Intellectual Property Design with PUF-based Hardware Security

Devi Pradeep Podugu
Anil Neerukonda Institute of Technology and Sciences, India
pradeep.ece06@gmail.com

A .Kamala Kumari
Andhra University College of Engineering, Visakhapatnam, India
kamalakumari99.anala@gmail.com

Srinivas Sabbavarapu
Anil Neerukonda Institute of Technology and Sciences, India
srinivas.sabbavarapu@gmail.com (corresponding author)

Received: 4 April 2024 | Revised: 4 May 2024 and 21 May 2024 | Accepted: 29 May 2024

 DOI: https://doi.org/10.48084/etasr.7413

ABSTRACT

With the advent of networked systems in almost all current applications, security poses a great threat to the design industry. The participation of several people in different design abstract stages in the hierarchical design industry makes the design vulnerable to security threats. To address these security issues, this study used PUFs to create signatures on Intellectual Property (IP) to protect against malicious attacks. The proposed method exhibits significant resilience to ML-based attacks.

Keywords-trojan; hardware security; Intellectual Property (IP)

I. INTRODUCTION

The globalization of the Integrated Circuit (IC) design industry poses a challenging task in counterfeit IC products, as several people, even in several countries, are involved at different levels of design abstracts in the semiconductor supply chain [1]. A significant security risk is unavoidable for ICs in critical applications, such as military, health, and business, where IC parts may be invaded by trojan circuitry by exploiting uncontrollable outsourcing of fabrication [2]. In this context, hardware trojan attacks and hardware security have gained popularity over the last decade, as various hard-to-detect hardware trojans have emerged [3-5]. Several studies have highlighted the difficulty in identifying fraudulent and suspicious manufacturers that exploit more variants of attacks, activation mechanisms, and payloads [6-8]. Some real-world examples have been published [9-11], urging immediate research actions to tackle these problems.

However, technology scaling takes current dense ICs to a higher level of abstraction, where Intellectual Property (IP) cores play a vital role, making the sale and fabrication outsourcing of ICs common in the semiconductor industry, further raising concerns about hardware security [12]. Several attempts have been made in the recent past to ensure hardware security and reliability, as different hardware threats have emerged [13-17]. Also, several security techniques emerge, considering the detection of threats (hardware trojans), in particular, to the ASIC. This study proposes a technique to make the IP core more resilient to hardware attacks by appending Physically Unclonable Functions (PUF) for signature analysis of the IP response. This study uses Vivado and its IP library as a proof of concept.

II. BACKGROUND AND RELATED WORKS

A. Background

In the context of IC design, IP refers to pre-designed and pre-verified building blocks that can be used for faster development of complex digital systems. IP encryption refers to the process of encrypting the IP contained in electronic systems or devices to protect it from being stolen or copied by unauthorized individuals or competitors. There are different methods and techniques for encrypting IPs, and the specific approach used can depend on the type of IP protection and the level of security required. Examples include symmetric-key encryption, asymmetric-key encryption, hardware encryption, and obfuscation. A trojan is a form of malicious block that can be hidden within seemingly harmless files, programs, or attachments, and can be used to compromise the security of a computer system or network. As the field of hardware security continues to evolve, new hardware trojan types are likely to emerge, and new countermeasures need to be developed to address them. Several studies on this topic have been conducted in the last decade. A new class of hardware Trojans was introduced in [12] to convey secret information, through

physical side channels. This approach demonstrated the engineering of power side channels that can cause leakage of information below the threshold limit. The MOLES technique was proposed, exploring the power side channels below the noise power level to convey secret information. Different characterizations were presented in [18], and a gate-level characterization was employed in [19] for trojan detection.

B. Related Works

Architectural techniques aim to improve the likelihood of activating hardware trojans during testing. In [1], fake flip-flops were incorporated into the design to increase trojan activity. This approach used a transition probability threshold to determine where to insert the flip-flops. In [18], a method for employing ring oscillators was introduced to secure all gates in the design. This approach incorporated additional logic that transforms circuit pathways into ring oscillators, and trojans were located using variations in the frequency of the ring oscillators. In [6], voltage inversion was employed at alternate levels of the circuit to enhance the power consumption of an infected circuit. Methodologies that depend on side channels attempt to isolate the trojan's effects on the circuit without turning it on. The major goal is to attempt to very likely identify the existence of a trojan by identifying its design's overabundance on various design parameters, such as power or latency, in contrast to an uninfected circuit. In [20], path delay analysis was used to detect trojans.

In [21], off-chip leakage through side channels was addressed. In [22], consistency-based gate-level characterization was employed to address trojan detection on hardware. In [23], security constraints for wireless ICs were proposed. The study in [10] addressed the security barriers for reconfigurable devices. In [24, 25], state-of-the-art trojan detection techniques were presented for IP cores and different ICs, respectively. In [26], a scalable trojan detection approach was presented. In [27], PUFs were used in FIR filters to ensure the security of the coefficients. In [28], signature-based security was implemented. In [29], the Falcon post-quantum digital signature scheme was used to provide signature-based hardware security.

III. METHODOLOGY

The proposed method consists of two steps. The first step deals with generating the customized IP and the next step ensures security by adding a PUF, as shown in Figure 1.

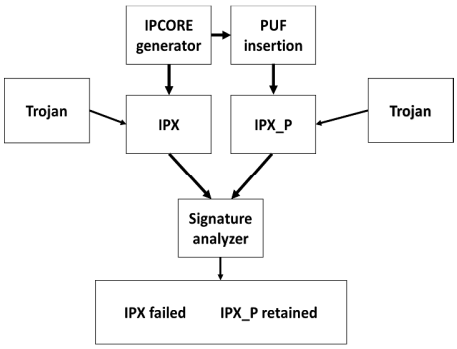

Fig. 1. Block diagram.

As shown in Figure 1, IP customization is performed in conjunction with PUF insertion. Trojans can infect the IP cores generated in regular procedures compared to the PUF-enabled IP cores.

A. Generating and Customizing IP from IP Core

An adder/subtractor IP in Vivado Xilinx was selected as a proof of concept. The core parameters for the adder/subtractor IP in Xilinx Vivado depend on the specific necessity of the design. However, some of the common core parameters that can be customized for the adder/subtractor IP are data width, operation mode, input and output ports, overflow mode, clocking options, implementation options, bit growth, and IP customization. Different modes to customize the adder/subtractor IP are the add mode, carry in, carry out, bypass, synchronous controls and Clock Enable (CE), sync set and clear (reset) priority, borrow in/out sense, active high, and active low. To corrupt the selected IPcore, a combinatorial trojan was implanted in the design, which is a specific event-triggered trojan.

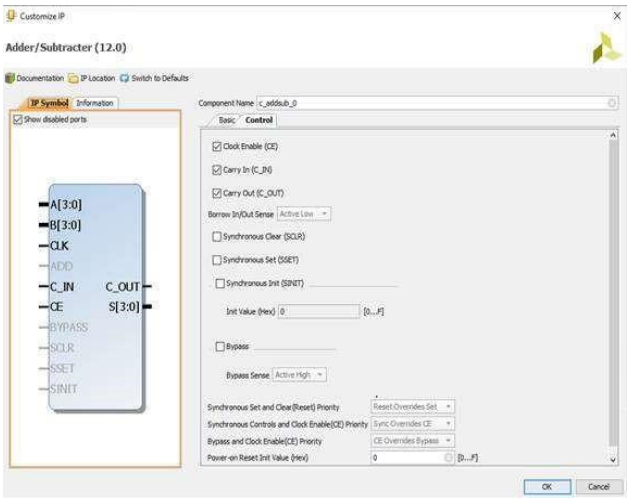

Fig. 2. Core design for the adder/subtractor IP.

B. PUF Insertion

To protect the system against the trojan, a PUF was employed to create signatures for the design to keep it tractable. A PUF and a True Random Number Generator (TRNG) are primary primitives [30]. PUF has the advantage of being compatible with minimal computational resources over the current classical cryptography types. In [31], the effective design, implementation, and analysis of these hardware-based security primitives were described. PUF circuits are used to create unique and reliable signatures for certain electronic circuits [32]. The two primary types of PUFs are strong and weak PUFs. In [33], strong PUF implementations and their use for low-cost authentication were described along with weak PUF implementations and their use in key generation applications. This study also discussed error correction techniques, such as pattern matching and index-based coding.

IV. IMPLEMENTATION AND VALIDATION

Figures 2 and 3 show the implementation of the customized IP of interest, while Figure 4 shows its corresponding validation.

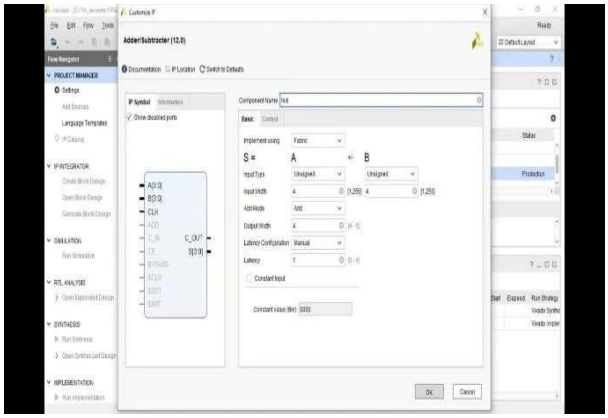

Fig. 3. Design of core IP and customization.

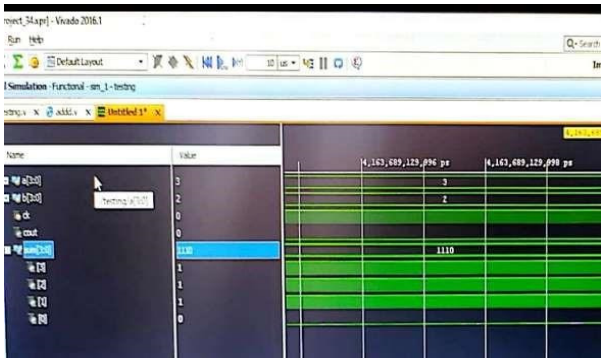

Fig. 4. Simulation results of adder/subtractor IP with PUF.

A. *Trojan Insertion*

To validate the proposed method, a combinational circuit-based trojan was introduced in the IP core, which changes the output at a certain combination of the input pattern. The simulation results show the erroneous output with the insertion of the malicious circuit.

B. *PUF Insertion*

A Butterfly PUF, using the architecture shown in Figure 5 [34], was designed using the Verilog Hardware Description Language (VHDL) to generate the unique signature for the IP of interest.

Initially, the excite signal is set to high to begin the operation of the Butterfly PUF. The Butterfly PUF circuit reaches an unstable operating point because the inputs and outputs of both latches are opposite signals. The excite signal is set low after a few clock pulses. This initiates the transition of the PUF circuit to one of the two stable states, 0 or 1, of the output signal. A Butterfly PUF can generate a single bit, i.e., 0 or 1, for a single clock pulse. Since 8-bit data are needed, the output of the PUF for 8 clock pulses was obtained and then stored in a register to be used as a signature for the IP.

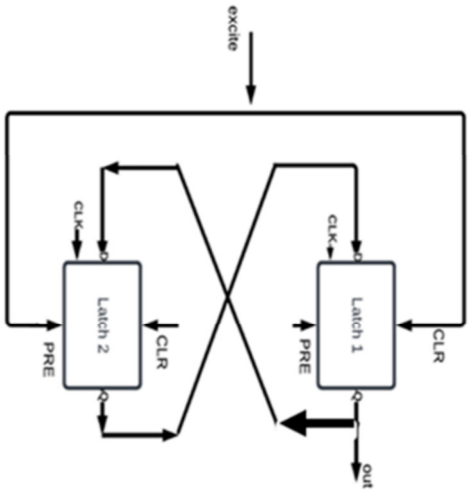

Fig. 5. Butterfly PUF.

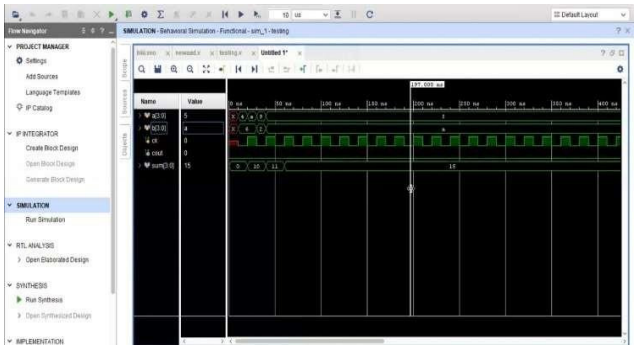

Fig. 6. Simulation results of adder/subtractor IP with PUF.

Figure 7 shows the synthesized schematic of the PUF and Table II shows its synthesis report. Furthermore, Uniqueness (UQ) and Reliability are important metrics that are defined as measures of security.

TABLE I. CELL USAGE REPORT

Cell	COUNT
ADD/SUB IP	1
BUFG	1
CARRY4	48
LUT2	34
LUT4	4
FDRE	8
IBUF	16
OBUF	16

TABLE II. UTILIZATION REPORT

	With PUF		Without PUF		% overhead
Block	Used	Available	Used	Available	
Slice LUTs*	40	63400	32	63400	10
Logic LUTs	40	63400	32	63400	10
Slice Registers	16	126800	16	126800	0
Register as Flip Flop	16	126800	16	126800	0
Bonded IOB	24	210	24	210	0

C. *Uniqueness (UQ)*

Uniqueness is defined as the average inter-chip Hamming Distance (HD) among p devices, where, C is a challenge, and X_i *and* X_j are the respective n-bit responses of i^{th} and j^{th} chips as:

$$Uniqueness = \frac{2}{p(p-1)} \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} \frac{HD(X_i, X_j)}{n} \times 100\% (1)$$

The ideal value for Uniqueness is 50%.

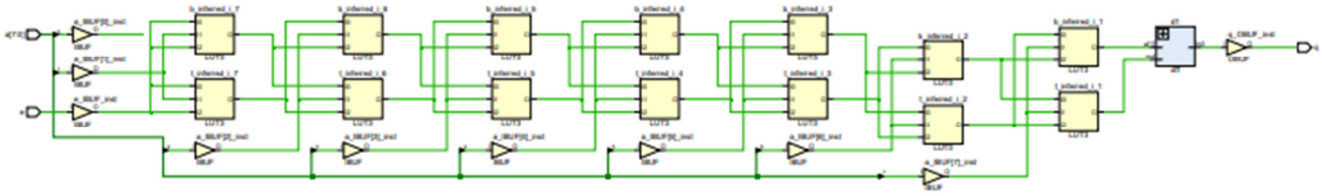

Fig. 7. Synthesis diagram of PUF.

D. *Reliability (RE)*

Reliability measures the consistency in the PUF responses.

$$HD_{INTRA_i} = \frac{1}{s} \sum_{t=1}^{s} \frac{HD(X_i, X_{i,t})}{n} \times 100\%$$

$$Reliability_i = 100\% - HD_{INTRA_i}$$

$$Average\ Reliability = \frac{1}{p} \sum_{i=1}^{p} Reliability_i$$

The ideal value for reliability is 100%.

Table III presents the effect of different sets of coefficients. Different excitations are considered for the signature generation and their Reliability and Uniqueness are observed to be similar. Furthermore, different PUF architectures can be tried to justify the security of the signal processing blocks using PUFs. Although Uniqueness is appreciated, some effort is needed to retrieve the original data from the different coefficients. Every time there is a possibility of changing coefficients. However, the overall response may not change, as shown in Table III.

TABLE III. PERFORMANCE METRICS

Design	Reliability	Uniqueness
Ideal Value	100%	50%
Set 1	98.34	49.00
Set 2	98.57	49.24
Set 3	99.19	48.10
Set 4	98.01	49.10

The design was further validated with ML-based attacks, proving to be effective and efficient in protecting the IP up to 98%. The Butterfly PUF has low hardware complexity and decent accuracy compared to RO-based PUF and Arbiter PUF, as shown in Table IV. The IP was prototyped on an Artix 7 FPGA using Xilinx Vivado. The Butterfly PUF for generating the signature to protect the IP was implemented and validated. The behavioral simulation validates the functional behavior of the entire architecture. The elaborated RTL is shown in Figure 7, and synthesis details are presented in Tables I and II.

The synthesis report shows the hardware utilization of the FPGA, which hardly differs by 2% with and without PUF architectures and is insignificant compared to the security it provides to the IP. Furthermore, the power consumption is presented in Table V, where it presents significant power savings with Butterfly PUF.

TABLE IV. HARDWARE COMPLEXITY WITH DIFFERENT PUFS

	With PUF				Without PUF	
	USED			AVB		
Block	**RO**	**AB**	**BF**		**Used**	**AVB**
Slice LUTs*	54	45	40	63400	32	63400
Logic LUTs	54	45	40	63400	32	63400
Slice Registers	20	18	16	126800	16	126800
Register as Flip Flop	16	16	16	126800	16	126800
Bonded IOB	24	24	24	210	24	210

*RO: Ring Oscillator, AB:Arbiter, BUF: Butterfly, AVB: Available

TABLE V. POWER CONSUMPTION WITH DIFFERENT PUFS (IN MW).

PUF	Static	Dynamic	Total
RO	0.0023	0.45	0.4523
Arbiter	0.0028	0.665	0.6678
Butterfly	0.0018	0.125	0.1268

CONCLUSION

It is essential to implement robust security measures during the design and development stages to detect and prevent such trojans. This study presented a method to ensure the security of IP cores by introducing PUFs to generate signatures that are later analyzed in the design cycle. The nominal hardware overhead is ignored due to the significant security advantages. In addition, the Butterfly PUF had significantly less power consumption than the RO and Arbiter PUFs.

REFERENCES

[1] U. Guin, K. Huang, D. DiMase, J. M. Carulli, M. Tehranipoor, and Y. Makris, "Counterfeit Integrated Circuits: A Rising Threat in the Global Semiconductor Supply Chain," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1207–1228, Aug. 2014, https://doi.org/10.1109/JPROC.2014.2332291.

[2] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, "Trojan Detection using IC Fingerprinting," in *2007 IEEE Symposium on Security and Privacy (SP '07)*, Berkeley, CA, USA, May 2007, pp. 296–310, https://doi.org/10.1109/SP.2007.36.

[3] G. T. Becker, F. Regazzoni, C. Paar, and W. P. Burleson, "Stealthy dopant-level hardware Trojans: extended version," *Journal of Cryptographic Engineering*, vol. 4, no. 1, pp. 19–31, Apr. 2014, https://doi.org/10.1007/s13389-013-0068-0.

[4] S. Ghandali, G. T. Becker, D. Holcomb, and C. Paar, "A Design Methodology for Stealthy Parametric Trojans and Its Application to Bug Attacks," in *Cryptographic Hardware and Embedded Systems – CHES 2016*, Santa Barbara, CA, USA, Aug. 2016, pp. 625–647, https://doi.org/10.1007/978-3-662-53140-2_30.

[5] C. Paar, "Hardware Trojans and Other Threats against Embedded Systems," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, Abu Dhabi, United Arab Emirates, Dec. 2017, https://doi.org/10.1145/3052973.3053885.

[6] S. Bhunia, M. S. Hsiao, M. Banga, and S. Narasimhan, "Hardware Trojan Attacks: Threat Analysis and Countermeasures," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1229–1247, Dec. 2014, https://doi.org/10.1109/JPROC.2014.2334493.

[7] M. Tehranipoor and F. Koushanfar, "A Survey of Hardware Trojan Taxonomy and Detection," *IEEE Design & Test of Computers*, vol. 27, no. 1, pp. 10–25, Jan. 2010, https://doi.org/10.1109/MDT.2010.7.

[8] K. Xiao, D. Forte, Y. Jin, R. Karri, S. Bhunia, and M. Tehranipoor, "Hardware Trojans: Lessons Learned after One Decade of Research," *ACM Transactions on Design Automation of Electronic Systems*, vol. 22, no. 1, Feb. 2016, Art. no. 6, https://doi.org/10.1145/2906147.

[9] S. Adee, "The Hunt For The Kill Switch," *IEEE Spectrum*, vol. 45, no. 5, pp. 34–39, May 2008, https://doi.org/10.1109/MSPEC.2008.4505310.

[10] Y. Jin and Y. Makris, "Hardware Trojans in Wireless Cryptographic ICs," *IEEE Design & Test of Computers*, vol. 27, no. 1, pp. 26–35, Feb. 2010, https://doi.org/10.1109/MDT.2010.21.

[11] S. Skorobogatov and C. Woods, "Breakthrough Silicon Scanning Discovers Backdoor in Military Chip," in *Cryptographic Hardware and Embedded Systems – CHES 2012*, Leuven, Belgium, 2012, pp. 23–40, https://doi.org/10.1007/978-3-642-33027-8_2.

[12] O. Bronchain, L. Dassy, S. Faust, and F. X. Standaert, "Implementing Trojan-Resilient Hardware from (Mostly) Untrusted Components Designed by Colluding Manufacturers," in *Proceedings of the 2018 Workshop on Attacks and Solutions in Hardware Security*, Toronto, Canada, Jan. 2018, pp. 1–10, https://doi.org/10.1145/3266444.3266447.

[13] S. Yu, C. Gu, W. Liu, and M. O'Neill, "A Novel Feature Extraction Strategy for Hardware Trojan Detection," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, Seville, Spain, Oct. 2020, pp. 1–5, https://doi.org/10.1109/ISCAS45731.2020.9180479.

[14] K. Hasegawa, M. Yanagisawa, and N. Togawa, "Trojan-feature extraction at gate-level netlists and its application to hardware-Trojan detection using random forest classifier," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, Baltimore, MD, USA, May 2017, pp. 1–4, https://doi.org/10.1109/ISCAS.2017.8050827.

[15] K. Yang, M. Hicks, Q. Dong, T. Austin, and D. Sylvester, "A2: Analog Malicious Hardware," in *2016 IEEE Symposium on Security and Privacy (SP)*, San Jose, CA, USA, May 2016, pp. 18–37, https://doi.org/10.1109/SP.2016.10.

[16] S. Bhasin and F. Regazzoni, "A survey on hardware trojan detection techniques," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, Lisbon, Portugal, May 2015, pp. 2021–2024, https://doi.org/10.1109/ISCAS.2015.7169073.

[17] R. Kumar, P. Jovanovic, W. Burleson, and I. Polian, "Parametric Trojans for Fault-Injection Attacks on Cryptographic Hardware," in *2014 Workshop on Fault Diagnosis and Tolerance in Cryptography*, Busan, Korea (South), Sep. 2014, pp. 18–28, https://doi.org/10.1109/FDTC.2014.12.

[18] S. Pagliarini, J. Sweeney, K. Mai, S. Blanton, L. Pileggi, and S. Mitra, "Split-Chip Design to Prevent IP Reverse Engineering," *IEEE Design & Test*, vol. 38, no. 4, pp. 109–118, Dec. 2021, https://doi.org/10.1109/MDAT.2020.3033255.

[19] J. Rajendran, V. Jyothi, O. Sinanoglu, and R. Karri, "Design and analysis of ring oscillator based Design-for-Trust technique," in *29th VLSI Test Symposium*, Dana Point, CA, USA, May 2011, pp. 105–110, https://doi.org/10.1109/VTS.2011.5783766.

[20] A. Waksman and S. Sethumadhavan, "Silencing Hardware Backdoors," in *2011 IEEE Symposium on Security and Privacy*, Oakland, CA, USA, May 2011, pp. 49–63, https://doi.org/10.1109/SP.2011.27.

[21] M. Potkonjak, A. Nahapetian, M. Nelson, and T. Massey, "Hardware Trojan horse detection using gate-level characterization," in *Proceedings of the 46th Annual Design Automation Conference*, San Francisco, CA, USA, Jul. 2009, pp. 688–693, https://doi.org/10.1145/1629911.1630091.

[22] L. Lin, W. Burleson, and C. Paar, "MOLES: malicious off-chip leakage enabled by side-channels," in *Proceedings of the 2009 International Conference on Computer-Aided Design*, San Jose, CA, USA, Nov. 2009, pp. 117–122, https://doi.org/10.1145/1687399.1687425.

[23] Y. Alkabani and F. Koushanfar, "Consistency-based characterization for IC Trojan detection," in *Proceedings of the 2009 International Conference on Computer-Aided Design*, San Jose, CA, USA, Nov. 2009, pp. 123–127, https://doi.org/10.1145/1687399.1687426.

[24] A. Baumgarten, A. Tyagi, and J. Zambreno, "Preventing IC Piracy Using Reconfigurable Logic Barriers," *IEEE Design & Test of Computers*, vol. 27, no. 1, pp. 66–75, Feb. 2010, https://doi.org/10.1109/MDT.2010.24.

[25] X. Zhang and M. Tehranipoor, "Case study: Detecting hardware Trojans in third-party digital IP cores," in *2011 IEEE International Symposium on Hardware-Oriented Security and Trust*, San Diego, CA, USA, Jun. 2011, pp. 67–70, https://doi.org/10.1109/HST.2011.5954998.

[26] R. Torrance and D. James, "The state-of-the-art in semiconductor reverse engineering," in *Proceedings of the 48th Design Automation Conference*, San Diego, CA, USA, Mar. 2011, pp. 333–338, https://doi.org/10.1145/2024724.2024805.

[27] S. Wei and M. Potkonjak, "Scalable Hardware Trojan Diagnosis," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 6, pp. 1049–1057, May 2012, https://doi.org/10.1109/TVLSI.2011.2147341.

[28] J. M. K. K. A. Mehdi, "A Distributed-bit SEC-DED RAM with a Self-Testing and Repairing Engine," *International Journal of Performability Engineering*, vol. 1, no. 1, Jul. 2005, Art. no. 79, https://doi.org/10.23940/ijpe.05.1.p79.mag.

[29] N. Q. Luc, T. T. Nguyen, D. H. Quach, T. T. Dao, and N. T. Pham, "Building Applications and Developing Digital Signature Devices based on the Falcon Post-Quantum Digital Signature Scheme," *Engineering, Technology & Applied Science Research*, vol. 13, no. 2, pp. 10401–10406, Apr. 2023, https://doi.org/10.48084/etasr.5674.

[30] S. S. Kumar, J. Guajardo, R. Maes, G. J. Schrijen, and P. Tuyls, "Extended abstract: The butterfly PUF protecting IP on every FPGA," in *2008 IEEE International Workshop on Hardware-Oriented Security and Trust*, Anaheim, CA, USA, Jun. 2008, pp. 67–70, https://doi.org/10.1109/HST.2008.4559053.

[31] B. N. Bukke, K. Manjunathachari, and S. Sabbavarapu, "Implementation of a Finite Impulse Response Filter using PUFs to Avoid Trojans," *Engineering, Technology & Applied Science Research*, vol. 13, no. 6, pp. 12151–12157, Dec. 2023, https://doi.org/10.48084/etasr.6133.

[32] N. N. Anandakumar, S. K. Sanadhya, and M. S. Hashmi, "Design, Implementation and Analysis of Efficient Hardware-Based Security Primitives," in *2020 IFIP/IEEE 28th International Conference on Very Large Scale Integration (VLSI-SOC)*, Salt Lake City, UT, USA, Oct. 2020, pp. 198–199, https://doi.org/10.1109/VLSI-SOC46417.2020.9344097.

[33] K. Dey, M. Kule, and H. Rahaman, "PUF Based Hardware Security: A Review," in *2021 International Symposium on Devices, Circuits and Systems (ISDCS)*, Higashihiroshima, Japan, Mar. 2021, pp. 1–6, https://doi.org/10.1109/ISDCS52006.2021.9397896.

[34] C. Herder, M. D. Yu, F. Koushanfar, and S. Devadas, "Physical Unclonable Functions and Applications: A Tutorial," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126–1141, May 2014, https://doi.org/10.1109/JPROC.2014.2320516.