IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 October 2024, accepted 2 December 2024, date of publication 5 December 2024, date of current version 13 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3512419

APPLIED RESEARCH

A True Random Number Generator Based on Race Hazard and Jitter of Braided and Cross-Coupled Logic Gates Using FPGA

HOSSAM O. AHMED[1], (Senior Member, IEEE), DONGHOON KIM[2], (Senior Member, IEEE), AND WILLIAM J. BUCHANAN[3]

[1]College of Engineering and Technology, American University of the Middle East, Egaila 54200, Kuwait
[2]Department of Aerospace Engineering and Engineering Mechanics, University of Cincinnati, Cincinnati, OH 45221, USA
[3]School of Computing, Engineering and the Built Environment, Edinburgh Napier University, EH10 5DT Edinburgh, U.K.

Corresponding author: Hossam O. Ahmed (Hossam-omar@aum.edu.kw)

ABSTRACT In the contemporary digital landscape, security has become a vital element of our existence. The growing volume of sensitive information being stored and transmitted over networks necessitates the implementation of robust security measures. Cryptographic algorithms, which are critical for protecting user data privacy, rely on cryptographic keys to ensure data security. True Random Number Generators (TRNGs) are essential to numerous vital security applications. In this paper, we propose a novel Braided and Hybrid Cross-Coupled Entropy Source (B+HCCES) TRNG module. The proposed B+HCCES TRNG module generates random numbers based on the race hazard and jitter of braided and cross-coupled combinational logic gates. The B+HCCES architecture has been designed using VHDL, and the targeted Field-Programmable Gate Array (FPGA) is the Intel Cyclone V 5CGXFC9D6F27C7 chip. The B+HCCES module operates at a fixed sampling frequency of 300 MHz, generated by an embedded phase-locked loop. The B+HCCES module demonstrates an enhanced throughput of 3.33 times compared to the state-of-the-art, while still maintaining a comparably lightweight architecture. The experimental results demonstrate that the generated random sequence successfully passes the NIST SP800-90B and BSI AIS-31 tests.

INDEX TERMS True random number generator (TRNG), race hazard, ring oscillator, field programmable gate array (FPGA), jitter.

I. INTRODUCTION

True Random Number Generators (TRNGs) are important in various fields due to their ability to generate random sequences that are both unpredictable and non-reproducible. In security protocols, TRNGs generate robust encryption keys, enhancing data protection and preventing unauthorized access [1], [2]. Networking relies on TRNGs for secure data transmission, safeguarding information as it traverses interconnected systems [3]. The widespread importance of TRNGs highlights their crucial role in preserving the confidentiality, integrity, and availability of sensitive information in our increasingly digital world. Thus, many applications will benefit from integrating TRNGs to secure the data exchange processes such as mobile applications [4], aerospace and avionics [5], [6], and Internet of Things (IoT) applications [7] in the near future.

The prevailing standard for securing mobile systems involves storing a confidential key in nonvolatile memory, which ensures that the key remains intact even when the device is powered off. However, over the past two decades, the security of this method has been compromised by numerous advanced tampering techniques. Micro-probing, which involves physically accessing and manipulating the circuitry, can extract sensitive information directly from the device [8]. Focused Ion Beam (FIB) techniques allow for precise cutting and editing of circuits at the microscopic level, potentially exposing the stored keys [9]. Glitch attacks, which introduce

The associate editor coordinating the review of this manuscript and approving it for publication was Marco Anisetti.

faults into a system to cause unexpected behavior, can lead to the leakage of cryptographic keys [10]. Side-channel attacks exploit indirect information, such as power consumption or electromagnetic emissions, to infer the stored keys [11]. These sophisticated attacks have demonstrated significant vulnerabilities in the traditional approach of storing keys in cryptographic hardware devices, highlighting the need for more advanced security measures to protect sensitive information in mobile systems [9], [10].

In the domain of digital TRNG hardware implementation, the choice between FPGA (Field-Programmable Gate Array) and ASIC (Application-Specific Integrated Circuit) architectures necessitates a meticulous evaluation of several critical factors. FPGAs offer unparalleled flexibility and rapid prototyping capabilities, facilitating iterative development cycles and adaptability to evolving specifications [12], [13], [14], [15], [16], [17], [18]. However, this flexibility often results in higher power consumption and potentially longer latency compared to ASICs, which are custom-designed for optimized performance, power efficiency, and compactness. Conversely, ASIC development demands significant upfront investment and longer design cycles due to the intricacies of semiconductor fabrication [19], [20]. The selection process involves balancing considerations of development time, initial capital outlay, design flexibility, and the specific performance demands of the TRNG application. This deliberation guides practitioners and researchers toward choosing the platform that aligns best with their project's technical and economic objectives [21].

This paper presents a novel TRNG module named Braided and Hybrid Cross-Coupled Entropy Source (B+HCCES). This module introduces a unique approach to generating random numbers by utilizing race hazards and time jitter found in a novel mixture of braided and cross-coupled combinational Logic Gates (LGs). These physical phenomena are leveraged to extract entropy, ensuring the unpredictability and randomness of the generated sequences. The B+HCCES TRNG module offers a robust solution for applications requiring high-quality random numbers, particularly in cryptographic and security-sensitive contexts.

Moreover, this paper provides a detailed exploration of the design principles and operational mechanisms of the B+HCCES TRNG, highlighting its potential advantages over conventional entropy sources in terms of reliability and effectiveness.

The rest of this brief is organized as follows. In Section II, we discuss the related work. In Section III, the structure and working principle of the proposed B+HCCES architecture are described in detail. In Section IV, we introduce the FPGA implementation of B+HCCES TRNG and give the overall experimental architecture. NIST (National Institute of Standards and Technology) and AIS (Application notes and Interpretations of the security requirements for electronic Signatures)-31 test suite analysis and other experimental results are reported and discussed in Section V. In Section VI, we compare our proposed design with the state-of-the-art of TRNGs. Lastly, conclusions are drawn in Section VII.

II. RELATED WORK

The relationship between an entropy source and a TRNG is vital for maintaining the integrity and reliability of the random numbers produced. On one hand, an entropy source is a mechanism, whether physical or computational, responsible for producing randomness. It may utilize unpredictable physical occurrences (such as thermal fluctuations in electronic systems or radioactive decay) or computational techniques that gather and blend diverse data inputs (like mouse movements, keyboard timings, or variations in network traffic). On the other hand, a TRNG is a system or algorithm that leverages an entropy source to generate random numbers that exhibit statistical properties identical to true randomness. In contrast to Pseudo Random Number Generators (PRNGs), which utilize deterministic algorithms and seeds to produce sequences that mimic randomness but are inherently predictable, TRNGs depend on genuinely unpredictable entropy sources. Subsequently, the randomness quality of a TRNG is significantly influenced by the quality and rate of entropy from its source. If the entropy source lacks adequate randomness, or if it introduces biases or predictability, the output of the TRNG may show patterns or biases that undermine its randomness integrity. TRNGs can be generated using various methods, which can broadly be categorized into physical and computational approaches. The physical TRNG methods could be based on thermal noise, radioactive decay, photon arrival times, inductor-less bistable Josephson junction circuits, magnetic tunnel junctions, or chaotic systems [22], [23], [24], [25], [26]. The computational methods are based on both environmental noise and either hash functions or algorithmic approaches [26], [27], [28]. However, the power consumption and the throughput of generating these true random values are key factors depending on the targeted applications.

In this paper, we will depend on digital LGs within the same silicon fabric architecture of an FPGA chip. LGs themselves do not typically directly produce TRNGs because LGs are deterministic—they produce outputs based strictly on their inputs according to predefined rules (truth tables). However, it is possible to use the LGs to act as a TRNG module by intentionally creating instability in LGs and forcing them to operate under different uncertain conditions.

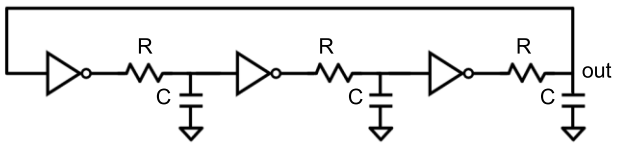


FIGURE 1. An example of an RO with three NOT gates.

The techniques that could lead to such phenomena are:

- **Ring Oscillations (ROs):** Creating a feedback loop where the output of a gate is fed back into its input can

cause oscillations. For example, connecting the output of a NOT gate back to its input will create a simple oscillator that continually switches states [29].

- **Race conditions from signal feedback:** In complex circuits, feedback loops can create race conditions where the outcome depends on the relative timing of the signals. This can make the circuit behave unpredictably [17], [18].
- **Race-hazards glitches and time jitter:** Designing circuits with hazards (unintended paths) that can cause glitches. For instance, in a combinational circuit, if there are different paths with different delays leading to the same output, it can cause temporary incorrect outputs [30].
- **Metastability glitches:** If the input changes at a time when the gate is not ready (e.g., during the transition period), it can enter a metastable state where the output is unpredictable for a short time [13].

Also, other techniques can be used, such as the setup and hold time violations, clock skew, noise and power supply variations, temperature variations, and fault injection [31], [32], [33]. In this paper, the main source of entropy in our proposed B+CCLGES module is based on the combination of the race-hazards glitches and the ROs.

A. RING OSCILLATORS

The RO is a type of oscillator circuit commonly used in digital integrated circuits to generate a continuous square wave output. As depicted in Fig. 1, it consists of an odd number of inverting stages (typically inverters) connected in a loop or ring configuration. Each inverter in the ring introduces a propagation delay (t_{pd}), which is described as

$$t_{pd} = k.C_{load}.R_{eq} \tag{1}$$

where k is a technology-dependent constant, C_{load} is the load capacitance per inverter, and R_{eq} is the equivalent resistance seen by each inverter.

The total delay (T) around the ring oscillator loop is the sum of the propagation delays of all the inverters. If there are (N) inverters in the ring, then:

$$T = N.t_{pd} \tag{2}$$

The frequency (f) of oscillation is inversely proportional to (T) and is given by

$$f = \frac{1}{T} = \frac{1}{N.t_{pd}} \tag{3}$$

B. CROSS-COUPLED LOGIC GATES

Cross-coupled (CC) LGs can indeed be used to construct random number generators, particularly those based on chaotic systems or feedback loops that exhibit sensitive dependence on initial conditions. CC-LGs refer to a class of logic circuits where the outputs of gates are fed back as inputs to other gates in a cyclical manner. One common example is the Set-Reset (SR) latch, which uses CC NOR, XOR, or NAND gates. CC-LGs, such as those in an SR latch, are essential for creating stable memory elements and sequential logic circuits.

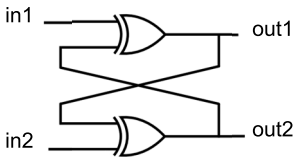


FIGURE 2. An example of CCX LGs.

They rely on feedback to maintain state, making them useful in applications where memory or state retention is required. Variations in manufacturing, temperature, or power supply can affect the delay of each inverter, leading to unpredictable variations in the oscillation frequency. This variability can be exploited to generate random numbers. As illustrated in Fig. 2, the CC XOR (CCX) is a logic circuit that can function as an oscillator and generate various race hazards. The CCX module has been explicitly explained in [2]. This cross-coupled XOR configuration enables the generation of varying output behaviors based on the input port values of *in1* and *in2*. Table 1 shows that the CCX has three distinct output modes: oscillation, stable, and unpredictable. When *in1*=*in2*=0, the CCX functions as two head-to-tail buffers. In this state, *out1* and *out2* hold the same unknown value, with a phase difference determined by the sum of the buffer delays and the wire delays. When *in1* is not equal to *in2*, the CCX is comparable to an inverter with a buffer attached to its head and tail.

In this configuration, *out1* and *out2* behave as a ring oscillator. When *in1*=*in2*=1, the CCX effectively operates as two inverters connected in a head-to-tail configuration. In this state, *out1* and *out2* maintain opposite values, with their phase difference determined by the cumulative delay of the equivalent circuit. Thus, the CCX is an intriguing module that can function as an oscillator and produce random outputs when the circuit is continuously fed with random input values.

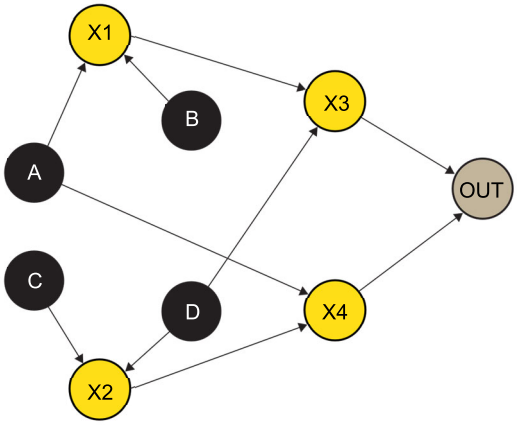


FIGURE 3. An example of a complex braided XOR logic network.

TABLE 1. The CCX output signal modes.

in1	in2	Output signal mode
0	0	Stable
0	1	Oscillation
1	0	Oscillation
1	1	Unpredictable

C. COMPLEX BRAIDED XOR LOGIC NETWORK

Braided XOR circuits are characterized by their interconnected XOR gates, often forming intricate patterns to achieve certain logical functions or provide diffusion in cryptographic algorithms. To understand the real need to create a braided XOR The CCX module has been logic network, we first compare a traditional XOR circuit and the braided XOR network. In a traditional XOR circuit, the signals typically follow a more straightforward path through a series of XOR gates. In this case, the propagation delay is generally more predictable because the signal path is less complex.

Algorithm 1 The Demonstration of the Sequential Processing of Inputs Through Multiple XOR Gates to Generate the Final Output

Initialization:
1: Set variables A, B, C, and D to represent the inputs.
Step 1: Compute intermediate values
2: Calculate X1 = A XOR B
3: Calculate X2 = B XOR C
4: Calculate X3 = C XOR D
5: Calculate X4 = A XOR D
Step 2: Compute final outputs
6: Calculate Y1 = X1 XOR X2
7: Calculate Y2 = X2 XOR X3
8: Calculate Y3 = X3 XOR X4
Step 3: Calculate the defuzzification output
9: Display or store the values of Y1, Y2, and Y3 as the final outputs.

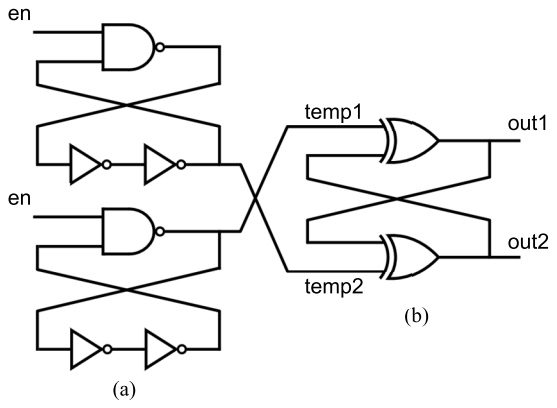


FIGURE 4. Proposed HCCLG module.

Additionally, the XOR gate contributes its intrinsic delay to the overall propagation delay. In contrast, in a braided XOR logic network, the signals may traverse multiple intertwined paths through various XOR gates. The interconnection of gates in a braided manner increases the complexity of the signal paths. Subsequently, these aspects lead to the propagation delay being more variable and potentially longer due to the increased number of gates and the complexity of the interconnections.

The complexity and interconnectivity of these circuits can significantly impact propagation delay due to:

- Number of gates: The more XOR gates a signal must pass through, the greater the cumulative delay.
- Interconnections: The complexity and length of interconnections between gates add to the delay.
- Gate loading: Each gate adds capacitive loading to the previous stage, increasing the delay.
- Signal integrity: Complex braiding can affect signal integrity, leading to potential timing issues and increased delay.

Fig. 3 presents an example of a single-output braided XOR logic network. The diagram features nodes for input variables (A, B, C, D) shown as black circles, intermediate XOR gates (X1, X2, X3, X4) represented as yellow circles, and the final output depicted as a gray circle. Arrows within the diagram indicate the signal pathways, illustrating the sequential processing of inputs through various XOR gates to produce the final output. Additionally, Algorithm 1 outlines the sequential operation of a distinct braided XOR logic network featuring multiple output signals. These examples of braided XOR logic networks illustrate the range of options available for implementing different diffusion techniques in cryptographic algorithms. However, it is vital to engage in comprehensive trial and error to determine the optimal circuit design that complies with the NIST SP800-90B and BSI AIS-31 standards.

III. PROPOSED B+HCCES ARCHITECTURE

In this work, we introduce a new TRNG architecture is based on CC-LGs combined with a complex braided XOR network module. This design aims to produce a robust entropy source capable of generating high-quality random sequences that meet the standards of NIST SPA800-90B and BSI AIS 20/31 specifications. The proposed B+HCCES TRNG module primarily relies on four sub-modules to generate high-quality random samples: two Hybrid CC-LG (HCCLG) modules, a Braided XOR LG (B-XOR-LG) module, a sampler, and a post-processing module.

A single HCCLG module consists of two stages, as shown in Fig. 4. The right-side stage comprises two ROs, which generate two noisy square pulses, *temp1,* and *temp2*, when the *en* signal is high. The left-side stage is a CCX module. The inputs to the CCX module are *temp1* and *temp2*. To increase the ambiguity of the output signals of the CCX module, *out1* and *out2*, we did not impose constraints on the *temp1* and *temp2* signals. Thus, the CCX module can operate in any of three different modes: unpredictable, oscillation, or stable,

depending on the instantaneous values of the input signals. The main function of the CCX module is to assist the ROs in enhancing the randomness of the samples generated by the HCCLG module.

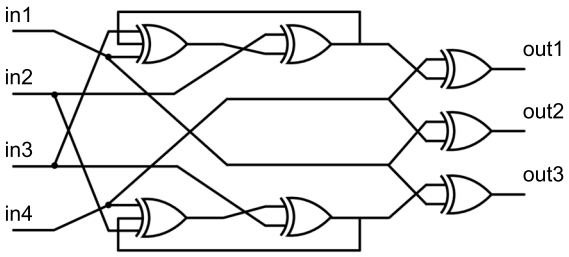


FIGURE 5. Proposed B-XOR-LG module.

The B-XOR-LG module receives four different output signals from the two preceding HCCLG modules, as illustrated in Fig. 5. These four inputs (*in1*, *in2*, *in3*, and *in4*) are then processed through the proposed B-XOR network, which heightens the race hazards at the three output signals (*out1*, *out2*, and *out3*) due to the multiple paths and the presence of two active feedback loops. This proposed B-XOR-LG module was developed through extensive trial and error until it successfully passed all NIST SP800-90B and BSI AIS-31 tests. Additionally, the two feedback signals between each pair of successive XORs in the intermediate stage significantly enhanced the confusion of the final output signal. The overall randomness of the final output was enhanced by using the B-XOR-LG module with three output signals, as the different jitter produced by these outputs will contribute to greater diversity in the subsequent sampling stage.

The third module of the B+HCCES is the sampler. In a TRNG, the sampler is a fundamental module for several reasons. It gathers raw data from various sources, ensuring that sufficient entropy is accumulated to generate true random numbers. This raw noise data typically requires preprocessing to eliminate biases and achieve a uniform distribution. The sampler is capable of executing these functions, converting raw noise into a format conducive to random number generation. Furthermore, as the raw data may display certain correlations, the sampler facilitates the de-correlation of samples to enhance the overall randomness. The post-processing stage encompasses a set of operations or techniques applied to the output data following its initial generation or collection, with the objective of improving its quality, usability, or adherence to specific standards.

In a TRNG, post-processing serves several purposes. One of its primary functions is to ensure that the random numbers exhibit a uniform distribution by eliminating any biases present in the raw data. This is necessary for producing genuinely random numbers that have an equal probability of occurrence, which is a fundamental requirement for numerous applications. Additionally, post-processing works to reduce or eliminate correlations between samples. Raw data from physical noise sources can sometimes exhibit correlations, which can compromise randomness. By decorrelating these samples, post-processing enhances the overall randomness and independence of the generated numbers. Furthermore, post-processing aims to increase the unpredictability and randomness of the data. This involves employing various techniques to extract maximum entropy from the noise source, ensuring that the random numbers are as unpredictable as possible. This unpredictability is important for applications in cryptography, security, and other fields where high-quality random numbers are necessary. Overall, post-processing in a TRNG is vital for refining the raw output into high-quality random numbers suitable for a wide range of critical applications.

IV. HARDWARE IMPLEMENTATION OF B+HCCES

The detailed hardware architecture of the proposed B+HCCES system is presented in Figure 6. This system is divided into two primary sub-modules. The first sub-module, known as the B+HCCES TRNG architecture, consists of two HCCLG modules. These modules share a common input signal labeled '*en*', which activates them to generate output pulses; otherwise, the output remains at logic 0.

To deliberately increase the propagation delay within the interconnections between modules, we designed a configuration where the four outputs from the two HCCLG modules are closely interconnected with the B-XOR-LG module. Specifically, the first and second output pins of the first HCCLG module are connected to the second and fourth input pins of the B-XOR-LG module, respectively. Similarly, the first and second output pins of the second HCCLG module are linked to the first and third input pins of the B-XOR-LG module. This interconnected configuration also contributes to the system's ability to introduce additional time jitter. The B-XOR-LG module outputs three signals, each of which is directly connected to an input pin of a D Flip-Flop (DFF) module. These DFFs collectively form the sampler module within the B+HCCES TRNG architecture. The sampler module plays a pivotal role in capturing the random values generated by the B-XOR-LG module. The operation of the sampler module is synchronized by a 300 MHz clock signal provided by an embedded Phase-Locked Loop (PLL) unit. This clocking mechanism ensures that the sampling process occurs at a consistent and precise rate, facilitating the accurate capture of random signals. The sampler module is critically important as it enables the generation of high-quality random numbers that adhere to statistical properties such as uniform distribution, independence, and unpredictability.

By directly interfacing with the outputs of the B-XOR-LG module, the sampler ensures that the inherent randomness of the signals is faithfully captured and converted into digital data. Furthermore, the sampler plays a pivotal role in minimizing biases and correlations that may arise during the analog-to-digital conversion process. This helps to preserve the integrity and reliability of the generated random numbers, making them suitable for a wide range of applications including cryptography, simulations, and statistical analyses.

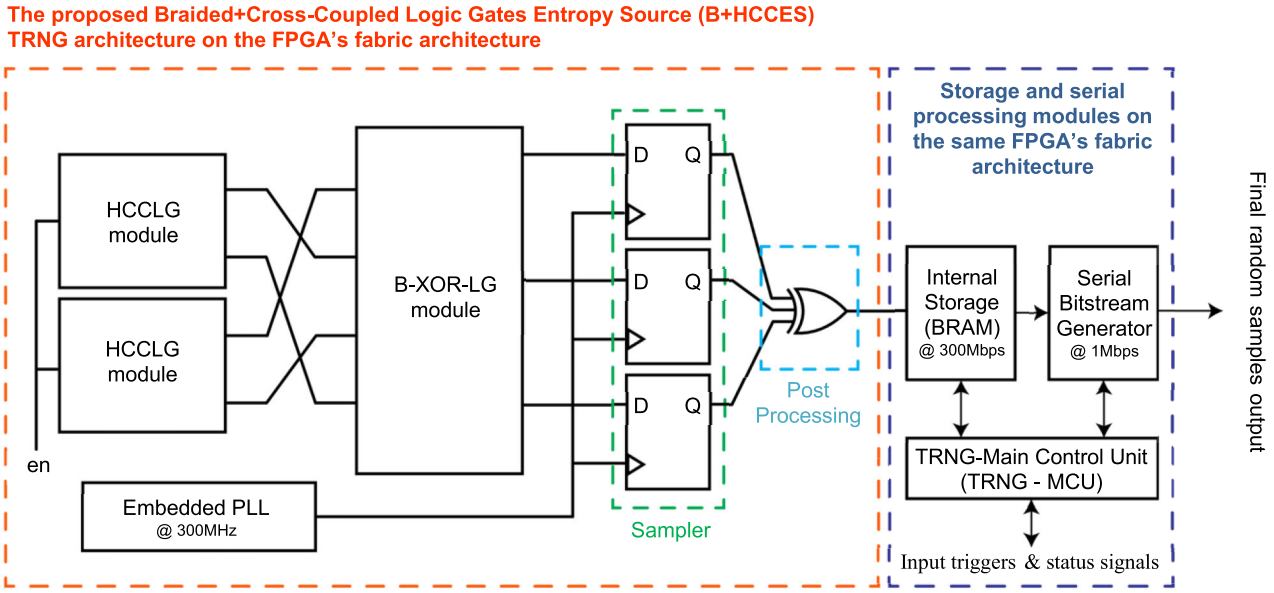


FIGURE 6. The detailed structure of the B+HCCES TRNG architecture with the storage control module, and the serial module.

TABLE 2. Summary of computational performance and power consumption for the proposed B+HCCES TRNG architecture with the storage control module, and serial module.

Targeted FPGA device name	INTEL 5CGXFC9D6F27C7 (low-end Cyclone V GT)
Total block memory bits (BRAM)	8,388,608 / 12,492,800 (67 %)
External memory usage	None
TRNG sampler frequency	300 MHz (from PLL)
Total used embedded PLLs	1 / 17 (6 %)
Number of logic elements	698 / 113,560 (< 1 %) ALMs
Total number of registers	335 registers
Estimated I/O thermal power dissipation	6.43 mW @23mm heat sink with 200 LFpM airflow + 12.5% toggle I/O signal rate
Estimated core static thermal power dissipation	533.50 mW
Total thermal power dissipation	539.93 mW

TABLE 3. Table 1. The summary of computational performance and power consumption for the proposed B+HCCES unit across the FPGA chip.

Combinational ALUT	23
Dedicated Flip-flops	3
ALMs needed [A-B+C] *	11.5
Combinational cell thermal power dissipation	0.02 mW
Clock enable block thermal power dissipation	1.79 mW
Register cell thermal power dissipation	0.04 mW
I/O thermal power dissipation	2.46 mW
Total thermal power dissipation	4.31 mW
Energy efficiency (pJ/bit)	0.01436

* *A: ALMs used in final placement. B: Estimate of ALMs recoverable by dense packing. C: Estimate of ALMs unavailable.*

The three outputs of the sampler module are connected to the post-processing module, as illustrated in Fig. 6. The post-processing stage plays a pivotal role in the domain of TRNGs by refining raw outputs into high-quality random numbers essential for various applications. Key tasks performed by post-processing algorithms include bias removal, pattern elimination through whitening techniques, and rigorous statistical testing to validate randomness. This process is vital for enhancing the reliability and security of the generated random numbers, guaranteeing they satisfy the rigorous criteria necessary for cryptographic applications, simulations, and other fields where impartial and unpredictable randomness is crucial. The XOR gates are commonly employed in the enhancement and de-correlation of data from TRNGs. XOR operations generate less predictable outputs when using multiple bits, enhancing the randomness of the produced numbers. Aggregating noise from various sources improves the output's reflection of true randomness. XOR gates, suitable for real-time applications in TRNG systems due to their simple implementation and low propagation delays, are efficient in hardware. The second sub-module of the proposed B+HCCES system handles the storage of generated numerical values and facilitates their transmission to a computer for subsequent analysis, such as the NIST SP90-B test and the BSI AIS-31 test. Initially, 8,388,608 bits of the available 12,492,800 bits in the embedded Block Random Access Memory (BRAM) of the INTEL 5CGXFC9D6F27C7 FPGA were allocated for storing random numbers generated by the B+HCCES TRNG architecture.

The clock speed for both read and write operations was synchronized with the clock speed generated by the embedded PLL, to simplify the design. Subsequently, a serial bitstream generator was developed to read data from these BRAM memory locations at a clock speed of 300 MHz. This generator transformed the incoming bitstream into serial frames before transmitting them to a laptop at a baud rate

of 1 MHz. The orchestration of synchronization and control signals among these modules is managed by the TRNG-Main Control Unit (TRNG-MCU). This unit oversees the overall data processing flow, ensuring seamless operation and efficient communication between the different components involved in the data storage, extraction, and transmission processes.

The detailed circuit structure of the second sub-module within the proposed B+HCCES system exhibits a high level of complexity, incorporating sophisticated mechanisms to synchronize diverse elements during various data transfer operations. Furthermore, it is equipped with dual operational modes to cater to both standard data streaming and restart burst streaming, specifically designed to comply with the stringent criteria outlined in the NIST SP90-B testing standards.

The computational performance and power consumption of the proposed B+HCCES TRNG system are illustrated in Table 2. The results show that the entire system consumed only 698 Adaptive Logic Modules (ALMs) out of the total available resources of the INTEL 5CGXFC9D6F27C7 FPGA chip. One of the key features in digital system design is the power consumption per the designed chip. In Intel FPGAs, thermal power dissipation is systematically divided into three key components: Estimated I/O thermal power dissipation, Estimated core static thermal power dissipation, and Total thermal power dissipation. Estimated I/O thermal power dissipation pertains to the thermal energy generated by input/output pins during data transfer, which is contingent upon the switching activity and the operational load from external devices. Estimated core static thermal power dissipation refers to the power consumed by the FPGA's internal logic during periods of inactivity, predominantly arising from leakage currents and the intrinsic characteristics of the silicon substrate. Total thermal power dissipation combines both the I/O and core static components, yielding a holistic measure of the FPGA's overall heat generation. This comprehensive evaluation is critical for effective thermal management strategies and ensuring sustained operational efficiency.

The projected static thermal power dissipation of the core is determined to be 533.50 mW, contributing to a total thermal dissipation of 539.93 mW. Furthermore, an estimated 6.43 mW of thermal dissipation is assigned specifically to input/output operations. These figures are derived under specific conditions: utilizing a 23mm heat sink in an environment with air flows at 200 Linear Feet per Minute (LFpM) airflow and considering a 12.5% toggle rate for input/output signals. Although the proposed B+HCCES TRNG system has demonstrated highly optimized results in terms of logic utilization and power consumption, conducting a direct comparison with other state-of-the-art contributions would be inequitable for two reasons. First, many existing studies typically focus solely on modules responsible for deriving final entropy values rather than considering the entirety of the TRNG system. Second, many state-of-the-art implementations use an ASIC design flow, which inherently shows very low power consumption. This is because, in the ASIC flow, power consumption is measured based on the actual power used by the logic cells, unlike FPGA implementations.

Therefore, we decided to analyze the proposed B+HCCES TRNG architecture independently, excluding other modules used for additional functions, such as internal storage, synchronization, control operations, and serial data exchange. The computational performance of the proposed B+HCCES TRNG architecture, as illustrated in Table 3, reveals that this FPGA implementation is notably efficient in both resource utilization and energy consumption. The design employs 23 combinational ALUTs and 3 dedicated flip-flops, resulting in a total requirement of 11.5 ALMs. Thermal power dissipation is meticulously distributed, with 0.02 mW for combinational cells, 1.79 mW for the clock enable block, 0.04 mW for register cells, and 2.46 mW for I/O operations, contributing to a total thermal power dissipation of 4.31 mW.

Furthermore, the implementation achieves an impressive energy efficiency of 0.01436 pJ per bit, underscoring its potential for low-power, high-performance applications. One of the most critical aspects of this design is the elimination of all IP (Intellectual Property) cores, except for the PLL. This constraint was strictly enforced for several reasons, which will be detailed in the following sections.

In the context of FPGAs, choosing to develop proprietary IP rather than relying on third-party IP cores presents several advantages. Firstly, creating our own IP can lead to substantial cost savings.

By avoiding the licensing fees typically associated with third-party IP, we reduced the prototype expenses. Additionally, in-house IP development allows us to tailor solutions precisely to specific needs, optimizing performance and efficiency for the application.

Another advantage is the increased independence and flexibility gained from developing in-house IP. By avoiding reliance on external vendors, organizations can handle updates, bug fixes, and support internally, resulting in faster development cycles and greater ease in making necessary adjustments. Moreover, controlling IP internally provides comprehensive oversight of its design and implementation, enhancing security and mitigating risks associated with vulnerabilities in third-party solutions. Retaining full intellectual property rights further strengthens control over the technology. Custom IP can be designed to integrate seamlessly with existing systems, thereby minimizing potential compatibility and integration issues. Additionally, developing in-house IP promotes the growth of the team's technical skills and expertise, which can be beneficial for future projects and innovation.

Avoiding vendor lock-in is another significant benefit. Dependence on third-party IP can limit future projects to specific vendors and their ecosystems. Developing proprietary IP allows for greater freedom in selecting tools and platforms. Additionally, managing in-house IP ensures that long-term maintenance and support can be tailored to specific needs, rather than relying on external vendors whose support might

change or be discontinued. However, it is essential to consider these benefits alongside potential drawbacks, such as the time required for development, the complexity of managing proprietary IP, and the need for specialized knowledge.

V. EXPERIMENTAL RESULTS

This section presents an entropy analysis of the non-deterministic sampling module in the proposed B+HCCES TRNG architecture. Entropy, a measure of the randomness or information content of a data source, is critical in evaluating the performance of TRNSs. Specifically, Shannon entropy is emphasized, as it quantifies the unpredictability of a random bit stream, serving as a metric for the level of disorder or uncertainty inherent in the data.

Higher entropy values signify a more secure and unpredictable output, which is necessary for cryptographic applications. In this context, the entropy analysis aims to assess the quality and robustness of the random bit stream generated by the proposed B+HCCES TRNG architecture, ensuring its compliance with security standards and its suitability for use in various security-sensitive applications. To validate the performance of the proposed B+HCCES TRNG architecture, we conducted two primary entropy tests: the NIST SP800-90B tests and the AIS 31 tests. Additional tests for evaluating the statistical properties of a TRNG include the Diehard and Dieharder Tests. However, we opted to use the NIST SP800-90B tests and the BSI AIS-31 tests due to their robust and reliable results.

Both tests utilize a range of statistical measures to assess the randomness of outputs from TRNGs. The first test is Excursion, which evaluates deviations from the expected mean, highlighting potential biases. The next test is NumDirectionalRuns, which counts sequences of consecutive identical bits. Following that is LenDirectionalRuns, which analyzes the lengths of these runs to determine consistency in randomness. The fourth test, NumIncreasesDecreases, tracks the total number of increases and decreases, revealing trends within the output. The fifth test is NumRunsMedian, which provides insights into the median lengths of runs, reflecting typical output characteristics. Next, LenRunsMedian assesses the median length of runs of identical bits.

The seventh test, AvgCollision, assesses the average number of collisions among outputs, indicating their uniqueness. The eighth test is MaxCollision, which identifies the maximum number of collisions for any single value. The ninth test, Periodicity, checks for repeating patterns in the output at various intervals (1, 2, 8, 16, and 32 bits). The tenth test is Covariance, which evaluates dependencies between bits at specified intervals. Next, Compression analyzes the output's compressibility, with high compressibility suggesting non-randomness. The twelfth test, Chi-square Independence, examines whether the observed frequency distribution aligns with expected randomness. Following that is the Chi-square Goodness of Fit, which assesses how well the data conforms to a theoretical distribution. The fourteenth test is the Length of the Longest Repeated Substring Test, which identifies the longest repeating substring to detect patterns. Finally, the last test is the Restart Test, which investigates how frequently the output returns to an initial state, revealing potential periodic behavior. Subsequently, these tests offer a comprehensive evaluation of the output's unpredictability and its adequacy for cryptographic applications.

TABLE 4. Results of NIST SP90-B test.

Test	C[0]	C[1]	C[2]
Excursion	18	0	6
NumDirectionalRuns	6	0	9
LenDirectionalRuns	17	6	0
NumIncreasesDecreases	66	0	6
NumRunsMedian	53	1	5
LenRunsMedian	420	6	0
AvgCollision	6	0	9
MaxCollision	4	2	28
periodicity (1)	70	0	6
Periodicity (2)	21	0	6
Periodicity (8)	6	0	37
Periodicity (16)	64	0	6
Periodicity (32)	56	1	5
Covariance (1)	7	0	6
Covariance (2)	15	0	6
Covariance (8)	12	0	6
Covariance (16)	6	0	28
Covariance (32)	136	0	6
Compression	6	0	9
Chi-square independence	p-value = 0.594202		
Chi-square goodness of fit	p-value = 0.517031		
length of longest repeated substring test	Passed		
Restart test	Passed		
Min entropy per bit	0.992343		
Min entropy per Byte	7.938744		

VI. NIST SP800-90B TESTS

The NIST SP 800-90B recommendations are fundamental for evaluating TRNGs because of their rigorous criteria for assessing randomness and security. The NIST SP800-90B test's evaluation of a TRNG entropy is strengthened with the incorporation of three distinct c[i] values, allowing for the calculation of statistical measures of randomness like sample entropy, Shannon entropy, and conditional entropy. Assessing randomness with varied sample sizes or conditions enhances the assessment's reliability.

The test more accurately identifies potential weaknesses in the output by comparing its values to predefined thresholds. This analysis, with its multifaceted approach, strengthens the evaluation of the TRNG's fitness for cryptographic uses, thereby generating reliable and secure random numbers. Standardization through NIST SP 800-90B is important for comparing various TRNGs and confirming their adherence to consistent security and performance standards. Compliance with these guidelines is vital for ensuring that TRNGs deliver the high level of security necessary for cryptographic applications that depend on random numbers.

As shown in Table 4, the proposed B+HCCES TRNG architecture successfully passed all evaluated tests under the NISF SP800-90B framework. This comprehensive evaluation

demonstrates the effectiveness and reliability of the TRNG design in producing high-quality random numbers. The Chi-square independence test yielded a p-value of 0.594202, well above the typical threshold of 0.05, indicating that there is no significant deviation from expected random behavior. Thus, the numbers generated by the TRNG are independent and unbiased. Similarly, the Chi-square goodness-of-fit test produced a p-value of 0.517031. This result further confirms that the distribution of the generated random numbers closely aligns with the expected uniform distribution, reinforcing the TRNG's capability to produce random numbers with the desired statistical properties. The length of the longest repeated substring test and the restart test were both passed, demonstrating the TRNG's effectiveness in avoiding long sequences of repeated patterns and its ability to successfully restart its operation.

These outcomes are essential for maintaining the randomness and unpredictability of the generated numbers.

The restart test is notably relevant within the NIST SP 800-90B suite, as it measures the TRNG's capacity to produce unbiased and unpredictable numbers following a restart or reset. If a restart or reset results in a predictable sequence, it may undermine the randomness and security of the generated numbers, which is particularly vital for applications requiring high levels of security, such as encryption. To visualize the random sequences generated after a restart, the chip was restarted 1,000 times, with 1,000 bits of random samples (nonces) collected during each restart operation. Fig. 7 shows the binary streams of 3 bytes after six restart operations. Fig. 8 presents four distinct power-up batches, each comprising 250 nonces, with each nonce measuring 125 bytes.

In terms of entropy, the TRNG demonstrated a minimum entropy per bit of 0.992343 and a minimum entropy per byte of 7.938744. These high entropy values indicate that the TRNG produces random numbers with a high degree of uncertainty and unpredictability, which is significant for applications requiring secure and reliable random number generation.

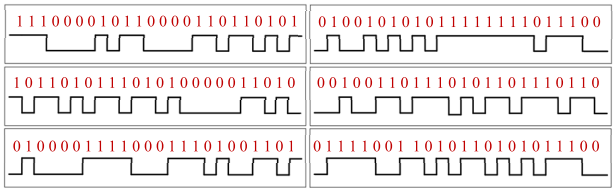


FIGURE 7. Test results of six restart tests.

A. AIS 31 TESTS

As illustrated in Table 5, the results from the AIS 31 tests demonstrate that the proposed B+HCCES TRNG (True Random Number Generator) architecture successfully passed all evaluations. This outcome is significant, as it underscores the TRNG's capability to meet the stringent requirements of the AIS 31 standard, which is crucial for ensuring high-quality randomness in the generated numbers. The AIS

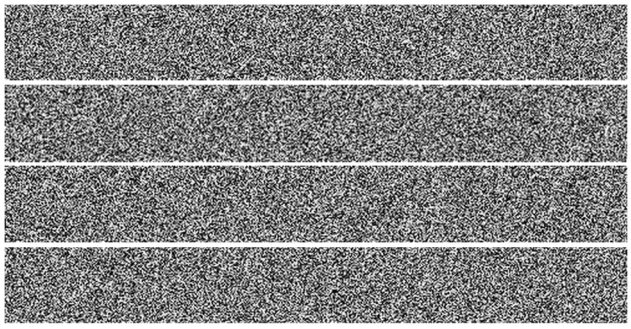


FIGURE 8. Distributions of four power-up batches: Each batch consists of 250 samples, each with a 1000-bit length.

31 tests assess various statistical properties of random numbers, including their uniformity, independence, and unpredictability. Passing all these tests confirms that the TRNG consistently produces random numbers with the desired characteristics, making it suitable for security-sensitive applications. The successful completion of these tests reflects the TRNG's reliability and adherence to established standards for randomness. Additionally, the entropy score of 7.996781 achieved by the TRNG is noteworthy. Entropy, in this context, quantifies the uncertainty or unpredictability in the random numbers produced. The near-maximum entropy score of 8 bits per byte indicates that the TRNG generates random numbers with a high degree of randomness and minimal bias. This high entropy value is critical for ensuring that the random numbers are truly unpredictable and suitable for applications requiring robust cryptographic security.

TABLE 5. Results of AIS-31 test.

	TEST	Pass Rate
P1/T0	Disjointness	Passed
P1/T1	Monobit	257/257
P1/T2	Poker	257/257
P1/T3	Run	257/257
P1/T4	Long run	257/257
P1/T5	Auto-correlation	257/257
T6-a	Uniform dist. (S<0.025)	|P(1) - 0.5| = 0.001730
T6-b	Uniform dist. (S<0.020)	p(01) = 0.50044 p(11) = 0.49811 |p_(01) - p_(11)| = 0.002329
T7-a	Comparative multinomial, width=3 (S<15.13)	test size [0] = 0.269121 test size [1] = 0.006480
T7-b	Comparative multinomial, width=4 (S<15.13)	test size [0] = 0.095220 test size [1] = 3.836958 test size [2] = 0.144500 test size [3] = 0.856983
T8	Entropy (S>7.976)	7.996781

VII. COMPARISON WITH EXISTING TRNGS BASED ON FPGA

Applications requiring high throughput from TRNGs include a range of fields that demand rapid generation of random

TABLE 6. Performance comparison with the state-of-the-art trng implementations.

Work	Entropy Source	Platform	Hardware Resources	Throughput (Mbps)	Power per system (W)	Energy Per system (nJ/bit)	Power per module (mW)	Energy per module (pJ/bit)
[12]	RO	Spartan-6	4LUT and 3DFF	0.76	NA	NA	NA	NA
[14]	STR	Spartan-6	56LUT and 19DFF	100	NA	NA	1.15	0.0115
[15]	DCFL	Cyclone-IV	298LUT	150	NA	NA	NA	NA
[16]	STR	Virtex-5	320LUT and 320DFF	200	NA	NA	NA	NA
[17]	MSFRO	Virtex-6	24LUT and 2DFF	290	3.687	12.712	NA	NA
[2]	DCCX	Artix-7	12LUT and 10DFF	150	NA	NA	NA	NA
	DCCX	Kintex-7	12LUT and 10DFF	200	NA	NA	NA	NA
This work	Race hazard and jitter	Cyclone-V GT	23LUT and 3DFF	**300**	**0.5399**	**1.799**	**4.31**	**0.0143**

numbers for security and efficiency. For instance, in cryptographic systems, such as those used in secure communications, TRNGs need to produce random numbers at rates exceeding 1 Gbps to effectively support encryption and decryption processes. Similarly, in online gaming and gambling, where fairness is crucial, platforms often require TRNGs to deliver random outcomes at data rates around 100 Mbps to manage multiple simultaneous user requests without latency. Financial services, particularly in online banking, also rely on high-throughput TRNGs, needing generation rates of up to 10 Mbps for secure token generation during transactions. Thus, the high throughput of the TRNG systems is important in many applications that require high speed to handle multiple requests simultaneously.

As depicted in Table 6, distinct differences emerge when comparing various TRNGs based on their hardware resources, portability, and throughput. The RO TRNG [12], implemented on a Spartan-6 FPGA, demonstrates the lowest throughput of 0.76 Mbps, with minimal hardware resources (4 LUTs and 3 DFFs). Despite using more hardware resources (9 SLICEs), it still represents a notable step up from the RO TRNG in terms of performance. The STR (Serial TRNG) [14], on the same Spartan-6 platform, achieves a throughput of 100 Mbps, using 56 LUTs and 19 DFFs. This implementation strikes a balance between hardware resource usage and performance, offering a significantly higher throughput compared to the RO and LXOR TRNGs. The DCFL (Dual Clock Frequency LFSR) TRNG [15], implemented on a Cyclone-IV FPGA, provides a throughput of 150 Mbps with a more substantial hardware footprint of 298 LUTs. While it delivers a higher throughput than the Spartan-6-based designs, its resource requirements may affect its suitability for resource-constrained applications. The STR TRNG [16] on a Virtex-5 FPGA achieves an even higher throughput of 200 Mbps, using 320 LUTs and 320 DFFs. This design improves upon the throughput of the DCFL TRNG but demands considerable hardware resources, which may limit its applicability in environments with stringent resource constraints.

The MSFRO (Multi-Stage Frequency Random Oscillator) TRNG [17], implemented on a Virtex-6 FPGA, delivers a very high throughput of 290 Mbps while using only 24 LUTs and 2 DFFs. This TRNG demonstrates excellent performance in terms of throughput relative to its hardware resource requirements, making it an efficient choice for high-performance applications. The DCCX (Dual Clock Chaotic XOR) TRNG [2], implemented on both Artix-7 and Kintex-7 FPGAs, shows a throughput of 150 Mbps and 200 Mbps, respectively, with consistent hardware resource usage of 12 LUTs and 10 DFFs. This implementation provides flexibility in choosing different FPGA platforms while maintaining a balance between throughput and hardware resource consumption. Finally, the B+HCCES TRNG, implemented on a Cyclone-V GT FPGA, exhibits the highest throughput of 300 Mbps with a hardware resource requirement of 23 LUTs and 3 DFFs.

This design not only delivers superior throughput but also demonstrates efficient use of hardware resources, making it a strong candidate for applications requiring high-speed random number generation. Overall, the B+HCCES TRNG outperforms its peers in terms of throughput while maintaining a reasonable hardware footprint and offering a compelling balance between performance and resource utilization. Most state-of-the-art contributions do not address the power and energy aspects of their TRNG modules and systems, focusing instead on logic utilization and overall throughput of the TRNG circuit. However, incorporating these two additional key metrics is crucial to completing the design cycle, as understanding power consumption is essential for determining the suitability of the TRNG for various applications. Additionally, it is important to compare the power consumption of only the TRNG module, as well as the power usage after incorporating the additional modules responsible for bitstream aggregation, storage, and transmission, which together form the complete TRNG system.

As illustrated in Table 6, the proposed B+HCCES TRNG in this work has a power consumption of 0.5399 W, whereas

the MSFRO system [17] exhibits a much higher power consumption of 3.687 W, making it approximately 6.83 times more power-hungry. Similarly, in terms of energy consumption, the system in this work consumes 1.799 nJ/bit, while the MSFRO system [17] shows a consumption of 12.712 nJ/bit, indicating that it uses about 7.06 times more energy. These comparisons highlight that the MSFRO system [17] is considerably less efficient than the system discussed in this work, suggesting that the latter may be better suited for applications with strict power and energy constraints.

In comparing the performance metrics of the two TRNG modules, notable distinctions can be observed in terms of power consumption, energy efficiency, and throughput. The TRNG module presented in this work has a power consumption of 4.31 mW and an energy consumption of 0.0143 pJ/s, achieving a throughput of 300 Mbps.

In contrast, the STR module [14] exhibits a slightly lower power consumption of 1.15 mW and energy consumption of 0.0115 pJ/s, with a throughput of 100 Mbps. While the STR module [14] demonstrates lower power and energy metrics, the TRNG module in this work significantly outperforms it in throughput, with an increase of 200 Mbps. This comparison highlights the trade-off between power efficiency and throughput capabilities, suggesting that the TRNG module presented here may be more suitable for applications requiring higher data rates, despite its higher power consumption. The overall power consumption of the STR system [14] is not specified. However, we are almost confident that our proposed solution will demonstrate promising results, as we did not rely on any intellectual property (IP) cores or external modules; instead, we designed and optimized our own TRNG bitstream aggregation and storage units. Conversely, the STR system utilized Xilinx MicroBlaze IPs, which are generic rather than optimized solutions. Furthermore, they did not specify any memory block or provide an explanation of how they stored the bitstream prior to transmitting it via the relatively slow UART module to the computer in detail.

The previous comparisons with state-of-the-art TRNG modules highlight the key contributions and promising outcomes of this work. However, it is important to note that while the proposed B+HCCES TRNG module is optimized, the overall B+HCCES TRNG system is not fully optimized, and its maximum speed is constrained by the internal design choices we made in the Storage and serial processing architecture.

FPGAs are robust and adaptable devices utilized in numerous applications, yet they are also vulnerable to various types of attacks. Multiple attacks can target TRNGs, such as temperature attacks, underpower attacks [34], electromagnetic attacks [35], and hardware Trojan attacks [36]. Power, temperature, and frequency injection attacks are particularly effective at compromising the entropy of TRNGs that rely on ring oscillators. These methods significantly weaken the randomness generated by these devices.

FPGA power attacks targeting TRNGs take advantage of variations in power consumption to extract sensitive data or interfere with the TRNG's operations. By examining power traces during the random number generation process, attackers can detect patterns that might expose internal states or specific outputs. Power-wasting circuits, like ROs, can be utilized to deliberately overload the power regulator, and studies have indicated that such power attacks significantly reduce the randomness of RO-based TRNGs [37]. Additionally, the injected ring oscillator circuits inevitably produce excess heat, which causes fluctuations in the overall temperature of the FPGA chip over time.

Therefore, we implemented 2,000 ROs as a hard macro and programmed them into the FPGA, following a similar approach to the experiment described in [21]. These ROs feature varying numbers of NOT gate stages, specifically 1, 3, 5, 7, 9, and 11 stages. The generated random numbers were evaluated using NIST 800-90B and AIS31 testing standards. The results indicated that fluctuations in voltage and temperature did not affect the proposed B+HCCES TRNG module, as it successfully passed all test suites.

The frequency injection attack method consists of injecting a signal with a designated frequency into the clock lines or essential timing components of an FPGA. This interference can severely disrupt the FPGA's typical functioning, resulting in incorrect signal interpretations or the execution of unintended tasks [38]. In this research, we exposed the proposed B+HCCES TRNG module to four specific subharmonic frequencies: 150 MHz (first subharmonic), 100 MHz (second subharmonic), 75 MHz (third subharmonic), and 60 MHz (fourth subharmonic). After the injection, we collected the output bitstream sequences and analyzed them according to the NIST SP800–90B and BSI AIS-31 standards. The results indicated that all random numbers successfully passed the tests, with the minimum entropy of the sequences fluctuating by no more than 0.1087% both before and after the frequency injection.

Based on these security validation observations, our proposed TRNG demonstrated robust and resilient performance, successfully operating under significant threats. While further testing could provide additional insights into its capabilities, such evaluations would necessitate more expensive equipment and specialized environments. Additionally, some tests are exclusively applicable to ASIC-based chips, limiting their relevance for our FPGA implementation. Overall, the findings affirm the TRNG's effectiveness in maintaining security under challenging conditions [39], [40], [41], [42].

It is essential to emphasize that researchers in this domain primarily focus on the ability of their systems to generate high-quality random numbers rather than solely comparing output entropy values with other leading methods. As long as the generated bitstream successfully passes all statistical tests set forth by recognized standards, such as NIST SP800-90B and BSI AIS-31, and meets the established thresholds for randomness, the emphasis shifts toward practical

performance. Consequently, the primary objective is to propose an innovative TRNG circuit capable of producing diverse and high-quality noise bitstreams that can withstand scrutiny [43], [44], [45]. Additionally, significant attention is devoted to optimizing various parameters that influence noise generation. This includes factors such as the utilization of logic resources, the throughput of the bitstream, and overall energy consumption, all of which are crucial for ensuring efficiency and effectiveness in real-world applications. These aspects are further elaborated in Table 6, which outlines the specific optimizations and their implications for the performance of the TRNG.

Also, it is important to note that we employed a single FPGA chip for bitstream generation, without relying on any external hardware components. This method provides a considerable cybersecurity advantage, reinforcing the resilience of our TRNG module against potential hardware hacking threats.

VIII. CONCLUSION

The B+HCCES TRNG architecture, implemented on a Cyclone-V GT FPGA, achieves an impressive throughput of 300 Mbps while efficiently utilizing only LUTs and 3 DFFs. This remarkable design not only meets but surpasses the stringent evaluations of both the NIST SP800-90B and BSI AIS-31 tests, which are critical benchmarks in assessing the quality and reliability of random number generators. Furthermore, when compared to existing FPGA-compatible TRNGs, the B+HCCES TRNG demonstrates exceptional performance metrics. The comparative analysis reveals that the proposed TRNG excels in striking an optimal balance between resource utilization, throughput, and the overall quality of the generated output sequence. This efficient use of resources allows for scalability and adaptability in various applications without compromising performance. As a result, the B+HCCES TRNG sets a new standard in the field of random number generation, offering a highly efficient and high-performing solution specifically tailored for FPGA-based systems. This advancement not only enhances the security and reliability of applications that depend on random number generation but also paves the way for future innovations in TRNG design and implementation.

REFERENCES

[1] Y. Chen, Y. Tian, R. Zhou, D. M. Castro, D. Guo, and Q. Zhou, "NDSTRNG: Non-deterministic sampling-based true random number generator on SoC FPGA systems," *IEEE Trans. Comput.*, vol. 73, no. 5, pp. 1313–1326, May 2024.

[2] S. Yang, H. Liang, R. Hu, L. Yao, Z. Huang, M. Yi, and Y. Lu, "Lightweight hybrid entropy source true random number generator based on jitter and metastability," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 71, no. 7, pp. 3513–3517, Jul. 2024.

[3] M. A. Qureshi and A. Munir, "PUF-RAKE: A PUF-based robust and lightweight authentication and key establishment protocol," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 4, pp. 2457–2475, Jul. 2022.

[4] Q. Wang and D. Wang, "Understanding failures in security proofs of multi-factor authentication for mobile devices," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 597–612, 2023.

[5] H. O. Ahmed and D. Wyatt, "Adaptive prognostic malfunction based processor for autonomous landing guidance assistance system using FPGA," *IEEE Access*, vol. 12, pp. 2113–2122, 2024.

[6] H. O. Ahmed, "Coarse grained FLS-based processor with prognostic malfunction feature for UAM drones using FPGA," in *Proc. Integr. Commun., Navigat. Surveill. Conf. (ICNS)*, Apr. 2023, pp. 1–6.

[7] M. Safi, B. Simpson, E. Clark, H. Idriss, and T. Idriss, "Lightweight random obfuscation protocol: A PUF-based mutual authentication protocol for IoT devices," in *Proc. IEEE Int. Conf. Artif. Intell., Blockchain, Internet Things (AIBThings)*, Sep. 2023, pp. 1–5.

[8] Y.-W. Lee, Y. Lee, M. Moon, and S. Kang, "Tunable compact probing detector with fast analysis time against invasive attacks," in *Proc. Int. SoC Design Conf. (ISOCC)*, Jeju, Korea (South), Oct. 2019, pp. 115–116.

[9] H. Li, G. Du, C. Shao, L. Dai, G. Xu, and J. Guo, "Heavy-ion microbeam fault injection into SRAM-based FPGA implementations of cryptographic circuits," *IEEE Trans. Nucl. Sci.*, vol. 62, no. 3, pp. 1341–1348, Jun. 2015.

[10] X. Liu and P. Ampadu, "Distributed on-chip power supply for security enhancement in multicore NoC," in *Proc. IEEE 34th Int. Syst.-Chip Conf. (SOCC)*, Las Vegas, NV, USA, Sep. 2021, pp. 212–217.

[11] A. Ghosh, M. Nath, D. Das, S. Ghosh, and S. Sen, "Electromagnetic analysis of integrated on-chip sensing loop for side-channel and fault-injection attack detection," *IEEE Microw. Wireless Compon. Lett.*, vol. 32, no. 6, pp. 784–787, Jun. 2022.

[12] R. Della Sala, D. Bellizia, and G. Scotti, "A novel ultra-compact FPGA-compatible TRNG architecture exploiting latched ring oscillators," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 3, pp. 1672–1676, Mar. 2022.

[13] R. Della Sala, D. Bellizia, and G. Scotti, "High-throughput FPGA-compatible TRNG architecture exploiting multistimuli metastable cells," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 12, pp. 4886–4897, Dec. 2022.

[14] X. Wang, H. Liang, Y. Wang, L. Yao, Y. Guo, M. Yi, Z. Huang, H. Qi, and Y. Lu, "High-throughput portable true random number generator based on jitter-latch structure," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 2, pp. 741–750, Feb. 2021.

[15] X. Wu and S. Li, "A new digital true random number generator based on delay chain feedback loop," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2017, pp. 1–4.

[16] A. Cherkaoui, V. Fischer, L. Fesquet, and A. Aubert, "A very high speed true random number generator with entropy assessment," in *Proc. 15th Int. Workshop Cryptograph. Hardware Embedded Syst. (CHES)*, Santa Barbara, CA, USA, Aug. 2013, pp. 179–196. [Online]. Available: https://ujm.hal.science/ujm-00859906

[17] J. Cui, M. Yi, D. Cao, L. Yao, X. Wang, H. Liang, Z. Huang, H. Qi, T. Ni, and Y. Lu, "Design of true random number generator based on multi-stage feedback ring oscillator," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 3, pp. 1752–1756, Mar. 2022.

[18] T. Ni, Q. Peng, J. Bian, L. Yao, Z. Huang, A. Yan, S. Wang, and X. Wen, "Design of true random number generator based on multi-ring convergence oscillator using short pulse enhanced randomness," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 70, no. 12, pp. 5074–5085, Dec. 2023.

[19] T. Chen, S. Jia, Y. Ma, Y. Cao, N. Lv, W. Wang, J. Yang, and J. Lin, "A design of high-efficiency coherent sampling based TRNG with on-chip entropy assurance," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 70, no. 12, pp. 5060–5073, Dec. 2023.

[20] A. Peetermans and I. Verbauwhede, "Characterization of oscillator phase noise arising from multiple sources for ASIC true random number generation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 71, no. 3, pp. 1144–1157, Mar. 2024.

[21] Y. Luo, W. Wang, S. Best, Y. Wang, and X. Xu, "A high-performance and secure TRNG based on chaotic cellular automata topology," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 12, pp. 4970–4983, Dec. 2020.

[22] E. Elmitwalli and S. Köse, "Bistable Josephson junction-based true random number generator without inductors," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 70, no. 4, pp. 1615–1619, Apr. 2023.

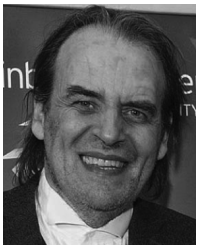
[23] A. Tamakoshi, N. Onizawa, H. Yamagata, H. Fujita, and T. Hanyu, "Design of an energy-efficient true random number generator based on triple read-write data-stream multiplexing of MTJ devices," in *Proc. 18th IEEE Int. New Circuits Syst. Conf. (NEWCAS)*, Jun. 2020, pp. 283–286.

[24] J. J. Tatarkiewicz and W. Kuzmicz, "Steganographic stream cipher encryption using true random number generator," in *Proc. 30th Int. Conf. Mixed Design Integr. Circuits Syst. (MIXDES)*, Jun. 2023, pp. 244–247.

[25] M. S. Sajal and M. Dandin, "True random number generation using dark noise modulation of a single-photon avalanche diode," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 71, no. 3, pp. 1586–1590, Mar. 2024.
[26] F. Yuan, S. Li, Y. Deng, Y. Li, and G. Chen, "Cu-doped TiO_{2-x} nanoscale memristive applications in chaotic circuit and true random number generator," *IEEE Trans. Ind. Electron.*, vol. 70, no. 4, pp. 4120–4127, Apr. 2023.
[27] R. Patgiri, "Rando: A general-purpose true random number generator for conventional computers," in *Proc. IEEE 20th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, China, Oct. 2021, pp. 107–113.
[28] R. Patgiri and N. B. Muppalaneni, "Stealth: A highly secured end-to-end symmetric communication protocol," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, China, Jul. 2022, pp. 1–8.
[29] I. Baturone, R. Román, and Á. Corbacho, "A unified multibit PUF and TRNG based on ring oscillators for secure IoT devices," *IEEE Internet Things J.*, vol. 10, no. 7, pp. 6182–6192, Apr. 2023.
[30] P. Kirtonia, S. Williams, and M. Bayoumi, "Jitter-based true random number generator with dynamic selection bit reconfiguration," in *Proc. IEEE 67th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2024, pp. 162–166.
[31] Y. Yao, X. Chen, W. Kang, Y. Zhang, and W. Zhao, "Thermal Brownian motion of skyrmion for true random number generation," *IEEE Trans. Electron Devices*, vol. 67, no. 6, pp. 2553–2558, Jun. 2020.
[32] Y. Yamanashi and N. Yoshikawa, "Superconductive random number generator using thermal noises in SFQ circuits," *IEEE Trans. Appl. Supercond.*, vol. 19, no. 3, pp. 630–633, Jun. 2009.
[33] F. Tehranipoor, P. Wortman, N. Karimian, W. Yan, and J. A. Chandy, "DVFT: A lightweight solution for power-supply noise-based TRNG using dynamic voltage feedback tuning system," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 6, pp. 1084–1097, Jun. 2018.
[34] Y. Cao, V. Rožic, B. Yang, J. Balasch, and I. Verbauwhede, "Exploring active manipulation attacks on the TERO random number generator," in *Proc. IEEE 59th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Oct. 2016, pp. 1–4.
[35] P. Bayon, L. Bossuet, A. Aubert, V. Fischer, F. Poucheret, B. Robisson, and P. Maurine, "Contactless electromagnetic active attack on ring oscillator based true random number generator," in *Proc. Int. Workshop Construct. Side-Channel Anal. Secure*, Dec. 2012, pp. 151–166.
[36] A. P. Johnson, R. S. Chakraborty, and D. Mukhopadhyay, "A novel attack on a FPGA based true random number generator," in *Proc. Workshop Embedded Syst. Secur.*, Oct. 2015, pp. 1–6.
[37] D. Mahmoud and M. Stojilovic, "Timing violation induced faults in multi-tenant FPGAs," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Florence, Italy, Mar. 2019, pp. 1745–1750.
[38] A. T. Markettos and S. W. Moore, "The frequency injection attack on ring-oscillator-based true random number generators," in *Cryptographic Hardware and Embedded Systems—CHES 2009* (Lecture Notes in Computer Science), vol. 5747, C. Clavier and K. Gaj, Eds. Lausanne, Switzerland: Springer, Sep. 2009, pp. 317–331, doi: 10.1007/978-3-642-04138-9_23.
[39] J. Rajski, M. Trawka, J. Tyszer, and B. Wlodarczak, "A lightweight true random number generator for root of trust applications," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 42, no. 9, pp. 2815–2825, Sep. 2023.
[40] A. M. Garipcan and E. Erdem, "Hardware design and analysis of ring oscillator based noise source for true random number generators," in *Proc. Int. Conf. Artif. Intell. Data Process. (IDAP)*, Malatya, Turkey, Sep. 2018, pp. 1–6.
[41] K. Gai, Y. Ding, A. Wang, L. Zhu, K. R. Choo, Q. Zhang, and Z. Wang, "Attacking the Edge-of-Things: A physical attack perspective," *IEEE Internet Things J.*, vol. 9, no. 7, pp. 5240–5253, Apr. 2022.
[42] F. Zokaee, F. Chen, G. Sun, and L. Jiang, "Sky-sorter: A processing-in-memory architecture for large-scale sorting," *IEEE Trans. Comput.*, vol. 72, no. 2, pp. 480–493, Feb. 2023.
[43] S. Akter, S. Williams, K. Khalil, and M. Bayoumi, "A hybrid random number generator based on MetaStability-ring oscillator linear feedback shift registers (MSRO-LFSR)," in *Proc. IEEE 67th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Springfield, MA, USA, Aug. 2024, pp. 1135–1139.
[44] R. D. Sala, D. Bellizia, and G. Scotti, "Unveiling the true power of the latched ring oscillator for a unified PUF and TRNG architecture," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, early access, Sep. 4, 2024, doi: 10.1109/TVLSI.2024.3448503.
[45] R. Zhang, H. Zhang, X. Wang, Y. Ziyang, K. Liu, S. Nishizawa, K. Niitsu, and H. Shinohara, "De-correlation and de-bias post-processing circuits for true random number generator," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 71, no. 11, pp. 5187–5199, Nov. 2024.

HOSSAM O. AHMED (Senior Member, IEEE) received the B.S. degree in communications and computer engineering from El-Shorouk Academy, Egypt, in 2007, and the M.Sc. and Ph.D. degrees in electronics and electrical communications engineering from Ain Shams University, Egypt, in 2015 and 2019, respectively. He is currently an Assistant Professor with the Computer Engineering Department, American University of the Middle East (AUM), Kuwait. His research interests include wireless sensor network applications, artificial intelligence hardware accelerators, intelligent control systems, fuzzy systems applications, cybersecurity, and control systems design using VHDL/FPGA. Also, he is an active member of SASD Voting Members [Standards Activity Subdivisions Committee (SASD)].

DONGHOON KIM (Senior Member, IEEE) is currently an Assistant Professor with the Department of Aerospace Engineering and Engineering Mechanics, University of Cincinnati (UC). Before joining UC, he was an Assistant Professor with the Department of Aerospace Engineering, Mississippi State University (MSU) for two years. During his time with MSU, he chaired the Autonomy Research Group and established the Autonomous System Research Laboratory. Prior to that, he was with the Robotics Division, LG Electronics, South Korea, for two and a half years as a Senior Research Engineer, where he designed and developed various service robotic platforms. His research interests include autonomous system design and development, fault-tolerant control, health monitoring of autonomous systems, celestial mechanics, and applications of machine and deep learning. He is a Senior Member of the American Institute of Aeronautics and Astronautics (AIAA). Additionally, he serves on the technical committee of AIAA Astrodynamics and Space Automation and Robotics, the International Federation of Automatic Control (IFAC) Optimal Control, Intelligent Autonomous Vehicles, Robotics, Computational Intelligence in Control, and Adaptive and Learning Systems.



WILLIAM J. BUCHANAN is currently a Professor with the School of Computing, Edinburgh Napier University, Edinburgh, U.K. He also leads the Blockpass ID Laboratory and the Centre for Cybersecurity, IoT, and Cybephysical. He has published more than 30 academic books and over 400 academic research articles. He works in the areas of blockchain, cryptography, trust, and digital identity. He has one of the most extensive cryptography sites in the World (asecuritysite.com). He is involved in many areas of novel research and teaching. Along with this, his work has led to many areas of impact, including three highly successful spin-out companies, along with awards for excellence in knowledge transfer, and teaching. In 2024, he was elected as a fellow of the Royal Society of Edinburgh. He was awarded the "Outstanding Contribution to Knowledge Exchange" Award and an OBE.