Implementation of a Finite Impulse Response Filter using PUFs to Avoid Trojans

Balaji Naik Bukke
GITAM (Deemed to be University), India
bbukke@gitam.edu

Kamsali Manjunathachari
GITAM (Deemed to be University), India
mkamsali@gitam.edu

Srinivas Sabbavarapu
Anil Neerukonda Institute of Technology and Sciences, India | IIT Hyderabad, India
ssrinivas.ece@anits.edu.in (corresponding author)

Received: 23 June 2023 | Revised: 15 July 2023, 5 August 2023, and 7 August 2023 | Accepted: 7 September 2023

ABSTRACT

In the modern era of signal processing, digital filters play an important role in real-time applications such as communication, consumer electronics, digital signal processing, audio, etc. In digital filter design, Finite Impulse Response (FIR) filters are highly preferable due to their linear phase and inherent stability. These filters benefit from being time-invariant and simple to implement with minimal computational requirements. Therefore, the hardware security of FIR filters is essential for good performance and reliable results. On the other hand, there is the possibility of hardware threats, such as tampering, reverse engineering, hardware Trojans, etc., as the design of an FIR filter involves many stages. Such hardware attacks on FIR filters can cause several problems, including performance degradation, leakage of confidential information, lack of stability, etc. This study presents the design and implementation of a Trojan-aware FIR filter using Physical Unclonable Functions (PUFs). The key feature of PUFs is that they generate a unique and unpredictable response for each given challenge. In the proposed design, PUFs were used to generate the FIR filter coefficients that are unique and unpredictable by attackers/trojans to improve security. The security of FIR with PUF was tested using ML-based challenges, and the results showed approximately 30% more reliability and consistency compared to the FIR without PUFs.

Keywords-Finite Impulse Response (FIR) filter; hardware threats; Physical Unclonable Functions (PUF); hardware trojan; hardware security

I. INTRODUCTION

The process of placing thousands of transistors on a single chip to create an Integrated Circuit (IC) is known as Very Large-Scale Integration (VLSI). Electronic circuits can consist of RAM, ROM, CPU, and other combinatorial logic elements, and VLSI allows an IC designer to incorporate all of them into a single chip [1]. As the integration density increases with advancements in device and CAD technology, the threat of IC malfunctions at different abstraction levels increases at a huge pace and quantity [2]. There are many hardware threats, such as reverse engineering, tampering, hardware trojans, etc. A hardware trojan is a malicious modification of circuitry. Hardware security aims to counteract these trojans by securing the hardware component of the IC design against unauthorized access. Hardware security is critical as it lays the foundation for the entire system to run. There are many attempts in progress to make the hardware blocks more secure.

There are many hardware security techniques, such as obfuscation, hardware-based cryptography, physical tamper resistance, randomization, and Physical Unclonable Functions (PUFs) that can be used against attacks. PUFs and True Random Number Generators (TRNGs) are the primary primitives. A PUF has the advantage of being compatible with minimal computational resources over the classical cryptography types [2]. PUF is a hardware security method that generates a unique and unpredictable output. The main application of PUFs is low-cost, as authentication uses Strong PUFs and secret-key generation uses weak PUFs [3]. PUFs have many benefits due to their uniqueness, unpredictable nature, low power consumption, scalability, and other qualities. PUFs come in a variety of forms, including SRAM PUF, Arbiter PUF, Ring Oscillator PUF, Butterfly PUF, etc. This study selected Butterfly PUF due to its simplicity and robustness [4].

This study investigated the design of a secure Finite Impulse Response (FIR) filter since it is the basic building block in almost every signal processing application. FIR and Infinite Impulse Response (IIR) filters are two different categories of digital filters that both have benefits and drawbacks of their own. However, FIR filters are frequently used in the design of digital filters due to their linear phase and stability [5]. FIR filters can be implemented as specialized hardware, such as Application-Specific Integrated Circuits (ASICs) [6-7]. A digital filter with an FIR is one whose impulse response has a finite duration and settles to zero in a finite amount of time. Compared to IIR filters, FIR filters are different because they have linear phase, unconditional stability, and easy implementation. FIR filters are extensively used in many applications such as signal processing, image processing, biomedical signal processing, etc. [8-9]. A general FIR filter equation can be expressed as:

$$Y(n) = \sum_{k=0}^{N-1} w(k).x(n-k) \qquad (1)$$

II. RELATED WORKS

The VLSI sector has undergone numerous changes in recent decades, including the breakdown of Integrated Device Manufacturers (IDMs) into foundries [1]. Due to the increasing growth in the size and complexity of the VLSI design, it became challenging to achieve the speed and quality requirements of the IC design. As a result, in [10], an effective model for quick floor planning was presented in the VLSI top-down physical design flow that used active logic reduction technology to accelerate the design flow while reducing internal logic units and maintaining design quality. Logic optimization is the process of finding an equivalent representation of the logic circuit within predetermined constraints and is crucial to VLSI design. Logic optimization is carried out at the gate level in normal custom and semi-custom design flows by converting the input truth table into Boolean values. The streamlined gate logic is used to infer the final transistor netlist that will be organized on a chip. In [11], a genetic method was presented for the gateless VLSI design to eliminate the intermediate gate-level optimization phase and produce optimized transistor netlists. Due to the growing complexity of the semiconductor industry, traditional rule-based technologies cannot address the issues of Electronic Design Automation (EDA). Machine learning (ML), which is expanding rapidly and has recently been incorporated into EDA applications [12], can be used as a remedy for this. Integrating CAD tools with the algorithm design environment to produce hardware designs can be used effectively in the design of VLSI communication systems [13]. Digital Signal Processing (DSP) systems rely heavily on filters, which are also frequently used in VLSI design. A digital filter with a finite period of impulse response is known as an FIR filter. In [8], several design strategies and the necessary parameters for the implementation of FIR were discussed.

As several medicinal applications, signal processing applications, de-noising applications, etc., use FIR filters, designing an effective FIR filter with low power consumption and high speed is crucial. In [9], an effective implementation of FIR filters with high-speed adders was presented for signal-processing applications. In [7], the register minimization retiming technique was used to design an effective filter. The design of low-complexity FIR filters has been the focus of active study over the past 20 years. Current methods for integrating the removal of common subexpressions with the synthesis of filter coefficients are primarily based on bit-width truncation, which raises the implementation cost. In [14], a brand new cost-aware sensitivity-driven method was proposed for the design of FIR filters. In [15], a straightforward and effective design for variable fractional delay FIR filters was presented. Hardware security and trust have become a major security concern as the semiconductor supply chain has become globalized. In [16-17], the security lifecycle for hardware technologies was discussed to explore hardware security, vulnerabilities, trojans, reverse engineering, and other hazards. In [18], an overview of Trojan detection and prevention methods for hardware security was presented. Hardware trojan detection has proven to be extremely difficult due to the improvement of numerous entities in the VLSI design cycle. In [19], a novel algorithm was presented that used netlists of legitimate and trojan-injected circuits to find harmful nets. Recently proposed Trojan horse detection techniques rely on Trojan horse activation to look for false outputs or other measurable abnormal side-channel signals. From an authentication perspective, the time to activate the hardware Trojan circuit is a big issue. In [20], a method was presented to improve hardware Trojan detection and reduce the startup time of Trojans.

As hardware-based security primitives are crucial to a system's protection, it is becoming essential to protect sensitive data as well as the underlying technology. A PUF has the advantage of being compatible with minimal computational resources over the present classical cryptography types. In [2], the effective design, implementation, and analysis of these hardware-based security primitives were described. In [21], PUF circuits were proposed to create unique and reliable signatures for certain electronic circuits. In [22], the security of the IoT framework was thoroughly examined. In [23], the design of feedforward XOR PUFs (FFXOR-PUFs) was presented, where each component PUF used an FF to create an FFPUF. In [24], various homogeneous and heterogeneous FFXOR PUFs were presented and evaluated. A new classification for reconfigurable PUFs (RPUFs) was presented, namely Algorithm-based RPUF (A-RPUF) and Circuit-based RPUF (C-RPUF), and two XOR-based RPUF circuits and an XOR-based Reconfigurable Ring Oscillator PUF (XRRO PUF)) were proposed. The two primary subtypes of PUFs are Strong and Weak PUFs. In [3], strong PUF implementations and their use for low-cost authentication were described along with weak PUF implementations and their use in key generation applications. Additionally, this study explored error correction techniques like pattern matching and index-based coding. In [25], a SEC-DED RAM system was proposed, where a Built-In Self-Test (BIST) and a repairing scheme (STAR) were used to make it fault-tolerant and dynamically reconfigurable. In [26], reversible gates, such as Feynman, Peres, and Toffoli gates, were used to design a concurrent detectable carry select adder that can detect faults. In [27], security issues were addressed at the device level.

As the literature review showed that there is no current design for a secured FIR filter, this study focused on the PUF-based trojan-aware FIR filter implementation.

III. METHODOLOGY

A physical unclonable FIR filter was chosen due to its wide use in signal processing applications, where MAC units play a vital role [28]. The unique fingerprint generated by a PUF is derived from the inherent physical variations in the hardware components such that it cannot be duplicated or cloned. PUFs have many benefits due to their uniqueness, unpredictable nature, low power consumption, and scalability. PUFs come in a variety of forms, including SRAM PUF, Arbiter PUF, Ring Oscillator PUF, Butterfly PUF, etc. Butterfly PUF has high reliability, which makes it hard for attackers to clone the device, and is simple in its architecture and implementation.

A. Effect of Trojan on the FIR filter

Assume an FIR filter designed for an X band of frequency. The response depends on the weights of the filter (coefficient). If $c1$, $c2$, and $c3$ define the $f1$ frequency, $c1d$, $c2d$, and $c3d$ may not produce the same frequency $f1$. If the coefficients change ∂c, then frequency also may vary by ∂f. The change in the coefficient may be caused due to some Trojans. Therefore, it is evident that PUFs can be used to secure the filter response from the Trojans in the coefficient space.

B. Design of a General FIR Filter

The proposed PUF FIR filter was developed as shown in Figure 1, following these steps: (a) Design a general FIR filter, (b) generation of coefficients from the Butterfly PUF, and (c) mapping of the generated coefficients to the designed FIR filter.

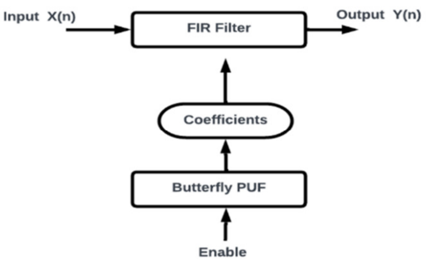


Fig. 1. Flowchart of the proposed filter.

A basic 4-tap FIR filter was designed using the Verilog Hardware Description Language (HDL). The FIR filter works by convolving the input signal with a set of filter coefficients or taps to produce the filtered output, as shown in Figure 2. Initially, the input signal is sampled at regular intervals to obtain a discrete-time signal. The sampled input signal is then multiplied by a set of filter coefficients, which are predetermined and stored in the filter. The products of the input samples and the filter coefficients are added together, producing a sum. The sum obtained is the output, i.e. the filtered signal for that input sample. After the filtered output is obtained, the input samples are shifted by one position, and the process is repeated with the new input sample.

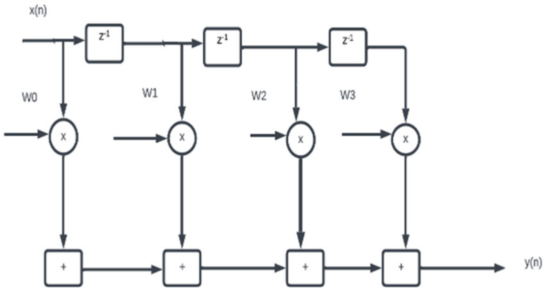


Fig. 2. Basic FIR filter.

A Trojan could be in the coefficients path, where the change in coefficients may change the entire response. Therefore, to protect this path, a secured hardware block can be used to generate the coefficients and prevent any Trojan from being able to damage the filter response.

C. Generation of Coefficients from the Butterfly PUF

PUFs are devised by generating the security key and considering the unique characteristics inherent in a physical device [13]. The physical properties of silicon devices are determined by exploring a set of inputs (challenges) and a set of outputs (responses). PUFs can also be used in extracting the signature of chips, device authentication, IP protection, seeding a Pseudo-Random Number Generator (PRNG), securing IPs, etc. [6-9, 15]. Most FPGA PUFs are delay-based working around the race conditions, frequency variations found in IC, and meta-stability [14, 16]. PUF architectures are classified as strong and weak [5, 17]. Strong PUFs can be used in authentication directly due to the richness of Challenge-Response Pairs (CRPs), making it impossible for the attacker to modify the CRPs in a restricted time frame [18]. However, weak PUFs limited numbers of CRPs and can be used to generate cryptographic keys in PRNG, protect IPs, and identity generation. In these weak PUFs, an attacker cannot access the response, as these responses must be kept private [18]. Here, the proposed hybrid PUF architecture is claimed to be inefficient on weak PUFs as a result of the CRP limitation.

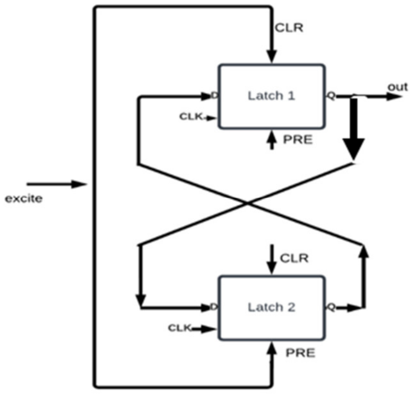


Fig. 3. Butterfly PUF.

Figure 3 shows the architecture of a Butterfly PUF to generate the unique coefficients for the FIR filter that act as security keys for the filter. Initially, the exciting signal is set to high to begin the operation of the Butterfly PUF. The butterfly PUF circuit reaches an unstable operating point because the input and output 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 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. The output of the PUF for 8 clock pulses is obtained since 8-bit data are needed, and it is then placed in a register so that it can be used as a coefficient for the FIR filter. Four 8-bit coefficients are produced by four 4-Butterfly PUFs for a 4-tap FIR Filter, and N-Butterfly PUFs are needed for an N-tap FIR Filter. Figure 5 presents the different sets of coefficients generated using these PUFs.

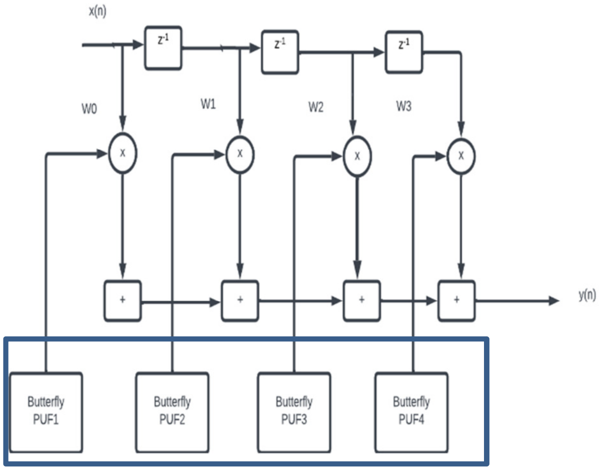
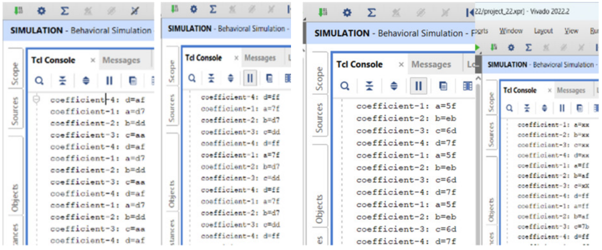


Fig. 4. Block diagram of the secured FIR filter - Secured Coefficient generation through PUFs.


SIMULATION - Behavioral Simulation
Tcl Console
Messages
Scope
Sources
Objects
22/project_22.xpr] - Vivado 2022.2
Window
Layout
View

```
coefficient-4: d=af
coefficient-1: a=d7
coefficient-2: b=dd
coefficient-3: c=aa
coefficient-4: d=af
coefficient-1: a=d7
coefficient-2: b=dd
coefficient-3: c=aa
coefficient-4: d=af
coefficient-1: a=d7
coefficient-2: b=dd
coefficient-3: c=aa
coefficient-4: d=af
```

```
coefficient-4: d=ff
coefficient-1: a=7f
coefficient-2: b=d7
coefficient-3: c=dd
coefficient-4: d=ff
coefficient-1: a=7f
coefficient-2: b=d7
coefficient-3: c=dd
coefficient-4: d=ff
coefficient-1: a=7f
coefficient-2: b=d7
coefficient-3: c=dd
coefficient-4: d=ff
```

```
coefficient-1: a=5f
coefficient-2: b=eb
coefficient-3: c=6d
coefficient-4: d=7f
coefficient-1: a=5f
coefficient-2: b=eb
coefficient-3: c=6d
coefficient-4: d=7f
coefficient-1: a=5f
coefficient-2: b=eb
coefficient-3: c=6d
coefficient-4: d=7f
```

```
coefficient-1: a=xx
coefficient-2: b=xx
coefficient-3: c=xx
coefficient-4: d=xx
coefficient-1: a=ff
coefficient-2: b=af
coefficient-3: c=xX
coefficient-4: d=ff
coefficient-1: a=ff
coefficient-2: b=af
coefficient-3: c=7b
coefficient-4: d=ff
```

Fig. 5. Four different sets of coefficients.

D. *Mapping the Generated Coefficients to the Designed FIR Filter*

The generated coefficients of the Butterfly PUF are given to the designed FIR filter, as indicated in Figure 4. After applying these coefficients to the filter, it performs a convolution operation to produce the filtered output, i.e. the coefficients will be multiplied with the input signal and then all the products will be added together to produce a sum, which is the output of the FIR Filter. Let the generated coefficients of the Butterfly PUF for a 4-tap FIR Filter be $W0$, $W1$, $W2$, and $W3$ and the input be $x(n)$. Then the output equation of the 4-tap FIR Filter can be written as:

$$Y = W_0 \ X(n) + \ W_1 \, X(n-1) + \ W_2 \, X(n-2) + W_3 \, X(n-3) \tag{2}$$

As the coefficients generated from the Butterfly PUF are unique, it is difficult to replicate or reverse engineer the filter. This combination of security and uniqueness makes PUF-based FIR filters more secure, and they can be used for applications such as secure signal processing, cryptographic key generation, and secure communication.

IV. IMPLEMENTATION AND RESULTS

The proposed design was implemented on an Xilinx Vivado. targeting Artix 7 FPGAs. Figure 6 shows the results obtained by simulating the design of the Butterfly PUF. Figure 7 shows the results of the simulated FIR filter, which was examined with different coefficients to check its consistency, proving its suitability for being a valid candidate to avoid any type of Trojan attacks with the help of PUFs.

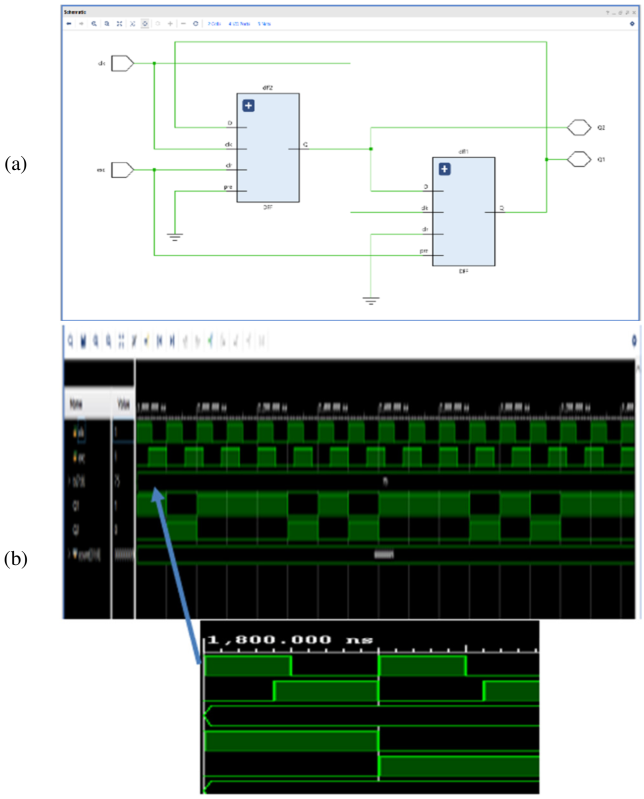


Fig. 6. (a) Schematic and (b) output of the butterfly PUF.

Tables I and II show the cell utilization, where it is evident that the utilization did not increase significantly with the introduction of the PUF circuitry. Security was further validated by applying 1000 ML attacks iteratively 10 times. The proposed PUF FIR design can negate hardware overhead with a decent security improvement. Figure 7 shows the schematic of the designed PUF FIR filter. Figure 8 shows the corresponding slice using VIVADO 2016.1, which was drawn from the block diagram shown in Figure 4. Figure 9 presents the corresponding output of the FIR filter for the different sets of coefficients shown in Figure 5.

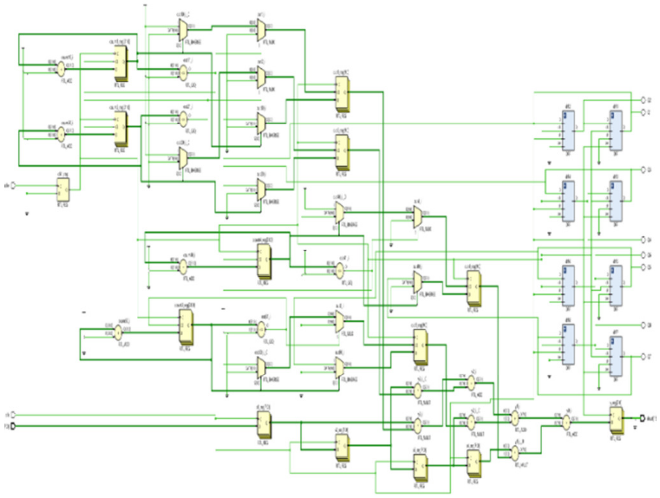


Fig. 7. FIR Schematic diagram.

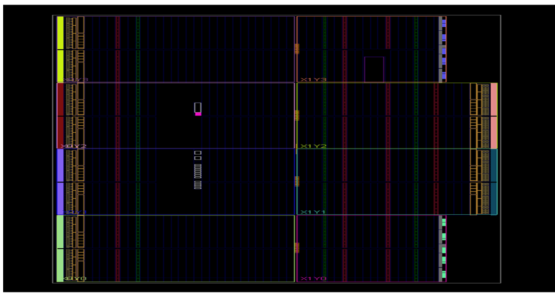
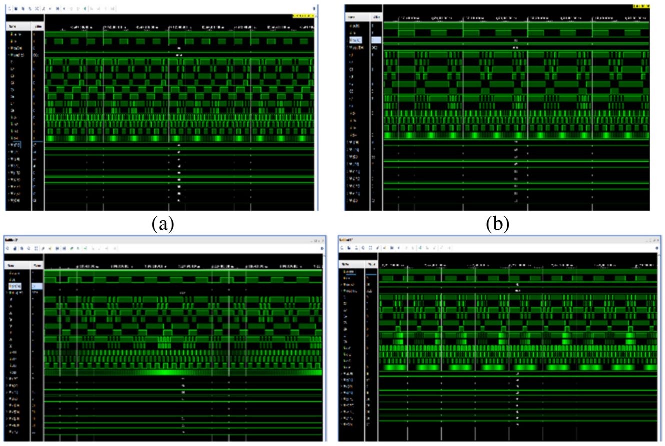


Fig. 8. FIR Slice using VIVADO 2016.1.



(a)

(b)

(c)

(d)

Fig. 9. FIR outputs for different sets of coefficients (Figure 5): (a) Set 1, (b) Set 2, (c) Set 3, and (d) Set 4.

TABLE I. CELL USAGE REPORT

Cell	COUNT
BUFG	1
CARRY4	30
LUT2	48
LUT4	4
FDRE	16
IBUF	1
OBUF	24

TABLE II. UTILIZATION REPORT

	With PUF		Without PUF		%
Block	**Used**	**Available**	**Used**	**Available**	**overhead**
Slice LUTs*	48	63400	36	63400	25
Logic LUTs	48	63400	36	63400	25
Slice Registers	16	126800	16	126800	0
Registers as Flip Flop	16	126800	16	126800	0
Bonded IOB	25	210	25	210	0

Furthermore, two important performance metrics, Uniqueness (UQ) and Reliability were defined and summarized. The ideal value of uniqueness is 50%. Here, X_i and X_j are the respective n-bit responses of the i-th and j-th chips for the same challenge C, and uniqueness is expressed as the average inter-chip Hamming Distance (HD) among p devices.

$$uniqueness = \frac{2}{p(p-1)}\sum_{i=1}^{p-1}\sum_{j=i+1}^{p}\frac{HD(X_i,X_j)}{n}\times 100\% \quad (3)$$

Reliability (RE) is the measure of consistency in the responses generated by the PUF.

$$HD_{INTRA_i} = \frac{1}{s}\sum_{t=1}^{s}\frac{HD(X_i,X_{i,t})}{n}\times 100\% \quad (4)$$

$$Reliability_i = 100\% - HD_{INTRA_i} \quad (5)$$

$$Average\ Reliability = \frac{1}{p}\sum_{i=1}^{p} Reliability_i \quad (6)$$

The ideal value of reliability is 100%, and for HD_{INTRA} is 0%. Here, X_i is the reference response of the i-th chip, $X_{i,t}$ is the current response at time t, and s is the number of responses for similar challenges. Table III presents the effect of different sets of coefficients. Different coefficients were considered for the filter, and their Reliability and Uniqueness were observed to be similar. 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 and the different response curves in Figure 9.

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

V. CONCLUSION

This paper presented a secure design of FIR filters using PUFs. As PUFs generate unique responses, it is difficult to replicate, clone, or reverse engineer the filter. Here, the coefficients generated from the PUF act as a secret key for the FIR filter, which is unpredictable. Therefore, the security of the FIR filter is improved, making it secure from hardware Trojans or other hardware attacks. Security was further validated by iteratively applying several ML attacks, showing around 30% consistency. The proposed design with PUFs can negate the hardware overhead of around 25% with decent security improvement. Therefore, PUFs offer a promising approach to hardware security, as they provide a means of generating unique and unpredictable responses that are resistant to many types of hardware attacks. This study can be extended to improve security to different communication modules such as OFDM and MIMO blocks, where, the frequency selection may be vulnerable to hardware threats.

REFERENCES

[1] J. Y. Chen, "Transformation of VLSI technologies, systems and applications the rise of foundry and its ecosystem," in *2013 International Symposium on VLSI Technology, Systems and Application (VLSI-TSA)*, Hsinchu, Taiwan, Apr. 2013, pp. 1–2, https://doi.org/10.1109/VLSI-TSA.2013.6545604.

[2] 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, Jul. 2020, pp. 198–199, https://doi.org/10.1109/VLSI-SOC46417.2020.9344097.

[3] 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, Dec. 2014, https://doi.org/10.1109/JPROC.2014.2320516.

[4] 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.

[5] R. Pal, "Comparison of the design of FIR and IIR filters for a given specification and removal of phase distortion from IIR filters," in *2017 International Conference on Advances in Computing, Communication and Control (ICAC3)*, Mumbai, India, Sep. 2017, pp. 1–3, https://doi.org/10.1109/ICAC3.2017.8318772.

[6] L. Merani and S. L. Lu, "A self-timed approach to VLSI digital filter design," in *Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing*, Victoria, BC, Canada, Feb. 1993, vol. 2, pp. 402–406 vol.2, https://doi.org/10.1109/PACRIM.1993.407336.

[7] D. Yagain and K. A. Vijaya, "FIR filter design based on retiming automation using VLSI design metrics," in *2013 International Conference on Technology, Informatics, Management, Engineering and Environment*, Bandung, Indonesia, Jun. 2013, pp. 17–22, https://doi.org/10.1109/TIME-E.2013.6611956.

[8] M. B. Trimale and Chilveri, "A review: FIR filter implementation," in *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bangalore, India, Feb. 2017, pp. 137–141, https://doi.org/10.1109/RTEICT.2017.8256573.

[9] S. Akash, M. Ajeeth, and N. Radha, "An Efficient Implementation of FIR Filter Using High Speed Adders For Signal Processing Applications," in *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, Jul. 2020, pp. 1047–1051, https://doi.org/10.1109/ICIRCA48905.2020.9183114.

[10] Y. Zhou, Y. Yan, and W. Yan, "A method to speed up VLSI hierarchical physical design in floorplanning," in *2017 IEEE 12th International Conference on ASIC (ASICON)*, Guiyang, China, Jul. 2017, pp. 347–350, https://doi.org/10.1109/ASICON.2017.8252484.

[11] M. Shoaib, N. Mahammad Sk, and V. Kamakoti., "A genetic approach to gateless custom VLSI design flow," in *2007 Internatonal Conference on Microelectronics*, Cairo, Egypt, Sep. 2007, pp. 403–406, https://doi.org/10.1109/ICM.2007.4497739.

[12] L. Wang and M. Luo, "Machine Learning Applications and Opportunities in IC Design Flow," in *2019 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, Apr. 2019, pp. 1–3, https://doi.org/10.1109/VLSI-DAT.2019.8742073.

[13] R. Jain, C. Chien, E. Cohen, and L. Ho, "Simulation and synthesis of VLSI communication systems," in *Proceedings Eleventh International Conference on VLSI Design*, Chennai, India, Jan. 1998, pp. 336–341, https://doi.org/10.1109/ICVD.1998.646629.

[14] J. Chen, J. Tan, C. H. Chang, and F. Feng, "A New Cost-Aware Sensitivity-Driven Algorithm for the Design of FIR Filters," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 6, pp. 1588–1598, Jun. 2017, https://doi.org/10.1109/TCSI.2016.2557840.

[15] H. Zhao and J. Yu, "A simple and efficient design of variable fractional delay FIR filters," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, no. 2, pp. 157–160, Oct. 2006, https://doi.org/10.1109/TCSII.2005.856673.

[16] H. Khattri, N. K. V. Mangipudi, and S. Mandujano, "HSDL: A Security Development Lifecycle for hardware technologies," in *2012 IEEE International Symposium on Hardware-Oriented Security and Trust*, San Francisco, CA, USA, Jun. 2012, pp. 116–121, https://doi.org/10.1109/HST.2012.6224330.

[17] W. Hu, C. H. Chang, A. Sengupta, S. Bhunia, R. Kastner, and H. Li, "An Overview of Hardware Security and Trust: Threats, Countermeasures, and Design Tools," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 6, pp. 1010–1038, Jun. 2021, https://doi.org/10.1109/TCAD.2020.3047976.

[18] R. S. Chakraborty, S. Narasimhan, and S. Bhunia, "Hardware Trojan: Threats and emerging solutions," in *2009 IEEE International High Level Design Validation and Test Workshop*, San Francisco, CA, USA, Aug. 2009, pp. 166–171, https://doi.org/10.1109/HLDVT.2009.5340158.

[19] S. Rajendran and M. L. Regeena, "A Novel Algorithm for Hardware Trojan Detection Through Reverse Engineering," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 4, pp. 1154–1166, Apr. 2022, https://doi.org/10.1109/TCAD.2021.3073855.

[20] H. Salmani, M. Tehranipoor, and J. Plusquellic, "A Novel Technique for Improving Hardware Trojan Detection and Reducing Trojan Activation Time," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 1, pp. 112–125, Jan. 2012, https://doi.org/10.1109/TVLSI.2010.2093547.

[21] 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.

[22] B. E. Sabir, M. Youssfi, O. Bouattane, and H. Allali, "Towards a New Model to Secure IoT-based Smart Home Mobile Agents using Blockchain Technology," *Engineering, Technology & Applied Science Research*, vol. 10, no. 2, pp. 5441–5447, Apr. 2020, https://doi.org/10.48084/etasr.3394.

[23] S. V. S. Avvaru, Z. Zeng, and K. K. Parhi, "Homogeneous and Heterogeneous Feed-Forward XOR Physical Unclonable Functions," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2485–2498, 2020, https://doi.org/10.1109/TIFS.2020.2968113.

[24] W. Liu *et al.*, "XOR-Based Low-Cost Reconfigurable PUFs for IoT Security," *ACM Transactions on Embedded Computing Systems*, vol. 18, no. 3, Dec. 2019, https://doi.org/10.1145/3274666.

[25] 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, pp. 79-88, Jul. 2005, https://doi.org/10.23940/ijpe.05.1.p79.mag.

[26] V. N. and K. Sambath, "Implementation of Normal Urdhva Tiryakbhayam Multiplier in VLSI," *International Journal of Performability Engineering*, vol. 17, no. 6, pp. 511–518, Jun. 2021, https://doi.org/10.23940/ijpe.21.06.p3.511518.

[27] 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.

[28] K. V. Gowreesrinivas, S. Srinivas, and P. Samundiswary, "FPGA Implementation of a Resource Efficient Vedic Multiplier using SPST Adders," *Engineering, Technology & Applied Science Research*, vol. 13, no. 3, pp. 10698–10702, Jun. 2023, https://doi.org/10.48084/etasr.5797.

AUTHORS PROFILE

Balaji Naik Bukke obtained the B.Sc. in Electronics and Communication Engineering in 2004 from GITAM College of Engineering and an M.Sc. in VLSI design and Embedded Systems in 2008 from the National Institute of Technology Rourkela. He is currently pursuing his Ph.D. He is currently an Assistant Professor in EECE, School of Technology, GITAM (Deemed to be University), Hyderabad, India. His research interest lies in Low-Power VLSI Design and Hardware Security.

K. Manjunathachari obtained a Ph.D. from JNT University, Kakinada, and an M.Sc. from JNT University, Hyderabad. He has more than 17 years of teaching and 3 years of industry experience. Currently, he is a professor and Head of the EECE department at GITAM University, Hyderabad, Telangana, India. He has published many papers in Visual Signal and Image Processing.

Srinivas Sabbavarapu obtained a BSc. in Electronics and Communication Engineering in 2004 from V. R. Siddhartha Engineering College and an M.Sc. in VLSI design and Embedded Systems in 2008 from the National Institute of Technology Rourkela, India. He received his Ph.D. from the Indian Institute of Technology (IIT), Hyderabad, India. He is an Associate Professor at the ANITS (Autonomous) Engineering College, Visakhapatnam. His research interest lies in CAD for VLSI, Low Power VLSI Design, and Hardware Security.