PUFsecurity

Anti-Tampering Designs in Hardware Security

Meng-Yi Wu

Kent Kai-Hsin Chuang

Book Series on Hardware Security

Founding Editor: Charles Ching-Hsiang Hsu

Anti-Tampering Designs in Hardware Security

Meng-Yi Wu, Kent Kai-Hsin Chuang

Book Series on Hardware Security

Founding Editor: Charles Ching-Hsiang Hsu

Quantum Tunneling PUF: Basics, Circuits, and Security Applications

By Kent Kai-Hsin Chuang

PUF-based Security Solutions and Applications

By Lawrence Liu

Anti-Tampering Designs in Hardware Security

By Meng-Yi Wu, Kent Kai-Hsin Chuang

Random Number: Generation, Emulation, and Practice

By Balance Chun-Heng You, Kent Kai-Hsin Chuang

Contemporary Cryptography: Fundamentals and Algorithms

By Chun-Yuan Yu, Danny Yung Chih Chen, Wayne Wen-Ching Lin

TPM and HSM: Implementation and Application

By Shih-Li Hsu

Anti-Tampering Designs in Hardware Security

Written by Meng-Yi Wu, Kent Kai-Hsin Chuang
Foreword by Charles Ching-Hsiang Hsu
Edited by Ada Ying-Yun Huang, Andrew Irvin, Ann Yi-An Lin, Evans Ching-Song Yang

Printed by
PUFsecurity Corporation

First Edition, 2024

PUFsecurity

About the Author

Meng-Yi Wu

Dr. Meng-Yi Wu joined PUFsecurity Corporation in 2019 and is the R&D Director, leading the team in research and development in physical unclonable functions (PUF), secure storage, anti-tampering IP, and security chip design. Before joining PUFsecurity, Dr. Wu worked at eMemory Technology as the Department Manager of Technology Development on NeoFuse and the Director of the NeoPUF Product Line. He was one of the main inventors of both NeoFuse and NeoPUF, which won the ISSCC 2018 Takuo Sugano Far East Best Paper Award.

Dr. Wu received his Ph.D. degree in electrical engineering from Taiwan National Tsing Hua University in 2006. His expertise is in semiconductor devices, non-volatile memory of trapped-charge and conventional Flash Memory, embedded OTP (one-time-program) memory, and PUF technology. His current research interests include memory, PUF, and hardware security. He holds over 200 semiconductor-related patents and has published over 10 academic papers on semiconductors. Several of his publications include “Quantum Tunneling PUF: A Chip Fingerprint for Hardware Security”, “A PUF Scheme Using Competing Oxide Rupture with Bit Error Rate Approaching Zero”, and “Highly Reliable Anti-fuse Technology in Sub-16nm Technologies for Security Applications.”

Kent Kai-Hsin Chuang

Dr. Kent Kai-Hsin Chuang joined PUFsecurity Corporation in 2020, where he currently holds the position of a R&D manager in security technology. His current research interest includes Root of Trust circuits, PUF-based security solutions, and attack/protection techniques for cryptographic implementations.

Dr. Chuang received his Ph.D. degree in electrical engineering from KU Leuven, Belgium, in 2020. During his Ph.D. study, he worked in the COSIC research group of KU Leuven and the reliability research group of IMEC, where he was in charge of developing highly reliable PUFs in CMOS and emerging memory technologies. He holds 14 worldwide issued and pending patents and has more than 15 international publications. Several of his publications include “Highly Reliable Physically Unclonable Functions” and “A Physically Unclonable Function Soft Oxide Breakdown Featuring 0% Native BER and 51.8 fJ/bit in 40-nm CMOS.” He has been invited as a tutorial speaker at two IEEE conferences and also served as a reviewer of IEEE journals more than 20 time.

Foreword

“I was seldom able to see an opportunity until it had ceased to be one.”

For many years, this quote from Mark Twain has been imprinted in my mind, reminding me that good ideas, whenever they arrive, must be seized to become genuine opportunities. So when I met Tang Ma (馬騰桂) in 2015, shortly after his retirement from Maxim, his explanation of the possibilities of PUF (Physically Unclonable Function) for semiconductors, and the critical role it could play in security, felt like an opportunity worth pursuing.

As a semiconductor device physicist, I immediately thought that the characteristics associated with a device’s physical dimensions could potentially produce unclonable features, due to the natural variances during the fabrication process. The thickness and quality of the film, for example, or the length and breadth of a transistor gate, parameters such as those produce minute differences in a transistor’s microstructure. When extracting and comparing the adjacent transistors’ electrical characteristics, we can register the difference as 0 or 1 if it is distinct enough within the measurable region.

So, the critical factor is identifying the parameter with a linear difference in its geometry, yet the electrical behavior measured from the terminals is exponential. Drawing on my decades of work on electron transport in thin gate dielectrics, I immediately thought that the tunneling current between gate dielectrics would be

significantly different, even if the thickness or quality of the dielectric retains only minimal variations.

To verify this idea, I brought Wei-Jer (翁偉哲), Meng-Yi (吳孟益), Hsin-Ming (陳信銘), and Evans (楊青松) to collaborate together, by setting up an experiment that used our existing NeoFuse (anti-fuse using oxide tunneling) with two oxide capacitors in parallel. The results produced two critical findings:

1. *Only one of the oxide capacitors will have a significant tunneling current when the applied voltage is large enough.*
2. *The occurrence of the tunneling current between the pair of oxide capacitors is random, arriving equally on either the left or right capacitor.*

I knew this technology would be foundational for securing the future of interconnected computing. Since then, these two critical findings have established the silicon fingerprint, NeoPUF, which led to the founding of our company, PUFsecurity, and the development of our integrated suite of security subsystems that play a vital role in the Hardware Security ecosystem. After establishing PUFsecurity, we were encouraged to try and enlighten the broader semiconductor community on the risk of unsecured chips through the education and promotion of Quantum Tunneling PUF, which ultimately led to publishing this book series to introduce the fundamentals of PUF-based Hardware Security.

These books will cover a wide range of topics, including quantum tunneling PUF, an overview of PUF-based solutions, tamperproof

design, random number generation, and the importance of cryptography to Hardware Security. We will also cover applications like AI and IoT, as well as provide an overview of the latest security standards and regulations. Our hope is that they can play a role in furthering our collective understanding of Hardware Security and its impact on the future of computing.

As the internet enables more and more connected devices, deploying a traditional physically secure boundary for a system is no longer sufficient. We must embrace the principles of “Zero Trust,” explicitly verifying every linked device and decentralizing the secure boundaries of the network to a solution embedded on each device. It is similar to how we biometrically identify each of us through our fingerprints.

A PUF can play the role of a chip fingerprint, uniquely identifying both the device and the chip in which it is embedded. It can then be integrated with an NVM OTP to establish a Hardware Root of Trust that can generate, store, and safely manage Keys. Then, we can implement a Security Coprocessor by combining a PUF-based Hardware Root of Trust with Crypto Engines to provide a device with a fully integrated Security Subsystem.

I want to offer my sincerest gratitude to everyone that helped realize this project, in particular, its authors; Dr. Kent Chuang (莊愷莘), Lawrence Liu (劉持志), Dr. Meng-Yi Wu (吳孟益), Balance You (游鈞恆), Danny Chen (陳勇志), Dr. Wayne Lin (林文景), Chun-Yuan Yu (游鈞元), Dr. Li Hsu (徐世理) and Matthew Yu (于立宏). I would also like to thank Ada Huang (黃楹芸), Andrew Irvin

(張安筑), and Ann Lin (林奕安) for their diligent work editing, formatting, and publishing these books. Lastly, I would like to thank Dr. Evans Yang (楊青松) for his tireless dedication in guiding this project.

Thank you all.

Charles Ching-Hsiang Hsu
April 2023

Preface

Kerckhoff's Principle: A cryptosystem should be secure even if everything about the system, except the key, is public knowledge.

"The enemy knows the system being used [1]." As stated by Claude Shannon, if the only secret in a secure system, which is the key, is revealed, the entire system becomes insecure. In other words, even when the most advanced encryption system is utilized, such as post-quantum cryptography (PQC), there is no security if the key is leaked. Therefore, protecting the key becomes crucial and fundamental in a secure system.

In a secure chip, the keys, which constitute the hardware root of trust (HRoT) of the chip, must be fully protected. An HRoT is the most important design in chip security, providing the fundamental trust (root keys), hardware identification code (UID), hardware unique keys (HUK), and entropy. Consequently, it often becomes the target of hacker attacks. If there are no effective designs to prevent attacks, the keys that serves as the foundation for secure operations can be easily obtained by attackers, leading to various security issues in applications, such as identity authentication cracking, data encryption breaking, and theft of product design know-how.

There are three criteria that need to be considered when designing an HRoT:

1. Secure Storage: To store important security information of the chip, a storage unit must include access control permissions and management, data obfuscation, and data encryption for data read and write operations to prevent information from being read out during power-on or power-off situations, as well as from electrical and physical reverse engineering.
2. Trusted Environment: The design must include dedicated logic circuits and registers. Other auxiliary circuits are also required to detect abnormal behaviors within the entire block. It is necessary to mitigate all possible vulnerabilities in the circuit design, or hackers could exploit techniques like power analysis to steal confidential data.
3. Authorized Secured Operation: The operation of the design must have permission control to determine whether the reading of confidential information is legal. It prevents hackers from controlling or obtaining important information during operation by creating errors in the chip's logic functions through fault injection.

Therefore, a qualified HRoT requires a comprehensive anti-tampering design and a complete security policy to protect the entire system. Whether chip protection design can guarantee secured operation or not, relevant design guidelines and certification standards must be referenced. The certification from a credible third-party security certification laboratory must be obtained to confirm that anti-tampering designs effectively reduce attack risks and enhance chip operation security. Such assessments typically involve white-box design and physical chips

for conducting penetration tests to evaluate the risks of confidential data protection across various aspects, such as components, circuits, and functional designs. The security report or certificate will be issued by the certification laboratory after passing the security risk assessment.

In this book, we will take PUFsecurity's product PUFrt as an example, which features one-time programmable (OTP), static entropy NeoPUF, and dynamic random number (TRNG). Starting with introducing the system architecture of memory, the book will state various attack methods on chips and the threat models they generate. The countermeasures and protective technologies based on various threats will then be provided. We hope our readers gain a thorough knowledge of HRoT design and understand the design concepts of a secure system as well as what to be aware of.

Meng-Yi Wu
February 2024

Hardware Security Overview

We are all eagerly embracing the technology revolution underway with the Internet of Things (IoT), artificial intelligence (AI), e-finance, and electric vehicles. However, we need to safeguard against the security risks they create. While these technologies are rapidly evolving, the number of connected devices will soar in the near future. Securing increasingly sophisticated device networks has become a critical security topic. To accomplish the security goals, it is important to make connected devices secure by design, which requires intensive effort to make hardware security a fundamental part of the conceptual stage for electronic devices and systems.

The primary consideration for hardware security is the use of dedicated components as gatekeepers in a system. This leads to three important research topics in this field: hardware acceleration, countermeasures against attacks, and Hardware Roots of Trust. Implementing cryptographic algorithms in hardware is typically faster than in software. Especially for public-key encryption algorithms and signature schemes, performing operations in software may be unacceptably slow within some platforms. Hardware implementations for cryptographic algorithms are inherently more efficient and elegant.

For edge devices deployed anywhere in the world, the risks of hijacking by malicious parties and vulnerability to attack are high. Under such circumstances, a device, and the system on top of it, could be vulnerable even if standardized cryptographic algorithms

and security protocols are applied. This weakness can be exploited by advanced attack techniques, including side-channel attacks [2], fault attacks [3], and invasive physical attacks [4]. Consequently, we need to understand possible vulnerabilities caused by new attack techniques and design relevant hardware countermeasures.

A Hardware Root of Trust (HRoT) should be the first element in the Chain of Trust of a security system to adequately protect the physical layer. It should provide an unpredictable and tamperproof secret that enables the required hardware security features, which, historically raises security issues around generation and storage.

Hardware Security Book Series Overview

PUFsecurity Corporation published a Hardware Security Book Series, covering a wide range of topics on hardware security. The book series starts with introducing the essential knowledge about Physically Unclonable Functions (PUFs) and PUF-based Root of Trust (RoT) solutions. The next milestone is understanding how security applications can benefit from high-quality PUF and RoT. The next book in the series introduces several integrated PUF-based IPs as a fully comprehensive security solution. The discussions of these solutions will cover hardware architecture, functionality, specifications, and important use cases in Artificial Intelligence (AI) and the Internet of Things (IoT).

More details about generic RoT solutions using NeoPUF, a technology developed by eMemory Technology, are covered in two following books. One focuses on anti-tampering features of

NeoPUF and RoT solutions, including design techniques applied to mitigate common security threats. The other book describes true random number generators, their essential concepts, and demonstrations of their design integration within RoT solutions.

Three types of cryptography are introduced given their theory and design methods in circuit implementation, which include Cryptographic Hash Functions, Symmetric-Key Ciphers, and Public-Key Cryptography. It is crucial to understand the significant differences between algorithms executed in software compared to those in hardware. Finally, two modules, Trusted Platform Module (TPM) and Hardware Security Module (HSM), are illustrated in the last book. Through the specification explanation, we will learn the types of TPMs and HSMs that are secure enough to protect keys during the lifecycle and learn methods to improve the relevant designs of the modules.

This book series addresses the methodologies, applications, and market insights related to the core technology in hardware security comprehensively. It serves as a practical and handy tool for readers at all levels, from beginners to experts. Readers can acquire better and deeper understandings about PUF, RoT, PUF-based solutions, and hardware security, which will further assist them in pursuing excellence in academia and industry.

Contents

List of Figures

List of Tables

1

Why is chip security important? Chips store the software necessary for a device to operate. Vulnerabilities in chips can potentially lead to tampering with the pre-running software, thereby altering the protective measures and permission control originally inherent in the software. This chapter provides an overview of each chapter in the book, giving readers a preliminary understanding of the subject matter.

1. Introduction

In recent years, the rapid advancement of physical tampering techniques has posed severe security threats to chips and systems. To prevent tampering attempts that could breach security measures, anti-tampering designs must be implemented across all layers of the security system. Serving as the root of trust for chip security, one of the hard design requirements for PUFrt is its tamper-proof nature [5]. By thoroughly studying the anti-tampering designs of PUFrt, this book aims to emphasize the importance of anti-tampering measures and demonstrate how these countermeasures effectively ensure the tamper-proof integrity of PUFrt.

1.1. Book Overview

To provide a comprehensive overview of tamper-proof designs, this book is organized as follows:

The first half of the book includes the introduction of anti-tampering designs. In Chapter 2, it discusses the basics of physical attacks, explaining the reasons why we need the chip design to be tamper-proof. Chapter 3 outlines the general requirements for using a NVM secure macro as the root of trust in an SoC. Chapter 4 provides a high-level overview of the anti-tampering designs implemented in the PUFrt, offering detailed explanations of each anti-tampering design in both the hard macro and the RTL wrapper.

The second half of the book focuses on experiment results concerning the resilience of PUFrt against physical attacks in practices. Chapter 5 presents findings from physical inspections of the OTP and PUF macros. Chapter 6 gives a few examples of side-channel attacks performed on PUFrt. Chapter 7 details the procedure and results of attempts to attack PUFrt using the InGaAs photoemission microscopy technique. Finally, Chapter 8 concludes the book.

2

Starting with an overview of attack types, this chapter provides brief introductions and examples of three different methods used to attack chips: invasive attacks, semi-invasive attacks, and non-invasive attacks. This foundational knowledge will help readers grasp the basic concepts necessary for understanding subsequent defense measures.

2. Basic of Physical Attacks

The quest for tamper-proof design stems from the vulnerability to physical attacks that can compromise security systems, a concern documented extensively in literature since the 1990s [6]-[13]. Unlike cyberattacks, which often exploit vulnerabilities in application or protocol layers, physical attacks target the hardware itself by monitoring or modifying its physical behaviors. Even though physical attacks typically impose more environmental restrictions and resource requirements compared to cyberattacks, their potential consequences cannot be underestimated. Moreover, implementing hardware fixes after mass production poses significant challenges due to these attacks' invasive nature.

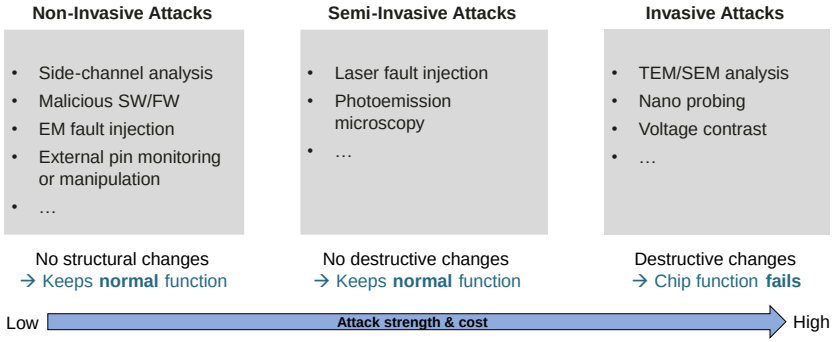

Figure 2-1 Categorization and examples of physical attacks.

Physical attacks can be divided into three categories: non-invasive attacks [6],[7], semi-invasive attacks [8]-[11], and invasive attacks [12],[13], depending on the required level of penetration to the

target. The major differentiations and examples of these three types of attacks are illustrated in *Figure 2-1*. This chapter will briefly introduce the three different types of physical attacks and common countermeasures. More detailed countermeasures will be discussed in chapter five.

2.1. Invasive Attacks

An invasive attack, also known as a penetration attack, is a type of physical attack where the attacker gains access to a system or network by physically entering a restricted area or breaking through physical barriers. The goal of an invasive attack is to bypass security measures that are in place to protect the system and extract sensitive information or cause damage to the system.

Invasive attacks differ from semi-invasive and non-invasive attacks in terms of the level of physical access required. Semi-invasive attacks involve physical access to the device but do not require breaking through physical barriers, such as opening the device's casing to access the internal components. Non-invasive attacks, on the other hand, do not require physical access to the device at all and can be executed remotely over a network or through social engineering tactics. The invasive is the most powerful type of attack among these three. Despite being very powerful, the invasive attack has the least applicability due to its high resource consumption and its damage to the target.

Figure 2-2 shows some examples of the required techniques to perform physical attacks, including the sample preparations

needed before the actual attacks. An invasive attack targeting a chip requires its package to be removed or drilled, and it typically also requires substrate thinning and metal delayering. After the sophisticated sample preparation process, the chip can then be subjected to microscopic investigation using a scanning electron microscope (SEM) [14], transmission electron microscope (TEM) [15], passive voltage contrast (PVC) [16], or other similar equipment. It is also possible to directly probe certain signal lines through micro-probing techniques [17] or even add/modify signal lines using Focused Ion Beam (FIB) techniques [18].

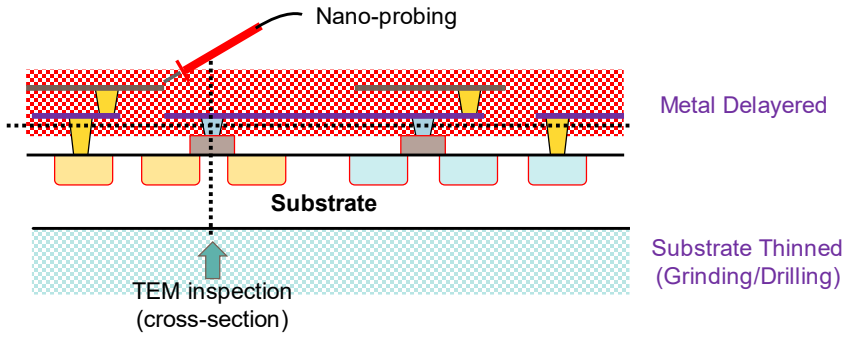

Figure 2-2 Examples of invasive attacks and the required sample preparations.

A successful invasive attack may allow access to the forbidden region of a chip, providing an attacker the ability to monitor private signal lines, change values of the internal registers, or scan out the protected memory contents. On the other hand, invasive attacks have poor practicability due to the high requirements on the equipment and expertise. Moreover, since most invasive

attacks will leave unrecoverable damages to the targets and make them nonfunctional, their practicability is quite limited.

To protect the chip or device against invasive attacks, one common methodology is to apply anti-tampering packaging techniques, such as adding an anti-tamper seal around the chip [20],[21] to block physical accesses or using back-to-back packaging [22] to prevent accesses from the backside of the chip. There are also chip-level countermeasures that can be applied, such as metal shielding techniques that can prevent invasive attacks that require metal delayering. Using circuit elements that are intrinsically immune to microscopic investigation is also an effective countermeasure, such as replacing the visible eFuse cells with invisible anti-fuse cells [24]. A total solution that is highly resilient to invasive attacks usually requires a combination of these approaches.

2.2. Semi-invasive Attacks

The physical attacks that fit into the second category are semi-invasive attacks. As shown in *Figure 2-3,* semi-invasive attacks require fewer resources but are less powerful than invasive attacks. Preparing the samples still requires a certain level of penetration, such as drilling through the package and grinding the silicon substrate. The main difference in this step is that it requires keeping the chip functional after sample preparation. Afterward, the attacker may inject laser pulses [9], supply voltage glitches [25], or clock glitches [26] into the chip. One approach is to use these means to change the internal states of the chip, aiming for

bypassing or disabling the security measures. The other approach is to monitor the responses of the chip in accordance with these injection activities, and consequently, the internal behavior or secret parameters within the chip may be captured using statistical analysis.

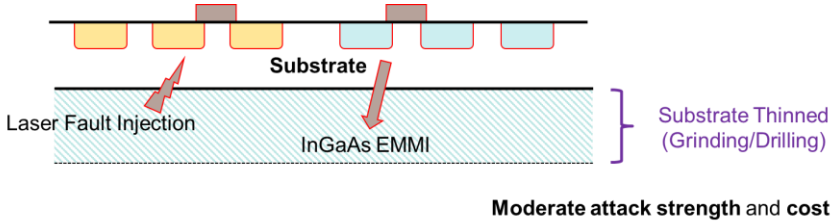

Figure 2-3 Examples of semi-invasive attacks and the required sample preparations.

A successful semi-invasive attack may lead to security breaches such as unauthorized access, malicious code execution, sensitive data and key leaks, function failures, and other severe consequences. Compared to invasive attacks, a semi-invasive attack has less capability of focusing on a specific target, that is, a single attack may affect multiple targets within a chip and make it more difficult for an attacker to have the desired consequences. On the other hand, since there are fewer requirements for sample preparation and equipment, semi-invasive attacks are more practical than invasive attacks.

To protect the chips from semi-invasive attacks, chip-level anti-tampering techniques are usually applied. For the faulty events injected by the attackers, one can add filters or sensors within the

chip to block or detect them [27]. In addition, one can also increase the error tolerance of the circuit by adding redundancies [28], which may prevent the faults from occurring and being exploited.

2.3. Non-invasive Attacks

A physical attack that does not require invasive penetration to its target is categorized into non-invasive attacks. Unlike invasive and semi-invasive attacks, a non-invasive attack can be performed without sample preparation. As shown in *Figure 2-4,* when performing the attack, an attacker will try to probe the external signal lines or use electromagnetic (EM) probes to sense or affect the internal behavior of the chip [29].

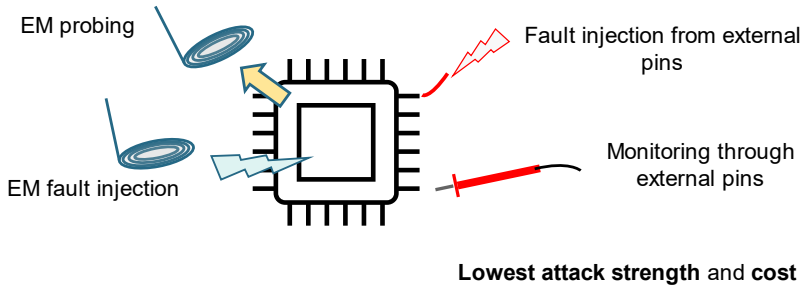

Figure 2-4 Examples of non-invasive attacks.

One type of non-invasive attack is also based on fault injection but without penetrating into the package or chip. For example, voltage or clock glitches may be injected by coupling them into the external signal lines using capacitive probes. Alternatively, EM pulses can also be injected into certain chip areas using EM probes [30].

Sometimes, it is also possible to induce similar consequences by software execution, such as the row-hammer attack that injects faults into the dynamic random-access memory (DRAM) cells by repeatedly accessing the adjacent rows [31]. Changing the ambient temperature is also a possible approach to induce faults or misbehavior in the chips, for example, the data remanence effect of the static random-access memory (SRAM) cells may be exploited [32]. By significantly lowering the chip temperature, sensitive contents stored in the SRAM may not be correctly erased after a full reset, making it possible for non-privileged users to read sensitive contents.

On the other hand, a more commonly practiced type of non-invasive attack is based on analyzing the side-channel leakages. While performing certain executions, the power dissipation, EM emission, or execution time may differ, leaving traces that reveal the internal behavior of the chip during such executions. This type of attack is known as the side-channel attack, which is usually used to help guess the secret keys used in cryptographic operations. If the same key is used to perform multiple cryptographic operations, an attacker may collect power/EM traces that correspond to known inputs or outputs. If leakage information resides in those traces, an attacker can use statistical techniques such as correlation analysis to narrow down the search space of the secret keys and eventually guess the secret key correctly [6],[7].

To protect the chips against non-invasive fault injection attacks, one can add countermeasures such as detection circuits, and it is

also helpful to make the circuit more environmentally invariant. To protect the chips against side-channel attacks, there are techniques to reduce information from leaking out through these unintended side channels. Commonly used techniques are adding analog filters [33], utilizing low-leakage circuit cells [34], and employing masking designs [35] in cryptographic implementations.

3

The hardware root of trust is foundational to device security. Activated upon the first power-up, it is crucial throughout the device's lifecycle. From the perspective of a hardware root of trust, this chapter discusses the necessary security features required and its threat models. We provide readers with a comprehensive understanding of the considerations needed for chip security design.

3. Hardware Root of Trust

Modern SoC designs typically incorporate embedded nonvolatile memory (NVM) solutions such as eFuse, OTP, or eFlash, which are circuit macros that are hardened in specific technology platforms and are thus denoted as the hard macro. A hard macro is primarily designed following a full-custom design flow. The hard macros mainly include memory cells in array forms, analog peripheral circuits, and digital controllers. The analog circuits include sense amplifiers, bandgap voltage generators, high-voltage charge pumps, etc. The digital controllers are typically simple logic in gate-level design, such as the address decoder and output registers.

When security features are required in an SoC, there are sensitive parameters that must be stored internally, and the embedded NVM macro is the natural choice for this purpose. However, a standalone NVM macro is typically not resilient to physical attacks, making it risky to store the sensitive parameters within the NVM macros. In order to fundamentally improve the security of an SoC, it is important to make sure the secrets are kept safe, which is done by transforming a regular NVM macro into a secure macro.

3.1. General Requirements

Conventional NVM hard macros usually incorporate proprietary interfaces and protocols, while the SoCs are usually in favor of

more standardized architecture and protocols, such as the Advanced Microcontroller Bus Architecture (AMBA) [36]. In order to allow communications between the host system and the hard macros in an SoC, it is necessary to design a controller between the main bus and the hard macros. The host system performs read-and-write instructions to the memory-mapped registers, and this controller converts the register instructions into signals that can drive the NVM macro to perform the intended operations. Shortly speaking, this controller is the translator between the standard protocol and the proprietary protocol. *Figure 3-1* illustrates how a traditional NVM macro is connected to the main SoC with the help of a digital controller designed in RTL, denoting the RTL wrapper.

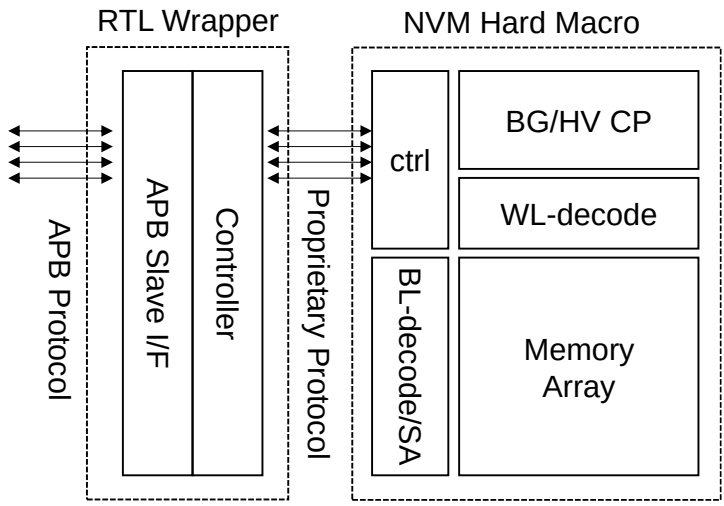

Figure 3-1 Conventional NVM IP.

The next section describes the security vulnerabilities typically encountered with such a conventional NVM design.

3.2. Design and Threats for Building a NVM Secure Macro

The NVM in the chip is used to protect the most important secrets, such as keys or secure information. Thus, the NVM in a secure chip is also often regarded as an HRoT (hardware root of trust) [37]. When an NVM storing critical assets is used, it may be exposed to several possible security threats. An attacker may use hardware or software attacks to attack the chip and try to gain access to these essential hardware secrets.

As far as the HRoT is concerned, it usually consists of non-volatile memory (e.g., eFuse, OTP, and eFlash) and a digital RTL design for the controller. Non-volatile memory is typically attacked in various ways. *Table 3-1* below shows a hypothetical threat model of NVM macro integration macros where the HRoT can be attacked, and the confidential data be obtained based on invasive, semi-invasive, and non-invasive approaches. In general, device-related or layout-related approaches are considered invasive or semi-invasive, and attackers can do this against the secure IP through a delayer or a FIB on visible pins [17],[18]. Modern ways of SCA (Side-channel Attacks) or fault injection are considered non-invasive attacks [6]-[11]. Attackers can analyze data by gathering large data to statistically distinguish power consumption or leakage during reading and writing, or use fault injection by laser,

power glitch, or clock glitch to make the macro generate misoperations to cause confidential data leaks. The next section will provide a brief introduction to these models.

Table 3-1 Threat model for the security macro as a hardware root of trust

Invasive Attacks	• SEM, FIB, TEM, or optical inspection (OBIC/OBIRCH) on NVM cells • Passive Voltage Contrast by FIB/SEM on NVM cells • Delayering and Nano-Probing
Semi-invasive Attacks	• Fault injection on secure registers to bypass security policies • Photon emission inspection (InGaAs/EMMI) during read operation
Non-invasive Attacks	• Side-channel Attacks by correlation/differential power analysis (CPA/DPA) • Unauthorized access to secret keys or sensitive data using malicious FW/SW • Fault injection through external pins to alter addresses or operation modes

3.3. Threat Models for Secure Macro

As listed in *Table 3-1*, there are several threats that are particularly impactful for the NVM secure macros. Under a successful attack, the NVM secure macro may leak sensitive information or enter a faulty state, which will then lead to security breaches. Operating principles and the resulting threat models against the NVM secure macro will be discussed in this section.

3.3.1. Invasive Attacks

Three main types of invasive attacks could potentially be exploited to reveal the content of the NVM. First, an attacker may decapsulate the chip and then remove the metal layers to expose the NVM cells for microscopic inspection. These steps may involve chemical and mechanical processing and FIB techniques.

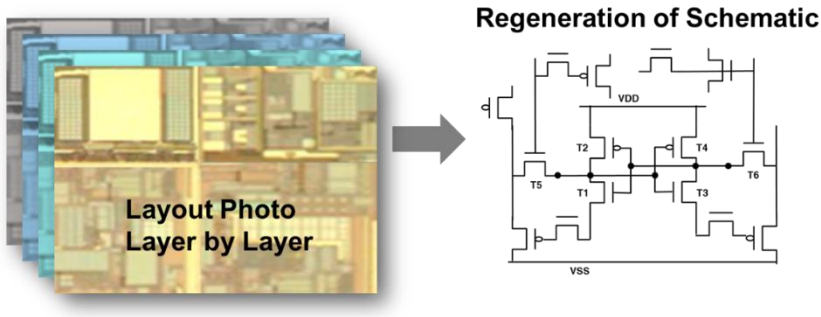

Figure 3-2 Illustration of how to perform reverse-engineering that recovers the internal design or secret within a chip.

Afterward, the attacker can utilize SEM, TEM, or OBIRCH techniques to inspect the NVM cell and then obtain images of these NVM cells. If the resulting images can be correlated to the logical values stored within the NVM cells, the attacker may be able to uncover the data stored within the NVM array.

Alternatively, the attacker can utilize passive voltage contrast (PVC) techniques to inspect the NVM cells. This technique requires injecting electrons or ions onto the NVM devices or the

interconnects using SEM or FIB. *Figure 3-3* illustrates the FIB technique. Using an electron detector, the power of reflected electrons at different injected locations can be recorded. By scanning through the NVM array, a 2D heat map can be obtained, showing the reflection ratio at different locations. Typically, such a heat map can reveal hidden conduction paths, or the charges stored within the floating-gate structure, which may relate to the logic values stored within the NVM cells. Consequently, the attacker may uncover the data stored within the NVM array using this technique.

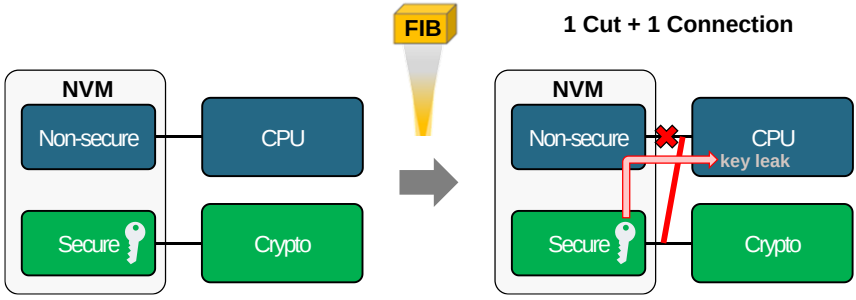

Figure 3-3 How does Focus Ion Beam (FIB) technique steals secrets.

In another scenario, the attacker may apply the nano-probing technique on the interconnects that are exposed after sample preparation. If the chip can still function in such a case, the attacker can try to probe the signal lines and then obtain the transmitted secret data. Otherwise, the attacker may need to use multiple probes to bias the NVM array and then monitor the output to obtain the stored secret data.

In summary, these techniques can target the NVM array of the secure macro directly to uncover the secret data stored within it, leading to serious security threats against the NVM secure macro.

3.3.2. Semi-invasive Attacks

Two main types of semi-invasive attacks are potentially threatful to the NVM secure macro. Different from the aforementioned invasive attacks, the minimum sample preparation requirement for these attacks is only IC decapsulation. Metal delayering or substrate thinning is not always required.

In the first attack scenario, the attacker may inject faults onto the secure macro, either affecting the IO of the NVM array or affecting the registers in the soft macro. There are various means to inject faults into the chip, including laser pulses, EM pulses, and voltage/clock glitches. One example of fault injection is shown in *Figure 3-4*. Because of the photoelectric effects, faults may be induced and side-channel leakage may be observed.

Note that some of these methods may be categorized as non-invasive attacks, depending on how closely they interact with the chips. If faults are injected into the chip successfully, they may change the security-critical states or change the read values of the security flags/configurations stored within the NVM array. Both these faults may severely impact the system security relying on the NVM secure macro.

In the second attack scenario, the attacker may apply emission microscopy (EMMI) techniques to inspect the NVM array of secure macro, typically from the backside of the chip. For chips manufactured in advanced technology nodes, the InGaAs EMMI technique is particularly effective due to its high detection capability in the near-infrared spectrum. Using InGaAs EMMI, it becomes feasible to monitor the photoemission behavior during operation of the NVM macro.

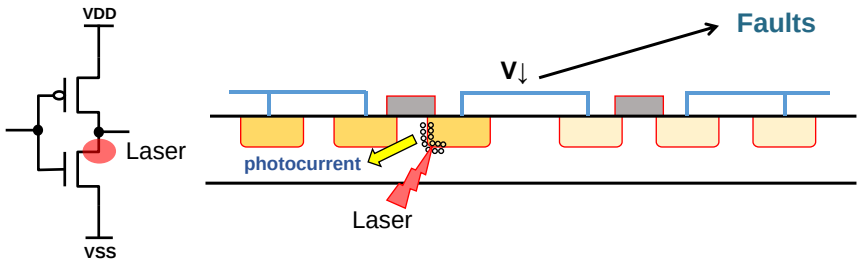

Figure 3-4 Backside laser fault injection.

For example, when a PN junction is forward-biased, the recombination of Electron-Hole pairs can emit photons. Finding the spots that have stronger photoemission may help identify the locations of forward-biased PN junctions. In certain operations such as reading data from the NVM array, it is possible that NVM cells or peripheral circuits behave differently depending on the stored data. If the differences in these circuit behaviors result in variances in different photoemission spectra, the stored data may be revealed by the InGaAs EMMI inspection.

3.3.3. Non-invasive Attacks

Three main types of non-invasive attacks are potentially threating to the NVM secure macro. These attacks can be executed without performing additional sample preparation processes, and they also do not require high-end techniques such as TEM, SEM, or FIB.

First, an attacker can perform side-channel attacks based on power analysis techniques, such as simple power analysis (SPA), differential power analysis (DPA), or correlation power analysis (CPA). By monitoring the power consumption of the SoC, the attacker can collect power traces that may contain side-channel information about certain circuit operations. In the case of an NVM secure macro, an attacker may collect power traces when data is being read from the NVM array.

As shown in *Figure 3-5*. By applying power analysis techniques to these power traces, the attackers aim to find the value of the read data. Similarly, an attacker may also apply the same technique when the NVM macro is performing write operations. In some cases, the NVM secure macro is embedded with a cryptographic engine, the attacker may also apply side-channel attacks while the macro is performing data encryption/decryption, aiming to find the secret keys used in these operations.

Second, since the NVM secure macro is usually embedded in an SoC with processors, its content could be vulnerable to unauthorized parties who can run SW/FW in the SoC. In such

cases, an attacker, who is an unprivileged user in this system, is trying to the secret parameters owned by a privileged user that are kept within the NVM secure macro. In most systems, since both an unprivileged user and a privileged user are sharing the same hardware components, there is no physical isolation from the internal attackers. As a result, the content stored within the NVM secure macro is mostly protected by security policies. If an internal attacker can break or bypass the security policies, it is possible for them to gain access to secret parameters stored within the NVM secure macro.

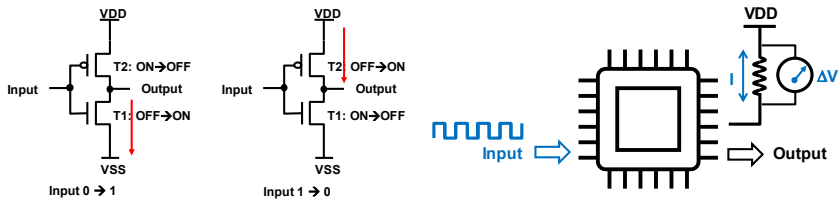

Figure 3-5 Input-dependent power consumption reveals chip information.

Third, the NVM secure macro may also be vulnerable to faults that are injected through external pins or other channels that do not require physical penetration. For example, an attacker may inject glitches into the power pin, temporarily changing the voltage level of the macro's power supply. Such changes can make the circuit operate outside its intended operating range, and some of them may result in transient faults that alter the circuit's behavior. If these faults are successfully injected into the NVM secure macro, they might result in changes in the data address or operating

modes, which may be exploited by the attackers to obtain the secret parameters stored within the NVM secure macro.

3.4. Requirements for Anti-tampering Design

To prevent the aforementioned physical attacks, countermeasures for each possible attack should be implemented. Generally, one should first identify the assets in the NVM secure macro. It is also necessary to recognize the critical security properties that need to be protected. For example, if we need to store a public key pair within the NVM, both confidentiality and integrity are critical for the private key, while only integrity is critical for the public key.

After identifying the assets that require protection, the next step is to assess the potential vulnerabilities of these critical assets against viable attacks. With an understanding of the attack methodologies and the potential victims, one can design and implement the necessary anti-tamper techniques into the NVM secure macro to make it tamper-proof.

4

Demonstrated as an example of a hardware root of trust, PUFrt is available on the market. We will dissect its design concepts and comprehensively analyze the 17 different anti-tampering design aspects within it. Building upon the threat models introduced in Chapter 3, this chapter will guide you on how to address each threat model with specific anti-attack designs and illustrate the chip's anti-tampering block diagram.

4. Hardware Root of Trust Designs and Anti-Tampering Methods

The previous two chapters provide a general overview of the kinds of threats we are encountering these days and the basic requirements for safeguarding the secure macro and the main SoC from these threats. Starting from this chapter, we will provide an in-depth explanation of the comprehensive anti-tamper solution of the PUFrt, an HRoT solution from PUFsecurity corporation.

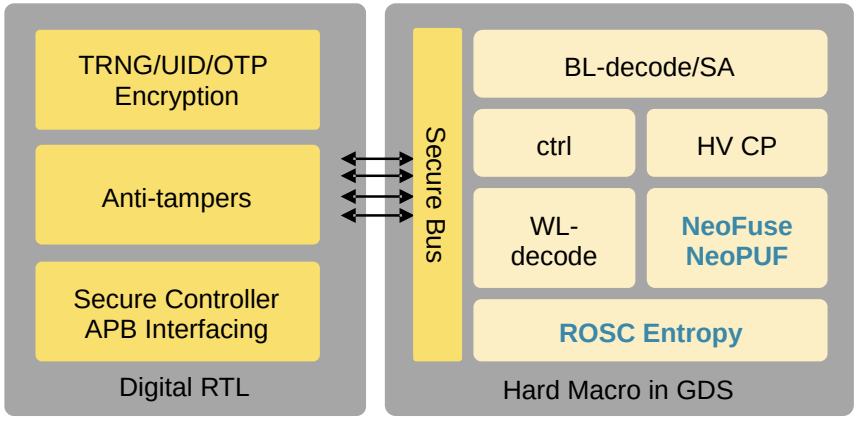

Figure 4-1 PUFrt functional blocks in RTL and hard macro.

As illustrated in *Figure 4-1*, the PUFrt can be partitioned into the hard macro part and the digital RTL part. The functionalities provided by each part are also shown in this figure. In addition to the OTP and PUF [38], PUFrt includes a true random number

generator (TRNG) that contains dynamic entropy using a hardened ring oscillator (ROSC) noise source [39] and uses the inherent PUF static entropy source as a conditioning algorithm. The digital RTL part consists of the controller for the hard macro and anti-tamper features.

4.1. Design Concept

The hardware root of trust solution, PUFrt, is a mixed-signal design with analog and digital parts. In order to make it tamper-proof, we need to implement different anti-tampering designs for different blocks and threat models. For example, if the protection of digital circuits focuses on fault injection, we can use Cyclic Redundancy Check (CRC) algorithms [40] to protect the integrity of the registers. If the protection of analog circuits or memory focuses on physical attacks, we need to implement some layout and device implementation protection for hard macro. Furthermore, if the overall macro encounters side-channel attack, such as DPA, we need to implement the relevant anti-SCA attack or protection actions.

In *Figure 4-2*, it shows the diagram of the PUFrt anti-tampering design. Compared with the traditional design, it is implemented with different protection methods according to different functional blocks. The hard macro part consists of a bandgap reference (BG), a high voltage generator (HV) with a charge pump (CP), WL/BL decoders, and an entropy source based on ring oscillators (ROSC). The hard macro is also equipped with countermeasures based on

layout and analog design techniques against physical inspection, side-channel attacks, and fault attacks.

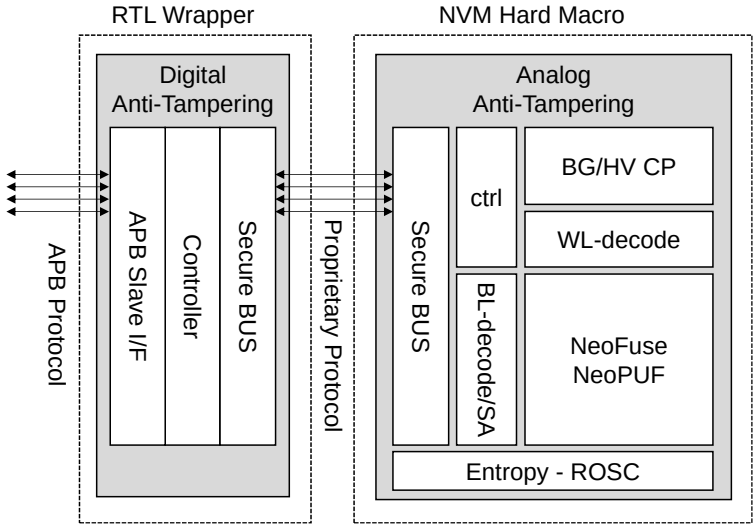

Figure 4-2 PUFrt consists of hard macro and digital functions.

The RTL wrapper consists of the controller, interface to the hard macro, and the standard APB interface. The RTL wrapper uses countermeasures based on digital design techniques against register fault injection. In addition, the data bus connecting these two blocks also needs to be protected by obfuscated layout and integrity checks of critical signals. The overall design scheme is shown in *Figure 4-2*, which contains the planning of each anti-tampering design, while the real details of its design will be further described in the next chapter. Here is a high-level design concept for the PUFrt anti-tampering features. In *Figure 4-3*, the key idea

is to create a multi-layered ring of protection by using the static entropy from PUF, the dynamic entropy from PUF-based TRNG, and a tamper-proof shell to protect the sensitive data stored in the central OTP.

The static entropy of PUF can be used as the secret parameter to protect the confidentiality of the stored data. The dynamic entropy can secure the operation flow and obfuscate the data being transmitted through the bus interface of the hard macro. Combining the NeoFuse OTP storage that has inherent high security against physical inspection with both analog and digital anti-tampering designs, PUFrt is a trusted platform that provides high resistance against physical tampering.

Figure 4-3 PUFrt design concept to provide highly secure HRoT.

The complete PUFrt anti-tampering designs are shown in the following *Table 4-1*. The design techniques can be categorized by the main defense target into three types of physical attacks: invasive, semi-invasive, and non-invasive attacks.

Table 4-1 PUFrt anti-tampering features

Types		Security Features
Invasive Attack	1	Intrinsic Physical Security
	2	Voltage Contrast Attack Countermeasures
	3	Data Address X-Y Scrambler and IO Shuffler using PUF
Semi-Invasive Attack	4	(Optional) Top Metal Shielding
	5	Security-oriented IP Layout
	6	Active Sense-Amplifier READ Protection
	7	Hidden and Obfuscated Data Interface (inside macro)
	8	Output Data Fault Detection
Non-Invasive Attack	9	Pin Integrity Protection on Mode and Array Selection
	10	Word Lock; Non-Accessible Post-Masking (on OTP)
	11	Zeroization and Post-Masking (on PUF)
	12	Built-in Secure Repair and Test-mode Lock
	13	(Optional) Random Dummy Insertion READ
	14	PUF Health Check
	15	Fault Injection Prevention (Mode/Address/Post-masking)
	16	Unified Write Power to Prevent Electrical Analysis
	17	Power Detection -VDD/VDDIO Floating

Invasive attacks refer to the use of physical methods to attack the chip, such as SEM [14] or TEM [15]. Non-invasive attacks usually refer to contactless electrical attacks. A semi-invasive method is a reverse attack on the chip using both methods above. The shaded items are implemented in the hard macro and the other ones are implemented in the RTL wrapper. Each implemented anti-tampering technique is assigned a number, which is in accordance with the order of descriptions in section 4.5, and these numbers are also used in the illustrations in this chapter to indicate their positions and purposes.

4.2. PUFrt: Hardware Root of Trust Design Example

PUFrt is a PUF-based (Physically Unclonable Function) secure macro that provides primitive security features, including robust unique secret generation, secure storage for sensitive data, and an endless supply of true random bits. *Figure 4-4* is the architecture of PUFrt.

The PUFrt can be partitioned into two parts, the hard macro in a full-custom layout (GDS) format and the soft macro in RTL design. The hard macro comprises a one-time programmable (OTP) storage block using NeoFuse technology and a PUF block using NeoPUF technology. The soft macro part of the PUFrt provides the digital functions for controlling the hard macro, managing the security policy, and interconnecting to the main system. Both NeoFuse and NeoPUF are CMOS-compatible technologies

developed and qualified in different process platforms by eMemory, providing secure OTP storage and inborn PUF functionality respectively. In the rest of the book, unless otherwise specified, the OTP terms will represent the secure OTP using NeoFuse in the PUFrt, and the PUF terms will represent the inborn PUF (NeoPUF) in the PUFrt.

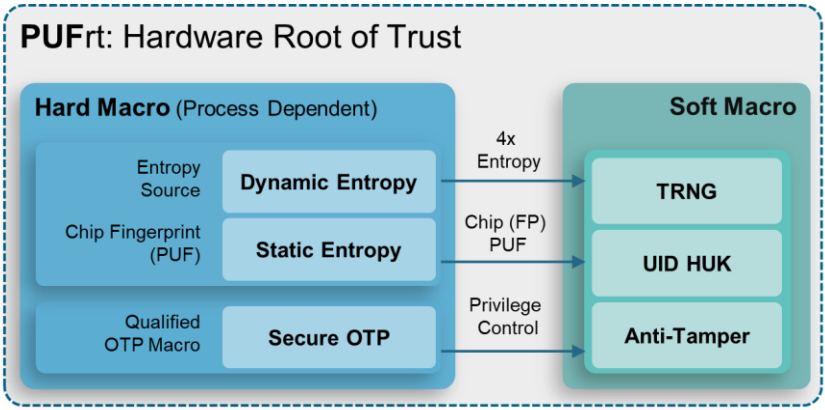

Figure 4-4 PUF hardware root of trust architecture.

PUFrt has five main features, namely, Secure OTP, Integrated TRNG, Inborn PUF, Dual interface, and Anti-Tamper designs. As shown in *Figure 4-5*, the Dual interface offers a secure controller with a standard APB interface, making PUFrt controllable by accessing the memory-mapped registers. The access policy of both OTP and PUF is fully adjustable by the users, which is done by setting the word lock, secure range, and post-masking features. Inborn PUF offers inborn unique keys. A PUF-based unique key can serve as a unique chip ID, a hardware unique key (HUK), or

an entropy source. Secure OTP is a secure storage that is protected by the built-in PUF-based data encryption technique. The data stored in Secure OTP is transformed into ciphertext through data shuffling and physical word address scrambling based on each chip's unique PUF secret. Integrated TRNG is a true random number generator that continually supplies random bits using a ring-oscillator-based noise source and a PUF-based conditioning engine. Finally, a comprehensive anti-tamper design is also an important feature that is implemented in the PUFrt.

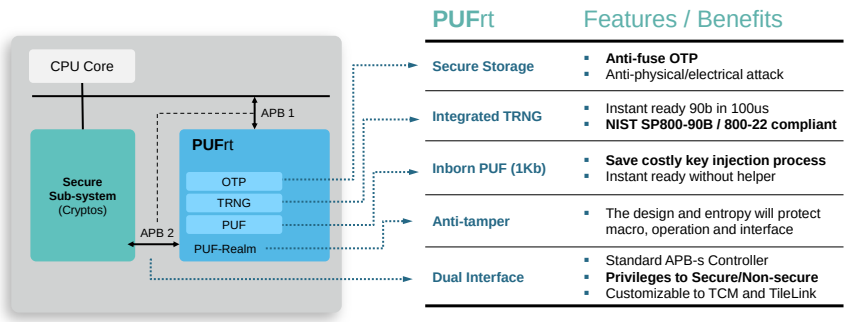

Figure 4-5 PUF hardware root of trust features.

In PUFrt, both the hard macro part and soft macro part are protected by the anti-tamper designs. The hard macro portion of the PUFrt has passed the third-party security assessments and is certified for set-top box (STB) applications. There are several examples that are currently in mass production and for sale in the market known for their high safety requirements. PUFrt is based on these fundamental and reliable tamper-proof designs of the hard macro, combined with the addition of digital anti-tamper designs to meet high security standards.

4.3. Features of PUFrt Anti-tampering Design

As discussed in sections 4.1 and 4.2, there are 17 anti-tampering features implemented in the PUFrt, which will be discussed in detail in the following paragraphs. Besides attributing these features by hard macro and RTL partitioning, each feature has the attribute of being effective when SoC power is On or Off.

1. **Intrinsic Physical Security** (*Hard macro*, *OFF-Power*)

PUFrt memory cells rely on the mechanism of quantum tunneling for operation. The energy for both the writing and enrollment operations is strictly controlled, so neither operation will cause oxide breakdown (oxide wear-out) due to over-programming, such as the case with other types of anti-fuse memories. From an SEM or TEM cross-sectional view, there is no physically traceable mark. That is, it is impossible to tell from the inspection which cells are programmed or unprogrammed.

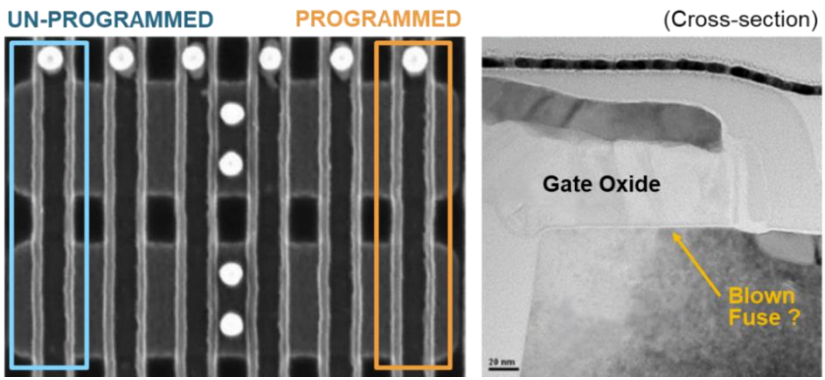

Figure 4-6 Quantum tunneling paths are invisible.

2. Against Voltage Contrast attack (*Hard macro*, *OFF-Power*)

When the system is powered off, the charge during PVC inspection will be evenly distributed to each leak point of oxide since the memory array has connected/shared poly and OD architectures. It is impossible to accurately discern any data from bright spot analysis. In short, PUFrt memory arrays are naturally PVC-resistant designs.

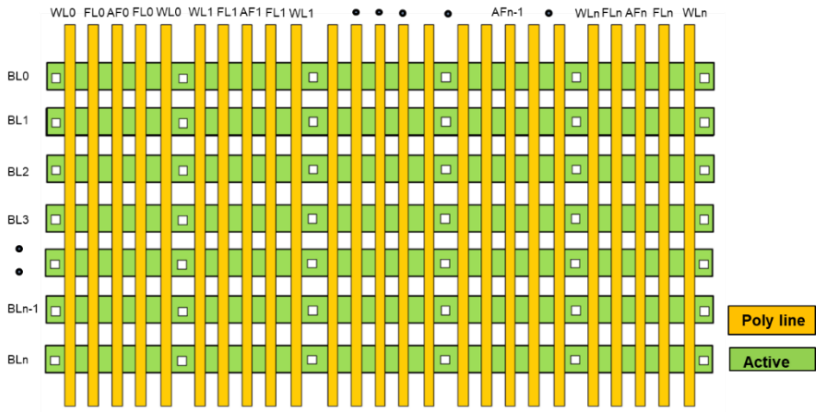

Figure 4-7 Memory cells design of against PVC attacks.

3. Data Address X-Y Scrambler and IO Shuffler (Digital, ON-Power)

PUFrt utilizes a XiP (execute-in-replace) secure OTP and read-only PUF extraction using a novel lightweight encryption (with ultra-low latency of ~1ns) algorithm for address scrambling and data masking. It uses 7 bits from the PUF for X-Y address mapping and 64 bits of random data for IO swapping. These 64 bits are derived by extending (through key expansion methods) 20 PUF bits to 64 bits. As such, all protected data and secure UID stored

on PUFrt have a relationship with the auxiliary (not user-accessible) on-chip 1Kb PUF secret (entropy).

4. **Metal Shielding** (*Hard macro*, ON-Power)

Optionally available, the addition of extra top metal layers of the PUFrt hard macros can be used for metal shielding, to prevent hackers from malicious probing or FIB attacks from the top level of the chip and/or to act as an EM shield.

5. **Security-oriented IP Layout** (*Hard macro*, ON-Power & OFF-Power)

Scrambling (obfuscation) is also physically implemented in the cell array on the word line, bit line, and I/O addresses. The component design and routing of PUFrt have been carefully designed. For example, the word line and bit line signals are divided into multi-layer routes, making an invasive attack (e.g., FIB) infeasible to manipulate the cell array. Moreover, following a security-oriented layout style, the lower metal layers are used for signal/control/address routing, with the higher metal layers reserved for power routing.

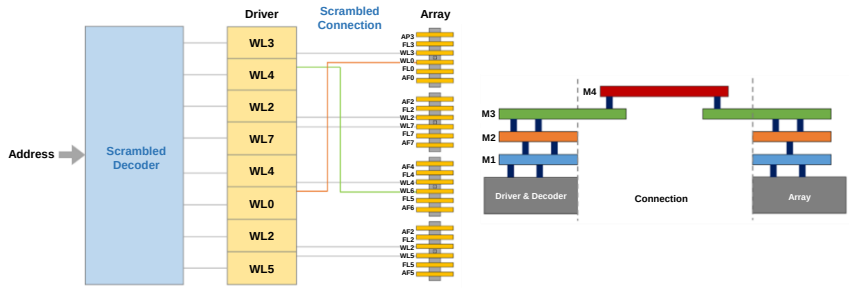

Figure 4-8 Security-oriented layout style.

6. Active Sense Amplifier Protection (*Hard macro*, ON-Power)

Attackers can use knowledge of the current differences when reading “0” or “1” data, to steal sensitive information from a monitored sense amplifier. However, for the PUFrt, the sensing circuit has very similar current spectrums when reading "0" and reading "1", so it will be more difficult to use attack methods such as EM or photon inspection to extract data from a sense amplifier.

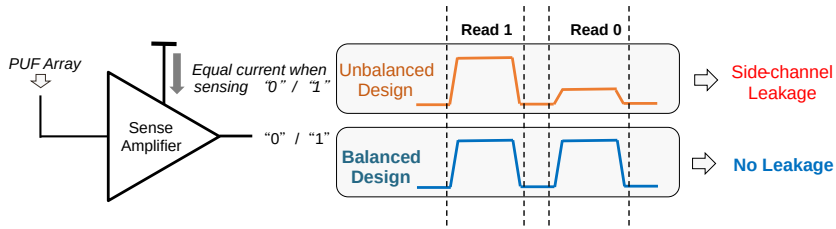

Figure 4-9 Balancing the read current protection.

7. Hidden and Mixed-Up Data Interface (*Hard macro*, ON-Power & *OFF-Power*)

The important signal routes and IO interface of the Hard Macro are realized using the middle metal layers. In addition, by request, the following protection schemes are available for the pin out/interface of hard macro:

- The pins can be arranged in a predetermined and out-of-order sequence.
- The minimum metal width and spacing can be used.
- Fake pins can be inserted with behavior like real outputs, created from the logical combination of real pins.

- The I/O pins can be arranged using dual layers of metal. For example, by using M2/M1, M1 can carry the actual signal, with M2 metal acting as an overlay to inhibit FIB attacks on the M1 lines. This can help reduce the possibility of the success of a FIB attack.

8. Output Data Fault Detection (Digital, ON-Power)

The SA out and Data out signals are compared with each other (same signal, two different tap points) to protect the data output path from being maliciously forced to 0 or 1. When such an attempt to force the output pins is detected, an alarm is raised, toggling the PINTRPT pin high.

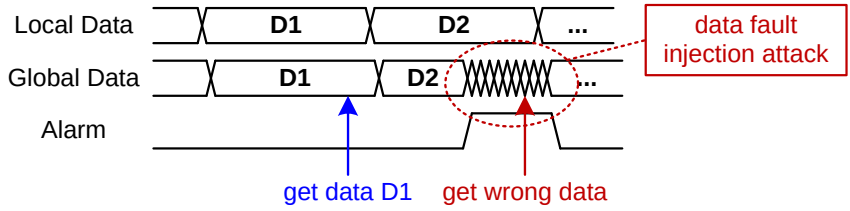

Figure 4-10 Fault injection detection design.

9. Pin Protection on Mode (Integrity) and Data (IO Shuffler) (Digital, ON-Power)

The hard macro's IO pins are more easily accessible, making them particularly vulnerable to attack, so extra anti-tampering features have been added to protect those pins:

- Address pins have parity checks.
- Additional, redundant pins for mode protection.

- All pins are shuffled using PUF bits. The first-stage circuit for shuffling the IO sequence is implemented on the hard macro.

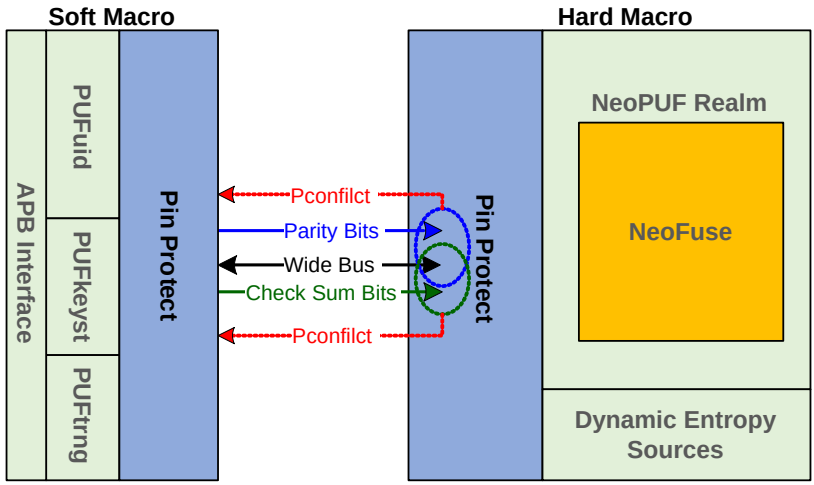

Figure 4-11 Pin protection at hard macro boundary.

10. Word Lock; Non-Accessible Post-Masking (on OTP) (Digital, ON-Power)

OTP locations can be individually set using the Word Lock function with three different access policies: RW, RO, and NA (read 0 forever). Post-masking is a function to temporarily set the access policy of the OTP to NA. This allows the user to read the OTP once at boot up, then apply the post-masking register so that OTP can no longer be read out. This prevents replay and power analysis attacks that rely on unrestricted and continuous access to the OTP. Post-masking is a temporary function – the access policy is reset to its original setting upon power reset.

11. **Zeroization and Post-Masking (on PUF)** (Digital, ON-Power & *OFF-Power*)

PUFrt performs enrollment to populate the PUF with random numbers so that it can act as a source for the static entropy pool and PUFuid. Similar to post-masking for the OTP, post-masking for the PUF forces it to behave as a one-time read memory. For example, after power-on, a user could use post-masking to prevent any further reading of the unique ID. Replay attacks, continuous power analysis, or photonic attacks are avoided because no further power signatures can be obtained after post-masking has been enabled.

As mentioned in point #10, post-masking is a temporary function that is reset after the power turns off, and then on again. For more permanent protection of the unique ID, zeroization of PUFuid is supported, which will permanently delete the UID.

12. **Built-in Secure Repair and Test-mode Lock** (Digital, ON-Power)

After the secure repair has been executed, any empty and unused repair bits should be programmed to “0” to avoid security vulnerabilities. Test-mode lock disables all test modes (which includes secure repair) so that only the user mode functions are available to use.

13. **Random Dummy Read** (Digital, ON-Power)

When a conventional memory is read, the read instruction will always be executed at a fixed location (the read address.)

Therefore, attackers may repeatedly apply a read command to the same location to analyze the power consumption and energy signatures for that address/data pair. EMMI optical detection or Optical Beam Induced Resistance Change (OBIRCH) laser facility can also be used to investigate signals and locate a specific address for PFA to reverse the data. In addition, fault injections or glitch attacks on the IO pins or sensing circuits have recently become more and more common. All such attacks need to be prevented to avoid incorrect data being fetched and/or secret data being stolen from the secure storage. Hence, random dummy read is required.

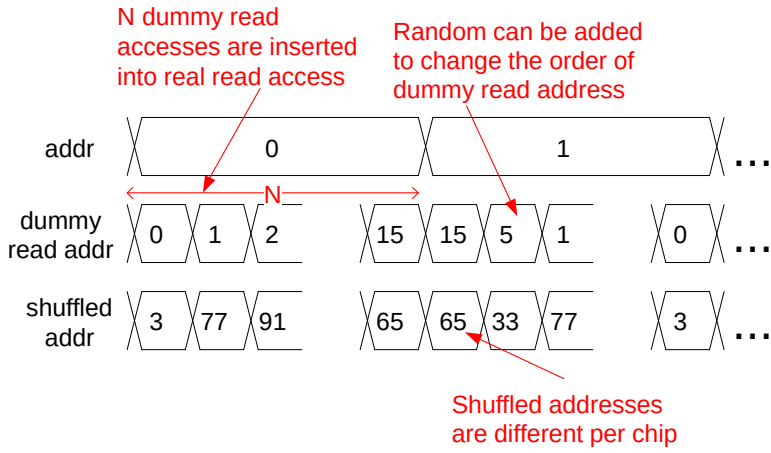

Figure 4-12 Random dummy insertion read.

Figure 4-12 illustrates the random dummy read mechanism. Random dummy read creates an unpredictable series of internal reads (using PUFtrng as the entropy source), padding the desired read operation with an additional number of dummy reads (using

either real, random locations, or fabricated garbage data). So instead of one read command from the user translating to one internal read operation executed by PUFrt, one user read command becomes multiple internal read operations, executed in a random order, with only one of the read operations using the user-given read address. This approach has several advantages:

- Prevention of replay attack by power analysis, laser, or optical inspection: since the physical address access sequence is now randomized, the exact physical location on the die would be difficult to locate, as it would seem to change for each read command, even for repeated reads to the same address. If post-masking is added as well, then each post-masked location can only be read once, making it even more difficult for an attacker to quickly locate and perform power consumption analysis or optical/laser investigation.

- Prevention of glitch and fault injection attacks: for each read, up to an additional 15 random locations will be read (in random order). This series of 16 reads will be performed twice but in a different read order the second time (also in a randomized order). For example, if the desired read address is “3”, then the first read sequence could be 1/2/3/…/14/15/16 while the second read sequence is 16/15/14/…/3/2/1 so that the reading of location “3” occurs at a different time for each of the read sequences. The results from both read operations of location “3” will then be compared with each other to prevent malicious IO attacks, making it difficult for such an attack to change the read data. A hacker would need to attack

precisely at the exact two times that the desired location (3 in this example) is being read.

Random dummy read is an optional tamper-proof design feature because it greatly increases the read access times of each location.

14. PUF Health Check (Digital, ON-Power)

NeoPUF needs to be enrolled before it can be used. PUFrt will not have a reliable static entropy source if the enrollment process was maliciously skipped during the manufacturing process, or if enrollment was performed, but the post-enrollment PUF contains insufficient entropy.

PUFrt has two indicators to alert the user if its built-in PUF does not contain sufficient entropy; that is if the PUF contains a true silicon fingerprint or not. The first indicator is the PUF Health Check function that checks the randomness of the entire 2Kb of NeoPUF by using its built-in statistical test. The second indicator is the flag that is automatically written after enrollment has been executed. Using the PUF Health Check in conjunction with the examination of the enrollment flag, users can determine if the PUF contains enough randomness to be used, otherwise, the interrupt pin will rise and indicate an error.

15. Fault Injection Detection/Protection (Digital, ON-Power)

To prevent attackers from injecting faults to bypass the security features of PUFrt, cyclic redundancy check (CRC) codes have

been implemented to detect the tampering of those registers responsible for storing security configuration settings and input addresses/commands. In addition, single-bit security flags/settings have been upgraded to multi-bit, to lower the possibility of a fault injection attack from disturbing secure operations. Finally, the sequencer/finite state machine has been hardened to protect the integrity of state transitions, raising an interrupt if a fault injection attack on the state machine has been detected.

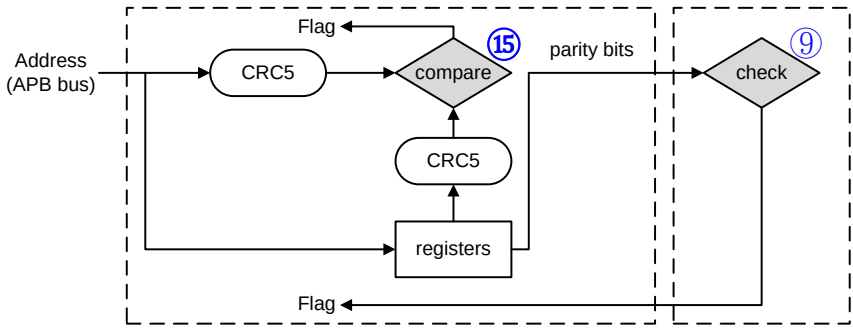

Figure 4-13 Fault injection protection by cyclic redundancy check.

16. Unified Read/Write Power to Prevent Electrical Analysis (*Hard macro*, ON-Power)

When reading the hard macro, analog techniques can be employed for unifying the read current for data 0 or 1, so that once the power signatures of reading 0 and 1 are the same, side-channel analysis of power/current usage becomes ineffective. When adversaries try to use read-by-write/enrollment and

distinguish whether the cells were already programmed/enrolled or not, it is not possible to use side-channel analysis to achieve it.

17. Power Detection -VDD/VDDIO Floating (*Hard macro*, ON-Power)

To prevent the malicious use of FIB to cut the power supply for the hard macro and make the hard macro outputs uncontrollable, floating voltage detectors have been placed in the hard macro. If a power supply is cut, PUFrt will respond by toggling PINTRPT, warning the SoC, and allowing it to decide on the next course of action. Both VDD and VDDIO power supplies are protected by floating voltage detectors.

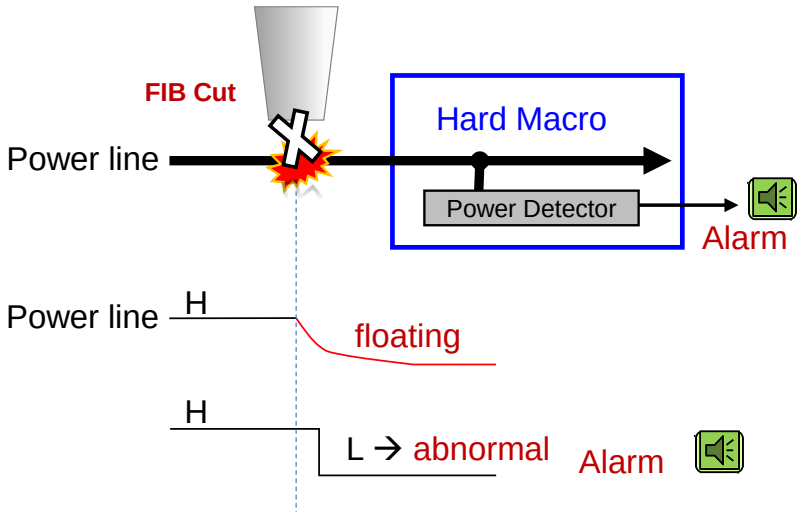

Figure 4-14 Power Detection by VDD/VDDIO.

4.4. Threat Models and Relevant Countermeasures

As more physical attacks arise, chips are facing numerous security threats in real practices. To safeguard the on-chip security system from being breached, the hardware root of trust has been more widely employed. If the most fundamental secrets were only being kept within the root of trust component, an attacker could only discover hints about the key or other secret parameters by tampering with it using the aforementioned physical attacks.

As discussed in section 4.1, a root of trust consists of nonvolatile memory and digital RTL design for controller or interfacing. There are many ways of attacking nonvolatile memory, such as SEM and TEM inspection. It is also possible to obtain information of the data written into or read from the NVM through power analysis. For the digital designs, since the control flow is based on the states kept in the registers, an attacker can corrupt the operation by injecting faults into the registers, and subsequently gaining access to the protected memory regime.

Table 4-2 lists the relevant threat models for the PUFrt and their corresponding countermeasures. The shaded items are implemented in the hard macro and the other ones are implemented in the RTL wrapper. The circled numbers (①②③……⑰) are in accordance with the anti-tampering features listed in *Table 4-1*.

Table 4-2 Relevant threat models of PUFrt

Threat Model	Implemented Countermeasures
SEM, FIB, TEM, optical inspection (OBIC/OBIRCH)	Intrinsic physical security ①
Passive voltage contrast	Sharing poly and OD array ②
Locate address, delayer, and nano probing optical inspection (OBIC/OBIRCH)	Top metal shielding, security-oriented IP layout, inter-metal routing ④⑤⑦
	Address/IO scrambler, post-masking, random dummy read ③⑩⑪⑬
Power analysis on SA during read	Active SA protection during reading ⑥
Fault injection on visible IO or mode select	Output Data Fault Detection, Pin Protection, IO-Shuffler, FI prevention ③⑧⑨⑮
Rollback, replay attack, and software access	Assess Permission and Post-Masking on OTP and PUF ⑩⑪
Secure setting or reserved bit leakage/revise	Secure repair and test-mode protection by lock ⑫
Fault injection or glitch protection	Output Data Fault Detection, Random dummy read, FI Prevention ⑧⑬⑮
Power analysis on CP or maliciously cut power	Unified operating power and power floating detection ⑯⑰
Skip enrolling PUF	PUF health check and flag check ⑭
Photon emission inspection	Active SA protection during reading ⑥
	Address/IO scrambler, post-masking, random dummy read ③⑩⑪⑬

4.5. Anti-Tampering Design in Hard Macro

The tamper-proof design in hard macro is to protect the related operations of OTP and PUF and to avoid reverse attacks that lead to data leakage or compromised security. *Figure 4-15* illustrates how the analog anti-tampering techniques are implemented in different functional blocks of the hard macro to make it tamper-proof.

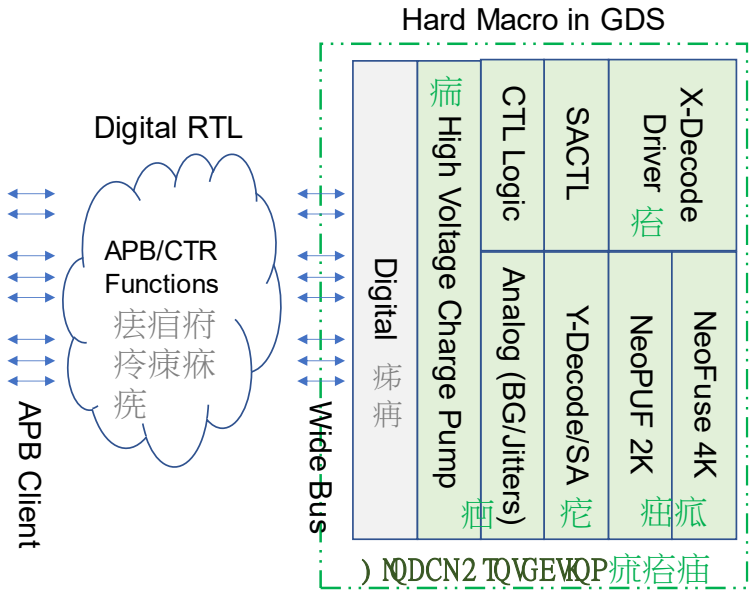

Figure 4-15 PUFrt anti-tampering features in hard macro.

The analog anti-tampering designs can help protect memory cells, arrays, analog peripheral circuits, and signal routing. As previously

listed in *Table 4-2*, attacks that are commonly applied to the hard macro are:

1. Optical inspection using electron microscopies, such as SEM or TEM.
2. Photon emission inspection, such as EMMI or InGaAs.
3. Power attacks, including power analysis or FIB cuts to float the power supply.
4. Malicious probing, such as PVC or fault injection.

The anti-tampering techniques highlighted in *Figure 4-15* are dedicated to countering the threats brought by each of these common attacks.

4.6. Anti-Tampering Design in RTL

In order to make the entire NVM secure macro tamper-proof, the digital RTL wrapper of the integrated macro requires secure operating flow, permission control, and channel protection. *Figure 4-16* shows the functional blocks and tamper-proof designs implemented in the RTL wrapper.

These designs help protect the bus interfaces and the exposed pins from malicious attacks, add dedicated access management to prevent illegal access of important data, hide important data by leveraging unique PUF secrets to scramble the address and data, and minimize the information leakage from photoemission that may be inspected by specific instruments.

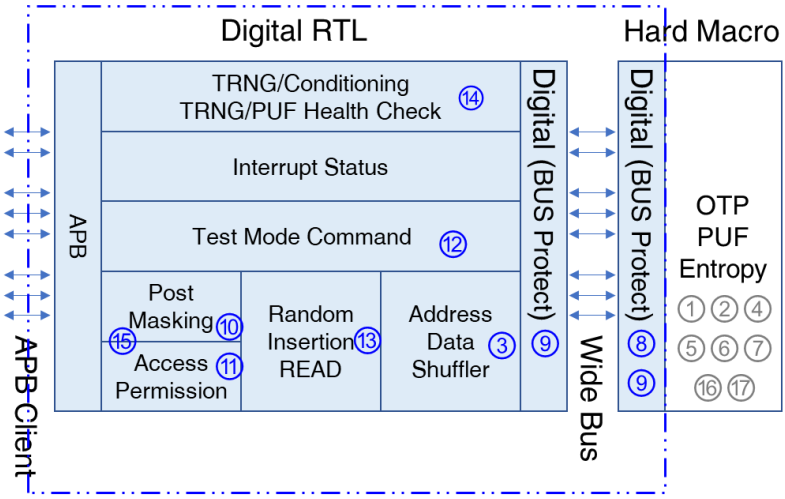

Figure 4-16 PUFrt anti-tampering features through digital design.

5

Precise measurement techniques may potentially observe the data stored in memory. Therefore, TEM and SEM, along with related penetration tests, are indispensable. Using a hardware root of trust available on the market, PUFrt, as an example, readers will gain insights in this chapter into the security penetration testing required for chips.

5. Resistance to Invasive Attacks: SEM & TEM Inspection

The PUFrt is composed of the hard macro of Secure OTP and PUF, and the soft macro of interfacing, controller, and security functions. The protection in the hard macro is mainly to prevent physical inspection of SEM and TEM and differential power analysis from side-channel attack. The following three sections of this chapter will carry out relevant experimental analysis on SEM, TEM, and physical imprinting observation from simulated end-of-life samples.

5.1. Resistance to Physical Inspection

The hard macro of PUFrt relies on the mechanism of quantum tunneling for cell operations. The energy for both the program and enrollment operations is controlled to form the quantum tunneling path in the gate oxide, shown in *Figure 5-1*.

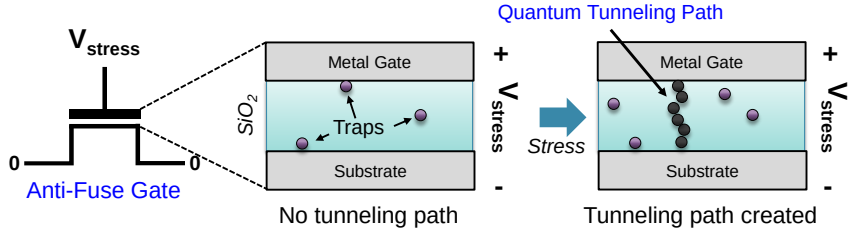

Figure 5-1 Mechanism of quantum tunneling for enrollment operations.

Scanning Electron Microscope (SEM) and Transmission Electron Microscopy (TEM) can be leveraged by hackers to inspect the micro-structures of semiconductor devices. As shown in *Figure 5-2*, the programmed and un-programmed Secure OTP cells cannot be distinguished by the SEM planar view. From the TEM cross-sectional view, there is also no physically traceable mark in the gate oxide of a programmed cell. It is believed the current conduction in the gate oxide is majorly based on electrons hopping through the traps which are in the form of defects or dangling bonds that are invisible to modern SEM/TEM inspection tools. Therefore, it is impossible to tell which cells have been programmed by SEM and TEM inspection and which have not.

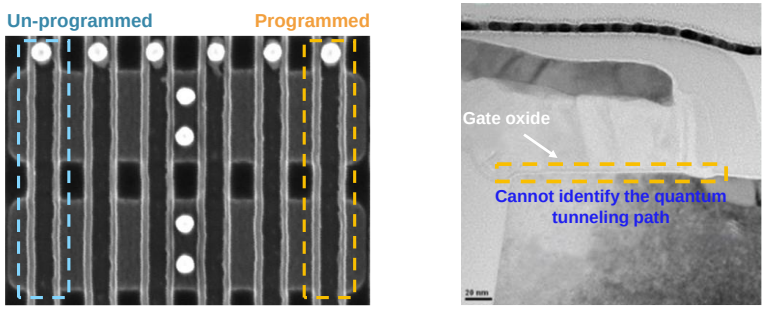

Figure 5-2 (a) SEM of PUF/Secure OTP. (b) TEM of PUF/Secure OTP.

5.2. Resistance to post-HTOL-induced Physical Imprint

In order to understand if imprinting is present in those samples with aging effects, SEM and TEM examinations of the NVM cells in the

PUFrt after long-term HTOL (High Temperature Operating Life [41]) stress were also evaluated. *Figure 5-3* shows how the imprinting phenomenon occurs and if there is secondary ion reflection by SEM if oxide leakage occurs at that location, or if any trace of data leakage can be observed by TEM inspection. The following penetration tests will demonstrate that no imprinting was found even when the memory cells were stressed under prolonged HTOL conditions.

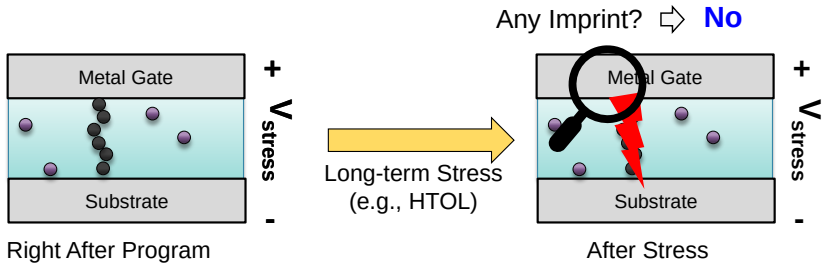

Figure 5-3 Post-programming(left); after long-term HTOL stress(right).

Anti-fuse OTP is an NVM technique that uses oxide leakage to store a 0 or 1. In general, VC (voltage contrast), SEM, or TEM can be used to observe whether there is a physical scratch from oxide rupture. However, the studies reported in these papers are from individual transistors with isolated POs/gates or high-energy oxide volcano ruptures. The latter is usually seen in severe oxide hard breakdown events that have dielectric breakdown induced epitaxy. In contrast, a soft breakdown [42] formed by oxide traps or an oxide breakdown between soft and hard breakdown [43] is generally not easy to observe.

In this experiment, PUFrt test chips will be used to observe whether physical traces such as imprints can be observed after post-HTOL by VC, SEM, and TEM experiments. It demonstrates that both the PUF and the secure OTP are resistant to invasive physical attacks.

5.3. SEM Inspection

The test chip used in this experiment is implemented in the TSMC 55nm Ultra Low Power Process. Before going under SEM inspection, the samples are first subjected to a 1000-hour high temperature operating life (HTOL) stress. The goal of the experiment is to observe the data pattern of the random PUF code, in which the “0”s and “1”s are randomly distributed with an equal probability of 0.5. Since “0”s are represented by PUF cells that have a quantum tunneling spot at the read path, the attack method is primarily locating these quantum tunneling spots.

The experiment is performed by a 3rd party examination laboratory, Ma-Tek Inc., incorporating their high-end sample preparation techniques, SEM equipment, and analysis tools. There are two types of inspection results from the experiment, one is based on the samples that are de-layered down to the contact layer, and the other one is based on the samples that are de-layered down to the active layer.

As shown in *Figure 5-4*, the PUF array is delayered to the contact layer, with the poly-silicon lines exposed under SEM inspection. As shown in the zoomed-in figures of three different PUF regions,

there is no visual difference among them. Since the three different regions are composed of different PUF cells, they provide very different data patterns. If the data pattern can be visually seen, there should be a clear difference between these three zoomed-in figures, which is clearly not true. Consequently, using the SEM technique to observe the PUF array delayered to the contact layer is ineffective in finding the PUF data.

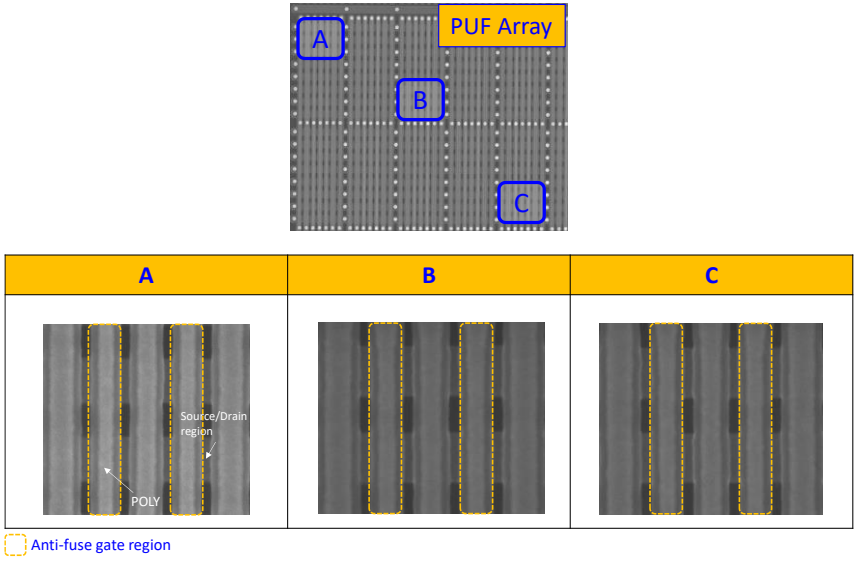

Figure 5-4 SEM inspection results on the PUF array delayered to contact.

In the other case as shown in *Figure 5-5*, where the PUF array is delayered down to the active region, the source and drain area of the transistors are exposed under SEM inspection. Similar to the previous results, the three different PUF regions are visually identical, which implies that using the SEM technique is unable to

find the PUF data in this type of sample. Summing up these findings, we can conclude that the PUF is resilient to SEM inspection.

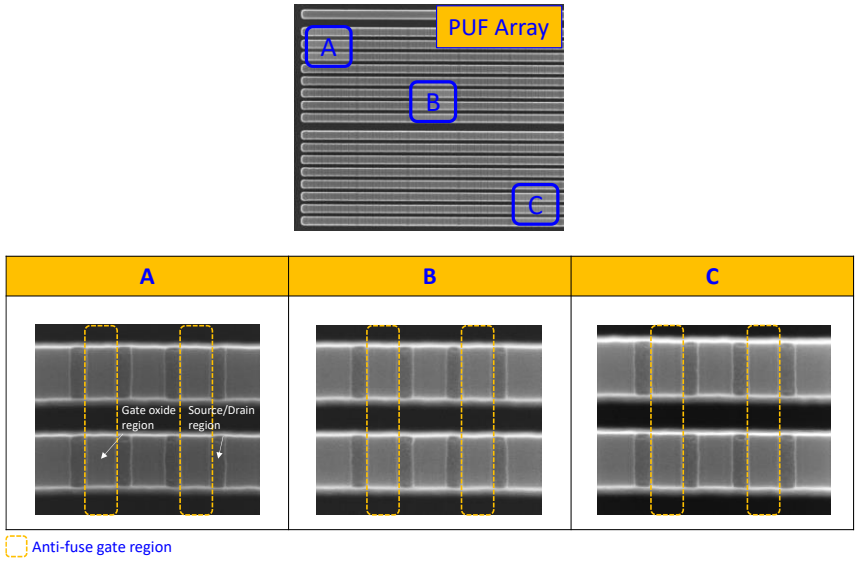

Figure 5-5 SEM inspection results on the PUF array delayered to active area.

5.4. TEM Inspection

The test chips used in TEM inspection experiments are implemented in the TSMC 22nm Ultra Low Leakage Process. Similarly, the samples are first subjected to a 1000-hour HTOL stress before the experiment. The goal of this experiment is to find the data stored in the Secure OTP cells through TEM inspection.

The experiment is also performed by Ma-Tek Inc., using their TEM tools and other supporting services. The inspection results are

shown in *Figure 5-6*. Since the OTP is programmed with the checkboard pattern, wherein each of the zoomed-in OTP regions, the four OTP cells are having a 1010 pattern as illustrated. If the TEM is able to identify the data pattern stored in the OTP cells, there should be a visual difference between the cells with “1”s and “0”s, which is not the case. Consequently, we can conclude that TEM inspection is ineffective in attacking the OTP array to obtain the stored data.

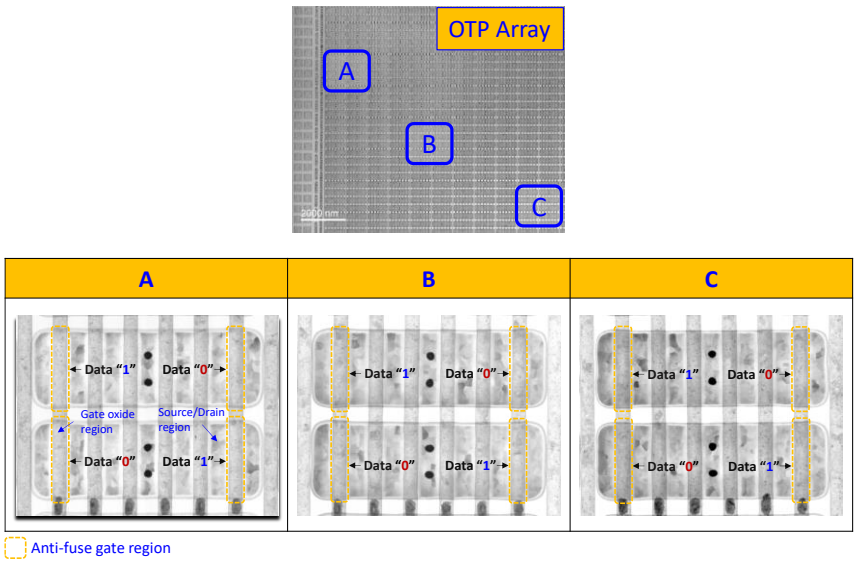

Figure 5-6 TEM inspection results on the OTP array.

5.5. Conclusion

Microscopic inspection techniques are powerful tools that have been used for data piracy of the NVM storages. Benefits from their physical properties, the Secure OTP and inborn PUF are known to

be highly resistant to microscopic techniques. The subsequent experiments on the test chips have demonstrated that inborn PUF is resilient to SEM and Secure OTP is resilient to TEM. Since the two share the same cell structure, the results of one can imply the other. In conclusion, the OTP and PUF in the PUFrt have been proven resilient to physical attacks based on SEM and TEM.

6

Without leaving any traces of tampering with the chip, attackers can quickly obtain cryptographic keys during operation through side-channel attacks. Additionally, attackers can compromise the availability of secure chips through EM waves or high-temperature operations, causing the loss of the chip's original security protections. In this chapter, we will introduce examples of hardware roots of trust that meet market demands for addressing these three types of attacks.

6. Side-Channel Attack: Timing and Power Side Attacks

A side-channel attack is a common method for adversaries to “peek” into the secret within the device without damaging it. In this section, power and timing side-channel attacks are used as examples to explain how the attackers may break the confidentiality of an OTP.

6.1. Timing Side-Channel Attacks

This attack is based on the concept of “read by write” and requires the adversaries to manipulate the IP interface or to take the cover of the SoC’s runtime control. The adversaries will intentionally proceed with the program/enroll operation in the IP macro and attack the IP macro maliciously.

When the users intend to execute the programming (OTP) or enrollment (PUF) process, the high voltage needs to be set up across the gate and finally form the quantum tunneling path. The conduction current (100~200uA level) will surge through the virgin cell transistors and finally be collected at the bit line. The gate oxide will become a low-resistive path for the current conduction.

Adversaries will attempt to re-execute the program process on the low resistive gate oxide of the target cell transistor. A similar surge

current will be repeated but at a ***faster*** pace as the oxide layer has a very high conductive path. The adversaries would compare two kinds of surge current waveforms/profiles to judge the secret to be data “1” or data “0” through “read by OTP write/ PUF enrollment process.” In this case, data “1” is defined as the virgin state of the cell.

Timing side-channel attack needs to re-program/enroll the cell array to extract the secret. By writing data “0”s into the array (applying high voltage to every cell), the programmed/enrolled cells will exhibit the program/enroll surge current in a faster path than the non-programmed/enrolled cells. Data “0”s and “1”s can be determined hypothetically by observing the surge current timestamp difference observed from programming/enrolling distinct cells. Therefore, data contents can be leaked by means of the “read by OTP write/ PUF enrollment process.”

Experiments were performed to test if the “read by OTP write/PUF enrollment” is capable of extracting data from the cell array or not. The program/enrollment current was monitored from the VDD2 terminal and program/enroll operations were executed on programmed/enrolled cells. The resulting waveforms are demonstrated in *Figure 6-1* and *Figure 6-2*.

The pink curve indicates the transient voltage profile on the terminal of the resistor added to the VDD2 pad during the programming process on the original virgin cells. The voltage profile can be used to extract the conduction current profiles.

Please note that data “1” represents the Virgin cell while data ”0” represents the Program cell.

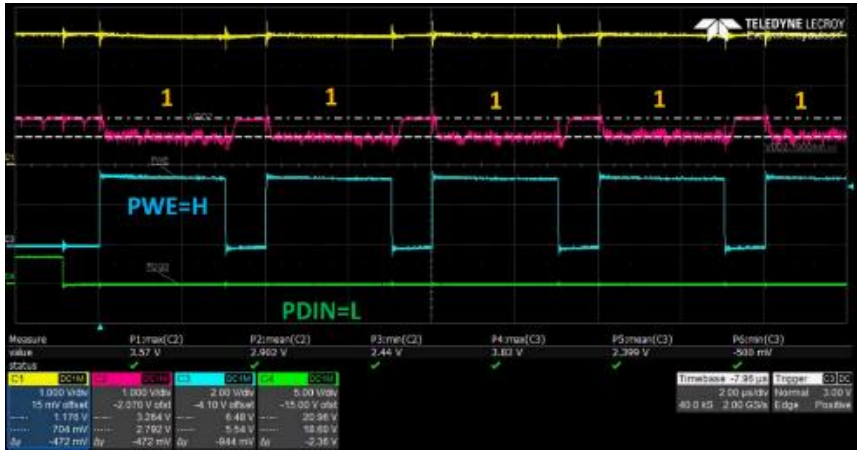

Figure 6-1 Timing side-channel attacks I.

The result shows no significant difference in surge current timestamp between the first and the second programming operations. The monitored VDD2 current signatures contain the cell current and the required operation current for charge pumping/high voltage control circuits. Even if the cell current has the difference between data “1” and data “0”, the differences will be hidden behind the large current consumption variations with charge pumping to support the write and enrollment process.

The pink curve indicated the transient voltage profile on the terminal of the resistor added to the VDD2 pad during the re-programming process on those cells programmed with the check

board (CKBD) pattern: "1010...10". After the "programming all 0" process, the data will become all “0” s.

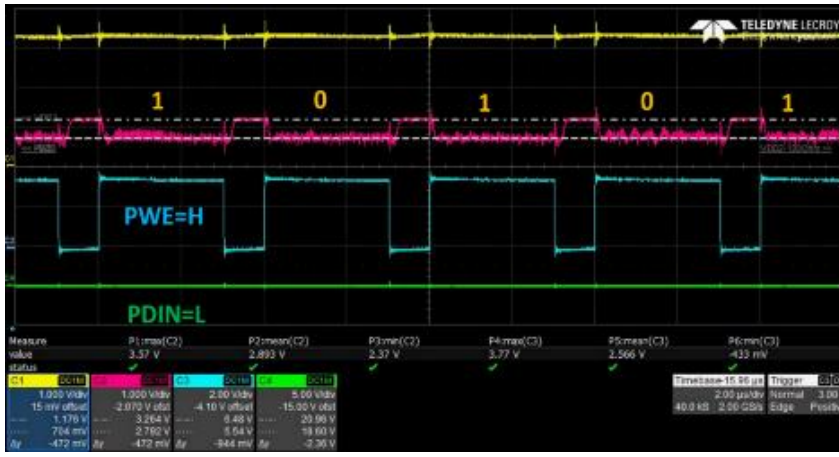

Figure 6-2 Timing side-channel attacks II.

6.2. Power Side-Channel Attacks

Power side-channel attack also utilizes the information leaked from the power ports to obtain the data that is stored. This attack assumes that the SoC chip has exposed the OTP/PUF voltage supply to the external environment. However, this method would not be successful due to various factors that add confusion to cell state judgment. Based on the intrinsic physical characteristics of the program/enrollment process, the following factors will strongly affect the currents:

- The fluctuations in the voltage used for program and enrollment.

- The variations of time form the quantum tunneling path over different cells.

All the above factors could lead to a time difference in the current surge and result in a spread in timing and power values observed for different cells. Furthermore, the current consumption from the internal charge pump overshadows the current level difference from a distinct cell state. Therefore, a power side-channel attack executed during the write/enrollment process would not be successful. In conclusion, The Secure OTP and PUF exhibit intrinsic resistance against timing and power side-channel attacks.

6.3. Confidentiality and Integrity under Extreme Operation

Adversaries may try to tamper with the cell data information by applying different environmental conditions to destroy the data integrity. However, secret data stored in Secure OTP and PUF blocks is not susceptible to environmental conditions such as temperature and radiation. The following experiments demonstrate that data integrity can be kept in Secure OTP and inborn PUF under harsh environmental conditions.

6.3.1. Data Retentivity in High Temperature

Three distinct temperatures (175°C /250°C /275°C) were selected to test the data retention for Secure OTP test chips using high temperature baking stress. The *Figure 6-3* testing result showed that even under 275°C baking, after 1000 hours, the data stored

inside the selected cell remains as good as its initial condition at 0 hours. Thus, data retentivity is considered stable even under high temperature stress. Furthermore, this experiment proves that E_A (Activation energy) is irrelative to temperature. The secret can be kept within PUFrt for a long duration.

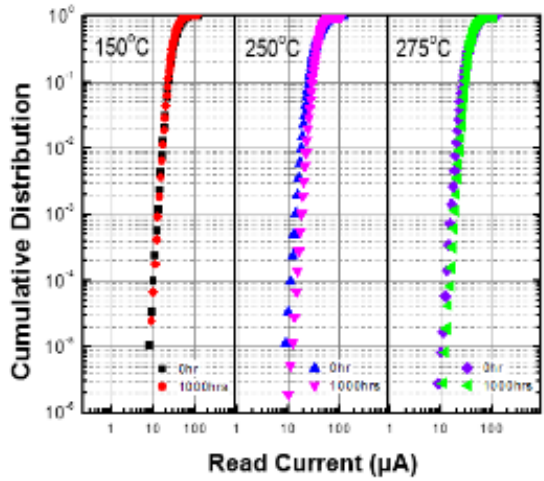

Figure 6-3 Secure OTP data retention test result.

6.3.2. Electromagnetics Radiations (Semi-Invasive Attack)

To study the resistance of Secure OTP against electromagnetic (EM) radiation, experiments were performed where samples were exposed to 100KGy high dose Gamma ray. The samples contained a checkboard pattern (50% cells with data “1” and 50% cells with data “0”) as the stored secret in the Secure OTP array, any change in data due to radiation would be easy to conceive.

Figure 6-4 demonstrates the shmoo plot for the read functionalities. The result shows all read accesses are still functional well after 100Kgy Gamma ray radiation and no data are lost after such EM irradiation.

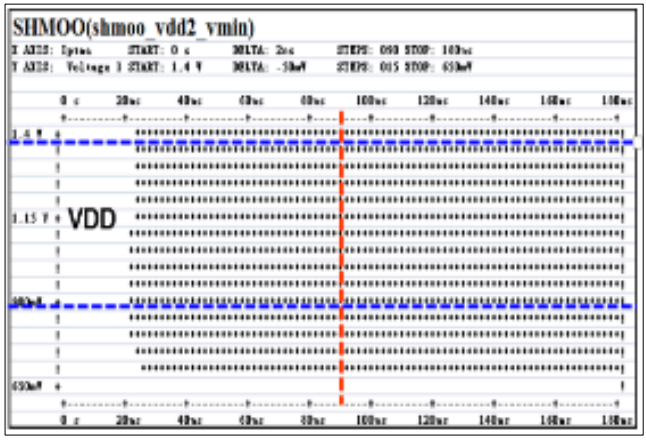

Figure 6-4 Radiation test of Secure OTP shown in shmoo plot.

In conclusion, since the secure OTP and inborn PUF in PUFrt are implemented based on an irreversible program/enrollment process, environmental variations, non-invasive attacks such as Gamma-ray irradiation, and even UV light exposure would not affect the data integrity.

7

In this chapter, readers will gain valuable insights into the experimentation related to the hardware sample, specifically its decapsulation for backside imaging. The chapter encompasses a comprehensive experiment setup, results, and analysis, focusing on various operation modes applied to an example of Hardware Root of Trust (PUFrt) circuit. It covers the implications of regular reads, repeated resets, and DC reads, ultimately providing a summary of the experiment's findings.

7. PUFrt InGaAs Backside Imaging Analysis

Photon Emission Microscopy is widely used for performing failure analysis of integrated circuits. Using an emission microscope with photodetectors made from InGaAs material (InGaAs EMMI, or InGaAs), the leakage current locations on a chip can be pinpointed by detecting the Near Infrared (NIR) emission [44] caused by electron-hole recombination. Since IR has a higher transmission rate over the silicon substrate than visible light, it can be detected from the backside of a chip without the need for invasive sample preparation, such as removing the metal layers. Given these reasons, the InGaAs is well-suited for non-invasive attacks on secret data generated by PUFs or stored in memory elements.

The purpose of this experiment is to determine the capability of InGaAs backside imaging technique for attacking devices within PUFrt and to verify the effectiveness of the anti-tampering techniques embedded in PUFrt. While reading data from the memory, a PUF cell that generates logic-0 will conduct current through its oxide tunneling path, which is expected to emit photons that will transmit through the substrate and be detected by the InGaAs detector. If an attacker can locate those PUF cells that output logic-0, they can then derive the correct values of the PUF contents through InGaAs backside imaging.

To guard against this, the anti-tampering techniques in PUFrt are used to protect against the logic-0 cells from being located. To give a summary of the results, this experiment shows that the logic-0 cells can be indeed located if an attacker can repeatedly send read commands to PUFrt, but the actual PUF contents are not derivable from these locations because of the address/data shuffler, which decouples the physical locations of those logic-0 cells from the real PUF values. Moreover, this experiment also shows that locating the logic-0 cells becomes impossible once the random dummy read technique has been enabled.

7.1. Experiment Setup

The hardware sample for this experiment is the same as used for regular PUFrt evaluations, except that the chip is decapsulated for backside imaging. The sample is placed inside a Hamamatsu PHEMOS-1000 emission microscope for backside InGaAs image inspection.

In this experiment, several operation modes are applied on the PUFrt circuit to represent different threat models, as listed below:

1. **Regular Read:** assuming that an attacker is using the InGaAs imaging technique to monitor the read operations performed by an authorized user.
2. **Repeating Reset:** assuming the attacker can repeatedly reset the entire PUFrt to observe the read operations during the initialization-after-reset procedure.

3. **DC Read:** assuming the attacker can force the read to enable the pin of the Hard Macro to stay at high, i.e., the read mode is always enabled.

7.2. Experiment Results

The results of different sub-experiments will be demonstrated in this section, together with more detailed descriptions of each threat model and experiment condition.

7.2.1. Regular Read

In this experiment, the threat comes from an attacker who cannot send commands directly to the PUFrt circuit. As an alternative, this attacker will wait for an authorized user to send read commands to PUF and try to capture the corresponding chip activities using the InGaAs imaging technique. The goal of this attack is to discover the secrets generated by the PUF array by analyzing the InGaAs image.

Normally, an authorized user will read a 128-bit or 256-bit PUF code for key derivation or other purposes. Once this operation is completed, there is no need to read the same PUF code again because a PUF code only serves one purpose, and the outcome is stored within an internal register, which will only be erased during power-off or reset.

If the PUF code is only read once, the resulting InGaAs image will capture the chip activities when the PUF cells are idle, or at best

capture one photon emission event from the PUF cells during the read operation, which cannot provide sufficient energy to the InGaAs detector. Consequently, the InGaAs image taken under this condition, as shown in *Figure 7-1*, shows no hotspots in the PUF cell region.

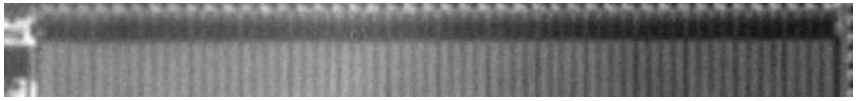

Figure 7-1 InGaAs image when the PUF code is read only once.

Given that there are no detectable hotspots on the resulting InGaAs image, one may raise a reasonable doubt on whether the experimental setup is correct or not. For clarification, the image area is zoomed out to cover the entire Hard Macro, as shown in *Figure 7-2*.

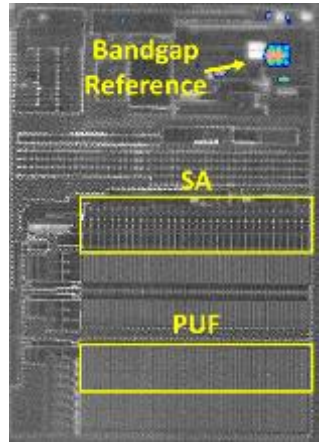

Figure 7-2 InGaAs image superimposed with the die-photo of the hard macro.

While the highlighted PUF region shows no hotspots, there are clear hotspots found in the region of the bandgap reference circuits. This figure confirms that this setup can indeed detect photon emissions from the backside of the test chip, but there are no detectable signals from the PUF cells during the regular read operations. In *Figure 7-2*, there are also no detectable hotspots at the location of sense amplifiers (SAs), showing that the SAs are also not valid targets for InGaAs imaging analysis.

7.2.2. Attacking the 1Kb PUF Secret

The previous experiment shows that PUFrt is intrinsically immune to InGaAs imaging attacks. In addition to this intrinsic anti-tampering feature, the address/data shuffling algorithm provides PUFrt with another layer of protection. The shuffling algorithm randomized the address and data sequences based on the 1Kb PUF secret, which is unique for every chip. In summary, the data read from PUF or OTP is protected by the 1Kb PUF secret.

If an attacker wants to break the shuffling algorithm, one possible approach is to attack the 1Kb PUF secret using InGaAs. In the regular operation mode of PUFrt, the 1Kb PUF secret will be read only once during the initialization step after power-up or system reset and will become non-accessible afterward. Consequently, it is impossible for an attacker to repeatedly read this 1Kb secret and reveal its data on an InGaAs image. The only remaining option for an attacker is to repeatedly reset the PUFrt to force the read operation of this 1Kb secret after each reset.

In an experiment to test the feasibility of attacking the PUF secret, the PUFrt sample is continually reset at the maximum speed of one reset per 25μs, which is limited by the required initialization steps (including reading the 1Kb PUF secret). The InGaAs image of this experiment is shown in *Figure 7-3*, which shows no hotspots at the PUF cell location in the Hard Macro. As a result, it can be concluded that any attempts to reveal the PUF secret by repeatedly resetting PUFrt cannot succeed, and the shuffling algorithm, therefore, cannot be broken by attacking the PUF secret.

Figure 7-3 InGaAs image of repeatedly resetting the PUFrt on 1Kbits PUF cells.

7.2.3. DC Read

The read operations at Hard Macro are enabled by a signal called PCLK, which enables currents to flow from the PUF cells to the sense amplifiers (SAs) when PCLK is set at high. It is straightforward to suspect that currents will be continuously drawn if PCLK is set to be always high, leading to clear hotspots in the InGaAs image. This concern is solved by one of the analog anti-tampering features, which will cut off the current path from PUF cells to SAs after the outputs of SAs have reached their stable states. As a result, even if the PCLK signal can be forced to be

always high, the current will be only drawn in a short period after the rising edge of PCLK, which is equivalent to a duty ratio of almost zero.

In the subsequent experiment, the PUFrt sample is set to read mode with a fixed read address after a reset. Meanwhile, the PCLK signal is set from low to high once and is kept at high until the reset signal is sent. The InGaAs image of this experiment is shown in *Figure 7-4*, which shows no hotspots as expected. This experiment, therefore, confirms the effectiveness of this analog anti-tampering current cut-off feature.

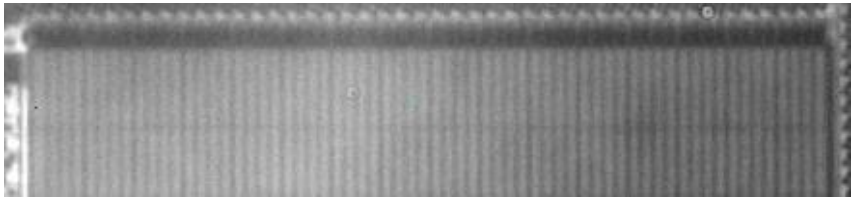

Figure 7-4 InGaAs image when performing DC read on 1Kbits PUF cells.

7.3. Experiment Summary

The results shown in this section show that no secret data can be obtained through InGaAs imaging analysis, even if the anti-tampering features are disabled. These results are summarized in *Table 7-1*.

Table 7-1 Experiment summary of InGaAs imaging analysis

Test mode	Hotspots detected?
Regular Read	No
Repeating Reset	No
DC Read	No

Based on the experiments, the vulnerability against InGaAs imaging analysis of individual blocks is also summarized in *Table 7-2*. All individual blocks of PUFrt are immune to InGaAs attack by nature and are all additionally protected by digital anti-tampering techniques, implying that it is impossible to derive secrets using InGaAs imaging technique.

Table 7-2 Summary of InGaAs attack vulnerability for individual blocks in PUFrt

Block		Scrambling	Dummy Read	InGaAs hotspots	Derive secret data?
PUFrt	1Kb Secret PUF	No	No	No	No
	PUFuid	Yes	Yes	No	No
	OTP	Yes	Yes	No	No

8

Conclusion of the book. This chapter provides a concise review of the key points from all preceding chapters.

8. Conclusion

PUFrt and other types of NVM secure macro are the root of trust for modern SoC designs. As discussed throughout this book, these root of trust components must be tamper-proof in order to make the entire system secure. Physical attacks are the primary concerns that these root of trust macros face, as they can attack the chip from different angles, building up a comprehensive tamper-proof shell around our design has become a must these days.

In PUFrt, the hard macro part is secured by various anti-tampering features based on analog design techniques, including secure layout, balanced sensing, metal shielding, pin protection, and power detections. Furthermore, the RTL wrapper also provides another layer of protection through data/address obfuscation, random dummy insertion, privilege control, and redundancy checks. By combining these techniques, we can effectively make PUFrt tamper-proof.

Through the experiments in real silicon, insights about how physical attacks are practically conducted are presented. Being subjected to these real-world malicious attacks, the actual tamper resistance of PUFrt is thoroughly examined. It has been proven that PUFrt is well-protected against different types of attacks, and it is truly a tamper-proof root of trust design.

9

Appendix of the book, including Abbreviations, References, and Index.

Abbreviations

A	AMBA	Advanced Microcontroller Bus Architecture
	APB	Advanced Peripheral Bus
C	CPA	Correlation Power Analysis
	CRC	Cyclic Redundancy Check
D	DPA	Differential Power Analysis
	DRAM	Dynamic Random Access Memory
E	EM	Electromagnetic
	EMMI	Emission microscopy
F	FI	Fault Injection
	FIB	Focused Ion Beam
H	HTOL	High Temperature Operating Life
	HRoT	Hardware Root of Trust
	HUK	Hardware Unique Key
I	IC	Integrated Circuit
	IP	Intellectual Property
N	NIR	Near Infrared
	NVM	Non-volatile Memory
O	OBIRCH	Optical Bean Induced Resistance Change
	OTP	One-Time Programmable memory
P	PFA	Physical Failure Analysis
	PUF	Physically Unclonable Function
	PVC	Passive Voltage Contrast
R	ROSC	Ring Oscillator
	RTL	Register Transfer Level
S	SEM	Scanning Electron Microscopy
	SoC	System-on-Chip
	SPA	Simple Power Analysis

	SRAM	Static Random Access Memory
T	TEM	Transmission Electron Microscopy
	TRNG	Ture Random Number Generator

References

[1] S. Claude, "Communication Theory of Secrecy Systems." Bell System Technical Journal, vol 28, 4, 1949, pp. 656-715.

[2] P. Kocher, J. Jaffe, and B. Jun, “Differential power analysis,” *Annual International Cryptology Conference*., Springer, Berlin, Heidelberg, 1999.

[3] G. Piret, and J.-J. Quisquater, “A differential fault attack technique against SPN structures, with application to the AES and KHAZAD.” *International workshop on cryptographic hardware and embedded systems,* Springer, Berlin, Heidelberg, 2003.

[4] D. Nedospasov, et al., “Invasive PUF analysis.” *2013 Workshop on Fault Diagnosis and Tolerance in Cryptography,* IEEE, 2013.

[5] J. Chou, “PUFrt: Solving Chip Security’s Weakest Link”, Whitepaper, PUFsecurity

[6] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," Annual International Cryptology Conference., Springer, Berlin, Heidelberg, 1999.

[7] E. Brier, C. Clavier, and F. Olivier. "Correlation power analysis with a leakage model." International workshop on cryptographic hardware and embedded systems. Springer, Berlin, Heidelberg, 2004.

[8] G. Piret, and J.-J. Quisquater, "A differential fault attack technique against SPN structures, with application to the AES and KHAZAD." Int. Workshop on cryptographic hardware and embedded systems, Springer, Berlin, Heidelberg, 2003.

[9] S. P. Skorobogatov and R. J. Anderson. "Optical fault induction attacks." International workshop on cryptographic hardware and embedded systems. Springer, Berlin, Heidelberg, 2002.

[10] J. G. J. van Woudenberg, M. F. Witteman and F. Menarini, "Practical Optical Fault Injection on Secure Microcontrollers," 2011 Workshop on Fault Diagnosis and Tolerance in Cryptography, 2011, pp. 91-99.

[11] C. H. Kim and J. -J. Quisquater, "Faults, Injection Methods, and Fault Attacks," in IEEE Design & Test of Computers, vol. 24, no. 6, pp. 544-545, Nov.-Dec. 2007.

[12] M. T. Rahman et al., "Physical Inspection & Attacks: New Frontier in Hardware Security," 2018 IEEE 3rd International Verification and Security Workshop (IVSW), 2018, pp. 93-102.

[13] 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.” J. Emerging Technologies in Computing System, vol. 13, 1, Article 6, 2017.

[14] F. Courbon, S. Skorobogatov, and C. Woods. "Reverse engineering flash EEPROM memories using scanning electron microscopy." International Conference on Smart Card Research and Advanced Applications. Springer, Cham, 2017.

[15] D. B. Williams and C. Barry Carter. "The transmission electron microscope." Transmission electron microscopy. Springer, Boston, MA, 1996. 3-17.

[16] C. Kison, J. Frinken, and C. Paar, "Finding the aes bits in the haystack: Reverse engineering and sca using voltage contrast." Int. Workshop on Cryptographic Hardware and Embedded Systems. Springer, Berlin, Heidelberg, 2015.

[17] S. Skorobogatov, "How Microprobing Can Attack Encrypted Memory," 2017 Euromicro Conference on Digital System Design (DSD), 2017, pp. 244-251.

[18] C. Boit, C. Helfmeier and U. Kerst, "Security Risks Posed by Modern IC Debug and Diagnosis Tools," 2013 Workshop on Fault Diagnosis and Tolerance in Cryptography, 2013, pp. 3-11.

[19] C. Helfmeier, C. Boit and U. Kerst, "On charge sensors for FIB attack detection," 2012 IEEE International Symposium on Hardware-Oriented Security and Trust, 2012, pp. 128-133.

[20] H. Eren and Lucas D. Sandor. "Fringe-effect capacitive proximity sensors for tamper proof enclosures." 2005 Sensors for Industry Conference. IEEE, 2005.

[21] V. Immler, et al. "Secure physical enclosures from covers with tamper-resistance." IACR transactions on cryptographic hardware and embedded systems, 2019, pp. 51-96.

[22] Lau, J.H. (2021). “3D IC Integration and 3D IC Packaging.” in Semiconductor Advanced Packaging. Springer, Singapore. https://doi.org/10.1007/978-981-16-1376-0_7.

[23] J. -M. Cioranesco et al., "Cryptographically secure shields," 2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), 2014, pp. 25-31, doi: 10.1109/HST.2014.6855563.

[24] R. Shen, H. -M. Chen and M. -Y. Wu, "Highly reliable anti-fuse technology in sub-16nm technologies for security applications," 2016 International Conference on IC Design and Technology (ICICDT), 2016, pp. 1-4.

[25] C. Bozzato, Riccardo Focardi, and Francesco Palmarini. "Shaping the glitch: optimizing voltage fault injection attacks." IACR Transactions on Cryptographic Hardware and Embedded Systems (2019): 199-224.

[26] T. Korak, et al. "Clock glitch attacks in the presence of heating." 2014 Workshop on Fault Diagnosis and Tolerance in Cryptography. IEEE, 2014.

[27] K. M. Zick, et al. "Sensing nanosecond-scale voltage attacks and natural transients in FPGAs." Proceedings of the ACM/SIGDA international symposium on Field programmable gate arrays. 2013.

[28] E. Dubrova, “Fault-tolerant design.” Springer, New York, 2013.

[29] D. Agrawal, et al. "The EM side—channel (s)." International workshop on cryptographic hardware and embedded systems. Springer, Berlin, Heidelberg, 2002.

[30] P. Maurine, "Techniques for EM Fault Injection: Equipments and Experimental Results," 2012 Workshop on Fault Diagnosis and Tolerance in Cryptography, 2012, pp. 3-4.

[31] D. Gruss, C. Maurice, and S. Mangard, "Rowhammer. js: A remote software-induced fault attack in javascript." International conference on detection of intrusions and malware, and vulnerability assessment. Springer, Cham, 2016.

[32] N. A. Anagnostopoulos, et al. "Low-temperature data remanence attacks against intrinsic SRAM PUFs." 2018 21st Euromicro Conference on Digital System Design (DSD). IEEE, 2018.

[33] D. Das, S. Maity, S. B. Nasir, S. Ghosh, A. Raychowdhury and S. Sen, "High efficiency power side-channel attack immunity using noise injection in attenuated signature domain," 2017 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), 2017, pp. 62-67.

[34] K. Tiri, et al. "Prototype IC with WDDL and differential routing–DPA resistance assessment." International Workshop on Cryptographic Hardware and Embedded Systems. Springer, Berlin, Heidelberg, 2005.

[35] E. Prouff, and M. Rivain. "Masking against side-channel attacks: A formal security proof." Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, Berlin, Heidelberg, 2013.

[36] D. Flynn, "AMBA: enabling reusable on-chip designs." IEEE micro 17.4, pp. 20-27, 1997.

[37] M.-Y. Wu, "Hardware Root-of-Trust Design Based on on-chip PUF for AIoT Applications." 2022 International Symposium on VLSI Design, Automation and Test (VLSI-DAT), IEEE, 2022.

[38] M. -Y. Wu et al., "A PUF scheme using competing oxide rupture with bit error rate approaching zero," 2018 IEEE International Solid - State Circuits Conference - (ISSCC), 2018, pp. 130-132.

[39] B. Valtchanov, et al., "Characterization of randomness sources in ring oscillator-based true random number generators in FPGAs." 13th IEEE Symposium on Design and Diagnostics of Electronic Circuits and Systems, IEEE, 2010.

[40] W. W. Peterson and D. T. Brown, "Cyclic Codes for Error Detection," in Proceedings of the IRE, vol. 49, no. 1, pp. 228-235, Jan. 1961.

[41] M. Stitch, G. M. Johnson, B. P. Kirk and J. B. Brauer, "Microcircuit Accelerated Testing Using High Temperature Operating Tests," in IEEE Transactions on Reliability, vol. R-24, no. 4, pp. 238-250, Oct. 1975.

[42] M. Depas, T. Nigam and M. M. Heyns, "Soft breakdown of ultra-thin gate oxide layers," in IEEE Transactions on Electron Devices, vol. 43, no. 9, pp. 1499-1504, Sept. 1996.

[43] B. Kaczer, R. Degraeve, M. Rasras, K. Van de Mieroop, P. J. Roussel and G. Groeseneken, "Impact of MOSFET gate oxide breakdown on digital circuit operation and reliability," in IEEE Transactions on Electron Devices, vol. 49, no. 3, pp. 500-506, March 2002.

[44] J. C. H. Phang et al., "A review of near infrared photon emission microscopy and spectroscopy," Proceedings of the 12th International Symposium on the Physical and Failure Analysis of Integrated Circuits, 2005. IPFA 2005., 2005, pp. 275-281.

Index

Our Thanks

We dedicate this book to our friends, families and, of course, the whole team at **PUFsecurity** that made this project possible.

Together, we gave our time and effort to develop this from only a nascent concept into a fully realized series of books about PUF based technology. This would have never happened without your continued support and commitment.

So, to each of you, we say thank you!

PUFsecurity Corporation.

8F-1, No. 5, Tai-Yuan 1st St., Jhubei City,
Hsinchu County,
302082, Taiwan
Tel: +886-3-560-1010
www.pufsecurity.com