prior work on analog hardware security tailored for locking analog circuits can be categorized as follows:

- Locking the biasing circuit
- Locking the calibration module
- Locking the digital section within a mixed-signal circuit
- Locking via emerging technologies
- Compound locking

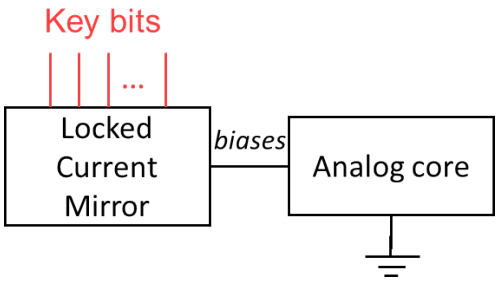
In the sections that follow, we use these categories to present and discuss previous work on locking analog circuits.

2.1.1.1 *Locking the Biasing Circuit*

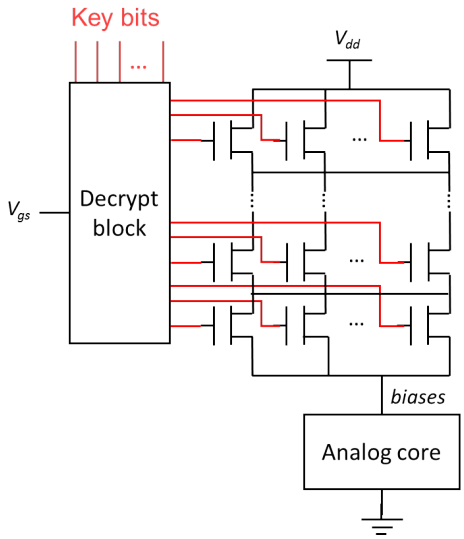
Analog circuit performance is very sensitive to changes in biasing conditions, i.e., currents and voltages. In more detail, the analog circuits are designed through two steps: performance design and biasing design. In the first step, the circuit is designed to perform a specific function. During this step, the biasing conditions for the transistors are set to achieve the



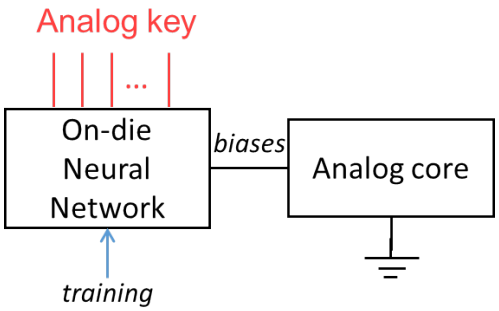
(a) Biasing locking based on transistor width obfuscation [85].



(b) Biasing locking based on current mirror locking [86].



(c) Biasing locking using mesh-based transistor obfuscation [87].



(d) Locking through neural network-based biasing [88].

Figure 2.1: Existing biasing locking methodologies for analog ICs.

targeted performance. The biasing circuit is then designed to keep the desired operating points for these transistors at a certain level. This is critical for analog circuit functionality and reliability. Another important aspect of the biasing circuit is that it is not part of the analog circuit's core, but rather is connected to it. Based on that, biasing circuits can provide an efficient way for locking analog circuits.

Biasing locking re-designs the biasing circuit so that the secret key can be used to control its parameters and thus its output. Applying a wrong key leads to non-negligible fluctuations in the biasing conditions, which force the circuit to violate one or more of its performances. Using the secret key, on the other hand, causes the circuit to behave as expected. Biasing locking must be designed to have a large enough key size to prevent brute-force attacks while also ensuring that no incorrect keys can drive the analog circuit into or close to its normal performance.

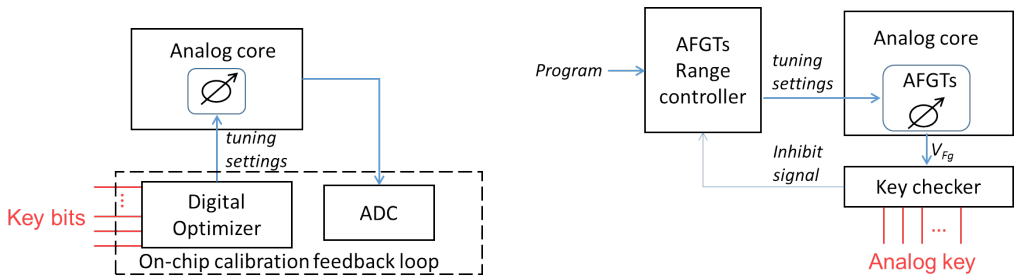
In [85], a transistor that sets the current or voltage bias in a node is obfuscated by replacing it with parallel-connected transistors whose gates are controlled by key-bits as illustrated in Fig. 2.1(a). The key-bits activate transistors whose aggregate width equals the nominal width of the original transistor. However, the proposed technique does not guarantee that only one key can activate the biasing circuit or that the wrong keys will degrade the performance well enough. In [86], it is shown how to redesign a current mirror so as to insert key-bits as illustrated in Fig. 2.1(b). Extra mirroring branches are inserted, where each branch is comprised of the mirroring transistor and possibly several switches that are controlled by the key-bits. The resultant current bias will depend on which branches are switched-on, as well as on the geometry of the mirroring transistor in these branches. They used a Satisfiability Modulo Theory (SMT) solver to overcome the issues in the previous technique. The solver is used to search for the appropriate size for each mirroring transistor so that only the correct key-bits make the circuit to perform well while other keys cannot. In [87], a mesh-based obfuscation of biasing transistors is proposed as illustrated in Fig. 2.1(c). Each transistor in the mesh is of a different size and is controlled by a key-bit. The valid key sets the effective transistor length and width to generate the correct bias. In [88], it is proposed to add on-chip a neural network that is trained to map the secret analog key, which is in the form of analog DC voltages presented as inputs to the neural network, to the correct bias as illustrated in Fig. 2.1(d). For invalid keys, the neural network is trained to give the same erroneous bias.

Biasing locking is a generic methodology applicable to any analog IC and offers an elegant way for inserting a digital key-enabled lock into an analog IC. Although the lock is not inserted into the analog core, biasing circuits are fundamental units for proper operation of analog ICs and their design should be carefully done to meet requirements such as biasing accuracy, temperature stability, bandwidth, input/output compliance voltage, input/output resistance, etc. The above works have not considered the effect of locking on the performance of the biasing circuit.

On the other hand, counter-attacks were proposed recently that break biasing locking allowing the attacker to recover the secret key or to remove the lock by re-synthesizing the biasing circuit [89]–[92].

2.1.1.2 *Locking the Calibration Module*

Another category of locking methodologies considers inserting the lock into the on-chip calibration mechanism. In this case, locking acts on the tuning knobs that compensate for process variations and non-idealities. In [93], a calibration loop is considered that uses an ADC to digitize the output of the circuit, followed by a digital optimizer that maps the output



(a) Logic locking of digital optimizer in on-chip calibration loop [93].

(b) Locking of the programming range of AFGTs [94].

Figure 2.2: Existing locking via the calibration module for analog ICs.

to appropriate tuning knobs that improve the performance trade-off of the circuit. It is proposed to insert the lock into the digital optimizer using logic locking as shown in Fig. 2.2(a). The used logic locking technique is known as Stripped Functionality Logic Locking (SFLL) [44]. Unless the valid key is provided, the tuning operation is affected.

Although no modifications are needed in the analog part of the IC, which is important for the analog designer, the proposed technique is vulnerable to removal attacks as the locked optimizer module is not part of the circuit's core and can be easily replaced with a non-locked optimizer. Moreover, SFLL can only protect a portion of the feasible input patterns, necessitating a diligence selection of the secured pattern and tuning knobs to ensure the protection of the chip. Logic locking, also, results in some justifiable yet non-negligible areas and power overheads. Finally, the system protection is determined by the strength of the employed logic locking technique, which means that breaking the employed logic locking technique renders the analog circuits unprotected.

In [94], a calibration scheme is considered enabled by Analog Floating-Gate Transistors (AFGTs) as shown in Fig. 2.2(b). A locking principle is proposed where the lock controls the programmability range of the AFGTs. The full tuning range is inhibited unless the AFGTs are first programmed in a certain order and with certain voltages, which are termed waypoints and constitute the secret analog key. This condition is validated by a key checker block. The limitations of this approach are that AFGTs are not standard tuning knobs to enable programmability and that the lock mechanism can be straightforwardly removed by the attacker, thus this approach offers resilience only against overbuilding.

2.1.1.3 *Locking the Digital Section within a Mixed-Signal Circuit*

The third locking methodology, shown in Fig. 2.3, leverages logic locking of digital sections within a mixed-signal circuit to gain control over the signal-processing flow [55]. They suggested that, by employing a state-of-

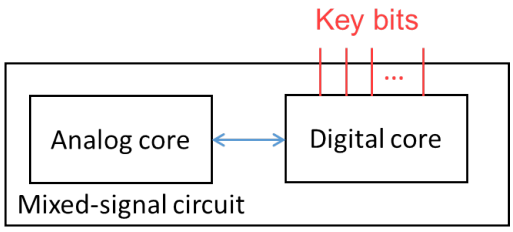


Figure 2.3: Locking the digital section within a mixed-signal circuits [55].

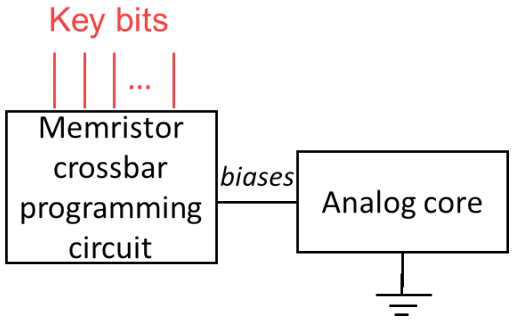


Figure 2.4: Biasing locking based on memristor crossbar [96].

the-art logic locking technique, e.g., SFLL, it is feasible to lock the entire system functionality. Using an invalid key, the output of the locked digital sections will be corrupted, which will in turn corrupt the mixed-signal IC performance trade-off in a complex and unpredictable way.

In [95], the authors presented a real showcase of their work in the context of audio applications. They used the technique that they proposed in [55] to lock an audio input interface by locking the underlying decimation filter in the ΣΔ ADC, and then they showed the effect of using the incorrect key as well as the correct key on the recorded audio.

Similar to [93], logic locking will result in some justifiable yet non-negligible area and power overheads and the system protection is determined by the strength of the employed logic locking. Removing and redesigning the entire digital part is not an easy task and is beyond the assumed capabilities of the attacker.

2.1.1.4 *Locking via Emerging Technologies*

The use of emerging technologies such as memristors is another feasible way to lock analog circuits. In [96], a key-enabled biasing circuit is proposed based on two memristor crossbars as shown in Fig. 2.4. Only the valid key programs correctly the memristor crossbars to generate the correct biases, while by applying an invalid key the memristor breakdown voltage can be reached, thus limiting the number of trials an attacker can attempt. However, in order to determine the secret code that will correctly activate the biasing circuit, each memristor must be adjusted

after the fabrication process. Furthermore, as it is an emerging technology, memristor integration with existing silicon technologies is still not mature enough.

2.1.1.5 *Compound Locking*

Finally, a compound locking methodology can be considered. For example, in [97], it is proposed to lock the analog core with biasing locking and the digital core with logic locking creating shared key dependencies so as to stop an attacker from breaking the analog and digital cores' locking mechanism independently.

2.1.2 *Analog Camouflaging*

Existing physical design obfuscation techniques for AMS and RF circuits are based on obfuscating the threshold voltage of transistors [98] and camouflaging the layout geometry of analog components [47].

In [98], it is proposed to design the analog circuit with a mix of transistors whose various threshold voltage (Vth) is one of many, i.e., high, normal, and low, to protect the analog circuit netlist from RE attacks. Since the operating point of the transistors depends on Vth, changing Vth drives the circuit to behave differently. By wisely dividing the used transistors into groups each of which uses a different Vth, it can help the protection of the analog circuit netlist. The applicability of this technique is feasible since the Process Design Kits (PDKs) typically include more than one Vth for the same transistor. On the other hand, identifying the Vth of each transistor during the RE process is extremely difficult. However, this technique requires a redesign of the circuit, which is not appealing for analog designers.

In [47], it is proposed to camouflage the effective geometry of the design layout by using a mix of real and fake contacts, which is inspired by gate camouflaging in the digital regime. This concept is used to build a library of "obfuscated" geometry components such as transistors, resistors, and capacitors. For instance, a portion of the gate fingers of the transistors is disabled, while some units in the resistors and the capacitors are disconnected. As a result, the actual sizing of the components, i.e. the effective width of the transistors and the values of the resistors and the capacitors, is obfuscated. From the attacker's perspective, these fake contacts appear to be electrically connected while in fact they are disconnected. As a result, the reverse-engineered netlist contains components that are wrongly sized. Furthermore, they proposed a methodology for effectively camouflaging analog designs, i.e., the selection criteria for the components to be obfuscated so as to reduce the needed number of camouflaged components and the position of the selected components

so as to reduce the effect of the camouflaged components on parasitics, circuit's performances, and area overhead. They also argued that the proposed technique can be used for camouflaging an already designed circuit or for introducing camouflaging into the design process. However, the proposed technique uses fake contacts, which requires very careful handling during the fabrication process as these contacts may be accidentally connected, resulting in a yield reduction. Also, the space needed to electrically disconnect these contacts may be too small, increasing the system parasitic, or it may be too large, facilitating the attacker to detect the fake contact.

2.1.3 *Analog Split Manufacturing*

Split manufacturing of integrated circuits proposes a method for lowering the required cost of having a fabrication plant as well as reducing the rising security concerns associated with outsourcing IC fabrication. It divides a design's fabrication process so that the malicious foundry fabricates the high end (aka FEOL layer), i.e., transistors and lower metal layers, and the design house fabricates the low end (aka BEOL layer), i.e., higher metal layers. The principle of split manufacturing of analog ICs is demonstrated in [99]. It is argued that an untrusted foundry not knowing coil and capacitor sizings, grants a higher level of security of split manufacturing for RF than for digital designs. In particular, the BEOL layer in analog circuits not only contains the metal interconnections as in the digital designs but also contains capacitors and inductors sizing. Since the analog designs contain a small number of transistors, it is more critical to hide capacitors and inductors sizing than to conceal the metal interconnections. As a result, it is important to remove the top metal layers down to the last layer containing any information regarding the size of capacitors and inductors. In TSMC 0.18 μm technology, for example, the BEOL layer should include the top two metal layers, since they contain capacitors and inductors, whereas the FEOL layer contains the remaining layers, i.e., four metal layers and transistors. On the other hand, known attacks on digital split manufacturing, such as the proximity and recognition attacks [99], [100], do not apply to analog split manufacturing. The authors also suggested using non-functional rings and creating empty zones to obfuscate the exclusion of metal layers beneath inductors and capacitors, as the attacker may be able to predict the size and position of such devices if obfuscation is not used.

2.2 PRIOR ART ON ANALOG HTS

2.2.1 *Analog Circuits for Triggering HTs*

Among different methods for triggering HTs, e.g., no trigger, rare-event trigger, or sequential rare-events trigger, the sequentially triggered HTs are the most difficult to detect. But, it requires the use of finite state machine (FSM) or counters to detect those rare events, which increases the possibility of HT detection. However, HTs can benefit from a tiny analog switch-capacitor circuit to detect a sequence of rare events as proposed in [20]. The authors proposed the usage of a simple analog circuit they called "A2" to detect high frequency toggling events. A2 can be inserted into already routed circuits to monitor a specific pulse signal. During normal operation, the HT remains silent as the monitored signal is toggled with low frequency, causing leakage currents to discharge the capacitor. When the targeted signal is toggled at a high frequency for a certain time window, it charges the capacitor above a certain threshold, which in turn triggers the payload. In [101], the authors suggested different design configurations based on the previous concept, i.e., using analog switch-capacitor circuits for sequentially triggered HTs.

2.2.2 *HT Attacks in the Analog Domain*

In [102]–[105], HT attacks are demonstrated for wireless ICs aiming at leaking secret information within a legitimate signal transmission. The attacker leverages the data transmission capability of the HT-infected device to establish a covert side-channel, without the need to gain physical access to the device. For example, the HT could forward bit-by-bit the content of the cipher key register of the crypto-core to the analog transmitter. In [102], [103], [105], the idea is to exploit the margins that exist between the operating point of the transmitter and the boundaries defined by the transmitter and communication standard specifications. In particular, the HT performs minute modifications in the parameters of the transmitted signal, such as amplitude and frequency, to leak sensitive information from the tampered device. Two HT payload mechanisms are shown in [105], one that uses a single pole double throw switch and a pair of resistors to alter the input termination impedance of the power amplifier, and another one that reprograms the gain stages. In [104], it is proposed to use spread spectrum techniques to hide an unauthorized transmission signal within the legitimate signal below the noise level. In both cases, the IC passes all conventional specification tests and the transmission signal still obeys the transmission specifications and is within the margins allowed because of process variations. Therefore, the inconspicuous receiver cannot interpret the minute change in the

transmitted signal as malicious. However, the attacker knowing the HT payload mechanism can listen to the channel and recover the key.

Another interesting direction for HT design is to exploit the fact that an analog IC may have undesired states or operating modes. In this case, the HT attack consists of bringing the analog IC into one of these states to cause undesirable operation. This HT type has been demonstrated for a multitude of basic analog circuits, such as current mirrors, filters, oscillators, bandgap reference sources, and operational amplifiers [16], [106]–[109].

2.2.3 *HT Defenses in the Analog Domain*

In Section 1.3.2, we discussed some general HT prevention and detection techniques. Herein, we will focus specifically on HT prevention and detection techniques in the analog domain.

2.2.3.1 *HT Prevention in the Analog Domain*

To inject a HT, the attacker must first understand the circuit in order to construct a stealthy HT by setting its trigger, payload, and location so that it can hide easily in the original circuit and avoid detection by the defender. Obfuscation countermeasures such as locking and camouflaging can be used to prevent the attacker from understanding the circuit's functionality or design and thus preventing him from inserting stealthy HTs or facilitating their detection.

2.2.3.2 *HT Detection in the Analog Domain*

HT detection can be done before the fabrication, i.e., pre-silicon, so as to facilitate removing any potential HTs from the design before being fabricated or after the fabrication, i.e., post-silicon, so as to detect the presence of HTs.

1. *Pre-silicon detection techniques*
   In [110], a methodology is proposed for detecting potential wires that can be used to carry HT's triggering signal based on a control value concept. Based on this methodology, they created a tool called "FANCI", which uses Boolean functional analysis to detect such wires. In [111], they argued that information flow tracking, a well-known methodology used for ensuring data confidentiality in electronic circuits, could be used to protect sensitive data from the risk of leakage not only in digital circuits but also in analog and mixed-signal circuits.

2. *Post-silicon detection techniques*
   In [104], the authors proposed a technique for exposing HTs injected

in the analog/RF section of the wireless local area network (WLAN). They used the adaptive channel estimation (ACE) technique, a technique that is based on leveraging the channel estimation capabilities of orthogonal frequency division multiplexing (OFDM) systems, to effectively separate potential HT activity from the legitimate signal as well as the noise and thus detect the presence of HT. The effectiveness of their technique was evaluated by detecting minute changes in the transmitted signal power properties caused by a HT designed to leak sensitive data and was injected in the transmitter's power amplifier (PA). In [105], the authors extended their work by demonstrating the efficacy of the ACE technique in defending against the threat posed by amplitude-modulating analog/RF hardware Trojans on WLAN systems. In [103], it has been demonstrated that statistical side-channel fingerprinting, which is based on trained one-class classifiers, is capable of detecting HTs that carefully conceal leaked sensitive data within the allowed transmission specification margins for process variations. Another method is to perform careful analysis of the transmitted signal spectrum to detect such HTs [104]. In [61], the authors demonstrated that HTs can be detected using side-channel crypt-analysis techniques. They proposed a methodology for creating golden fingerprints for the targeted chip using side-channel information such as power, temperature, and EM radiations. These golden fingerprints are then used to detect potential HTs. In [112], the authors proposed a medium-cost visual technique, i.e., optical microscopic picture, to detect the existence of HTs. They argued that comparing the GDSII layout of the targeted chip with an optical image of the chip's top metal layer is sufficient to detect any potential HTs.
A run-time HT detection technique tailored for analog HTs, such as the one proposed in [20], is proposed [113]. The underlying principle of this technique is to monitor a set of specific signals, and upon abnormal toggling events, it initiates a hardware interrupt request.

2.3 PRIOR ART ON PUF

There are a multitude of PUF designs in the literature that can be categorized based on their operating principle, the number of CRPs that the PUF can support, etc. Because of the enormous number of publications relating to PUF, below we briefly discuss the underlying concept for PUFs based on their operating principle,

- *Delay-based PUFs* exploit some race condition that is built-up inside the circuit. For example, the arbiter PUF uses two delay paths with an identical layout built by a serial connection of k multiplexers and a latch at the end that acts as the arbiter [114]–[118]. A challenge is

composed of k bits configuring the multiplexers. The overall path delay is a function of process variations, thus for an input edge that propagates along both paths, one of the paths will be faster, which is decided by the arbiter that generates the single response bit. The arbiter uses n delay path circuits built by a serial connection of multiplexers to generate an n-bit PUF response. On the other hand, the Ring Oscillator (RO) PUF consists of N ROs and a control logic that compares the oscillation frequency of two different ROs [119]–[122]. A challenge selects a pair of ROs and depending on which RO is faster a single response bit is generated. There are $N(N-1)/2$ possible pairings, but due to correlations a maximum of $\log(N!)$ bits can be extracted.

- *Memory-based PUFs* exploit the instability of volatile memory cells, e.g., Static Random-Access Memory (SRAM), flip-flop, and latch. For example, an SRAM cell has two stable states, i.e., 1 or 0, and positive feedback that forces the cell into one of these two states to prevent undesired transitioning [123]–[126]. The SRAM is powered-up with no write operation and each SRAM cell relaxes into either the 1 or 0 state depending on process variations within it. An n-bit PUF response can be extracted by selecting the logic state of n SRAM cells.

- There exist also *non-electrical PUFs* that are not appealing for integration in electronic circuits, such as optical PUFs [70], [127], [128]. The underlying principle of optical PUFs is the application of a laser beam to an inhomogeneous material from a specific position with a specific angle. A charge-coupled device (CCD) camera then captures the reflected beam that represents a random interference pattern known as a "speckle". This speckle is then analyzed to determine the scattering of the beam. A challenge is represented by the position and angle of the laser beam, while a response is represented by the speckle.

- *Nano-based PUFs* exploit the process variations in nano-electronic devices such as memristors, carbon nanotubes, silicon nanowires, etc. For example, memristors have been used to build PUFs similar to the SRAM-PUFs [129], [130]. Memristor-crossbar is used as memory-cells, and then a write pulse with a fixed duration is applied to each memristor. Because of process variations, this pulse causes some memristors to turn ON while others remain OFF. The duration and amplitude of the write pulse are considered the challenge in this case, while the memristor status, i.e., ON or OFF, represents the PUF's response. Although there are benefits of utilizing nano-electronic devices for building PUFs, i.e., low power consumption and tiny footprint, these technologies suffer from instability and

their compatibility with current CMOS technology is insufficiently mature.

2.4 CONCLUSION

In this chapter, we reviewed the state-of-the-art on analog locking, HT design and defenses in the analog domain, and PUFs. In general, the existing analog locking techniques in the literature suffer from one or more disadvantages. For example, biasing locking does not consider the effect on the biasing circuit performance and is vulnerable to attacks. Calibration locking techniques are vulnerable to removal attacks. Techniques leveraging logic locking present some non-negligible area and power overheads, and also depend on the strength of the employed logic locking technique. In Chapter 3, we will address all these issues for the class of programmable analog ICs by proposing a lock-less technique.

The reviewed HT attacks in the analog domain are easily detectable via physical inspection, side-channel fingerprinting, testing, and other specific measurements. In Chapter 4, we will propose a novel HT attack that exploits the test infrastructure and is undetectable in the analog regime.

Current PUF architectures rely on physical cell redundancy to generate a large number of key-bits. As a result, they present large area and power overheads. In Chapter 5, we will propose a novel PUF architecture based on a single spiking neuron that uses temporal redundancy to generate an arbitrarily large number of key-bits, which effectively reduces the area and power overheads.

3

LOCKING BY UNTUNING: A LOCK-LESS APPROACH FOR ANALOG AND MIXED-SIGNAL ICS

3.1 INTRODUCTION

In this chapter, we focus on IC/IP piracy and we propose an anti-piracy countermeasure for analog ICs that embed digitally-controlled programmability. IC/IP piracy includes reverse engineering and different counterfeiting types, namely cloning, overbuilding, remarking, and recycling as discussed in Chapter 1.

As described in more detail in Chapter 2, a common characteristic of the existing analog locking methodologies is that they all insert a lock into the design. The lock is thoughtfully inserted into peripheral circuitry, i.e., biasing circuitry or on-chip calibration mechanism, or into digital sections with the aim to be as non-intrusive as possible to the sensitive analog core. These lock insertion approaches also offer the possibility to use a large key size, which is a prerequisite for thwarting counter-attacks aiming at recovering the key. Still, as discussed in Chapter 2, these lock insertion approaches may require changes in the analog design flow, may degrade analog performance, may result in some justifiable yet non-negligible area and power overheads, and some are also vulnerable to counter-attacks. In this chapter, we argue that a lock-less solution can be envisioned in the case where the analog IC offers multiple-bit programmability.

In particular, we argue that the tuning knobs within the programmable analog IC can naturally serve as a locking mechanism. Specifically, the programming bits controlling the tuning knobs serve as key-bits and each configuration setting, i.e., programming bits that configure the IC in a specific operation mode demanded by the application, is treated as a secret key. Naturally, when invalid programming bits are provided the functionality of the circuit breaks. We discuss the practical implementation of this locking approach, its benefits compared to existing locking techniques, and its resilience against foreseen attacks. We demonstrate it with simulation on a 64-bit programmable $\Sigma\Delta$ modulator and with hardware measurements on a 194-bit programmable $\Sigma\Delta$ modulator. These modulators are used in highly-digitized, multi-standard RF receiver applications.

The rest of the chapter is structured as follows. In Section 3.2, we provide a general overview of programmability embedded into analog ICs. In Section 3.3, we present the proposed locking methodology and we discuss its benefits and its resilience against foreseen counter-attacks.

In Section 3.4 we present simulation results on an RF receiver. In Section 3.5, we present hardware measurements on an RF ΣΔ modulator. Section 3.7 concludes the chapter.

3.2 PROGRAMMING OF ANALOG ICS

Analog ICs are often demanded to be programmable (or configurable) with the aim to: (a) Compensate for process variations and inherent non-idealities so as to achieve the desired performance trade-off and boost yield; (b) Configure the circuit into different operation modes demanded by the application; (c) Adapt the performance to changes in the environment, e.g. towards moderating power consumption; (d) Enable fault tolerance in the presence of aging, latent defects, single event upsets, etc.

Programmability (or configuration) is enabled by judiciously inserting tuning knobs (or actuators) into the design that act on the circuit performances. Typically, tuning knobs are programmable bias sources that set the current of voltage bias in a node or are implemented by tunable single components, i.e., resistors, capacitors, varactors, etc. Ideally, tuning knobs should act orthogonally on the circuit performances so as to facilitate finding a good balance among multiple competing performance goals; however, this orthogonality property is difficult to achieve in practice and a tuning knob typically acts simultaneously on multiple performances invoking a trade-off, which makes the programming more tangled. Typically, tuning knobs are digitally-controlled, that is, a configuration setting is a digital word.

The programming is driven by a calibration algorithm that uses performance indicators, i.e., direct measurement of performances or information-rich measurements, to search in space of tuning knob settings so as to achieve the target performance objectives. The calibration algorithm returns the configuration setting (or programming bits) that sets the optimal performance trade-off given the target objective. It is an optimization process that involves multiple testing/tuning iterations where in each iteration the next best tuning knob setting is selected based on the current trade-off of measured performances. The calibration algorithm can be implemented off-chip or on-chip.

Off-chip calibration requires that the chip is interfaced with the Automatic Test Equipment (ATE). The ATE applies test stimuli, analyses the test response, and generates the tuning knob values. The calibration algorithm also runs on the computer of the ATE. The tuning knobs are accessed and controlled from a primary pin via a test bus. The calibration can be driven directly by the measured performances or can be assisted by DfT structures with the aim to reduce the test cost, i.e., interface the chip to low-cost ATE, the number of test configurations, and test time.

For example, in [131], a built-in envelope detector is used to extract the low-frequency envelope of the output of an RF transmitter from which multiple RF performances are predicted implicitly in a single test step using the alternate test principle. In [132], the loop-back test is used to analytically compute the parameters of the RF transceiver using baseband test signals and, thereafter, these parameters are used for pre-distortion or post-distortion to digitally calibrate the RF transceiver. In [133]–[135], it is shown that by adding on-chip process variation-aware sensors that are not electrically connected to the circuit under calibration, the calibration can be performed in "one-shot" based on machine learning algorithms. An off-chip implementation of the calibration algorithm can address objectives (a)-(c). For objectives (b)-(c), operation modes and adaptation levels need to be pre-specified based on the anticipated range of applications and environmental conditions, resulting in multiple pre-specified configuration settings. In this case, the calibration algorithm returns a LUT with the multiple pre-specified configuration settings that is pre-loaded into the chip before deployment and stays fixed during its entire lifetime. Based on the application and the environmental conditions met in the field, the appropriate configuration setting is selected from the LUT.

An on-chip implementation [132], [136]–[140] requires an on-chip infrastructure that includes, for example, measurement acquisition sensors for obtaining performance indicators, ADCs for digitizing the measurements, digital post-processing circuitry that drives the tuning knob values optimization given the current measurements, and Digital-to-Analog Converters (DACs) if the actuating signals are analog. An on-chip calibration is automated and can be completed faster compared to off-chip calibration since the process takes place entirely on-chip and there is no need to offload test signals and perform off-chip analysis. An on-chip implementation can address all aforementioned objectives (a)-(d). For objectives (b)-(c), the LUT approach can be followed. For objective (c), in the case where the changes in the environment cannot be anticipated, a fully embedded on-chip calibration scheme offers the possibility to run the calibration on-chip upon request. For objective (d), this is required since fault scenarios are manifold. Clearly, an on-chip implementation offers larger flexibility compared to an off-chip implementation. It can decide on the best configuration setting considering the current status of the chip. However, this comes at the expense of area overhead and design complexity, thus oftentimes it is not the preferred solution by designers.

Often the same calibration mechanism is used to achieve multiple of the above objectives. For example, the configuration setting for each operation mode may take into account process variations and non-idealities so as to achieve the most advantageous performance trade-off for each operation mode. In this case, the configuration settings are unique for each chip.

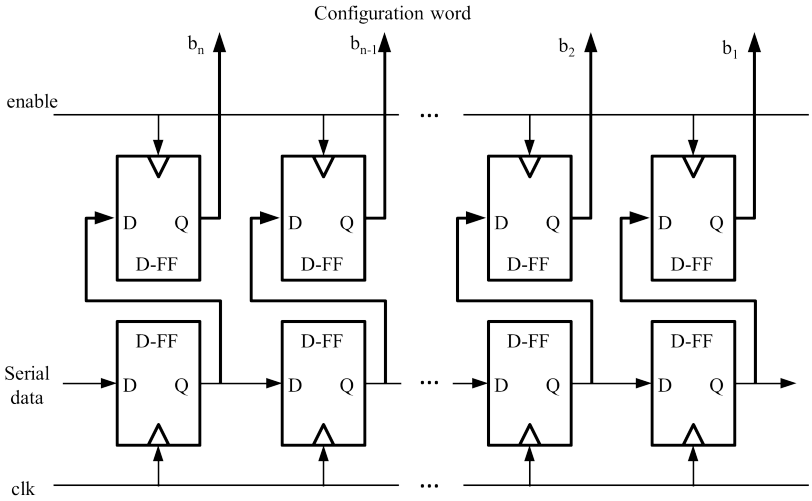


Figure 3.1: Configuration word loading and storage.

The above discussion remains quite general; in fact, the calibration scheme varies from one circuit class to another, and for a given circuit class it varies also from one architecture to another. Moreover, programmability may vary from a few bits for calibrating single blocks to tens of bits for calibrating complete systems. An example of a highly programmable system is multi-standard RF transceivers that aim at a maximum hardware share amongst different communication standards at a minimum power consumption [141]–[143].

During calibration, the programming bits $\{b_1, b_2, \cdots, b_n\}$ of the current configuration word are streamed into the chip propagating through a serial shift register composed of cascaded D-FFs, as illustrated in Fig. 3.1. Once the full configuration word is loaded into the serial register, it is latched to a parallel register where it is stored, activating the programmable analog IC. The parallel registered is refreshed in every iteration of the calibration algorithm. In the field of operation, the configuration words resulting from calibration are pre-stored in the memory, and the configuration word corresponding to the desired operation mode is loaded.

The tuning knobs in the analog design are typically of two types, namely programmable binary-weighted passive arrays setting the passive elements values, i.e., resistors and capacitors, and programmable binary-weighted current mirrors setting the biases of active blocks.

Fig. 3.2 depicts a programmable binary-weighted current mirror. The programming bits control which mirroring branches are turned on, contributing to the adjustment of the mirroring ratio. The resultant bias current is given by $\sum_{i=1}^{n} b_i \cdot 2^i \cdot I_{ref}$. Fig. 3.3 depicts a programmable capacitor array. In this case, the programming bits control which capacitors contribute to the equivalent capacitance, which is given by $\sum_{i=1}^{n} b_i \cdot 2^i \cdot C$.

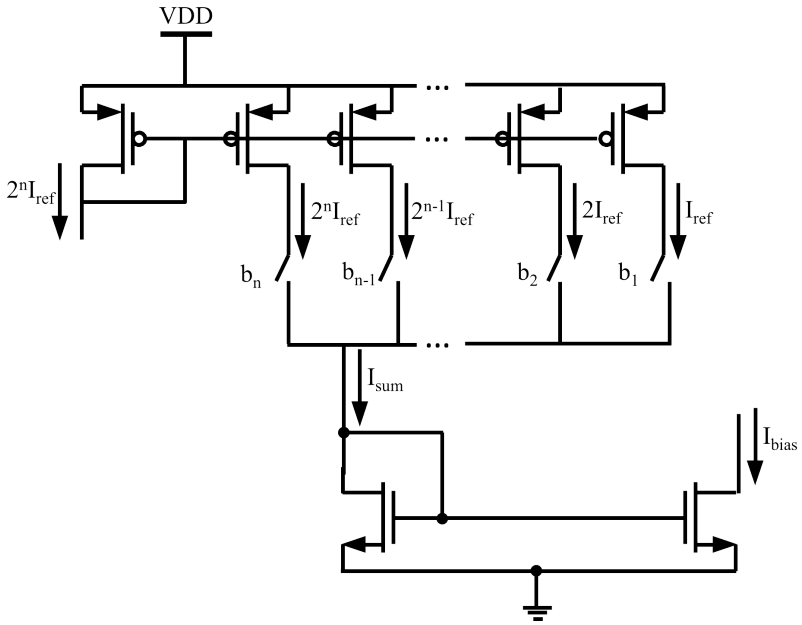


Figure 3.2: Programmable current mirror.

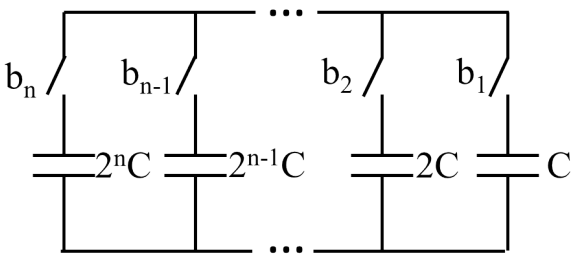


Figure 3.3: Programmable capacitor array.

3.3 LOCK-LESS LOCKING OF AMS ICS

3.3.1 *Locking Principle*

We argue that for highly-programmable analog ICs with off-chip calibration it is not required to insert additional circuitry on-chip, i.e., a lock, in order to introduce key-bits. Instead, we can take advantage of the embedded programmable fabric so as to naturally perform a lock-less locking operation, as shown in Fig. 3.4. Specifically, the configuration settings resulting from the calibration algorithm can be treated as secret keys or, equivalently, the programming bits can be treated as secret key-bits. The calibration algorithm that produces the configurations settings is also kept secret and is not shared with any untrusted and potentially malicious party. Using invalid programming bits will result in untuning the circuit inciting complete loss of functionality or significant performance

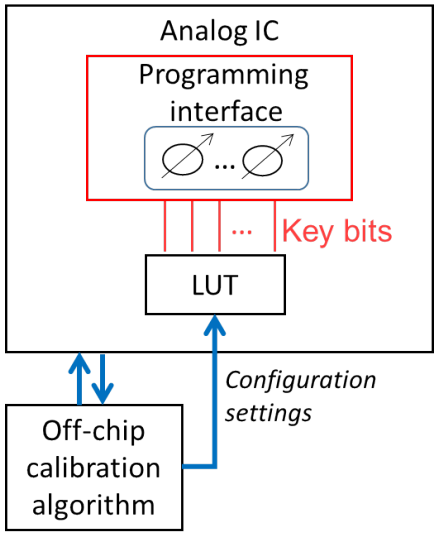


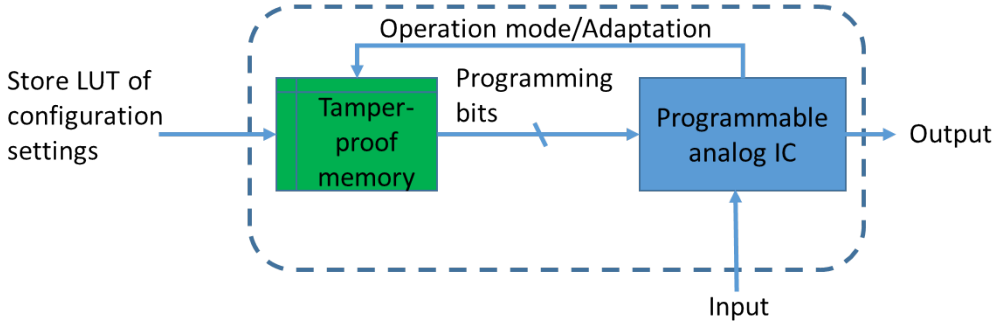
Figure 3.4: Lock-less locking via the programmability interface.

degradation, that is, one or more performances will lie far outside their allowable specification range.

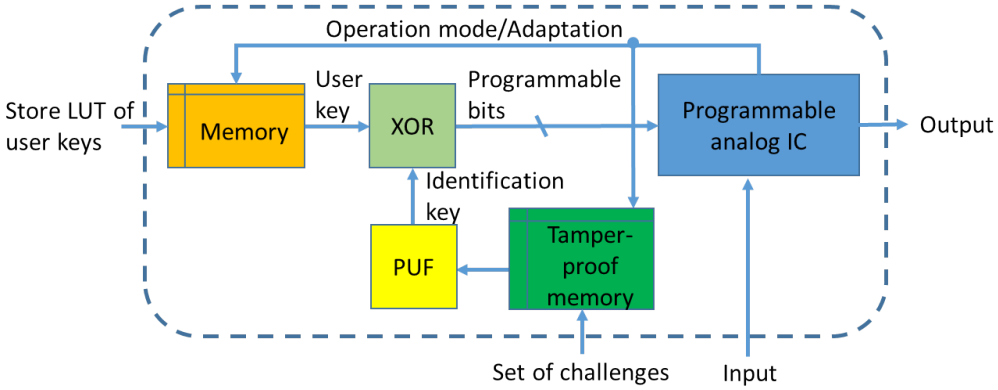
Exploiting the embedded programmability for performing the locking operation presents significant advantages. Unlike known approaches for locking analog ICs which all modify to some degree the design [55], [85]–[88], [93], [95], with the proposed approach the design is left completely intact. Therefore, there is no need for redesign, no extra design iterations, and, most importantly, no performance degradation. In addition, the proposed approach does not increase the power or area of the analog IC itself. The power and area overheads are only due to the key management scheme, which can be shared for enabling security for all other blocks on the same die, i.e., in the context of a SoC.

In the case of on-chip calibration, an attacker that extracts the netlist will also have at hand the hardware that implements the calibration feedback loop and, thereby, it may be fairly easy to extract the calibration algorithm. In this scenario, we can envision logic locking of the digital section of the calibration feedback loop [93], thus in this work we do not treat this scenario.

The possible secret key management schemes are the same ones used by any locking methodology, for digital or analog ICs alike. One option is to directly store the LUT with the configuration settings into a TPM, as shown in Fig. 3.5(a). A second option, illustrated in Fig. 3.5(b), makes use of a PUF. The PUF needs to take at least as many challenges as the total number of configuration settings. This key management scheme uses a number of triplets {challenge, user key, configuration setting} that equals the number of configuration settings. The challenges are pre-stored in a TPM. For a target configuration setting, the corresponding challenge is driven to the PUF from the TPM and the PUF generates a secret



(a) Key management scheme based on directly storing the LUT of configuration settings into a TPM.



(b) Key management scheme using a PUF.

Figure 3.5: Key management schemes.

identification key. The user key is defined such that when XORed with the identification key the target configuration setting is produced. In both schemes, for updating its operation mode or for an adaptation to the environment, the circuit commands dynamically the memories to load the corresponding programming bits.

The proposed locking methodology can serve as a countermeasure against cloning by a malicious SoC integrator, foundry, or end-user that performs reverse-engineering, as well as overbuilding against a malicious foundry. It can offer resilience against recycling only if the key management scheme in Fig. 3.5(b) is used and the unique user keys are re-loaded every time at power-on.

There are four options for performing the testing/calibration phase in such a way that: (a) resilience is achieved against remarking by gaining control over the number of activated functional chips; (b) the secrecy of the calibration algorithm and configuration settings is preserved. These options are:

1. The testing/calibration phase is performed in a trusted test facility.

2. The foundry returns the manufactured chips to the design house and owner of the IC, where the testing/calibration steps are performed in a secured environment.

3. Only structural defect-oriented testing is performed at the untrusted test facility, which does not require calibrating the chip first. Thereafter, calibration is performed by the trusted design house. Performing structural testing only is possible for designs with well-centered performances that have an 100% parametric yield. A metric to assess the centering of a performance is its C_{pk} value, where $C_{pk} = \min\left[\frac{USL-\mu}{3\sigma}, \frac{\mu-LSL}{3\sigma}\right]$, USL and LSL denote the upper and lower specification limits of the performance, respectively, and μ and σ denote the mean and standard deviation of the performance, respectively. Structural test approaches that are generic or specific to an analog IC class are continuously being proposed [144]–[148], and recently introduced industrial analog defect simulators [149], [150] assist in performing efficiently fault simulation towards test generation and test quality assessment.

4. The testing/calibration phase is performed in an untrusted test facility using secured remote calibration based on asymmetric cryptography [25]. More specifically, multiple test/calibration iterations are carried out for searching for the best configuration settings. In each step of the calibration algorithm, the next configuration setting is dictated by tests done using the current configuration setting. The trusted design house holds and runs the calibration algorithm and generates the next best configuration setting, while the untrusted test facility generates the test results for a given configuration setting. In each step, the design house communicates the configuration setting securely to the chip which interfaced to the ATE in the untrusted test facility. The test result produced in the test facility is sent back to the trusted party to drive the generation of the next configuration setting, and so forth. If the calibration ends based on stop criteria and the performance specifications are not met, then the IC is labeled as faulty. To avoid remarking, the design house can deliberately load into the on-chip TPM largely offset configuration settings to render the chip totally malfunction. For functional ICs, at the end of the calibration, the configuration settings that correctly activate the IC are stored into the on-chip TPM. Note that the calibration is automated on both sides and that the test results do not need to be secured.

3.3.2 *Security Analysis*

We consider the most favorable threat model for an attacker. We assume that the attacker has full capabilities, i.e., has the circuit netlist and access to an unlocked oracle chip. Herein, we list the foreseen attacks, based also on all the known attacks in the literature in both the analog and digital domains, and we argue about the resilience offered by the proposed locking methodology.

3.3.2.1 *Attacks in Digital Domain*

Known attacks in the digital domain aiming at breaking logic locking, such as the most lethal Boolean Satisfiability (SAT)-based attack [151], are not applicable since the proposed locking methodology does not employ logic locking.

3.3.2.2 *Removal Attacks*

Removal attacks aim at removing or bypassing the lock. They are not applicable since the proposed locking methodology is lock-less; the key directly applies to existing tuning knobs into the design.

3.3.2.3 *Attacks on Biasing Locking*

The proposed locking methodology uses as key-bits the programming bits of tuning knobs, where a large class of tuning knobs is bias sources. All attacks on biasing locking work by considering the obfuscated component within the bias source [89]–[92]. In the proposed locking methodology, bias sources are not obfuscated; only their programming bits are kept secret. Therefore, attacks on biasing locking are not applicable.

3.3.2.4 *Brute-Force and Multi-Objective Optimization Attacks*

The brute-force attack consists in applying random combinations of programming bits, i.e., keys, until a key is found that unlocks the circuit, i.e., brings all the performances within the acceptable specification range. Instead, the attacker can employ a multi-objective optimization algorithm, such as gradient descent, simulated annealing, Genetic Algorithm (GA), etc., to search more efficiently in the key space. For analog ICs it is likely that a number of keys result in a satisfactory performance trade-off, although this number is typically a very small fraction of all keys. Furthermore, the fact that the tuning knobs are binary-weighted has also a security implication: no two different configuration words can produce the same tuning knobs, i.e. bias currents, resistances, or capacitances. This limits the number of keys that can establish an acceptable performance trade-off. Only the keys that produce tuning knob values in the

neighborhood of the nominal tuning knob values can meet this objective, and this number of keys is a tiny fraction of the key space. These attacks require simulating the circuit, computing the performances at every iteration, and comparing them to the performances in the datasheet or the performances of the oracle chip.

Resilience against these attacks is proportional to the key size and to the simulation time. For analog ICs one simulation run can be extremely time-consuming, thus the attacker in practice can afford carrying out just a few iterations. The key size for our case study circuit is very large, and the simulation time is on the order of one day. Thus, these attacks are infeasible.

One workaround for the attacker could be dividing the circuit into sub-blocks, tracing key-bits to sub-blocks, and enabling smaller brute-force and multi-objective optimization attacks at sub-block level. This is not possible for two reasons. First, an analog circuit typically has internal feedback loops that involve multiple sub-blocks each, thus sub-blocks cannot be considered individually. Second, the performances of sub-blocks are not documented in the datasheet and the oracle does not offer access to sub-blocks for measurements.

With that said, there is another important defense against these attacks which is the fact that configuration settings are unique for each chip taking into account inter-die variations. These attacks become meaningful only if the extracted key from simulation of the nominal design can be used to set a good starting point for launching a gradient search for quickly calibrating any chip.

3.3.2.5 *Revealing the Calibration Algorithm*

The attacker may target speculating the calibration algorithm by studying the circuit architecture. However, very often the calibration algorithm is very specific and esoteric to the design, that is, intended for or understood only by the designer. Thus, the attacker must have a very high and specialized expertise and a thorough understanding of the design so as to be able to conceive the underlying calibration algorithm. Revealing the calibration algorithm is a new type of attack specific to the countermeasure that is proposed. In general, it opens a discussion for securing and obfuscating calibration algorithms when they are considered to be a valuable IP of the design.

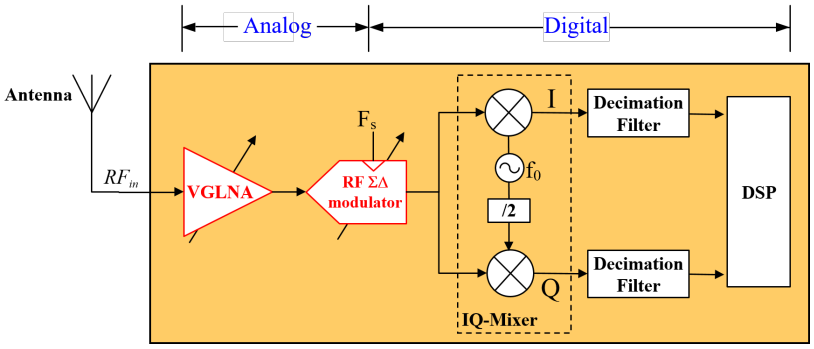


Figure 3.6: Architecture of programmable multi-standard RF receiver.

3.4 SIMULATION RESULTS

3.4.1 *RF Receiver Architecture and Programmability*

The recent increase in wireless communication standards has pushed the development of multi-standard RF transceivers [141]–[143], where the same RF transceiver circuit serves for establishing communication using several standards. Different standards have different requirements in terms of sensitivity, center frequency, bandwidth, resolution, etc. Therefore, it is necessary that many blocks within the RF transceiver are made programmable, in order to be able to adapt its specifications to the requirements imposed by the target standard. The configuration of the blocks is performed thanks to judiciously inserted tuning knobs which are controlled by digital programming bits.

Our simulation case study for demonstrating the proposed locking methodology is the analog section of a highly-digitized, multi-standard RF receiver that is designed in 65 nm CMOS process, as illustrated in Fig. 3.6. The ΣΔ modulator is used to directly convert the RF signal at the output of the Variable Gain Low Noise Amplifier (VGLNA) to the digital domain. It over-samples the analog input signal and generates a high-frequency, low-resolution 1-bit digital signal at its output. This signal is then down-converted by a digital mixer and filtered using a digital decimation filter. It is made re-configurable such that it can serve for establishing communication using several standards within the frequency range from 1.5 GHz to 3.0 GHz, including Bluetooth, ZigBee, WiFi 802.11b, as well as several standards dedicated to the second generation 2G and the fourth generation 4G broadband cellular networks, such as GSM, GPRS, EDGE, LTE1800, LTE2100, and LTE2600.

The analog section is composed of a VGLNA and a 2nd order band-pass (BP) RF ΣΔ modulator. In total, there are 64 programming bits embedded into the analog section.

The block-level schematic of the VGLNA is shown in Fig. 3.7. It is composed of five gain stages with a resistive feedback. It features a 4-bit

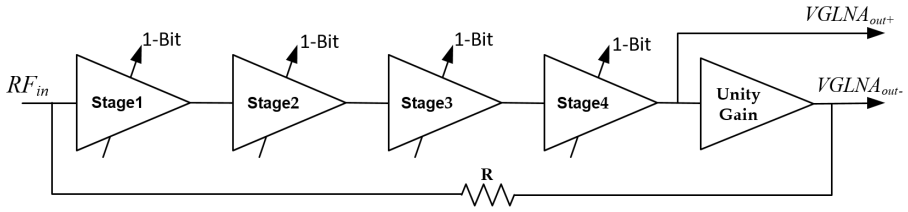


Figure 3.7: Architecture of tunable variable gain LNA.

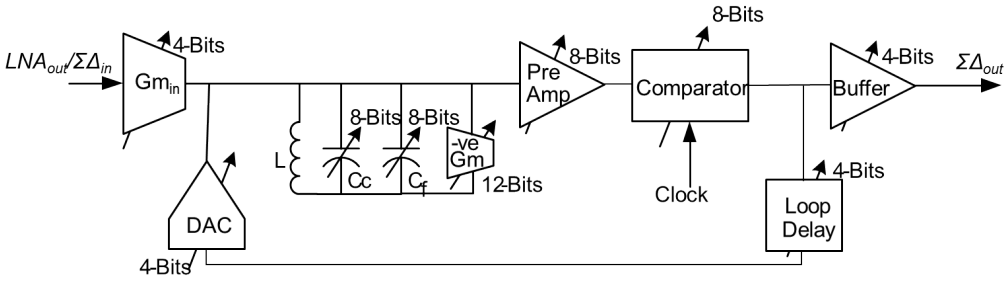


Figure 3.8: Architecture of tunable 2nd order BP RF ΣΔ modulator.

configuration word with which the VGLNA can attain 16 different gain levels so as to adapt the sensitivity and dynamic range of the RF receiver to the specifications imposed by the target standard.

The block-level schematic of the BP RF ΣΔ modulator is illustrated in Fig. 3.8 [152]. It is composed of an input transconductance, Gm_{in}, an LC bandpass loop filter with two capacitor arrays C_c and C_f for coarse- and fine-tuning, respectively, a pre-amplifier, a comparator, a tunable loop delay, a feedback DAC, and an output buffer. For a target standard, the modulator uses a 60-bit configuration word to tune the center frequency and quality factor of the LC bandpass loop filter in the presence of process variations, as well as to trim the biasing current of the other blocks in order to compensate for process variations and improve the performance trade-off.

3.4.2 *Calibration Algorithm*

For identifying the proper programming bits for a target standard, the RF receiver employs an off-chip calibration algorithm that is design-based, complex, and only known by the designer, thus it can be kept secret. For a certain standard, the calibration algorithm is used in a precised iterative manner until the required performance is fulfilled. The first steps of the calibration procedure are illustrated in Fig. 3.9. In more detail, the calibration procedure is as follows:

1. The comparator is configured as a buffer by deactivating its driving clock.

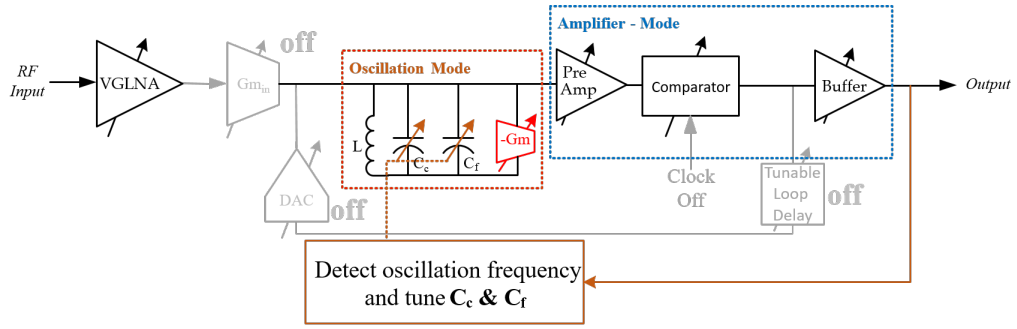


Figure 3.9: The 2nd order ΣΔ modulator configured for calibrating the LC loop filter.

2. The output buffer is configured to adapt the output of the BP RF ΣΔ modulator to its off-chip load during calibration. This output buffer is removed from the signal path in normal operation mode.
3. The RF input signal is disabled by turning off the input transconductor Gm_{in}.
4. The feedback loop with the DAC and loop delay is turned off.
5. Having deactivated the feedback loop, the LC loop filter is put in oscillation mode by setting its Q-enhancement transconductor, -Gm, to its maximum.
6. The capacitor arrays C_c and C_f of the LC tank are tuned until the output frequency is equal to the desired center frequency.
7. The Q-enhancement transconductor, -Gm, is reduced gradually until oscillation vanishes.
8. The feedback loop is restored.
9. The BP RF ΣΔ modulator is put in the operating mode by applying an RF input signal with frequency F_0.
10. The sampling frequency is set to $F_s = 4 \cdot F_0$.
11. The loop delay is set according to F_s.
12. The VGLNA is tuned to set the appropriate sensitivity and dynamic range.
13. The input transconductance Gm_{in}, the feedback DAC, the preamplifier and the comparator are initialized to their nominal values determined by simulation.

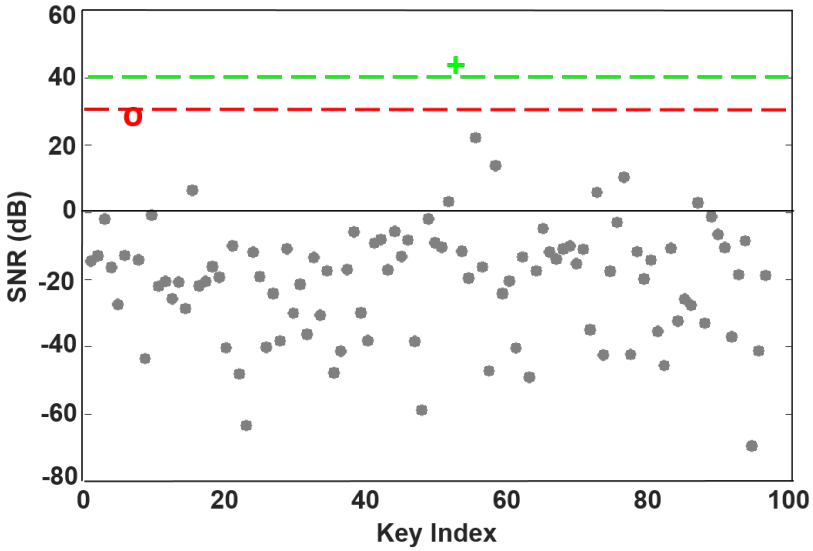


Figure 3.10: SNR for correct key (green cross) and invalid keys (gray dots and red dot with index 7) computed at the output of the BP RF ΣΔ modulator.

14. An iterative procedure is used to determine the configuration words of these blocks through the improvement of the measured Signal-to-Noise Ratio (SNR) and spurious-free dynamic range (SFDR) of the BP RF ΣΔ modulator.

3.4.3 *Locking Results*

For a given center frequency or standard, there is an optimal combination of the 64 programming bits composing a secret key that unlocks functionality. We will consider the maximum center frequency, i.e., 3 GHz, and we will demonstrate the locking efficiency when applying invalid keys. The circuit has several performances, including SNR, dynamic range, SFDR, etc., and locking succeeds when at least one performance violates its specification.

We assume that the attacker has extracted the netlist of the circuit and can simulate it at the transistor-level with the ability to monitor internal nodes that shed more light into the operation. We consider first the SNR observed at the output of the BP RF ΣΔ modulator for an input sinusoidal signal with frequency 3 GHz and power -25 dBm. The SNR is computed for an Oversampling Ratio (OSR) of 64 and based on an 8192 point FFT. Fig. 3.10 shows the SNR across 100 randomly generated keys and the correct key. As it can be seen, the correct key stands out resulting in an SNR of over 40 dB, while for invalid keys the SNR is less than 30 dB. In fact, for most invalid keys the SNR is below 0 dB, which means that the input signal gets buried under the noise level or there are harmonics

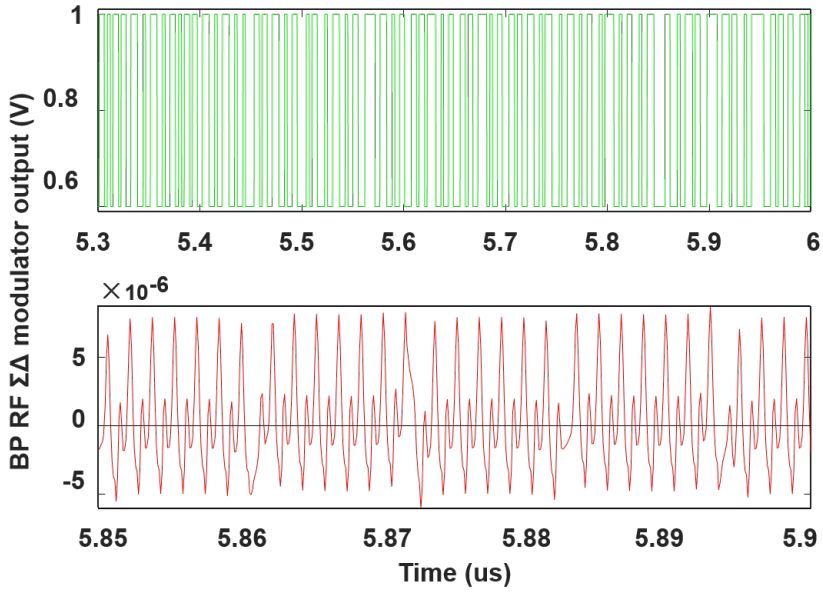


Figure 3.11: Transient output of BP RF ΣΔ modulator for the correct key (top) and the invalid key with index 7 in Fig. 3.10 (bottom).

within the band-of-interest. However, there are 4 invalid keys that have an SNR higher than 10 dB, and among them one has an SNR of about 30 dB. This key with index 7 in Fig. 3.10 naturally would attract the attention of the attacker. But, in fact, it turns out to result in a “deceptive" SNR. Looking into the programming bits corresponding to this key, the feedback loop of the BP RF ΣΔ modulator is open and, in addition, the comparator operates as a buffer. In this way, the analog signal passes without being digitized, thus there is no quantization noise being added. Fig. 3.11 shows the transient output of the BP RF ΣΔ modulator for the correct key and the invalid key with index 7. The correct output is an oversampled bitstream, while the output for the invalid key is an analog waveform showing no analog-to-digital conversion. This analog waveform when it passes directly through the digital section of the RF receiver will show a reduced SNR. This is shown in Fig. 3.12, where now the SNR measured at the output of the RF receiver is plotted. The SNR for the correct key does not change as expected and for some invalid keys the SNR is further reduced. In short, all invalid keys show an SNR of less than 10 dB, that is, the functionality is significantly corrupted.

Fig. 3.13 shows the power spectral density (PSD) at the output of the BP RF ΣΔ modulator for the correct key and the invalid key with index 7. As it can be seen, for the invalid key there is no noise shaping, which is the main characteristic of the BP RF ΣΔ modulator.

Fig. 3.14 shows the dynamic range of the RF receiver for the correct key and the invalid key with index 7. The input range is divided into three segments, e.g. [-85:-45], [-60:-20], and [-40:0], and for each segment

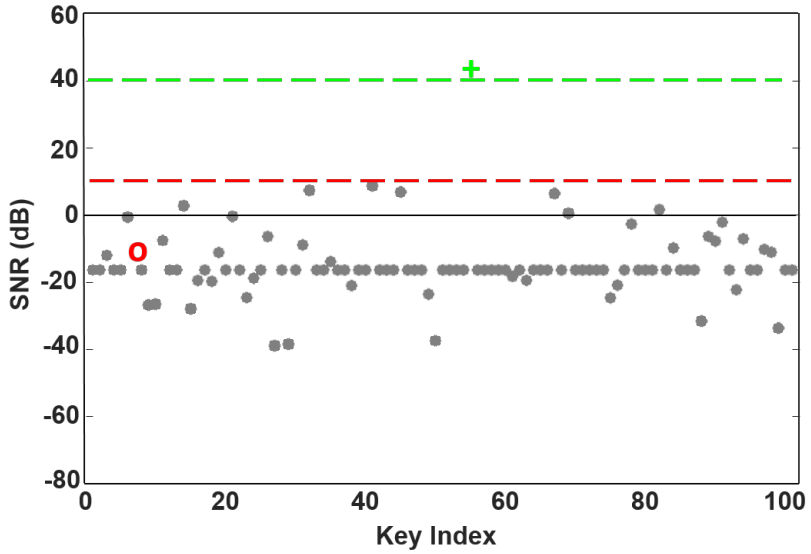


Figure 3.12: SNR for correct key (green cross) and invalid keys (gray dots and red dot with index 7) computed at the output of the RF receiver.

the VGLNA is tuned to have the appropriate gain level and sensitivity. While Figs. 3.10 and 3.12 show the SNR just for an input power of -25 dBm, Fig. 3.14 plots the SNR for different input power values with a step of 5 dBm. As it can be seen, the behavior of the locked circuit across the input range is very different as compared to the unlocked circuit.

Fig. 3.15 shows the SFDR for the correct key and the invalid key with index 7. SFDR is measured by applying a two-tone input, where the two tones have the same power and a frequency difference of 10 MHz. SFDR is the difference between the power of the fundamental and the third harmonic. As it can be seen, the locked circuit has a much lower SFDR.

Finally, the same experiment was repeated for other center frequencies and qualitatively the results were identical.

3.5 EXPERIMENTAL RESULTS

3.5.1 ΣΔ *Modulator Architecture and Programmability*

Our experimental case study for demonstrating the proposed locking methodology is a 6th order programmable BP RF modulator [153] designed in the context of a highly-digitized, multi-standard RF receiver, as illustrated in Fig. 3.6.

The BP RF ΣΔ modulator is designed in a 65nm CMOS process and is a re-spin of the design presented in Section 3.4.1. Its block-level schematic is illustrated in Fig. 3.16. At the higher level, it is composed of a BP loop filter, a comparator, feedback DACs, a tunable delay block, and an output buffer. It is designed specifically for an RF receiver that establishes

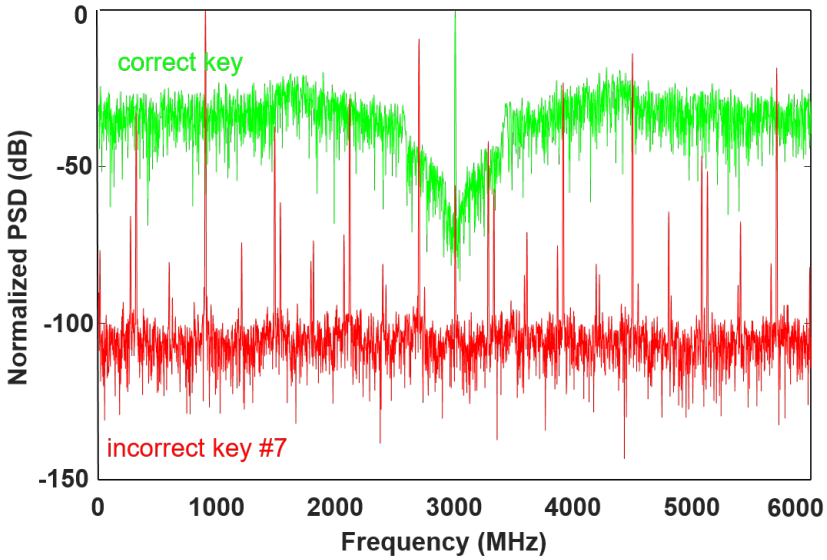


Figure 3.13: PSD at the output of the BP RF ΣΔ modulator for the correct key (green) and the invalid key with index 7 in Figs. 3.10 and 3.12 (red).

communication using several standards within the frequency range from 1.7 GHz to 2.8 GHz, such as Bluetooth, ZigBee, and WiFi 802.11b, etc. All the sub-blocks of the ΣΔ modulator are made programmable to set its operation mode according to the target standard and at the same time to tune its performance trade-off on a per-chip basis in the presence of process variations. In total, the ΣΔ modulator uses a 194-bit programming word. The partitioning of the 194 programming bits into the different sub-blocks and their utility is as follows:

- The center frequency of the loop filter is tuned in the presence of process variations. For this purpose, in each LC task inside the loop filter there are two capacitor arrays, namely an array C_c with a 6-bit configuration word for coarse-tuning and another array C_f with a 6-bit configuration word for fine-tuning.
- The quality factor of each LC tank is tuned in the presence of process variations using a 10-bit configuration word that controls the current of a negative transconductance $-Gm$.
- A 8-bit configuration word is dedicated to trimming the biasing current of the tunable delay block so as to adapt it to the sampling frequency F_s of the modulator.
- The 56-bit configuration word dedicated to the comparator, the 7-bit configuration word dedicated to each Gm inside the loop filter, and the 7-bit configuration word dedicated to each DAC in the feedback loop are independent of the center frequency tuning and are used to trim the biasing currents so as to compensate for process variations.

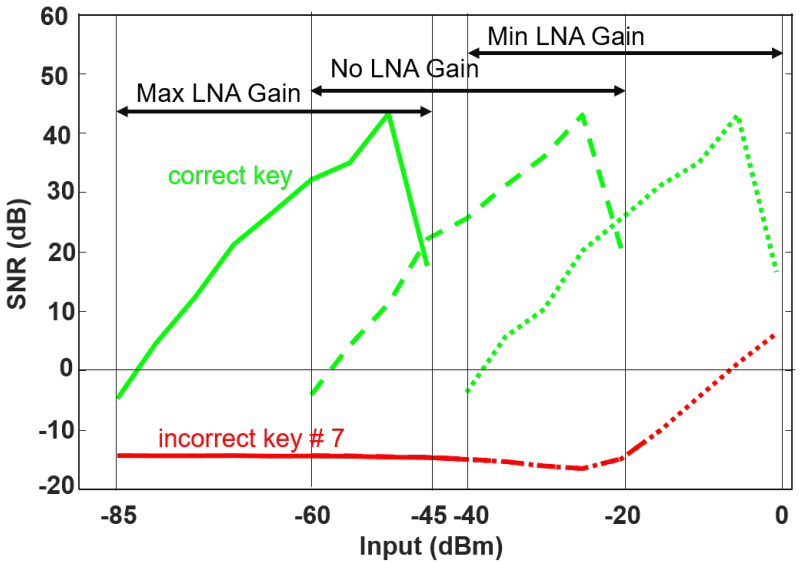


Figure 3.14: SNR versus input power with different LNA gain settings for the correct key (green) and the invalid key with index 7 in figures 3.10 and 3.12 (red).

- The biasing current of the output buffer is controlled through a 8-bit configuration word with the objective of adapting the 1-bit output of the modulator to its off-chip load.

3.5.2 *Calibration Algorithm*

An off-chip calibration algorithm is used to tune a fabricated chip for a given standard in the presence of process variations. Thus, the configuration settings per standard are unique for each fabricated chip. The calibration algorithm is as follows with the first steps illustrated in Fig. 3.17:

1. The comparator is configured as a buffer by deactivating its driving clock.
2. The input signal is disabled by turning off the input transconductance Gm_1.
3. The feedback DACs current is turned off.
4. The third LC tank in the loop filter is put in oscillation mode by setting its Q-enhancement transconductance $-Gm$ to its maximum.
5. The capacitor arrays C_c and C_f of the third LC tank in the loop filter are tuned until the output frequency is equal to the desired center frequency dictated by the standard.

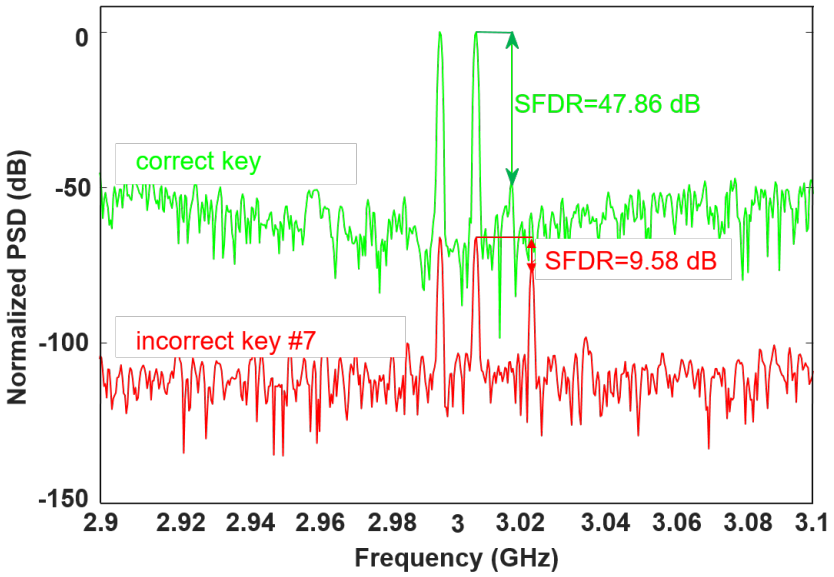


Figure 3.15: SFDR for the correct key (green) and the invalid key with index 7 in Figs. 3.10 and 3.12 (red).

6. The Q-enhancement transconductance $-Gm$ of the third LC tank in the loop filter is reduced gradually until oscillation vanishes.
7. Steps 1-6 are repeated for the second and first LC tanks in the loop filter.
8. The feedback loop is restored.
9. The $\Sigma\Delta$ modulator is put in the operating mode by applying an RF input signal with frequency F_0.
10. The sampling frequency is set to $F_s = 4 \cdot F_0$.
11. The tunable delay is set according to F_s.
12. The tuning knobs of the input transconductance Gm_1, the feedback DACs, and the comparator are initialized to their nominal values determined by simulation.
13. An iterative procedure is used to determine the optimal configuration words of these blocks in the presence of process variations through the improvement of the measured SNR and SFDR of the $\Sigma\Delta$ modulator.

3.5.3 *Locking Results*

Our experiment is conducted using hardware measurements on the actual fabricated chip. Fig. 3.18 shows a photo of the test bench used for

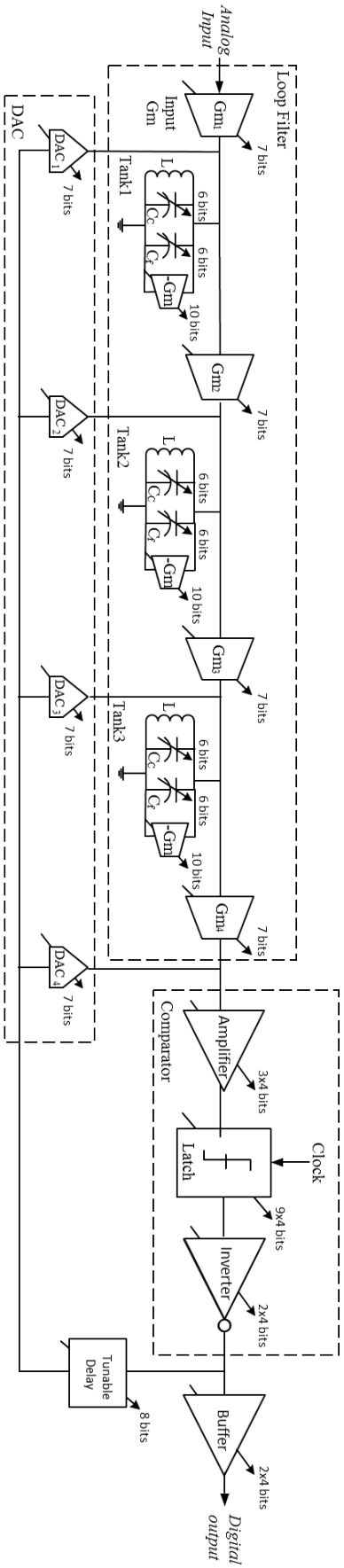


Figure 3.16: Architecture of ΣΔ modulator.

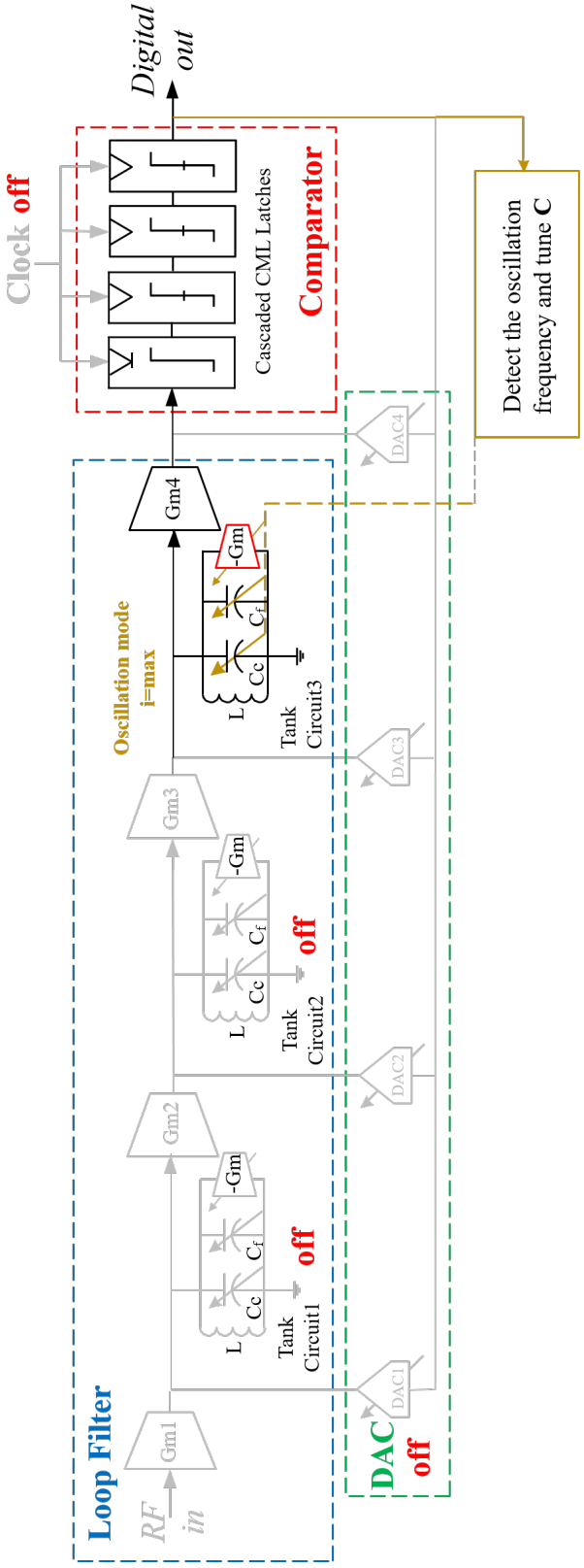


Figure 3.17: The ΣΔ modulator configured for calibrating the LC loop filter.

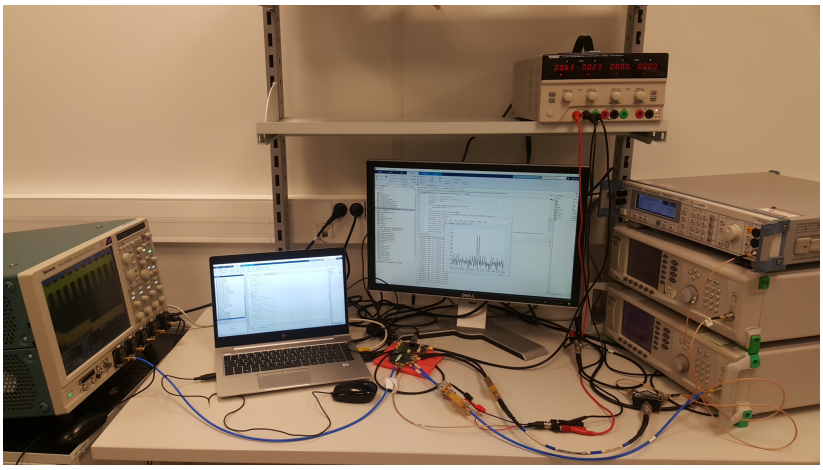


Figure 3.18: Test bench for measuring the performance of the modulator under the unlocking and the different locking scenarios.

measuring the performance of the modulator under the unlocking and the different locking scenarios. Without loss of generality, we consider the operation mode with the maximum center frequency, e.g., 2.8 GHz.

The circuit has several performances, including SNR, SFDR, input third-order intercept point (IIP3), etc. and locking succeeds when at least one performance violates its specification for any incorrect key. Ideally, for an incorrect key all performances should violate their specifications, and the further they lie away from their specifications the more efficient the locking is.

The locking efficiency is first assessed by applying random keys and comparing the resultant SNR with the SNR of the unlocked circuit when the correct key is applied. For this measurement, we consider an input sinusoidal signal with a frequency of 2.8 GHz and a power of -14 dBm. Fig. 3.19 shows the SNR across 5000 randomly generated keys and the correct key. As it can be seen, the correct key stands out resulting in an SNR of over 60 dB, while for incorrect keys the SNR is less than 30 dB. The average SNR across incorrect keys is around 10 dB, while the maximum and minimum observed SNR is 26.6 dB and -21.1 dB, respectively. This experiment shows that applying incorrect keys degrades drastically the SNR performance, thus the locking objective is achieved.

Next, we present measurement results for the rest of the performances considering three keys, namely the correct key and the “best" and the “worst" incorrect keys in Fig. 3.19, i.e., the incorrect keys resulting in the highest and the lowest observed SNR, respectively.

Fig. 3.20 shows the PSD at the output of the ΣΔ modulator. As it can be seen, for the “worst" incorrect key there is no noise shaping shown by the “V” shape in the band-of-interest, which is the main characteristic of a BP ΣΔ modulator. The signal is permanently buried under the noise level.

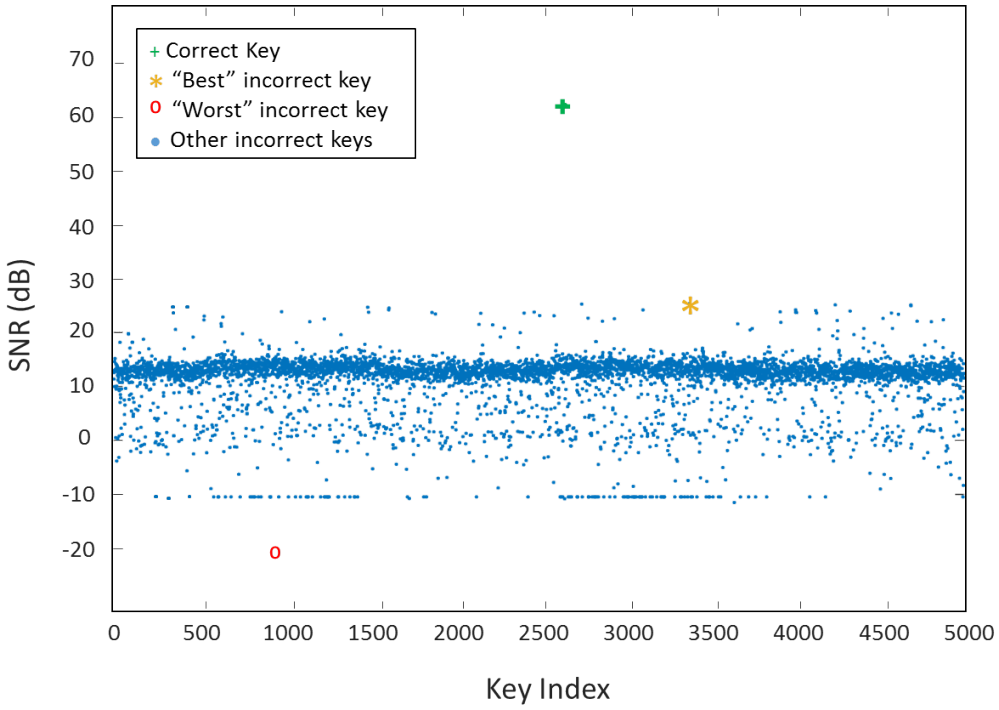


Figure 3.19: SNR for correct (green cross) and invalid (blue dots) keys. Invalid key (orange star) has a relatively high SNR, and invalid key (red circle) has the min SNR.

For the "best" incorrect key, noise shaping is observed; however, there are some harmonics in the band-of-interest reducing the SNR to 26.6 dB.

Fig. 3.21 plots the dynamic range. While Fig. 3.19 shows the SNR for an input power of -14 dBm corresponding to the maximum obtained SNR, Fig. 3.21 plots the SNR for different input power values with a step of 1 dBm. As it can be seen, the "best" incorrect key results in significantly reduced dynamic range, while for the "worst" incorrect key the curve does not show the expected knee behavior and is permanently under 0 dB.

Fig. 3.22 plots the SFDR measured by applying a two-tone input, where the two tones have the same power and a frequency difference of 10 MHz. As it can be seen, the nominal SFDR is 51.39 dB, whereas the "best" and "worst" incorrect keys result in significantly reduced SFDR values of 14.41 dB and 4.15 dB, respectively.

Fig. 3.23 shows the output fundamental power and the third-order intermodulation (IM_3) product versus the input power, from which the IIP3 can be determined. We apply a two-tone input, where the two tones have the same input power and a frequency difference of 2 MHz. The output fundamental power and IM_3 are measured by sweeping the input power from -13 dB to -17 dB with a step size of -1 dB. As it can be seen, the unlocked circuit has a nominal IIP3 of 8 dBm, while the "best" and

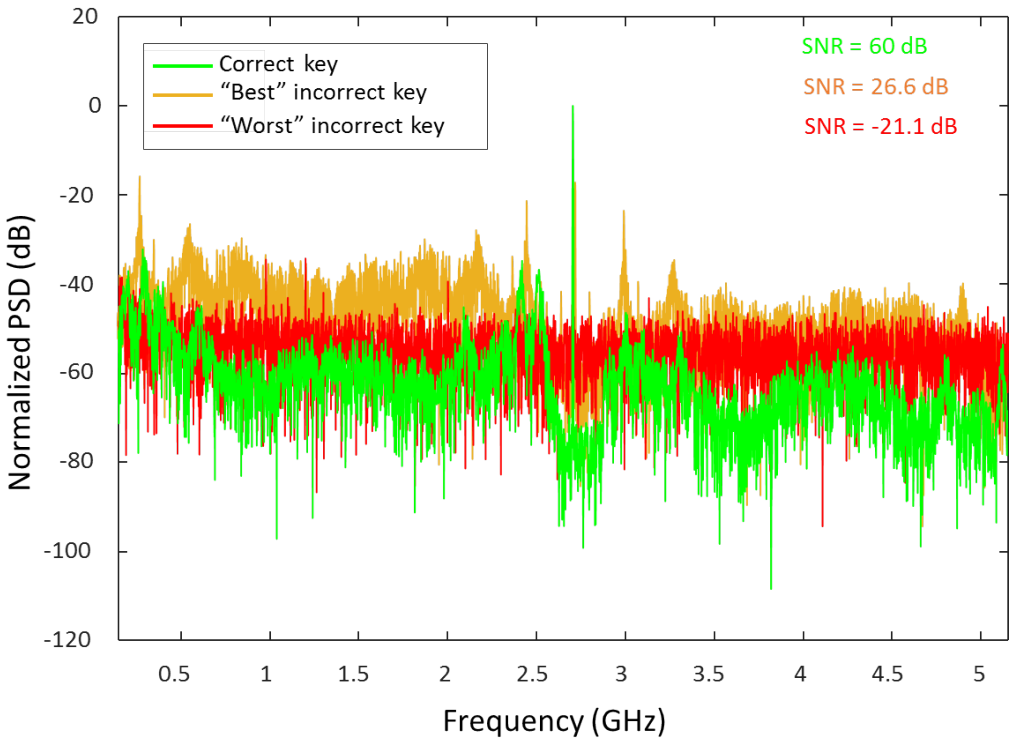


Figure 3.20: PSD at the output of the BP RF ΣΔ modulator for the correct key (green) and the invalid key with max SNR (orange) and the invalid key with min SNR (red) in Fig. 3.19.

the "worst" incorrect keys result in a significantly reduced IIP3 of -8 dBm and -12 dBm, respectively.

As a final note, the same experiment was repeated for other center frequencies and qualitatively the results were identical.

3.6 DISCUSSION ON RESILIENCE AGAINST FORESEEN ATTACKS

3.6.1 *Brute-Force and Multi-Objective Optimization Attacks*

The key space in our case studies is very large, i.e., 2^{64} for the RF receiver case study using simulation and 2^{194} for the ΣΔ modulator case study using hardware measurements, thus the search space for brute-force and multi-objective optimization attacks is huge. Besides, these attacks are implemented at simulation-level, and one transistor-level simulation to compute the circuit performances is extremely time-consuming. More specifically, let us consider a single key and an 8192-point Fast Fourier Transform (FFT). For the 2nd order BP ΣΔ, it takes about 20 minutes to simulate the SNR for a given input power, 3 hours to simulate the SNR across the complete input range, and 30 minutes to simulate the SFDR. For the 6th order BP ΣΔ modulator, it takes about 30 minutes to simulate

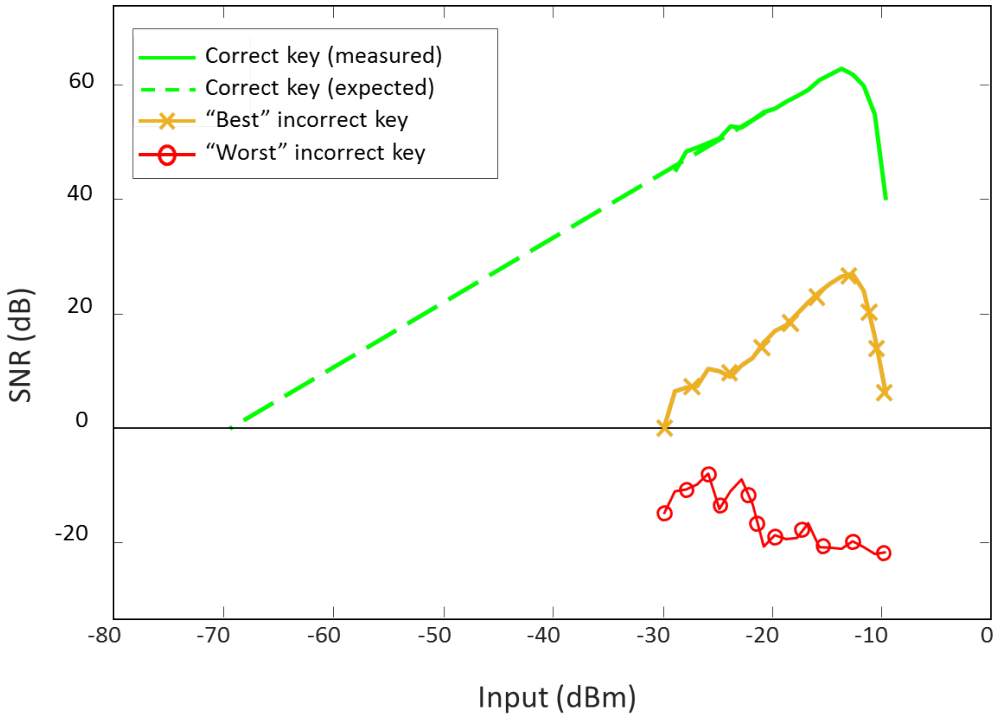


Figure 3.21: SNR versus input power for the correct key (green) and the invalid key with max SNR (orange) and the invalid key with min SNR (red) in Fig. 3.19.

the SNR for a given input power, 4.5 hours to simulate the SNR across the complete input range, 45 minutes to simulate the SFDR, and 3.75 hours to simulate the IIP3. Finally, the overall simulation time per iteration will be in the order of hours, thus only a few iterations can be performed in practice and it will be impossible to "hit" a good performance trade-off within an affordable simulation time.

Furthermore, most sub-blocks are included in a feedback loop which makes it impossible to calibrate individual sub-blocks. Also, to calibrate a sub-block, the rest of the sub-blocks need to be conditioned appropriately. Besides, the circuit carries high-frequency signals and re-fabbing a chip with intermediate taps for observing internal signals would result in performance loss.

Despite the key space is very large, it is very unlikely that many key-bit combinations could result in satisfactory performance trade-off. For instance, current mirrors and capacitor arrays are binary-weighted, thus for a desired current or capacitor value there is a unique sub-key.

3.6.2 *Revealing the Calibration Algorithm*

There are many aspects in the algorithm that make it very complex, thus hindering the attacker's ability to recover it. It is design-specific and its

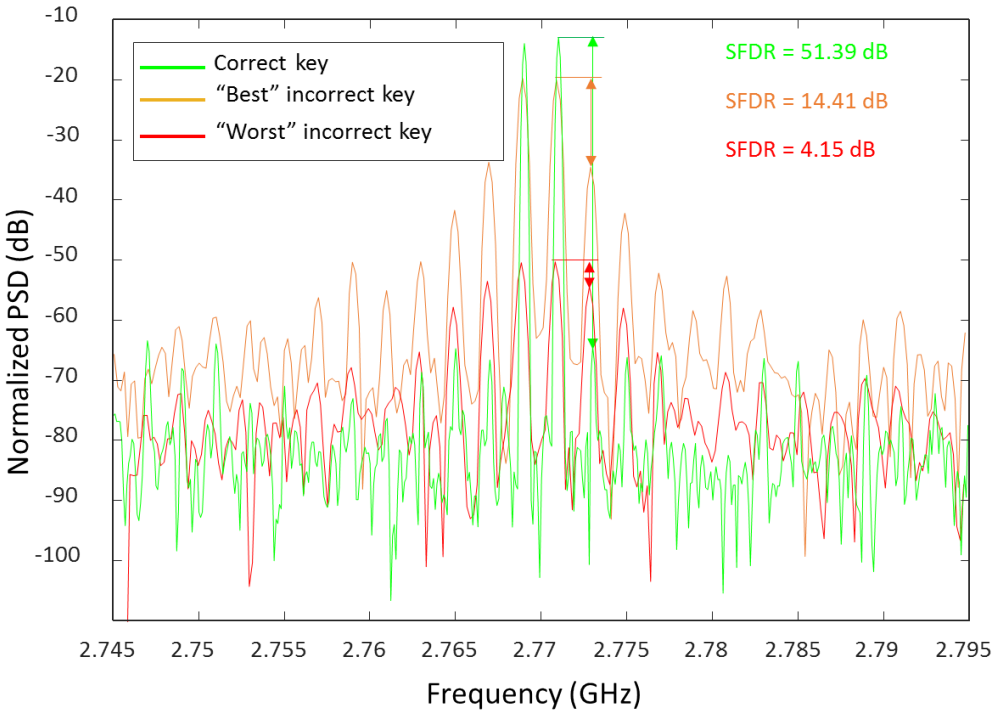


Figure 3.22: SFDR versus input power for the correct key (green) and the invalid key with max SNR (orange) and the invalid key with min SNR (red) in Fig. 3.19.

steps cannot be easily retraced by conjecture even under the assumption that the attacker has strong analog design expertise. In particular: (a) The circuit needs to be reconfigured into appropriate test benches multiple times during calibration; (b) The order with which the different sub-blocks should be calibrated is very specific; (c) Most sub-blocks are included in a feedback loop which prohibits calibrating sub-blocks individually, given also that the target performances of individual blocks per standard are unknown to the attacker; (d) The calibration of many sub-blocks requires initial programming bits that are dictated by design simulation and are unknown to the attacker. If other than these programming bits are used then convergence in a reasonable time is not guaranteed.

3.7 CONCLUSION

In this chapter, we proposed a locking methodology for the class of programmable analog ICs that leverages the existing programmability based on tuning knobs. No lock needs to be inserted into the design since the existing tuning knobs take the role of sub-locks. The key is the concatenation of the programming bits of tuning knobs. By applying an invalid key, the best scenario from the attacker's perspective is that

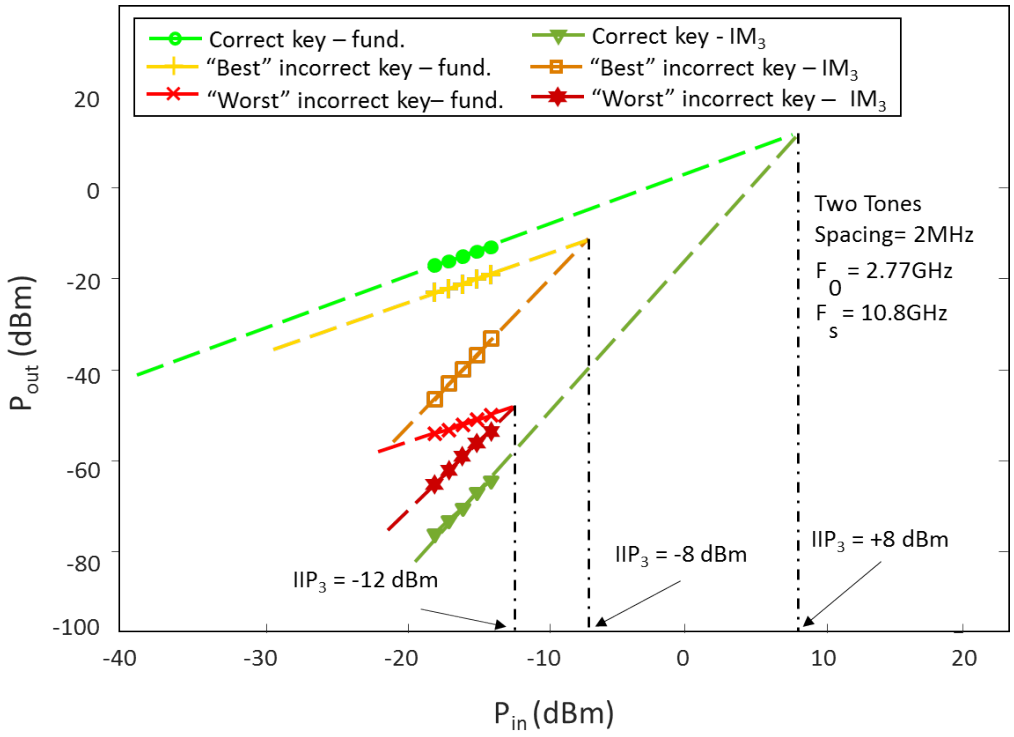


Figure 3.23: IIP3 under unlock and lock operation.

the IC is untuned, i.e., the performances show a degraded trade-off. The owner of the IC needs to keep secret the configuration settings per standard and per chip, as well as the calibration algorithm that is used to generate these configuration settings. The locking methodology presents many advantages that allow its wide adoption by analog designers, i.e., it is strictly non-intrusive to the design, it incurs zero power and area overheads except for the overheads due to the key management scheme that are common to any locking methodology, and it neither requires any change in the analog design flow nor any design re-iterations. We demonstrated the locking methodology on two ΣΔ modulators that have a 64-bit and 194-bit programmability so as to be part of a multi-standard RF receiver. They were designed in the 65nm STMicroelectronics technology. Simulations and hardware measurements showed very efficient locking as any invalid key resulted in dramatically degraded performance trade-offs.

4

DIGITAL-TO-ANALOG HARDWARE TROJAN ATTACKS

4.1 INTRODUCTION

To date, the vast majority of HT attacks and defenses have been demonstrated for digital ICs [154]. Few HT designs have been demonstrated in the analog domain, including HTs inserted into the RF front-end aiming at leaking sensitive information via a covert side-channel [102]–[105] and HTs that bring the analog IC into an undesired state or operation mode [16], [106]–[109], as reviewed in Chapter 2.

In general, designing HTs for analog ICs is very challenging since all criteria that make up an effective HT are difficult to meet. First, it is difficult to design stealthy HT since analog signal paths are typically very sensitive and a HT circuitry tapping into them is likely to result in some non-negligible performance degradation, thus it will be difficult for the infected IC to pass testing. Second, it is difficult to design small footprint HTs that will evade optical reverse engineering since analog designs comprise few components or can be clearly divided into sub-blocks or stages each comprising few components. Third, on any analog IC we can extract several information-rich measurements, such that it is unlikely not to be able to find a measurement subspace wherein the statistical fingerprints of HT-infected and HT-free instances are clearly distinguished.

In this chapter, we propose a HT attack for analog ICs with its key characteristic being that it is invisible in the analog domain. This is achieved by exploiting the on-chip test infrastructure that is common to digital and analog cores within the SoC. In particular, the HT trigger mechanism resides in a digital IP core and the payload mechanism resides in the test bus that links all IP blocks in the SoC in a daisy network. The HT is triggered in the dense digital section of the SoC, thus posing a challenge for HT prevention or detection. The HT payload is transferred to the victim analog IP via the test bus and the interface of the analog IP to the test bus. The interface can include DfT blocks, i.e., sensors and actuators, and programmability fabric for the purpose of calibration. The proposed HT is demonstrated on two case studies. The first case study shown with simulation is a LDO regulator where the HT infects it via its DfT interface. The second case study shown with hardware measurements is an RF receiver front-end where the HT infects it via its programmability fabric.

The rest of the chapter is structured as follows. In Section 4.2, we provide an overview of DfT techniques for AMS and RF ICs. In Section 4.3, we provide an overview of calibration schemes present in AMS and RF ICs. In Section 4.4, we review a modern test infrastructure and its use for accessing and controlling DfT structures and the programmability fabric. In Section 4.5, we present the proposed HT attack scenarios. In Sections 4.6 and 4.7, we demonstrate the HT attack on the two case studies. Section 4.8 concludes the chapter.

4.2 DFT FOR AMS AND RF ICS

DfT consists in embedding test structures on-chip with the aim to improve defect coverage and/or facilitate testing, i.e., reduce the test cost by speeding up test application time and/or alleviating the dependence on complex ATE. Built-in Self-test (BIST) is a special form of DfT where the test procedure takes place entirely on-chip without needing to interface the chip to external ATE. In post-manufacturing testing, BIST can offer significant test cost savings at the expense of larger area overhead. For safety-critical or mission-critical applications, it can be reused in the field of operation to perform on-line test in idle times or concurrent error detection.

In general, for AMS and RF ICs, the DfT circuitry can comprise one or more of the following test structures: test access points, digitally-controlled re-configuration schemes, and test instruments, i.e., test stimulus generators, actuators, sensors, checkers, and test response analyzers.

Examples of generic DfT techniques include oscillation-based testing [144], topology modification [146], and symmetry-based BIST [148]. In oscillation-based testing, the circuit is re-configured in a positive feedback loop to force oscillation. Then, the oscillation frequency and amplitude are measured on-chip using as test response analyzer a counter and amplitude detector, respectively. Deviations from the nominal expected oscillation frequency and amplitude point to defect detection. In topology modification, 1-bit controlled Pull-Up (PU) and Pull-Down (PD) transistors are used to tie a node to V_{dd} or ground, respectively, with the aim of re-configuring the circuit such that defects are better exposed. In symmetry-based BIST, invariant properties, i.e., properties that hold true in error-free operation but are violated in the presence of defects, are built and monitored by checkers.

There exist also DfT techniques that are specific to the circuit class, i.e., linear time-invariant circuits, Phase-Locked Loops (PLLs), data converters, RF transceivers, etc., and oftentimes specific to different architectures within a circuit class.

For linear time-invariant circuits, concurrent error detection is achieved by checkers that monitor checksums [155] or create a pseudo-duplicated

response that by default in error-free operation converges to the circuit output [156].

For ADCs, traditional BIST schemes for static linearity testing, i.e., Differential Non-Linearity (DNL) and Integral Non-Linearity (INL), use test stimulus generation performed by ramp generators [157] and a test response analyzer that computes the histogram [158], which could be done also based on reduced-code collection [159]. The requirement for a high-resolution test stimulus can be relaxed by using non-linear stimulus generators combined with advanced post-processing techniques of the converter's output [160]. Traditional BIST schemes for dynamic testing, i.e., SNR, use test stimulus generation performed by sinusoidal signal generators [161], [162] or $\Sigma\Delta$ bitstreams encoding sinusoidal signals [163], and test response analyzers that perform spectral analysis [164] or sine-wave fitting analysis [163].

For PLLs, BIST techniques have been proposed for measuring on-chip the jitter [165], [166]. The PLL response is under-sampled and the count of unstable bits at the clock rising edges is correlated to the high-frequency jitter. Defect-oriented BIST for PLLs has been proposed in [147], where a digital Pseudo-Random Bit Sequence (PRBS) injected in the charge pump perturbs the PLL, and the cross-correlation of the PRBS pattern with the output of the phase/frequency detector is considered for defect detection.

For RF transceivers, a common BIST technique consists in creating a loop-back connection between the transmitter and the receiver, in order to test the whole RF transceiver, e.g., measure the Error Vector Magnitude (EVM), using baseband only signals [167]–[169].

Sensor-based testing is another common DfT technique. Current sensors [170] and amplitude detectors [171] can be used to monitor current or voltage on internal nodes. There exist also non-intrusive sensors that extract information without being electrically connected to the circuit under test, e.g., temperature sensors [145] and process variation-aware sensors [172].

4.3 CALIBRATION OF AMS AND RF ICS

Calibration schemes are oftentimes utilized in AMS and RF ICs with the aim of boosting yield, i.e., by compensating against process variations and non-idealities, and to program different operation modes, e.g., in the case of multi-standard RF transceivers [141], [142], [153].

At a minimum, a calibration scheme utilizes digitally-controlled tuning knobs that act on the circuit performances. Tuning knobs may include bias voltages, current sources, or single tunable components, such as resistors, capacitors, and varactors.

The standard calibration algorithm consists in multiple testing/tuning iterations where in each step the performances are measured and the

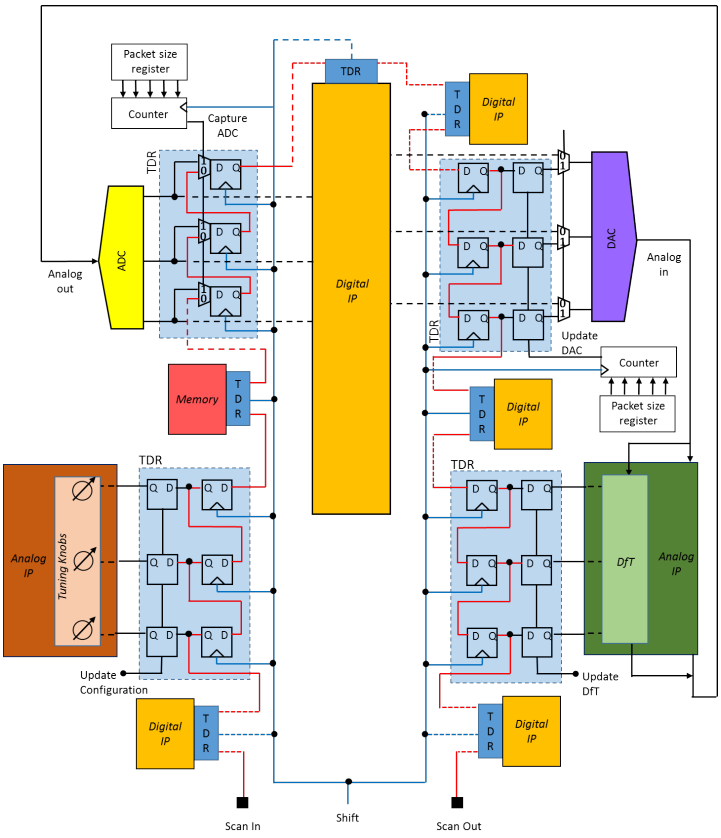


Figure 4.1: Scan access including analog IPs (adapted from [174]).

next best tuning knob setting is dictated based on some optimization algorithm.

The calibration scheme may also utilize on-chip sensors for performance measurement which can speed-up the test cycle and alleviate the dependence on complex ATE. For example, one-shot calibration schemes based on process-variation-aware sensors and machine learning are proposed in [134], [173].

The most advance calibration schemes are fully implemented on-chip rendering the circuit self-healing. These schemes can also be used during the lifetime of the circuit to compensate against aging. They comprise tuning knobs or actuators, sensors for extracting information-rich measurements or directly the performances, and a digital processor engine that maps the outputs of the sensors to tuning knob values and aims at driving the optimization so as to identify a good balance among multiple competing performance goals [136], [137], [140].

4.4 TEST ACCESS AND CONTROL MECHANISM IN SOCS

A modern SoC can embed numerous IP blocks and, in turn, each IP block typically embeds a variety of DfT structures, i.e., test access points

and test instruments, and comes with a calibration scheme that uses a programmability interface. The total number of DfT structures and tuning knobs in a SoC can easily be in the order of hundreds or even thousands, and accessing them separately from primary pins is prohibitive.

To this end, an on-chip test infrastructure is used that connects all the DfT structures and programming interfaces to a common test bus, in order to gain access and control them, manage the test and calibration process, and offload the test response to the ATE for off-chip test response analysis, all-in-all using a limited number of dedicated primary pins [175]. The test bus is interfaced on the chip boundary with the test access port (TAP). Typically, each IP block has its own test program, i.e., IP-level test patterns, and once the SoC design is finalized and the test infrastructure is added, the test programs are re-targeted to the top-level design. The test programs can be simply concatenated, but to achieve higher test time efficiency they can be regrouped enabling concurrency.

The test infrastructure is standardized driven by the needs for test portability and re-use [175]. Portability refers to reusing a test program independently of the position of the target IP block inside the SoC hierarchy, reusing a test program at different steps, i.e., post-manufacturing test, debugging, in-field test towards dependable designs, etc., and reusing a test program independently of the ATE platform. Standardization dissociates the IP block-level DfT structure and programming interface design from test application and calibration operations at SoC-level, i.e., access, control, observe, program, etc., and allows all actors, i.e., IP providers, SoC integrators, test infrastructure providers, to speak the same language. It also enables test automation using CAD tools and leads to significant SoC-level test time reduction.

The latest standard for test infrastructure controllability and observability is the IEEE Std. 1687 [176]. It deals with the great number of DfT structures and connects them serially via programmable segment insertion bits (SIBs) to a reconfigurable scan network (RSN) between the scan in (SI) and scan out (SO) ports. When the SIB of a DfT structure is opened, its test data register (TDR) becomes part of the RSN such that it is accessed from the SI port and its output is streamed to the SO port.

IEEE Std. 1687 was developed with digital ICs in mind. The standard for analog test access is the IEEE Std. 1149.4 [177] and dates from the 1990s. It proposes a test bus paradigm that is still used today, but it requires a minimum of two additional test pins which is too costly and often prohibitive as many designs are pin-limited. To this end, nowadays there is an IEEE working group extending IEEE Std. 1687 [178] to include properties demanded by analog ICs, such as periodic sampling. The envisioned test access standard will be compatible for both analog and digital IPs in a SoC connecting them onto a common test infrastructure.

The principle for connecting analog IPs to the common test infrastructure is proposed in [174]. An example is shown in Fig. 4.1, depicting two analog IPs and several digital IPs connected to a common scan path. For simplicity, the SIBs are not included. To be able to connect analog IPs to the test infrastructure it is required that analog test stimuli and analog test responses are first digitized. This is achieved by using on-chip DACs and ADCs, respectively, which could be shared among several analog IPs if these are tested sequentially and the voltage ranges are consistent. Four types of connections to the scan path are shown in Fig. 4.1 for analog IPs: (a) a DAC connecting an analog node inside the IP or the DfT structure, e.g., for forcing an analog test stimulus; (b) an ADC connecting an analog node inside the IP or the DfT structure, e.g., for monitoring a test response signal; (c) a direct connection to the DfT, e.g., for activating a digitally-controlled re-configuration scheme or embedded test instrument; and (d) a direct connection to the digital tuning knobs used for calibration. Fig. 4.1 shows for simplicity 3-bit data converters and 3-bit words controlling the tuning knobs and DfT structures, but in fact any TDR size can be accommodated into the scan path. It also shows a number of intervening or appended TDRs that connect digital IPs to the scan path, as well as the case where an analog signal inside an analog IP is digitized via the ADC and driven into a digital IP and the case where the output of a digital IP is converted via the DAC to analog and drives an analog input of the analog IP. Finally, Fig. 4.1 shows the three main control signals, namely shift, update, and capture. The shift operation shifts the data serially through the scan path one bit per clock cycle. The update operation latches the data to the input of the ADC, to the input of the DfT structure, or to the programming bits of the tuning knobs. The capture operation offloads a digitized test response into the scan path to be scanned out for subsequent off-chip analysis. For each ADC and DAC, a counter and a packet size register are used that set the periodicity of the TDR update and capture operations. For a more detailed description of the test infrastructure, the interested reader is referred to [174].

4.5 PROPOSED HT ATTACK

4.5.1 *Threat Model*

We assume that the attacker has access to the SoC design and can manipulate a digital IP and the test infrastructure. The attacker also needs to have some minimum knowledge of the victim analog IP so as to design the HT payload. Based on this threat model, the attacker could be the SoC integrator or the foundry.

4.5.2 *Attack Scenario*

It is well-known that the test infrastructure can be a vehicle for attacks. From a general point of view, such attacks can be categorized into external threats and internal threats. External threats consider an unauthorized user that gains control of the chip via the TAP. They include launching several types of scan attacks aiming at stealing secret keys [179], [180], performing reverse-engineering and device cloning [181], performing memory dumping [182], and modifying memory values to attain privilege escalation [183]. Internal threats consider that the attack takes place entirely inside the chip. For example, a third-party malicious IP connecting to the scan network can aim at sniffing confidential data or can act as a tampering IP corrupting the test data of another IP as they are being transported across the scan network [184]. For a more detailed description and a taxonomy of security threats in IEEE test standards, the interested reader is referred to [185].

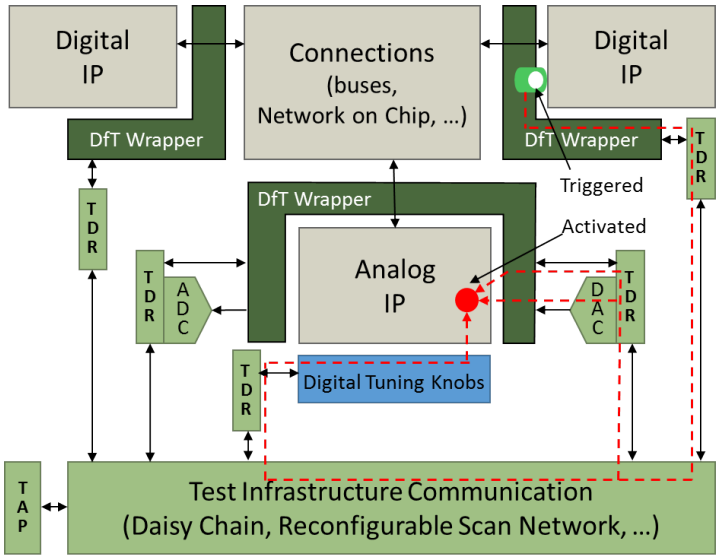


Figure 4.2: HT scenario exploiting the SoC test infrastructure.

In this chapter, we propose a novel HT attack scenario where the test infrastructure is a vehicle for infecting an analog IP. The HT attack, illustrated in Fig. 4.2, exploits the fact that analog and digital IPs coexist in the SoC and are linked together via the shared common scan network, as described in Section 4.4. The key characteristic of the proposed HT attack is that the HT is not hidden inside the analog IP itself, thus neither detection nor prevention is possible in the analog domain. Instead, the HT is well hidden inside a digital IP and the scan network. More specifically, the triggering mechanism is hidden inside a digital IP. Upon activation, the payload is generated and transported to the victim analog IP via the scan network. All the IPs apart from the targeted analog IP can be bypassed thanks to the programmability features of the RSN. The

payload has two parts, a bit that opens a SIB of a DfT structure or the programmability interface of the analog IP, and a digital word that is a malicious test pattern applied to the input of the DfT structure or a malicious tuning knob setting. Essentially, during mission mode, the payload switches the analog IP in test mode or re-configures the analog IP in an undesired operation mode. It can be smartly designed so as to result either in performance degradation or denial-of-service for the analog IP. In fact, numerous malicious test patterns and tuning knob settings can serve this objective, and in practice it will suffice to activate just a single DfT structure controlled by few bits or change just one tuning knob value. In turn, if the analog IP controls other digital IPs, then the operation of the entire SoC can be jeopardized.

In Sections 4.6 and 4.7, we present two examples of how this scenario might play out in a SoC. In the first example, the HT infects an LDO via its DfT interface. Although the LDO is the direct victim of the HT, given that the LDO supplies one or more digital IPs inside the SoC, the HT infects implicitly other digital IPs too. In the second example, the HT infects an RF receiver via its programmability interface. In both examples, we design HT payloads that lead to performance degradation or denial-of-service.

4.5.3 *HT Design*

The proposed HT attack scenario can make use of any triggering mechanism, i.e., combinational, sequential, or more complex triggering mechanisms, as discussed in section 4.1. Several benchmark triggering mechanisms can be found in Trust-Hub [154]. For this reason, herein we do not cover in more detail this aspect of the HT design, and we will focus only on the payload mechanism aspects, by proposing several different examples.

The general payload mechanism consists in a malicious digital pattern applied at the interface of the analog IP during normal operation. In the case of infection via the DfT interface, the malicious pattern corresponds to an incorrect DfT pattern, either semi-activating or fully-activating the DfT structure, thus forcing the analog IP into either an incorrect test mode or the correct test mode. In the case of infection via the programming interface, the malicious pattern forces a malicious programming setting, either one that corresponds to a different operation mode or one that is invalid, i.e., not corresponding to any documented usage.

The result of unexpectedly activating the DfT or modifying the programming during normal operation can be either performance degradation or denial-of-service. The attacker can design the malicious pattern according to the objective, which can be simply generated by flipping bits in the DfT pattern that disables DfT or flipping bits of a given pro-

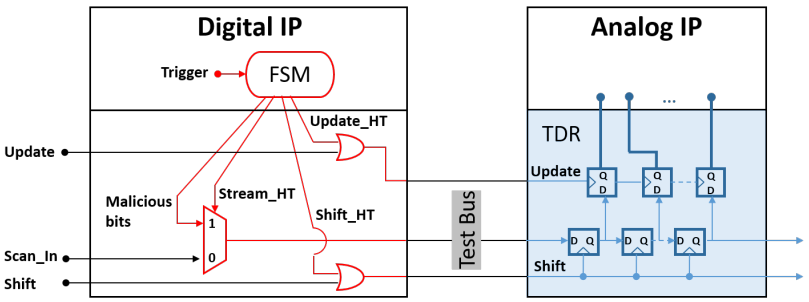


Figure 4.3: Payload mechanism based on transporting the malicious bit pattern to the victim analog IP.

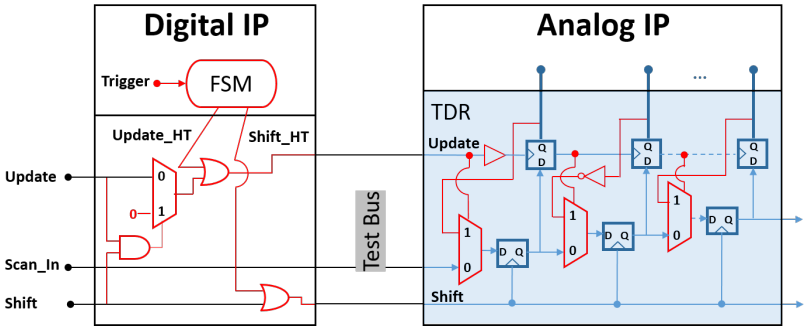


Figure 4.4: Payload mechanism based on updating the TDR of the victim analog IP.

gramming setting. Since the operation of analog circuits is very sensitive, flipping just one bit can lead to the desired effect for the attacker.

Fig. 4.3 shows a payload mechanism that generates the intent malicious pattern inside the digital IP and then transports it to the DfT or programming interface of the victim analog IP via the test bus. The HT design is shown in red color. Upon activation of the trigger, the malicious pattern is generated by a FSM, which also controls its transporting via the test bus. Signal Stream_HT is set to 1 to toggle the multiplexer and feed the malicious pattern into the test bus. Signal Shift_HT is also set to 1 to shift the malicious pattern downwards via the test bus for a number of clock cycles required to reach the victim analog IP. Then, signals Stream_HT and Shift_HT return to 0, and signal Update_HT is set to 1 to update the parallel data register of the TDR of the victim IP and force the malicious bit pattern.

Fig. 4.4 shows a payload mechanism that refreshes the parallel data register of the TDR of the victim IP while flipping a select set of bits to generate the malicious pattern. Prior to activation of the trigger, the parallel data register stores either the pattern that disables the DfT of the analog IP or a programming setting corresponding to a specific operation mode of the analog IP. Upon activation of the trigger, the FSM sets signals Update_HT and Shift_HT simultaneously to 1, which is not a valid

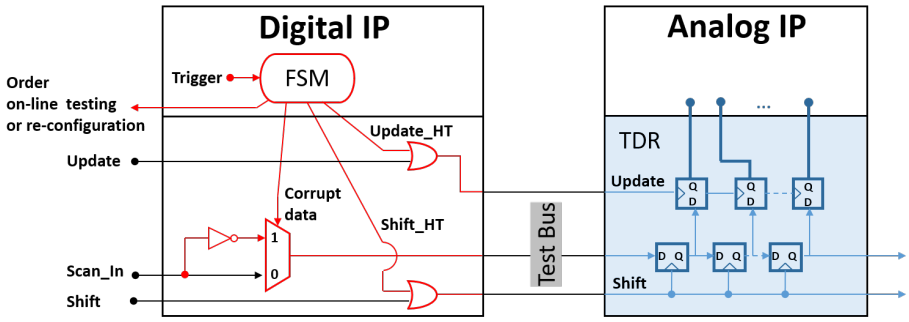


Figure 4.5: Payload mechanism based on requesting on-line testing or re-configuration and subsequently corrupting the transported data.

condition during normal test mode. This generates the desired malicious pattern from the pattern stored in the parallel data register using inverters in appropriate positions, writes this malicious pattern into the serial shift register that is a segment of the scan chain, and updates the parallel data register writing into it the malicious pattern. A buffer is used in the update path to ensure that the write operation of the malicious pattern into the scan chain will be completed before the parallel data register is finally updated. Note that the pattern existing in the scan chain prior to the HT activation is a “don’t care" pattern since the test bus is in idle mode during the payload application. Thus, altering this pattern is not an issue as it would have been eventually streamed off-chip and dumped.

Fig. 4.5 shows a third possibility where the payload mechanism requests on-line testing or re-configuration of the victim analog IP and corrupts the transported data as they pass by the scan network segment of the malicious digital IP. In this case, the attacker is aware of the geometry of the scan network and knows exactly the number of clock cycles needed for the data to reach the digital IP. At this point, the digital IP orders corrupting a select subset of bits in the data as they are being shifted to reach the victim analog IP.

A fourth scenario of payload mechanism could be the corruption of a test pattern or a programming setting stored in the memory. The memory re-write operation takes place upon activation of the HT, but the payload is applied later when the analog IP is subject to on-line testing or a re-configuration is demanded by the application.

4.5.4 *Discussion on Countermeasures*

The HT resides outside the analog IP and the payload is naturally applied to the analog IP via its DfT or programming interface. Thus, the HT is totally transparent to the analog IP and cannot be prevented or detected in the analog domain. Countermeasures against the proposed HT attack can only be implemented in the digital domain.

Countermeasures in the digital domain aiming at HT insertion prevention or post-silicon detection were discussed in Chapter 1 and are generally applicable. However, their effectiveness is questionable since the proposed HT is both stealthy and with a tiny footprint. Note that all HT designs proposed in Section 4.5.3 are transparent in normal test mode. Only the HT design in Fig. 4.4 can be detected with a test command that simultaneously sets the update and shift signals to 1, but this test command can be easily suppressed by the attacker, as shown in Fig. 4.4. More specifically, this test command sets the output of the AND gate to 1, which changes the update signal to 0. In this way, when this command is applied neither the TDR of the victim analog IP is updated nor the data in the scan chain are corrupted. This is the expected behavior in HT-free operation, thus the HT goes undetected.

On the other hand, there have been many recent works aiming at improving the trust in the test infrastructure, defending against the external and internal threats described in Section 4.5.2. A comprehensive overview and classification of such countermeasures can be found in [185]. Possible countermeasures include: (a) test infrastructure access authentication, e.g., by inserting a password inside the TAP controller [186] or implementing a challenge-response protocol [187], [188]; (b) scan network access authentication, e.g., by locking the SIB [189], implementing a challenge-response protocol [190], or obfuscating the geometry of the RSN structure [189], [191]; (c) privileged-based access restriction [188], which extends the access authentication techniques by assigning different privileges to the users according to the trust level they have; (d) assure data confidentiality, e.g., by test data encryption [184], [191] or isolating any untrusted instruments when confidential data are being shifted through the scan network [188]; (e) bidirectional IP block authentication, e.g., implementing a challenge-response protocol at the chip-level [184]; (f) assure data integrity [184], i.e., assure that the test patterns have not been modified during their transportation across the scan network; (g) on-line detection aiming at detecting the execution of the attacks while they are running, e.g., by setting rules to verify the test pattern compliance to a legitimate behavior [191]. The proposed HT attack is an internal threat and all the proposed HT designs are essentially tampering mechanisms corrupting test or programmability patterns. Countermeasures (a)-(c) defend against external threats only, thus the proposed HT attack can evade or bypass them, given also its insertion phase. Countermeasures (d)-(g) can defend against internal threats and the protection they can offer against the proposed HT attack should be evaluated. More specifically, data encryption cannot protect against the proposed HT designs since decrypting at the analog IP interface a randomly generated test or programmability pattern, or corrupting an encrypted test or programmability pattern by flipping many bits at

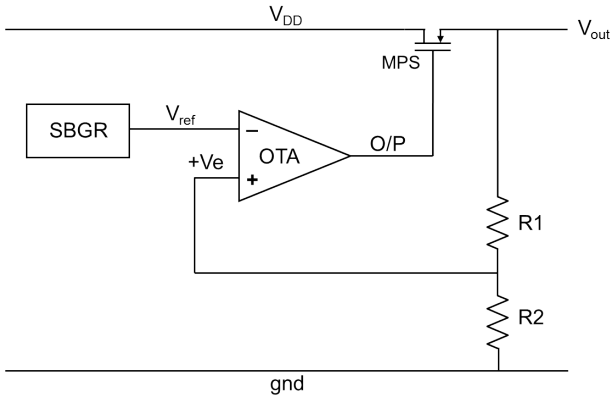


Figure 4.6: Block-level schematic of the LDO.

random and subsequently applying it to the analog IP, will still infect the analog IP. The HT effect, however, will not be controllable by the attacker, and probably the HT will cause dramatic performance degradation or complete malfunction. Bidirectional IP block authentication and assuring data integrity can defend against the proposed HT attack, but they require significant extra on-chip resources, thus the overhead the defender has to pay is significant. On-line detection at the DfT or programming interface of the analog IP can be bypassed at the phase where the proposed HT is being inserted.

4.6 CASE STUDY: LDO

4.6.1 *LDO Regulator Design*

The LDO is one of the most popular power management systems to supply the sub-blocks of a SoC. It is a perfect target of a HT as the infection will spread to other IPs inside the SoC. We designed an LDO in the 65nm technology by STMicroelectronics using the free open-source OCEANE tool [192]. Its block-level schematic is shown in Fig. 4.6. It consists of a sub-band gap reference voltage generator (SBGR), an error amplifier implemented with an operational transconductance amplifier (OTA), a power p-MOS transistor, and a feedback resistor network. The error amplifier monitors a fraction V_e of the LDO output voltage V_{out} through the resistor feedback network and compares it with the output voltage V_{ref} of the SBGR. If V_e is higher (lower) than V_{ref}, then the error amplifier drives the gate of the power transistor to decrease (increase) its output voltage so as to maintain a constant V_{out}. Figs. 4.7 and 4.8 show the schematics of the OTA and SBGR. Fig. 4.9 shows the schematic of the self-biased operational transconductance amplifier (SOTA) inside the SBGR.

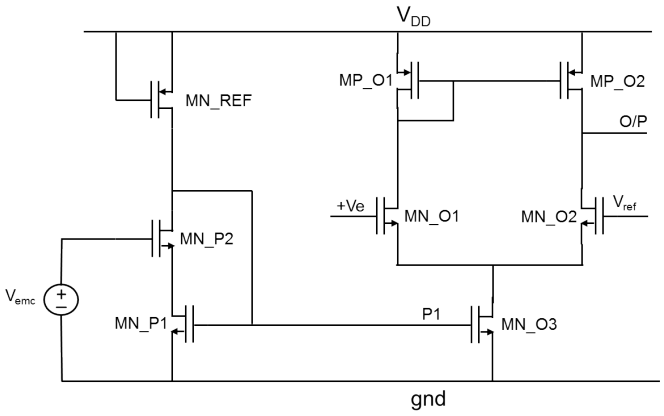


Figure 4.7: Schematic of the error amplifier within the LDO implemented with an OTA.

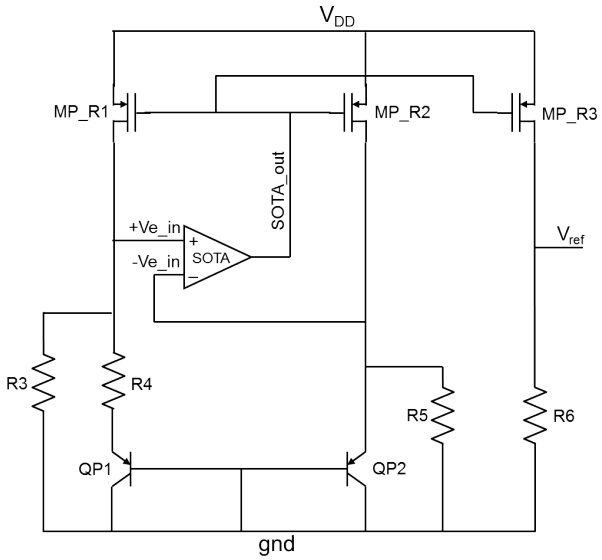


Figure 4.8: Schematic of SBGR generator.

The green curves in Figs. 4.10-4.12 show the nominal LDO performance in the HT-free scenario. Specifically, Fig. 4.10 shows the LDO output variation as a function of power supply voltage variations at 27°C. As it can be seen, V_{out} shows a 33.4mV variation when V_{dd} varies from 1.4V to 3V. Fig. 4.11 shows the LDO output dependence on temperature variations for a V_{dd} equal to 1.5V. As it can be seen, V_{out} shows a 10mV variation when temperature varies from -55°C to 125°C. Fig. 4.12 shows the transient response of the LDO for a variation of load current from 50mA to 0mA and then from 0mA to 50mA, which corresponds to removing the load and then adding it back. The maximum overshoot is 44.9mV and settles after 875ns, while the maximum undershoot is 53.2mV and settles after 800ns.

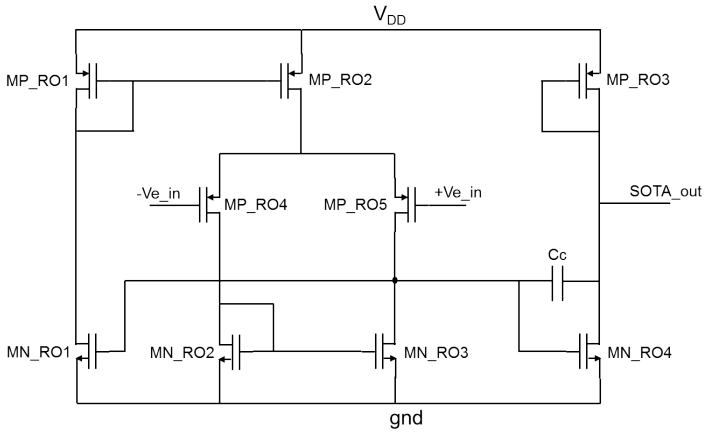


Figure 4.9: Schematic of SOTA.

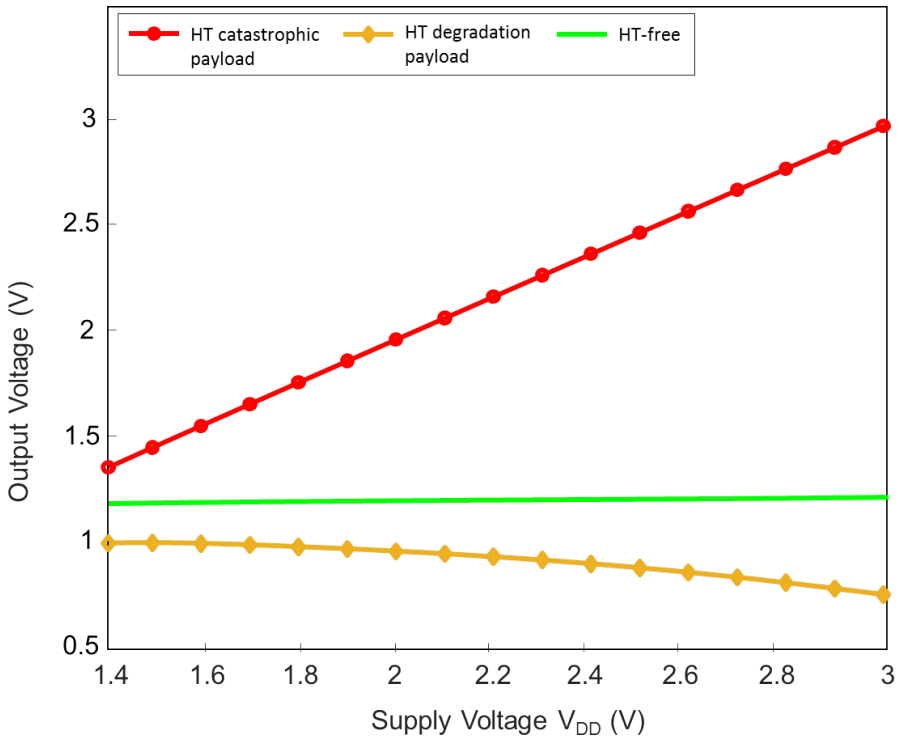


Figure 4.10: LDO output variation as a function of power supply variation.

4.6.2 *DfT*

We use a generic defect-oriented DfT concept for low-frequency analog ICs proposed in [146]. The DfT principle is based on topology modification (or re-configuration) enabled by the addition of PD and PU transistors. A PD transistor connects a circuit node to ground, while a PU transistor connects a circuit node to the power supply. PD and PU transistors are activated by applying a logic 1 and 0 at their gates, respectively. If N PD and PU transistors are added, then the circuit can be configured into 2^N topologies, including the original one where all PD

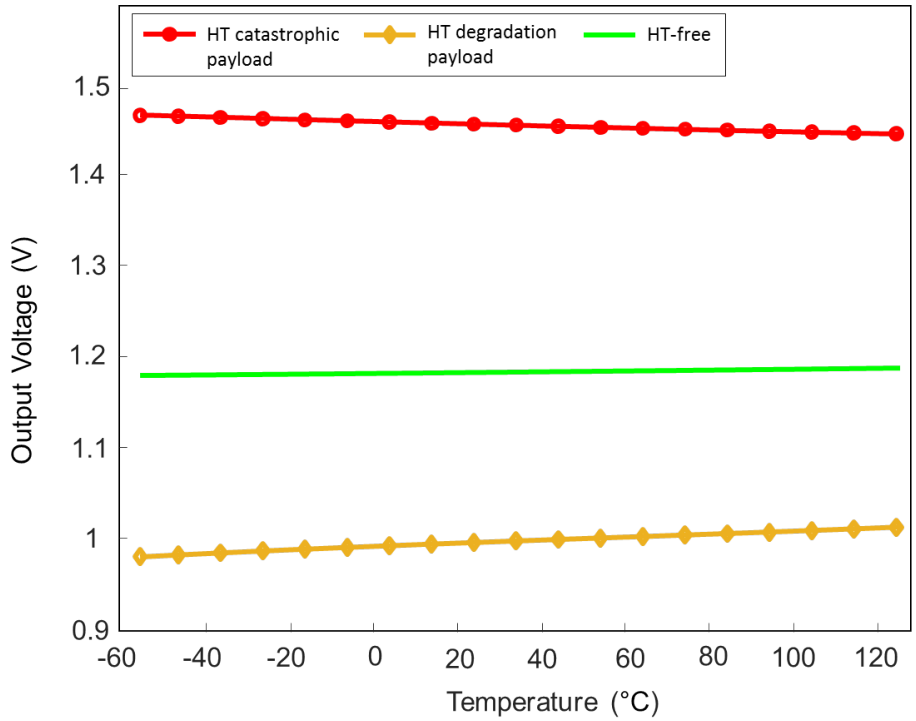


Figure 4.11: LDO output variation as a function of temperature variation.

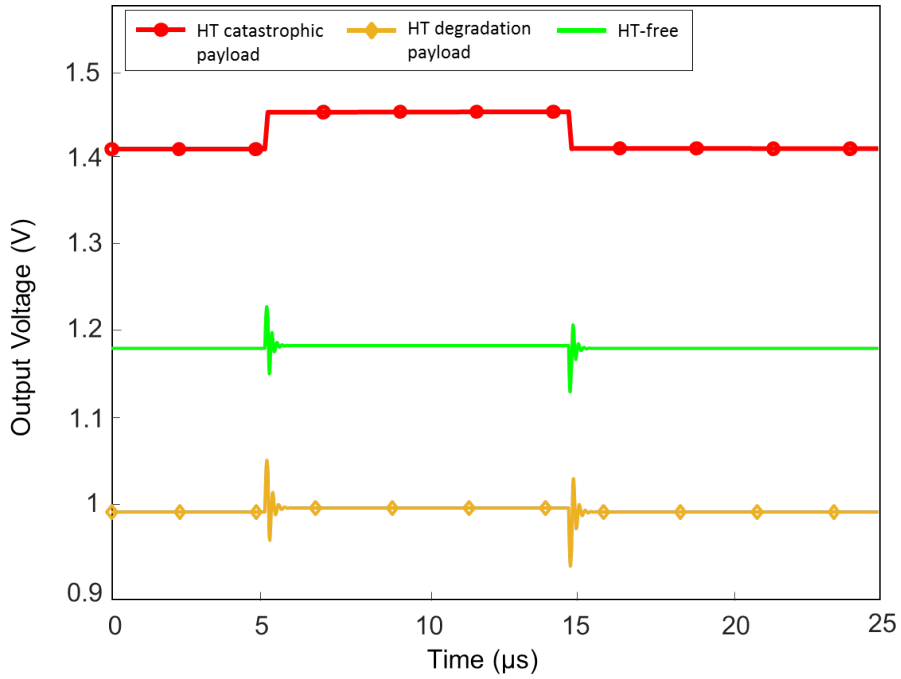


Figure 4.12: Transient response of the LDO for a variation of load current.

and PU transistors are deactivated. The underlying principle is that by these re-configurations we are able to expose the presence of additional defects that are undetectable in the original topology.

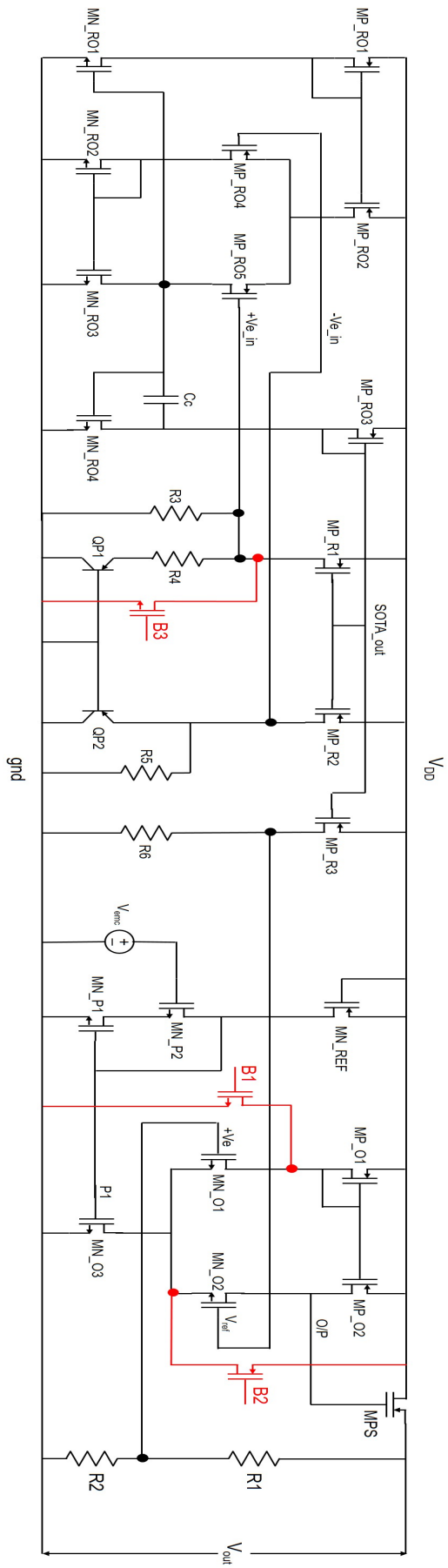


Figure 4.13: LDO with DfT. The added PD and PU transistors to enable topology modifications are shown in red color.

A DC test is used for the LDO. In particular, the LDO is self-activated and its output is used as the test output. In the defect-free case, for each test configuration, a different nominal test output value $V_{test,j}$ may be observed, where j denotes the configuration number. To account for process variations and avoid yield loss, we consider a tolerance window $\pm k * V_{test,j}$, $k > 0$. For the purpose of our experiment, we set $k = 0.1$.

The defect simulation is performed at transistor-level and in an automated workflow using the Tessent®DefectSim tool by Mentor®, A Siemens Business [149]. We cycle through all configurations and for each configuration defects are injected one by one. If $V_{test,j}$ is outside the tolerance window then the defect is deemed detectable by the test configuration.

We use the default defect model of the tool [149]. In particular, for MOS transistors we use only gate open and drain-to-source short defects. Similarly, for bipolar transistors, we consider base open and collector-emitter short defects. We consider the default short resistance of 10 ohms. Regarding opens, a weak pull-up or pull-down is assigned to each open defect to account for the facts that an ideal open does not exist and, besides, it cannot be handled by a SPICE simulator [149]. For passive elements, i.e., resistors and capacitors, we consider ±50% variations. In total, the defect model contains 60 defects. Furthermore, any of the N added PU or PD transistors could also contain defects, which increases the number of defects by 2N. We consider the absolute defect coverage defined as the percentage of detected defects.

A defect coverage of 80% is reached using only the original topology. We applied the DfT idea considering that in a given re-configuration only one PU or PD transistor can be enabled. The LDO has 14 nodes in total, thus the number of possible re-configurations is 28. We performed an exhaustive search and we identified 3 nodes where PD and PU transistors can be added to result in a defect coverage of 100%. The complete LDO schematic with the embedded DfT structure is shown in Fig. 4.13. One PD and one PU transistor, labelled by B1 and B2, respectively, are used inside the error amplifier, and one PD transistor, labeled by B3, is used inside the SBGR. The DfT is disabled with the pattern [B1,B2,B3]=010, while the patterns for enabling the three test configurations are [B1,B2,B3]=110, [B1,B2,B3]=000, and [B1,B2,B3]=011.

4.6.3 *HT Payload Design*

An interesting aspect of this DfT approach is that the DfT interface inside the LDO, i.e., transistors B1, B2, and B3, has a digital word input and can be connected directly to the scan network without using a DAC. Another interesting aspect specific to the LDO is that the LDO is self-driven without needing to specify an analog test input.

The HT payload consists in applying a malicious DfT pattern during normal operation. We identified two such DfT patterns that result in degradation of the LDO performance and to complete malfunction, respectively. In turn, the HT can affect indirectly all digital IPs inside the SoC that are supplied by the LDO, thus resulting in degradation or complete malfunction of a large part or even the entire SoC.

In particular, applying the DfT pattern [B1,B2,B3]=110 results in shifting the LDO output by about 15% and also results in small variation of the LDO output for temperature and V_{dd} variations, as shown by the orange curves in Figs. 4.10-4.12. In more detail, enabling B1 results in zero gate voltage for transistors MP_O1 and MP_O2 which increases the current flowing through them. However, the sum of the currents stays fixed since it equals the current flowing through MN_O3 which is fixed. As the voltages of all terminals of MP_O1 are fixed, it turns out that the current through MP_O2 reduces, which is enabled by the increase of the drain voltage of MP_O2. This voltage drives the gate of the power p-MOS transistor MPS and, thereby, the current that flows through MPS reduces, which reduces the LDO output. In turn, this reduces the voltage on the $+V_e$ terminal which points to reduction of the source voltage of MN_O1 since the current flowing through MP_O1 is fixed. This feedback effect reduces the drain voltage of MN_O2 which is the gate voltage of MPS. In the end, as it can be seen from Figs. 4.10-4.12, the LDO output settles at a slightly lower value of around 1V.

Applying the DfT pattern [B1,B2,B3]=011 results in a catastrophic effect in the operation of the LDO, as shown by the red curves in Figs. 4.10-4.12. In more detail, setting B3=1 connects the $+V_{e\_in}$ terminal of the SOTA to ground. The result is that V_{ref} follows V_{dd} instead of being stabilized at 0.7V. Since the output of the LDO follows V_{ref}, it shows a linear relationship with V_{dd} acting like a non-stabilized power supply. In addition, once the load is removed the response overshoots and never settles back unless the load is added again.

4.7 CASE STUDY: RF RECEIVER

4.7.1 *RF Receiver Programmable Architecture*

Our second case study is a programmable highly-digitized multi-standard RF receiver whose high-level architecture is illustrated in Fig. 3.6. A BP RF ΣΔ ADC is used to directly convert the RF signal at the output of the low noise amplifier (LNA) to the digital domain. The signal is then down-converted to DC by a digital mixer and filtered using a digital decimation filter. The RF receiver is designed with programmable sub-blocks such that it can serve for establishing communication using several standards within the frequency range from 1.7 GHz to 2.8 GHz, including Bluetooth,

ZigBee, WiFi 802.11b, etc. The programmability aims at adapting the specifications in terms of sensitivity, center frequency, bandwidth, and resolution, according to the target standard. The programmability is enabled by judiciously inserting digitally-controlled tuning knobs into the different sub-blocks.

The designer uses a complex calibration algorithm to find appropriate programming settings per standard, as detailed in Section 3.5.2.

The calibration is performed following testing/tuning iterations towards optimising the performance trade-off per standard and per chip. The programming setting visited in each iteration is driven to the programmability interface via the scan network. For a given chip, once the calibration has been completed, the final matrix of the programming setting per standard is stored in an on-chip memory. During application, when the programming setting is to be updated, the new programming setting is called from the associated memory address and driven to the RF receiver via the scan network where it is latched into the register of the programmability interface.

In our example, we infect the RF receiver via the critical modulator block of the BP RF ΣΔ ADC. We rely on a recent design in the 65nm technology by STMicroelectronics [153] whose block-level architecture is illustrated in Fig. 3.16. The functionality of the modulator is adjusted using a 194-bit programming word. Fig. 3.16 shows the number of bits of the programming word controlling the operation of each sub-block. The programming aims at meeting the specifications of the target standard while at the same time compensating for process variations and non-idealities so as to improve the overall performance trade-off.

Our experiment is conducted using hardware measurements on the actual fabricated chip. Fig. 3.18 shows a photo of the test bench used for measuring the performance of the modulator under the HT-free and different HT-infection scenarios.

For the purpose of our experiment, without loss of generality, we consider that the RF receiver operates with center frequency 2.77 GHz. The four main performances are plotted in Figs. 4.14, 4.15, 4.16, and 4.17. The green curves correspond to the nominal HT-free operation. More specifically, Fig. 4.14 shows the PSD for an input power of -14 dBm. As it can be seen, the modulator has a nominal SNR of 60 dB. Fig. 4.15 shows the SNR at different input power values with a step of 1 dBm defining the dynamic range (DR) of the modulator. Fig. 4.16 shows the SFDR measured by applying two tones at the input with the same power and frequency difference of 2 MHz. As it can be seen, the modulator has a nominal SFDR of 51.39 dB. Finally, Fig. 4.17 shows the output fundamental power and the IM_3 product versus the input power, from which the IIP_3 can be determined. The modulator has a nominal IIP3 of 8 dB.

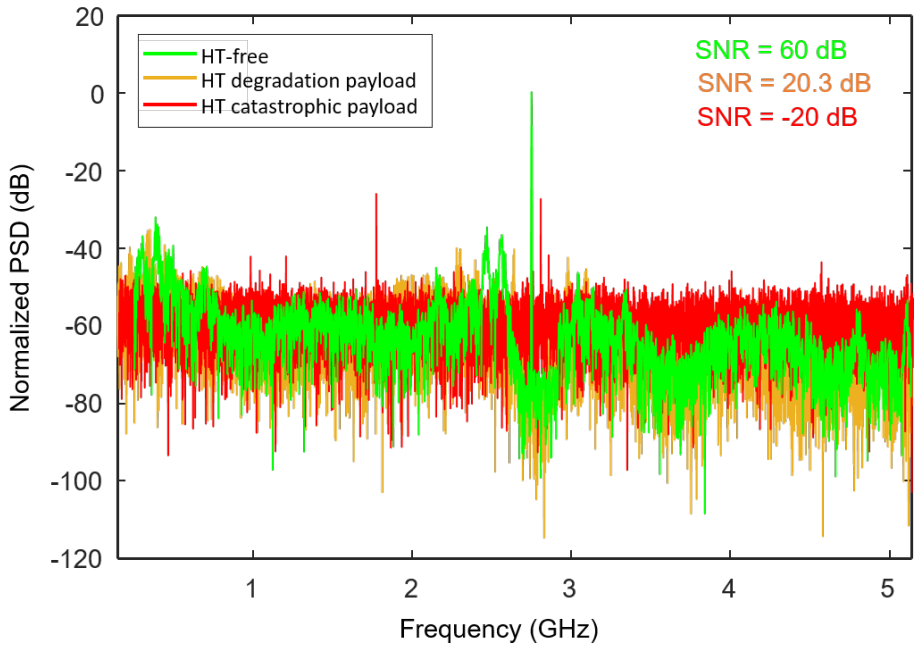


Figure 4.14: PSD under HT-free and HT-infected operation.

4.7.2 *HT Payload Design*

The HT payload consists in unexpectedly altering the programming setting during normal operation. The issue here is that the final programming settings per operation mode and per chip are defined during testing time, while the HT is planted at an earlier phase. In other words, the matrix of final programming settings is unknown to the attacker and, in any case, will change from one chip to another. The attacker can still attain a controllable HT effect. The reason is that the programming setting is divided into segments each controlling a different sub-block, as shown in Fig. 3.16. Each segment of the programming setting can have one of two roles, namely either calibrating against process variations or setting the desired operation mode. Thus, for the HT to cause complete malfunction, i.e., denial-of-service, it suffices that it randomly flips bits in segments of the programming setting that are used for setting the operation mode. Accordingly, for the HT to cause performance degradation, it suffices that it randomly flips bits in segments of the programming setting that are used to calibrate against process variations. In fact, as we demonstrate below, in both scenarios it suffices that the HT flips just one bit in the nominal programming setting, that is, the malicious and nominal programming settings can have a Hamming distance of 1 which facilitates the HT payload design.

Returning to our case study, a candidate block where the HT can act to incite performance degradation is the negative transconductance -Gm in LC tank 1. The programmability of -Gm is responsible for improving the

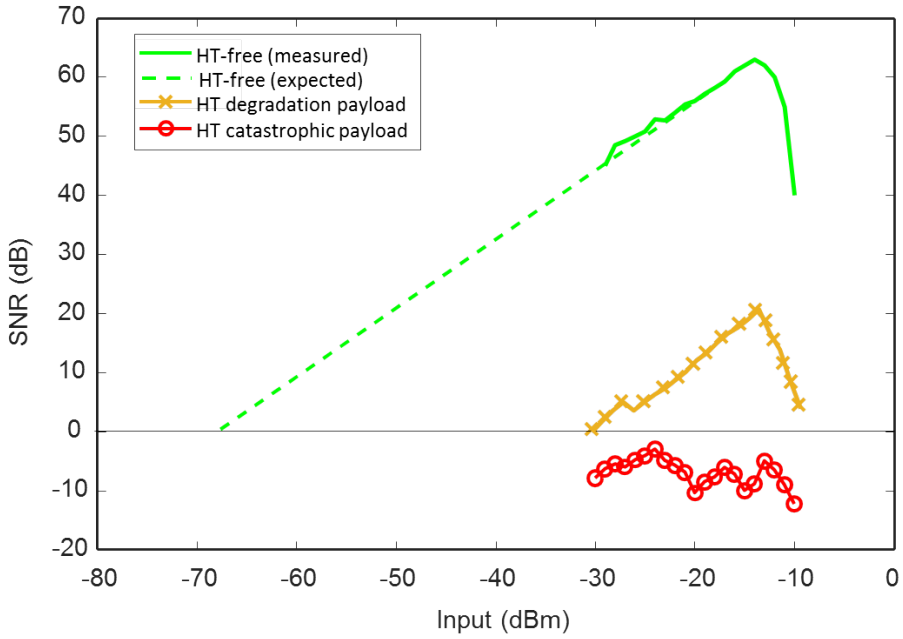


Figure 4.15: Dynamic range under HT-free and HT-infected operation.

quality factor of the LC filter. Flipping one single bit in the programmability of -Gm will inevitably decrease the quality factor, thus untuning the RF receiver performance and degrading the SNR. The orange curves in Figs. 4.14-4.17 show the HT-infected performances in this scenario. As it can be seen, SNR, SFDR, and IIP3 are decreased by about 66.71%, 60.69%, and 212.5%, respectively, and the DR is also decreased.

A candidate block where the HT can act to incite complete malfunction is the tunable delay block in the feedback loop. The tunable delay block is responsible for controlling the center frequency of the noise shaping. It consists of delay elements and the programming connects or disconnects them so as to control the delay time. Flipping one bit in its programmming will inevitably set the RF receiver in another operation mode, most likely in an undocumented and invalid operation mode, thus leading to complete malfunction. The red curves in Figs. 4.14-4.17 show the HT-infected performances in this scenario. As it can be seen, there is no noise shaping and the signal now is buried under the noise floor.

As a final note, the measurements carried out in Chapter 3 and illustrated in Figs. 3.19-3.23 were the result of 5K randomly generated keys, whereas here a single programming bit, i.e. key-bit, is flipped in a deterministic manner to degrade the circuit's performance or to cause malfunction. More specifically, we selected a certain sub-block to attack so as to degrade the circuit's performance or to cause malfunction. We flip just one bit in the configuration word of the targeted sub-block and we measure the effect on the circuit's performance.

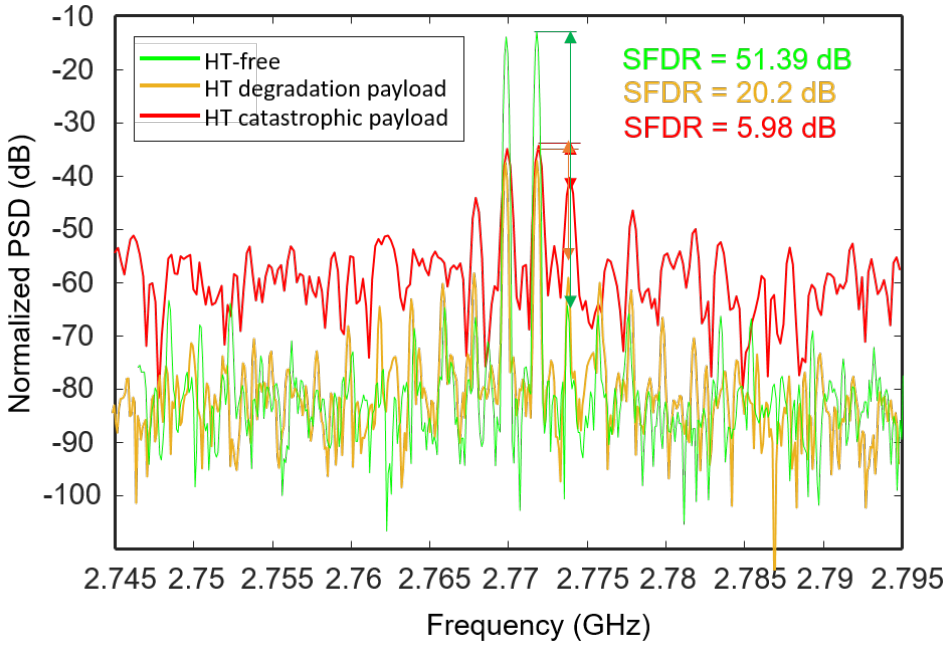


Figure 4.16: SFDR under HT-free and HT-infected operation.

4.8 CONCLUSION

We proposed a novel HT attack scenario targeting infecting analog IPs embedded in a SoC. The HT lies in the dense digital circuitry and transports its payload to the victim analog IP via the test bus. The payload consists of a corrupted DfT pattern or programmability setting and is applied to the analog IP via the DfT circuitry or programmability fabric that are accessed via the test bus. We proposed four different HT designs, namely generating the corrupted data inside a digital IP, refreshing the TDR of the analog IP to corrupt the stored data, ordering on-line testing of re-configuration and corrupting the data as they are being transported to the analog IP, and re-writing data in the memory aiming at infecting the analog IP in the next on-line test cycle or re-configuration. The proposed HT attack was demonstrated on two case studies, namely an LDO regulator and an RF receiver. In the LDO case study, we considered an effective DfT approach and we derived DfT patterns that can lead to performance degradation or denial-of-service. In the RF receiver case study, we demonstrated with hardware measurements that the infection can succeed by flipping only a select bit in the programming of the ADC that digitizes the received signal. The key characteristic of the proposed HT is that it is totally invisible in the analog domain. It is hidden in its entirety into the digital part of the SoC having a very small footprint and evades detection during test time.

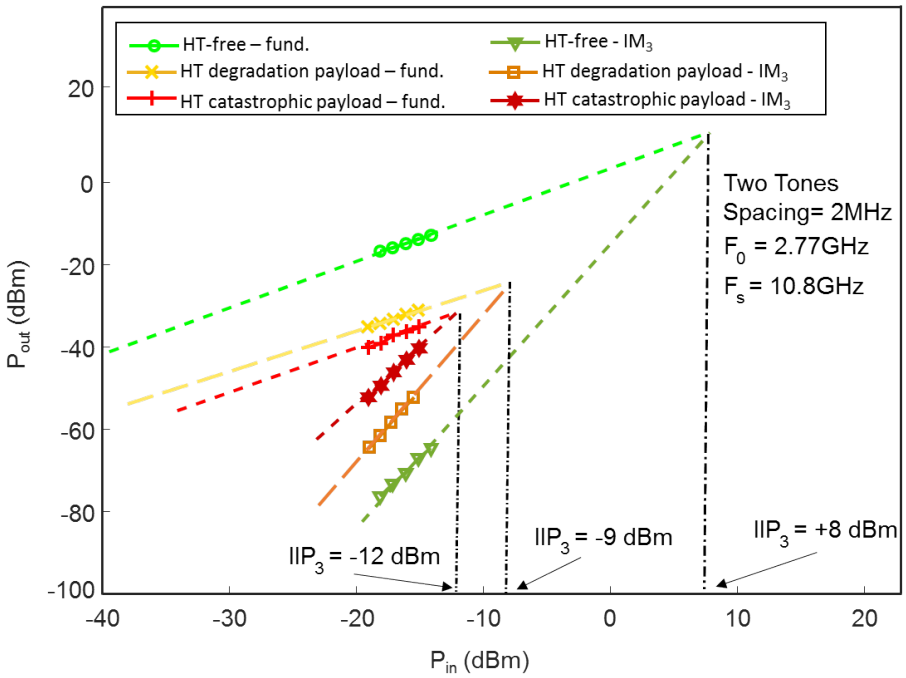


Figure 4.17: IIP3 under HT-free and HT-infected operation.

5

NEURON-PUF: PHYSICAL UNCLONABLE FUNCTION BASED ON A SINGLE SPIKING NEURON

5.1 INTRODUCTION

As discussed in Chapter 1, PUFs are a class of hardware security primitives that find several applications. In this chapter, we propose a novel PUF class, called *neuron-PUF*, that uses a single spiking neuron as the source of entropy. A spiking neuron produces a pulse wave whose pre-set duty cycle and periodicity will depend on random process variations. This analog signature is processed by a key extractor to generate the digital key. A stability-enhancement technique based on masking is used to drop native unstable bits identified during testing. Neuron-PUF uses a composite and multidimensional challenge which makes it a candidate for implementing a strong PUF.

Previously proposed PUFs use space redundancy, i.e., multiple PUF cells, to generate the PUF response as discussed in Chapter 2. In contrast, the proposed neuron-PUF uses a single PUF cell based on a single spiking neuron and extracts an arbitrarily long PUF response in a serialized fashion using temporal redundancy. Therefore, the proposed neuron-PUF significantly reduces area and power overheads.

The rest of the chapter is structured as follows. In Section 5.2, we discuss spiking neurons focusing on the specific spiking neuron that we employed. In Section 5.3, we present the neuron-PUF architecture. In Section 5.4, we present the PUF quality metrics used to characterize the proposed PUF. In Section 5.5, we present the results. Section 5.6 concludes the chapter.

5.2 SPIKING NEURONS

Spiking neurons are biologically-inspired neuron models that serve as the fundamental building block of neuromorphic systems. Neuromorphic systems aim at emulating the brain functionality for efficiently solving cognitive tasks, i.e., visual recognition and motion control. A large number of neuromorphic systems have been demonstrated in the recent years [193]–[196]. Spiking Neural Networks (SNNs) constitute the third generation of neural networks aiming at bridging the gap between the biological brain and machine learning in terms of recognition speed and power consumption [197], [198].

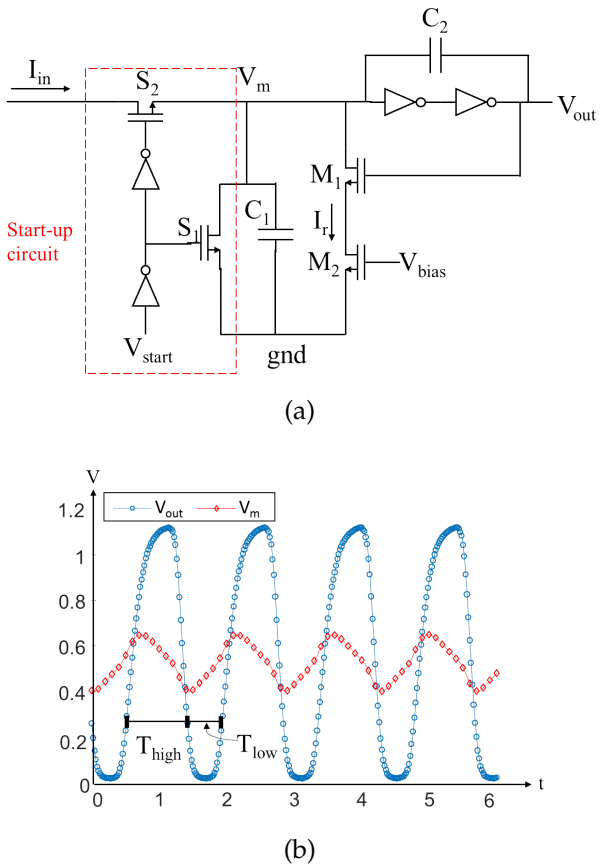


Figure 5.1: The axon hillock circuit: (a) schematic; (b) transient response.

There are several spiking neuron models of different complexities, ranging from biophysical models to phenomenological models, such as the most popular Integrate & Fire (I&F) model, which can have a hardware-friendly implementation and can still be designed to reproduce a large variety of spiking firing patterns observed in biological neurons [199]. An I&F spiking neuron receives and integrates input spikes from neurons in the previous layer via the synaptic connections. If its state reaches a certain threshold, then it fires a spike of its own that propagates to the neurons in the next layer via the synaptic connections. In addition, it resets its state so that it can fire again.

There are several hardware implementations of I&F spiking neurons of different complexities [199]. For the purpose of this work, as a proof of concept, we use the axon hillock circuit proposed in [200], whose schematic is shown in Fig. 5.1(a).

When $V_{start} = 0$, the input is disconnected and the capacitor C_1 is discharged to o. When $V_{start} = V_{dd}$, the switch S_2 turns on and the switch S_1 turns off. The input current I_{in} starts charging the capacitor C_1, which models the membrane capacitance of the neuron's cell. When the membrane voltage V_m reaches the threshold V_ℓ of the first inverter, then the inverters switch state and the output fires a pulse, i.e., V_{out} goes to V_{dd}. The capacitor C_2 models the inherent positive feedback of the

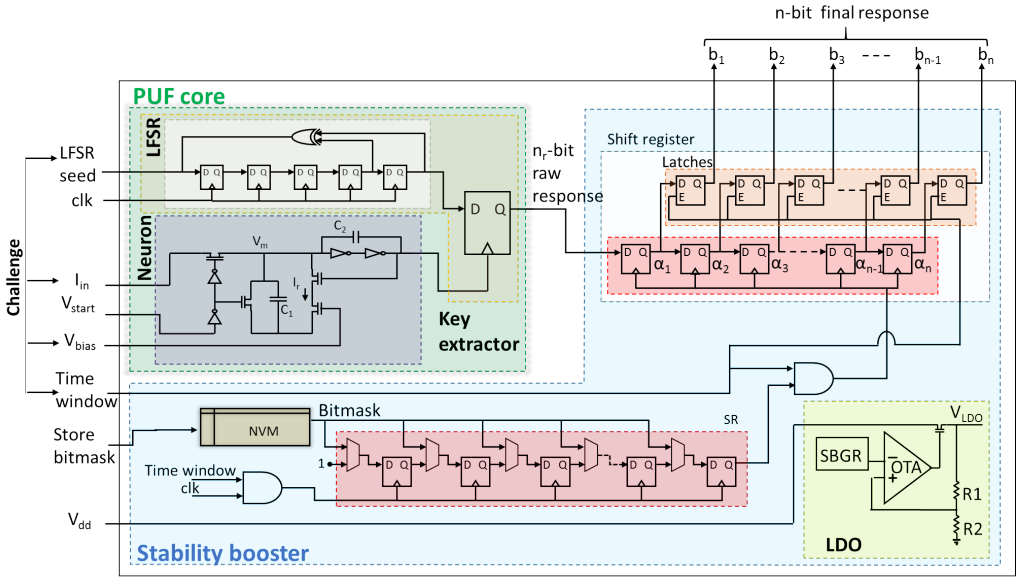


Figure 5.2: Neuron-PUF architecture.

neuron cell. As V_{out} increases, V_m increases due to the positive feedback even faster. Setting a V_{bias} higher than the threshold voltage of transistor M_2 introduces a leaking mechanism that helps the neuron to reset itself once it has fired a spike. In particular, when V_{out} increases to V_{dd}, C_1 starts discharging to 0 via the current I_r flowing through transistors M_1 and M_2, until a point where V_m drops below the inverter's threshold. At this point, the output voltage goes rapidly to ground and the pulse is terminated. Then, for a constant input current, the neuron starts again the integration process to fire another pulse. A simulation of this circuit is shown in Fig. 5.1(b).

It can be shown that the spike duration is given by

$$T_{high} \simeq \frac{C_2 \times V_{dd}}{I_r(V_{bias}) - I_{in}}, \tag{5.1}$$

and the time between two consecutive spikes is given by

$$T_{low} \simeq \frac{C_1 + C_2}{I_{in}} V_\ell. \tag{5.2}$$

5.3 NEURON-PUF ARCHITECTURE

The architecture of the proposed neuron-PUF is illustrated in Fig. 5.2. It consists of the PUF core, which takes the challenge as input and produces the PUF raw response, and the stability booster, which enhances the stability of the PUF final response.

The PUF core consists of the spiking neuron, which generates an analog signature, and the key extractor, which processes the analog signature to generate the digital PUF response. In more detail, the PUF starts generating its response when V_{start} goes high. The spiking neuron integrates a current I_{in}, and produces at its output a spiking pattern in the form of a pulse wave, whose duty cycle and periodicity are defined by T_{high} and T_{low} in Eqs. (5.1)-(5.2). The key extractor consists of a linear-feedback shift register (LFSR) and an edge triggered flip-flop. The flip-flop receives its input from the LFSR and is clocked with the pulse wave, producing a serialized raw digital PUF response in the form of a bitstring.

Since the sampling is non-coherent, the raw response can be arbitrarily long. Therefore, fixed time windows on the raw response starting with any delay t_d with respect to V_{start} can be considered to extract an n_r-bit raw response. The time window equals n_r/f_s, where f_s is the system clock. Notice that the time window could also be divided into multiple intervals.

The challenge of the PUF is composed of the input current I_{in}, the bias voltage V_{bias} which controls the leaking mechanism of the neuron, the seed of the LFSR, and the delay t_d with respect to V_{start}. The input current I_{in} can be generated using a well-matched programmable current mirror controlled with a digital word of d_1 bits. Such a current mirror has d_1 mirroring branches, each composed of the mirroring transistor and a pass transistor controlled by one bit of the digital word. The bias voltage V_{bias} can be generated from a resistive ladder with equal resistors forming a voltage divider, which can be trimmed for precision. Using d_2 resistors we can program d_2 different V_{bias} values. The space of seeds is 2^{d_3}, where d_3 is the length of the LFSR. The delay t_d can take any value of d_4 clock cycles. Considering a single time window, the CRP space is equal to $2^{d_1} \cdot d_2 \cdot 2^{d_3} \cdot d_4$. The CRP space can be further increased by segmenting the time window into multiple parts. The CRP space can be made very large, which makes the neuron-PUF a good candidate for implementing a strong PUF.

The main source of entropy is process variations within the spiking neuron affecting capacitor values, leakage current I_r, and inverter threshold V_ℓ, which in turn alter the period and duty cycle of the firing pattern as shown in Eqs. (5.1)-(5.2).

The stability booster consists of an LDO regulator and a masking circuit. The LDO regulator stabilizes the supply for all the sub-blocks against temperature and external power supply variations. An already existing LDO inside the chip can be used to power the PUF. In our implementation, we use the same LDO architecture as in Chapter 4, which consists of an SBGR voltage generator, an error amplifier implemented with an OTA, a power p-MOS transistor, and a feedback resistor network. The bitmask is chip-specific and is computed during testing time. In particular, the

chip is exercised by varying temperature and power supply and the n_r-bit PUF response is collected several times. The bitmask is a bitstring of length n_r that has 1 when the corresponding bit has been shown to be stable and 0 otherwise. The bitmask is stored inside the chip in a non-volatile memory (NVM) and loaded into a shift-register at power-up. When the time window starts, the raw PUF response is driven into a shift-register that is clocked with the bitmask. Thus, only the stable bits are serially shifted into the register, while the unstable bits are dropped. At the completion of the time window, the stable n-bit PUF response, where $n \leqslant n_r$, is latched using negative edge-triggered flip-flops.

The advantageous property of the proposed neuron-PUF is that it uses a single compact cell to produce an arbitrarily long key, thus it can offer significant power and area cost savings compared to existing PUFs that use one cell per key-bit.

Since the neuron-PUF generates serially the key and the key-bits are closely spaced, the key cannot be easily correlated to power traces, thus reading-out the key at run-time via side-channel analysis is unworkable. The neuron also presents complex dynamics albeit having a simple structure, i.e., the challenge is related to the derivative of the membrane potential that defines the spike firing. These two properties, combined with the composite and very large CRP space, make the neuron-PUF highly resilient against modelling attacks. Finally, the neuron-PUF is resilient to memory attacks for stealing the key. Stealing the bitmask from the NVM only reveals the position of stable bits and not the key itself, but this also requires knowing part of the challenge, e.g. the delay t_d.

5.4 PUF QUALITY METRICS

There are several PUF quality metrics proposed in the literature [201], [202]. Herein, we use a distinct set of metrics that are the most vital for characterizing the reliability and randomness of a PUF. The set of metrics includes uniformity, uniqueness (or inter-PUF variation), diffuseness, and stability (or intra-PUF variation). The first three metrics also characterize the unpredictability of the PUF against modeling attacks.

We use the following notation in describing PUF metrics:

- N is the number of PUFs evaluated on N different chips.
- n is the number of bits in the PUF response.
- m is the number of challenges of the PUF.
- r_{ijk} is bit j in the PUF response of chip i for challenge k.
- $\mathbf{R}_{ik}$ is the n-bit PUF response of chip i for challenge k, i.e., $\mathbf{R}_{ik} = [r_{i1k}, \cdots, r_{ink}]$.

- T is the number of PUF response measurements over time and different operating conditions, i.e., changes in ambient temperature and supply voltage fluctuations.
- $HD(\mathbf{R}_{ik}, \mathbf{R}_{jk})$ is the hamming distance (HD) of the PUF responses in chips i and j for challenge k. Similarly, $HD(\mathbf{R}_{ik_1}, \mathbf{R}_{ik_2})$ is the HD of the PUF responses in chip i for two different challenges k_1 and k_2, and $HD(\mathbf{R}_{ik}^{t_1}, \mathbf{R}_{ik}^{t_2})$ is the HD of the PUF responses in chip i for challenge k for two different measurements t_1 and t_2.

Uniformity estimates how uniform the proportion of 0 and 1 is in the PUF response bits. For a given chip i and challenge k it is expressed as:

$$\text{Uniformity} = \frac{1}{n}\sum_{j=1}^{n} r_{ijk} \times 100\%. \tag{5.3}$$

Ideally, uniformity should be equal to 50%.

Uniqueness represents the ability of a PUF to uniquely distinguish a chip among a set of identical chips. For a given challenge k it is expressed as:

$$\text{Uniqueness} = \frac{1}{\binom{N}{2}}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\frac{HD(\mathbf{R}_{ik}, \mathbf{R}_{jk})}{n} \times 100\% \tag{5.4}$$

Ideally, uniqueness should be equal to 50%.

Diffuseness measures the difference in the PUF responses when the PUF is queried with different challenges. For a given chip i it is expressed as:

$$\text{Diffuseness} = \frac{1}{\binom{m}{2}}\sum_{k_1=1}^{m-1}\sum_{k_2=k_1+1}^{m}\frac{HD(\mathbf{R}_{ik_1}, \mathbf{R}_{ik_2})}{n} \times 100\% \tag{5.5}$$

Ideally, diffuseness should be equal to 50%.

Stability captures the capability of the PUF to reproduce its response bits under temperature variations, power supply fluctuations, noise, and aging. For a given chip i and challenge k it is expressed as:

$$\text{Stability} = \left(1 - \frac{1}{T}\sum_{t=1}^{T}\frac{HD(\mathbf{R}_{ik}^{0}, \mathbf{R}_{ik}^{t})}{n}\right) \times 100\%, \tag{5.6}$$

where $\mathbf{R}_{ik}^{0}$ denotes an enrollment of the PUF response at nominal operating condition and is used as a reference. Ideally, stability should be equal to 100%.

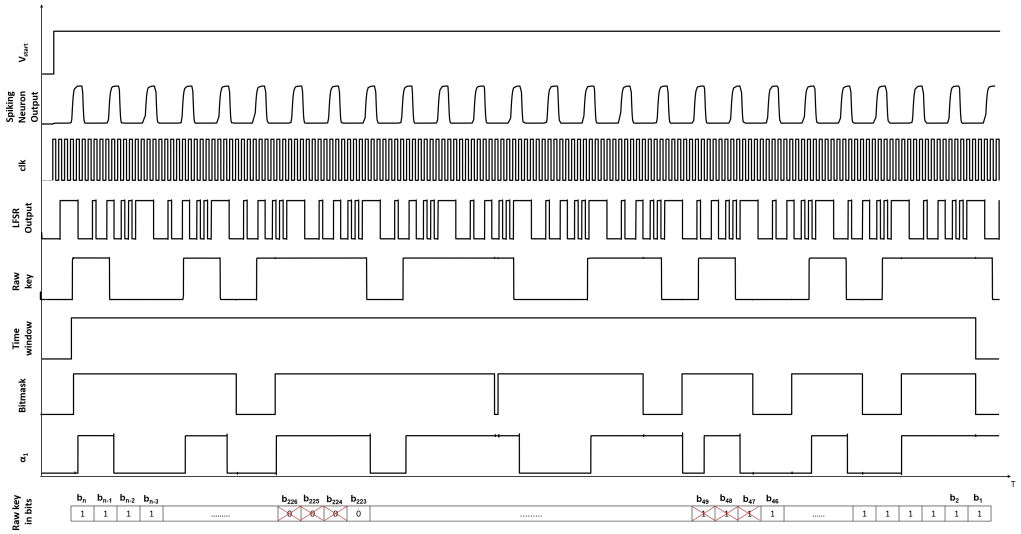


Figure 5.3: Transient simulation of neuron-PUF showing relevant signals.

5.5 RESULTS

The neuron-PUF is designed in the 65nm CMOS technology by STMicroelectronics. For the purpose of simplicity, in our demonstration we consider that the PUF challenge is defined only based on I_{in}, i.e., V_{bias}, seed of LFSR, and delay t_d of the time window with respect to V_{start} are fixed.

For the spiking neuron we use minimum size transistors so as to increase the impact of process variations. We set $I_{in} = 14.6\mu A$ and $V_{bias} = 0.6V$.

We used an LFSR with feedback polynomial $x^5 + x^4 + 1$ and seed "01110", resulting in a pattern that is repeated every 22 clock cycles. It turns out that the choice of the LSFR, i.e., length, polynomial, and seed, affects the PUF metrics. For example, increasing the LFSR length from 3 to 6 bits increases uniqueness by 10-25% and changing the seed can offer a further 4-7% improvement. The chosen LFSR results by performing such trials, but this is a quick analysis since PUF metrics close to their ideal values can be obtained in just a handful of trials.

We opt for generating a 256-bit raw PUF response. Considering a clock period of 2.56ns, this sets the time window to 0.4µs. We consider that the rising edge of the time window comes $t_d = 3$ clock cycles after V_{start} goes high.

The output of the LDO stabilizes around $V_{LDO} = 1.2V$ for the nominal power supply $V_{dd} = 1.6V$. In particular, it shows a 33.4mV variation when V_{dd} varies from 1.4V to 3V and a 10mV variation when temperature varies from -55°C to 125°C. The transient response for a variation of load current from 50mA to 0mA and then from 0mA to 50mA, shows a maxi-

Table 5.1: Neuron-PUF quality metrics.

	Uniformity	Uniqueness	Diffuseness	Stability (T)	Stability (V_{dd})
PUF core	53.83%	48.54%	54.07%	61%	60.51%
With LDO only	53.8%	48.54%	49.43%	90.04%	96.11%
With complete stability booster	47.49%	48.42%	46.25%	100%	100%

mum overshoot of 44.9mV and settles after 875ns, while the maximum undershoot is 53.2mV and response settles after 800ns.

Fig. 5.3 shows a transistor-level transient simulation of the neuron-PUF using the above settings. We plot all the relevant signals and at the bottom we show the raw PUF response and cross out the unstable bits. Bit α_1 corresponds to the output of the first flip-flop of the shift register, as shown in Fig. 5.2. When the bit of the bitmask is 1, α_1 is shifted by one position, i.e., $\alpha_2 = \alpha_1$, and α_1 is updated to store the bit of the current raw PUF response. When the bit of the bitmask is 0, α_1 retains its value. Thus, the first bit in the α_1 bitstring shown in Fig. 5.3 corresponds to bit b_n of the stable PUF response, and the last bit in the α_1 bitstring shown in Fig. 5.3 corresponds to bit b_1 of the stable PUF response.

A set of $N = 100$ PUF instances emulating PUFs from 100 different chips is generated by performing a Monte Carlo (MC) analysis with 100 runs, considering both mismatch and inter-die variations, and using the actual statistical PDK of the technology.

Uniqueness and average uniformity and diffuseness across the 100 PUF instances are calculated by simulating the PUF instances at nominal conditions of 25°C and $V_{dd} = 1.6V$. For diffuseness, we change I_{in} from 13.8µA to 15.3µA with step size of 0.15µA. Stability for each instance is calculated by changing temperature from -25°C to 100°C in steps of 25°C and V_{dd} from 1.3V to 1.9V in steps of 0.1V. Thereafter, we report average stability across all instances.

The four PUF quality metrics are summarized in Table 5.1 considering the PUF raw response at the output of the PUF core, and the PUF response after stability boosting using the LDO only and the complete stability booster. As it can be seen, for the complete system, uniformity, uniqueness and diffuseness are 47.49%, 48.42%, and 46.25%, respectively, i.e. close to their ideal 50% value. Without the stability booster uniformity increases to 53.38% and diffuseness increases to 54.07%. Without the stability booster, the percentage of stable bits under V_{dd} and temperature variations is around 60.5% and 61%, respectively. Adding the LDO enhances the stability by around 35.6% and 29% against V_{dd} and temperature variations, respectively. With the complete stability booster in place, we obtain a fully stable 199-bit PUF response.

The PUF core consumes only 44.39nW/bit or 0.114fJ/bit, considering a clock period of 2.56ns and that there are 256 raw bits. For the same technology node, corresponding reported values for SRAM and RO PUFs are 1100fJ/bit and 474.8fJ/bit, respectively [125]. Using the stability booster, power consumption raises to 0.64µW/bit, but a stability booster is needed in all PUF architectures and can be excluded from the direct comparison.

Finally, the layout area of the PUF core is 13.4µm × 20.2µm. Considering that there are 199 stable bits, area per bit is $322F^2$, where F=65nm is the minimum feature size. For the same technology node, corresponding reported values for SRAM and RO PUFs are $806F^2$ and $39000F^2$, respectively, computed as the ratio of the array area and the number of stable bits [125].

In summary, the neuron-PUF offers significant reductions in area and power overheads compared to SRAM and RO PUFs.

5.6 CONCLUSION

We proposed neuron-PUF, a novel PUF design that uses a single spiking neuron as the source of entropy. Neuron-PUF is a single-cell PUF that uses temporal redundancy to generate a digital key of arbitrary size. It has a composite and large CRP space making it a candidate for implementing a strong PUF. Simulation results show that the neuron-PUF achieves close to ideal PUF metrics. It also has high potential for resisting modeling and memory attacks. The single-cell property results in minimum energy per bit and area per bit compared to all popular PUFs.

6

CONCLUSION AND PERSPECTIVE

6.1 CONCLUSION

Posed by the advancements in fabrication technology that raise the cost of building and maintaining state-of-the-art fabrication premises, more companies have gone fabless in recent years. Furthermore, the advent of new applications and the need for shorter time-to-market led to the prominence of new actors in the semiconductor supply chain such as SoC integrators and IP vendors. This caused the IC design flow to be distributed in a globalized fashion in which the design house has no control over the off-shore entities. This fragility in the face of technological advancements exposes the IC/IP to emerging hardware security vulnerabilities, e.g., counterfeiting, reverse engineering, and hardware Trojans. As a result, design firms have to protect their IC/IP so as to preserve their revenue, market share, and reputation. This security precariousness affects not only the IC/IP owner but also governments and consumers, making hardware security and trust a priority research domain. While hardware security and trust for digital ICs has been extensively studied over the past years, hardware security and trust for analog ICs is an emerging field with a largely unexplored solution space.

In this thesis, we explore this space by shedding light on hardware security and trust challenges for analog ICs, such as the analog design's sensitivity to any modification or insertion of new components and the difficulty in inserting locks with multiple-bit keys into analog designs, and we propose new countermeasures tailored for analog ICs. These countermeasures, while keeping the design intact, can be used to protect analog ICs from various types of threats such as counterfeiting and reverse engineering. We also argued that HT attacks that are initiated in the digital core and are targeting the analog core in the context of SoCs are feasible. From what we discussed, it is clear that developing adequate protection mechanisms for AMS chips requires a thorough understanding of their functionality and design, as well as the threats they face throughout the supply chain.

6.2 CONTRIBUTIONS OF THE THESIS

The main contributions of this thesis can be summarized as follows:

IN CHAPTER 3, we proposed a lockless technique [79] that leverages the programmability embedded in digitally-controlled analog ICs to

allow securing analog ICs against IC-piracy threats such as reverse engineering, cloning, remarking, overproduction, etc. In contrast to current locking methodologies that require a careful insertion of the locking mechanism into the design, i.e., into the biasing circuit, into the on-chip calibration mechanism, or into the digital section, the proposed technique does not make any changes to the original design. We argued that the programming bits controlling the tuning knobs can serve as key-bits and each configuration setting, i.e., programming bits that configure the IC in a specific operation mode demanded by the application, can be treated as a secret key. We demonstrated the proposed technique with simulation and hardware measurements on a programmable ΣΔ modulator used in highly-digitized, multi-standard RF receiver applications. We showed that when invalid programming bits are provided the functionality of the circuit, i.e., SNR, SFDR, IIP3, breaks. We also discussed its advantages over existing locking techniques, as well as its resistance to anticipated attacks.

IN CHAPTER 4, we proposed a stealthy hardware Trojan attack on analog ICs employed in a SoC [80]. The main feature of the proposed HT is that it is unperceivable in the analog domain. This is done by exploiting the on-chip test infrastructure that is shared by the SoC's digital and analog cores. The HT is triggered in the dense digital section of the SoC, thus posing a challenge for HT prevention or detection. The HT payload is transferred to the victim analog IP via the test bus and the interface of the analog IP to the test bus. We proposed four different HT designs, explored the threat model, and discussed the foreseen countermeasures. We demonstrated the proposed HT on two case studies. The first simulation-based case study is an LDO infected by the HT via its DfT interface. The second case study with hardware measurements is an RF receiver front-end infected by the HT through its programmability fabric. We demonstrated the ability of the proposed HT to affect the chip performance in each case study. In particular, flipping just one bit in the DfT pattern or the programming setting can result in the desired effect for the attacker, namely chip performance degradation or complete chip failure.

IN CHAPTER 5, we proposed the *neuron-PUF*, a physical unclonable function that derives its entropy from a single spiking neuron [81]. It generates an analog signature, i.e., pulse wave, whose pre-set duty cycle and periodicity depends on random process variations. A key extractor then uses this signature to generate the digital key. We used a masking-based stability-enhancement technique to remove natively unstable bits discovered during testing. *Neuron-*

PUF uses a composite and multidimensional challenge, making it a good candidate for implementing a strong PUF. The proposed PUF differs from popular PUFs in that it uses a single PUF cell to extract an arbitrarily long PUF response in a serialized fashion using temporal redundancy. As a result, the proposed *neuron-PUF* reduces area and power overheads significantly. Although this is preliminary work, our results provide a starting point to address this new concept.

6.3 FUTURE WORK AND PERSPECTIVE

In terms of future work, the contributions made in this thesis could be enhanced and extended in a variety of ways.

Regarding analog locking, the problem of protecting purely analog cores is still open. Biasing locking protects the biases, but these can be extracted by an attacker as was recently shown in [89]–[92]. By leveraging logic locking we lock the mixed-signal circuit at system-level, but the analog core is still left unprotected. In our proposed locking approach, the design is not obfuscated but its programming bits are protected instead. Another direction is to search for an effective generic methodology for protecting any type of analog circuits, i.e., op-amp, PLL, ADC,DAC, filter, etc., since effective analog locking methods known today are specific to the analog design. For example, locking the calibration module assumes a specific feedback loop that is not commonly met because it is complex to design [93] or the use of AFGTs [94] which are not frequently used either by analog designers. Our proposed locking technique assumes highly-programmable analog designs, thus it is not a generic technique.

Another perspective is to evaluate potential digital-to-analog HT attacks, such as the HT attack that we proposed, and develop effective methods to defend against such attacks.

We are also planning to continue the verification and experimental validation towards proving the *neuron-PUF* robustness and resilience to mainstream attacks. Furthermore, we are planning to investigate the use of different types of spiking neurons [199] in the context of the *neuron-PUF* architecture.

Another direction that we are currently interested in is the protection of hardware accelerators for deep neural networks (DNNs). DNN-based computing techniques have made significant progress in solving long-standing artificial intelligence (AI) and machine learning (ML) problems, e.g., object recognition, natural language processing, autonomous systems, etc. Despite recent advancements in DNN hardware accelerator design, detailed DNN hardware accelerator security methodologies lag behind. The most common DNN hardware accelerators employed today are graphical processing units (GPUs), field-programmable gate arrays

(FPGAs), and tensor processing units (TPUs). However, ASIC implementations improve DNN performance by orders of magnitude and are under intense research. As a result, new security challenges emerge to protect the intellectual property of ASIC hardware accelerators and, in general, evaluate the vulnerability of AI hardware accelerators to side-channel attacks, fault-injection attacks, HT attacks, etc. [203]–[207].

Part II

APPENDIX

BIBLIOGRAPHY

[1] Bill McClean, *Fabless Company Share of IC Sales to Set New Record in 2020 at 32.9%*, Dec. 2020. [Online]. Available: https://www.icinsights.com/data/articles/documents/1328.pdf (visited on 04/01/2021) (cit. on p. 3).

[2] ——, *Intel to Keep Its Number One Semiconductor Supplier Ranking in 2020*, Nov. 2020. [Online]. Available: https://www.icinsights.com/data/articles/documents/1320.pdf (visited on 04/01/2021) (cit. on p. 3).

[3] H. Hung, Y. Chiu, and M. Wu, "Analysis of competition between IDM and fabless–foundry business models in the semiconductor industry," *IEEE Transactions on Semiconductor Manufacturing*, vol. 30, no. 3, pp. 254–260, 2017 (cit. on p. 3).

[4] A. Barenghi, L. Breveglieri, I. Koren, and D. Naccache, "Fault injection attacks on cryptographic devices: theory, practice, and countermeasures," *Proceedings of the IEEE*, vol. 100, no. 11, pp. 1207–1228, 2012 (cit. on p. 5).

[5] S. Bhunia and M. Tehranipoor, "Chapter 8 - side-channel attacks," in *Hardware Security*, S. Bhunia and M. Tehranipoor, Eds., Morgan Kaufmann, 2019, pp. 193–218 (cit. on p. 5).

[6] H. Choukri and M. Tunstall, "Round reduction using faults," *Workshop on Fault Diagnosis and Tolerance in Cryptography*, vol. 5, pp. 13–24, 2005 (cit. on p. 5).

[7] I. Polian, "Security aspects of analog and mixed-signal circuits," in *Proc. IEEE International Mixed-Signal Testing Workshop*, 2016 (cit. on p. 5).

[8] K. Tiri and I. Verbauwhede, "A VLSI design flow for secure side-channel attack resistant ICs," in *Proc. Design, Automation & Test in Europe*, 2005 (cit. on p. 5).

[9] N. Beringuier-Boher, K. Gomina, D. Hely, J.-B. Rigaud, V. Beroulle, A. Tria, J. Damiens, P. Gendrier, and P. Candelier, "Voltage glitch attacks on mixed-signal systems," in *Proc. Euromicro Conference on Digital System Design*, 2014, pp. 379–386 (cit. on p. 5).

[10] S. Bhunia, M. S. Hsiao, M. Banga, and S. Narasimhan, "Hardware trojan attacks: threat analysis and countermeasures," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1229–1247, 2014 (cit. on pp. 5, 8).

[11] K. Xiao, D. Forte, Y. Jin, R. Karri, S. Bhunia, and M. Tehranipoor, "Hardware Trojans: lessons learned after one decade of research," *ACM Transactions on Design Automation of Electronic Systems*, vol. 22, no. 1, 6:1–6:23, 2016 (cit. on pp. 5, 8).

[12] R. Karri, J. Rajendran, K. Rosenfeld, and M. Tehranipoor, "Trustworthy hardware: identifying and classifying hardware trojans," *Computer*, vol. 43, no. 10, pp. 39–46, 2010 (cit. on p. 5).

[13] X. Wang, T. Mal-Sarkar, A. Krishna, S. Narasimhan, and S. Bhunia, "Software exploitable hardware trojans in embedded processor," in *2012 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, 2012, pp. 55–58 (cit. on p. 5).

[14] G. T. Becker, F. Regazzoni, C. Paar, and W. P. Burleson, "Stealthy dopant-level hardware trojans: extended version," *Journal of Cryptographic Engineering*, vol. 4, no. 1, pp. 19–31, 2014 (cit. on pp. 5, 6).

[15] J. Clements and Y. Lao, "Hardware trojan design on neural networks," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2019, pp. 1–5 (cit. on p. 5).

[16] X. Cao, Q. Wang, R. L. Geiger, and D. J. Chen, "A hardware trojan embedded in the inverse widlar reference generator," in *Proc. IEEE International Midwest Symposium on Circuits and Systems*, 2015 (cit. on pp. 5, 25, 57).

[17] Yier Jin and Y. Makris, "Hardware trojan detection using path delay fingerprint," in *2008 IEEE International Workshop on Hardware-Oriented Security and Trust*, 2008, pp. 51–57 (cit. on pp. 5, 11).

[18] Y. Shiyanovskii, F. Wolff, A. Rajendran, C. Papachristou, D. Weyer, and W. Clay, "Process reliability based trojans through NBTI and HCI effects," in *NASA/ESA Conference on Adaptive Hardware and Systems*, 2010, pp. 215–222 (cit. on p. 6).

[19] L. Lin, T. G. M. Kasper, C. Paar, and W. Burleson, "Trojan side-channels: lightweight hardware trojans through side-channel engineering," in *Cryptographic Hardware and Embedded Systems - CHES 2009*, Springer Berlin Heidelberg, 2009, pp. 382–395 (cit. on p. 6).

[20] K. Yang, M. Hicks, Q. Dong, T. Austin, and D. Sylvester, "A2: analog malicious hardware," in *Proc. IEEE Symposium on Security and Privacy*, 2016, pp. 18–37 (cit. on pp. 6, 24, 26).

[21] R. Torrance and D. James, "The state-of-the-art in semiconductor reverse engineering," in *Proc. IEEE/ACM Design Automation Conference*, 2011, pp. 333–338 (cit. on pp. 6, 8).

[22] B. Lippmann, M. Werner, N. Unverricht, A. Singla, P. Egger, A. Dübotzky, H. Gieser, M. Rasche, O. Kellermann, and H. Graeb, "Integrated flow for reverse engineering of nanoscale technologies," in *Proc. Asia and South Pacific Design Automation Conference*, 2019, pp. 82–89 (cit. on pp. 6, 8, 11).

[23] U. Guin, K. Huang, D. DiMase, J. M. Carulli, M. Tehranipoor, and Y. Makris, "Counterfeit integrated circuits: a rising threat in the global semiconductor supply chain," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1207–1228, 2014 (cit. on pp. 7, 12, 13).

[24] M. Rostami, F. Koushanfar, and R. Karri, "A primer on hardware security: models, methods, and metrics," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1283–1295, 2014 (cit. on pp. 7, 9).

[25] J. A. Roy, F. Koushanfar, and I. L. Markov, "EPIC: Ending piracy of integrated circuits," *IEEE Computer*, vol. 43, no. 10, pp. 30–38, 2010 (cit. on pp. 7, 10, 12, 36).

[26] S. E. Quadir, J. Chen, D. Forte, N. Asadizanjani, S. Shahbazmohamadi, L. Wang, J. Chandy, and M. Tehranipoor, "A survey on chip to system reverse engineering," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 13, no. 1, pp. 1–34, 2016 (cit. on p. 8).

[27] S. Adee, "The hunt for the kill switch," *IEEE Spectrum*, vol. 45, no. 5, pp. 34–39, May 2008. [Online]. Available: https://spectrum.ieee.org/semiconductors/design/the-hunt-for-the-kill-switch (visited on 01/26/2021) (cit. on p. 8).

[28] A. Antonopoulos, C. Kapatsori, and Y. Makris, "Security and trust in the analog/mixed-signal/RF domain: a survey and a perspective," in *Proc. IEEE European Test Symposium*, 2017 (cit. on p. 8).

[29] J. Robertson and M. Riley, *The big hack: inside the chinese cyberspies' bag of tech tricks*, Oct. 2018. [Online]. Available: https://www.bloomberg.com/news/articles/2018-10-04/the-big-hack-inside-the-bag-of-tech-tricks-used-by-china-spies (visited on 04/12/2021) (cit. on p. 8).

[30] S. Bhasin and F. Regazzoni, "A survey on hardware trojan detection techniques," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2015, pp. 2021–2024 (cit. on p. 8).

[31] X. Guo, R. G. Dutta, J. He, M. M. Tehranipoor, and Y. Jin, "QIF-Verilog: quantitative information-flow based hardware description languages for pre-silicon security assessment," in *2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, 2019, pp. 91–100 (cit. on p. 9).

[32] K. Xiao, A. Nahiyan, and M. Tehranipoor, "Security rule checking in ic design," *Computer*, vol. 49, no. 8, pp. 54–61, 2016 (cit. on p. 9).

[33] D. Zhang, Y. Wang, G. E. Suh, and A. C. Myers, "A hardware design language for timing-sensitive information-flow security," *SIGARCH Comput. Archit. News*, vol. 43, no. 1, pp. 503–516, 2015 (cit. on p. 9).

[34] X. Li, M. Tiwari, J. K. Oberg, V. Kashyap, F. T. Chong, T. Sherwood, and B. Hardekopf, "Caisson: a hardware description language for secure information flow," in *Proceedings of the 32nd ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2011, pp. 109–120 (cit. on p. 9).

[35] X. Li, V. Kashyap, J. K. Oberg, M. Tiwari, V. R. Rajarathinam, R. Kastner, T. Sherwood, B. Hardekopf, and F. T. Chong, "Sapper: a language for hardware-level security policy enforcement," in *Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems*, 2014, pp. 97–112 (cit. on p. 9).

[36] K. Shim, "A survey of public-key cryptographic primitives in wireless sensor networks," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 577–601, 2016 (cit. on p. 9).

[37] A. D. Biagio, A. Barenghi, G. Agosta, and G. Pelosi, "Design of a parallel AES for graphics hardware using the CUDA framework," in *2009 IEEE International Symposium on Parallel Distributed Processing*, 2009, pp. 1–8 (cit. on p. 9).

[38] N. J. Oishi, A. Mahamud, and Asaduzzaman, "Short paper: enhancing wi-fi security using a hybrid algorithm of blowfish and RC6," in *2016 International Conference on Networking Systems and Security (NSysS)*, 2016, pp. 1–5 (cit. on p. 9).

[39] P. Swierczynski, M. Fyrbiak, P. Koppe, and C. Paar, "FPGA trojans through detecting and weakening of cryptographic primitives," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 8, pp. 1236–1249, 2015 (cit. on p. 9).

[40] M. Tehranipoor and C. Wang, Eds., *Introduction to Hardware Security and Trust*. Springer-Verlag New York, 2012 (cit. on p. 9).

[41] S. Bhunia, S. Ray, and S. S.-K. (Eds.), *Fundamentals of IP and SoC Security*. Springer International Publishing AG, 2017 (cit. on p. 9).

[42] M. Yasin, J. Rajendran, O. Sinanoglu, and R. Karri, "On improving the security of logic locking," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 9, pp. 1411–1424, 2016 (cit. on p. 10).

[43] N. Limaye, E. Kalligeros, N. Karousos, I. G. Karybali, and O. Sinanoglu, "Thwarting all logic locking attacks: dishonest oracle with truly random logic locking," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1–1, 2020 (cit. on p. 10).

[44] M. Yasin, A. Sengupta, M. T. Nabeel, M. Ashraf, J. Rajendran, and O. Sinanoglu, "Provably-secure logic locking: from theory to practice," in *Proc. ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1601–1618 (cit. on pp. 10, 20).

[45] M. Yasin, S. M. Saeed, J. Rajendran, and O. Sinanoglu, "Activation of logic encrypted chips: pre-test or post-test?" In *Proc. Design, Automation & Test in Europe Conference & Exhibition*, 2016, pp. 139–144 (cit. on p. 10).

[46] C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: a tutorial," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126–1141, 2014 (cit. on p. 10).

[47] J. Leonhard, A. Sayed, M.-M. Louërat, H. Aboushady, and H.-G. Stratigopoulos, "Analog and mixed-signal IC security via sizing camouflaging," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2020 (cit. on pp. 10, 11, 22).

[48] M. Li, K. Shamsi, T. Meade, Z. Zhao, B. Yu, Y. Jin, and D. Z. Pan, "Provably secure camouflaging strategy for IC protection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 8, pp. 1399–1412, 2019 (cit. on p. 10).

[49] X. Zhang and M. Tehranipoor, "Case study: detecting hardware trojans in third-party digital IP cores," in *IEEE International Symposium on Hardware-Oriented Security and Trust*, 2011, pp. 67–70 (cit. on p. 11).

[50] R. S. Chakraborty, F. Wolff, S. Paul, C. Papachristou, and S. Bhunia, "MERO: a statistical approach for hardware trojan detection," in *Cryptographic Hardware and Embedded Systems - CHES 2009*, Springer Berlin Heidelberg, 2009, pp. 396–410 (cit. on p. 11).

[51] V. R. Surabhi, P. Krishnamurthy, H. Amrouch, K. Basu, J. Henkel, R. Karri, and F. Khorrami, "Hardware trojan detection using controlled circuit aging," *IEEE Access*, vol. 8, pp. 77 415–77 434, 2020 (cit. on p. 11).

[52] M. Hicks, M. Finnicum, S. T. King, M. M. K. Martin, and J. M. Smith, "Overcoming an untrusted computing base: detecting and removing malicious hardware automatically," in *IEEE Symposium on Security and Privacy*, 2010, pp. 159–172 (cit. on p. 11).

[53] K. Xiao, D. Forte, and M. Tehranipoor, "A novel built-in self-authentication technique to prevent inserting hardware trojans," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 12, pp. 1778–1791, 2014 (cit. on p. 11).

[54] K. Shamsi, M. Li, K. Plaks, S. Fazzari, D. Z. Pan, and Y. Jin, "IP protection and supply chain security through logic obfuscation: a systematic overview," *ACM Transactions on Design Automation of Electronic Systems*, vol. 24, no. 6, 65:1–65:36, 2019 (cit. on p. 11).

[55] J. Leonhard, M. Yasin, S. Turk, M. Nabeel, M.-M. Louërat, R. Chotin-Avot, H. Aboushady, O. Sinanoglu, and H.-G. Stratigopoulos, "MixLock: securing mixed-signal circuits via logic locking," in *Proc. Design, Automation & Test in Europe Conference*, 2019 (cit. on pp. 11, 12, 20, 21, 34).

[56] M. Yasin, J. Rajendran, and O. Sinanoglu, *Trustworthy Hardware Design: Combinational Logic Locking Techniques*. Springer, 2020 (cit. on pp. 11, 12).

[57] J. Rajendran, M. Sam, O. Sinanoglu, and R. Karri, "Security analysis of integrated circuit camouflaging," in *Proc. ACM Conference on Computer and Communications Security*, 2013, pp. 709–720 (cit. on p. 11).

[58] T. D. Perez and S. Pagliarini, "A survey on split manufacturing: attacks, defenses, and challenges," *IEEE Access*, vol. 8, pp. 184 013–184 035, 2020 (cit. on p. 11).

[59] T. Sugawara, D. Suzuki, R. Fujii, S. Tawa, R. Hori, M. Shiozaki, and T. Fujino, "Reversing stealthy dopant-level circuits," *Journal of Cryptographic Engineering*, vol. 5, no. 2, pp. 85–94, 2015 (cit. on p. 11).

[60] F. Stellari, P. Song, A. J. Weger, J. Culp, A. Herbert, and D. Pfeiffer, "Verification of untrusted chips using trusted layout and emission measurements," in *IEEE International Symposium on Hardware-Oriented Security and Trust*, 2014, pp. 19–24 (cit. on p. 11).

[61] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, "Trojan detection using IC fingerprinting," in *IEEE Symposium on Security and Privacy*, 2007, pp. 296–310 (cit. on pp. 11, 26).

[62] S. Narasimhan, W. Yueh, X. Wang, S. Mukhopadhyay, and S. Bhunia, "Improving IC security against trojan attacks through integration of security monitors," *IEEE Design & Test of Computers*, vol. 29, no. 5, pp. 37–46, 2012 (cit. on p. 11).

[63] D. Forte, C. Bao, and A. Srivastava, "Temperature tracking: an innovative run-time approach for hardware trojan detection," in *IEEE/ACM International Conference on Computer-Aided Design*, 2013, pp. 532–539 (cit. on p. 11).

[64] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *ACM/IEEE Design Automation Conference (DAC)*, 2007, pp. 9–14 (cit. on p. 12).

[65] A. R. Sadeghi, I. Visconti, and C. Wachsmann, "Enhancing RFID security and privacy by physically unclonable functions," in *Towards Hardware-Intrinsic Security. Information Security and Cryptography*, A. R. Sadeghi and D. N. (eds), Eds., Berlin, Heidelberg: Springer, 2010, pp. 281–305 (cit. on p. 12).

[66] J. Guajardo, S. S. Kumar, G. Schrijen, and P. Tuyls, "Brand and IP protection with physical unclonable functions," in *IEEE International Symposium on Circuits and Systems*, 2008, pp. 3186–3189 (cit. on p. 12).

[67] B. Chatterjee, D. Das, and S. Sen, "RF-PUF: IoT security enhancement through authentication of wireless nodes using in-situ machine learning," in *Proc. IEEE International Symposium on Hardware Oriented Security and Trust*, 2018, pp. 205–208 (cit. on p. 12).

[68] M. J. Mahmod and U. A. Guin, "Robust, lowcost and secure authentication scheme for IoT applications," *MDPI Cryptography*, vol. 4, no. 1, p. 8, 2020 (cit. on p. 12).

[69] K. Lofstrom, W. R. Daasch, and D. Taylor, "IC identification circuit using device mismatch," in *IEEE International Solid-State Circuits Conference*, 2000, pp. 372–373 (cit. on p. 12).

[70] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, 2002 (cit. on pp. 12, 27).

[71] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *ACM Conference on Computer and Communications Security*, 2002, pp. 148–160 (cit. on p. 12).

[72] R. Maes, A. V. Herrewege, and I. Verbauwhede, "PUFKY: a fully functional PUF-based cryptographic key generator," in *Cryptographic Hardware and Embedded Systems – CHES 2012*, Springer Berlin Heidelberg, 2012, pp. 302–319 (cit. on p. 12).

[73] M. Hiller, D. Merli, F. Stumpf, and G. Sigl, "Complementary IBS: application specific error correction for PUFs," in *IEEE International Symposium on Hardware-Oriented Security and Trust*, 2012 (cit. on p. 12).

[74] S. K. Mathew, S. K. Satpathy, M. A. Anders, H. Kaul, S. K. Hsu, A. Agarwal, G. K. Chen, R. J. Parker, R. K. Krishnamurthy, and V. De, "A 0.19pJ/b PVT-variation-tolerant hybrid physically unclonable function circuit for 100% stable secure key generation in 22nm CMOS," in *IEEE International Solid-State Circuits Conference*, 2014, pp. 278–279 (cit. on p. 12).

[75] J. Delvaux, D. Gu, D. Schellekens, and I. Verbauwhede, "Helper data algorithms for PUF-based key generation: overview and analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 6, pp. 889–902, 2015 (cit. on p. 12).

[76] B. Colombier, L. Bossuet, V. Fischer, and D. Hély, "Key reconciliation protocols for error correction of silicon PUF responses," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 8, pp. 1988–2002, 2017 (cit. on p. 12).

[77] A. Antonopoulos, C. Kapatsori, and Y. Makris, "Trusted analog/mixed-signal/RF ICs: a survey and a perspective," *IEEE Design & Test*, vol. 34, no. 6, pp. 63–76, 2017 (cit. on p. 12).

[78] M. M. Alam, S. Chowdhury, B. Park, D. Munzer, N. Maghari, M. Tehranipoor, and D. Forte, "Challenges and Opportunities in Analog and Mixed Signal (AMS) Integrated Circuit (IC) Security," *Journal of Hardware and Systems Security*, vol. 2, no. 1, pp. 15–32, 2018 (cit. on p. 12).

[79] M. Elshamy, A. Sayed, M.-M. Louërat, A. Rhouni, H. Aboushady, and H.-G. Stratigopoulos, "Securing Programmable Analog ICs Against Piracy," in *Proc. Design, Automation and Test in Europe Conference*, 2020 (cit. on pp. 14, 91).

[80] M. Elshamy, G. Di Natale, A. Pavlidis, M.-M. Louërat, and H.-G. Stratigopoulos, "Hardware Trojan Attacks in Analog/Mixed-Signal ICs via the Test Access Mechanism," in *Proc. IEEE European Test Symposium*, 2020 (cit. on pp. 14, 92).

[81] M. Elshamy and H.-G. Stratigopoulos, "Neuron-PUF: Physical Unclonable Function Basedon a Single Spiking Neuron," in *The 27th IEEE International Symposium on On-Line Testing and Robust System Design*, 2021 (cit. on pp. 14, 92).

[82] Maxim Integrated, *MAX36051: DeepCover Security Manager with 128 Bytes of Nonimprinting Memory* (cit. on p. 17).

[83] ——, *Deepcover security manager for low-voltage operation with 1kb secure memory and programmable tamper hierarchy*, 2010. [Online]. Available: `https://www.maximintegrated.com/en/products/embedded-security/security-managers/DS3660.html` (visited on 04/17/2021) (cit. on p. 17).

[84] P. Tuyls, G.-J. Schrijen, B. Škorić, J. van Geloven, N. Verhaegh, and R. Wolters, "Read-proof hardware from protective coatings," in *Cryptographic Hardware and Embedded Systems - CHES 2006*, L. Goubin and M. Matsui, Eds., Springer Berlin Heidelberg, 2006, pp. 369–383 (cit. on p. 17).

[85] V. V. Rao and I. Savidis, "Protecting analog circuits with parameter biasing obfuscation," in *Proc. IEEE Latin American Test Symposium*, 2017 (cit. on pp. 18, 19, 34).

[86] J. Wang, C. Shi, A. Sanabria-Borbon, E. Sánchez-Sinencio, and J. Hu, "Thwarting analog IC piracy via combinational locking," in *Proc. IEEE International Test Conference*, 2017 (cit. on pp. 18, 19, 34).

[87] V. V. Rao and I. Savidis, "Mesh based obfuscation of analog circuit properties," in *IEEE International Symposium on Circuits and Systems*, 2019 (cit. on pp. 18, 19, 34).

[88] G. Volanis, Y. Lu, S. Govinda, R. Nimmalapudi, A. Antonopoulos, A. Marshall, and Y. Makris, "Analog performance locking through neural network-based biasing," in *Proc. IEEE VLSI Test Symposium*, 2019 (cit. on pp. 18, 19, 34).

[89] N. G. Jayasankaran, A. Sanabria-Borbón, A. Abuellil, E. Sánchez-Sinencio, J. Hu, and J. Rajendran, "Breaking analog locking techniques," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 10, pp. 2157–2170, 2020 (cit. on pp. 19, 37, 93).

[90] V. V. Rao, K. Juretus, and I. Savidis, "Security vulnerabilities of obfuscated analog circuits," in *IEEE International Symposium on Circuits and Systems*, 2020 (cit. on pp. 19, 37, 93).

[91] R. Y. Acharya, S. Chowdhury, F. Ganji, and D. Forte, "Attack of the genes: finding keys and parameters of locked analog ICs using genetic algorithm," in *IEEE International Symposium on Hardware Oriented Security and Trust*, 2020, pp. 284–294 (cit. on pp. 19, 37, 93).

[92] J. Leonhard, M. Elshamy, M.-M. Louërat, and H.-G. Stratigopoulos, "Breaking analog biasing locking techniques via re-synthesis," in *Proceedings of the 26th Asia and South Pacific Design Automation Conference*, 2021, pp. 555–560 (cit. on pp. 19, 37, 93).

[93] N. G. Jayasankaran, A. S. Borbon, E. Sanchez-Sinencio, J. Hu, and J. Rajendran, "Towards provably-secure analog and mixed-signal locking against overproduction," in *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 2018 (cit. on pp. 19–21, 34, 93).

[94] S. G. R. Nimmalapudi, G. Volanis, Y. Lu, A. Antonopoulos, A. Marshall, and Y. Makris, "Range-controlled floating-gate transistors: a unified solution for unlocking and calibrating analog ICs," in *Proc. Design, Automation and Test in Europe Conference*, 2020 (cit. on pp. 20, 93).

[95] J. Leonhard, M.-M. Louërat, H. Aboushady, O. Sinanoglu, and H.-G. Stratigopoulos, "Mixed-signal hardware security using MixLock: demonstration in an audio application," in *International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design*, 2019 (cit. on pp. 21, 34).

[96] D. H. K. Hoe, J. Rajendran, and R. Karri, "Towards secure analog designs: a secure sense amplifier using memristors," in *Proc. IEEE Computer Society Annual Symposium on VLSI*, 2014 (cit. on p. 21).

[97] K. Juretus, V. V. Rao, and I. Savidis, "Securing analog mixed-signal integrated circuits through shared dependencies," in *Proc. ACM Great Lakes Symposium on VLSI*, 2019 (cit. on p. 22).

[98] A. Ash-Saki and S. Ghosh, "How multi-threshold designs can protect analog IPs," in *Proc. IEEE International Conference on Computer Design*, 2018, pp. 464–471 (cit. on p. 22).

[99] Y. Bi, J. S. Yuan, and Y. Jin, "Beyond the interconnections: split manufacturing in RF designs," *Electronics*, vol. 4, no. 3, pp. 541–564, 2015 (cit. on p. 23).

[100] J. Rajendran, O. Sinanoglu, and R. Karri, "Is split manufacturing secure?" In *2013 Design, Automation Test in Europe Conference Exhibition DATE*, 2013, pp. 1259–1264 (cit. on p. 23).

[101] M. M. Bidmeshki, K. S. Subramani, and Y. Makris, "Revisiting Capacitor-Based Trojan Design," in *2019 IEEE 37th International Conference on Computer Design*, IEEE, 2019, pp. 309–312 (cit. on p. 24).

[102] Y. Jin and Y. Makris, "Hardware trojans in wireless cryptographic ICs," *IEEE Design & Test of Computers*, vol. 27, no. 1, pp. 26–35, 2010 (cit. on pp. 24, 57).

[103] Y. Liu, Y. Jin, A. Nosratinia, and Y. Makris, "Silicon demonstration of hardware trojan design and detection in wireless cryptographic ICs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 4, pp. 1506–1519, 2017 (cit. on pp. 24, 26, 57).

[104] S. Chang, G. Bhat, U. Ogras, B. Bakkaloglu, and S. Ozev, "Detection mechanisms for unauthorized wireless transmissions," *ACM Transactions on Design Automation of Electronic Systems*, vol. 23, no. 6, 70:1–70:21, 2018 (cit. on pp. 24–26, 57).

[105] K. S. Subramani, N. Helal, A. Antonopoulos, A. Nosratinia, and Y. Makris, "Amplitude-modulating analog/RF hardware trojans in wireless networks: risks and remedies," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3497–3510, 2020 (cit. on pp. 24, 26, 57).

[106] Z. Liu, Y. Li, Y. Duan, R. L. Geiger, and D. Chen, "Identification and break of positive feedback loops in trojan states vulnerable circuits," in *Proc. IEEE International Symposium on Circuits and Systems*, 2014, pp. 289–292 (cit. on pp. 25, 57).

[107] Q. Wang, R. L. Geiger, and D. Chen, "Hardware trojans embedded in the dynamic operation of analog and mixed-signal circuits," in *Proc. National Aerospace and Electronics Conference*, 2015, pp. 155–158 (cit. on pp. 25, 57).

[108] C. Cai and D. Chen, "Performance enhancement induced trojan states in op-amps, their detection and removal," in *Proc. IEEE International Symposium on Circuits and Systems*, 2015, pp. 3020–3023 (cit. on pp. 25, 57).

[109] Q. Wang, D. Chen, and R. L. Geiger, "Transparent side channel trigger mechanism on analog circuits with PAAST hardware trojans," in *IEEE International Symposium on Circuits and Systems*, 2018 (cit. on pp. 25, 57).

[110] A. Waksman, M. Suozzo, and S. Sethumadhavan, "FANCI: identification of stealthy malicious logic using boolean functional analysis," in *Proc. ACM Conference on Computer and Communications Security*, Association for Computing Machinery, 2013, pp. 697–708 (cit. on p. 25).

[111] M.-M. Bidmeshki, A. Antonopoulos, and Y. Makris, "Information flow tracking in analog/mixed-signal designs through proof-carrying hardware IP," in *Proc. Design, Automation and Test Conference in Europe*, 2017 (cit. on p. 25).

[112] S. Bhasin, J.-L. Danger, S. Guilley, X. T. Ngo, and L. Sauvage, "Hardware Trojan Horses in Cryptographic IP Cores," in *Workshop on Fault Diagnosis and Tolerance in Cryptography*, IEEE, 2013, pp. 15–29 (cit. on p. 26).

[113] Y. Hou, H. He, K. Shamsi, Y. Jin, D. Wu, and H. Wu, "On-chip analog trojan detection framework for microprocessor trustworthiness," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 10, pp. 1820–1830, 2019 (cit. on p. 26).

[114] S. Tajik, E. Dietz, S. Frohmann, J.-P. Seifert, D. Nedospasov, C. Helfmeier, C. Boit, and H. Dittrich, "Physical characterization of arbiter PUFs," in *Cryptographic Hardware and Embedded Systems - CHES 2014*, 2014, pp. 493–509 (cit. on p. 26).

[115] T. Machida, D. Yamamoto, M. Iwamoto, and K. Sakiyama, "A new mode of operation for arbiter PUF to improve uniqueness on FPGA," in *2014 Federated Conference on Computer Science and Information Systems*, 2014, pp. 871–878 (cit. on p. 26).

[116] U. Chatterjee, R. S. Chakraborty, H. Kapoor, and D. Mukhopadhyay, "Theory and application of delay constraints in arbiter puf," *ACM Trans. Embed. Comput. Syst.*, vol. 15, no. 1, Jan. 2016 (cit. on p. 26).

[117] S. S. Zalivaka, A. A. Ivaniuk, and C. Chang, "Reliable and modeling attack resistant authentication of arbiter PUF in FPGA implementation with trinary quadruple response," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 1109–1123, 2019 (cit. on p. 26).

[118] S. V. S. Avvaru, Z. Zeng, and K. K. Parhi, "Homogeneous and heterogeneous feed-forward xor physical unclonable functions," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2485–2498, 2020 (cit. on p. 26).

[119] C. Q. Liu, Y. Cao, and C. H. Chang, "CRO-PUF: a low-power, reliable and aging-resilient current starved inverter-based ring oscillator physical unclonable function," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 12, pp. 3138–3149, 2017 (cit. on p. 27).

[120] A. Cherkaoui, L. Bossuet, and C. Marchand, "Design, evaluation, and optimization of physical unclonable functions based on transient effect ring oscillators," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1291–1305, 2016 (cit. on p. 27).

[121] J. Zhang, X. Tan, Y. Zhang, W. Wang, and Z. Qin, "Frequency offset-based ring oscillator physical unclonable function," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 4, no. 4, pp. 711–721, 2018 (cit. on p. 27).

[122] M. T. Rahman, D. Forte, J. Fahrny, and M. Tehranipoor, "ARO-PUF: an aging-resistant ring oscillator PUF design," in *2014 Design, Automation Test in Europe Conference Exhibition DATE*, 2014, pp. 1–6 (cit. on p. 27).

[123] K. Liu, Y. Min, X. Yang, H. Sun, and H. Shinohara, "A 373-f2 0.21%-native-BER EE SRAM physically unclonable function with 2-d power-gated bit cells and V_{ss} bias-based dark-bit detection," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 6, pp. 1719–1732, 2020 (cit. on p. 27).

[124] J. Li, T. Yang, and M. Seok, "A technique to transform 6T-SRAM arrays into robust analog PUF with minimal overhead," in *IEEE International Symposium on Circuits and Systems*, 2017 (cit. on p. 27).

[125] A. B. Alvarez, W. Zhao, and M. Alioto, "Static physically unclonable functions for secure chip identification with 1.9–5.8% native bit instability at 0.6–1 V and 15 fJ/bit in 65 nm," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 3, pp. 763–775, 2018 (cit. on pp. 27, 89).

[126] L. T. Clark, S. B. Medapuram, D. K. Kadiyala, and J. Brunhaver, "Physically unclonable functions using foundry SRAM cells," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 3, pp. 955–966, 2019 (cit. on p. 27).

[127] M. R. Carro-Temboury, R. Arppe, T. Vosch, and T. J. Sørensen, "An optical authentication system based on imaging of excitation-selected lanthanide luminescence," *Science Advances*, vol. 4, e1701384, 2018 (cit. on p. 27).

[128] C. Mesaritakis, M. Akriotou, A. Kapsalis, E. Grivas, C. Chaintoutis, T. Nikas, and D. Syvridis, "Physical unclonable function based on a multi-mode optical waveguide," *Scientific Reports*, vol. 8, p. 9653, 2018 (cit. on p. 27).

[129] S. Zeitouni, E. Stapf, H. Fereidooni, and A.-R. Sadeghi, "On the security of strong memristor-based physically unclonable functions," in *2020 57th ACM/IEEE Design Automation Conference DAC*, 2020, pp. 1–6 (cit. on p. 27).

[130] G. S. Rose, N. McDonald, L.-K. Yan, and B. Wysocki, "A write-time based memristive PUF for hardware security applications," in *2013 IEEE/ACM International Conference on Computer-Aided Design ICCAD*, 2013, pp. 830–833 (cit. on p. 27).

[131] V. Natarajan, S. Sen, A. Banerjee, A. Chatterjee, G. Srinivasan, and F. Taenzler, "Analog signature-driven postmanufacture multidimensional tuning of RF systems," *IEEE Design & Test of Computers*, vol. 27, no. 6, pp. 6–17, 2010 (cit. on p. 31).

[132] J. W. Jeong, A. Nassery, J. N. Kitchen, and S. Ozev, "Built-in self-test and digital calibration of zero-IF RF transceivers," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 6, pp. 2286–2298, 2016 (cit. on p. 31).

[133] Y. Lu, K. S. Subramani, H. Huang, N. Kupp, K. Huang, and Y. Makris, "A comparative study of one-shot statistical calibration methods for analog/RF ICs," in *Proc. IEEE International Test Conference*, Paper 21.3, 2015 (cit. on p. 31).

[134] M. Andraud, H.-G. Stratigopoulos, and E. Simeu, "One-shot non-intrusive calibration against process variations for analog/RF circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 11, pp. 2022–2035, 2016 (cit. on pp. 31, 60).

[135] F. Cilici, M. J. Barragan, S. Mir, E. Lauga-Larroze, S. Bourdel, and G. Leger, "Yield recovery of mm-wave power amplifiers using variable decoupling cells and one-shot statistical calibration," in *IEEE International Symposium on Circuits and Systems*, 2019 (cit. on p. 31).

[136] C. Maxey, G. Creech, S. Raman, J. Rockway, K. Groves, T. Quach, L. Orlando, and A. Mattamana, "Mixed-signal SoCs with in situ self-healing circuitry," *IEEE Design & Test of Computers*, vol. 29, no. 6, pp. 27–39, 2012 (cit. on pp. 31, 60).

[137] S. Bowers, K. Sengupta, B. Parker, and A. Hajimiri, "Integrated self-healing for mm-wave power amplifiers," *IEEE Transactions on Microwave Theory and Techniques*, vol. 61, no. 3, pp. 352–363, 2013 (cit. on pp. 31, 60).

[138] S. Sen, V. Natarajan, S. Devarakond, and A. Chatterjee, "Process-variation tolerant channel-adaptive virtully zero-margin low-power wireless receiver systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 12, pp. 1764–1777, 2014 (cit. on p. 31).

[139] D. Banerjee, S. K. Devarakond, X. Wang, S. Sen, and A. Chatterjee, "Real-time use-aware adaptive RF transceiver systems for energy efficiency under BER constraints," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 8, pp. 1209–1222, 2015 (cit. on p. 31).

[140] S. Lee, C. Shi, J. Wang, A. Sanabria, H. Osman, J. Hu, and E. Sánchez-Sinencio, "A built-in self-test and *In Situ* analog circuit optimization platform," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 10, pp. 3445–3458, 2018 (cit. on pp. 31, 60).

[141] M. Ingels, V. Giannini, J. Borremans, *et al.*, "A 5 mm^2 40 nm LP CMOS transceiver for a software-defined radio platform," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 12, pp. 2794–2805, 2010 (cit. on pp. 32, 39, 59).

[142] S. Li, J. Li, X. Gu, H. Wang, C. Li, J. Wu, and M. Tang, "Reconfigurable All-Band RF CMOS Transceiver for GPS/GLONASS/-Galileo/Beidou With Digitally Assisted Calibration," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 9, pp. 1814–1827, 2015 (cit. on pp. 32, 39, 59).

[143] D. Haghighitalab, D. Belfort, A. Kilic, A. Benlarbi-Delai, and H. Aboushady, "A 2.4 GHz ISM-band highly digitized receiver based on a variable gain LNA and a subsampled Sigma-Delta ADC," *Analog Integrated Circuits and Signal Processing*, vol. 95, no. 2, pp. 259–270, 2018 (cit. on pp. 32, 39).

[144] G. Huertas, D. Vázquez, E. J. Peralías, A. Rueda, and J. L. Huertas, "Testing mixed-signal cores: a practical oscillation-based test in an analog macrocell," *IEEE Design & Test of Computers*, vol. 19, no. 6, pp. 73–82, 2002 (cit. on pp. 36, 58).

[145] L. Abdallah, H.-G. Stratigopoulos, S. Mir, and J. Altet, "Defect-oriented non-intrusive RF test using on-chip temperature sensors," in *Proc. IEEE VLSI Test Symposium*, 2013 (cit. on pp. 36, 59).

[146] A. Coyette, B. Esen, W. Dobbelaere, R. Vanhooren, and G. Gielen, "Automatic generation of test infrastructures for analog integrated circuits by controllability and observability co-optimization," *Integration, the VLSI Journal*, vol. 55, pp. 393–400, 2016 (cit. on pp. 36, 58, 70).

[147] M. Ince, E. Yilmaz, W. Fu, J. Park, K. Nagaraj, L. Winemberg, and S. Ozev, "Digital built-in self-test for phased locked loops to enable fault detection," in *IEEE European Test Symposium*, 2019 (cit. on pp. 36, 59).

[148] A. Pavlidis, M. .-M. Louërat, E. Faehn, A. Kumar, and H. .-G. Stratigopoulos, "SymBIST: symmetry-based analog and mixed-signal built-in self-test for functional safety," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2021 (cit. on pp. 36, 58).

[149] S. Sunter, K. Jurga, and A. Laidler, "Using mixed-signal defect simulation to close the loop between design and test," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 12, pp. 2313–2322, 2016 (cit. on pp. 36, 73).

[150] V. Zivkovic and A. Schaldenbrand, "Requirements for industrial analog fault-simulator," in *International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design*, 2019, pp. 61–64 (cit. on p. 36).

[151] P. Subramanyan, S. Ray, and S. Malik, "Evaluating the security of logic encryption algorithms," in *Proc. IEEE International Symposium on Hardware Oriented Security and Trust*, 2015 (cit. on p. 37).

[152] A. Ashry and H. Aboushady, "A 4th order 3.6GS/s RF Sigma-Delta ADC with a FoM of 1pJ/bit," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 10, pp. 2606–2617, 2013 (cit. on p. 40).

[153] A. Sayed, T. Badran, M.-M. Louërat, and H. Aboushady, "1.5-to-3.0 GHz tunable RF ΣΔ ADC with a fixed set of coefficients and a programmable loop delay," *IEEE Transactions on Circuits and Systems - II: Express Briefs*, vol. 67, no. 9, pp. 1559–1563, 2020 (cit. on pp. 44, 59, 75).

[154] B. Shakya, T. He, H. Salmani, D. Forte, S. Bhunia, and M. Tehranipoor, "Benchmarking of hardware trojans and maliciously affected circuits," *Journal of Hardware and Systems Security*, vol. 1, no. 1, pp. 85–102, 2017 (cit. on pp. 57, 64).

[155] A. Chatterjee, "Concurrent error detection and fault-tolerance in linear analog circuits using continuous checksums," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 1, no. 2, pp. 138–150, 1993 (cit. on p. 58).

[156] H.-G. D. Stratigopoulos and Y. Makris, "Concurrent detection of erroneous responses in linear analog circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 5, pp. 878–891, 2006 (cit. on p. 59).

[157] G. Renaud, M. Diallo, M. J. Barragan, and S. Mir, "Fully differential 4-V output range 14.5-ENOB stepwise ramp stimulus generator for on-chip static linearity test of ADCs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 2, pp. 281–293, 2019 (cit. on p. 59).

[158] F. Azais, S. Bernard, Y. Bertrand, and M. Renovell, "Optimizing sinusoidal histogram test for low cost ADC BIST," *Journal of Electronic Testing: Theory and Applications*, vol. 17, no. 3-4, pp. 255–266, 2001 (cit. on p. 59).

[159] G. Renaud, M. J. Barragan, A. Laraba, H.-G. Stratigopoulos, S. Mir, H. L. Gall, and H. Naudet, "A 65nm CMOS ramp generator design and its application towards a BIST implementation of the reduced-code static linearity test technique for pipeline ADCs," *Journal of Electronic Testing: Theory and Applications*, vol. 32, no. 4, pp. 407–421, 2016 (cit. on p. 59).

[160] T. Chen, X. Jin, R. L. Geiger, and D. Chen, "USER-SMILE: ultrafast stimulus error removal and segmented model identification of linearity errors for ADC built-in self-test," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 7, pp. 2059–2069, 2018 (cit. on p. 59).

[161] B. Dufort and G. W. Roberts, "On-chip analog signal generation for mixed-signal built-in self-test," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 3, pp. 318–30, 1999 (cit. on p. 59).

[162] H. Malloug, M. J. Barragan, and S. Mir, "Practical harmonic cancellation techniques for the on-chip implementation of sinusoidal signal generators for mixed-signal BIST applications," *Journal of Electronic Testing: Theory and Applications*, vol. 34, no. 3, pp. 263–279, 2018 (cit. on p. 59).

[163] M. Barragan, R. Alhakim, H.-G. Stratigopoulos, M. Dubois, S. Mir, H. L. Gall, N. Bhargava, and A. Bal, "A fully-digital BIST wrapper based on ternary test stimuli for the dynamic test of a 40nm CMOS 18-bit stereo audio ΣΔ ADC," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 11, pp. 1876–1888, 2016 (cit. on p. 59).

[164] H. Chauhan, Y. Choi, M. Onabajo, I.-S. Jung, and Y.-B. Kim, "Accurate and efficient on-chip spectral analysis for built-in testing and calibration approaches," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 3, pp. 49–506, 2014 (cit. on p. 59).

[165] S. Sunter and A. Roy, "On-chip digital jitter measurement, from megahertz to gigahertz," *IEEE Design & Test of Computers*, vol. 21, no. 4, pp. 314–321, 2004 (cit. on p. 59).

[166] H. Le-Gall, R. Alhakim, M. Valka, S. Mir, H. Stratigopoulos, and E. Simeu, "High frequency jitter estimator for SoCs," in *IEEE European Test Symposium*, 2015 (cit. on p. 59).

[167] J.-S. Yoon and W. R. Eisenstadt, "Embedded loopback test for RF ICs," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 5, pp. 1715–1720, 2005 (cit. on p. 59).

[168] A. Valdes-Garcia, J. Silva-Martinez, and E. Sanchez-Sinencio, "On-chip testing techniques for RF wireless transceivers," *IEEE Design & Test of Computers*, vol. 23, no. 4, pp. 268–277, 2006 (cit. on p. 59).

[169] E. S. Erdogan and S. Ozev, "Detailed characterization of transceiver parameters through loop-back-based BiST," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 6, pp. 901–911, 2010 (cit. on p. 59).

[170] M. Cimino, H. Lapuyade, Y. Deval, T. Taris, and J.-B. Bégueret, "Design of a 0.9v 2.45 GHz self-testable and reliability-enhanced CMOS LNA," *IEEE Journal on Solid-State Circuits*, vol. 43, no. 5, pp. 1187–1194, 2008 (cit. on p. 59).

[171] Y.-C. Huang, H.-H. Hsieh, and L.-H. Lu, "A built-in self-test technique for RF low-noise amplifiers," *IEEE Transactions on Microwave Theory and Techniques*, vol. 56, no. 2, pp. 1035–1042, 2008 (cit. on p. 59).

[172] L. Abdallah, H.-G. Stratigopoulos, S. Mir, and C. Kelma, "Experiences with non-intrusive sensors for RF built-in test," in *Proc. IEEE International Test Conference*, Paper 17.1, 2012 (cit. on p. 59).

[173] A. Antonopoulos, G. Volanis, Y. Lu, and Y. Makris, "Post-production calibration of analog/RF ICs: recent developments and a fully integrated solution," in *International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design*, 2019, pp. 77–80 (cit. on p. 60).

[174] S. Sunter, J.-F. Côté, and J. Rearick, "Streaming access to ADCs and DACs for mixed-signal ATPG," *IEEE Design & Test*, vol. 33, no. 6, pp. 38–45, 2016 (cit. on pp. 60, 62).

[175] M. Portolan, "Automated testing flow: the present and the future," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 10, pp. 2952–2963, 2020 (cit. on p. 61).

[176] "IEEE standard for access and control of instrumentation embedded within a semiconductor device," *IEEE Std 1687-2014*, 2014 (cit. on p. 61).

[177] "IEEE standard for a mixed-signal test bus," *IEEE Std 1149.4-2010 (Revision of IEEE Std 1149.4-1999)*, 2011 (cit. on p. 61).

[178] "IEEE standard for describing analog test access and control, https://standards.ieee.org/project/1687_2.html," *IEEE Std P1687.2*, (cit. on p. 61).

[179] B. Yang, K. Wu, and R. Karri, "Secure scan: a design-for-test architecture for crypto chips," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 10, pp. 2287–2293, 2006 (cit. on p. 63).

[180] J. Da Rolt, G. Di Natale, M. Flottes, and B. Rouzeyre, "New security threats against chips containing scan chain structures," in *IEEE International Symposium on Hardware-Oriented Security and Trust*, 2011 (cit. on p. 63).

[181] L. Azriel, R. Ginosar, and A. Mendelson, "Revealing on-chip proprietary security functions with scan side channel based reverse engineering," in *Proc. Great Lakes Symposium on VLSI*, 2017, pp. 233–238 (cit. on p. 63).

[182] I. M. Breeuwsma, "Forensic imaging of embedded systems using JTAG (boundary-scan)," *Digital Investigation*, vol. 3, no. 1, pp. 32–42, 2006 (cit. on p. 63).

[183] F. Majeric, B. Gonzalvo, and L. Bossuet, "JTAG combined attack - another approach for fault injection," in *IFIP International Conference on New Technologies, Mobility and Security*, 2016, pp. 1–5 (cit. on p. 63).

[184] K. Rosenfeld and R. Karri, "Attacks and defenses for JTAG," *IEEE Design Test of Computers*, vol. 27, no. 1, pp. 36–47, 2010 (cit. on pp. 63, 67).

[185] E. Valea, M. D. Silva, G. D. Natale, M.-L. Flottes, and B. Rouzeyre, "A survey on security threats and countermeasures in IEEE test standards," *IEEE Design & Test*, vol. 36, no. 3, pp. 95–116, 2019 (cit. on pp. 63, 67).

[186] F. Novak and A. Biasizzo, "Security extension for IEEE std 1149.1.," *Journal of Electronic Testing*, vol. 22, pp. 301–303, 2006 (cit. on p. 67).

[187] A. Das, J. D. Rolt, S. Ghosh, S. Seys, S. Dupuis, G. D. Natale, M.-L. Flottes, B. Rouzeyre, and I. Verbauwhede, "Secure JTAG implementation using schnorr protocol," *Journal of Electronic Testing*, vol. 29, pp. 193–209, 2013 (cit. on p. 67).

[188] R. Baranowski, M. A. Kochte, and H. Wunderlich, "Fine-grained access management in reconfigurable scan networks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 6, pp. 937–946, 2015 (cit. on p. 67).

[189] J. Dworak, A. Crouch, J. Potter, A. Zygmontowicz, and M. Thornton, "Don't forget to lock your SIB: hiding instruments using P1687," in *IEEE International Test Conference*, 2013 (cit. on p. 67).

[190] C. Clark, "Anti-tamper JTAG TAP design enables DRM to JTAG registers and P1687 on-chip instruments," in *IEEE International Symposium on Hardware-Oriented Security and Trust*, 2010, pp. 19–24 (cit. on p. 67).

[191] S. Kan, J. Dworak, and J. G. Dunham, "Echeloned IJTAG data protection," in *IEEE Asian Hardware-Oriented Security and Trust*, 2016 (cit. on p. 67).

[192] J. Porte, *Outil pour la conception et l'enseignement d'électronique analogique (OCEANE)*, https://www-soc.lip6.fr/equipe-cian/logiciels/oceane/, Online (cit. on p. 68).

[193] S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana, "The SpiNNaker Project," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 652–665, 2014 (cit. on p. 81).

[194] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014 (cit. on p. 81).

[195] M. Davies, N. Srinivasa, T. Lin, *et al.*, "Loihi: a neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018 (cit. on p. 81).

[196] L. A. Camuñas-Mesa, Y. L. Domínguez-Cordero, A. Linares-Barranco, T. Serrano-Gotarredona, and B. Linares-Barranco, "A configurable event-driven convolutional node with rate saturation mechanism for modular convnet systems implementation," *Frontiers in Neuroscience*, vol. 12, p. 63, 2018 (cit. on p. 81).

[197] W. Maass, "Networks of spiking neurons: the third generation of neural network models," *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, 1997 (cit. on p. 81).

[198] L. A. Camuñas-Mesa, B. Linares-Barranco, and T. Serrano-Gotarredona, "Spiking neural networks and their memristor-CMOS hardware implementations," *Materials*, vol. 12, no. 17, p. 2745, 2019 (cit. on p. 81).

[199] G. Indiveri, B. Linares-Barranco, T. Hamilton, *et al.*, "Neuromorphic silicon neuron circuits," *Frontiers in Neuroscience*, vol. 5, 2011, Article 73 (cit. on pp. 82, 93).

[200] C. Mead, *Analog VLSI and Neural Systems*. Addison Wesley, 1989 (cit. on p. 82).

[201] S. Katzenbeisser, U. Kocabaş, V. Rožić, A.-R. Sadeghi, I. Verbauwhede, and C. Wachsmann, "PUFs: myth, fact or busted? a security evaluation of physically unclonable functions (PUFs) cast in silicon," in *Cryptographic Hardware and Embedded Systems – CHES 2012*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 283–301 (cit. on p. 85).

[202] A. Maiti, V. Gunreddy, and P. Schaumont, "A systematic method to evaluate and compare the performance of physical unclonable functions," in *Embedded Systems Design with FPGAs*, N. S. (P. Athanas D. Pnevmatikatos, Ed., New York, NY: Springer, 2013, pp. 245–267 (cit. on p. 85).

[203] Y. Liu, L. Wei, B. Luo, and Q. Xu, "Fault injection attack on deep neural network," in *2017 IEEE/ACM International Conference on Computer-Aided Design ICCAD*, 2017, pp. 131–138 (cit. on p. 94).

[204] Z. Liu, J. Ye, X. Hu, H. Li, X. Li, and Y. Hu, "Sequence triggered hardware trojan in neural network accelerator," in *2020 IEEE 38th VLSI Test Symposium (VTS)*, 2020, pp. 1–6 (cit. on p. 94).

[205] H. Naghibijouybari, A. Neupane, Z. Qian, and N. Abu-Ghazaleh, "Rendered insecure: gpu side channel attacks are practical," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 2139–2153 (cit. on p. 94).

[206] B. Salami, E. B. Onural, I. E. Yuksel, F. Koc, O. Ergin, A. Cristal Kestelman, O. Unsal, H. Sarbazi-Azad, and O. Mutlu, "An experimental study of reduced-voltage operation in modern FPGAs for neural network acceleration," in *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2020, pp. 138–149 (cit. on p. 94).

[207] J. Wei, Y. Zhang, Z. Zhou, Z. Li, and M. A. Al Faruque, "Leaky DNN: stealing deep-learning model secret with GPU context-switching side-channel," in *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2020, pp. 125–137 (cit. on p. 94).