Research paper

Hardware Security in Digital Integrated Circuits A Review of Threats, Attacks, and Defense Mechanisms

Ashish Kumar1, Rita Devi1*

[1]Department of Electronics and Communication Engineering Jawaharlal Nehru Government Engineering College Sundernagar, Himachal Pradesh, India

[1]Corresponding author. E-mail: ritadevi@gmail.com

Abstract
Hardware security has emerged as a major concern in modern digital integrated circuits due to globalization of semiconductor manufacturing, increasing complexity of IC design, and widespread deployment of IoT devices. Modern hardware systems are vulnerable to attacks such as hardware Trojans, side-channel attacks, reverse engineering, counterfeit IC insertion, and intellectual property piracy. Researchers have proposed several defense mechanisms including logic locking, physically unclonable functions (PUFs), secure boot architectures, hardware obfuscation, and side-channel resistant design techniques. This review paper discusses major hardware security threats, analyzes recent research developments from 2016 onward, compares existing protection methods, and highlights future challenges in secure digital IC design.

Keywords: Hardware Security, Digital ICs, Hardware Trojans, Side-Channel Attacks, Logic Locking, PUFs, Semiconductor Security

How to cite this article: Kumar A, Devi R. Hardware Security in Digital Integrated Circuits: A Review of Threats, Attacks, and Defense Mechanisms. Int J Drug Deliv Technol. 2026;16(53s): 269-276. DOI: 10.25258/ijddt.16.53s.30

1. Introduction

1.1. The Rise of the Fabless Semiconductor Model

Today, digital **Integrated Circuits (ICs)** are the "brains" behind almost everything we use, from smartphones and autonomous cars to medical pacemakers and the **Internet of Things (IoT)**. As these chips become more advanced, manufacturing them becomes incredibly expensive. To manage these costs, the semiconductor industry has largely shifted to a **fabless model**.

In this globalized supply chain, a design house (such as Apple or AMD) creates the circuit design and hardware description code, while the actual physical manufacturing is outsourced to third-party foundries located across the world.

1.2. The Problem: A "Zero-Trust" Environment

Although the global supply chain reduces manufacturing costs, it also creates major security vulnerabilities. During fabrication and assembly, the complete IC layout becomes exposed to third-party facilities.

Therefore, modern semiconductor systems must operate under a **zero-trust assumption**, where foundries, distributors, or even internal employees may potentially act as adversaries.

Historically, cyber security focused mainly on software-level protection using mechanisms such as firewalls and antivirus software. However, if the underlying hardware itself is compromised, software-level security can easily be bypassed.

1.3. The Goals of Hardware Security

Hardware security focuses on protecting the physical micro architecture of integrated circuits to ensure that the chip behaves exactly as intended and nothing more. Similar to software security, hardware security is based on the **CIA Triad**:

1. Confidentiality
Ensuring that sensitive information such as passwords, encryption keys, or private user data does not leak through physical properties including power consumption, timing behavior, heat generation, or electromagnetic radiation.

2. Integrity
Guaranteeing that no malicious modifications have been made to the logic gates, wires, registers, or internal functionality of the chip.

3. Availability
Ensuring that the integrated circuit continues to function correctly and cannot be intentionally disabled or disrupted by attackers.

1.4. Emerging Threats in Digital ICs

Because integrated circuits pass through multiple stages and organizations before reaching consumers, they face several hardware-level threats.

One major threat is the insertion of **Hardware Trojans**, which are hidden malicious circuits embedded inside the IC. These Trojans may remain inactive for long periods and activate only when specific trigger conditions occur.

Attackers may also perform **Side-Channel Attacks (SCAs)**, where secret information is extracted by monitoring physical leakages such as:

*Author for Correspondence: ritadevi@gmail.com

- Power consumption
- Timing information
- Electromagnetic emissions
- Thermal characteristics

Additionally, the semiconductor market faces the growing problem of **Counterfeit ICs**. These include illegally overproduced chips, recycled components from electronic waste, or reverse-engineered cloned devices.

1.5. Designing for Trust

To address these threats, researchers and engineers are increasingly adopting **Design-for-Trust (DfT)** methodologies directly during the IC design process.

One important security primitive is the **Physically Unclonable Function (PUF)**, which generates a unique hardware fingerprint for every chip based on microscopic manufacturing variations.

Another widely used approach is **Logic Locking**, where additional logic gates are inserted into the design so that the circuit operates correctly only when the correct secret key is supplied.

However, integrating security features introduces design trade-offs. Engineers must carefully balance security techniques against the chip's **PPA (Power, Performance, and Area)** constraints to ensure that the hardware remains efficient, fast, and energy-conscious.

2. Literature Review

2.1. The Early Focus: Waking Up to Supply Chain Threats (Pre-2020)

While the industry focused on supply chain threats, they were also battling the longstanding threat of Differential Power Analysis (DPA)—a technique introduced by [1]Kocher et al. in 1999 that extracts secret keys simply by monitoring a chip's power consumption during cryptographic operations

Back in the early 2010s, the semiconductor industry was going through a massive shift. To save on the incredibly high costs of building factories, companies started fully embracing the **fabless model**. They would design their chips in-house but send the blueprints overseas for manufacturing. While this made chips much cheaper, researchers quickly realized it opened up a giant security loophole: the physical design of the chip was suddenly exposed to untrusted third parties.

One of the first big wake-up calls for the hardware security community was the **counterfeit crisis**. A landmark 2014 study by [2] Guin *et al.* shed light on how fake chips were quietly sneaking into critical systems, including aerospace, defense, and medical devices. These were not simply cheap knock-offs. Counterfeiters were taking old, used chips from electronic waste, scrubbing off their labels, and selling them as brand new—a practice known as **recycling**.

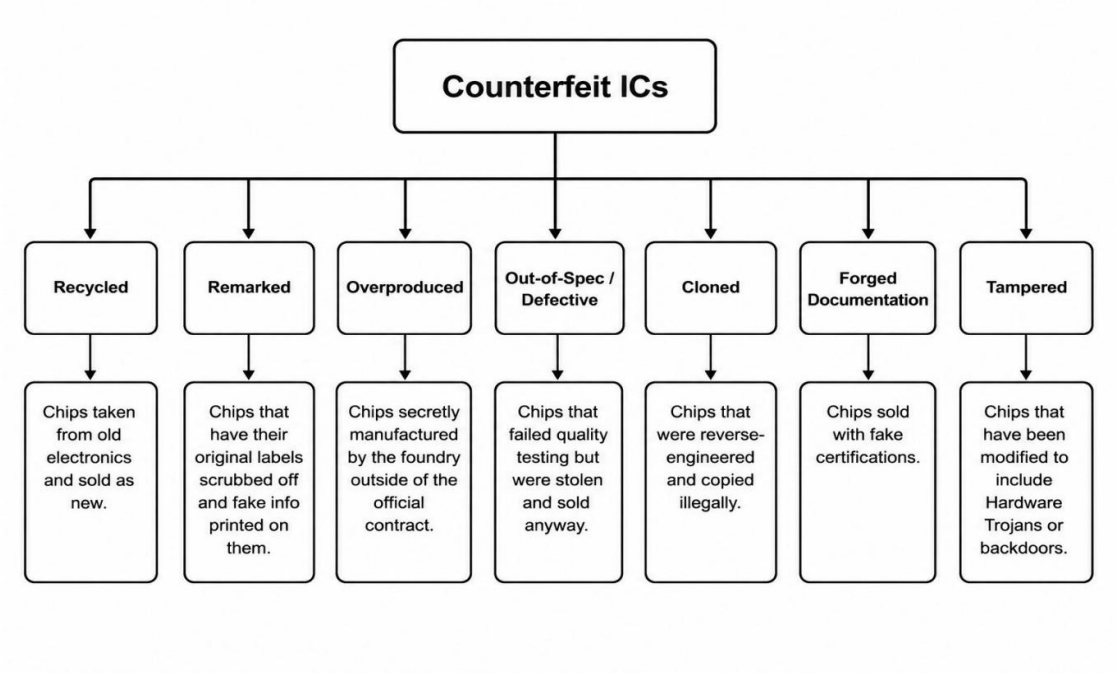

Figure 1: Adapted from the Taxonomy of Counterfeit ICs by Guin et al.

To catch these fakes, early researchers had to get creative. Because used chips slowly wear out over time (a process that degrades the transistors), researchers started measuring the exact time it took for electrical signals to travel through the chip's data paths. If the signal was slightly too slow compared to a fresh chip, it was a strong clue that the chip was recycled.

As chips became more complex, designers started buying premade circuit blocks called **Intellectual Property (IP) cores** from other companies to speed up their work. However, this raised a serious concern: what if a malicious designer hid a **Hardware Trojan** inside that IP core? These Trojans act like hidden time-bombs, waiting for a very specific and rare input to trigger them into leaking secrets or disrupting the system. Because they are designed to remain dormant during normal

testing, they were extremely difficult to detect.

The community's understanding of these threats was heavily shaped by [3]Karri et al. (2010), who developed a comprehensive taxonomy classifying hardware Trojans based on their physical characteristics, activation mechanisms (e.g., always-on, internally triggered), and malicious payloads

Trojan Characteristic	Categories & Examples
Insertion Phase	Specification, Design, Fabrication, Testing, Assembly and Package
Abstraction Level	System level, Register-Transfer Level (RTL), Gate level, Physical level
Activation Mechanism	*Always On* *Triggered Internally:* Time-based (Counters), Condition-based (Specific inputs) *Triggered Externally:* User input, Component output
Malicious Effects	Change functionality, Downgrade performance, Leak in-formation, Denial of Service (DoS)
Physical Location	Processor, Memory, I/O, Power supply, Clock grid

Table 1: Table adapted from the Hardware Trojan Taxonomy presented by Hu et al. [4].

To solve this problem, the research community realized they needed mathematical proof that a chip was safe. In 2016, [4] Hu *et al.* introduced an ingenious method called **Gate-Level Information-Flow Tracking (GLIFT)**. Instead of simply feeding random inputs into the chip and observing the outputs, GLIFT tags sensitive data (such as a secret key) and mathematically tracks exactly how it flows through every single AND, OR, and XOR gate. If sensitive data accidentally flows to an unauthorized output, the tool flags a potential Trojan.

However, there was a major drawback: tracking every individual logic gate in a modern processor was computationally very slow. To overcome this bottleneck, [5] Ardeshiricham *et al.* (2017) extended the concept and introduced **Register Transfer Level Information Flow Tracking (RTLIFT)**. By shifting the analysis to the RTL phase where Verilog or VHDL code is analyzed before synthesis into logic gates—they achieved security verification speeds more than five times faster.

Finally, because engineers knew they could not fully trust external foundries, the pre-2020 era gave rise to several clever **Design-for-Trust** techniques that protected chips from the inside out. One major breakthrough was [6]**Logic Locking**. Designers deliberately scrambled the circuit by inserting additional logic gates. As a result, the manufactured chip would only produce meaningless outputs until the correct secret multi-bit key was entered to unlock the intended functionality.

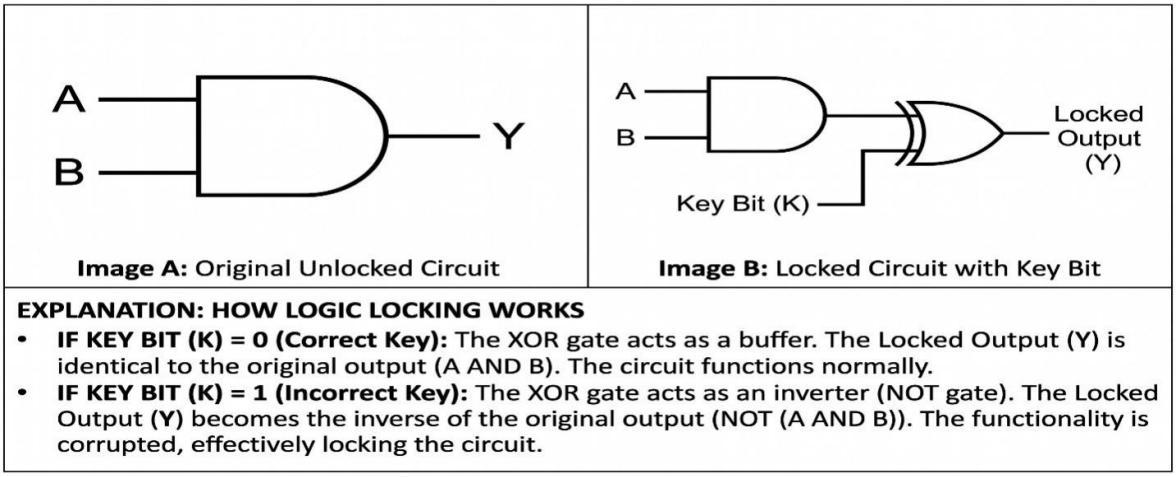

Figure 2: Logic Locking

[7][8]Alongside logic locking, engineers also explored **Physically Unclonable Functions (PUFs)**. They realized that semiconductor manufacturing is never perfectly precise; tiny and uncontrollable microscopic variations always exist in the silicon. PUFs were designed to exploit these natural imperfections to

generate a unique "silicon fingerprint" for every individual chip. Since these variations are entirely random, not even the original manufacturer can create a perfect physical clone of a PUF. This provided a highly effective and foundational method for verifying chip authenticity.

2.2. The Rise of Physical & Side-Channel Threats (2020-2022)

Foundational studies by [9]Ru¨hrmair et al. demonstrated that strong PUFs, particularly Arbiter PUFs, are highly susceptible to ML modeling attacks, as their linear additive delay structures can be accurately predicted given enough challenge-response pairs.

By 2020, the hardware security community had established strong foundational defenses like Logic Locking and Physically Unclonable Functions (PUFs). However, the cat-and-mouse game continued. Instead of trying to break the digital math of the chip, attackers began focusing heavily on its physical properties and started using a powerful new weapon: Machine Learning (ML).

1. **Machine Learning Breaks the "Unclonable" Fingerprint**

PUFs were originally thought to be the ultimate defense because they rely on random, microscopic manufacturing flaws that are impossible to clone. However, re-searchers quickly realized that PUFs especially complex ones with millions of possible challenge-response pairs were highly vulnerable to ML modeling attacks [10].

An attacker could feed a subset of inputs (challenges) into the chip, record the outputs (responses), and train a neural network on that data. Because the physical variations in a chip follow the laws of physics, the ML model could eventually "learn" the chip's unique hardware behavior and accurately predict its future re-sponses, effectively breaking the device's security without ever physically opening it.

To quantify this vulnerability, [11]Ganji et al. introduced 'PUFmeter', a property-testing tool designed to rigorously assess and evaluate the resilience of various PUF architectures against these emerging mathematical modeling attacks.

To counteract these algorithmic threats, defenders began developing machine learning-resistant architectures, such as deceptive multiplexer-based logic locking, to actively confuse structural-exploiting ML models. [12]

2. **Supercharging Side-Channel Attacks**

Machine learning did not just threaten PUFs; it also revolutionized Side-Channel Attacks (SCAs). In a traditional side-channel attack, such as Differential Power Analysis (DPA), an attacker uses statistical analysis to guess a secret cryptographic key by measuring the tiny fluctuations in power consumption of the chip. However, as Dimopoulos et al. demonstrated in 2021, attackers started shifting toward Deep Learning [13].

By profiling a device, a neural network can learn exactly what a chip's power or electromagnetic (EM) emissions look like when processing specific data. Once trained, these ML models require a significantly smaller number of power traces to accurately extract secret keys from the hardware, making SCAs faster and much more dangerous

3. **Defending the Physics: Masking and Hiding**

To fight back against advanced power and EM attacks, hardware designers had to become more creative at the circuit level. Oswald and Howe (2021) outlined two primary hardware countermeasures: *Hiding* and *Masking* [14].

- **Hiding**

Hiding attempts to make the chip's power consumption appear identical regardless of the data being processed. Designers achieve this using specialized logic gate architectures, such as *Dual Rail Precharge* logic, where two complementary wires are used for every signal so that the energy consumption remains nearly constant.

- **Masking**

Masking (or secret sharing) takes a different approach by splitting secret data into random, meaningless pieces called masks before computation. Since the resulting power consumption becomes tied to randomized values instead of the actual secret key, the attacker's ML models are effectively fed with useless noise.

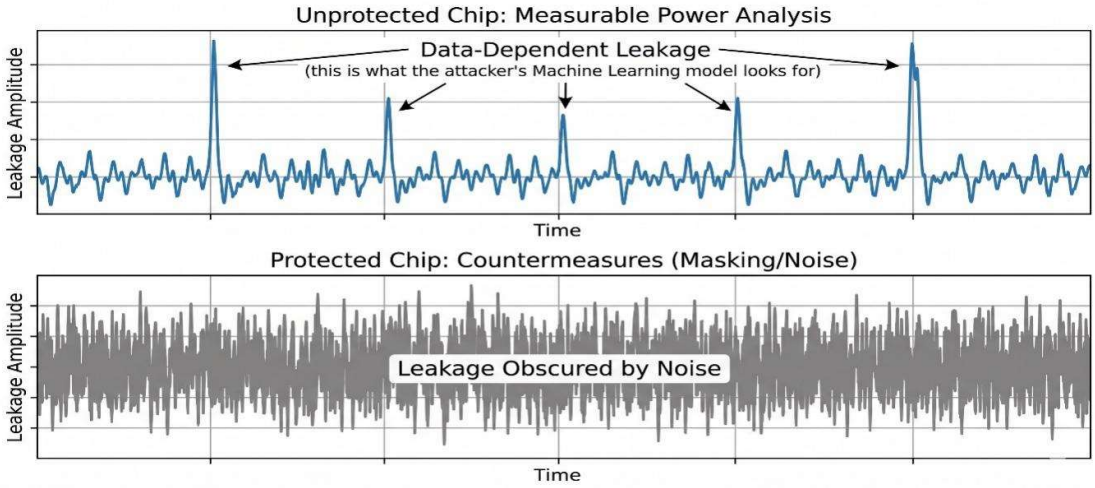

Figure 3: The effect of masking and noise countermeasures on a chip's power consumption traces. The unprotected circuit (top) shows clear data-dependent patterns, while the protected circuit (bottom) obscures the leakage.

However, implementing these countermeasures introduces significant Power, Performance, and Area (PPA) overheads, forcing designers into difficult engineering trade-offs.

4. **Beyond Traditional Silicon:**

As traditional CMOS scaling reaches its physical limits, researchers are exploring emerging technologies like Memristors and Carbon Nanotubes (CNTs) to build inherently secure primitives that are naturally immune to certain physical probing attacks. [15]

Advanced circuit designs are even incorporating self-destructive polymorphic logic that permanently alters or destroys its own functionality if it detects active physical tampering or voltage glitching. [16]

5. **A New Dimension of Vulnerability: 3D Stacking**

During this same period, Moore's Law began slowing down. To continue improving performance, the semiconductor industry started stacking chips vertically, creating 2.5D and 3D Integrated Circuits connected by tiny copper pillars known as Through-Silicon Vias (TSVs).

Although 3D stacking greatly improves performance and integration density, Rao et al. highlighted that it also creates entirely new security challenges [18].

Stacking dies vertically traps heat, causing the temperature of a 3D chip to fluctuate much more than traditional 2D chips. Attackers began designing *Thermal-Activated Hardware Trojans*. Instead of relying on complex digital triggers, these Trojans monitor the chip temperature. When the poorly ventilated 3D chip becomes sufficiently hot, the thermal variations naturally alter transistor delays, activating the Trojan to leak sensitive data or cause system failures.

Furthermore, [17]Dofe et al. highlighted that an untrusted foundry could easily hide malicious circuits inside the Through-Silicon Vias (TSVs) connecting the different layers, effectively creating a denial-of-service or data leakage threat hidden between the planes

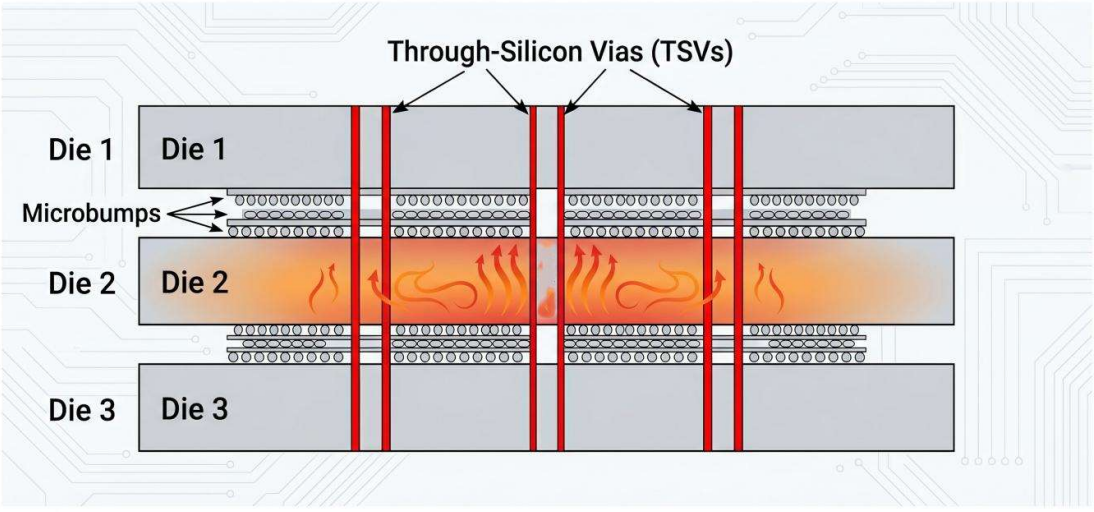

Figure 4: Cross-section of a 3D Integrated Circuit. Heat gets trapped in the middle layers, and attackers can hide Trojans inside the vertical TSV pillars

Defense Technique	How it Works	Protection Level	Implementation Overhead (Area/Power)
Noise Injection	Adds dummy operations to confuse attackers.	Low Moderate	**5% – 20%** (Low cost, good for IoT)
First Order Masking	Splits secret data into random shares.	Moderate	**30% – 80%** (Medium cost)
Dual Rail Logic	Uses two complementary wires to balance power perfectly.	High	**200% – 400%** (Very high cost, used in Smart-cards)

Table 2: PPA Overhead of Side-Channel Countermeasures

2.3. The AI and Quantum Era: A New Frontier (2023-2026)

By 2023, the hardware security landscape experienced two massive tectonic shifts: the explosion of generative Artificial Intelligence (AI), and the impending reality of quantum computers. During this era, researchers realized that the tools used to design and protect chips had to fundamentally change to keep up with these new technologies.

1. **AI Enters the Fab**

Early exploratory studies, such as the 'Chip-chat' framework by [19]Blocklove et al., demonstrated the initial successes and challenges of conversational hardware design, proving that general-purpose LLMs could assist in drafting specifications and generating basic modules Large Language Models Writing Hardware you are likely familiar with using Chat GPT to write Python or C++ code. Recently, researchers began asking a bold question: can AI write the code for physical microchips? The answer is yes, but it is complicated. Hardware Description Languages (HDLs) like Verilog are much stricter than software languages, and early Large Language Models (LLMs) struggled to write functional chips without making errors . To solve this, researchers started fine-tuning models specifically on massive datasets of open-source Verilog code, creating specialized hardware AI assistants like *VeriGen* and *AutoChip* . These tools can take a natural language prompt such as "design a 16-bit processor"—and automatically generate the RTL (Register Transfer Level) code, even interacting with compilers to automatically fix their own syntax bugs. [20]

2. **The Dual-Edged Sword of AI in Security**

While AI makes chip design faster, it also acts as a powerful weapon for both defenders and attackers . On the defensive side, engineers are using LLMs to automatically scan millions of lines of Verilog code to find security vulnerabilities and generate "security assertions" (mathematical rules that prove a chip is safe) . However, on the offensive side, attackers are also using LLMs to design incredibly stealthy Hardware Trojans . Because LLMs can understand the complex context of a chip's design, they can be prompted to hide malicious triggers deep inside seemingly innocent code, such as RISCV processors or cryptographic modules. [20]

For example, the SCAR framework utilizes Graph Neural Networks and LLMs to automatically analyze Register-Transfer Level (RTL) code, boasting over 94% accuracy in localizing side-channel vulnerabilities and generating protective code. [21]

The offensive capabilities of AI have rapidly expanded into mixed-signal designs as well. Recent frameworks like LATENT utilize LLM guided feedback loops to autonomously insert highly stealthy, analog hardware Trojans into complex netlists, showcasing the dual-use nature of generative AI. [22]

To counteract 'AI hallucinations' and build trust in defensive tools, the field is rapidly pivoting toward Explainable AI (XAI). Frameworks like SALTY leverage Graph Neural Networks combined with XAI to not only detect gate-level Trojans but to visually contrast benign and malicious patterns, making the AI's decisions transparent to human engineers [23]

3. **The Quantum Threat and NIST's Response**

While AI was changing how chips were designed, physicists were building quantum computers. Traditional cryptography (like RSA and ECC, which protect everything from your bank account to secure boot processes) relies on math problems that take normal computers millions of years to solve . However, a powerful quantum computer could break these algorithms in hours. In response, the U.S. National Institute of Standards and Technology (NIST) finalized a completely new set of "Post-Quantum Cryptography" (PQC) standards in 2024, such as CRYSTALS Kyber and CRYSTALS Dilithium . These algorithms use complex math, like multi-dimensional lattices, which even quantum computers cannot easily solve. [24]

To protect intellectual property against these massive computational threats, emerging concepts like Quantum Logic Locking (QLL) encode secret keys directly into noncloneable qubits, making reverse engineering fundamentally impossible under the laws of quantum physics [25]

4. **The PPA Nightmare of Post-Quantum Cryptography**

For hardware engineers, the transition to PQC is a massive headache. As always, it comes down to the Golden Triangle: Power, Performance, and Area (PPA). Classical ECC keys are tiny around 64 bytes . The new PQC lattice-based signatures, however, are massive, often requiring 2 to 3 Kilobytes of data . When you try to run these massive PQC algorithms on a tiny, battery-powered IoT microcontroller, the memory footprint and processing delays become huge bottlenecks . To make quantum-proof IoT devices a reality, hardware designers from 2024 onward have been racing to build specialized "PQC hardware accelerators" custom logic gates dedicated strictly to performing lattice math incredibly fast without draining the battery.[24]

Feature	Classic Crypto (RSA, ECC	Post-Quantum Crypto (Kyber, Dilithium)
Security Basis	Prime factorization, discrete logarithms	Multi-dimensional Lattices, hashes
Quantum Resistance	**Vulnerable** (Broken by Shor's algorithm)	**Secure** (Designed to resist quantum attacks)
Hardware Key Sizes	Very Small (e.g., ~64 Bytes for ECC)	**Massive** (e.g., 1–3 Kilobytes)
Implementation	Fast, highly optimized for small chips	Slower, requires high memory overhead
Hardware Support	Widely available built in accelerators	Emerging; requires entirely new chip accelerators

"Table adapted from Kia et al. [24], highlighting the drastic increases in key sizes and memory requirements that hardware engineers must accommodate when transitioning to Post-Quantum Cryptography."

Table 3: The Shift from Classic to Post-Quantum Cryptography in Hardware

5. **The Speculative Execution Threat:**
Modern computing architectures are increasingly threatened by speculative execution vulnerabilities (such as Spectre and Meltdown), where attackers exploit processor optimization routines to leak sensitive data through micro architectural cache side channels [26]Finally, as hardware security matures, future design paradigms are actively integrating AI-assisted security verification with ecological sustainability, striving to design energy-efficient, green secure chips for next-generation 6G ecosystems. [27]

6. Conclusion

We have traveled through the fascinating, invisible battlefield of modern hardware security. The shift to a globalized, "fabless" semiconductor model brought us incredibly advanced and affordable chips, but it also forced us to abandon the idea that hardware is inherently safe. Today, we must operate under a "Zero-Trust" mindset, assuming that vulnerabilities can be introduced at any stage of a chip's journey from the design house to the foundry, and finally into our devices . If there is one major lesson to take away from the literature, it is that hardware security is a perpetual cat-and-mouse game. When engineers invented Physically Unclonable Functions (PUFs) to create unclonable silicon fingerprints, attackers learned to use Machine Learning to predict them. When designers built complex cryptographic accelerators to protect data, adversaries realized they could simply measure the chip's power consumption or temperature to steal the secret keys. Now, as we enter the era of Artificial Intelligence and Quantum Computing, both the weapons and the shields are becoming exponentially more sophisticated. For future VLSI designers, the most important takeaway is that security is never free. Every defensive measure we discussed is permanently tied to the Golden Triangle of chip design: Power, Performance, and Area (PPA) . Implementing quantum-proof mathematical locks, adding deceptive noise to confuse side-channel attackers, or placing sensors to detect laser tampering will inevitably consume more battery, take up more physical silicon space, and potentially slow down the processor . Because no single security feature is perfect, and because we cannot afford to protect every single wire on a chip, the industry is moving toward a "Defense-in-Depth" architecture . Instead of relying on one magic shield, designers layer their defenses: a PUF derives a unique key, which protects a Hardware Root of Trust, which securely boots the system, while the underlying math is protected by side-channel masking. Breaking this chain requires an attacker to defeat multiple independent security mechanisms simultaneously, making it practically impossible . Ultimately, securing digital Integrated Circuits is no longer just an optional add-on or an afterthought; it is a foundational requirement for the safety of our modern world . As you continue your studies in electronics and communication engineering, remember that the code you write in Verilog and the logic gates you place on a chip carry the immense responsibility of keeping our digital future trustworthy, reliable, and secure.

References

[1] Kocher, P., Jaffe, J., Jun, B. (1999). "Differential Power Analysis."

[2] Guin, U., Huang, K., DiMase, D., Carulli, J. M., Tehranipoor, M., Makris, Y. (2014). Title: "Counterfeit Integrated Circuits: A Rising Threat in the Global Semiconductor Supply Chain." Published in: Proceedings of the IEEE, 102(8), 1207-1228.

[3] Karri, R., Rajendran, J., Rosenfeld, K., Tehranipoor, M. (2010). "Trustworthy hardware: Identifying and classifying hardware trojans."

[4] Hu, W., Mao, B., Oberg, J., Kastner, R. (2016). Title: "Detecting Hardware Tro-jans with Gate-Level Information-Flow Tracking." Published in: Computer (IEEE Computer Society), 49(8), 32-40.

[5] Ardeshiricham, A., Hu, W., Marxen, J., Kastner, R.

(2017). Title: ”Register Transfer Level Information Flow Tracking for Provably Secure Hardware Design.” Published in: Proceedings of the Design, Automation , Test in Europe Conference , Exhibition (DATE), 2017, 1691-1696.

[6] Roy, J. A., Koushanfar, F., Markov, I. L. (2008). ”EPIC: Ending Piracy of Integrated Circuits.”

[7] Suh, G. E., Devadas, S. (2007). ”Physical Unclonable Functions for Device Authen-tication and Secret Key Generation.”

[8] Gao, Y., Al-Sarawi, S. F., Abbott, D. (2020). Physical Unclonable Functions. Nature Electronics, 3, 81–91. doi:10.1038/s41928-020-0372-5.

[9] Ru¨hrmair, U., et al. (2010/2013). ”Modeling Attacks on Physical Unclonable Func-tions.”

[10] Gao, Y., Al-Sarawi, S. F., Abbott, D. (2020). Title: ”Physical Unclonable Func-tions.” Published in: Nature Electronics, 3, 81–91.

[11] Ganji, F., Forte, D., Seifert, J.P. (2019). ”PUFmeter: A Property Testing Tool for Assessing the Robustness of Physically Unclonable Functions to Machine Learning Attacks.”

[12] Sisejkovic, D., et al. (2021). ”Deceptive logic locking for hardware integrity protection against machine learning attacks.”

[13] Dimopoulos, C., Fournaris, A. P., Koufopavlou, O. (2021). Title: ”Machine Learning Attacks and Countermeasures on Hardware Binary Edwards Curve Scalar Multipli-ers.” Published in: Journal of Sensor and Actuator Networks, 10(3), 56.

[14] Oswald, E., Howe, J. (2021). Title: ”Side Channels: Attacks, Defences, and Evalua-tion Schemes Part 1 , 2.” Published by: NIST Computer Security Resource Center.

[15] Knechtel, J. (2021). ”Hardware Security for and beyond CMOS Technology.”

[16] Roy, S., et al. (202x) / Forte, D. ”Sense and React: Self-Destructive Polymorphic Mechanism Against Voltage Tampered Active Physical Attacks.”

[17] Dofe, J., Yu, Q., Wang, H., Salman, E. (2016). ”Hardware Security Threats and Potential Countermeasures in Emerging 3-D ICs.”

[18] Rao, V. V., Sasan, A., Savidis, I. (2022). Title: ”Analysis of the Security Vulner-abilities of 2.5-D and 3-D Integrated Circuits.” Published in: Drexel University / University of California, Davis (Conference/Journal Proceedings).

[19] Blocklove, J., Garg, S., Karri, R., Pearce, H. (2023). ”Chip-chat: Challenges and opportunities in conversational hardware design.”

[20] Ghimire, A., et al. (2024). Title: ”Hardware Design and Security Needs Attention: From Survey to Path Forward.” Published in: arXiv preprint.

[21] Srivastava, A., et al. (2024). ”SCAR: Power side-channel analysis at RTL level.”

[22] Chaudhuri, J., et al. (2025). ”LATENT: LLM-Augmented Trojan Insertion and Eval-uation Framework for Analog Netlist Topologies.”

[23] Mahfuz, T., et al. (2025). ”Salty: Explainable artificial intelligence guided structural analysis for hardware trojan detection.”

[24] Kia, A., Storey, A. W., Imtiaz, M. (2026). Title: ”Advanced Hardware Security on Embedded Processors: A 2026 Systematic Review.” Published in: Electronics, 15(5), 1135.

[25] Rao, S. P., et al. (2025). ”Secure and Trustworthy Microelectronics: Vulnerabilities, Solutions, and Trends.”

[26] Mishra, J., Sahay, S. K. (2025). ”Modern Hardware Security: A Review of Attacks and Countermeasures.”

[27] Patil, R. C., Verma, S. (2025). ”Hardware Security and Trusted Integrated Circuit (IC) Design”