A Novel SRAM-based Cross-Coupled BRAM PUF Featuring Entropy-Driven Addressing for AMD FPGAs

AHMAD AL KHAS and SALIH BAYAR, Faculty of Engineering, Dept. of Electrical-Electronics Eng., Marmara University, Türkiye
IHSAN CICEK*, Faculty of Engineering, Dept. of Electronics Eng., Gebze Technical University, Türkiye

SRAM-based Physically Unclonable Functions (PUFs) are critical for securing FPGA-based embedded systems. However, traditional implementations face significant limitations: they typically require power cycling to extract entropy or rely on external complex circuitry to drive address sequences, potentially creating a larger attack surface for adversaries. This paper proposes a novel, fully standalone PUF architecture on AMD FPGAs that uses the timing violations of read and write operations. Unlike prior collision-based designs that depend on externally supplied random addresses, our proposed architecture introduces a mutual feedback loop where the metastable response of one Block RAM dynamically determines the address of its cross-coupled counterpart. This mechanism creates a self-sustaining entropy source that functions without depending on an external driver or power cycling. The design was implemented and validated on eight AMD Artix-7 FPGAs using a custom MicroBlaze-based data acquisition system. Experimental results demonstrate that the CC-RIWC PUF achieves an average uniqueness of 41.45% and reliability up to 98.37% across a wide temperature range ($-18^{\circ}C$ to $50^{\circ}C$). While the self-driving cross-coupled architecture introduces a trade-off in resource usage (44 slices) and throughput (900 Mbps) compared to externally-driven primitives, it provides a superior architectural advantage by delivering a complete, standalone entropy primitive suitable for resource-constrained, always-on embedded systems like Internet-of-Things.

CCS Concepts: • **Hardware → Reconfigurable logic applications**; • **Security and privacy → Embedded systems security**; **Hardware security implementation**.

Additional Key Words and Phrases: Physical Unclonable Function, PUF, SRAM PUF, Cryptography, FPGA, Embedded systems

1 Introduction

Physically unclonable functions (PUFs) play a vital role in cryptographic systems, enabling the extraction of unique device-specific identifiers from inherent physical variations. Their cost-effective and robust nature makes them particularly attractive for securing embedded systems. Specifically, they are cost-effective as they leverage existing hardware resources without requiring dedicated external circuitry, and they are robust due to their ability to generate stable, device-unique responses even under varying environmental and operating conditions. Hence, they have evolved as a promising solution, offering a foundation of security by using the inherent physical uniqueness of hardware, a vital countermeasure in a world of increasing digital fragility [3, 4].

PUFs have gained substantial attention in hardware security due to their ability to take advantage of microscopic manufacturing variations to generate unique device-specific responses to input challenges [15, 44]. This

*Corresponding author

Authors' Contact Information: Ahmad Al Khas, ahmadal@marun.edu.tr; Salih Bayar, salih.bayar@marmara.edu.tr, Faculty of Engineering, Dept. of Electrical-Electronics Eng., Marmara University, Istanbul, Türkiye; Ihsan Cicek, Faculty of Engineering, Dept. of Electronics Eng., Gebze Technical University, Kocaeli, Türkiye, ihsancicek@gtu.edu.tr.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM 1936-7414/2026/9-ART
https://doi.org/10.1145/3848637

Just Accepted

approach enables secure cryptographic key generation, robust authentication mechanisms, and other hardware-centric security applications. The utilization of PUFs in authentication protocols has gained significant attention primarily because they offer the potential to minimize the reliance on complex error correction technologies or fuzzy extractors. By achieving higher intrinsic reliability, such designs can significantly reduce the storage and computational expenses typically associated with stabilizing PUF responses across varying environmental conditions. Beyond their cryptographic utility, PUFs are crucial in protecting the ownership and integrity of hardware and intellectual property within electronic systems [17], highlighting their versatility in safeguarding assets across various applications, including, but not limited to, application-specific architectures. Recently, PUFs have also been explored for secure key generation, offering novel pathways to improve the resilience and confidentiality of digital assets in security-critical embedded and software applications [11, 43].

Since the advent of PUF technology, a diverse range of implementation techniques have been introduced in the literature. The diversity of these techniques has facilitated their integration into various applications. Physical Unclonable Functions are generally categorized into two classes based on their challenge-response space: Weak and Strong PUFs [6, 24, 37]. Weak PUFs typically support a limited number of challenges, often restricted by the number of physical components, and are primarily used for secret key storage. In contrast, Strong PUFs possess a challenge space so large that an exhaustive measurement of all challenge-response pairs (CRPs) is infeasible within a limited timeframe. These classifications help in selecting the appropriate PUF architecture for the specific application. The classification of the PUF, whether strong or weak, is not relevant to the security characteristics of the PUF; rather, it denotes the total number of unique CRPs that are facilitated [26]. Fig. 1 presents the categories of common PUF types found in the literature.

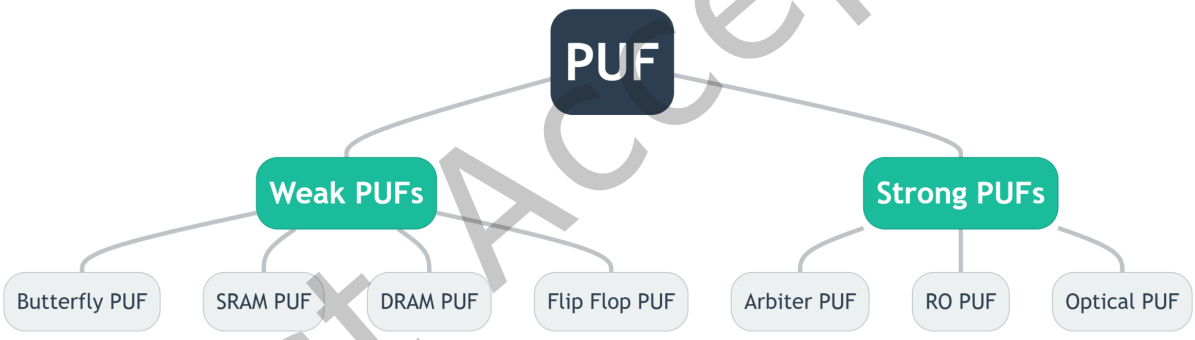

Fig. 1. A taxonomy of common PUF architectures

PUFs leverage inherent manufacturing variations to generate unique responses. Strong PUFs such as Arbiter PUFs and Ring Oscillator (RO) PUFs are widely studied for their entropy-rich outputs [8, 47, 49]. RO PUFs are based on frequency differences due to process-induced device parameter variations. However, RO PUFs are known to have high power consumption [19, 32, 40]. Similarly, Arbiter PUFs exploit delay variations in identically routed paths, with the response determined by the race's outcome between multiple signals [45], yet it consumes less resources than RO [60] . Their implementation on Field Programmable Gate Arrays (FPGAs) is challenging due to strict manual routing requirements [51, 55]. Optical PUFs, while offering high complexity and resistance to modeling attacks, are not suitable for integration into FPGA or ASIC-based platforms due to their reliance on physical optics and external image capture hardware [23]. Although strong PUFs provide higher entropy, their stability and implementation constraints must be considered carefully when selecting security-critical applications. In contrast, weak PUFs including Butterfly and SRAM PUFs, are widely studied for their adaptability, and high reliability [27, 28]. Butterfly PUFs utilize cross-coupled latch circuits to generate stable responses

Just Accepted

without requiring dedicated memory cells [27]. They offer resilience to environmental variations and are ideal for lightweight cryptographic applications thanks to their simple architecture [28, 48, 50].

In this paper, we propose a novel SRAM-based PUF design featuring a unique entropy-driven addressing mechanism that achieves high uniqueness and reliability. Our proposed architecture competes with existing designs in terms of throughput and area efficiency, while its state-dependent behavior allows for a non-additive expansion of the challenge-response space. Furthermore, similar to the methodologies presented by Cicek et al. [13] and Wild et al. [54], our architecture successfully extracts entropy at runtime via read-write collisions rather than relying on initial power-up states. While runtime generation without power-cycling has been demonstrated previously, retaining this capability remains a critical architectural asset of our design compared to conventional SRAM PUFs, making it highly suitable for on-demand key generation in continuously active, always-on embedded environments.

This dynamic scalability enables the construction of high-entropy primitives suitable for robust and secure authentication in hardware-constrained environments.

The contributions of this paper are:

- Entropy-Driven Addressing: We introduce a self-driving address generation scheme where the response of BRAM-0 dictates the address of BRAM-1 (and vice-versa). This eliminates the need for external driver circuitry or the designer's external choice of address required in our previous design, transforming the circuit into a standalone PUF.
- Dynamic CRP Expansion: We demonstrate a sequential challenge accumulation method (Section 6.1) that allows for on-demand expansion of the CRP space without resetting the device, a feature not present in standard weak PUF implementations.

The remainder of this paper is organized as follows. Section 2 reviews previous work on PUF architectures. Section 3 provides a detailed discussion on the operational principles of AMD FPGA Block RAMs (BRAMs). Section 4 presents the proposed SRAM-based PUF architecture that uses cross-coupled BRAMs to induce timing-based metastability without requiring a dedicated processor or power-cycling technique. Section 5 describes the lightweight and fully synthesizable hardware implementation utilizing minimal FPGA resources. Section 6 presents extensive experimental validation in eight FPGA devices, demonstrating the potential of the proposed design through a sequential challenge-based CRP space expansion mechanism. Finally, Section 7 concludes the paper by summarizing the key findings and describing the directions for future research.

2 Related Work

SRAM-based PUFs are among the most extensively studied types of PUFs, widely recognized for their strong security characteristics and practicality in hardware security applications [10, 11]. These PUFs exploit the intrinsic power-up behavior of SRAM cells, which produce device-specific, unpredictable responses due to random process variations. Each SRAM cell consists of six transistors: four arranged in a cross-coupled configuration forming the bistable latch, and two serving as access transistors [13, 14]. Upon power-on, SRAM cells randomly initialize to a logic '0' or '1' state, producing a unique bitstream that can be used as a hardware fingerprint [41].

In this traditional model, the start-up procedure serves as the physical mechanism that enables the extraction of entropy from the cross-coupled inverters. Within this framework, the memory addresses (locations) of the SRAM cells are defined as the challenges, while the resulting power-up states at those specific locations constitute the responses [21]. This behavior has been leveraged to enable secure secret key generation and authentication mechanisms. To enhance their robustness and efficiency, SRAM PUF designs have incorporated techniques such as body voltage control to reduce power and area overhead [34, 36], and memory retention strategies to increase response stability and repeatability under varying environmental conditions [43, 59]. Collectively, these advancements have solidified SRAM PUFs as viable candidates for security-critical applications.

Just Accepted

Expanding beyond traditional designs, Holcomb et al. [25] proposed the Bitline PUF, which differs from conventional SRAM PUFs by reading multiple SRAM columns simultaneously, producing a distinct form of entropy. Although this approach presents benefits for ASIC-based PUF applications, it fails to address limitations encountered when implementing PUFs on FPGAs. A fundamental challenge in FPGA-based PUF development stems from restricted access to internal memory circuitry, which is often proprietary and not fully controllable by the end users. Due to these limitations, researchers working with FPGAs have traditionally favored strong PUF architectures such as arbiter-based or RO PUFs.

Efforts to overcome these issues include the work by Wild et al. [54], who proposed using partial reconfiguration via the Internal Configuration Access Port (ICAP) to selectively reset BRAMs, thereby bypassing the automatic BRAM reset on FPGA power-up. While effective, accessing the ICAP whether governed by an embedded hard processor (like an ARM in a ZYNQ SoC), a deployed softcore processor, or a dedicated logic state machine inherently requires manipulating the FPGA's configuration layer. This introduces additional design complexity, security access considerations, and hardware overhead. In contrast, our proposed architecture generates runtime entropy entirely at the application-logic level through read-write collisions. Our design provides a highly portable, self-contained entropy source that minimizes architectural overhead without needing to manipulate the FPGA's configuration layer.

Recent advancements in collision-based PUFs have demonstrated the viability of using read-write conflicts to generate entropy on FPGAs. Notably, Cicek et al. [13] established the fundamental mechanism of inducing metastability in True Dual-Port (TDP) BRAMs, achieving high throughput by driving the address ports with external random patterns. However, such designs operate as driven primitives: they are dependent on an external entropy source or a host processor to supply non-deterministic address sequences which increase area. In contrast, the new design proposed in this work provides a standalone PUF architecture that does not depend on any external driver through implementing a feedback loop between cross-coupled BRAMs. This approach presents key challenges. First, the proprietary nature of BRAM implementations limits detailed knowledge beyond vendor documentation. Second, FPGA vendors typically enforce deterministic power-up behavior in BRAM cells (i.e. resetting all BRAM cells to zero (see page 33) [1]), undermining their potential as entropy sources. Third, the metastability-mitigating design of FPGAs complicates the use of power-cycling techniques commonly applied in ASIC PUF implementations. Thus, a thorough understanding of BRAM operation is essential to design a new SRAM-based PUF architecture that can address aforementioned challenges.

3 Functional Operation of FPGA BRAM

The 7-series devices of AMD FPGAs provide Block RAMs within the FPGA fabric to serve various applications. In our study, we used the FPGA device XC7A35T-1CPG236C, which offers 50 embedded BRAMs, each having a storage capacity of 36 Kbits. Each 36 Kbit size BRAM can be used as a whole or as two separate 18 Kbit BRAMs; with this, we can use up to 100 BRAMs if all BRAMs were configured as 18 Kbits [1]. FPGA BRAMs offer two modes of operation: TDP and simple dual port (SDP). The TDP mode offers the capability to independently use the input and output (I/O) data ports. In contrast, in the SDP mode, the ports are combined for the joint final input and output, facilitating read and write operations. The architectural differences between TDP and SDP modes are illustrated in Figures 1-1 and 1-6 of the official AMD documentation [1]. Successful reading and writing operations can be performed concurrently in TDP mode with configurable data widths up to 36 bits (32 bits for data and 4 bits for parity). The primary distinction in our context is that the SDP mode leverages the underlying hardware ports to provide a dedicated write interface and a dedicated read interface, often used to maximize data bus width through port concatenation. Unlike TDP mode, which can experience structural address collisions from two independent simultaneous write operations, the SDP configuration is inherently protected from write-write conflicts because it presents only a single logical write interface. However, as specified

Just Accepted

in the vendor documentation [1], SDP mode still allows for operational collisions when the read and write interfaces access the same memory location simultaneously. Our proposed architecture specifically exploits these operational collisions to trigger the nondeterministic timing behavior required for entropy generation.

For a regular read operation of the BRAM, the address is loaded before the start of the operation. Performing a read operation requires only one clock cycle; during the cycle and until it passes, the BRAM will load the stored data into its output latches [1]. Similarly, the write operation takes one cycle from enabling the write-enable (WE) signal (see Figure 1-2 for WRITE_FIRST operation [1]).

Other BRAM configurations determine the sequence in which read and write operations are executed, and these configurations are known as NO_CHANGE, READ_FIRST, and WRITE_FIRST [1]. We utilized the BRAM in its default WRITE_FIRST configuration in our proposed design. The default mode is used when it is desired that recently written data is loaded into the output port (see Figure 1-2 [1]). For instance, the commencement of a write operation at a particular edge of the applied clock will initiate the loading of the stored data at the output port in the subsequent cycle of the same clock edge. Consequently, to properly read any recently written data, the read operation should occur after a complete cycle of the clock from when the write operation has been initiated. Under the conditions of violating the recommended timing parameters, random data may be generated because of the non-compliant timing that triggers internal metastability. Metastability does no physical harm to the BRAM cells. Other device parameters remain unchanged regarding the violation of the read-write timing.

For our proposed PUF design, the BRAM has been configured in SDP mode. We have used the synchronous mode to guarantee prompt instability when the BRAM is operated under non-compliant conditions. We have developed a dedicated driver circuit for the PUF module to enforce the BRAM operating under violated timing conditions. This driver circuit induces nondeterministic behavior by asserting the read and write enable signals simultaneously for the same address within a single clock cycle, intentionally violating the read-during-write timing constraints specified in the vendor documentation.

4 Proposed SRAM-based PUF

The block diagram of the proposed design is shown in Fig. 2. Our proposed PUF design leverages a novel overflow-based approach to generate CRPs. The core of the PUF relies on a pair of cross-coupled blocks of RAMs as shown in Fig. 2.

An initial challenge value is written to BRAM-0, triggering a sequence of overflow operations that modify both BRAM contents. The overflow operations introduce non-linear dependencies between the input challenge and the output response derived from the combined state of both BRAMs. The overflow diagram illustrates the PUF's operation. The write enable signals of both BRAMs are controlled to update their values, further shaping the response selectively. Starting from an idle state, a new challenge is written to BRAM-0 with regular write operation. It is followed by inverting the challenge and writing it again on the same address, while the PUF controller creates a conflict between both operations. Hence, the response of BRAM-0 is loaded onto the output ports and passed to the second BRAM (BRAM-1). BRAM-1 undergoes the same operations as BRAM-0, but instead of using the challenge as the input source, the response generated from BRAM-0 is used. This operation triggers overflows that propagate to BRAM-1, modifying its content, and leading to the generation of the final response of the received challenge (response of BRAM-0). The final response is generated by combining the upper and lower bits of the BRAM-1 ports (A & B). Both BRAMs erase any stored data sequence written in the address after obtaining their responses.

The unstable behavior in BRAM is observed when there is a timing conflict during the interval between write and read operations, according to the official documentation of the FPGA device [1]. With this information available, we have designed a special driver module that generates timing conflict between the BRAM read and write operations. Algorithm 1 encapsulates the operational flow of the proposed Cross-Coupled Read Invert-Write

Just Accepted

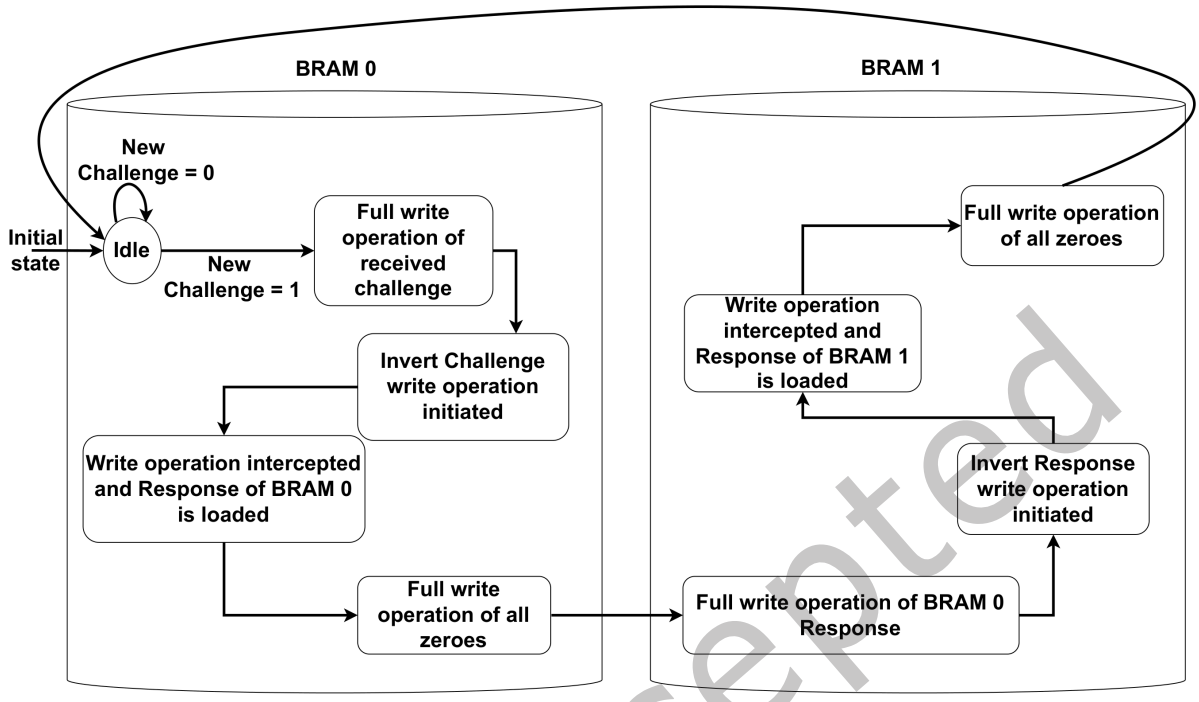

Fig. 2. Algorithmic state diagram of the proposed CC-RIWC PUF.

Just Accepted

Collision (CC-RIWC) PUF. The proposed PUF design harnesses the dynamic behavior induced by read-write collisions in two BRAMs within an FPGA. The algorithm unfolds in synchronization with the clock cycle's falling edge, orchestrating intricate operations on the two BRAMs (indexed as 0 and 1) based on the state transitions. In particular, the algorithm employs a sophisticated interplay of address generation and data manipulation to induce and capture the metastable states essential for PUF functionality. The utilization of BRAMs in a cross-coupled fashion adds a layer of complexity and richness to the generated CRPs, enhancing the security and unpredictability of the PUF.

The intricacies of algorithmic implementation involve careful handling of addresses, data transfers, and state transitions between the two BRAMs. Specifically, the algorithm checks for state transitions during each clock cycle and triggers the necessary read and write operations accordingly. The interleaved operations on the BRAMs, involving both input and output manipulations, result in the deliberate generation of metastable states. These states, captured in the form of CRPs, contribute to the unique and unpredictable characteristics of the CC-RIWC PUF. The algorithmic details provided serve as a comprehensive guide for understanding the underlying mechanism of this novel PUF design.

The selection of the next address in our CCRAM PUF is mathematically grounded in the need for entropy-driven feedback. Rather than employing a linear or arbitrary sequence, we utilize the output bits of the previous metastable result (Resp_{B0} or Resp_{B1}) to directly modulate the BRAM addressing registers. This design choice transforms the BRAM's internal address space into an entropy-dependent state machine. By coupling the response of one BRAM to the address input of the next, we ensure that the system state evolves as a function of the underlying silicon physical variations, effectively preventing the prediction of address sequences by an adversary.

Algorithm 1 Autonomous CRP Generation via Cross-Coupled BRAMs

Require: Input Challenge: C_i (32-bit), Number of CRPs: CRP_{Count}
Ensure: Sequence of Responses: R_{seq}
$Resp_{B1} \leftarrow 0$ ▷ BRAM initial value (0 by default)
for ($i = 0;\ i < CRP_{Count};\ i$++) **do**
  Phase 1: BRAM-0 Metastability Induction
  $Addr_{B0}[13:5] \leftarrow Resp_{B1}[14:6]$ ▷ Extract dynamic address
  $Addr_{B0}[4:0] \leftarrow 0$
  $Data_{B0} \leftarrow C_i[31:0]$
  TriggerCollision($BRAM_0$, $Addr_{B0}$, $Data_{B0}$)
  $Resp_{B0} \leftarrow$ Read $BRAM_0$ output (metastable result)
  Phase 2: Cross-Coupling (Feedback) ▷ Use BRAM-0 response to drive BRAM-1 autonomously
  $Addr_{B1}[13:5] \leftarrow Resp_{B0}[14:6]$ ▷ Extract dynamic address
  $Addr_{B1}[4:0] \leftarrow 0$
  $Data_{B1_Lower} \leftarrow Resp_{B0}[31:0]$ ▷ Lower 32 bits
  $Data_{B1_Upper} \leftarrow Resp_{B0}[35:32]$ ▷ Upper 4 bits (formerly parity)
  $CombinedData_{B1} \leftarrow \{Data_{B1_Upper}, Data_{B1_Lower}\}$
  Phase 3: BRAM-1 Metastability Induction
  TriggerCollision($BRAM_1$, $Addr_{B1}$, $CombinedData_{B1}$)
  $Resp_{B1} \leftarrow$ Read $BRAM_1$ output (metastable result)
  Phase 4: Output Generation & Update
  $FinalResponse \leftarrow Resp_{B1}$
  Store $FinalResponse$ in R_{seq}
end for
return R_{seq}

function TriggerCollision($BRAM$, $Addr$, $Data$)
  Cycle T1: Write $Data$ to $BRAM[Addr]$
  Cycle T2: Write ~ $Data$ (Inverted) to $BRAM[Addr]$
  Simultaneously: Read $BRAM[Addr]$ ▷ Causes Read-Write Conflict
end function

Just Accepted

The total number of clock cycles for one CRP generation is 12 cycles, 6 for each BRAM. The first and last two cycles are for regular write operations, whereas the third and fourth cycles are when the collision between write and read operations occurs. The final response is 36-bits in size, which is generated from a 32-bit challenge. We utilize the 4 parity bits that are normally used for error detection as standard storage bits to maximize the data width of the responses, treating the width as a raw 36-bit word. The address bit width for both BRAMs is 14-bits. In our implementation, the 9 most significant bits (MSBs) of the address for each BRAM are determined by the random data generated during the fourth clock cycle of the opposing BRAM, while the 5 least significant bits (LSBs) are tied to zero. This configuration stems from the BRAMs being operated in SDP mode with a 36-bit data width, which effectively aligns the memory into 32-bit (plus 4-bit parity) words. Thus, address increments occur in steps of 32, rendering the 5 LSBs redundant for word-level addressing. The address of BRAM-0 is determined by the output data from BRAM-1 at the fourth cycle and vice versa, increasing the complexity of the address sequence and reducing predictability compared to linear addressing. The design offers an output CRP space of 2^{36}, but it provides flexibility, as it can be customized for a more extended bit width, bringing the output space to at least 2^{72} if the CRPs are concatenated. The only drawback is the increased latency, which will take at least twice as long to generate the final response. Although we utilized only AMD FPGA devices and their integrated BRAMs in this study, our proposed design is principally applicable to any FPGA platform with a comparable BRAM module.

5 Hardware Design of the Data Acquisition System

A hardware prototype and a data acquisition system were necessary to validate the CC-RIWC PUF under real-world conditions. In the development of the first PUF prototype we used Verilog HDL to model the circuit to generate the intended behavior. However, using unconventional operating conditions and the lack of metastable behavior in vendor-provided BRAM models prevented the direct observation of the expected unstable behavior

in HDL simulations. Given the limitations in the simulation of unstable BRAM behavior, we have implemented a hardware design, as it allows direct and accurate observation of the PUF's performance.

A hardware-in-the-loop approach became imperative for experimenting with the prototype PUF. It requires only a CC-RIWC controller module and a 36-Kbit BRAM (as of two 18-Kbit BRAMs) for the construction of our SRAM PUF circuit based on CC-RIWC on the FPGA device. According to post implementation report from Vivado, our design used only 44 slices, 95 look-up tables (LUTs), 202 flip-flops (FFs), and one BRAM on an AMD XC7A35T-1CPG236C FPGA device, which means that only a small amount of the available slices are utilized. The utilization of FPGA hardware resources and performance metrics, including slices and PUF throughput, are summarized in Table 1.

Table 1. Comparison of the proposed PUF with other designs focusing on resource utilization and throughput

Work	**PUF Type**	**Hardware Platform**	**Resource Utilization**	**Throughput (Mbps)**
Swaroopa [50]	Butterfly	Nexys A7	256 slice REG	NA
Yue [58]	SRAM	Nexys DDR4	146 slices	NA
Chen [9]	SRAM	Cyclone-II	16 × 100-bits array	598.1
Avaroğlu [5]	RO	Cyclone-IV	128 RO	47.68
Yamamoto [56]	Latch	Spartan-3E	256 slices	192.7
Hisashi [22]	Logic LSI	Virtex-4	580 slices	12.5
Yan [57]	RO	Kintex-7	365 slices	0.625
Cherkaoui [12]	RO	Cyclone-II	> 255 slices	133
Dofe [18]	Dynamic	Artix-7	55 slices + 3 slice REG	NA
Li [29]	SRAM	Virtex-II Pro	369 slices	803
CC-RIWC PUF	SRAM	Artix-7	1 BRAM + 44 slices	900

The analysis of Table 1 reveals that our design exhibits a compact footprint, demonstrating high efficiency in both resource utilization and throughput compared to various PUF types documented in the literature. The two 18-Kbit BRAMs were configured in the SDP mode, featuring a 36-bit data port width comprising 32-bits of data and 4-bits for parity. Unlike previous implementation constrained by the narrower data paths of TDP configuration [13], transitioning to SDP mode allows us to leverage this maximum 36-bit width. This architectural optimization significantly increases the entropy density captured during read-write collisions. Both ports A and B were utilized for concurrent write and read operations. The operation of the CC-RAM controller adhered to Algorithm 1, orchestrating each cycle. Consequently, the CC-RAM PUF generated a 36-bit response for a 32-bit input challenge during each operational cycle. The BRAM ports were synchronized by a clock signal to ensure precision in timing control and maintain consistent unstable behavior.

AMD provides a softcore RISC processor known as MicroBlaze, custom-made for use in its FPGA devices, offering C/C++-based software control over its peripherals [2]. The use of a MicroBlaze processor yields a significant advantage in facilitating continuous challenge transmission, streamlining the automation of data acquisition, and expediting the process while enhancing the ease of debugging and validation. A custom softcore MicroBlaze processor-based system was configured using 16 KB of local memory and 100 MHz clock frequency. Integrating the MicroBlaze eases the control of hardware, data transfer, and its acquisition, enabling seamless collection of CRPs from the CC-RIWC-based SRAM PUF. The SRAM PUF is accessible through memory-mapped registers within the processor's address space. The integration includes module customization with AXI peripherals, facilitating seamless integration with the MicroBlaze soft-core processor.

Fig. 3 illustrates the integration of our custom CC-RIWC-based SRAM PUF design with MicroBlaze and its peripherals. Data transfer from the MicroBlaze processor to the computer is performed using the UART peripheral.

Just Accepted

In our design, the BRAMs are configured in SDP mode; consequently, the Write Enable (WEA) inputs associated with TDP functionality are tied to logic 0. Furthermore, all unused control registers and input ports are grounded to ensure a stable and predictable operating state.

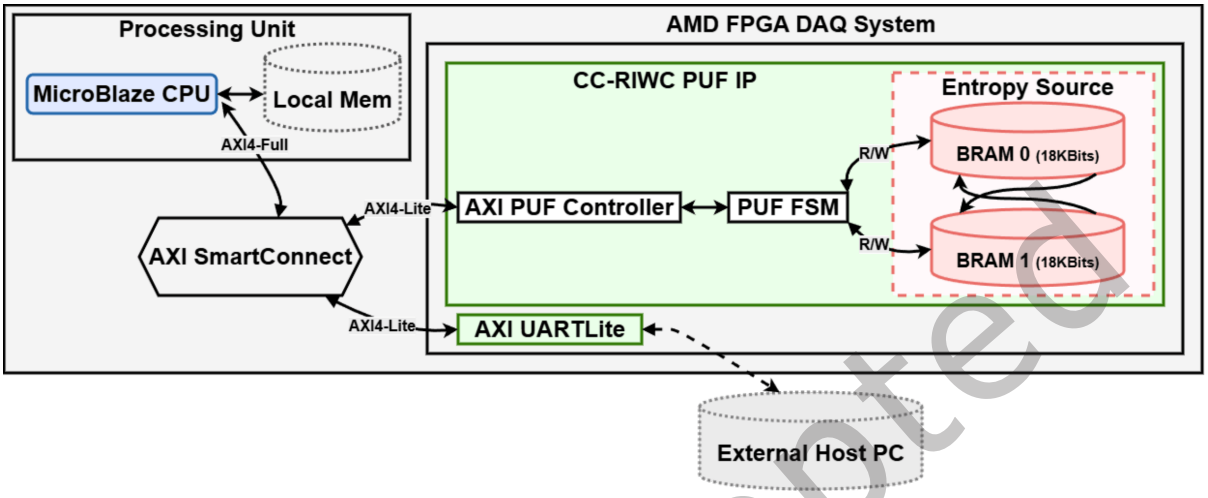

Fig. 3. Hardware Data Acquisition (DAQ) system diagram of the CC-RIWC-based PUF. The diagram illustrates the integration of the cross-coupled BRAM primitive with a MicroBlaze-based controller with the AXI-interconnect.

We have used the AMD Vitis Unified Software Platform to build the MicroBlaze firmware. This firmware, implemented as a concise C code, can initiate challenge writes, collect the corresponding responses, and continuously transmit the data to the computer via the serial port. Following hardware implementation and connection through AMD Vitis allowed the execution of our custom data acquisition firmware. From the computer side, we have verified and validated the proper control of the CC-RIWC-based SRAM PUF, and acquisition of a comprehensive CRP dataset. A substantial volume of CRP data is required for a thorough performance evaluation from a single device and across multiple devices. This is used to analyze variance and compute PUF metrics. Consequently, the identical MicroBlaze-based design was deployed on eight distinct BASYS-3 FPGA development boards that come with a XC7A35T-1CPG236C Artix-7 family device.

The same bitstream file created for the MicroBlaze-based design was used to program the eight FPGA boards. This approach was taken to maintain consistent placement of the BRAMs used, ensuring a uniform basis for the data acquisition process. At the end, for each of the eight FPGA devices used, a dataset consisting of one million 32-bit CRPs was collected for performance evaluation.

6 Performance Evaluation and Statistical Characterization

With the aim of effectively assessing our CC-RIWC-based SRAM PUF, we used a data collection-based tailored system to administer the challenges and capture their corresponding responses. The vast state space reaching up to 2^{32} CRPs posed a substantial evaluation challenge. We successfully gathered one million 32-bit CRPs from each FPGA device, recognizing the inherent variability in semiconductor devices due to uncontrollable factors during chip fabrication. Because the total challenge space of 2^{32} is too large for exhaustive evaluation, we utilize a randomly selected subset of CRPs. To ensure statistical significance, we targeted a minimum of 99% confidence levels. For a binomial distribution (where response bits evaluate to 0 or 1), a sample size of $n = 16,590$ is required to achieve a 1% margin of error (see Eq. 8.24) [35]. Our baseline dataset of 1,000,000 CRPs significantly exceeds this

Just Accepted

requirement, yielding a margin of error of approximately 0.13%. The objective is to ensure that PUFs exhibit the necessary characteristics to be qualified for cryptographic applications. A diverse range of analyses is imperative for a comprehensive performance evaluation, as different metrics are needed for different performance aspects.

The following run tests present widely accepted metrics employed in PUF evaluations, ensuring that the CC-RIWC-based SRAM PUF meets the required criteria for robust and secure cryptographic applications. However, our experiments did not encompass physical attacks, such as side-channel methods or probing techniques, nor did they involve analyses or tests based on machine learning or AI modeling attacks.

6.1 Dynamic CRP Space Expansion via State-Dependent Chaining

To evaluate the CRP space scalability of the proposed architecture, we investigated the PUF's state-dependent behavior. Because the entropy-driven addressing mechanism described in Section 4 utilizes the metastable response of the previous cycle to drive the subsequent address, the PUF possesses an inherent memory of recent operations resembling a time series. Applying a sequence of distinct challenges without asserting a reset effectively merges their responses into a single extended response. This mechanism provides a robust pathway for the dynamic expansion of the CRP space. Under normal conditions where 2^{32} CRP space is desired, the system may assert the reset followed by a controlled state transition.

Table 2. Representative evolution of the most reliable response for a fixed target challenge ($0xDEAD1234$) under varying sequence lengths k. MR_{900} represents the most reliable one out of 900 generated responses in the same k. *(Note: Hexadecimal values are illustrative examples.)*

	$Challenge_{target}$	**Example Response for Challenge = 0xDEAD1234**
Baseline measurement	$MR_{900}(Response_{target})$ [**0xABCD12345**]	0xABCD12345
k=1	$[discard(Response_{k=1}), MR_{900}(Response_{target})]$ [~~0x4587EFCDA~~, **0xEDF6789AB**]	0xEDF6789AB
k=2	$[discard(Response_{k=1}), discard(Response_{k=2}), MR_{900}(Response_{target})]$ [~~0x9584DDEF1~~, ~~0x64587AB0D~~, **0x0248ADEF5**]	0x0248ADEF5
⋮	⋮	⋮
k=29	$[discard(Response_{k=1}), \ldots, discard(Response_{k=29}), MR_{900}(Response_{target})]$ [~~0x54682D5AC~~, ..., ~~0x9875AC023~~, **0xFF5544889**]	0xFF5544889

To validate this concept, we conducted a targeted experiment on a selected FPGA instance using the following procedure:

- **Soft Reset Definition:** Before establishing baseline measurements, an internal logic reset was triggered. This soft reset instantaneously clears the data acquisition state machine and internal registers without requiring a full power-cycle or bitstream reconfiguration.
- **Baseline Measurement:** A fixed 32-bit target challenge was applied immediately following the reset. This process was repeated 900 times to establish the stable baseline response and the natural noise floor of the BRAM fabric.

Just Accepted

- **Sequence Chaining:** The same target challenge was applied but preceded by a sequence of k distinct pre-challenges, where k was increased from 1 to 29. The responses to the pre-challenges were discarded, and only the final response to the target challenge was recorded. This sequence was repeated 900 times for each step k while omitting the soft reset between chained operations. For each of the 30 attempts, the responses were selected from the 900 samples, representing the most common (i.e. the most reliable) response of the iteration.

As demonstrated in Table 2, the terminal response for the challenge 0xDEAD1234 is not a static mapping but a function of the preceding sequence length k. For instance, the baseline response (0xABCD12345) diverges immediately to 0xEDF6789AB when preceded by a single pre-challenge ($k = 1$), even though the targeted challenge remains unchanged. This behavior highlights the memory effect of the CC-RIWC architecture, where the metastable outcomes of the k pre-challenges act as an internalized, entropy-driven pre-conditioner for the target challenge. This ensures that an adversary cannot predict the output based solely on the current challenge, as the response is linked to the temporal history of the hardware's internal state transitions.

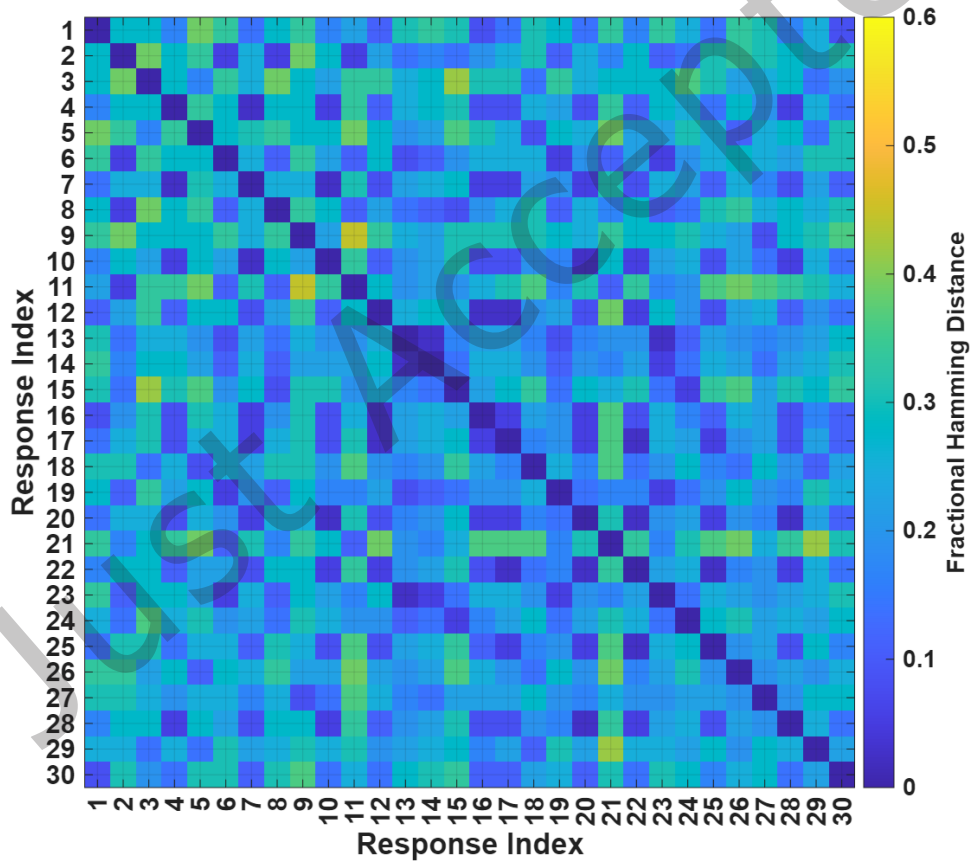

Fig. 4. Fractional Hamming Distance (FHD) heatmap comparing reliable responses generated by a fixed target challenge under different sequence lengths (k). The high decorrelation (non-zero FHD) illustrates CRP state dependency.

Just Accepted

As illustrated in the Fractional Hamming Distance (FHD) heatmap in Fig. 4, the experimental results reveal the impact of state-dependent chaining on the response space. To generate this figure, we picked the most reliable response for each state applied to each of the 30 experimental iterations (comprising the baseline and sequence lengths $k = 1$ to 29). The resulting heatmap compares these 30 dominant responses against one another. While a strictly stateless weak PUF would yield a heatmap dominated by near-zero distances indicating identical responses regardless of the preceding operations, our architecture demonstrates a high degree of decorrelation. The presence of non-zero FHD values across the matrix confirms that each unique sequence length k drives the cross-coupled BRAMs into a distinct internal state, forcing the fixed target challenge to map to unique regions of the entropy space.

This behavior provides empirical evidence for on-demand CRP space expansion. By sequentially chaining challenges, the effective challenge-response mapping becomes deeply contextual, allowing the system to logically construct a challenge space that extends well beyond the baseline 2^{32} limit. The diversity observed in the heatmap proves that this process is non-linear: the internal state transitions ensure that the final response is a function of the entire challenge history rather than a deterministic mapping of the current input. Consequently, the CC-RIWC PUF functions as a stateful hardware primitive, providing the high-entropy requirements necessary for scalable authentication systems.

This state-dependent behavior introduces a practical method for on-demand CRP space expansion. By sequentially chaining challenges, the effective response length multiplies, allowing the system to logically construct combinations well beyond the baseline 2^{32} CRP space. Importantly, this process is not merely additive: applying challenges A and B sequentially yields a final response that differs fundamentally from simply appending their independent responses. This supports the claim that the CC-RIWC PUF's behavior is deeply contextual and cannot be reduced to a deterministic mapping, highlighting its potential for scalable CRP space based authentication systems.

To clarify the operational characteristics of dynamic CRP expansion, it is necessary to define the physical mechanism of state retention and its implications for system performance. As the sequence progresses, the contextual state is maintained through the BRAM address registers rather than the response registers themselves. Specifically, before the 36-bit response values (Resp_{B0} and Resp_{B1}) are cleared, a specific portion of their generated bits is extracted to update the address registers of both BRAMs. Because these BRAM address registers are intentionally not reset between the k chained operations, they preserve the dynamic routing state that dictates the exact addresses where the subsequent CRP is generated. From a performance perspective, this stateful chaining introduces a deterministic trade-off. Because the hardware executes the chain sequentially, the effective throughput scales inversely with the sequence length k, by generating a final response for a sequence of length k requires $k \times 12$ clock cycles. Furthermore, while the experimental validation in this study utilized 900 iterations per sequence length to rigorously characterize the statistical noise floor and prove state divergence, practical deployment does not require such extensive sampling. In a live authentication protocol, a chained sequence is evaluated only once. Any compounding bit-level instability introduced by the extended operational time is resolved by the same standard cryptographic post-processing (e.g., fuzzy extractors) required for the baseline SRAM PUF responses.

6.2 Evaluation of Reliability

Reliability is an essential evaluation metric for assessing the effectiveness of any PUF architecture. It reflects a device's ability to reliably generate the same output responses given identical input challenges despite changes in environmental conditions. A widely used method for assessing the reliability involves determining the average Hamming distance (HD) between outputs (responses) evoked by an identical challenge. The Hamming Distance within a chip serves as an essential metric to evaluate reliability and is mathematically expressed by Eq. 1.

Just Accepted

$$\mathrm{HD}_{\mathrm{INTRA}} = \frac{1}{m}\sum_{t=1}^{m}\frac{\mathrm{HD}(R_i, R'_{i,t})}{n} \times 100\% \tag{1}$$

Eq. 1 determines the average count of unstable response bits, m indicates the number of responses collected in total. The function $H_D()$ represents the Hamming distance calculated among the response R_i, gathered at the reference temperature of 26°C, and the corresponding response $R'_{i,T}$, obtained at temperature T, both originating from the same challenge and device. Each response comprises n bits, with a total length of 36-bits. Consequently, the reliability of the PUF is assessed as the ratio of consistent device responses under different environmental conditions [31]. The intra-chip Hamming distance continues to be one of the most widely used metrics for assessing PUF reliability.

$$\text{Reliability} = 100\% - \mathrm{HD}_{\mathrm{INTRA}} \tag{2}$$

By Eq. 2, attaining higher levels of reliability requires reducing variations in the mean intra-chip Hamming distance. Reliability measurements are usually made by exposing the PUF to varying environmental conditions. Given that the performance of semiconductor devices is inherently temperature sensitive, their operational characteristics may deteriorate as a result of the natural aging of the devices. In our experimental setup, we assessed the reliability of device responses over a range of temperatures. Fluctuations in ambient temperature can adversely affect PUF responses, especially in weaker PUF designs. Before collecting CRPs, we used a custom-designed PID-controlled insulated chamber to keep the FPGA board at the desired temperature during experiments, as shown in Fig. 5.

Data collection was carried out at three separate ambient temperatures: 26°C (room temperature), −18°C (the lowest temperature achievable by the freezer) and 50°C. We evaluated the reliability of the PUF responses using Equations 1 and 2. The average reliability results, derived from the first test of the FPGA device using CRPs acquired from 2×10^5 (a constant random set of the 2^{32} possible set space of the CRPs), are presented in Table 3. It should be noted that while baseline metrics utilized a dataset of 1,000,000 CRPs, the reliability evaluation dataset was scaled to 200,000 CRPs. Collecting one million CRPs requires over 20 hours of continuous operation. Due to the safety constraints and equipment limitations of operating the PID-controlled thermal chamber at extreme temperatures (-18℃ and 50℃) without unattended oversight, the testing window was restricted to 4–5 hours. Notably, a sample of 200,000 CRPs still maintains a 99% confidence level with a margin of error under 0.3%, ensuring that the reliability results remain statistically robust.

FPGA-1 maintained an average reliability that exceeded 97.5% at −18°C and 98% at 50°C. The minimum reliability observed was 81.94%. These findings demonstrate a modest reliability reduction when the operating temperature deviates from the reference point of 26℃. Specifically, a decrease in reliability is observed both when the temperature drops to -18℃ (a 44℃ difference) and when it rises to 50℃ (a 24℃ difference). This indicates that the PUF's stability is highest at the characterization temperature, with environmental thermal stress in either direction inducing minor fluctuations in the response bits.

Fig. 6 illustrates the Cumulative Distribution Function (CDF) of the reliability for these two extreme temperature conditions. The CDF plot provides a clear visualization of the reliability distribution across the entire CRP population. For instance, the steep vertical rise near 98% indicates that the vast majority of CRPs exhibit very high stability. Even under extreme thermal stress, the distribution remains tightly clustered near the ideal 100% mark, with comprehensive statistical data provided in Table 3.

The histograms in Fig. 7 and Fig. 8 illustrate the reliability distribution of the first FPGA tested across the measured temperature range, exhibiting a clear Gaussian-like profile. On average, the reliability of FPGA-1 decreases by approximately 2.5% as the temperature deviates from the reference. The vertical lines in both histogram plots represent the overall average reliability of the rest of the devices tested. Compared to previous

Just Accepted

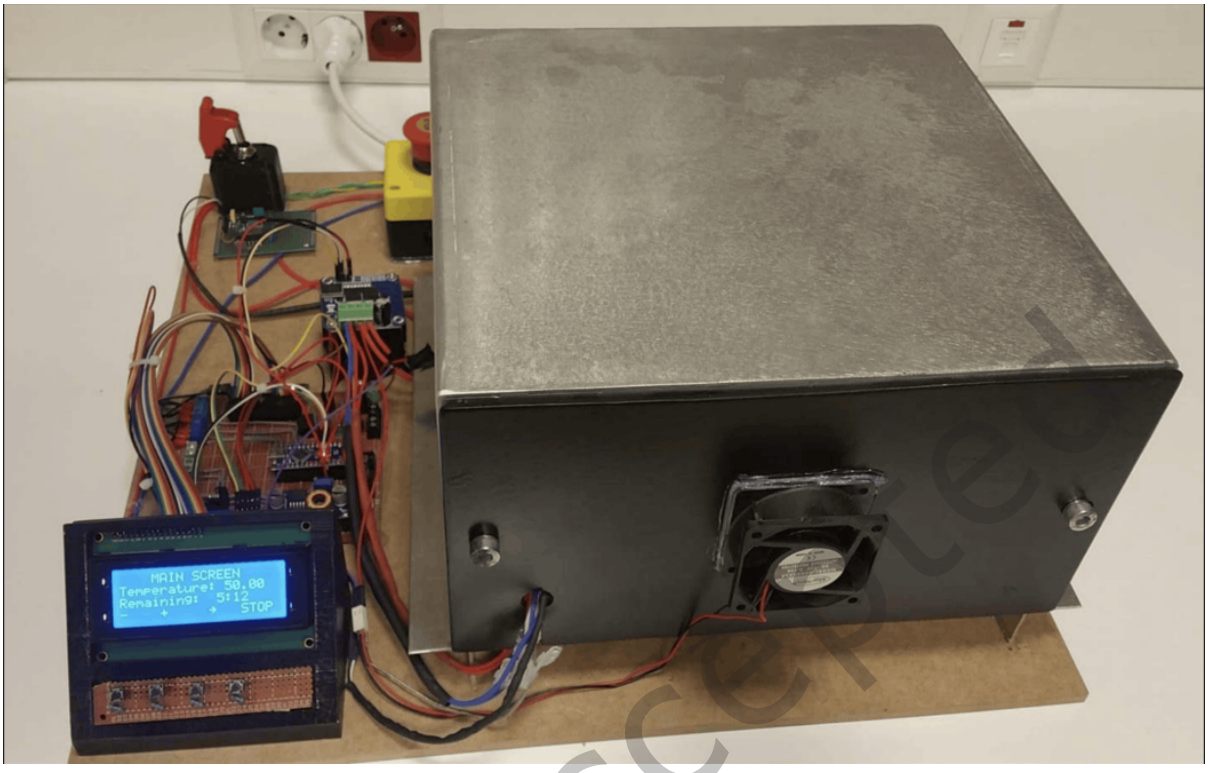

Fig. 5. Experimental setup for thermal reliability testing. The photograph shows the in-house developed PID-controlled insulated chamber used to maintain the FPGA temperature constant. This setup ensures that the CRP collection process occurs under stable environmental conditions, allowing for an accurate assessment of the PUF's thermal sensitivity and reliability.

Table 3. Reliability test results for first tested FPGA device at −18°C and 50°C temperatures.

Reliability (%)	**Min**	**Avg**	**Max**
FPGA-1 @ -18°C	81.94	97.52	100
FPGA-1 @ 50°C	83.33	98.37	100

work, such as [52], which documented an intrachip Hamming distance of 1.8% in their SRAM PUF design, equating to a reliability rate of 98.2%, the reliability values obtained here are promising.

6.3 Evaluation of Uniqueness

As a metric, uniqueness quantifies the dissimilarity between individual PUFs deployed on different FPGAs [7]. The calculation of uniqueness relies on assessing the distinction in the generated responses of identical challenges and conditions of PUF circuits employed on FPGA boards. The distinction is measured by applying the inter-chip hamming distance. We employed eight FPGA devices of the same model for our uniqueness evaluation, capturing responses for an identical set of challenges from the CC-RIWC SRAM PUFs employed on all FPGA devices. The uniqueness calculation used Eq. 3 as presented in [31].

Just Accepted

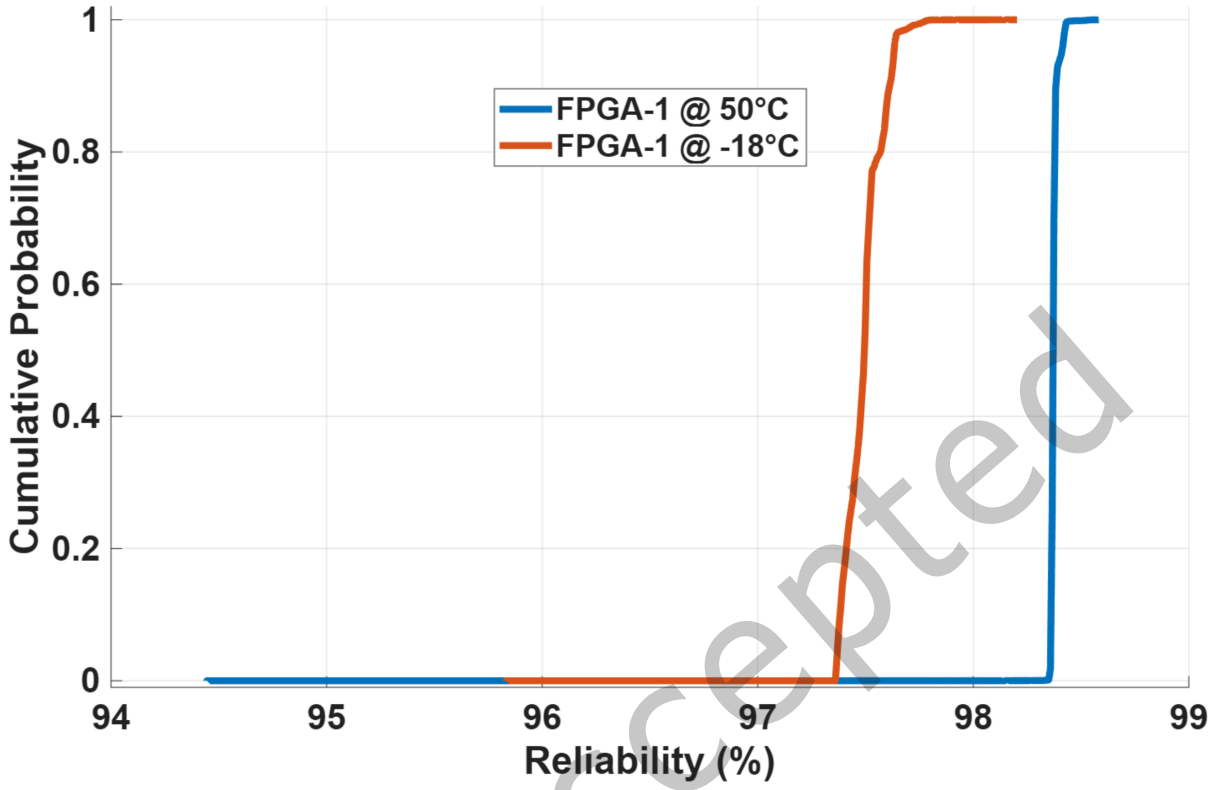

Fig. 6. Cumulative Distribution Function (CDF) of the intra-device reliability at -18℃ and 50℃. The plot illustrates the probability distribution of the reliability scores, showing that over 90% of the responses maintain a reliability level greater than 95% even under extreme temperature variations.

Just Accepted

$$\text{Uniqueness} = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{\text{HD}(R_i, R_j)}{n} \times 100\% \tag{3}$$

The uniqueness metric is expressed by the Hamming distance between responses R_i and R_j, where R_i and R_j are derived from i and j FPGA devices, respectively. Here, k indicates the total number of FPGA devices tested, while n denotes 36, the data width of the response. The test results for the uniqueness metric for eight FPGAs of the Artix-7 family are presented in Table 4.

Table 4. Uniqueness analysis results.

	Uniqueness (%)
Min	25.99
Avg	41.45
Max	52.48

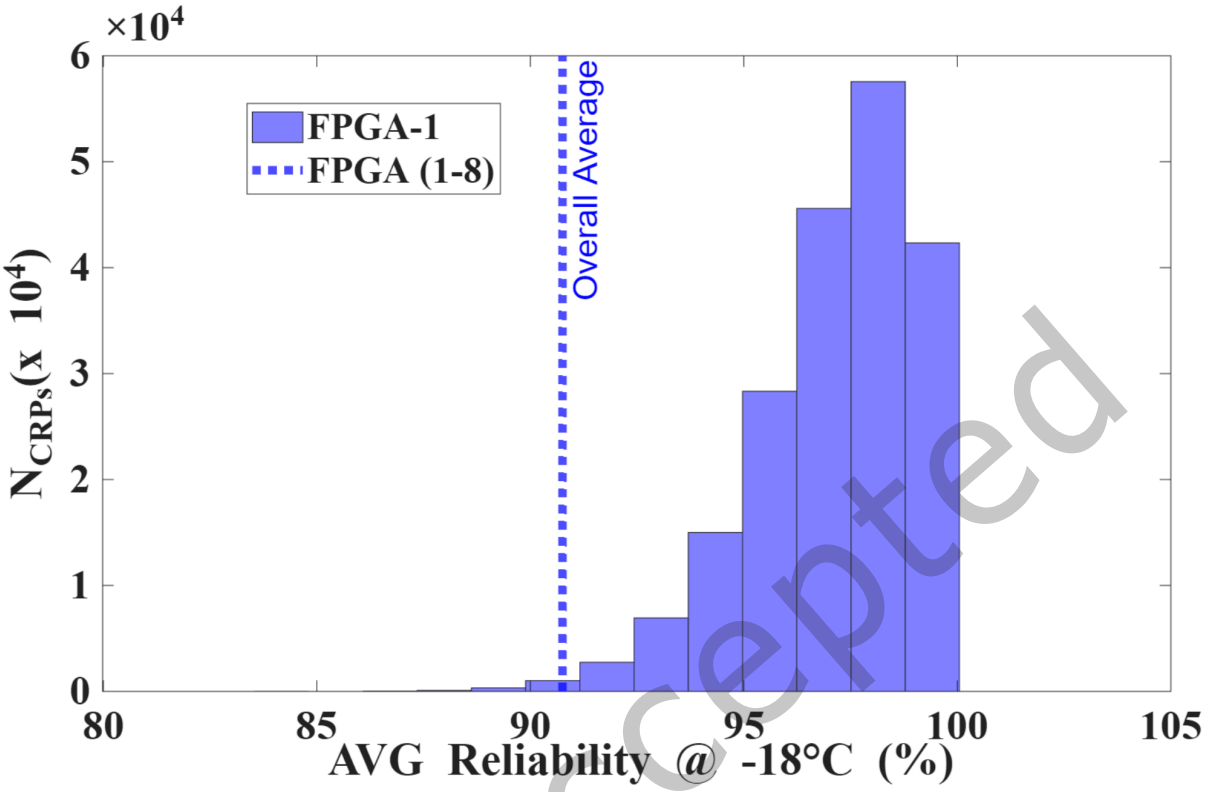

Fig. 7. Distribution of intra-device reliability for FPGA-1 at −18°C. The vertical blue line indicates the overall CRPs average across all tested FPGAs, highlighting the stability of the BRAM-based entropy at low temperature.

Just Accepted

As depicted in Table 4, the CC-RIWC PUF demonstrates an average uniqueness level of 41.45%. Although slightly lower than some conventional SRAM-based PUFs, it remains within an acceptable and functional range for raw hardware primitives. This signifies that each device is distinctly identified, as these values portray the difference of bits for the same type of response between 8 separate FPGA development boards when subjected to the same environmental and operating conditions, as well as same input challenges. In essence, there are distinct 36-bit responses gathered from each of the 8 FPGA devices, providing ample distinctiveness that uniquely identifies every FPGA device. The response distribution exhibits a Gaussian pattern with closely clustered mean values, as illustrated in Fig. 9. This performance is on par with several state-of-the-art SRAM-based PUF implementations documented in recent literature, which typically report uniqueness values ranging from 41% to 49.85% [39, 46]. Unlike earlier or less stable architectures that may exhibit significant global bias resulting in uncommonly low uniqueness levels, our entropy-driven addressing mechanism ensures a stochastic and balanced response distribution across different hardware instances. The reported values of maximum and minimum uniqueness are reported values for the single response, which means that these are the minimum and maximum registered uniqueness for a single response for each of randomly selected 10^6 CRPs collected.

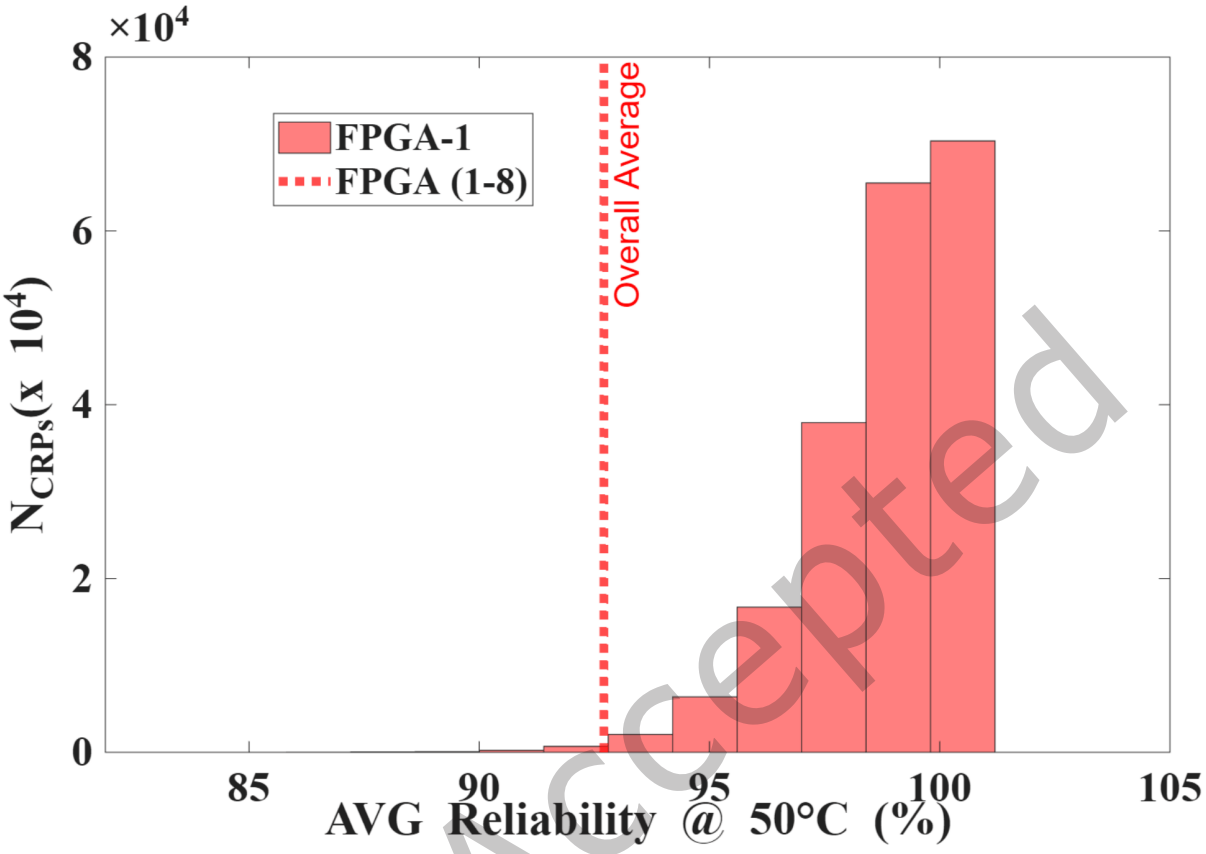

Fig. 8. Distribution of intra-device reliability for FPGA-1 at 50°C. The vertical red line indicates the overall CRPs average across all tested FPGAs, highlighting the stability of the BRAM-based entropy at high temperature.

Just Accepted

6.4 Evaluation of Bit-aliasing

Bit-aliasing pertains to imbalanced zeros and the ones distribution of the response vector of the PUF among different devices. The occurrence of bit-aliasing can lead to the generation of nearly identical PUF responses across distinct devices, which is undesirable output. This evaluation metric assesses the PUF and its tendency to produce distinguishable responses [31]. The percentage Hamming weight (HW) of the response R_i for the nth bit is $r_{i,n}$ and is examined in the k number of devices, as described by Eq. 4:

$$(\text{Bit} - \text{aliasing})_n = \frac{1}{k}\sum_{i=1}^{k} r_{i,n} \times 100\%. \tag{4}$$

For PUFs implemented on FPGAs, achieving a balanced distribution of ones and zeros (50% Hamming weight) for each bit across the test devices is crucial for optimizing the bit-aliasing metric. Deviations from this ideal level can result in PUF responses exhibiting nearly identical values across different FPGA devices, posing differentiation challenges. In our analysis, we experimentally computed the average distribution of 36-bit histograms; this was done on the eight FPGA devices and for each response bit, as illustrated in Fig. 10. We calculated the mean

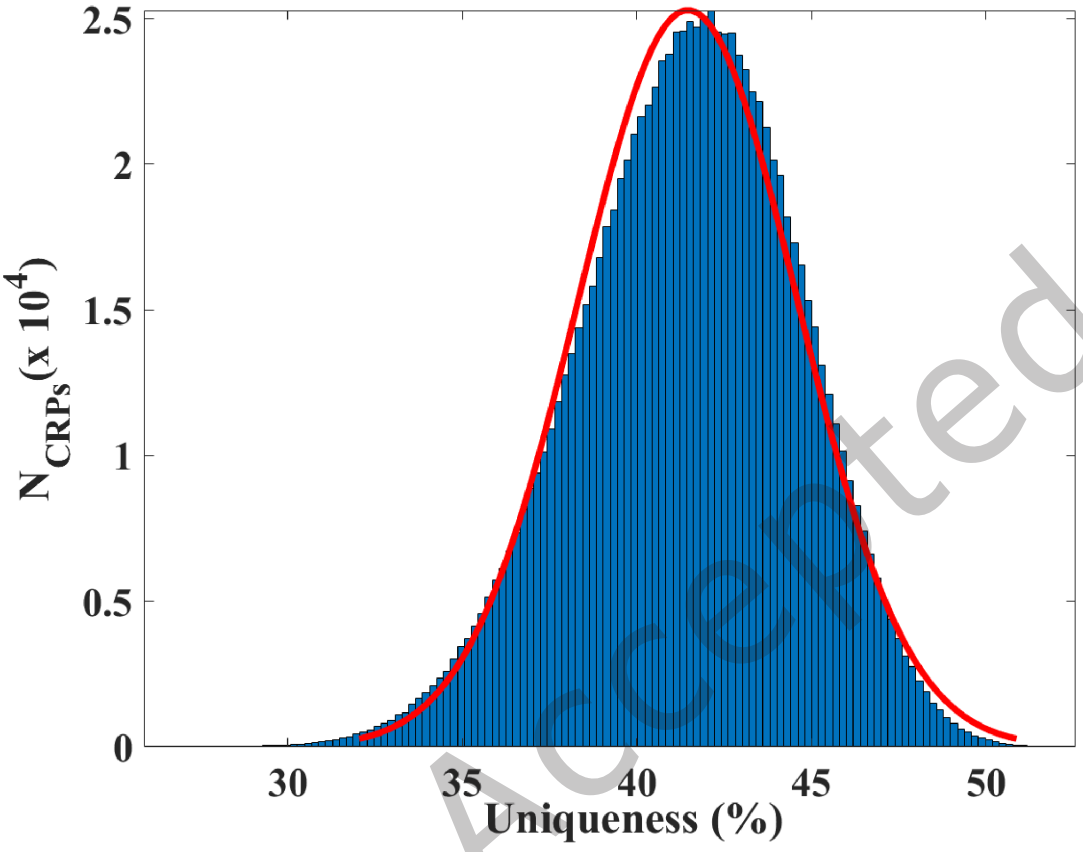

Fig. 9. Uniqueness analysis based on the Inter-device Hamming Distance (HD) distribution. The histogram represents the frequency of HD values calculated from CRPs collected from eight AMD Artix-7 FPGAs. The distribution is centered at a mean (μ) of 41.45%, indicating strong spatial decorrelation between devices. The shift from the ideal 50% is attributed to localized BRAM bias within the FPGA fabric, a common characteristic in SRAM-based primitives.

Just Accepted

values of bit-aliasing of the Gaussian-distributed histograms, resulting in an average value of 34.02%. Table 5 also presents a small portion of the CRP dataset bit-aliasing results per bit index.

The average bit-aliasing of 34.02% indicates a noticeable global bias inherent to the BRAM fabric, demonstrating a preference for the '0' state during collision resolutions. Since raw PUF responses with this level of bias do not possess full cryptographic entropy, they cannot be used directly as cryptographic keys. In practical authentication and key generation systems, these raw responses must be passed through standard post-processing stages, such as fuzzy extractors to compress the bitstream and ensure a uniform 50% entropy distribution suitable for secure protocols.

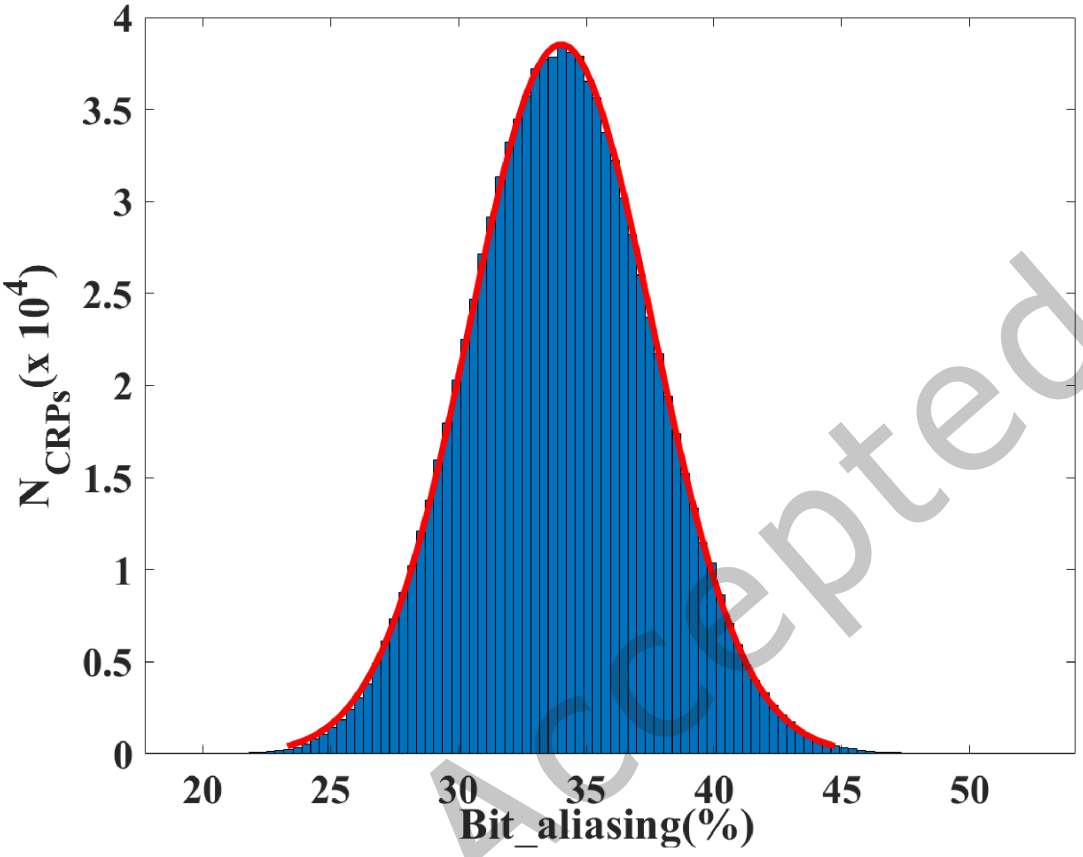

Fig. 10. Bit-aliasing distribution of the CC-RIWC PUF responses. The histogram illustrates the bit-wise bias across the eight tested FPGAs.

Just Accepted

6.5 Evaluation of Uniformity

$$(\text{Uniformity})_i = \frac{1}{n}\sum_{j=1}^{n} r_{i,j} \times 100\% \tag{5}$$

The uniformity metric evaluates the distribution in every response vector with respect to ones and zeros. This examination calculates the responses' percentage HW. The formula of the uniformity metric computation, which is denoted as $(Uniformity)_i$, is given by Eq. 5. Here, n represents the number of bits within the response $r_{i,j}$ generated by device i, where j indicating the bit's position in the response. Ideally, for highest entropy and random behavior, the percentage of HW should be 50%. However, achieving this ideal in practice may be challenging due to process variation-related biases. We have calculated the percentage HW value for each device's response of eight FPGA devices. Table 6 presents our findings, which are listed as the minimum and maximum uniformity registered for the eight FPGA devices and the best single-device average uniformity recorded among the eight devices. As indicated, the best average uniformity recorded for a single FPGA device was 53.04%. While the

Table 5. A small portion of the full bit-aliasing results for each response of the 10^6 CRPs and the average bit-aliasing (per bit index)

Bit Index	Bit-aliasing Response 1 (%)	Bit-aliasing Response 2 (%)	Bit-aliasing Response 3 (%)	...	Bit-aliasing Response 999,999 (%)	Bit-aliasing Response 1,000,000 (%)	Average Bit-aliasing (%)
1	50.00	50.00	25.00	...	50.00	37.50	**42.09**
2	12.50	62.50	12.50	...	50.00	50.00	**32.30**
3	37.50	0	37.50	...	50.00	12.50	**35.12**
...	...	...	...	...	...	...	...
35	25.00	50.00	12.50	...	0	12.50	**25.48**
36	75.00	25.00	50.00	...	62.50	50.00	**47.28**

average uniformity across the device population remains stable, a minimum uniformity value of 4.86% was recorded. This low outlier is attributed to a localized high-degree of manufacturing asymmetry in the specific BRAM, where the cross-coupled inverters exhibit a strong dominant state during the induced collision. Such extreme cases of bias are typical in raw, unconditioned intrinsic PUF responses.

This result goes along with state-of-the-art SRAM and arbiter-based PUF designs. Specifically, the proposed CC-RIWC PUF exhibits a uniformity (53.04%) that is comparable to the 52.6% in [39], 51% in [20], while outperforming the 46.81% uniformity reported for the arbiter PUF in [18].

Table 6. Uniformity analysis results.

	Uniformity (%)
Mean of Min	4.86
Best Avg	53.04
Mean of Max	71.88

Fig. 11 illustrates the distributions of the average uniformity levels calculated. As is obvious from Fig. 11, FPGA-5 has the closest mean distribution to the ideal mean value of 50% among the eight FPGAs. FPGA-7 and FPGA-8 also have fairly good average distributions of around 40%. While several other FPGA devices exhibit an average bias of around 30%, showing a tendency to produce more zeros than ones. It is important to note that this bias is not an artifact of the algorithm initializing the registers to zero; rather, during the induced read-write collision, the BRAM hardware resolves the metastable state based on its inherent physical asymmetries (such as bitline capacitance and default pre-charge routing), which in this specific FPGA fabric globally favors the '0' state. This hardware-induced bias is a well-known characteristic of raw SRAM-based primitives [33]. As discussed in Section 6.4, this hardware-induced bias dictates that the CC-RIWC PUF should serve as a raw entropy source, requiring subsequent cryptographic conditioning before the responses are utilized for device authentication. As demonstrated in Table 7, the proposed design achieves competitive results with respect to uniqueness and reliability. While bit-aliasing and uniformity are excluded from the general comparison table due to sparse reporting in referenced works, the individual analyses in Section 6.4 and Section 6.5 confirm that the CC-RIWC PUF remains comparable to, or surpasses, the performance of both conventional SRAM PUFs and strong-type PUFs.

Just Accepted

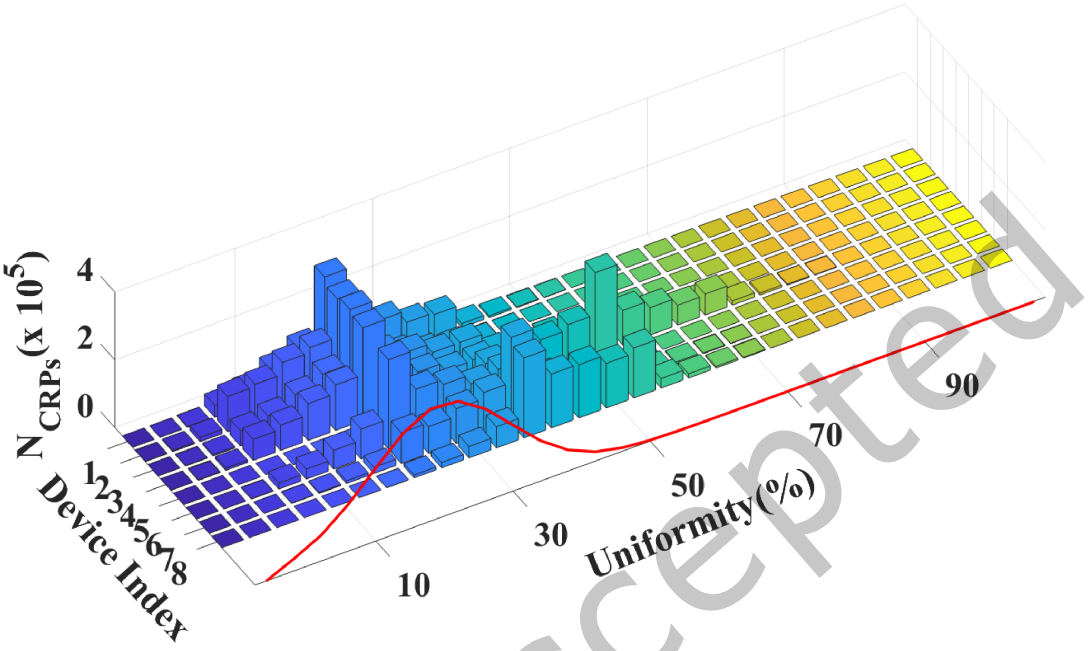

Fig. 11. Distribution of average uniformity levels across the eight tested FPGAs. Each data point represents the mean Hamming Weight (HW) of the PUF responses for a single device.

Table 7. Performance comparison with other works (* indicates ASIC implementations)

Work	Type	Uniqueness (%)	Reliability (%)
Dehghanzadeh [16]	SRAM*	49.37	97.5
Lin [30]	Arbiter*	38.9	NA
Sungyong [38]	SRAM*	49.57	97-98
Song [46]	SRAM*	49.85	98.08
Prasad [39]	SRAM*	41.2 - 45.1	94.264
Van der Leest [53]	Butterfly*	36	89
Gu [20]	Arbiter	48.52	96.46
Yan [57]	RO	NA	91
Ridley [42]	SRAM	NA	90.488
Dofe [18]	Arbiter	18.05	98
Usmani[52]	SRAM	NA	98.2
CC-RIWC PUF	SRAM	41.45	97.52 - 98.37

To contextualize the performance of the proposed CC-RIWC PUF, it is essential to distinguish between dependent PUF cores that requires external driver circuit and standalone PUF design. As detailed in Table 8, our

Just Accepted

previous work [13] achieved a significantly higher throughput of 6400 Mbps with lower resource utilization (17 slices). However, that design operated as a dependent PUF primitive, requiring an external driver circuit such as micro-controller with an RNG to drive the address ports of the BRAMs for applying challenges. When the hardware overhead of the required external circuitry is factored in, the true system-level footprint is considerably larger in terms of area.

In contrast, the proposed CC-RIWC design integrates the entropy-generation logic directly into the PUF core itself via the cross-coupled feedback loop thus eliminating the need for an external driver circuit. The observed resource utilization of 44 slices includes both the entropy source and the addressing logic integrated. While the feedback mechanism inherently limits the maximum frequency resulting in a throughput of 900 Mbps, this trade-off yields a security primitive that functions standalone without external circuitry.

Furthermore, a throughput of 900 Mbps is more than sufficient for the state-of-the-art cryptographic key generation and device authentication applications. For example, generating a 2048-bit RSA key or a 256-bit AES key occurs in mere microseconds, rendering the throughput difference negligible for practical security protocols. Thus, the reduction in raw throughput is a justified architectural trade-off for the gain in system level independence and the elimination of potential attack surfaces associated with external driving circuitry.

Table 8. Detailed comparison with our previously proposed study in [13]

Work	Cicek [13]	**CC-RIWC PUF**
PUF Type	SRAM	SRAM
Hardware Platform	Artix-7	Artix-7
Resource Utilization	**1 BRAM + 17 slices**	1 BRAM + 44 slices
Throughput (Mbps)	**6400**	900
Uniqueness (%)	37.36	**41.45**
Bit-aliasing (%)	**46.89**	34.02
Uniformity (%)	53.12	**53.04**
Reliability (%)	**98.92**	97.52 - 98.37
Addressing Mode	External Random	**Self-Generated**

7 Conclusion

This work presents a novel SRAM PUF architecture that addresses the limitations of conventional FPGA-based implementations by eliminating dependence on the initial state of SRAM cells at power-on. The proposed design utilizes a custom metastability-inducing circuit based on cross-coupled BRAMs, activated through regular read and write collisions. We integrated the design as an AXI4 peripheral within a MicroBlaze-based system for efficient data acquisition and performance evaluation. Experimental results, based on one million CRPs per device, demonstrate that the proposed PUF achieves strong uniqueness, competitive reliability, and a high throughput of 900 Mbps, with minimal hardware overhead – only 44 slices and one BRAM. Although bit-aliasing and uniformity values were moderate, the overall performance surpasses many weak-type PUFs reported in the literature. These results highlight the potential of our approach for lightweight, high-throughput, and integration-friendly PUF designs on commercial FPGAs.

Noting that our research scope and evaluation criteria did not include physical attacks, nor did they encompass metrics specific to machine learning or artificial intelligence, future work may focus on enhancing resistance to physical and side-channel attacks, as well as exploring applicability in secure key generation and authentication for edge and embedded systems.

Just Accepted

References

[1] 2019. *UG473: Xilinx 7 Series FPGAs Memory Resources User Guide.* Xilinx Incorporation, San Jose, CA. https://docs.amd.com/v/u/en-US/ug473_7Series_Memory_Resources

[2] 2019. *UG984: Xilinx MicroBlaze Processor Guide.* Xilinx Incorporation, San Jose, CA. https://www.xilinx.com/support/documentation/sw_manuals/xilinx2019_1/ug984-vivado-microblaze-ref.pdf

[3] Zain Ul Abideen, Rui Wang, Tiago Diadami Perez, Geert-Jan Schrijen, and Samuel Pagliarini. 2024. Impact of Orientation on the Bias of SRAM-Based PUFs. *IEEE Design & Test* 41, 3 (2024), 14–20. doi:10.1109/MDAT.2023.3322621

[4] Aref Al-Tamimi, Shawkat Ali, Yuan Cao, and Amine Bermak. 2024. Threshold Voltage based Dual Memristor Crossbar PUF. *AEU - International Journal of Electronics and Communications* 175 (2024), 155012. doi:10.1016/j.aeue.2023.155012

[5] Erdinç Avaroğlu. 2020. The implementation of ring oscillator based PUF designs in Field Programmable Gate Arrays using of different challenge. *Physica A: Statistical Mechanics and its Applications* 546 (2020), 124291. doi:10.1016/j.physa.2020.124291

[6] Mario Barbareschi, Franco Cirillo, and Christian Esposito. 2024. SRAM-PUF Authentication Schemes Empowered with Blockchain on Resource-Constrained Microcontrollers. In *2024 IEEE 27th International Symposium on Real-Time Distributed Computing (ISORC)*. 1–10. doi:10.1109/ISORC61049.2024.10551352

[7] Halak Basel. 2018. *Physically unclonable functions From Basic Design Principles to Advanced Hardware Security Applications.* Springer. 250 pages. doi:10.1145/1973009.1973111

[8] Georg T. Becker. 2015. On the Pitfalls of Using Arbiter-PUFs as Building Blocks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 34, 8 (2015), 1295–1307. doi:10.1109/TCAD.2015.2427259

[9] Shuai Chen, Bing Li, and Cenjun Zhou. 2018. FPGA implementation of SRAM PUFs based cryptographically secure pseudo-random number generator. *Microprocessors and Microsystems* 59 (2018), 57–68. doi:10.1016/j.micpro.2018.02.001

[10] Zhuojun Chen, Ming Wu, Yifeng Zhou, Renlong Li, Jinzhe Tan, and Ding Ding. 2023. PUF-CIM: SRAM-Based Compute-In-Memory With Zero Bit-Error-Rate Physical Unclonable Function for Lightweight Secure Edge Computing. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 31, 8 (2023), 1234–1247. doi:10.1109/TVLSI.2023.3277517

[11] Zhuojun Chen, Wenhao Yang, Jinghang Chen, Zujun Wang, and Ding Ding. 2024. Improving Radiation Reliability of SRAM-Based Physical Unclonable Function With Self-Healing and Pre-Irradiation Masking Techniques. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 32, 2 (2024), 372–381. doi:10.1109/TVLSI.2023.3332010

[12] Abdelkarim Cherkaoui, Viktor Fischer, Laurent Fesquet, and Alain Aubert. 2013. A Very High Speed True Random Number Generator with Entropy Assessment. In *Cryptographic Hardware and Embedded Systems – CHES 2013 15th International Workshop on Cryptographic Hardware and Embedded Systems – CHES 2013 (Security and Cryptology, Vol. 8086)*. Springer, Santa Barbara, California, United States, 179–196.

[13] Ihsan Cicek and Ahmad Al Khas. 2023. A new read–write collision-based SRAM PUF implemented on Xilinx FPGAs. *Journal of Cryptographic Engineering* 13, 1 (April 2023), 19–36. doi:10.1007/s13389-021-00281-8

[14] M. Cortez, A. Dargar, S. Hamdioui, and G. Schrijen. 2012. Modeling SRAM start-up behavior for Physical Unclonable Functions. In *2012 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*. 1–6.

[15] Abdolreza Darabi, Ebrahim Abiri, and Mohammad Reza Salehi. 2024. Data-dependent half-select free GSRAM cell with word line write-assist and built-in read buffer schemes for use in PUFs-based IoT devices. *AEU - International Journal of Electronics and Communications* 185 (2024), 155448. doi:10.1016/j.aeue.2024.155448

[16] Peyman Dehghanzadeh, Soumyajit Mandal, and Swarup Bhunia. 2025. MBM PUF: A Multi-Bit Memory-Based Physical Unclonable Function. *IEEE Transactions on Circuits and Systems I: Regular Papers* (2025), 1–14. doi:10.1109/TCSI.2025.3526884

[17] Riccardo Della Sala, Davide Bellizia, and Giuseppe Scotti. 2024. Unveiling the True Power of the Latched Ring Oscillator for a Unified PUF and TRNG Architecture. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 32, 12 (2024), 2403–2407. doi:10.1109/TVLSI.2024.3448503

[18] Jaya Dofe and Shailesh Rajput. 2024. Protecting against modeling attacks: design and analysis of lightweight dynamic physical unclonable function. *Cluster Computing* 28, 1 (Oct. 2024), 6. doi:10.1007/s10586-024-04736-5

[19] Susana Eiroa and Iluminada Baturone. 2011. An analysis of ring oscillator PUF behavior on FPGAs. In *2011 International Conference on Field-Programmable Technology*. 1–4. doi:10.1109/FPT.2011.6132673

[20] Chongyan Gu, Neil Hanley, and Máire O'neill. 2017. Improved Reliability of FPGA-Based PUF Identification Generator Design. *ACM Trans. Reconfigurable Technol. Syst.* 10, 3, Article 20 (May 2017), 23 pages. doi:10.1145/3053681

[21] Jorge Guajardo, Sandeep S. Kumar, Geert-Jan Schrijen, and Pim Tuyls. 2007. FPGA Intrinsic PUFs and Their Use for IP Protection. In *Cryptographic Hardware and Embedded Systems - CHES 2007*, Pascal Paillier and Ingrid Verbauwhede (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 63–80.

[22] Hisashi HATA and Shuichi ICHIKAWA. 2012. FPGA Implementation of Metastability-Based True Random Number Generator. *IEICE Transactions on Information and Systems* E95.D, 2 (2012), 426–436. doi:10.1587/transinf.E95.D.426

Just Accepted

[23] E. R. Henderson, J. M. Henderson, H. Shahoei, W. V. Oxford, E. C. Larson, D. L. MacFarlane, and M. A. Thornton. 2024. Designing a photonic physically unclonable function having resilience to machine learning attacks. In *Proc. SPIE 13028, Quantum Information Science, Sensing, and Computation XVI*, Vol. 13028. 130280B. doi:10.1117/12.3013126

[24] Charles Herder, Meng-Day Yu, Farinaz Koushanfar, and Srinivas Devadas. 2014. Physical Unclonable Functions and Applications: A Tutorial. *Proc. IEEE* 102, 8 (2014), 1126–1141. doi:10.1109/JPROC.2014.2320516

[25] Daniel E. Holcomb and Kevin Fu. 2014. Bitline PUF: Building Native Challenge-Response PUF Capability into Any SRAM. Cryptology ePrint Archive, Paper 2014/749. https://eprint.iacr.org/2014/749.

[26] Fei Hongming, Owen Millwood, Prosanta Gope, Jack Miskelly, and Biplab Sikdar. 2024. Attacking Delay-Based PUFs With Minimal Adversarial Knowledge. *IEEE Transactions on Information Forensics and Security* 19 (2024), 7259–7274. doi:10.1109/TIFS.2024.3433548

[27] Sandeep S. Kumar, Jorge Guajardo, Roel Maes, Geert-Jan Schrijen, and Pim Tuyls. 2008. Extended abstract: The butterfly PUF protecting IP on every FPGA. In *2008 IEEE International Workshop on Hardware-Oriented Security and Trust*. 67–70. doi:10.1109/HST.2008.4559053

[28] Kusum Lata and Linga Reddy Cenkeramaddi. 2023. FPGA-Based PUF Designs: A Comprehensive Review and Comparative Analysis. *Cryptography* 7, 4 (2023). doi:10.3390/cryptography7040055

[29] Dongfang Li, Zhaojun Lu, Xuecheng Zou, and Zhenglin Liu. 2015. PUFKEY: A High-Security and High-Throughput Hardware True Random Number Generator for Sensor Networks. *Sensors* 15, 10 (2015), 26251–26266. doi:10.3390/s151026251

[30] Lang Lin, Sudheendra Srivathsa, Dilip Kumar Krishnappa, Prasad Shabadi, and Wayne Burleson. 2012. Design and Validation of Arbiter-Based PUFs for Sub-45-nm Low-Power Security Applications. *IEEE Transactions on Information Forensics and Security* 7, 4 (2012), 1394–1403. doi:10.1109/TIFS.2012.2195174

[31] Abhranil Maiti, Vikash Gunreddy, and Patrick Schaumont. 2013. *A Systematic Method to Evaluate and Compare the Performance of Physical Unclonable Functions*. Springer New York, New York, NY, 245–267. doi:10.1007/978-1-4614-1362-2_11

[32] Honorio Martin, Pedro Peris-Lopez, Giorgio Di Natale, Mottaqiallah Taouil, and Said Hamdioui. 2019. Enhancing PUF Based Challenge–Response Sets by Exploiting Various Background Noise Configurations. *Electronics* 8, 2 (2019). doi:10.3390/electronics8020145

[33] Shayesteh Masoumian, Roel Maes, Rui Wang, Karthik Keni Yerriswamy, Geert-Jan Schrijen, Said Hamdioui, and Mottaqiallah Taouil. 2023. Modeling and Analysis of SRAM PUF Bias Patterns in 14nm and 7nm FinFET Technology Nodes. In *2023 IFIP/IEEE 31st International Conference on Very Large Scale Integration (VLSI-SoC)*. 1–6. doi:10.1109/VLSI-SoC57769.2023.10321895

[34] Shayesteh Masoumian, Georgios Selimis, Rui Wang, Geert-Jan Schrijen, Said Hamdioui, and Mottaqiallah Taouil. 2022. Reliability Analysis of FinFET-Based SRAM PUFs for 16nm, 14nm, and 7nm Technology Nodes. In *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 1189–1192. doi:10.23919/DATE54114.2022.9774735

[35] Douglas C. Montgomery and George C. Runger. 2018. *Applied Statistics and Probability for Engineers* (7th ed.). John Wiley & Sons, Hoboken, NJ. 187 pages.

[36] Jae-Won Nam, Ju-Hyeok Ahn, and Jong-Phil Hong. 2022. Compact SRAM-Based PUF Chip Employing Body Voltage Control Technique. *IEEE Access* 10 (2022), 22311–22319. doi:10.1109/ACCESS.2022.3153359

[37] Li Ni, Pengjun Wang, Yuejun Zhang, Xiangyu Li, Gang Li, Lin Ding, and Jiliang Zhang. 2024. SI PUF: An SRAM and Inverter-Based PUF With a Bit Error Rate of 0.0053% and 0.073/0.042 pJ/bit. *IEEE Transactions on Circuits and Systems II: Express Briefs* 71, 4 (2024), 2339–2343. doi:10.1109/TCSII.2023.3339296

[38] Sungyong Park, Minhyeok Jeong, Jaerok Kim, Donggyu Kim, and Yoonmyung Lee. 2024. A 6T-SRAM-Based Physically-Unclonable-Function With Low BER Through Automated Maximum Mismatch Detection. *IEEE Transactions on Circuits and Systems II: Express Briefs* (2024), 1–1. doi:10.1109/TCSII.2024.3359777

[39] Kailash Prasad, Neel Shah, Jinay Dagli, and Joycee Mekie. 2024. SDR-PUF: Sequence-Dependent Reconfigurable SRAM PUF with an Exponential CRP Space. In *2024 37th International Conference on VLSI Design and 2024 23rd International Conference on Embedded Systems (VLSID)*. 535–540. doi:10.1109/VLSID60093.2024.00095

[40] Matthias Probst, Lars Tebelmann, Moritz Wettermann, and Michael Pehl. 2024. Remote side-channel analysis of the loop PUF using a TDC-based voltage sensor. *Journal of Cryptographic Engineering* 15, 1 (Nov. 2024), 1. doi:10.1007/s13389-024-00366-0

[41] Vikash Kumar Rai, Somanath Tripathy, and Jimson Mathew. 2020. 2SPUF: Machine Learning Attack Resistant SRAM PUF. In *2020 Third ISEA Conference on Security and Privacy (ISEA-ISAP)*. 149–154. doi:10.1109/ISEA-ISAP49340.2020.235013

[42] Trevor Ridley, Julius Andrade, Cristian Luna, and Hayssam El-Razouk. 2024. Smart Car Temperature Monitoring System Using SRAM-Based PUF Sensor. In *2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC)*. 0555–0563. doi:10.1109/CCWC60891.2024.10427764

[43] F.J. Rubio-Barbero, F. de los Santos-Prieto, R. Castro-Lopez, E. Roca, and F.V. Fernandez. 2025. Harvesting random telegraph noise for true random number generation. *AEU - International Journal of Electronics and Communications* 196 (2025), 155801. doi:10.1016/j.aeue.2025.155801

[44] A. Santana-Andreo, P. Saraza-Canflanca, R. Castro-Lopez, E. Roca, and F.V. Fernandez. 2024. Reliability improvement of SRAM PUFs based on a detailed experimental study into the stochastic effects of aging. *AEU - International Journal of Electronics and Communications* 176 (2024), 155147. doi:10.1016/j.aeue.2024.155147

Under Accepted

[45] Niloufar Sayadi, Phuong Ha Nguyen, Marten van Dijk, and Chenglu Jin. 2025. Breaking XOR Arbiter PUFs With Chosen Challenge Attack. *IEEE Transactions on Information Forensics and Security* 20 (2025), 4971–4984. doi:10.1109/TIFS.2025.3566538

[46] Minte Song, Nan Liu, Shuaiyang Zhou, Zhengguang Wang, Zhanqiang Ru, Peng Ding, Wei Huang, and Helun Song. 2024. A Design of 2-Stage Voltage Ramp-Up SRAM Physical Unclonable Function. *Chinese Journal of Electronics* 33, 2 (2024), 371–379. doi:10.23919/cje.2022.00.406

[47] Kleber Stangherlin, Zhuanhao Wu, Hiren Patel, and Manoj Sachdev. 2023. Enhancing Strong PUF Security With Nonmonotonic Response Quantization. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 31, 1 (2023), 55–64. doi:10.1109/TVLSI.2022.3212271

[48] Franz-Josef Streit, Paul Krüger, Andreas Becher, Stefan Wildermann, and Jürgen Teich. 2021. Design and Evaluation of a Tunable PUF Architecture for FPGAs. *ACM Trans. Reconfigurable Technol. Syst.* 15, 1, Article 7 (Nov. 2021), 27 pages. doi:10.1145/3491237

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

[50] Sneha Swaroopa, Venkata Sreekanth Balijabudda, Rajat Subhra Chakraborty, and Indrajit Chakrabarti. 2025. PUFBind: PUF-Enabled Lightweight Program Binary Authentication for FPGA-based Embedded Systems. arXiv:2501.07868 [cs.CR] https://arxiv.org/abs/2501.07868

[51] Shahin Tajik, Heiko Lohrke, Jean-Pierre Seifert, and Christian Boit. 2017. On the Power of Optical Contactless Probing: Attacking Bitstream Encryption of FPGAs. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (Dallas, Texas, USA) *(CCS '17)*. Association for Computing Machinery, New York, NY, USA, 1661–1674. doi:10.1145/3133956.3134039

[52] Mohammad Usmani. 2018. *Applications Of Physical Unclonable Functions on ASICS and FPGAs*. M. Eng. thesis. University of Massachusetts Amherst.

[53] Vincent van der Leest, Geert-Jan Schrijen, Helena Handschuh, and Pim Tuyls. 2010. Hardware intrinsic security from D flip-flops. In *Proceedings of the Fifth ACM Workshop on Scalable Trusted Computing* (Chicago, Illinois, USA) *(STC '10)*. Association for Computing Machinery, New York, NY, USA, 53–62. doi:10.1145/1867635.1867644

[54] Alexander Wild and Tim Güneysu. 2014. Enabling SRAM-PUFs on Xilinx FPGAs. In *2014 24th International Conference on Field Programmable Logic and Applications (FPL)*. 1–4. doi:10.1109/FPL.2014.6927384

[55] Phillip Williams, Haytham Idriss, Mohammad Mohebbi, and Magdy Bayoumi. 2024. A Highly Secure Reconfigurable Memory-Based Strong PUF for Device Authentication in Internet of Things. In *SoutheastCon 2024*. 1333–1338. doi:10.1109/SoutheastCon52093.2024.10500202

[56] Dai Yamamoto, Kazuo Sakiyama, Mitsugu Iwamoto, Kazuo Ohta, Takao Ochiai, Masahiko Takenaka, and Kouichi Itoh. 2011. Uniqueness Enhancement of PUF Responses Based on the Locations of Random Outputting RS Latches. In *Cryptographic Hardware and Embedded Systems – CHES 2011*, Bart Preneel and Tsuyoshi Takagi (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 390–406.

[57] Wei Yan, Chenglu Jin, Fatemeh Tehranipoor, and John A. Chandy. 2017. Phase calibrated ring oscillator PUF design and implementation on FPGAs. In *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*. 1–8. doi:10.23919/FPL.2017.8056859

[58] Mengcheng Yue. 2024. A Two-Stage TMV SRAM PUF Preselection Method with Fewer ECC Resources. In *2024 IEEE 7th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Vol. 7. 528–533. doi:10.1109/IAEAC59436.2024.10503621

[59] Hu Zhang, Chenghua Wang, Chenggang Yan, Yijun Cui, Chongyan Gu, Maire O'Neill, and Weiqiang Liu. 2021. A Dynamic Highly Reliable SRAM-Based PUF Retaining Memory Function. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*. 1–5. doi:10.1109/ISCAS51556.2021.9401350

[60] Jiliang Zhang and Gang Qu. 2019. Recent Attacks and Defenses on FPGA-based Systems. *ACM Trans. Reconfigurable Technol. Syst.* 12, 3, Article 14 (Aug. 2019), 24 pages. doi:10.1145/3340557

Just Accepted