Leveraging SRAM for Counterfeit Detection and Secure 3DIC Integration

by

Gaines Odom

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
May 10, 2025

Keywords: SRAM, volatile memories, aging, data remanence, process variation, 3D ICs, heterogeneous integration, supply chain, provenance, whitelisting, blockchain

Approved by

Ujjwal Guin, Chair, Associate Professor of Electrical and Computer Engineering
Yadi Zhong, Assistant Professor of Electrical and Computer Engineering
Mehdi Sadi, Assistant Professor of Electrical and Computer Engineering

Abstract

Within the evolving domain of electronics security and counterfeit detection, researchers have historically placed a strong emphasis on techniques rooted in the analysis and recovery of non-volatile memory elements. These approaches often rely heavily on persistent data storage features and external references, such as golden chips or trusted databases, to validate authenticity and identify anomalies in suspect hardware. In particular, counterfeit integrated circuits (ICs) identification has been largely constrained to methodologies that assume the availability of a golden sample or access to other external sources of ground truth. While effective in controlled environments, such dependency significantly hampers scalability and real-world applicability, especially in distributed or resource-limited contexts.

In contrast, the detection of recycled or tampered volatile memory components—most notably, static random-access memory (SRAM)—has received comparatively little scholarly and industrial attention. This imbalance in research focus stems, in large part, from the widespread assumption that volatile memories lose all stored data immediately upon power-down. As a result, it is commonly believed that such memories offer little to no utility in postmortem security analyses. The volatile nature of SRAM has led to the prevailing view that these components are unsuitable for forensic examination or for use in security primitives that require persistent traceability. Consequently, recycled or subtly modified SRAM-based devices often evade detection using traditional security screening methods, leaving a critical blind spot in current counterfeit detection frameworks.

This thesis directly challenges these foundational assumptions by introducing a comprehensive and forward-thinking methodology designed to detect recycled SRAM-based electronics without relying on external references or trusted baselines. By leveraging the subtle,

yet repeatable, physical properties of SRAM cells that influence their power-up state under controlled conditions—including manufacturing-induced variations, process defects, and aging-related degradation—it becomes possible to derive device-specific signatures that persist beyond power loss. These signatures, when appropriately analyzed, provide a viable means of distinguishing authentic devices from recycled ones, even in the absence of traditional reference models. Experimental validation presented throughout this research demonstrates the feasibility and effectiveness of these proposed strategies under realistic conditions. Through a series of accelerated aging simulation experiments and SRAM state analyses, this work confirms that it is possible to detect recycled SRAM hardware without requiring any prior knowledge of its original, unaged behavior.

In addition to addressing the detection of recycled SRAM devices, this thesis further extends its contribution to the broader domain of hardware security by proposing an architecture for secure operation within heterogeneously integrated systems. With the increasing adoption of 2.5D and 3D integrated circuits, where dies from diverse fabrication origins are assembled into a single package, supply chain trustworthiness has emerged as a pressing concern. To this end, a high-level security concept is introduced that incorporates a whitelisting framework enabled by an SRAM-based logging mechanism. This logger passively monitors operational characteristics and verifies the legitimacy of chiplet activity against a pre-approved whitelist, thereby mitigating the risk posed by unverified or malicious components within the system-in-package. To strengthen the forensic and auditability aspects of this architecture, the design is further augmented with a blockchain-based ledger that records security-relevant events in an immutable and verifiable manner. The integration of blockchain technology ensures that tampering attempts or unexpected deviations from baseline behavior can be recorded transparently and traced back with cryptographic assurance.

By rigorously exploring and substantiating these novel concepts, this thesis makes a substantial contribution to the field of electronics security. It not only redefines the utility of

volatile memory in security-critical applications but also opens new pathways for counterfeit detection that are both reference-free and scalable. Ultimately, this work advances the state-of-the-art in memory-based forensics, device lifecycle validation, and secure system integration, setting the stage for more resilient and trustworthy hardware ecosystems in future semiconductor supply chains.

Acknowledgments

I would like to thank my advisor, Dr. Ujjwal Guin, for his support, ideas, assistance, and occasional scolding during my time as a graduate student at Auburn. His contributions to my research were critical to my success in my publications, and this thesis. I thank him, again, for the professional inroads he has helped pave for me.

I would like to extend my gratitude to the committee members of my thesis: Dr. Yadi Zhong and Dr. Mehdi Sadi, for their invaluable instruction, their interest, and taking their time to support and acknowledge my work.

I would also like to thank my colleague, Zakia Tamanna Tisha, for being an unfailingly supportive writing partner and assisting me in areas I was indisposed with during my time working in Dr. Guin's lab.

Additional thanks must be given to every undergraduate research assistant I have worked with. Their aid has been sine qua non for my success. Specific thanks to Amaar Ebrahim, Griffin Smith, and Joshua Paulsen (all current graduate students) for assistance to which I attribute my achievements in setting up and carrying out experiments.

My sincere thanks go out to all of my labmates for making researching in Dr. Guin's lab more enjoyable and rewarding. Your collaborative spirit and constant encouragement enriched both my academic and research experiences during my time as a Masters Student and Research Assistant at Auburn. Your support has meant very much to me.

Finally, I would like to express my gratitude to my parents for supporting and uplifting me during my journey through graduate school at Auburn. Your support is the foundation of my success, and nothing I've accomplished could have been done without you.

Contents

List of Figures

List of Tables

List of Abbreviations

SRAM	Static Random Access Memory
NMOS	N-channel Metal-Oxide Semiconductor
PMOS	P-channel Metal-Oxide Semiconductor
IC	Integrated Circuit
NBTI	Negative Bias Temperature Instability
HCI	Hot Carrier Injection
P1s	Presence of Logic-1s (expressed as percentages)
HI	Heterogeneous Integration

Chapter 1

Introduction

The turn of the 21st century has witnessed unprecedented growth and reliance on integrated circuits (ICs), driving innovation across a wide array of sectors, including communications, defense, computing, automotive systems, and critical infrastructure. In this modern era, dominance in semiconductor technology equates to national and economic security, much as dominance in traditional military resources has historically determined geopolitical power. Hence, securing electronic supply chains has become as critical as securing essential resources like food, water, and ammunition. The pursuit of advanced and efficient semiconductor designs has resulted in increasingly intricate IC architectures that, due to economic factors, have largely shifted offshore. Facilities abroad offer superior technology, reduced operational costs, proximity to related manufacturers, and significant governmental incentives, thereby dominating global production [1]. Notably, the Taiwan Semiconductor Manufacturing Corporation (TSMC) exemplifies this shift, expanding its market capitalization from approximately $14.84 billion in 1997 to over $1 trillion in 2024 [2]. Moreover, recent reports by the Bureau of Industry and Security highlight that 43% of electronic assembly processes are now conducted away from original equipment manufacturers (OEMs), often under limited oversight and trust conditions, thus intensifying concerns about intellectual property (IP) protection and authenticity [3–5].

The challenges posed by this offshoring trend are multifaceted, with threats like IP theft, unauthorized overproduction, piracy, and design tampering becoming increasingly prevalent. These threats compromise the integrity of the global electronics market, negatively impacting designers, manufacturers, and end-users alike. Significant efforts have been dedicated to developing effective countermeasures and robust detection methods [5–9]. However, another

prominent threat, recycling end-of-life (EOL) electronic components, remains inadequately addressed despite extensive research [10–12]. E-waste recycling involves extracting functional but aged IC components from discarded electronics, which are then reintroduced into supply chains under false pretenses. Although still operational, these recycled components are vulnerable to performance degradation and premature failure, posing significant risks in critical systems [12–15]. Furthermore, these components may retain sensitive data such as firmware or encryption keys, potentially exposing critical information to malicious entities [16].

1.1 Motivation

The semiconductor industry's globalization has dramatically accelerated advancements in integrated circuit design, fostering rapid evolution from system-on-chip (SoC) technologies to heterogeneous integration (HI) involving 2.5D and 3D IC packaging. This shift was necessitated by escalating demands for reduced latency, enhanced performance, power efficiency, and improved fabrication yields, particularly in high-performance computing (HPC), data centers, cloud computing, and AI-driven applications [17,18]. Prominent industry players, including TSMC and Samsung, are actively developing advanced HI solutions, further promoting this transition [19,20]. These advancements have also introduced sophisticated interconnect technologies such as the Universal Chiplet Interconnect Express (UCIe) and Open High-Bandwidth Interface (OpenHBI), enabling seamless die-to-die communications [21,22].

The demand for better performance has given rise to increasingly complex computing infrastructure. One such example are 3D ICs, smaller hardware modules separated into "chiplets" and integrated vertically, connected by through-silicon vias (TSVs) and placed into a singular package. A proposal made popular because of this is the premise of heterogeneous integration [17]. The idea is that the chiplets composing a 3D IC can now be fabricated at separate labs and integrated into a cohesive hardware design. Hardware Trojans—malicious alterations intentionally embedded within ICs during design or fabrication—pose an exponentially larger threat to 3D ICs due to their complex and densely

integrated nature. The intricate stacking and heterogeneous integration of multiple dies, often of varying or potentially dubious origin, in 3D ICs significantly increase the potential entry points for attackers, complicate detection efforts, and elevate the risk of functional disruption or information leakage. This complexity underscores the urgent need for enhanced Trojan detection and mitigation strategies specifically tailored to the unique vulnerabilities of advanced 3D integrated systems. We propose one such solution in this paper.

Moreover, the rising threat of counterfeit integrated circuits (ICs) and system-on-chip (SoC) obtained from discarded electronics being recycled and sold as new continues to grow due to the lack of effective detection techniques. The entry of such knock-off ICs into key systems comprising the critical global infrastructure can result in system and security failures with potentially disastrous consequences for societal well-being. IHS Inc. has reported that counterfeit ICs represent a potential annual risk of $169 billion in the global supply chain [23]. These recycled ICs often exhibit poorer reliability, reduced useful remaining lifetime, and degraded performance [10, 12]. The crude process of disassembly, cleaning, and restoration often employed to a recycled part as new can also create additional defects, resulting in electrostatic damage and other anomalies that can cause system malfunction [10–12, 14]. As most Department of Defense (DoD) infrastructures are designed well beyond their lifetime of electronics, they are critically dependent on a continuing supply of legacy commercial off-the-shelf (COTS) components for maintenance and repair, and often encounter recycled parts. It is thus essential to detect counterfeits efficiently to prevent the widespread infiltration of these parts in the semiconductor supply chain.

Although research towards securing electronics supply chains has seen recent progress [24–26], relatively little attention has been given to addressing threats that arise during runtime. Several studies have introduced techniques to defend advanced integrated circuits against hardware Trojans [27, 28]; however, these methods often fall short in mitigating risks that emerge after deployment. Despite decades of research in this area, reliably detecting hardware Trojans remains a persistent and complex challenge [9, 29].

It can be said succinctly that a variety of threats have become relevant concerns on account of the intricacies of the globalized semiconductor supply chain. These threats exist in many capacities throughout an IC's life cycle. For instance, these threats can emerge:

- Pre-integration, or prior to deployment, in cases of intellectual property (IP) theft.
- Post-integration, or during deployment, in cases of Hardware Trojan injection.
- Post-deployment, in cases of IC recycling.

It is because of these temporally persistent threats that nations take comprehensive measures to provide provenance and assure the security of their electronic hardware.

1.2 Contribution

In this thesis, we present a novel approach for detecting counterfeit devices utilizing SRAM architectures, such as CPUs, GPUs, FPGAs, discrete memory devices, and more [30]. We leverage the effects of negative bias temperature instability (NBTI) on PMOS transistors that cause changes in SRAM cells. These aging effets bias an SRAM cell towards the opposite value stored in the cell. Additionally, we propose a novel approach by which security vulnerabilities of heterogeneously integrated 2.5D/3D ICs introduced by a horizontal, global supply chain can be mitigated [31]. This approach bypasses the limitations of traditional hardware trojan detection methodologies by implementing a blockchain-enabled whitelisting framework, restricting the communication abilities of potentially malicious chiplets. There are four core contributions in the thesis, presented in topically-relevant pairs. This thesis introduces a series of novel approaches tailored specifically to mitigate critical threats in semiconductor supply chains, focusing on SRAM-based counterfeit detection and proactive runtime security in heterogeneous 2.5D/3D ICs. The key contributions of this thesis are summarized as follows:

- Firstly, we present an approach for providing the spatial uniformity to an SRAM power-up state necessary for fine-grain statistical analyses. Though many newer SRAM architectures boast low-bias designs, lending themselves to uniform power-up states, legacy SRAM devices often feature bitline interleaving, swapping the positioning of BL and $\overline{BL}$ and creating high amounts of capacitive bias along each column of SRAM cells. This cumulative bias results in "banded" power-up states, with certain "bands" being considerably more biased towards one discrete logic value or another.

- Secondly, we propose a novel method to detect counterfeit SRAM devices without golden samples or reference parameters. First, we pre-process the cumulative power-up state of the SRAM array, providing a relative amount of spatial uniformity. From here, we observe the processed cumulative power-up state of the IC over one hundred power-ups, wherein each SRAM cell is assigned a value from 0 to 100 based on how many times the cell powered up with logic-1. We divide this dataset into sections of varying sizes, or "blocks." Our analysis shows that aging-induced spatial non-uniformity can be identified by measuring the standard deviation (σ) of the average value held in each block.

- Thirdly, we introduce a high-level proposal for chiplet hardware for the purpose of securing 3D ICs against threats within their communication networks. We propose to implement a whitelist on the network I/O on the path to the chip-to-chip communication. The whitelist is implemented as a set of allowed source-destination pairs. Only messages from white-listed sources being sent to the corresponding white-listed destinations will be allowed out of the chip. We further propose adding an on-chip logger to network I/O. Any time there is a message that fails the whitelist, it generates an entry in the logger.

- Lastly, we offer a blockchain framework by which logger states - the sum of all failed communication attempts stored in the Network I/O's logger - can be appended to a

tamper-proof database, enabling the retrospective analysis and verification of communication events and potential security breaches within the system. The desktop or server hosting the blockchain client aggregates all the digitally signed transaction logs it has acquired from different chips. This consolidated set of logs is then appended to the blockchain, ensuring a tamper-proof record of the transactions.

1.3 Organization of the Thesis

The rest of this thesis is organized in the following manner: in Chapter 2, a description of SRAM, power-up states, and aging effects is provided. Our self-referencing approach for detecting counterfeit ICs using SRAM power-up states is explained and validated in Chapter 3. A novel means of securing 3D ICs after deployment is introduced in Chapter 4. Finally, I summarize my findings and highlight future research made possible by my contribution as I conclude this thesis (Chapter 5).

Chapter 2

Background and Related Work

In this chapter, we discuss the fundamentals of SRAM, aging, power-up states, 3D ICs, and other concepts fundamental to the approaches described in Chapters 3 and 4. We also discuss contributory research performed with respect to our frameworks.

2.1 SRAM

Static Random-Access Memory (SRAM) has been integral to the advancement of computing systems since its inception. SRAM is a volatile memory type characterized by its high-speed performance, simplicity in interfacing, and robustness against noise, making it suitable for various critical applications. The conventional SRAM cell employs a six-transistor (6T) architecture consisting of two cross-coupled CMOS inverters forming a bistable latch, complemented by two additional transistors that act as access gates.

The bistable latch configuration provides SRAM with the ability to store binary data reliably without the need for continuous refresh cycles, in contrast to Dynamic RAM (DRAM). This makes SRAM significantly faster and ideal for cache memory applications, including Level 1 (L1), Level 2 (L2), and Level 3 (L3) caches within microprocessors, substantially enhancing computational speed and efficiency.

Beyond performance-oriented applications, SRAM is widely utilized in security contexts, particularly in Physically Unclonable Functions (PUFs). The inherent variability in the manufacturing process of SRAM cells produces unique, device-specific fingerprints that can be exploited as security primitives. These SRAM-based PUFs offer robust solutions for device authentication, cryptographic key generation, and secure identity verification.

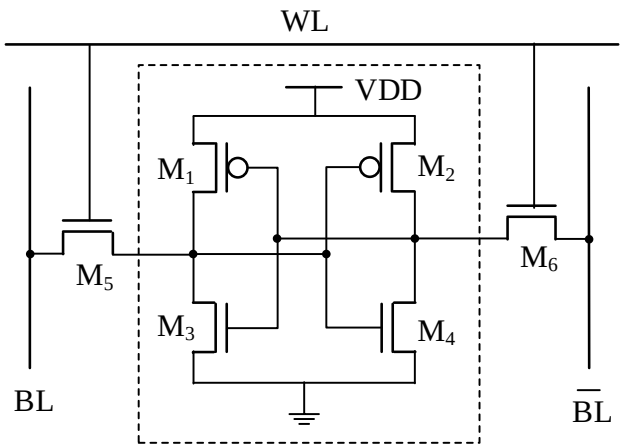

Figure 2.1: A traditional 6T SRAM architecture.

Due to its speed, reliability, and versatility, SRAM remains a fundamental component in modern electronic systems, driving ongoing research into enhancing its security, reliability, and performance under various operational and environmental conditions.

2.1.1 Power-Up States

SRAM cells are designed with perfect symmetry, as discussed in Section 2.1. A classical SRAM cell is shown in Figure 2.1. A typical SRAM is composed of two CMOS inverters, each having a PMOS-NMOS pair (where $M1$ & $M2$ act as PMOS transistors, while $M3$ and $M4$ act as NMOS transistors), as well as two access transistors connecting to bitline and bitline-bar for read and write operations ($M5$ and $M6$). Inverters are designed to be equal in scale and share identical features. When a 0 is applied to the gate of one inverter, a 1 is output from that inverter and subsequently applied to the gate of the other inverter, which outputs a 0 – in simpler terms, an SRAM's cross-coupled inverters create a latch that holds the logic values applied to them.

While SRAM cells are designed with perfect symmetry, SRAM devices are subject to bias - variations in bitline capacitance, channel length/width, doping concentration, or oxide thickness between inverters may cause mismatch, or a deviation from metastability. These

biases cause mismatch between the inverters, subsequently leading to differences in threshold voltage (V_T) values.

Leakage current significantly impacts the power-up state of SRAM cells, influencing the cell's initial state upon power restoration. Variations in transistor leakage currents caused by aging, temperature fluctuations, and manufacturing processes can create discernible imprints of previously stored data. These subtle changes facilitate data remanence attacks, highlighting potential vulnerabilities and underscoring the necessity of considering leakage currents in the secure design and disposal of SRAM-based systems.

A substantially large V_T differential between the two PMOS transistors present in the circuit will lead to variations in its power-up state. When, due to leakage current coming from across the channel of the PMOS transistors, a charge builds on the parasitic capacitance (capacitance present because of the cumulative dielectric properties of the materials composing the cell) at the output nodes of an SRAM cell, a charge builds on these nodes, which feeds into the gates of the inverters. When this voltage value becomes significantly large enough, it a "bridge" will create across the PMOS channels, and a 0 will be inferred from the output of the activating inverter. This will latch a logic value into the SRAM cell.

When a device is powered on, this phenomenon takes hold across the entire SRAM array, and all of its SRAM cells will be initialized to a discrete logic value. These initialization values are distinct to each cell, and principally caused by the V_T of its PMOS transistors.

2.1.2 Aging

The hypothesis of differential aging within SRAM cells posits that, over time, these cells may undergo distinctive aging degradation, thereby impacting their powerup characteristics. This premise delves into the dynamic nature of aging degradation, emphasizing the influence of non-uniform data stored in the array, which contributes to nonuniformity in the aging process. The initial power-up state of a new device exhibits uniformity, determined almost entirely by random manufacturing process variation. Previous research has indicated that

subjecting an SRAM cell containing a particular logic value during the aging results in a gradual reduction of the probability of the memory cell initializing to the same logic value upon successive power-ups [32].

Extensive study has been conducted to explore the factors contributing to aging degradation like negative bias temperature instability (NBTI) [33,34], hot carrier injection (HCI), and ionizing radiation [35]. NBTI causes asymmetric shifts in threshold voltages of SRAM cell transistors. When the PMOS transistors are negatively stressed, interface traps are created at the Si-SiO2 interface of PMOS transistor when the gate input is logic 0. Releasing the stress causes changes in threshold voltage of PMOS transistors [36]. Guin et al. [32] showed that the asymmetric shift in threshold voltages is responsible for the imbalance of 1s and 0s in the power-up state of an SRAM cell. Furthermore, the bits in the power-up state are determined by the contents of the aging data in the SRAM. This aging effect in the power-up state of SRAMs has been leveraged to develop a detection technique of recycled or aged ICs in the same study. A later study [30] showed that even COTS SRAMs, which do not power up with an equal distribution of 1s and 0s in their initial state, also possess an identifying aging characteristic: the percent of 1s in all subregions of a new SRAM cell is more statistically similar to each other than in an aged IC.

While the aging impacts of SRAMs have been widely covered in literature, very little has been explored about how aging influences data imprint in SRAMs, resulting in data remanence effects. Hovanes et al. [16] were the first to present an approach of data recovery from SRAMs under normal operating conditions. Their work showed that contrary to the conventional wisdom that SRAM loses its contents once powered off, SRAM actually develops an imprint of the stored values in the power-up state due to aging. The authors noted that the locations of SRAM cells that stabilize after aging differ from chip to chip to uncontrollable process variations, making the cells more likely to be stable or unstable as the chip ages. The hypothesis of this approach is based on the findings in [32], which state that the power-up state of SRAM ages inversely relative to the binary aging data. This makes it possible

to discern the data contained in SRAM by observing the differences between the power-up states before and after aging. The approach involves recovering data from multiple chips by using a majority voting method to improve the data reconstruction.

The reliability principles of NBTI and HCI state that when a PMOS transistor is aged with a voltage level consistent with logic-0 applied to its gate, its threshold voltage (V_t) will invariably rise due to heat-related degradation [32]. This is relevant to SRAM cells as, to re-iterate, a typical 6T SRAM cell is composed of a symmetric paired CMOS inverter structure, acting as a bistable latch which holds logic values in place, with two access transistors.

As a typical SRAM cell's structure is perfectly symmetrical, its power-up state is determined entirely by incidental biases and noise. Should one of the CMOS inverters "overpower" the other (i.e., have a lower PMOS threshold voltage), it will be the determinant of the logic value of the SRAM cell upon power-up. Because aging the PMOS transistor in a CMOS inverter with logic-0 raises its threshold voltage, and because an SRAM cell consists of two back-to-back latching inverters, that by aging an SRAM cell with logic-0, the likelihood of a 0 occurring on power-up decreases. Similarly, if an SRAM cell is aged with logic-1, the opposite inverter is aged with logic-0, making a logic-0 more likely upon power-up.

When this phenomenon is observed across the multitude of SRAM cells on a device, it can be noticed that an aging dataset creates a logical imprint onto its SRAM array. That is to say, for every cell that which is aged with 1, a 0 becomes more apparent, and for every cell that which is aged with 0, the opposite is true. Of course, because of Gaussian process variation, fabricated SRAM arrays are rife with cells that are not functionally symmetrical – in fact, most are not. Typically around 80% of SRAM cells have an inverter that reliably overpowers another, making these cells stable at a particular logic value. In some cases, this stability can be undone by aging with an opposing logic value, making them unstable, and in other cases, this stability is reinforced by aging with an agreeing logic value [16].

2.1.3 Counterfeit Detection

In Section 1.1, we discussed the threat posed by counterfeit and recycled ICs to the electronics supply chain and critical infrastructures. Because of these threats, measures to identify recycled electronics must be devised. Much research has been done in this field. Due to the inherently Gaussian nature of process variation, every SRAM cell is typically different, and parametrically asymmetrical. However, over a large number of cells, Guin et al [32] posits that a relatively equal number of logic-1 and logic-0 values will appear in an SRAM array upon power-up. As discussed in Section 1.1, recycled integrated circuits (ICs) pose a significant threat to the electronics supply chain and critical infrastructure. However, the characteristics associated with recycled devices can actually be leveraged to identify previously used SRAM components. Guin et al. demonstrated that the power-up state of an SRAM cell is influenced by the threshold voltages of its PMOS transistors, and these voltages, in turn, are impacted by the data stored in the cell during aging, as described earlier in Section 2.1.1. While individual SRAM cells can show significant variations due to manufacturing differences, collectively, these variations follow a Gaussian distribution, typically resulting in an approximately equal balance of logic-1 and logic-0 values. However, typical aging data stored in SRAM often contains significantly more logic-0 than logic-1 values. This uneven aging due to Negative Bias Temperature Instability (NBTI), as discussed in Section 2.1.2, skews the overall power-up state toward logic-1, allowing recycled SRAM components to be identified through their biased power-up patterns. This detection technique can effectively identify and remove counterfeit parts from the supply chain. Additionally, the usefulness of this approach extends beyond standalone SRAM chips, as most modern electronic systems incorporate SRAM that can be assessed using this method.

This observation implies a 50% time zero reference parameter (i.e., 50% of cells will store 1 on power-up prior to operation). While this position is true in some cases, particularly for SRAM designs utilizing modern process technology, it becomes difficult to establish for SRAM designs with tendentially high amounts of bias. For instance, in the 23LC line of

discrete SRAM ICs, bitline interleaving is used to aid in cumulative cell balancing. High amounts of capacitive bias build on bitlines (BLs), which leads to a strong bias upon power-up towards either a logic-1 or a logic-0. Further, the assumption that a deviation from this 50% reference parameter is indicative of significant use relies on aging data (the data loaded onto the device throughout its operational lifetime) being loaded relatively unequally with a dense amount of either logic-1 or logic-0 values. In cases where $Aging\ Data \not\ll 50\%$ or $Aging\ Data \not\gg 50\%$ (where 50% indicates an equal concentration of 1s and 0s, or a 50% presence of 1s), this approach struggles to effectively identify recycled ICs, particularly in consideration of the duration of operation necessary to make a substantiative impact on power-up values. A recycled IC could, in theory, be aged with all 1s for the first half of its addresses, and all 0s for the second half, and still maintain a 50% reference parameter. In summary, this approach faces two key issues:

1. It cannot be assumed, in all instances, that a COTS SRAM has an equal presence of logic-1 and logic-0 values upon power-up due to potential biases inherent in existing process technologies.

2. The assumption that the continued use of an SRAM will lead to a significant disruption of its 0/1 balance is not reliable, due to the unpredictability of aging datasets.

Of course, this approach can be modified on a chip-to-chip basis; A chip with a typical time zero reference parameter (obtained from a sufficiently large sample set of golden samples) of 70 $p1s$ (percentage of 1s) can be assumed to be counterfeit-likely if its power-up state demonstrates 50 or 90 $p1s$. A problem seen here is that it becomes necessary for a test lab to have access to typical time zero $p1s$ readings for all manner of integrated and discrete SRAM arrays, obviously with a certain amount of allowable deviation. Further, this adaptation is still predicated on the assumption that a chip's 0/1 balance will be markedly impacted by continued use.

We seek to improve upon this $p1s$-based counterfeit detection approach by addressing the two key issues faced by it. Rather than relying on an overall imbalance of 1 and 0 values in an entire SRAM array's collective power-up state, we will instead exploit an SRAM array's spatial $p1s$ incoherence to detect recycled chips without any need for a golden sample or zero time reference parameter. A benefit implicit in our approach is that it accounts for more variability in aging datasets. While a tendency away from 50 $p1s$ may be hard to assume or establish over a long period of time, a tendency for spatial disuniformity is much more likely, especially for SRAM cells utilized for firmware, etc., As the same data is loaded to them every power-up.

2.1.4 SRAM PUFs

SRAM power-up behaviors have enabled researchers to develop various physically unclonable functions (PUFs) and true random number generators (TRNGs) leveraging SRAM architecture [37–45]. Ideally, SRAM cells are designed with perfect symmetry to achieve the maximum static noise margin. However, unavoidable process variations introduce biases within each cell, causing a preference toward one inverter and leading to consistent power-up states over multiple cycles. Due to this inherent bias and external noise, SRAM power-up states can typically be classified into three categories: cells stable at logic 1 (S1), cells stable at logic 0 (S0), and unstable cells. Stable cells occur when the threshold voltage difference (V_{th}) between the cell's transistors (M1 and M2) significantly exceeds the external thermal noise. Conversely, unstable cells have a small enough V_T difference to be influenced by external noise, resulting in variable power-up states.

PUFs are widely studied hardware security primitives with numerous practical applications. A PUF is defined as a function whose precise behavior cannot be modeled or duplicated on identical architectures, even when its internal structure and physical access are known. This unclonability arises from the intrinsic, uncontrollable, and unpredictable process variations in transistor manufacturing, making PUFs effective for authenticating

hardware identity and ensuring device authenticity [46–48]. Over recent decades, researchers have explored various PUF architectures, including arbiter PUFs and ring oscillator PUFs (ROPUFs) [49–51]. Despite their effectiveness, these specialized PUF architectures are challenging to implement broadly since they require specific hardware designs within the chip.

SRAM-based PUFs, however, leverage existing SRAM components found in virtually all modern integrated circuits. An SRAM PUF is realized by extracting stable cells from the power-up states of SRAM arrays [37, 38, 40, 41, 52]. The resulting distribution of stable cells provides a device-specific fingerprint due to inherent process variations, allowing for secure and unclonable device identification. Because the internal transistor biases—and thus power-up states—can shift due to aging, SRAM PUF implementations often incorporate error correction mechanisms [44, 45, 52, 53]. Such error correction ensures the stability and reliability of the generated IDs over the device's lifetime, even under conditions of heightened external noise or environmental extremes, which could otherwise cause previously stable cells to become unstable.

2.2 3D ICs & Heterogeneous Integration

The globalization of the semiconductor supply chain brings rapid research and development (R&D) of chip fabrication and design, as well as swift adoption of the latest technology node. System-on-chip (SoC) has evolved in the past decades to combine different intellectual properties (IPs) into one design layout, and thus, a single die with multiple functions in one chip. However, the intensive computation workload in today's high-performance computers (HPC), data centers, cloud computing, and machine learning applications demands innovations beyond the current state-of-the-art SoC status quo. Driven by the need to further reduce latency and power consumption, increase throughput, and a better yield in IC fabrication, heterogeneous integration (HI) and 2.5D/3D packaging emerge as the new technological solution. This allows the horizontal and vertical stacking of multiple dies in a single package/chip, analogous to a system of mini-chips than the monolithic IC in SoC [17, 18]. It is

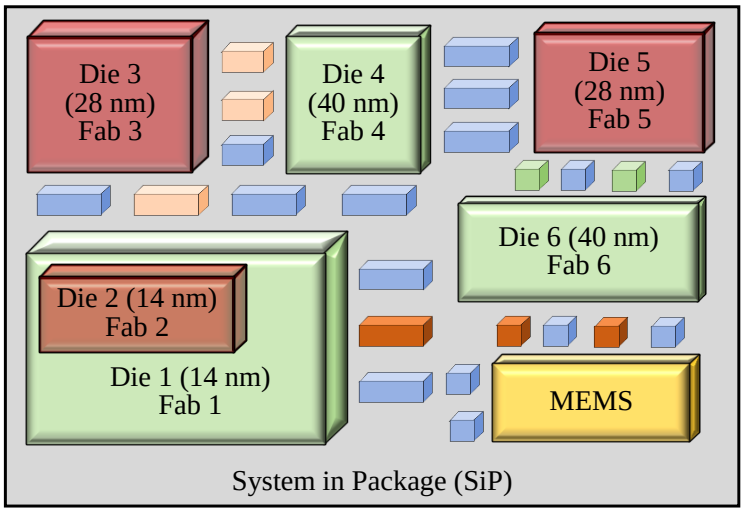

Figure 2.2: An abstract representation of heterogeneous integration where multiple chiplets are assembled in a SiP.

actively being researched and developed by multiple entities in the supply chain, e.g., TSMC 3DFabric^{TM} a 3D silicon stacking and advanced packaging technologies [19], and Samsung 3D-TSV (12 layers) DRAM Chip [20]. A ubiquitous interconnect, e.g., Universal chiplet interconnect express (UCIe) [21] and open high-bandwidth interface (OpenHBI) [22], at the package level to cover die-to-die (D2D) communication has been developed.

Unfortunately, the globalization of the semiconductor supply chain also opens the door for various threats to US critical infrastructures, where they are targeted by untrusted electronic products, counterfeit ICs, and devices with hardware Trojans [10, 12]. These threats originated from malicious third-party IP vendors, untrusted manufacturing facilities, and rogue distributors, including pirated and maliciously modified IPs, cloned and recycled ICs, etc. Bloomberg reported in 2018 and 2021 that the groundbreaking hardware hack with an extra tiny chip, covertly placed on board, can breach sensitive data from US companies [54, 55]. Although the published hack targeted pre-HI hardware, it is possible that an adversary can still execute similar hacks by incorporating malicious die(s) inside 2.5D/3D packages. When the hardware is compromised due to hardware Trojans in the chip or malicious chiplets, existing and additional software attacks can be mounted for malicious

purposes. When hardware is not authentic (compromised or cloned), the firmware and software running on it can be exploited by the attacker – easily bypassing the existing security measures implemented at the software level as the entry point to gain access to the device and/or system.

Chapter 3

Self-Referencing Approach for Recycled IC Detection

This section presents a novel self-referencing methodology for efficiently identifying counterfeit ICs [30]. The proposed approach eliminates significant expenses and the necessity of storing complex device parameters by employing a simple, cost-effective test setup to read the SRAM power-up state using a low-cost Raspberry Pi. The statistical comparison of power-up states between aged and new chips clearly and straightforwardly demonstrates differences. By analyzing the comparative standard deviation (σ) in the percentage of logic-1's ($p1s$) across multiple block sizes, a recycled chip can be reliably distinguished from a new one. The effectiveness of this self-referencing test has been successfully validated across six distinct ICs. This section begins with the establishment of the threat model.

3.1 Threat Model

Electronic supply chains face significant threats, notably through the inadvertent reintroduction of recycled components into critical applications. This approach assumes that an end-of-life SRAM device could re-enter our supply chain, become integrated into sensitive equipment performing critical functions, and subsequently fail, causing substantial operational disruptions, damage, or even loss of life. The recycling process for electronic components typically begins with genuine Original Component Manufacturer (OCM) parts that have initially been integrated into devices or equipment and subsequently discarded by end-users. These scrap electronics often find their way to reclaiming facilities, frequently located in developing nations, where circuit boards and components are crudely extracted under harsh, high-temperature conditions. Once extracted, these significantly aged and degraded components are repackaged for resale, thus reintroducing compromised devices into

the global supply chain [12]. Alarmingly, over 80% of counterfeit components originate from such recycled sources [56]. Despite improvements, recycling practices remain insufficiently regulated, with proper recycling rates in the United States historically as low as 10-18% in 2005, rising only modestly to approximately 25% by 2009. Consequently, a considerable portion of electronic waste continues to pose severe security and reliability risks upon re-entry into the supply chain.

To reiterate, this counterfeit detection schema does not require access to a golden sample or extensive pre-deployment characterization. As such, the detection schema can be deployed easily and cheaply by all manner of test labs and consumers.

3.2 Proposal

Figure 3.1 shows the overall concept of recycled IC detection using the self-referencing approach. In the power-up state of a new SRAM device, logic 1s, and logic 0s are uniformly distributed due to Gaussian manufacturing process variation, as depicted in Figure 3.1(a). Consequently, blocks of varying sizes, such as B_{01}, B_{02}, and smaller ones like b_{21}, b_{56}, will exhibit similar statistics of logic 1s, denoted as $p1s$. Conventionally, the data stored in SRAM during normal operation, such as firmware for an IoT device, does not exhibit a uniform distribution of 1s or 0s. Instead, there are distinct regions with varying concentrations of 1s and 0s, as illustrated in Figure 3.1(b). The aging process of an SRAM, influenced by non-uniform data, exerts a notable impact on the subsequent power-up states. This imprinting effect is evident in Figure 3.1(c). Consequently, regions like b_{21}, which were initially uniform, undergo a shift towards a higher presence of 0s when aged with predominantly 1s. Conversely, regions like b_{56} exhibit an increased presence of 1s after being aged with predominantly 0s.

We exploit this phenomenon by examining the occurrence of the percentage of 1 values ($p1s$) in memory cells on a block-by-block basis and monitoring the standard deviation (σ) of $p1s$. Aging effects lead to a deterioration in the uniformity of the SRAM power-up state. Consequently, the discrepancy between blocks intensifies, resulting in an elevated recorded

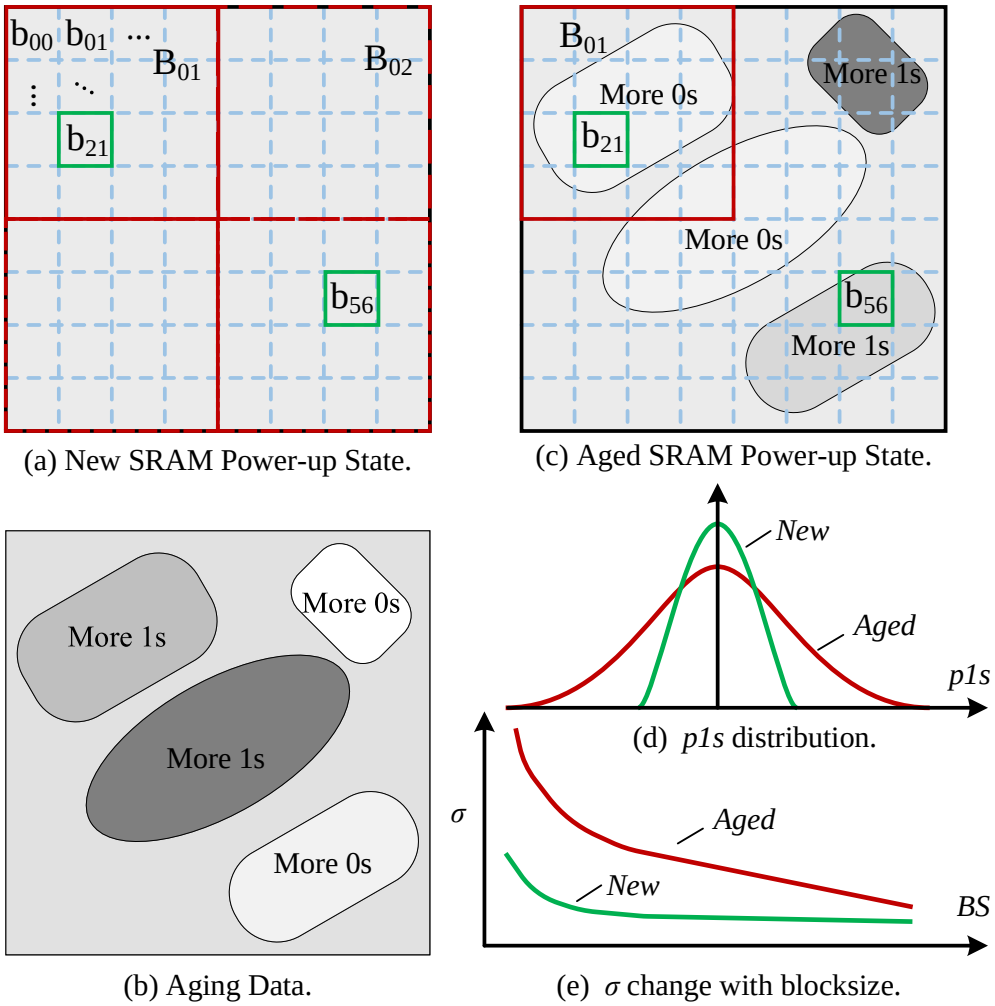

Figure 3.1: Overall self-referencing concept.

σ value, as illustrated in Figure 3.1(d). Finally, the shift in σ with varying block sizes illustrates the distinction between a new chip and an aged one, shown in Figure 3.1(e). This discrepancy arises due to the nature of the clustering of 1s and 0s across a power-up state. When considering the $p1s$ of a larger block like B_{01}, it tends to be akin to that of B_{02} even after aging. In such expansive regions, identifying non-uniformity becomes challenging, as concentrations of 1s and 0s tend to balance out overall. However, when focusing on smaller regions like b_{21} or b_{56}, these differences become more apparent, allowing for a clearer observation of concentrated pockets of 1s and 0s. Given the minimal non-uniformity in the power-up state of a new device, $p1s$ values remain consistently similar, leading to a more

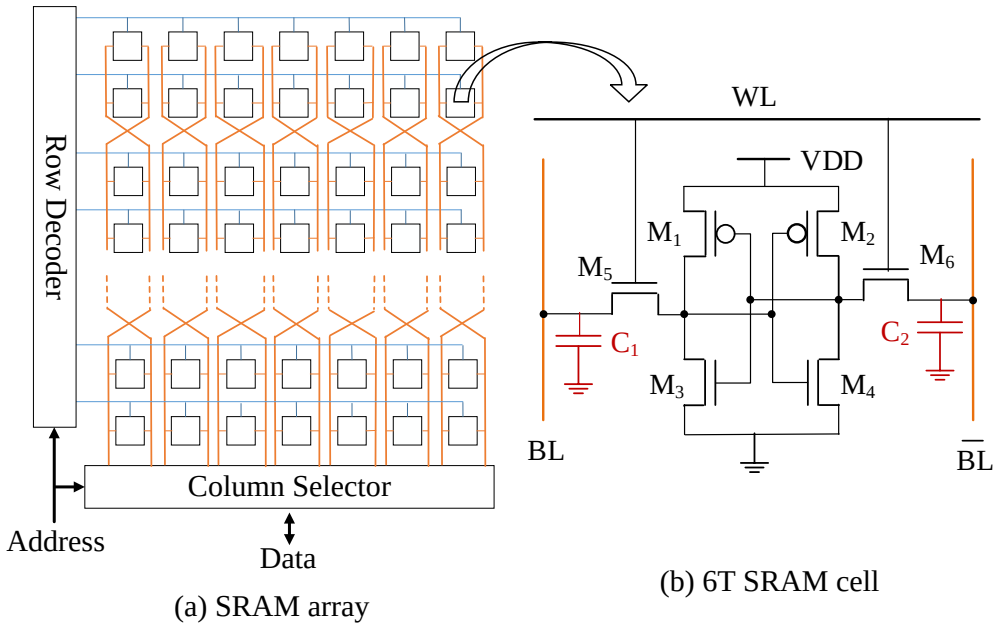

Figure 3.2: An SRAM array with bitline twisting.

stable σ with less variance across block sizes. By contrasting σ with block size, a definitive distinction between aged and new SRAM COTS becomes evident.

While the assumption that the power-up state of an SRAM is uniform holds true for newer technologies, legacy SRAM devices often feature bitline twisting, also known as bitline interleaving. As shown in Figure 3.2, this process swaps the positioning of BL and $\overline{BL}$ after a certain number of memory cells. This process typically occurs two or more times within one device, causing the bitmaps of these legacy chips to consist of "bands" in accordance with the number of twists. These bands are inherently biased to logic 1 or logic 0, causing a bitmap to inherit dark or light regions upon power-up. Though all SRAM cells are designed with perfect symmetry, the cumulative effect of node capacitors can skew the overall BL and $\overline{BL}$ capacitances [57], biasing portions of the power-up state to 0 or 1. It is because of this factor that the power-up state of a legacy SRAM device must be pre-processed.

Figure 3.3 shows the proposed flow of the self-referencing approach to detect recycled COTS SRAM chips manufactured with older technology nodes. The chip is first powered up to obtain the initial power-up states. To address the bitline flipping in legacy SRAMs,

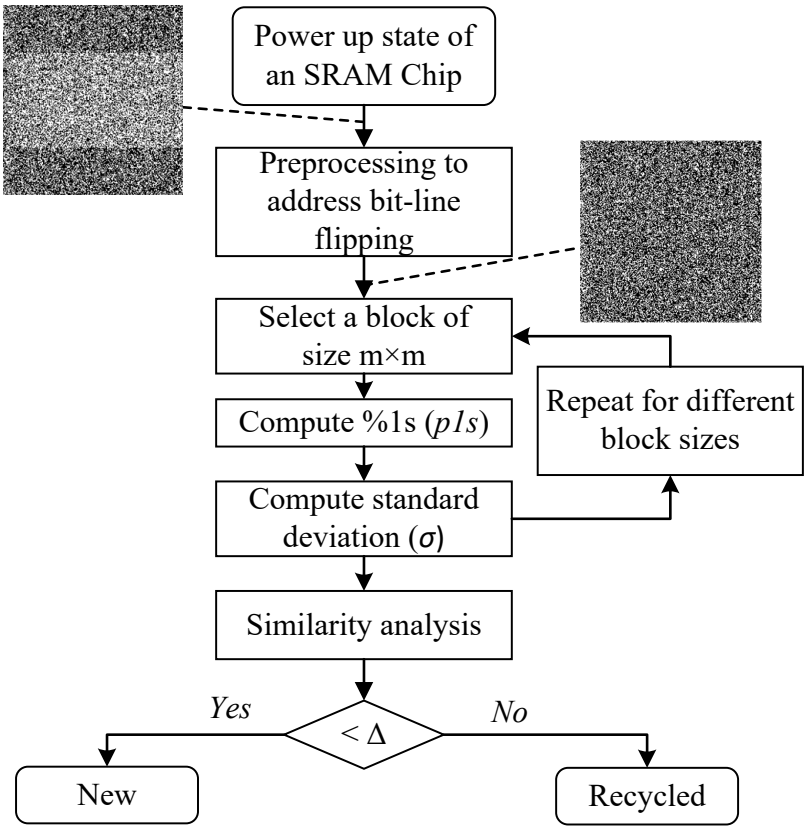

Figure 3.3: Proposed approach for detecting COTS SRAM chips.

it becomes necessary to pre-process the bitmap. This pre-processing operation involves inverting the power-up states of the complementary bands. Through this step, a relatively consistent power-up state is achieved, devoid of any discernible bands. After pre-processing, the uniform bitmap is split into blocks of size $m \times m$. The percentage of 1s, $p1s$, in the memory cells of each block is then computed. Following the computation of $p1s$, the standard deviation (σ) of the $p1s$ in each sub-block is calculated. This entire sequence, spanning from the division of the bitmap into blocks to the calculation of σ, is carried out for a range of n values. Subsequently, a comparative analysis is conducted between the blocks of varying sizes. This can be achieved by assessing the spread, σ, of $p1s$ for each block size. A decision on whether a chip is recycled can be made if the σs are well separated across the blocks. This is due to differential aging that creates an increased difference of $p1s$ in the power-up state.

Algorithm 1: Preprocessing of SRAM power-up states.

Input : SRAM power-up state (OPS), number of chip sections (B), sections to flip (F)
Output: Bitline compensated power-up state ($BCPS$)

```
1  function setSectionLimits (OPS,B) is
2      BL ← length(OPS)/B;
3      PSS[][] ← ∅;
4      for i ← 0 to B do
5          PSS[i][0 : (BL − 1)] ← OPS[i ∗ BL : ((i + 1) ∗ BL − 1)];
6      end
7      return PSS;
8  end
9  function invertSection (PSS) is
10     NPS[][] ← ∅;
11     foreach (i, j ∈ |PSS|) do
12         NPS[i][j] ← ¬PSS[i][j];
13     end
14     return NPS ;
15 end
16 function preProcessing (OPS, B, F) is
17     BCPS[] ← ∅;
18     PSS ← setSectionLimits(OPS, B);
19     for k ← 0 to B do
20         if (k == F) then
21             BCPS[k] ← invertSection(PSS[k]);
22         else
23             BCPS[k] ← PSS[k];
24     end
25     return BCPS;
26 end
```

On the contrary, σs of the $p1s$ of the new chips are expected to display greater resemblance across various block sizes.

Algorithm 1 outlines the overall pre-processing approach to compensate for the effect of bitline flipping for a legacy COTS SRAM power-up state. It is required to invert the flipped regions. We denote the original power-up state as OPS, the total number of regions in the chip as B, and the particular sections that need flipping as F. The `setSectionLimits` function takes OPS and B as inputs. The band length BL can be described as the total size of one region of OPS or the total size of OPS divided by B, Line 2. To divide the

Algorithm 2: Computation of standard deviation (σ).

Input : n $BCPSs$
Output: σ of $p1s$ for different block sizes.

```
1  function computePercentOnes (BCPS[]) is
2      block[][] ← ∅, X[][] ← ∅;
3      b[] ← Block sizes;
4      foreach ( m = 1 : |b| ) do
5          foreach ( i = 1 : n) do
6              foreach (j = 1 : b[m]) do
7                  block ← extractSubBlock(BCPS[i], j) ;
8                  X[m, i + j] ← calPercentOnes (block) ;
9              end
10         end
11     end
12     return X ;
13 end
14 function computeSigma (BCPS[]) is
15     X ← computePercentOnes(BCPS[]) ;
16     foreach ( m = 1 : |b| ) do
17         σ[m] ← standardDeviation (X[m, :]) ;
18     end
19     return σ ;
20 end
```

original power-up state into regions, the contents of each section are stored in a 2D sectioned power-up state array PSS, where each row has the contents of one region, Lines 4-6. The inversion of a region can be accomplished by the `invertSection` function, Lines 9-14. It takes PSS as its input, Line 9. A new power-up state array NPS is populated with the inverse of all elements in the input array PSS, Lines 10-12. The `preProcessing` function describes the general functionality of the pre-processing method, Lines 15-25. It takes inputs OPS, B, and F. The power-up state is broken into a sectioned power-up state array, Line 18. The bitline compensated power-up state is produced by inverting all sections indicated by input parameter F, Lines 19-25.

Algorithm 2 shows the computation of standard deviation, σ, for different block sizes. As we measure the power-up state multiple times, it takes n bitline compensated power-up states ($BCPS$) as the inputs and results σs for different block sizes. First, the `computePercentOnes`

function computes $p1s$ and is described in Lines 1-14. These $p1s$ values are used to calculate the standard deviation (σ) for their corresponding number of sub-blocks, Lines 15-21. The algorithm initiates with the input of an array of bitline compensated values, denoted as $BCPS[]$, in the `computePercentOnes` function, Line 1. The initialization of the 2D arrays, *block* and X, that represent a part of the power-up states and the percentage of logic 1s within these states, respectively, are described in Line 2. Another array, b, is initialized and denotes the number of blocks for which $p1s$ is to be computed and shown in Line 4. The function iterates through all power-up states, sub-divides them into blocks using the values in b, calculates the percentage of logic 1s, and stores these values in the array X, Lines 5-14. Algorithm 2 invokes `computeSSigma` function that takes n $BCPS$ as inputs, calls `computePercentOnes` iteratively to compute σ of $p1s$ for each block size across all power-up states, Lines 15-20.

3.3 Experimental Results

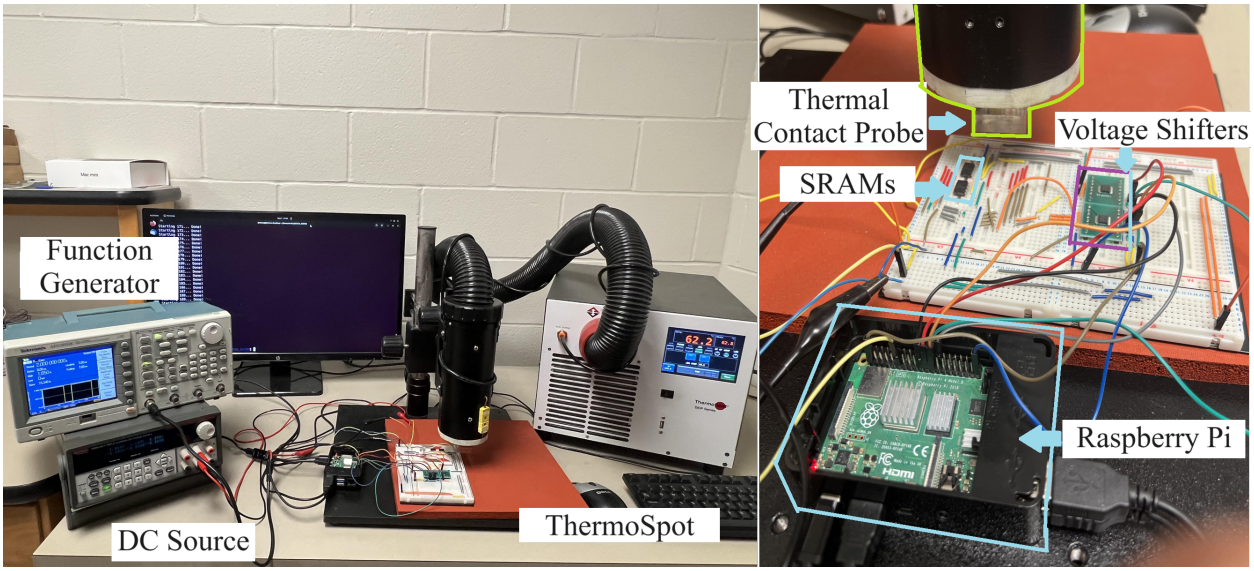

Figure 3.4: Experimental setup of accelerated aging using Thermospot system.

To validate the effectiveness of our proposed approach, we performed experiments with six different commercial off-the-shelf (COTS) 23A640-I/SN SPI Bus Low-Power Serial SRAM

memories [58]. Each SRAM chip had a total memory capacity of 64K bits. Figure 3.4 provides a detailed view of the experimental setup. The accelerated aging of the chips was performed using a Temptronic ThermoSpot DCP-201 system at the constant temperature of 100°C. All SRAMs were aged with the same black-and-white binary image to reflect the operational aging. One hundred power-up states of an SRAM were recorded every 6 hours of aging. This power-up state data was collected by the Raspberry Pi through the SPI interface. During power-up state collection, the SRAM was powered by a PWM signal from a function generator, generating a power-up state every two seconds. Simultaneously, a custom C program allowed the Raspberry Pi to read the complete power-up state at the positive edge of this signal. The collected SRAM data were then pre-processed using Algorithm 1 to compensate for bitline flipping via in-house Python scripts. Following that, the data was analyzed using Algorithm 2 in MATLAB.

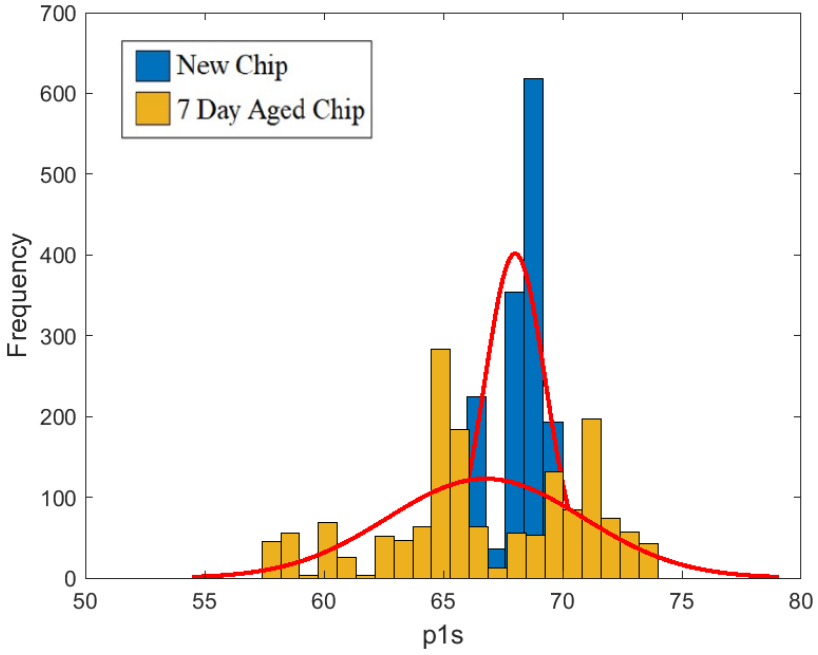

Figure 3.5: Distribution of $p1s$ of 64x64 blocksize of the power-up states of an SRAM Chip.

To validate our hypothesis on differential aging, we conducted experimental analyses of the distribution of 1s for various block sizes in both a new and an aged chip. As outlined in Section 2.2, we anticipate that an aged chip will demonstrate a broader distribution of

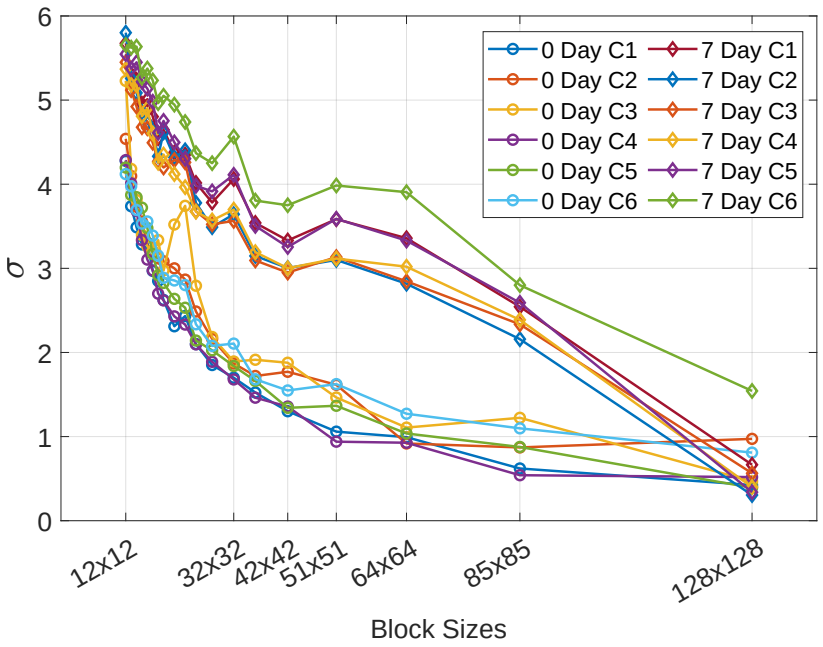

Figure 3.6: Standard deviation (σ) as a function of block size for new and aged SRAM chips.

$p1s$ compared to a new chip, especially for a specified block size. Illustrated in Figure 3.5 is the $p1s$ distribution in the power-up states of a new and an aged SRAM chip, focusing on a block size of 64×64. Notably, as the chip ages, the $p1s$ distribution displays a wider spread, coupled with a leftward shift of the mean. The mean of $p1s$ values has a left-bound tendency over time because more 1s are stored at the SRAM array. This observation strongly aligns with our hypothesis, indicating that aging introduces more dissimilarities in the percentage of 1s during the power-up state of the SRAM. Further justification for selecting a 64×64 block size for this investigation will be explained in the subsequent discussion.

A crucial consideration in the data analysis for our proposed approach involves the judicious selection of an optimal block size. To ensure a comprehensive examination of the percentage of 1s across diverse regions within the power-up state, we vary the block size, encompassing a range from smaller dimensions, such as 12×12, to larger configurations, extending up to 128×128. This systematic exploration allows us to scrutinize the impact of varying block sizes on the distribution of 1s in the power-up state, providing us with insights to choose the most suitable block dimension for our study.

Figure 3.6 shows the variation in σ of $p1s$ across different block sizes for both new and aged chips. The chips are denoted as C1, C2, and so forth. We observe the inverse relationship between σ values and block size, with this effect more pronounced in aged chips. For block sizes smaller than 32×32, a sharp reduction trend was noted across all chips, regardless of their age, attributed to the limited impact of the averaging effect on $p1s$ values within smaller blocks. The graph for a block size of 32×32 exhibited a slight increase, explained by the clustering nature of 1s and 0s in the aging data's power-up state for this specific block size. Notably, a significant difference in σ values was observed between aged chips (top curves) and new chips (bottom curves) for larger block sizes. For instance, at a block size of 42×42, the σ of new chips ranged from approximately 1.3 to 1.9, while for old chips, it ranged from about 3.0 to 3.8. In subsequent block sizes, the spread for new chips remained relatively stable, around or below 1.0, whereas σ values for old chips remained significantly larger and varied widely across different block sizes. However, for very large block sizes, such as 128×128, the averaging effect of $p1s$ within the blocks balances out concentrations of 1s and 0s on the power-up states, lowering the standard distribution and rendering it unsuitable for our study. Based on the above analysis, it is evident that in new chips, there is a consistent and uniform behavior across all blocks. However, in the case of old chips, a distinctive pattern emerges, indicating a gradual decrease in σ as the block size increases. This insightful observation forms the crux of our proposed self-referencing-based recycled IC detection approach. The systematic reduction in σ with larger block sizes in old chips serves as a unique signature, offering a robust foundation for developing a reliable method to detect recycled integrated circuits.

Table 3.1: Δ for different chips.

Chips	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**
New Δ	0.63	0.64	1.00	0.42	0.97	0.81
7 days aged Δ	2.92	2.79	2.57	2.72	3.25	2.44

Table 3.1 presents the similarity results for new and aged chips. Due to the pronounced change in slope observed up to the block sizes of 42×42, we recommend utilizing the Δ calculated from block sizes ranging between 51×51 to 128×128 for the purpose of recycled IC detection. We define the similarity index as the slope of these curves, expressed by the following equation.

$$\Delta = \sigma_{51\times51} - \sigma_{128\times128} \tag{3.1}$$

The columns specify the chips (C1 to C6), and rows indicate the Δ values for the new and 7-day aged chips. The values within the table cells represent the calculated the differential σ for each respective chip under the specified conditions. We note a consistent Δ pattern with values consistently below 1.00 for new chips. In contrast, aged chips exhibit Δ values exceeding 2.40, resulting in a distinct separation. This observed distinction forms the foundational basis for recycled IC detection.

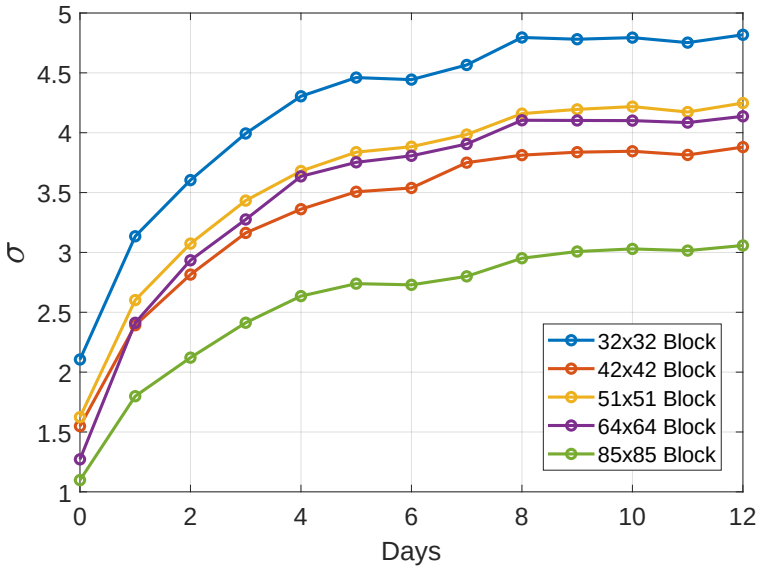

Figure 3.7: The change of σ of an SRAM Chip over a period of 12 days.

We performed two additional experiments to demonstrate that more aging increases the spread of $p1s$. To demonstrate the effect of aging on the standard deviation of $p1s$, we carried out the same accelerated aging on an SRAM chip for 12 days. The obtained values

of σ were plotted for five different block sizes, as depicted in Figure 3.7. Before starting the aging process, a hundred power-up states were recorded in the initial state, yielding σ values in the range of 0.8 to 2.2 for different block sizes. As the chip undergoes 6 hours of daily aging, the σ for all the blocks exhibited an upward trend. This experiment provides insight into the nature of the relationship between σ and aging duration: as the chip undergoes aging, the rise of σ shows the well-known logarithmic rise.

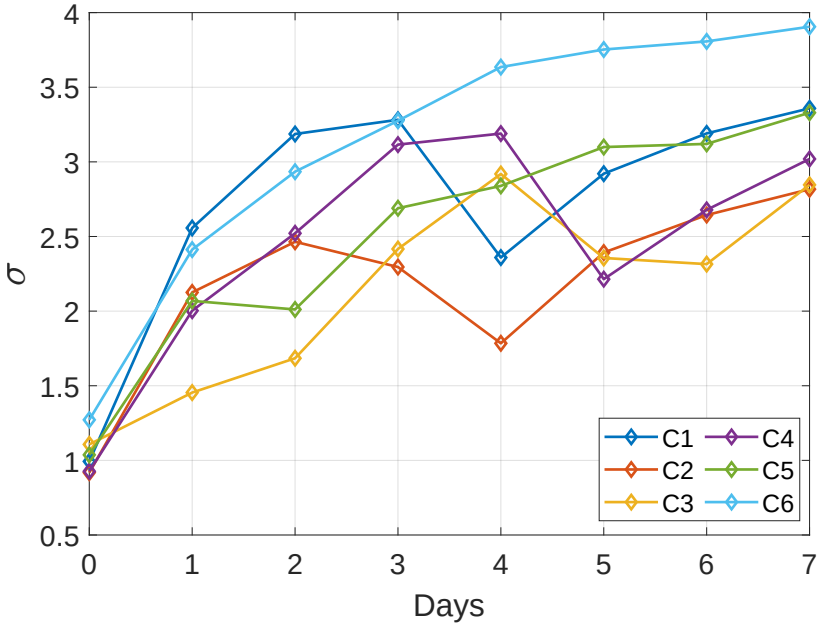

Figure 3.8: The change of σ for six SRAM chips over a period of 7 days.

In conclusion, our findings reveal consistent behavior across all chips studied. The selection of a 64×64 block size was based on the distinct differentiation observed in the corresponding σ data between new and aged chips, as detailed earlier. To validate the effectiveness, we subjected six chips to a seven-day aging process. Figure 3.8 illustrates the variation in the standard deviation of the $p1s$ of the chips throughout the aging process. Notably, a decrease in σ was observed on Day 4 for chips C1 and C2 as they underwent a reverse aging process. Similarly, chips C3 and C4 experienced reverse aging on Day 5 and Day 6, leading to a corresponding decrease in σ. In contrast, chips C5 and C6 underwent continuous aging throughout the week. The observed σ values followed the anticipated

pattern of increase and decrease, aligning with our aging and reverse-aging procedures. While reverse aging effectively diminishes the variation, it never fully restores the value back to its original new state.

Chapter 4

Blockchain-Enabled Whitelisting Mechanisms for Securing 3D ICs

This section presents a novel conceptual approach aimed at mitigating the inherent risks posed by the lack of trust among entities within the supply chain [31]. We propose to secure against malicious IPs in 2.5D/3D ICs at runtime by constraining the communication abilities of chiplets. This is accomplished using a whitelisting technique inspired by security measures deployed in traditional networks. We also propose to use a logger to capture any communication rule violation that occurs during die-to-die communications across different chiplets. The logger state can be further uploaded to an immutable blockchain ledger for forensics purposes if an attack is identified. We begin by highlighting the threat potential posed by malicious IP in a heterogeneously integrated circuit.

As shown in Section 2.2, Figure 2.2 illustrates an example system-in-package (SiP) architecture, integrating multiple dies into a single package by combining 2.5D and 3D stacked dies into one unit. Within such configurations, a single chip can comprise multiple dies varying significantly in process nodes, functionality, and manufacturing origins. Consequently, chiplets manufactured offshore present significant security concerns, as they may be susceptible to malicious modifications, known as hardware Trojans, potentially without the knowledge of the original IP holder. Specifically, in Figure 2.2, chiplets 1, 4, and 6 are considered trustworthy, whereas chiplets 2, 3, and 5 are presumed untrusted. Unrestricted data transmission involving these untrusted chiplets introduces risk, as a hardware Trojan embedded in one die could compromise sensitive data or disrupt services throughout the permitted transmission network. Without robust runtime attack detection and protective mechanisms, malicious dies could have unchecked access to critical systems and sensitive data, underscoring the need for communication security within these integrated environments.

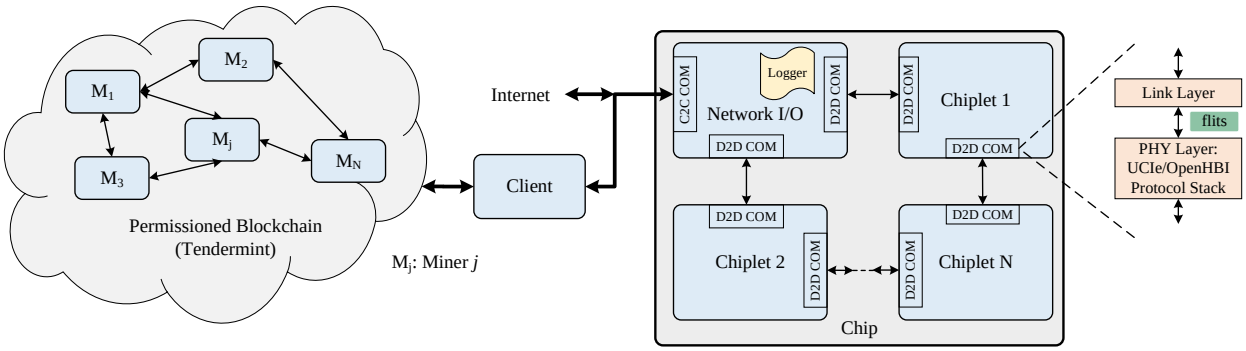

Figure 4.1: An abstract view of the proposed solution for securing 3D ICs.

Given the relatively recent emergence of 2.5D and 3D IC technologies, prior research addressing runtime security threats remains limited. Existing literature proposes methods for protecting these advanced ICs against hardware Trojans [27, 28], yet such approaches often do not adequately address threats encountered once chips are operational in the field. Despite extensive research efforts spanning several decades, the effective detection of hardware Trojans remains a challenging problem [9, 29]. In contrast to traditional Trojan detection methods, our proposed strategy proactively addresses these security risks through the deployment of a whitelisting approach.

4.1 Proposed Architecture

The core of our proposed architecture revolves around authorizing and controlling chiplet communication. Figure 4.1 depicts our overall proposed architecture for securing 3D ICs. Initially, we advocate for the implementation of an on-chip firewall to regulate data traffic flowing into and out of the chip. Firewalls represent a foundational element in network security, effectively managing communications between separate networks by blocking unauthorized data transfers. Such firewalls employ rule-based configurations, primarily via two fundamental filtering methods: blacklists and whitelists. Blacklists typically consist of addresses, domains, or protocols identified as harmful or untrustworthy, thus preventing associated data from traversing the firewall [59]. Conversely, whitelists operate under the restrictive principle of "deny all except," allowing data transmission solely from approved,

trusted sources to their specified destinations, and automatically blocking all other transmissions. Employing these methods in conjunction provides layered protection against various threats, ranging from malicious attacks to undesired communications [60].

The physical layer (PHY) of the router typically resides within the chiplet and employs a protocol stack—such as the widely adopted UCIe [21]—to handle data flits from the link layer. The initial flit in a die-to-die (D2D) communication, termed the header flit, contains essential routing information, including source and destination addresses, guiding the subsequent flits.

Our architecture integrates the whitelist directly within the network I/O module along the D2D communication pathway. The whitelist explicitly defines permitted source-destination pairs, allowing communication only from trusted chiplets to authorized destinations. To monitor compliance and detect violations, an on-chip logger records instances when transmissions fail whitelist verification—specifically, transmissions involving unauthorized source-destination pairs. This logger must have sufficient memory capacity to record multiple events, employing either a circular buffer approach (overwriting the oldest entries) or a selective drop method when exceeding capacity.

Each whitelist entry uniquely specifies an allowable source chiplet address paired with a designated destination address, thereby ensuring communications remain explicitly verified and secure. However, this introduces a trade-off: extensive whitelists increase vulnerability by expanding potential attack vectors, while excessively restrictive lists risk blocking legitimate operations. Hence, whitelist management must reside within a centralized security module embedded within the chiplet infrastructure, updated regularly based on ongoing security analyses, operational needs, and hardware configuration adjustments. These updates may be automated through scripts reacting to security advisories or manually overseen by network administrators.

To enhance logging security and traceability, we propose incorporating blockchain technology to periodically capture the logger's state. Leveraging blockchain infrastructure can

also support device provenance tracking, encompassing design, manufacturing, and distribution phases [26, 61]. A permissioned blockchain architecture, such as those offered by Tendermint [62] or Hyperledger Fabric [63], would be particularly suitable. Alternatively, privacy-preserving smart contracts on robust public blockchains could also be explored. Given that logger-generated data volumes are expected to remain within manageable limits, the blockchain system can effectively handle transactions, with optional data compression or selective admission control strategies if necessary.

A lightweight blockchain client operating on the host system would securely transmit logged data across the Internet—potentially via encrypted virtual private networks—to blockchain validator nodes. Each transaction incorporates a standardized message format containing a global timestamp, facilitating correlation and forensic analysis of logged communications across multiple chiplets. Events captured by loggers can be aggregated into periodic batch transactions for efficient blockchain recording. This blockchain-based infrastructure emphasizes modularity, scalability, and adaptability to future technological advancements and cryptographic standards, ensuring long-term security and operational viability. Although initial implementation requires investment in software, hardware, and human resources, the resulting security enhancements and counterfeit prevention benefits substantially outweigh the associated costs, ultimately proving cost-effective and sustainable for secure on-chip communications.

The private key associated with each chip derives from an SRAM-based physically unclonable function (PUF). Upon powering up the chip, inherent manufacturing variations within SRAM cells generate a unique and consistent power-up pattern, effectively serving as a distinctive hardware fingerprint. Immediately following initialization, this SRAM-generated fingerprint is securely communicated to the desktop or server hosting the blockchain client. This approach ensures the authenticity and integrity of transactions through reliable and tamper-proof cryptographic keys uniquely associated with each individual chip.

Ensuring secure operation involves digitally signing logger-generated messages using the chip's private key. To maintain key confidentiality, the secret key must reside exclusively within a tamper-proof memory accessible solely to the trusted network I/O module, safeguarded against unauthorized external access. Furthermore, storing the key in non-volatile memory ensures persistent availability and seamless retrieval, further reinforcing system integrity. Digitally signed logger entries are then transmitted securely over the internet to designated blockchain servers, completing the robust security framework integral to our proposed architecture.

Ensuring that the messages from the on-chip logger reach the blockchain will require careful hardware design. We propose incorporating a dedicated network interface card of the overall architecture. This specialized component plays a dual role, serving as a hardware blockchain client while effectively managing the digital signature process. Operating seamlessly as a liaison between the on-chip logger and the blockchain servers, this dedicated card ensures a swift and secure exchange of digitally signed messages over the network. Moreover, this dedicated network interface card can effectively handle the reception of messages from trusted loggers dispersed across various chips within the system. This strategic feature not only streamlines communication but also enhances the system's ability to gather data from multiple sources within the network. By providing this dual functionality, this card acts as a centralized hub for managing blockchain interactions and facilitating efficient communication between the on-chip loggers and the broader blockchain network. One can also use a local system (such as a laptop or desktop) that runs either a server node on the blockchain or has a secure client that can communicate with one or more server nodes on the blockchain. The data from the logger will be read by software running in a trusted manner and used to generate a message that is digitally signed using a secret key and submitted to the blockchain from the local node.

The overall whitelisting and validation process can be summarized as follows:

• *Step 1: Violation Occurance:* A violation occurs within a network inside of a chip consisting of many chiplets as shown in Figure 4.1. Specifically, when a chiplet i tries to transmit a packet beyond the chip boundaries, utilizing an invalid source-destination pair, the system detects a breach. The communication protocol stipulates that only whitelisted chiplets are authorized to engage in external communications. However, chiplet i lacks such permissions, possibly attributable to factors such as being manufactured in an untrusted environment. Consequently, this attempt is deemed a violation of the established network policies.

• *Step 2: Violation Logging:* In the event of a violation, it becomes imperative to capture the occurrence, even if the chiplet's attempt to communicate externally proves unsuccessful, for subsequent forensic analysis. Our proposed methodology involves recording crucial details, including the chiplet ID, destination address, and timestamp associated with the violation. Subsequently, this information is digitally signed using the chip's secret key, and the resulting signature is securely stored alongside the aforementioned data. Note that access to the logger state is restricted solely to authorized entities, ensuring confidentiality. Moreover, stringent controls are imposed on updates to prevent any malicious attempt to delete or manipulate records of the violation by potential adversaries.

• *Step 3: Logger State Collection:* The desktop/server orchestrating the blockchain client maintains regular communication intervals with a chip, executing periodic exchanges (e.g., every ten or fifteen minutes). A compact buffer is incorporated within the chip to store the logger state temporarily. In the event that the buffer reaches full capacity, a proactive mechanism is implemented. The chip autonomously triggers a request to the server, prompting the upload of the logger state.

• *Step 4: Overall Logger States Collection:* Despite the inherent diversity in resources and functionality among the various chips within the network—considering complex systems like smart grids—it is crucial to highlight that they share a common infrastructure. This infrastructure comprises multiple chiplets and a logger, as depicted in Figure 4.1, facilitating seamless chip-to-chip communications. All loggers in the network - every logger on every

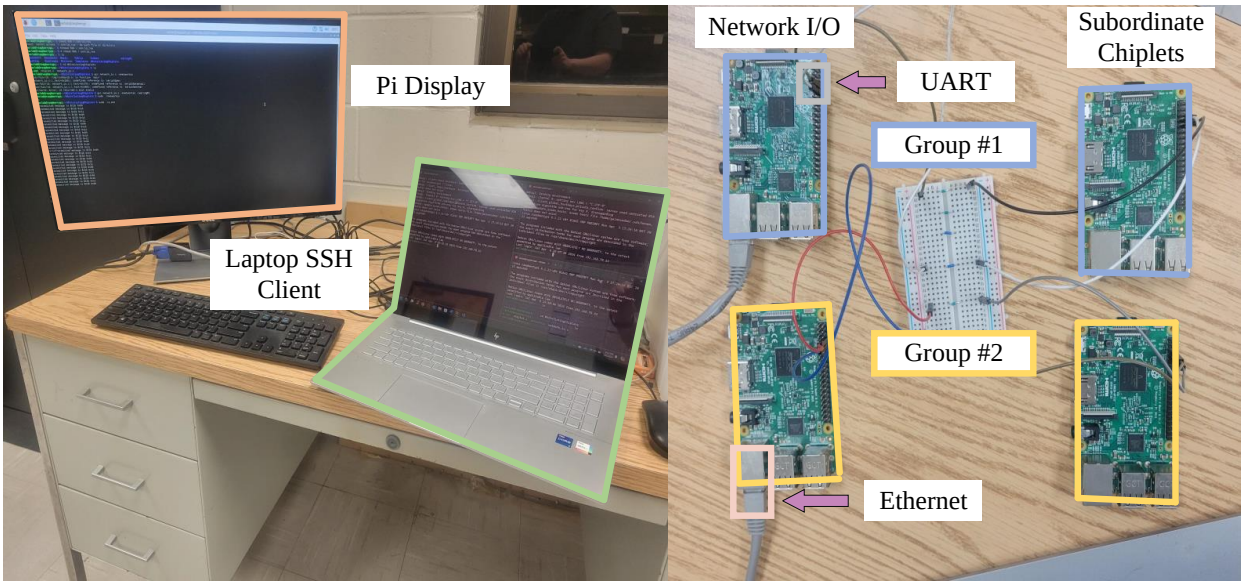

Figure 4.2: Experimental Setup for implementing whitelisting, where Raspberry Pis are modeled as chiplets.

chip - will record their states in a similar fashion at the time interval mentioned in *Step 3*. Note that all logger entries are digitally signed.

• *Step 5: Blockchain Entry:* The desktop or server hosting the blockchain client aggregates all the digitally signed transaction logs it has acquired from different chips. This consolidated set of logs is then appended to the blockchain, ensuring a tamper-proof record of the transactions. Once they are in the blockchain ledger, this data becomes a valuable resource for future forensics, enabling the retrospective analysis and verification of communication events and potential security breaches within the system.

4.2 Implementation Exhibition

To demonstrate the effectiveness of our proposed approach, we have implemented the whitelisting strategy on a network of 2 groups of Raspberry Pi 3 devices acting as chiplets. The Pi 3s comprising one SiP equivalent are connected physically by UART, and the Pi 3s acting as Network I/O chiplets are additionally connected outwardly by Ethernet, shown

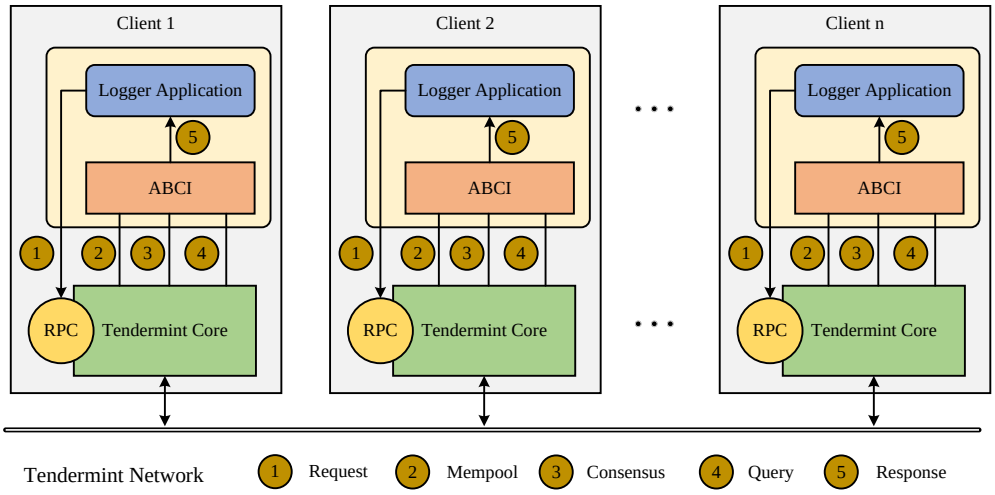

Figure 4.3: Tendermint implementation for logging runtime violations.

in Figure 4.2. Communications between chiplet models are conducted by in-house C programs. We opted to utilize a UART-based communication environment. UART facilitates straightforward serial communication, which is ideal for debugging and iterative testing without the overhead associated with setting up a UCIe environment. While UCIe is quickly becoming a widely adopted chiplet communication interface, offering efficiency, scalability, multi-protocol support (PCIe, CXL), and standardization for multi-chip architectures [21], leveraging UART allowed us to focus on prototyping and validation, ensuring a successful proof-of-concept.

The communication protocol among Pis operates as follows: messages transmitted from chiplets undergo deserialization and conversion into flattened messages. Subsequently, the messages undergo validation, where the header flits, containing the origin and destination information, are compared against a predefined whitelist. This whitelist comprises paired source and destination addresses, specifying allowed communications. If a message satisfies the whitelist criteria, it proceeds to the designated Network I/O interface for Ethernet. There, it undergoes reserialization before transmission over the wired Ethernet connection.

Figure 4.3 demonstrates the overall blockchain framework that enables the detection of run-time attacks originating from untrusted chiplets. Tendermint is a blockchain software for securely and consistently replicating an application on multiple machines [62]. At its

core, it is a Byzantine Fault Tolerant (BFT) state machine replication, and it can be used to create and manage a blockchain network. It is designed for flexibility and modularity by its Tendermint Core consensus mechanism and Application Blockchain Interface (ABCI). The ABCI facilitates communication between the blockchain and the application's operational logic. This modularity is crucial in the context of global adoption, as it allows the application to process specific transactions related to security events without interfering with the underlying blockchain dynamics. Clients play a pivotal role in the blockchain framework. They are responsible for several key functions and act as a bridge between the on-chip loggers situated within a chip's network I/O module and the blockchain. In particular, they collect, verify, and package the logged data from the chiplets, which include detailed records of communication activities and security policy violations. Once this data is prepared, the client then securely transmits it to the blockchain network for validation and recording. Clients ensure that the data adheres to the expected format and contains valid signatures before it's sent to the Tendermint network.

The secure recording process of logs begins with the collection of data from on-chip loggers positioned within the network I/O module of a chip. This logger is tasked with collecting logs from various chiplets integrated within the chip. It records instances of unauthorized communication attempts or breaches against the established whitelisting policy. The logger captures essential data such as the source and destination of the attempted communication, the timestamp of the event, and the type of policy violation. Each chip contributes to this security measure by appending its digital signature to the log data it generates, ensuring authenticity and integrity. This signed log data is securely encrypted by the logger and then transmitted to the client. The client, in turn, is responsible for relaying this data to the Tendermint blockchain network, where it undergoes further processing and is finally integrated into the blockchain ledger.

The integration of Tendermint Core with the application logic responsible for processing and verifying data from the on-chip logger is facilitated through three critical components:

consensus, mempool, and query. The mempool in Tendermint acts as a temporary buffer for transactions before they are processed by the consensus engine. Our security application allows the system to efficiently manage and prioritize the influx of valid logging data from the on-chip logger, ensuring that the blockchain can handle high volumes of violation data without any bottleneck scenario. Next, Tendermint's consensus mechanism ensures that all transactions representing logged security events from the on-chip logger are validated and agreed upon by all participating validator nodes in the network before being committed to the blockchain ledger. This BFT consensus mechanism is pivotal for maintaining the integrity and tamper-resistance of the ledger, ensuring that only verified events are recorded. Through ABCI, the application logic can query the blockchain state, enabling real-time monitoring and forensic analysis of the logged security events. This capability is crucial for identifying patterns of unauthorized communication attempts across chiplets, facilitating timely interventions, and enhancing the overall security of 2.5D/3D ICs. The Tendermint Core provides the foundational blockchain functionality, including the consensus engine, networking, and blockchain state management. Our security system leverages this Tendermint core to ensure a secure, consistent, and tamper-evident ledger of chiplet communication events that violate the whitelist policy.

Transactions originating from the on-chip loggers are encoded into a JSON format, providing a standardized and suitable medium for encapsulating the metadata associated with each blockchain transaction. The JSON-formatted data is then converted into a byte representation and subsequently encoded as a hexadecimal string. This hexadecimal encoding serves as a blockchain-compatible format facilitating the digital signing of the transaction. Digital signatures are appended to the transactions using private keys that are securely managed and stored within the chip infrastructure. Upon successful encoding and digital signing, the transactions are submitted to the blockchain network via an HTTP request to the Tendermint node's Remote Procedure Calls (RPC) interface [64]. This submission leverages the unique transaction format and incorporates the hexadecimal-encoded data as

a parameter in the request URL, ensuring that the transaction is appropriately broadcasted to the network for consensus processing.

Once a transaction is received for processing, it undergoes a decoding step where the hexadecimal-encoded data is converted into its original JSON format before being admitted into the blockchain. This preliminary decoding is essential for validating and processing the transaction through the consensus mechanism. Upon successful validation, the transaction is added to the blockchain, becoming a permanent and immutable record within the ledger. This procedure is critical for the forensic analysis and audit of security events within the 2.5D/3D IC ecosystem. The immutable nature of these records ensures that stakeholders can reliably query and examine historical data, providing invaluable insights into unauthorized communications and potential security breaches. This streamlined process of decoding followed by blockchain integration ensures the utility of the security framework in monitoring and safeguarding communications across multiple chiplets within and between chips.

Chapter 5

Conclusion and Future Research Direction

The growing complexities inherent in modern integrated circuits—particularly those involving heterogeneous integration and three-dimensional integrated circuits (3D ICs)—have significantly heightened concerns surrounding counterfeit components, unauthorized modifications, and the insertion of stealthy hardware Trojans. These advanced architectures, while offering substantial gains in performance, density, and power efficiency, also introduce new avenues for malicious exploitation due to their reliance on multi-source dies, increased interconnect density, and limited physical access for post-fabrication inspection.

In response to these emerging threats, this thesis has introduced a novel self-referencing methodology for counterfeit detection that fundamentally depart from traditional detection mechanisms. By removing the dependency on external characterization databases or golden samples the proposed technique presents a scalable and practical solution for diverse IC platforms, including those fabricated in untrusted or distributed supply chains. This shift toward self-contained authentication not only simplifies the logistics of implementation but also circumvents the limitations posed by the inaccessibility or unavailability of trusted reference data—particularly relevant in defense, aerospace, and legacy systems contexts.

Beyond detection, the research also explores proactive, architecture-level countermeasures aimed at ensuring runtime security in post-deployment phases. Specifically, it outlines strategies for protecting 2.5D and 3D ICs from dynamic threats through mechanisms that incorporate localized monitoring, whitelisting, and in-situ validation of inter-chiplet interactions. These techniques lay the foundation for future secure-by-design frameworks that embed resilience directly into the operational fabric of advanced heterogeneous systems.

Looking ahead, several avenues remain open for further investigation and refinement. One particularly promising direction involves the integration of machine learning (ML) and artificial intelligence (AI) techniques into the self-referencing detection framework. By training models to discern between aging-induced degradation patterns and subtle process-induced variations, ML algorithms could significantly increase the specificity and sensitivity of counterfeit identification systems. This would not only reduce false positives but also extend the applicability of the method across different technology nodes, fabrication processes, and usage profiles.

In parallel, there is a compelling need to develop streamlined and automated runtime security protocols tailored specifically to the unique characteristics of 2.5D and 3D ICs. Future work could explore hardware-software co-design strategies to tightly couple threat detection with response mechanisms, enabling real-time threat mitigation without sacrificing system performance or reliability. Such efforts may include lightweight runtime monitors, reconfigurable security enclaves, or adaptive control logic capable of responding to observed anomalies in inter-die communication patterns.

Furthermore, dynamic threat modeling and mitigation should be prioritized in future research, with an emphasis on developing systems capable of adapting to evolving adversarial tactics. Runtime security, when coupled with active reconfiguration or self-healing capabilities, has the potential to transform modern ICs from static targets into resilient platforms that can withstand and recover from sophisticated attacks.

Finally, given the increasing reliance on globally distributed semiconductor supply chains, future work should also prioritize system-level trust and traceability. Emerging technologies such as blockchain-based provenance tracking, trusted execution environments, and design-for-security methodologies offer complementary tools to enhance transparency and accountability throughout the IC lifecycle. By embedding such strategies into the broader framework of design, fabrication, and deployment, the industry can better guard against systemic vulnerabilities that originate far upstream of the final product.

In conclusion, the methodologies developed in this thesis lay a robust foundation for a new class of security and trust mechanisms suitable for next-generation ICs. Continued research in this direction—encompassing detection, protection, adaptability, and supply chain assurance—will be critical to maintaining the integrity and resilience of electronic systems in an increasingly adversarial and complex technological landscape.

Bibliography

[1] International Labour Office. The Distribution of Value Added Among Firms and Countries: The Case of the ICT Manufacturing Sector. Technical report, International Labour Organization, 2017. Accessed: 2025-04-14.

[2] TSMC (TSM) - Market Capitalization. `https://companiesmarketcap.com/tsmc/marketcap/`, 2025. Accessed: 2025-04-14.

[3] DiMase, Daniel and Collier, Zachary A and Muldavin, Jeremy and Chandy, John A and Davidson, Donald and Doran, Derek and Guin, Ujjwal and Hallman, John and Heebink, Joel and Hall, Ezra and Honorable Alan R. Shaffer. Zero Trust for Hardware Supply Chains: Challenges in Application of Zero Trust Principles to Hardware. *National Defense Industrial Association (NDIA)*, 2021.

[4] Ujjwal Guin, Ziqi Zhou, and Adit Singh. Robust Design-for-Security Architecture for Enabling Trust in IC Manufacturing and Test. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 26(5):818–830, 2018.

[5] Ujjwal Guin, Qihang Shi, Domenic Forte, and Mark M Tehranipoor. FORTIS: A Comprehensive Solution for Establishing Forward Trust for Protecting IPs and ICs. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 21(4):1–20, 2016.

[6] Ujjwal Guin, Ziqi Zhou, and Adit Singh. A Novel Design-for-Security (DFS) Architecture to Prevent Unauthorized IC Overproduction. In *VLSI Test Symposium (VTS)*, pages 1–6, 2017.

[7] Ujjwal Guin, Swarup Bhunia, Domenic Forte, and Mark M Tehranipoor. SMA: A System-Level Mutual Authentication for Protecting Electronic Hardware and Firmware. *IEEE Transactions on Dependable and Secure Computing*, 14(3):265–278, 2016.

[8] Andrew Stern, Dhwani Mehta, Shahin Tajik, Ujjwal Guin, Farimah Farahmandi, and Mark Tehranipoor. SPARTA-COTS: A Laser Probing Approach for Sequential Trojan Detection in COTS Integrated Circuits. In *IEEE Physical Assurance and Inspection of Electronics (PAINE)*, pages 1–6, 2020.

[9] Ayush Jain, Ziqi Zhou, and Ujjwal Guin. Survey of Recent Developments for Hardware Trojan Detection. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2021.

[10] Ujjwal Guin, Ke Huang, Daniel DiMase, John M Carulli, Mohammad Tehranipoor, and Yiorgos Makris. Counterfeit Integrated Circuits: A Rising Threat in the Global Semiconductor Supply Chain. *Proceedings of the IEEE*, 102(8):1207–1228, 2014.

[11] Ujjwal Guin, Daniel DiMase, and Mohammad Tehranipoor. Counterfeit Integrated Circuits: Detection, Avoidance, and the Challenges Ahead. *Journal of Electronic Testing*, 30(1):9–23, 2014.

[12] Mark M Tehranipoor, Ujjwal Guin, and Domenic Forte. *Counterfeit Integrated Circuits: Detection and Avoidance*. Springer International Publishing, 2015.

[13] Committee on Armed Services, House of Representatives. National Defense Authorization Act for Fiscal Year 2011, 2010. United States Congress.

[14] Ujjwal Guin, Daniel DiMase, and Mohammad Tehranipoor. A Comprehensive Framework for Counterfeit Defect Coverage Analysis and Detection Assessment. *Journal of Electronic Testing*, 30(1):25–40, 2014.

[15] Carlo Abesamis and Mark Leblanc. NASA Counterfeit Parts Awareness and Inspection, 2016.

[16] Joshua Hovanes, Yadi Zhong, and Ujjwal Guin. Beware of Discarding Used SRAMs: Information is Stored Permanently. In *IEEE Physical Assurance and Inspection of Electronics (PAINE)*, pages 1–7, 2022.

[17] Nidish Vashistha, Md Latifur Rahman, Md Saad Ul Haque, Azim Uddin, Md Sami Ul Islam Sami, Amit Mazumder Shuo, Paul Calzada, Farimah Farahmandi, Navid Asadizanjani, Fahim Rahman, and Mark Tehranipoor. ToSHI-Towards Secure Heterogeneous Integration: Security Risks, Threat Assessment, and Assurance. *Cryptology ePrint Archive*, 2022.

[18] Taiwan Semiconductor Manufacturing Company. Introducing TSMC 3DFabric: TSMC's Family of 3D Silicon Stacking, Advanced Packaging Technologies and Services. `https://www.tsmc.com/english/news-events/blog-article-20200803`, 2020.

[19] Taiwan Semiconductor Manufacturing Company. TSMC 3DFabric. `https://3dfabric.tsmc.com`.

[20] Samsung Electronics. Samsung Electronics Develops Industry's First 12-Layer 3D-TSV Chip Packaging Technology. `https://semiconductor.samsung.com/news-events/news/samsung-electronics-develops-industrys-first-12-layer-3d-tsv-chip-packaging-technology/`, 2019.

[21] Universal Chiplet Interconnect Express. UCIe. `https://www.uciexpress.org/`.

[22] Open Compute Project. OpenHBI Specification Version 1.0. `https://www.opencompute.org/documents/odsa-openhbi-v1-0-spec-rc-final-1-pdf`, September 2021.

[23] IHS iSuppli. Top 5 Most Counterfeited Parts Represent a $169 Billion Potential Challenge for Global Semiconductor Market. `http://press.ihs.com/press-release/design-supply-chain/top-5-most-counterfeited-parts-represent-169-billion-potential-cha`, 2011.

[24] Aritri P. Saha and Ujjwal Guin. Optimizing Supply Chain Management using Permissioned Blockchains. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–7, 2024.

[25] Sulyab T. Valapu, Aritri P. Saha, Bhaskar Krishnamachari, Vivek Menon, and Ujjwal Guin. Reward-based Blockchain Infrastructure for 3D IC Supply Chain Provenance. In *IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 1–11, 2025.

[26] Yadi Zhong, Amaar Ebrahim, Ujjwal Guin, and Vivek Menon. A Modular Blockchain Framework for Enabling Supply Chain Provenance. In *IEEE Physical Assurance and Inspection of Electronics (PAINE)*, pages 1–7, 2023.

[27] Siroos Madani, Mohammad R Madani, Indira Kalyan Dutta, Yamini Joshi, and Magdy Bayoumi. A Hardware Obfuscation Technique for Manufacturing a Secure 3D IC. In *IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 318–323. IEEE, 2018.

[28] Yang Xie, Chongxi Bao, Caleb Serafy, Tiantao Lu, Ankur Srivastava, and Mark Tehranipoor. Security and Vulnerability Implications of 3D ICs. *IEEE Transactions on Multi-Scale Computing Systems, vol. 2*, pages 108–122, 2016.

[29] Kan Xiao, Domenic Forte, Yier Jin, Ramesh Karri, Swarup Bhunia, and Mohammad Tehranipoor. Hardware Trojans: Lessons Learned after One Decade of Research. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 22(1):1–23, 2016.

[30] Gaines Odom, Zakia Tamanna Tisha, and Ujjwal Guin. A Novel Self-referencing Approach Using Memory Power-up States for Detecting COTS SRAMs. In *2024 IEEE 42nd VLSI Test Symposium (VTS)*, pages 1–7. IEEE, 2024.

[31] Gaines Odom, Hardhik Mohanty, Ujjwal Guin, and Bhaskar Krishnamachari. Blockchain-Enabled Whitelisting Mechanisms for Enhancing Security in 3D ICs. In *Proceedings of the Great Lakes Symposium on VLSI (GLSVLSI)*, pages 1–6, 2024.

[32] Ujjwal Guin, Wendong Wang, Charles Harper, and Adit D Singh. Detecting Recycled SOCs by Exploiting Aging Induced Biases in Memory Cells. In *IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 72–80, 2019.

[33] Dieter K Schroder and Jeff A Babcock. Negative Bias Temperature Instability: Road to Cross in Deep Submicron Silicon Semiconductor Manufacturing. *Journal of Applied Physics*, 94(1):1–18, 2003.

[34] Vijay Reddy, Anand T Krishnan, Andrew Marshall, John Rodriguez, Sreedhar Natarajan, Tim Rost, and Srikanth Krishnan. Impact of negative bias temperature instability on digital circuit reliability. *Microelectronics Reliability*, 2005.

[35] Umeshwarnath Surendanathan, Abishek S. Vellankanni, Aleksandar Milenkovic, Ujjwal Guin, and Biswajit Ray. Ionizing Radiation-Induced Data Imprinting Effects in SRAM Arrays. In *IEEE Transactions on Nuclear Science*, pages 1–7, 2025.

[36] Dieter K Schroder. Negative Bias Temperature Instability: What Do We Understand? *Microelectronics Reliability*, pages 841–852, 2007.

[37] Wendong Wang, Adit D Singh, and Ujjwal Guin. A Systematic Bit Selection Method for Robust SRAM PUFs. *Journal of Electronic Testing*, pages 1–12, 2022.

[38] Wendong Wang, Ujjwal Guin, and Adit Singh. Aging-Resilient SRAM-based True Random Number Generator for Lightweight Devices. *Journal of Electronic Testing*, 36:301–311, 2020.

[39] Jorge Guajardo, Sandeep S Kumar, Geert-Jan Schrijen, and Pim Tuyls. FPGA Intrinsic PUFs and Their Use for IP Protection. In *International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, pages 63–80. Springer, 2007.

[40] Daniel E Holcomb, Wayne P Burleson, and Kevin Fu. Power-up SRAM State as an Identifying Fingerprint and Source of True Random Numbers. *IEEE Transactions on Computers*, 58(9):1198–1210, 2009.

[41] Vincent Van der Leest, Erik Van der Sluis, Geert-Jan Schrijen, Pim Tuyls, and Helena Handschuh. Efficient Implementation of True Random Number Generator Based on SRAM PUFs. In *Cryptography and Security: From Theory to Applications*, pages 300–318. Springer, 2012.

[42] Aydin Aysu, Ege Gulcan, Daisuke Moriyama, Patrick Schaumont, and Moti Yung. End-to-End Design of a PUF-Based Privacy Preserving Authentication Protocol. In *International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, pages 556–576. Springer, 2015.

[43] Sanu K Mathew, David Johnston, Sudhir Satpathy, Vikram Suresh, Paul Newman, Mark A Anders, Himanshu Kaul, Amit Agarwal, Steven K Hsu, Gregory Chen, et al. μ RNG: A 300–950 mV, 323 Gbps/W All-Digital Full-Entropy True Random Number Generator in 14 nm FinFET CMOS. *IEEE Journal of Solid-State Circuits*, 51(7):1695–1704, 2016.

[44] Roel Maes. *Physically Unclonable Functions: Constructions, Properties and Applications*. Springer Science & Business Media, 2013.

[45] Kan Xiao, Md Tauhidur Rahman, Domenic Forte, Yu Huang, Mei Su, and Mark M. Tehranipoor. Bit Selection Algorithm Suitable for High-Volume Production of SRAM-PUF. In *2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pages 101–106. IEEE, 2014.

[46] Md Jubayer al Mahmod and Ujjwal Guin. A Robust, Low-Cost and Secure Authentication Scheme for IoT Applications. *Cryptography*, 4(1):8, 2020.

[47] Yadi Zhong, Joshua Hovanes, and Ujjwal Guin. On-Demand Device Authentication using Zero-Knowledge Proofs for Smart Systems. In *Proceedings of the Great Lakes Symposium on VLSI (GLSVLSI)*, pages 1–6, 2023.

[48] Joshua Hovanes, Yadi Zhong, and Ujjwal Guin. A Novel IoT Device Authentication Scheme Using Zero-Knowledge Proofs. In *GOMACTech Conference*, 2023. Presented at GOMACTech 2023.

[49] Blaise Gassend, Dwaine Clarke, Marten Van Dijk, and Srinivas Devadas. Silicon Physical Random Functions. In *ACM Conference on Computer and Communications Security (CCS)*, pages 148–160. ACM, 2002.

[50] G Edward Suh and Srinivas Devadas. Physical Unclonable Functions for Device Authentication and Secret Key Generation. In *44th ACM/IEEE Design Automation Conference (DAC)*, pages 9–14. IEEE, 2007.

[51] Xin Xin, Jens-Peter Kaps, and Kris Gaj. A Configurable Ring-Oscillator-Based PUF for Xilinx FPGAs. In *IEEE Euromicro Conference on Digital System Design (DSD)*, pages 651–657. IEEE, 2011.

[52] Charles Herder, Meng-Day Yu, Farinaz Koushanfar, and Srinivas Devadas. Physical Unclonable Functions and Applications: A Tutorial. *Proceedings of the IEEE*, 102(8):1126–1141, 2014.

[53] Geert-Jan Schrijen and Vincent Van Der Leest. Comparative Analysis of SRAM Memories Used as PUF Primitives. In *2012 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1319–1324. IEEE, 2012.

[54] Bloomberg. The Big Hack: How China Used a Tiny Chip to Infiltrate U.S. Companies. `https://www.bloomberg.com/news/features/2018-10-04/the-big-hack-how-china-used-a-tiny-chip-to-infiltrate-america-s-top-companies`, 2018.

[55] Bloomberg. The Long Hack: How China Exploited a U.S. Tech Supplier. `https://www.bloomberg.com/features/2021-supermicro/`, 2021.

[56] IHS. Leading up to HR 1540 National Defense Authorization Act for Fiscal Year 2012. `http://www.erai.com/presentations/General%20Session%201/Leading%20up%20to%20HR%201540%20National%20Defense%20Authorization%20Act%20for%20Fiscal%20Year%202012.pdf`, April 2012. Accessed: 2025-04-14.

[57] Bharadwaj S Amrutur and Mark A Horowitz. A Replica Technique for Wordline and Sense Control in Low-Power SRAM's. *IEEE Journal of Solid-State Circuits*, 33(8):1208–1219, 1998.

[58] Microchip 23A640/23K640: 64K SPI Bus Low-Power Serial SRAM. `http://ww1.microchip.com/downloads/en/DeviceDoc/22126E.pdf`.

[59] William R Cheswick, Steven M Bellovin, and Aviel D Rubin. *Firewalls and Internet Security: Repelling the Wily Hacker*. Addison-Wesley Professional, 2003.

[60] William Stallings and Lawrie Brown. *Computer Security: Principles and Practice*. Pearson, 2017.

[61] Pinchen Cui, Julie Dixon, Ujjwal Guin, and Daniel DiMase. A Blockchain-Based Framework for Supply Chain Provenance. *IEEE Access*, 7:157113–157125, 2019.

[62] Ethan Buchman. *Tendermint: Byzantine Fault Tolerance in the Age of Blockchains*. PhD thesis, University of Guelph, 2016.

[63] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, et al. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. In *Proceedings of the Thirteenth EuroSys Conference*, pages 1–15, 2018.

[64] Daniel Cason, Enrique Fynn, Nenad Milosevic, Zarko Milosevic, Ethan Buchman, and Fernando Pedone. The Design, Architecture and Performance of the Tendermint Blockchain Network. In *2021 40th International Symposium on Reliable Distributed Systems (SRDS)*, pages 23–33. IEEE, 2021.