Computer Systems Science & Engineering
DOI:10.32604/csse.2022.022280
Article

Tech Science Press

Design of Hybrid True Random Number Generator for Cryptographic Applications

S. Nithya Devi[1,*] and S. Sasipriya[2]

[1]Electronics and Communication Engineering, Dr. N.G.P. Institute of Technology, Coimbatore, 641110, India
[2]Electronics and Communication Engineering, Sri Krishna College of Engineering and Technology, Coimbatore, 641008, India
*Corresponding Author: S. Nithya Devi. Email: nithyadevis2015@gmail.com
Received: 03 August 2021; Accepted: 29 September 2021

Abstract: In real-time applications, unpredictable random numbers play a major role in providing cryptographic and encryption processes. Most of the existing random number generators are embedded with the complex nature of an amplifier, ring oscillators, or comparators. Hence, this research focused more on implementing a Hybrid Nature of a New Random Number Generator. The key objective of the proposed methodology relies on the utilization of True random number generators. The randomness is unpredictable. The additions of programmable delay lines will reduce the processing time and maintain the quality of randomizing. The performance comparisons are carried out with power, delay, and lookup table. The proposed architecture was executed and verified using Xilinx. The Hybrid TRNG is evaluated under simulation and the obtained results outperform the results of the conventional random generators based on Slices, area and Lookup Tables. The experimental observations show that the proposed Hybrid True Random Number Generator (HTRNG) offers high operating speed and low power consumption.

Keywords: True random number generators; lookup table; random number generator; digital circuit; seed

1 Introduction

Information Technology (IT) plays a major role in providing solutions to the modern problems. The global organizations are requesting Virtual Private Networking (VPN) functionality to provide feasibility to all the employees. Semiconductor retailers have the responsibility to assist the VPN access. They will offer specific terms that incorporate all the necessary security works on one device. A portion of these abilities includes computer system designs like encryption and decoding. The frequently unnoticed capacity to create random numbers is the basic square to retain a protected VPN framework. Most of the real-time scenarios, the cost of testing the Integrated Circuits (ICs) are more expensive. The main motive of the semiconductor industry is to provide cheap and efficient product delivery. The design verification and testing cost exceeds the manufacturing cost. Hence, the industry has framed new verification methodology

concepts like Open Verification Methodology, Universal Verification Methodology (UVM), etc… These methodologies will reduce the employee cost and the industry does not depend upon a particular employee.

1.1 Motivation

Globally, the produced arbitrary numbers should be uncorrelated and must fulfill the factual tests for irregularity. It is necessary to find out the loopholes and the corner cases of a design. This can be accomplished by either the "really arbitrary" "truly random" or "pseudo-random" numbers. The previous values show genuine randomness and the estimation of the next number remains unpredictable. The later one will be arbitrary in nature. The succession is identified on the explicit numerical calculations and this appears to be tedious and conventional. In any case, if the cycle time frame is exceptionally huge, the succession gives off an impression of being non-repetitive and irregular. Hence, the random generator must be cheap as well as arbitrary in nature. With this motivation, the proposed design is carried out with the modification of the True Random Number Generator (TRNG).

1.2 Applications of the Random Generators

The random generators are not only utilized in the semiconductor industry but also utilized in the following applications.

- Gaming
- Bio-medical
- Weather prediction
- Arithmetical and numerical sampling
- Cryptography and some other areas with sudden variation in their result.

The random numbers are important factors in most of the real-time applications. Most of the existing algorithms are achieved by this objective with the Pseudorandom-Number-Generator (PRNG), the Linear Feedback Shift Registers (LFSRs), the High Entropy Chaos-Based Truly Random Number Generator, etc.

1.3 Role of the Random Generators in the Semiconductor Industry

The general objective of any chip manufacturing is to provide low cost and high-speed operations. To accomplish these objectives, the random number generators are very useful in modeling the chip from the design phase to a silicon chip. Between the Register Transfer Language (RTL) and simulation, there is a stage called Design for Test (DFT). Most of the power level issues will be taken care of at Gate level simulations (GLS) stage. The Test Response Compactor (TRC), Multiple Input Signature Register (MISR) and PRNG are capable of generating the test vectors to test the Device under Test (DUT). These conventional methodologies will take a longer time to achieve 100% fault coverage and functional coverage. Mostly, the TRNGs offer an elevated level of non-determinism and irreproducibility. Yet, it doesn't follow an ideal uniform distribution and uniqueness. Thus, the post-processing unit is required for processing the output of a TRNG for reducing the repeating sequences and for generating the various sequences. TRNG consistently possesses the forward and the backward security features, since they have no deterministic segments. These factors may result in a complex circuit that decides the cost. Therefore, the models are very expensive and some of them are slow as well. With that concern, the major motive of the Hybrid TRNG is to build an automated Hybrid TRNG for the encryption standards. In this work, the Hybrid TRNG has been proposed for improving the random generation for reducing the testing progress and for enhancing the performance of a chip design. The performance metrics like delay and power would be calculated. The implementation would be carried out in the Xilinx ISE Design suite 14.7 simulation with the observation of the Lookup Table utilization, Slices and other internal components. The remainder of the paper has been organized as follows: Section 2 deals with the

literature survey and narrates the various available methods for the random generation. The suggestions and future enhancement concepts are also observed. Section 3 discusses the research methodology with the modification of the TRNG. Section 4 elaborates more about the experimental observations that will help in deciding on the proposed method and comparative analysis has also been presented in that section. Section 5 summarizes the research with some suggestions.

2 Literature Survey

Security measures are important factors that are facing challenging roles in the recent days. Most of the communication protocols utilize the standard computation algorithms for safeguarding the customer data. Callegari et al. [1] proposed a concept of Embeddable Analog-Digital Conversion (ADC)-based TRNG. They managed the design to adopt an uncorrelated binary sequence. The random source is generated based on the Kneading Matrix with the state aggregation process.

The Kneading matrix is given by:

$$K_{i,j} = \frac{\mu(X_i \cap M^{-1}(X_j))}{\mu X_i} \tag{1}$$

As per the Eq. (1) the entries of K is obtained from M, where μ is the common interval measure. The observed region states that the analog part of the design will lead in more noise and it can be compensated by substituting the XOR logic and verified using the National Institute of Standards and Technology (NIST) statistical analysis. Cryptographic frameworks prefer the random number generators for delivering the keys and the other mystery extents. Bagini and Bucci [2] targeted the characterization of the plan of a valid random number generator that has a straightforward and dependable usage with low cost. In the communication protocols, the symmetric and the public-key crypto-systems framed the Rivest–Shamir–Adleman (RSA) module that would help in the maintenance of a secure transmission with a random key.

The random key matrix played a major role in providing the Symmetric key cryptography applications. Nath et al. [3] considered a concept of the American Standard Code for Information Interchange (ASCII) code (0 to 255) for generating a random key matrix. They suggested the method for use in the public crypto applications. Even then, the TRNG is the one that is to be approximated. The limitations of the traditional methods are repeating the already generated sequence. Hence, TRNGs are capable of maintaining the non-repeatability property. Hu et al. [4] presented the concept of generating the 256-bit random number with the mouse movements.

Generally, the TRNG utilizes a non-deterministic approach with some post-processing abilities to create the arbitrariness. The deterministic calculations can be handled by the PRNG as it utilizes the information called as a seed. TRNGs are normally found on a wide range of physical marvels like thermal noise, coin distribution, etc. It will help predict the finite memory concepts for generating a sample. The algorithm must be arbitrary for providing the maximum security features. This will help in maintaining the hardware cryptographic systems. An industry-standard, System-On-a-Chip (SoC) level framework becomes progressively powerful in the IC development. In that, the Random Number Generators (RNGs) role will be required for securing the applications. But in most of the cases, the external noises like thermal noise, shot noise, etc., will exist in the development area. This happens more in the mixed-signal design units. Hence, the researchers have suggested the creation of new equipped strategies for delivering the irregular arrangements by utilizing the challenged noise sources.

The TRNG generous bit of scope is the algorithmic nature that effectively embeds them in any of the computerized circuit or framework. Physical-RNGs are the best estimates of the TRNGs and the discrimination factor is the one that is frequently related to them. Unfortunately, they may require very specific equipment and additional natural conditions; this makes the insertion process expensive. Ignoring

this risk, the security-related applications have been upgraded firmly by pushing their advancement and transmission; with the goal that the greater players in the Information Technology are currently presenting the physical-RNGs in their security stages. It would be attractive to plan for the RNGs to organize the highlights of the physical sources and, simultaneously, to plan for the friendliness of the computerized sources. Most of the recent methods have decreased the plan cost of the TRNGs by manipulating the Phase-Locked-Loops (PLLs) and the Field Programmable Gate Arrays (FPGAs) Fischer and Drutarovsky [5]. Sunar et al. [6] presented the concept of the TRNG to avoid the attacks from the unknown sources. They concentrated especially on the ring oscillators and their applications. Most of the complicated tasks can be solved by adopting the FPGA in generating the random nature. The authors Kohlbrenner and Gaj [7] presented the Configurable Logic Blocks (CLBs) for automatically self-testing the blocks.

The quality of the testing and the bit production rate could be determined by the aspects like thermal noise and telegraph noises. The thermal noise in the MOS semiconductor devices can be modeled by:

$$\sigma^2 = 4kT\gamma g_m \Delta f \tag{2}$$

As per the Eq. (2) as a normal random variable with zero mean and variance, the probability that the final metastable outcome is dictated by thermal noise and computed. Likewise, Tokunaga et al. [8] implemented a metastability-based feedback control, which may guide the random output bits with maximum entropy. Jiang et al. [9] presented a novel TRNG dependent on a diffusive memristor, a recently evolved unstable device that depends on the dispersion elements of the metal molecules in the memristive layer.

The device changes to a low-resistance state under a voltage beat after an irregular defer time and unwind back to the high-opposition state irrespective of the supply of the applied electrical inclination. They utilized the natural stochasticity of the defer time as the wellspring of the arbitrariness for fabricating the TRNG unit that comprises of just a diffusive memristor, a comparator, an AND-gate, and a counter. By comparing the existing TRNGs that are dependent on the non-unstable memristor self-off-exchanging conduct in the diffusive memristor extraordinarily decreases the vitality utilization since no RESET process is required. The Digital Clock Manager (DCM) and the Dynamic Partial Reconfiguration (DPR) are utilized for generating the random number presented by Johnson et al. [10]. DPR is generally a new improvement in the FPGA innovation, whereby the alterations to the predefined segments of the FPGA rationale texture is conceivable on–the–fly without influencing the ordinary use of the FPGA. The Xilinx Clock Management Tiles (CMTs) contain the Dynamic Reconfiguration Port (DRP) which permits the DPR to perform through a lot less difficult methods. Utilizing the DPR, the clock frequencies produced can be changed on–the–fly by modifying the comparing DCM parameters. The Xilinx Clock signal frequency is given by:

$$F_{CLKFX} = F_{CLKIN} \cdot \frac{M}{D} \tag{3}$$

As per Eq. (3), F_{CLKFX} is the frequency of the input signal and (M, D) are the multiplication and the division factors of the Xilinx DCM specifications. The DPR using the DRP is an additional preferred position in the FPGAs as it permits the client to tune the clock recurrence according to the need. Structure procedures exist to forestall any malevolent controls through the DPR which in different manners may unfavorably influence the security of the framework. The summary of survey shown in Tab. 1.

Table 1: Comparison of the Ring oscillator role in the TRNGs

Citation	Methodology	Merits	Suggestions
Vasyltsov et al. [11]	Ring oscillator with independent entropy sources	Digital TRNG, High entropy, Maximum throughput	To increase the quality of the TRNG
Bayon et al. [12]	Electromagnetic (EM) analysis on ring oscillator in TRNG	For improving previously published EM active attack	Security will be lost if the attacker heat up the device
Liu et al. [13]	Low-cost low-power ring oscillator-based TRNG	Applied in an SD card for encryption application	Further, power can be minimized
Robson et al. [14]	TRNG based on Ring Oscillator Utilizing Last Passage Time	Avoiding correlation and achieved high entropy	Try to make entropy very close to one.

Markettos and Moore [15] explained the concept of inspecting the activity of the ring oscillator, and clarified how the rule of infusion attack might be utilized by an assailant to assume the responsibility for this entropy source. If the random combinations are reduced, then it will be easy to find the exact payment card details with the security code. Non-uniform measurements may empower the attacker to figure the normal qualities or sequences. Entropy involves a source of vulnerability in a typically advanced framework. Failure of these properties in the inconspicuous stage has been limited in the cryptographic frameworks. Itaya and Jitsumatsu [16] proposed the concept of Beta encoder which is robust. Hence, the combination of the Beta encoder with the EXOR logic gate would result in strong correlation. Most of the researchers concentrated on the analog circuit design that may fail in producing the correlation.

Gabriel et al. [17] proposed a generator model for the unique random generations. Likewise, most of the researcher's are concentrating on the generation of purity of a continuous-variable quantum vacuum state for generating the unique random numbers, Ma et al. [18].

Cao et al. [19] implemented a silicon based random generation. Most of the recent methodologies implemented neural networks and machine learning concepts for deploying the TRNGs for generating many of the initial parameters. As per the industrial requirement, the standard CMOS-compatible TRNGs should hold full entropy, maximum throughput and small footprints are to be attracted for the integration into the security solutions of the commercial products.

In this section, we have reviewed the outline of the methodology that is conducting the exploration in the TRNG. The aspects of the block, size, security, delay and power have been dealt with. Hence, this research has focused on the implementation of the hybrid TRNG by avoiding the need of some analog components for eliminating the risk factors. Finally, this section concludes that the previous methodologies have less entropy while randomizing indirectly, consuming more time and cost in the chip productions.

3 Research Methodology

In TRNG, the random numbers are generated from the output of a naturally available physical source. Some commonly used sources are radioactive decay, the noise of a semiconductor diode, sound samples of a noisy environment, digitized videos of a lava lamp, etc. The outputs of these natural processes are truly random and therefore used for generating the random numbers. Some sources internal to the computer such as the timing of disk I/O response, memory, keystrokes, movement of hard disk heads, timing skew between the hardware and the software timers, etc., are also used for generating the true random numbers.

TRNGs are unpredictable, slow, not scalable, often biased and expensive. As per the recent research, the output of the ring oscillator has resulted in an excellent random bit generation. Yet, the design process appears to be complex and it may consume more time for the generations. The improvement of the TRNG with the modification of the Ring oscillator to the digital circuit is carried out through the multiplexer (MUX) and sequential circuits like Flip Flops. Digital circuits have been found to reduce energy and delay. In the semiconductor industry, time is an important factor for evaluating the chip design process. Hence, this section deals with the improvement of the TRNG with the modification of the Ring oscillator to the digital circuit through the multiplexer (MUX) and sequential circuits like Flip Flops (FFs), First In-First Out (FIFO), etc….

3.1 Conventional TRNG

As per the previous discussions, a ring oscillator based module based TRNG methods have been reviewed in this survey. The jitter is embedded inside the ring oscillators with an odd number of inverters in a circle routine. By this fashion, the values would change often because of twice of the delay in a single inverter with the total count. The jitter representation is shown in Fig. 1. There is a change in each cycle because of the oscillations. Generally, the oscillator based TRNG has been designed by combining the High and low-frequency oscillator modules, as shown in Fig. 2. As per the illustration, the signal frequency would change from the edges of the clock. With the addition of the D-Flip Flop (D-FF), the randomization concept could be observed. The D flip-flop is widely used in random generations and registers effectively as these are the types that are most commonly used. It holds more efficient way of CMOS implementation. Another major significance of the D-FF is that it can be used as a Shift register just for shuffling the random data's as in the order we want to as its output exactly follows the input. It is used for storing those random data and the same can be retrieved later.

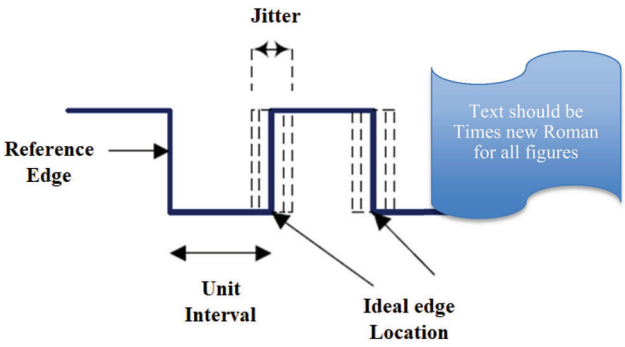


Figure 1: Representation of Jitter

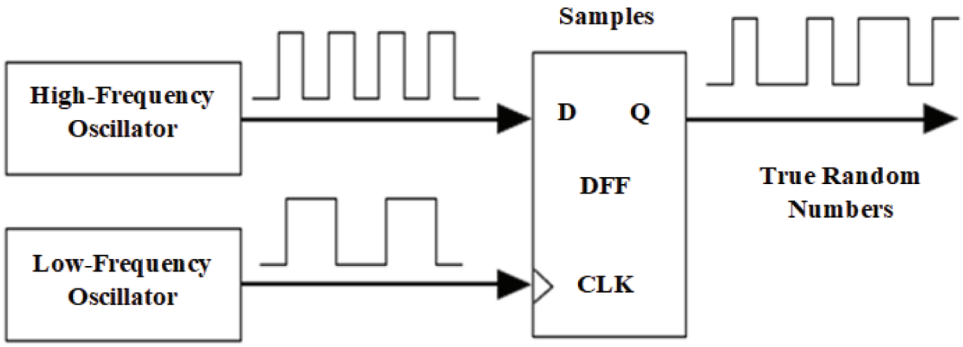


Figure 2: Oscillator based TRNG

Most of the TRNGs are modified by adding the ring oscillators by retaining its output to the XOR logic. Later, the obtained signal would be sampled with the DFF logic. The main drawback of such systems is the process of handling the switching activities in the oscillators. This case could be addressed by modifying the sampling stage with the DFF as shown in Fig. 3. They have proved that it has a minimum number of ring oscillators.

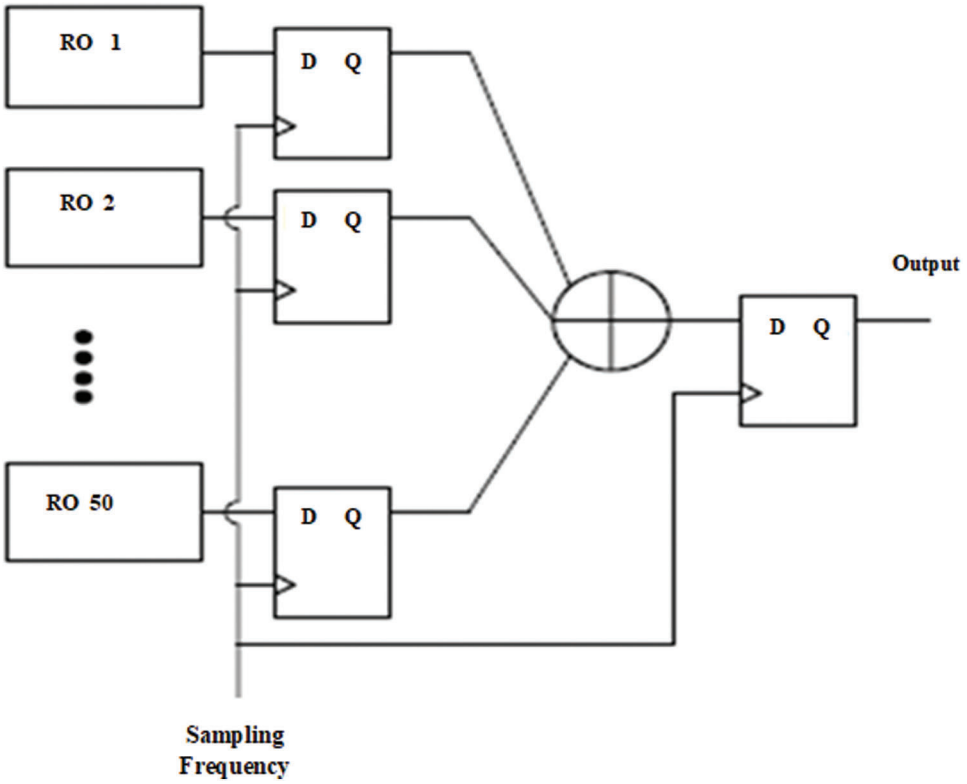


Figure 3: Word-based TRNG

The Physical Unclonable Function (PUF) and the TRNG would result in an efficient design frame for reusing the existing modules. The Proposed PUF-TRNG circuit design enables to have the same level of randomness without the need for an additional data post-processing, leading to a high throughput bit rate. Both the PUF and the TRNG operate within the quality requirements in terms of random bit throughput, uniqueness and reliability. This method is scalable and hence implemented for the complex structures. As shown in Fig. 4 the random generator module is modified with the output of the final XOR gate. It would be a TRNG bit stream or a single ring oscillator output that is subjected to the product register setting. In the PUF mode, the last yield would be taken care of to a counter for frequency estimation and afterward to the transport interface for examination. In the TRNG mode, the yield would be parallelized and put away into a memory-mapped register; it would read each of the examining clock cycles that are to be put away as an arbitrary piece stream in the off-chip memory.

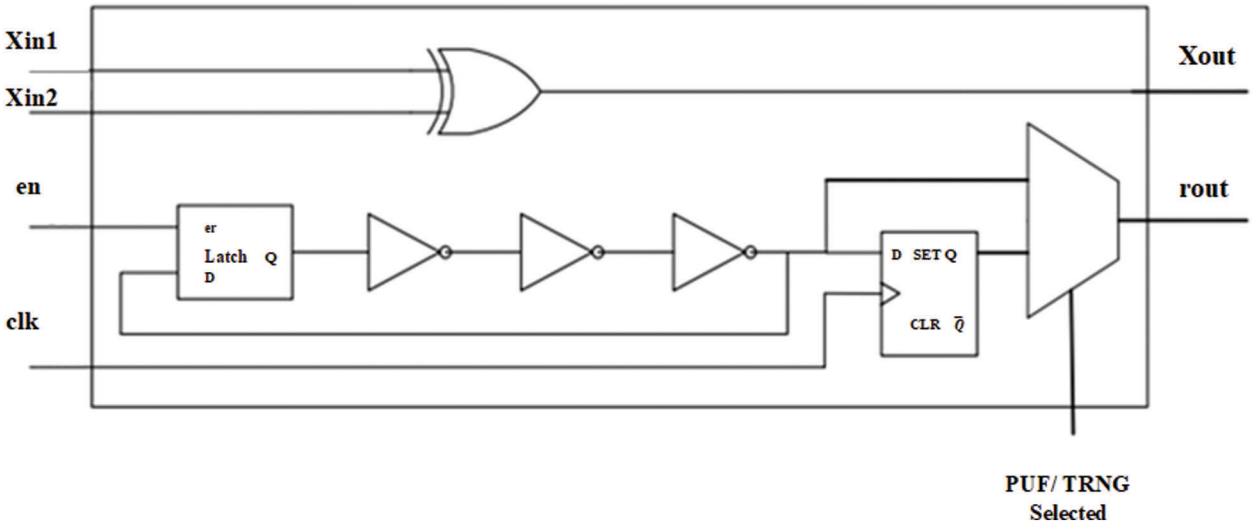


Figure 4: Physical unclonable function and TRNG module

3.2 Hybrid TRNG

The quality of the generated true random bits can be improved by modifying the TRNGs. Note that the ring oscillator based TRNGs, even though energizing, are very constrained regarding the randomness when indistinguishable ring oscillators are utilized. Equivalent length oscillator rings arranged in the FPGA are exceptionally related to one another due to the indistinguishable postponements and subsequently the XOR of the outcome from these rings returns for the most part zeros. This prompts poor arbitrariness from the plan. The proposed hybrid model has combined the encoding logic for replacing the inverter delays (d1, d2, d3) as, it could be controlled through the Pin Enable (en) as shown in the Fig. 5. The output of the encoder would provide the same result as that of the ring oscillator output. Then, the two stages of the D-FF are used for sampling the signals irrespective of the reset. The concept of applying the MUX is to select either the serial bit generator or the sampled signal. The combination of the single bit stream with the serial bit generator output would yield 8 bit, [7:0] as the resultant of the serial random sequence. Finally, the First In-First Out (FIFO) logic was deployed for generating the random bit sequence; it has been sequenced at the front in the order of the input and it would check the possible combinations as well.

The Hybrid TRNG comprises of the D-FF, the serial generator, the MUX, the FIFO, and a post-processing unit. To begin with, the control hardware begins with the encoder logic at the same time utilizing the ‘enable’ input. The DFF is at that point consolidated by the MUX and examined at various clocks, one at the reference clock and the other at 70.062 MHz. If the higher working frequency is utilized for the examination, then a ring oscillator may not be required. At that point, the units are programmable; various discrete levels are subjectively accepted for each of the inspecting clock. In this way, the examined bits are either taken care of at the post-processor unit or reasonably sent to the FIFO without post-processing. In this way, the outcome is either a simpler random bit stream selected using the control of the multiplexer and collected in squares of 8 bits utilizing the 8-bit register. 256 combinations are processed in a FIFO and sent back through a USB interface for the TRNG investigation. The FIFO logic would guide the generation of a random bit stream without any interference. Hence, the major outcome of the proposed logic would be carried out with low frequency without the use of a frequency divider. The obtained samples are collected from the TRNG and monitored continuously with various test cases, although it is necessary to write various test scenarios for verifying a design. The Hybrid TRNG

fully depends upon the digital circuit. Therefore, the natural test cases often checking the entropy would boost the repetition factor. The combination of the TRNG and the FIFO would generate additional test vectors and thus would assist in the completion of the verification process. Hence, the outcome would be non-deterministic.

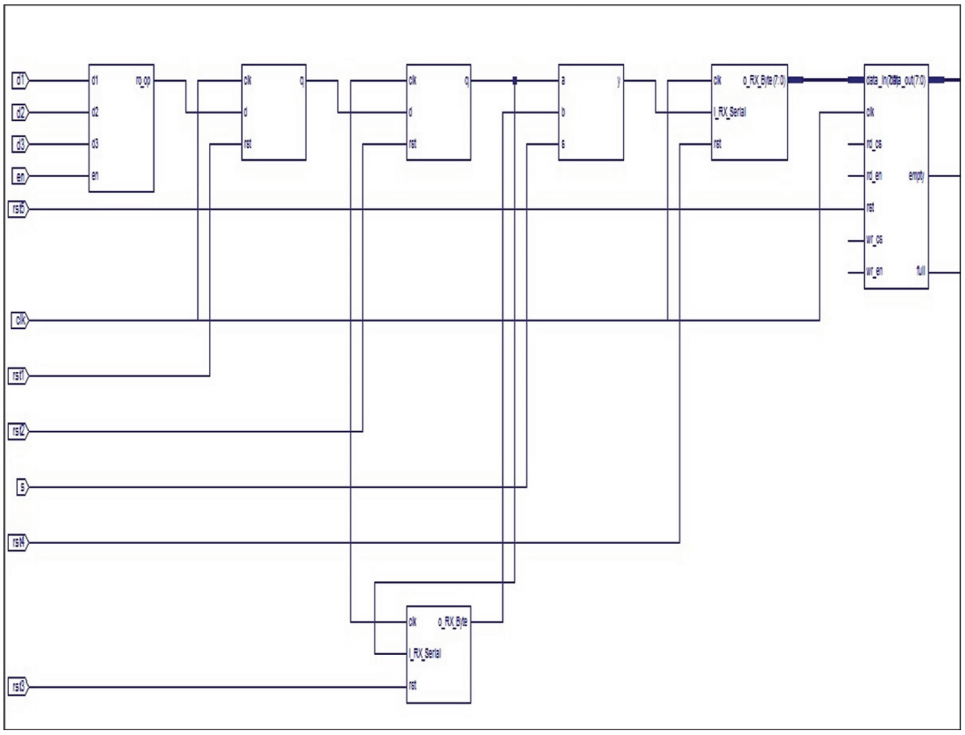


Figure 5: Proposed Hybrid TRNG

4 Experimental Results

The developed new Hybrid True Random Number Generator (HTRNG) without the Ring oscillator has generated a random bit sequence and has tested the same. Therefore, the algorithm is framed, implemented, verified and simulated using the XILINX ISE Design Suite. The quality of the HTRNG bit sequence is compared with that of the conventional random number generator methodologies, namely the TRNG with the Programmable Delays in the Oscillator-Ring proposed by Anandakumar et al. [20] and the Physical unclonable function and the true random number generator Maiti et al. [21]. The proposed method has been evaluated in Xilinx with a power consumption of 328.23 mW. The Look-Up Table (LUTs), Flip Flop (FFs), slices have been compared with the existing modules [22,23], which is shown in Tab. 2. The proposed method is completely implemented and tested with the XC2S600E platform.

Table 2: Parametric Comparison of the proposed HTRNG with the conventional methods

TRNG methodologies	LUTs	FFs	Slices	Power	Platform
TRNG with Programmable Delays in Oscillator-Rings	528	177	270	–	Spartan-3A (XC3S400A)
Physical unclonable function and TRNG	712	753	–	–	Spartan-3E (XC3S500E)
Proposed HTRNG	341	132	266	328.23 mW	XC2S600E

The proposed methodology is implemented in a Xilinx tool with a Verilog compiler, which is an open-source tool. Initially, the UCF file is generated and applied to the Xilinx ISE Design suite 14.2 for the bit stream file generation. Finally, the hardware module has been selected for implementing the proposed methodology. The logic synthesis is a procedure of observing the preferred circuit behavior, typically at the Register Transfer Level (RTL). The comparison of all the two conventional methods with the proposed method has been done based on the LUTs and the power FFs. The PUF and the TRNG has reported 712 LUTs. The proposed method has utilized half of the LUTs. Hence, the complexity would be reduced greatly. Initially, the ring oscillator is replaced with an encoding sequence as shown in Fig. 6. Later, the shift register logic is implemented and its resulted waveform is shown in Fig. 7.

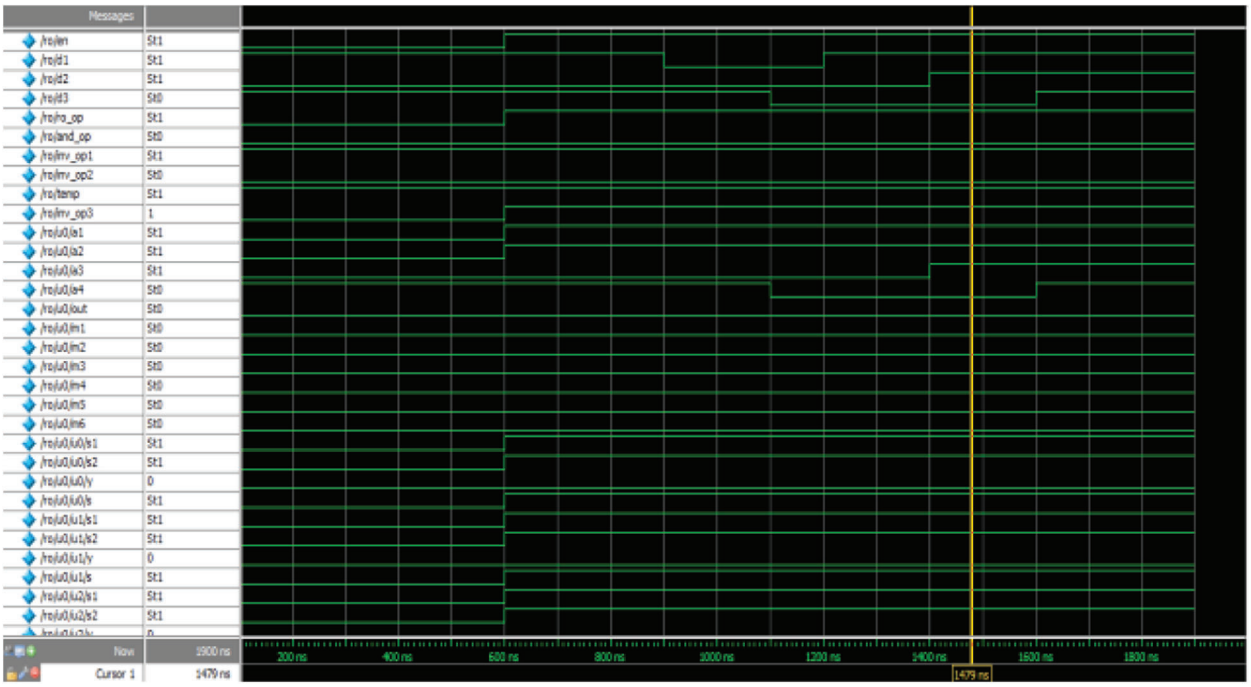


Figure 6: Initial sequence generator result

The RTL schematic for the proposed HTRNG architecture contains only one shift register, 2 DFF, a MUX and a FIFO. The post-processing procedure is not much complex. Hence, the randomization would result in reduced operating time and power. With the operating voltage of 1.80 v, the estimated power is 328.23 mW, as shown in Fig. 8. The control logic output would be processed with a control input Pin called the Enable (En). The results are shown in Fig. 9.

Finally, the post-processing signal would be generated as shown in Fig. 10. However, the byte level transactions are required for fulfilling the random sequence. This advancing methodology is processed into the Xilinx Spartan series with the target device of XC2S600E and the synthesis report is given in Fig. 11. It shows that about 341 LUTs have been consumed and 132 Flip Flops have been considered. The synthesis reports like Device utilization, logic distribution and memory locations are listed as shown in Fig. 11.

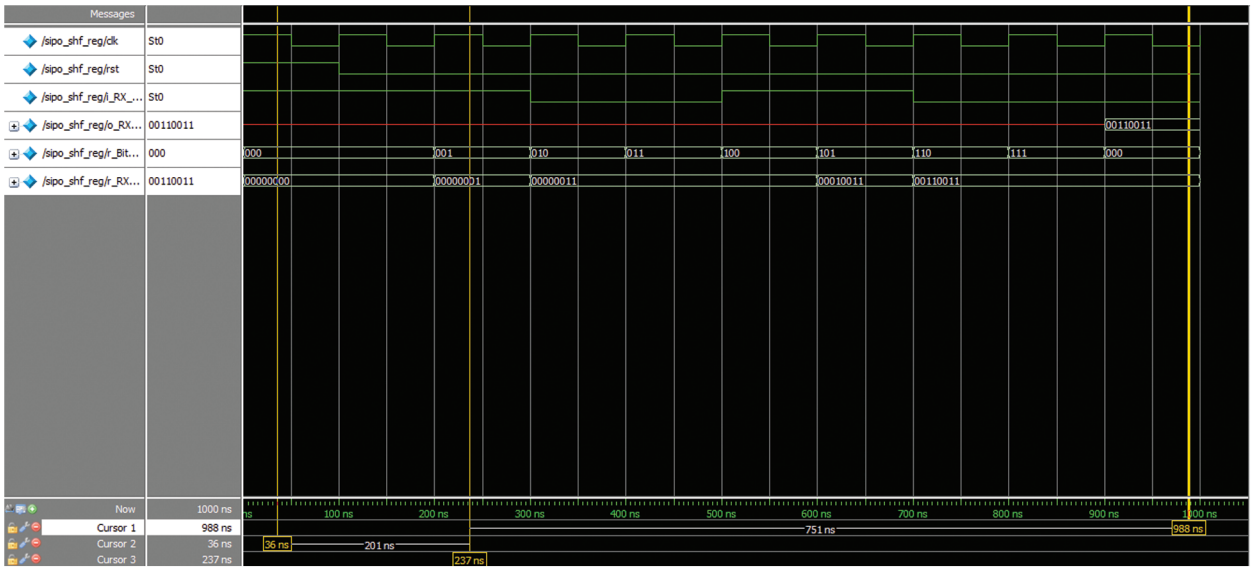
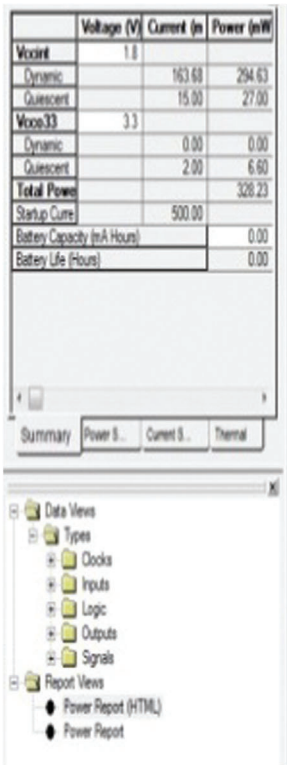


Figure 7: Shift Register outcome



Power summary:	I(mA)	P(mW)
Total estimated power consumption:		328
Vccint 1.80V:	179	322
Vcco33 3.30V:	2	7
Clocks:	155	279
Inputs:	8	15
Logic:	0	0
Outputs:		
Vcco33	0	0
Signals:	0	0
Quiescent Vccint 1.80V:	15	27
Quiescent Vcco33 3.30V:	2	7

Thermal summary:	
Estimated junction temperature:	30C
Ambient temp:	25C
Case temp:	30C
Theta J-A:	17C/W

Figure 8: Power consumption summary

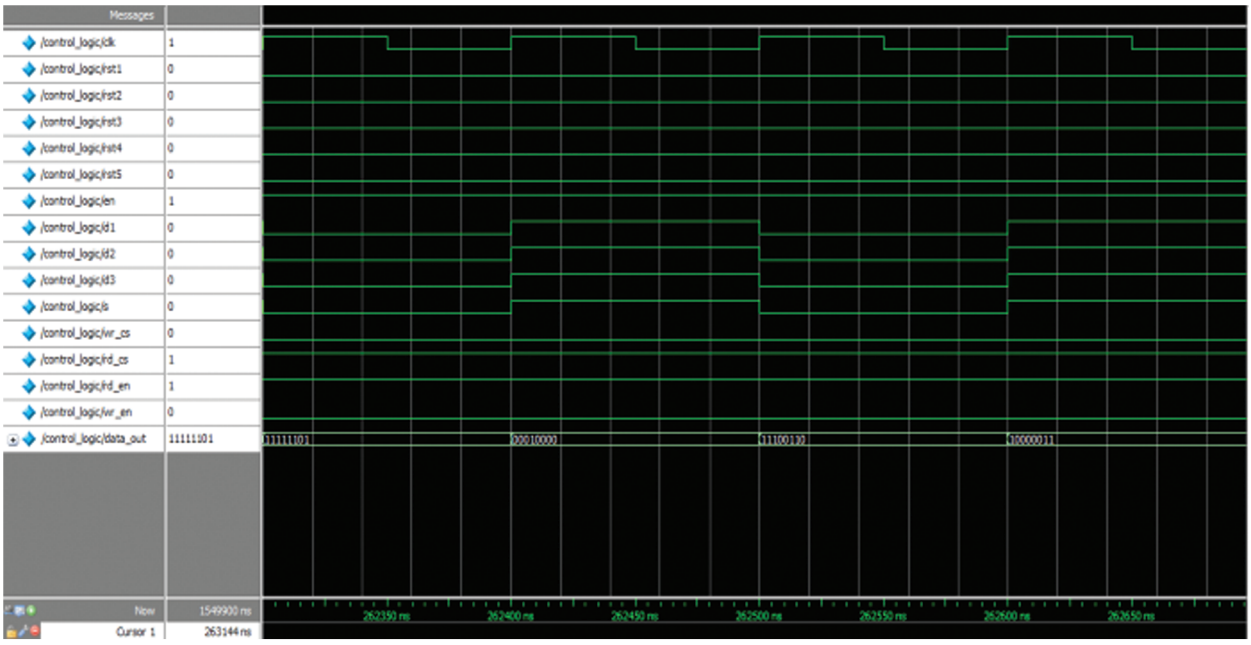


Figure 9: Control logic

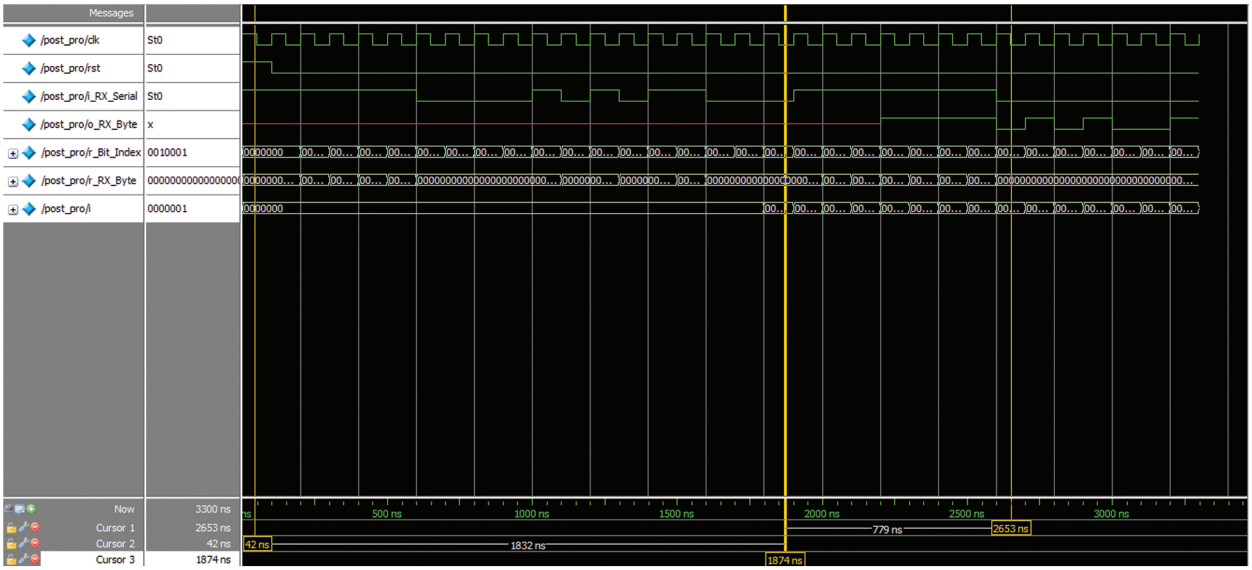


Figure 10: Post Processing sequence generation

To examine the relationship between the normal method with the additional control logic, shift register logic has been used in the proposed TRNG. An assessment was carried out to verify the random sequences, from various frequency conditions, for the proposed HTRNG design. The logics are sampled synchronously with respect to the reference clock. When multiple bits are required, the random logics are randomized into various slices and the LUTs could be further extended. The timing summary is shown in Fig. 12. This may cause some delay. The maximum clock and minimum clock delay will form jitter with an approximation of 8.565 ns. Meanwhile, the shift register delay would be noticed with a time delay of 14.273 ns.

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	132	13,824	1%	
Number of 4 input LUTs	341	13,824	2%	
Logic Distribution				
Number of occupied Slices	266	6,912	3%	
Number of Slices containing only related logic	266	266	100%	
Number of Slices containing unrelated logic	0	266	0%	
Total Number of 4 input LUTs	341	13,824	2%	
Number of bonded IOBs	15	510	2%	
Number of Block RAMs	1	72	1%	
Number of GCLKs	1	4	25%	
Number of GCLKIOBs	1	4	25%	
Total equivalent gate count for design	19,648			
Additional JTAG gate count for IOBs	768			

Figure 11: Utilization summary

```
Timing Summary:
---------------
Speed Grade: -7

   Minimum period: 14.273ns (Maximum Frequency: 70.062MHz)
   Minimum input arrival time before clock: 8.565ns
   Maximum output required time after clock: 8.129ns
   Maximum combinational path delay: No path found

Timing Detail:
--------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default period analysis for Clock 'clk'
  Clock period: 14.273ns (frequency: 70.062MHz)
  Total number of paths / destination ports: 5288 / 223
-------------------------------------------------------------------------
Delay:               14.273ns (Levels of Logic = 9)
  Source:            uu34/r_Bit_Index_1 (FF)
  Destination:       uu34/o_RX_Byte (FF)
  Source Clock:      clk rising
  Destination Clock: clk rising

  Data Path: uu34/r_Bit_Index_1 to uu34/o_RX_Byte
```

Figure 12: Delay analysis

5 Conclusion

The proposed HTRNG algorithm was implemented, analyzed and verified with different simulation levels. It maintains randomness with a simple hardware implementation than that of the past technique. The simulation has been done through the Xilinx where the hybrid TRNG technique achieves 14.273 ns delay and records the power to 328.23 mW and the complexity of the circuit is reduced when compared with that of the existing method. As of now, the RTL design process and its verification part have been accomplished. In future, if we tend to go with the synthesis and the chip level conversions, we can exactly measure the complexity. Based on the observations, the method is best suited for providing data security in the cryptographic/encryption applications. The random data which is generated would not be

repeated until it gets satisfied with all the combinations. Therefore, this research follows the cyclic nature of a random generation. Finally, the complexity of a circuit is greatly reduced by eliminating the ring oscillator and the hampers of all types of attacks. The schemes studied provide moderate data security, which can be found to be suitable for the commercial applications. The experimental results have proved that the proposed HTRNG is more efficient for testing the processor level designs. In the future, this work could be enhanced by incorporating other sequential circuits/arbitration logics, bit swapping, algorithmic concepts or machine learning algorithms that would direct and guide this randomization process with advancements. The key objective of those future works would be to reduce the delay and the area. Implementing bit swapping techniques could greatly improve the efficiency of the randomization techniques. The HTRNG implementation can also be considered in any one of the encryption standard based on the real-time applications.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Callegari, R. Rovatti and G. Setti, "Embeddable ADC-based true random number generator for cryptographic applications exploiting nonlinear signal processing and chaos," *IEEE Transactions on Signal Processing*, vol. 53, no. 2, pp. 793–805, 2005.

[2] V. Bagini and M. Bucci, "A Design of reliable true random number generator for cryptographic applications," in *Proc. Int. Workshop on Cryptographic Hardware and Embedded Systems*, Berlin, Heidelberg, Springer, pp. 204–218, 1999.

[3] A. Nath, S. Ghosh and M. A. Mallick, "Symmetric key cryptography using random key generator," in *Proc. International Conference on Security & Management, Las Vegas Nevada*, USA, pp. 234–242, 2010.

[4] Y. Hu, X. Liao, K. W. Wong and Q. Zhou, "A true random number generator based on mouse movement and chaotic cryptography," *Chaos, Solitons & Fractals*, vol. 40, no. 5, pp. 2286–2293, 2009.

[5] V. Fischer and M. Drutarovsky, "True random number generator embedded in reconfigurable hardware," in *Proc. Int. Workshop on Cryptographic Hardware and Embedded Systems*, Berlin, Heidelberg, Springer, pp. 415–430, 2002.

[6] B. Sunar, W. J. Martin and D. R. Stinson, "A provably secure true random number generator with built-in tolerance to active attacks," *IEEE Transactions on Computers*, vol. 56, no. 1, pp. 109–119, 2007.

[7] P. Kohlbrenner and K. Gaj, "An embedded true random number generator for FPGAs," in *Proc. ACM/SIGDA Int. Sym. on Field Programmable Gate Arrays*, USA, pp. 71–78, 2004.

[8] C. Tokunaga, D. Blaauw and T. Mudge, "True random number generator with a metastability-based quality control," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 78–85, 2008.

[9] H. Jiang, D. Belkin, S. E. Savel'ev, S. Lin, Z. Wang *et al.,* "A novel true random number generator based on a stochastic diffusive memristor," *Nature Communications*, vol. 8, no. 1, pp. 403, 2017.

[10] A. P. Johnson, R. S. Chakraborty and D. Mukhopadyay, "An improved DCM-based tunable true random number generator for Xilinx FPGA," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 4, pp. 452–456, 2017.

[11] I. Vasyltsov, E. Hambardzumyan, Y. S. Kim and B. Karpinskyy, "Fast digital TRNG based on metastable ring oscillator," in *Proc. Int. Workshop on Cryptographic Hardware and Embedded Systems*, Berlin, Heidelberg, Springer, pp. 164–180, 2008.

[12] P. Bayon, L. Bossuet, A. Aubert and V. Fischer, "Electromagnetic analysis on ring oscillator-based true random number generators," in *Proc. IEEE Int. Sym. on Circuits and Systems*, France, pp. 1954–1957, 2013.

[13] D. Liu, Z. Liu, L. Li and X. Zou, "A low-cost low-power ring oscillator-based truly random number generator for encryption on smart cards," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 63, no. 6, pp. 608–612, 2016.

[14] S. Robson, B. Leung and G. Gong, "Truly random number generator based on a ring oscillator utilizing last passage time," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 12, pp. 937–941, 2014.

[15] A. T. Markettos and S. W. Moore, "The frequency injection attack on ring-oscillator-based true random number generators," in *Proc. Int. Workshop on Cryptographic Hardware and Embedded Systems*, Berlin, Heidelberg, Springer, pp. 317–331, 2009.

[16] K. Itaya and Y. Jitsumatsu, "Random number generation using outputs from multiple beta encoders," in *Proc. Int. Sym. on Nonlinear Theory and Its Applications (NOLTA)*, Yugawara, Japan, pp. 249–252, 2016.

[17] C. Gabriel, C. Wittmann, D. Sych, R. Dong, W. Mauerer *et al.,* "A generator for unique quantum random numbers based on vacuum states," *Nature Photonics*, vol. 4, no. 10, pp. 711–715, 2010.

[18] X. Ma, F. Xu, H. Xu, X. Tan, B. Qi *et al.,* "Postprocessing for quantum random-number generators: Entropy evaluation and randomness extraction," *Physical Review A*, vol. 87, no. 6, pp. 062327, 2013.

[19] Y. Cao, E. C. Chidiebere, C. Fang, M. Zhou *et al.,* "Silicon-based true random number generators," in *Frontiers in Hardware Security and Trust: Theory, design and practice*, pp. 115, 2020.

[20] N. N. Anandakumar, S. K. Sanadhya and M. S. Hashmi, "FPGA-based true random number generation using programmable delays in oscillator-rings," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 3, pp. 570–574, 2020.

[21] A. Maiti, R. Nagesh, A. Reddy and P. Schaumont, "Physical unclonable function and true random number generator: A compact and scalable implementation," in *Proc. ACM Great Lakes Sym. on VLSI*, USA, pp. 425–428, 2009.

[22] N. Krishnaraj, M. Elhoseny, E. L. Lydia, K. Shankar and O. ALDabbas, "An efficient radix trie-based semantic visual indexing model for large-scale image retrieval in cloud environment," *Software: Practice and Experience*, vol. 51, no. 3, pp. 489–502, 2021.

[23] N. Krishnaraj, M. Elhoseny, M. Thenmozhi, M. M. Selim and K. Shankar, "Deep learning model for real-time image compression in Internet of Underwater Things (IoUT)," *Journal of Real-Time Image Processing*, vol. 17, no. 6, pp. 2097–2111, 2020.