International Journal of Scientific Research in Science, Engineering and Technology

Print ISSN - 2395-1990
Online ISSN : 2394-4099

OPEN ACCESS

Available Online at : www.ijsrset.com
doi : https://doi.org/10.32628/IJSRSET

A Power-Efficient FPGA Test Pattern Composer

K. Pasipalana Rao[1], Ch. Hari Chandrika[2], G. Ramya Krishna[3], G. Anitha Iswarya[4]

[1]Assistant Professor, Department of Electronics and Communication Engineering, Sri Vasavi Engineering, College, Tadepalligudem, Andhra Pradesh, India

[2,3,4] UG Student, Department of Electronics and Communication Engineering, Sri Vasavi Engineering, College, Tadepalligudem, Andhra Pradesh, India

ARTICLEINFO

Article History:

Accepted: 05 April 2024
Published: 19 April 2024

Publication Issue :
Volume 11, Issue 2
March-April-2024
Page Number :
333-341

ABSTRACT

The paper presents a novel approach to generating true random number sequences in Xilinx hardware using the random jitter of free-running oscillators. By employing programmable delay lines, the proposed method aims to reduce correlation between oscillator rings, enhancing randomness. Post- processing techniques such as Von-Neumann correction are applied to refine the generated sequences. Additionally, clock gating architecture is utilized to improve power efficiency by reducing switching activity. Notably, the design omits data and read/write lines in SRAM memory architecture due to LUT memory accessing not requiring write operations.

Keywords : LUT, SRAM, Von-Neumann Corrector, True-Random Number Generator, Clock gating, Random Jitter, and Xilinx.

I. INTRODUCTION

Introduction to Randon Number Generators and their importance: In today's world, computer systems and telecommunications are integral to various aspects of life, facilitating data transfer, personal tracking, online transactions, and communication. However, with the increasing reliance on digital means, the risk of data breaches and unauthorized access is heightened. Hence, there's a pressing need for robust methods and technologies to secure data, ensuring its confidentiality, integrity, and authenticity.

Types of Random Number Generators: Random number generation is essential for ensuring the privacy of electronic communications, forming a crucial part of encryption processes that render data unreadable to unauthorized users. The strength of encryption relies heavily on the randomness of binary numbers, necessitating the development of efficient random number generators (RNGs) to support secure cryptographic systems. Beyond cybersecurity, RNGs find widespread utility in various domains, including computer simulations, statistical sampling, and commercial applications like gaming. They play pivotal roles in authentication, secret key generation, game theory, and simulations within computer science, underscoring their significance beyond encryption.

Applications and Improvements in RNG Technology: In various applications, random numbers must exhibit good statistical properties and be both unpredictable and non-reproducible. Random number generation methods are typically categorized as deterministic or nondeterministic. Pseudo Random Number Generators (PRNGs), which are deterministic, offer fast, easy, and cost- effective solutions that are hardware-independent. PRNGs must meet specific requirements to be suitable for tasks such as authentication and key generation, often necessitating the addition of nondeterministic functions to ensure these criteria are met. On the other hand, True Random Number Generators (TRNGs), which are nondeterministic, provide slower and more expensive solutions that are hardware-dependent. Unlike PRNGs, TRNGs do not require additional components to meet certain requirements (R2, R3, R4). The high entropy and unpredictability of random numbers generated by TRNGs typically satisfy the R2 requirement, which implies that R3 and R4 requirements are also met.

Enhancements in TRNG Post-Processing: To enhance the quality of random numbers generated by True Random Number Generators (TRNGs), post-processing techniques are applied to eliminate statistical weaknesses and potential vulnerabilities. Recent studies have explored novel approaches to random number generation, including using human-based noise sources.

Advancements in Biomedical and Behavioral Signal-Based RNGs: Several studies have explored the use of physiological signals, such as mouse movements, ECG, and EEG, for random number generation. These signals were subjected to statistical analysis using the NIST test suite to assess their suitability for cryptographic applications. Successful results were obtained, indicating the potential of these signals for generating random numbers. Additionally, various approaches, including chaos-based methods and PRNG-based transformations, were proposed to improve the randomness and statistical properties of the generated numbers. These findings suggest promising avenues for utilizing physiological signals in random number generation for low-cost and real-world applications.

Evolution of Electronic Technologies and Their Impact on Modern Life: Modern cryptographic systems rely on random number generators (RNGs) to generate secure and unpredictable keys. These RNGs utilize physical phenomena, such as clock jitter in digital devices, to ensure randomness. Monitoring and utilizing jitter parameters are essential for estimating the unpredictability of generated numbers, as outlined in security standards like AIS-20/31 by the German Federal Office for Information Security (BSI). The evolution of electronics has had a profound impact on various fields, revolutionizing everyday life. Transistors, integrated circuits, and advancements like Moore's law have driven the rapid progress in semiconductor technology, enabling the fabrication of highly complex microprocessors with billions of transistors. These technological advancements have reshaped communication, media, transportation, and other aspects of human life.

The Critical role of TRNGs in Enhancing Cryptographic Security: True random number generators (TRNGs) are crucial for cryptographic applications due to their ability to generate truly random numbers using physical entropy sources. However, they often suffer from high power consumption and design complexity. The recently proposed TRNG based on Random Telegraph Noise (RTN) offers advantages such as low power and robustness for IoT applications but faces challenges like device selectivity and low speed. Despite these limitations, TRNGs play a vital role in cryptography, ensuring secure key generation and authentication protocols. Poor random number generators can compromise system security, highlighting the importance of TRNGs in cryptographic systems.

Assessment of TRNG randomness typically involves statistical tests like Diehard and NIST, along with entropy estimation per bit using stochastic models.

II LITERATURE SURVEY

Landauer's principle, proposed in 1961, states that irreversible circuits dissipate energy when information is lost, with a minimum energy dissipation of KTln2 joules per lost bit, where K is Boltzmann's constant and T is absolute temperature. In 1973, Bennett demonstrated that reversible logic, allowing the reproduction of inputs from outputs, can mitigate this energy dissipation. He showed that reversible systems can perform computations with the same efficiency as irreversible systems, leading to the development of reversible logic-based systems. Reversible gates must have an equal number of inputs and outputs to uniquely recover inputs from outputs at any point in time. A paper by Shibinu A.R and Rajkumar presents a 4-bit LFSR design using Muller expressions. The paper also introduces realizations of edge-triggered and level-triggered D flip-flops using reversible logic. A comparative analysis between conventional LFSR and Reversible LFSR demonstrates that the proposed technique is more efficient in terms of power, quantum cost, garbage output, and gate count. [3]. Muthih and Arockia Bazil Raj introduced a parallel architecture for high-speed LFSR, focusing on its application in BCH encoders and CRC operations. Their approach includes a linear transformation algorithm for converting serial LFSRs to parallel architecture and proposes enhancements through pipelining and retiming algorithms. [4]. The paper presents optimized design approaches for reversible D flip- flops with asynchronous set/reset, focusing on quantum cost, delay, and garbage outputs. It also discusses the design of a 3-bit LFSR and its application as a pseudo-random bit sequence generator, concluding with a comparative analysis of the proposed approaches. [5].Research paper [6] introduces automated techniques for minimizing layout area and power consumption in LFSR and D flip-flop implementations, crucial for IC self-testing. Paper [7] proposes a new data compression approach to reduce power dissipation in LFSR-based schemes, focusing on designing an 8-bit LFSR using reversible logic for low-power applications, emphasizing the importance of minimizing power dissipation in VLSI testing through BIST. BIST uses LFSR for pattern generation, offering improved area testability and cost-effectiveness compared to other methods.

Authors propose power reduction techniques like clock gating in LFSR, achieving significant reductions in power consumption, particularly in the Test Pattern Generation phase. Paper [10] presents an LFSR design aimed at reducing power dissipation during testing, achieved by minimizing transitions between successive patterns, leading to a 44.6% reduction in dynamic power. In [11], a stepped segment LFSR is proposed to address high power dissipation during testing in BIST, resulting in a 19.63% reduction in testing power with minimal impact on fault coverage. In [12], LFSR design in sub-threshold regime utilizes various flip-flop and XOR gate configurations, with TSPC-based D flip- flops showing superior performance in terms of operating frequency and power consumption, operating in the nano watt range. In [17], a low- power test pattern generator employs Low Transition Generalized Linear Feedback Shift Registers, reducing transitions between patterns and achieving significant dynamic power reduction through bit swapping and flip-flop bypassing. In [13], an LP- LFSR-based test pattern generator reduces switching activities through counter, gray code generator, and pipelined Braun array multiplier, aiming to decrease power consumption during testing. In [14], a lTohwe-power LFSR using gate diffusion input (GDI) technique minimizes hardware and power dissipation, achieving a reduction of 20% in hardware and 45.4% in power consumption, enhancing static power characteristics and logic level swings.

III IMPLEMENTATION METHODOLOGY

Existed Methods:

RO-based TRNGs face limitations in randomness when identical ROs are used due to correlated outputs. Incorporating PDLs in TRNGs can mitigate this issue, overcoming poor randomness, while flip-flop meta-stability was previously utilized for true random number generation. Our method introduces true randomness by leveraging the random jitter of free-running oscillators, facilitated by PDLs to equalize signal arrival times. By incorporating PDLs in oscillator rings, we induce large oscillation variations and introduce jitter, reducing correlation between oscillator outputs and enhancing randomness. This approach also introduces cycle-to-cycle variation in oscillator oscillations, further improving randomness through XOR operations.

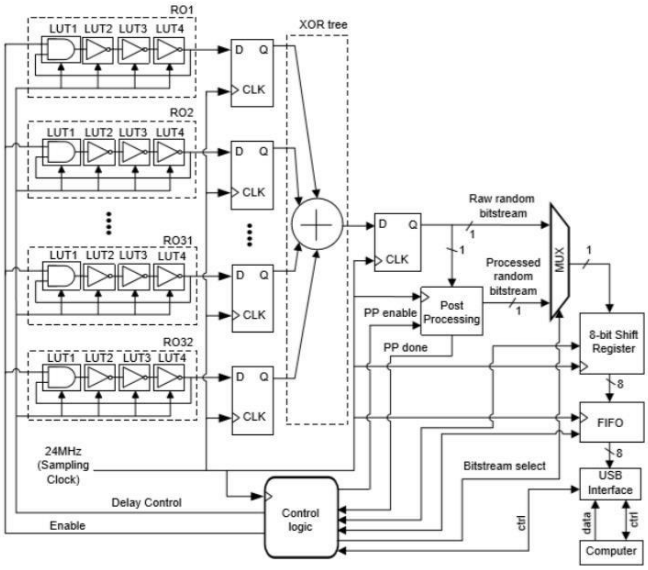

Fig. 1: Architecture of the Existing TRNG

The TRNG architecture comprises 32 ROs realized with 3 inverters and 1 AND gate each, controlled by programmable delays. RO outputs are XORed, sampled at 24 MHz, and processed or stored directly in a FIFO. Control circuitry manages RO activation, sampling, delay configuration, and data transfer to a PC via USB for statistical analysis.

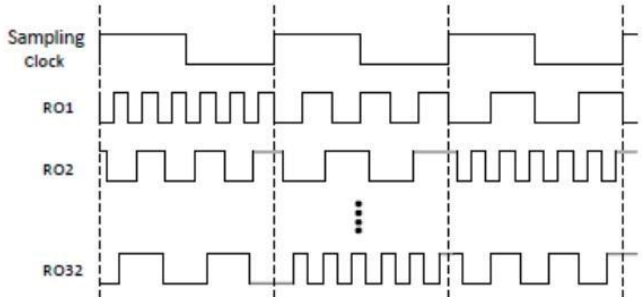

Fig 2: RO outputs for each sampling clock by using PDL

Programmable Delay Lines (PDLS)

Programmable delay lines introduce precision delay or phase shifts in high-frequency signals, achieved through electromechanical trombones or digital elements like circular buffers. Digital delay lines compensate for signal delays, with fractional delay lines handling non-integer delays. Analog delay lines, like bucket-brigade devices, provide continuous signal delay and are used in various applications, including PAL television standards and echo simulation. Before random access memory, delay line memory converted electrical pulses into sound waves, enabling data storage and processing during World War II radar improvements. FPGA Look-Up Tables (LUTs) leverage internal variations in propagation delays to offer diverse functionalities under different inputs.

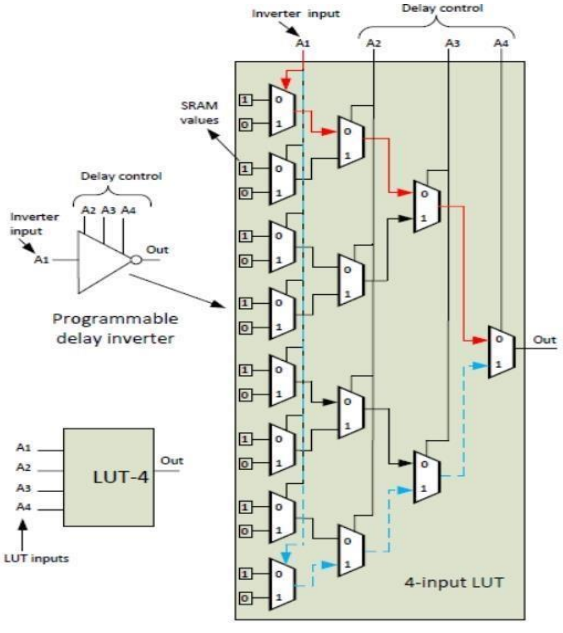

Fig 3: PDL using a 4-input LUT

This paper introduces a novel programmable delay inverter using a 4-input LUT. The inverter's output mirrors its first input (A1), while others (A2, A3, A4) serve as "don't-care" bits, influencing signal path. Experimentation shows shortest path when A2A3A4 = 000 and longest for A2A3A4 = 111. Thus, a delay inverter with 3 control inputs is achieved using 1 LUT, offering 8 discrete delay levels.

2. Look-Up Table:

In computer science, lookup tables serve to streamline runtime computations by replacing them with simple array indexing operations, significantly reducing processing time. They can be precalculated and stored in static program memory, calculated during initialization (memorization), or even implemented in hardware. Lookup tables validate input values against a list of valid items and are integral in FPGA configurations. Historically, they expedited complex hand calculations, such as trigonometric functions. Today, they're ubiquitous, from aiding school children in memorizing multiplication tables to enhancing spreadsheet functionalities like Excel's LOOKUP functions. Despite advancements in caching, application-level lookup tables continue to enhance performance for infrequently changing data.

Input A	Input B	Output C
0	0	0
0	1	0
1	0	0
1	1	1

Table 1: AND table

A Lookup Table (LUT) in an FPGA functions akin to a customized truth table, stored in a small RAM. Address pins A and B select entries, while pin C reads out the result based on the inputs. Essentially, the LUT serves as a directive for the chip's combinatorial logic, defining behaviour according to predetermined values. As inputs act as address lines for RAM cells, a two-input logic gate corresponds to a 4 x 1 bit RAM, scalable with increasing inputs. LUTs replace the need for multiple logic gates, simplifying circuit design. Within an FPGA, configurable logic blocks contain LUTs, crucial for chip functionality. LUTs store output values (LUT- Mask) in SRAM bits, enabling rapid computation without excessive gate usage. Moreover, SRAM's configurability enhances FPGA versatility, allowing for reconfiguration with each power cycle, a defining feature of FPGA chips.

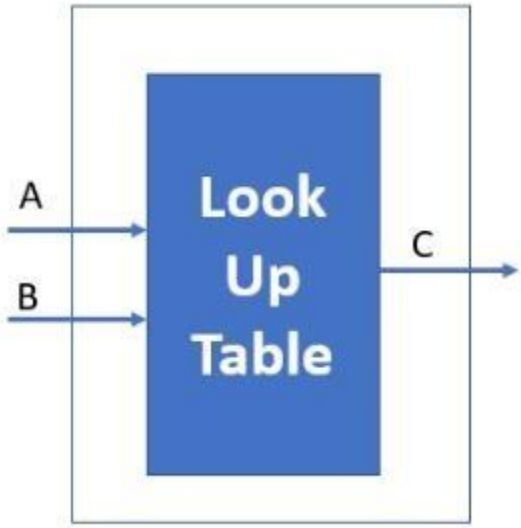

Fig 4: Look Up Table

3. Ring Counter:

A ring counter, comprised of a circular shift register, circulates data from the output of the last shift register to the input of the first. There are two main types: straight and twisted. A straight ring counter, also known as an Overbeck counter, circulates a single one (or zero) bit around the ring. In contrast, a twisted ring counter, or Johnson counter, circulates a sequence of ones followed by zeros. In both cases, the data pattern within the shift register recirculates with each clock pulse. However, proper operation requires pre-loading one register with a 1 (or 0) for a straight ring counter, or initializing all zeros for a twisted ring counter. Simply loading all 0's or 1's does not constitute a valid pattern.

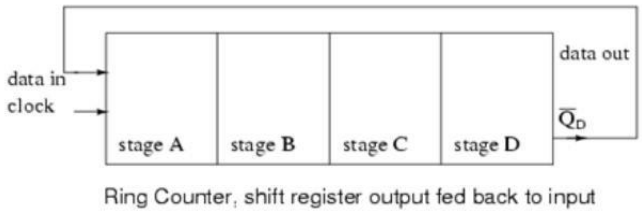

Fig 5 : Ring Counter using Shift Register

We make provisions for loading data into the parallel-in/ serial-out shift register configured as a ring counter below. Any random pattern may be loaded. The most generally useful pattern is a single 1.

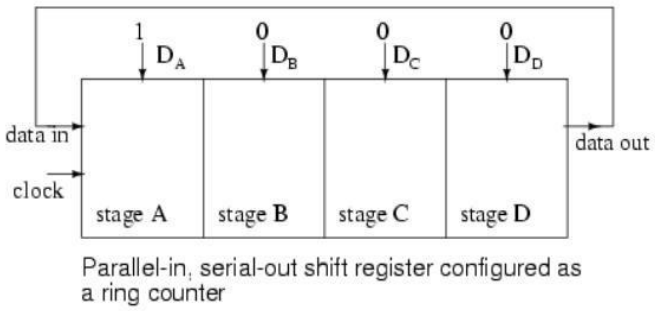

Fig 6 : Ring Counter using PISO

4. Post Processing

The existing implementation employs a simple post processing unit based on a Von Neumann corrector for enhancing the entropy and for removing any bias in the generated random bits. The post-processor also provides robustness of the TRNG output sequence. The employed Von Neumann corrector post-processing scheme is depicted in Fig. We read two bits at a time from the raw TRNG output and discard them if both of them are same (i.e. we eliminate 00 and 11 patterns). If the two bits are different (i.e. 01 or 10) then we take the first bit and discard the second bit. On an average, the post-processing unit requires 512 bits of raw input to generate 128 bits of post-processed output.

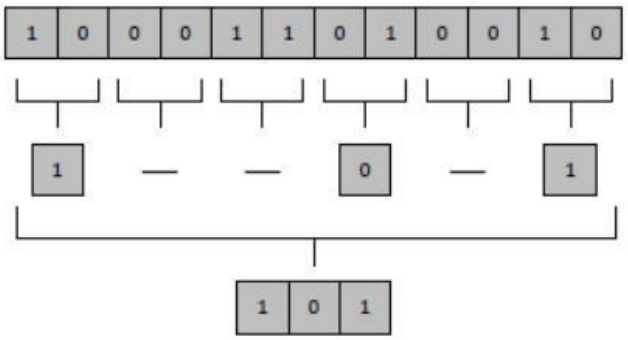

Fig 7: Principle of operation of Von Neumenn

Proposed Methods:

1. Random Access Memory (RAM):

The full form of RAM is Random Access Memory. The information stored in this type of memory is lost when the power supply to the PC or laptop is switched off. The information stored in RAM can be checked with the help of BIOS. It is generally known as the main memory or temporary memory or cache memory or volatile memory of the computer system.

A memory unit is a collection of storage cells together with associated circuits needed to transform information in and out of the device. Memory cells which can be accessed for information transfer to or from any desired random location is called random access memory (RAM).

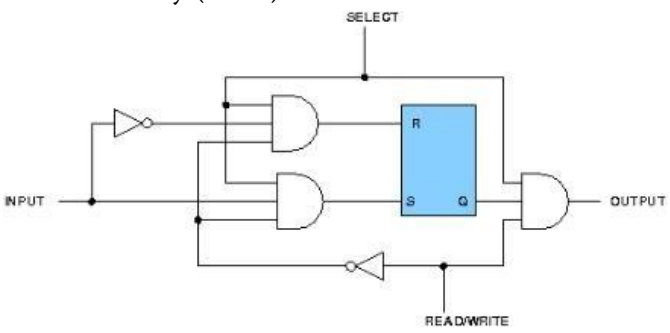

Fig 8: RAM Cell

This section outlines PJMEDIA's implementation of a delay buffer, akin to a fixed jitter buffer, delaying frame retrieval to ensure continuous playback. It proves valuable in scenarios where operations are

unevenly interleaved, accommodating bursts of put() followed by get() operations. The buffer adapts dynamically, learning and applying an optimal delay to the audio flow during runtime. It adjusts audio sample rates as needed, maintaining a consistent flow even when buffer levels fluctuate.

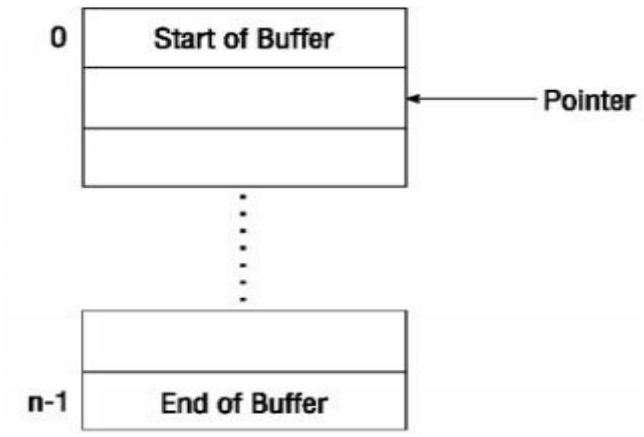

Fig 9: Buffer

2. Technique:

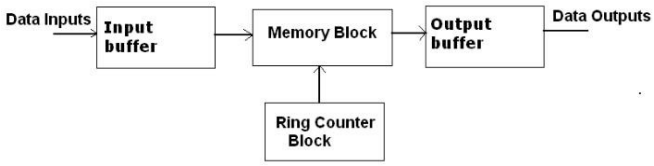

Fig 10 : Existing Block Diagram of Memory Organization

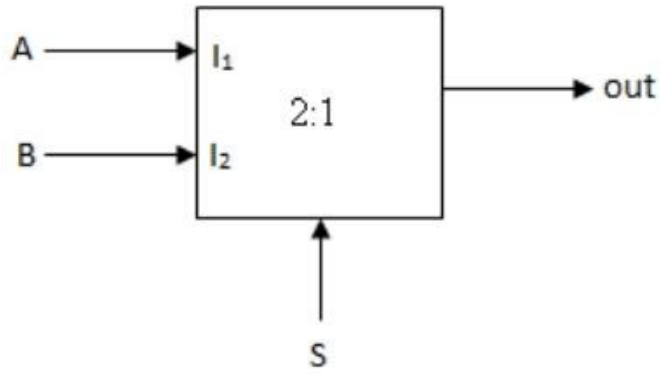

Fig 11 : Input Buffer

3. Input Buffer:

The input buffer, also known as the input area, temporarily stores incoming data before CPU processing. It's vital in managing data flow in computer memory and various hardware or software

S	I 1	I 2	Out
0	A	B	A
1	A	B	B

Truth Table for 2:1 Mux

4. Memory Block:

RAM, pivotal in computer data storage, allows random access to stored data. Serial scan designs in testability induce power dissipation and longer test times. Random Access Scan (RAS), resembling RAM, reduces state setting times but requires more resources. Recent research explores RAS as a DFT method, addressing serial scan limitations.

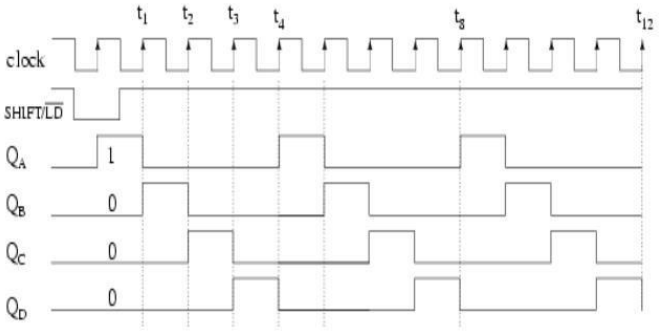

Fig 12: Loading 1000 into 4-stage ring counter and shifting

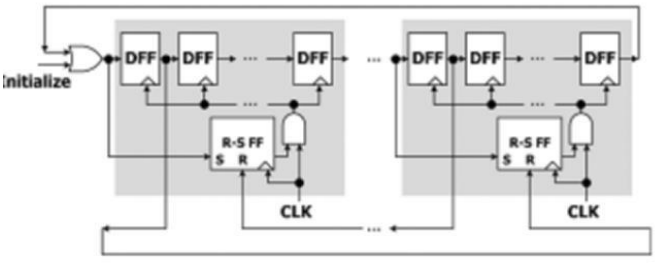

Fig 13: Ring Counter with SR Flip-Flop

IV COMPARISON RESULTS AND DISCUSSION

The output is either raw random bit stream or processed random bit stream selected via control input of the multiplexer and collected in blocks of 8 bits using the 8-bitshift register. Finally, each byte is stored in a FIFO of 64 by 8 width (i.e.512 bits) and sent to PC through a USB interface for the TRNG statistical analysis. The FIFO allows reading of raw/processed random bit stream without flow interruption. The control logic module enables the start and stop of the RO's, FIFO, 8-bit shift register, post-processing unit, and selection of the raw/processed random bit stream for transfer to the PC.

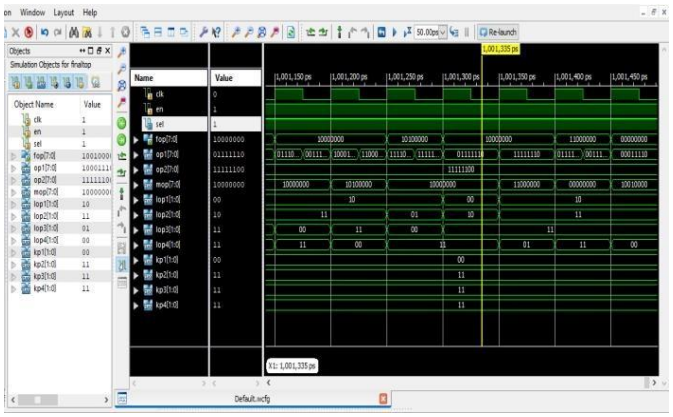

Fig 14: Existing simulation output

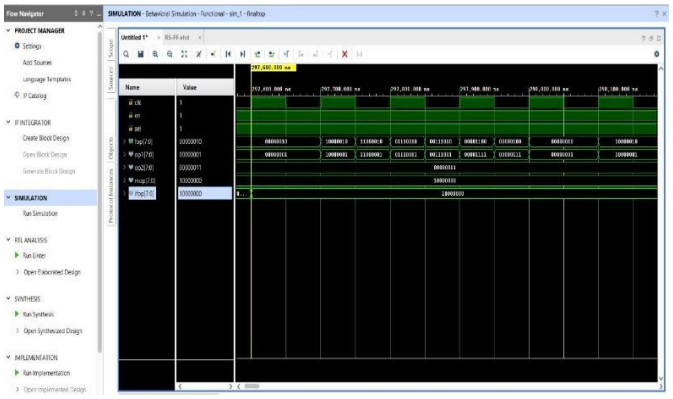

Fig 15: Proposed simulation output

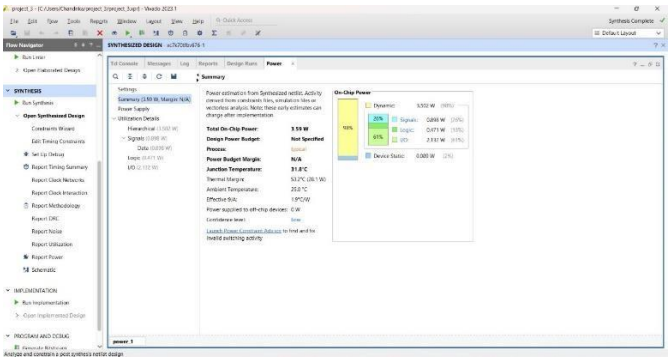

Fig 16: Existing Power Output

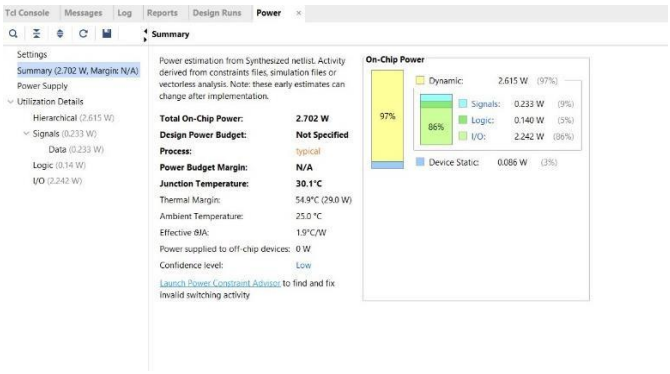

Fig 17: Proposed Power Output

V. CONCLUSION

A new design of RO-based TRNG is described and its implementation on Xilinx is presented in this project. The programmable delay of FPGA LUTs has been used to achieve random jitter and to enhance the randomness It has been demonstrated that the proposed implementation provides a very good area-throughput trade-off. Effectively, it is capable of producing a more throughput after post-processing with a low hardware footprint. In addition, the restart experiments show that the output of the proposed TRNG behaves truly random.

II. REFERENCES

[1]. K. Nohl, D. Evans, S. Starbug, and H. Plotz, "Reverse-engineering a ¨ Cryptographic RFID Tag," in Proceedings of the 17th Conference on Security Symposium. USENIX Association, 2008, pp. 185–193.

[2]. G.Marsaglia, “Diehard: A Battery of Tests of Randomness,” 1996.

[3]. Shibinu A.R , Rajkumar. et. al, “Implementation of power efficient 4-bit reversible linear feedback shift register for BIST,” Tech. Rep., 2010.

[4]. D. Muthih and A. Arockia Bazil Raj, “mplementation of high-speed LFSR design with parallel architectures,” 2014.

[5]. Jayasanthi M, Kowsalyadevi AK, “Low Power Implementation of Linear Feedback Shift Registers “,2019

[6]. M. Majzoobi, F. Koushanfar, and S. Devadas, “FPGA-Based True Random Number Generation Using Circuit Metastability with Adaptive Feedback Control,” in Cryptographic Hardware and Embedded Systems – CHES 2011. Springer Berlin Heidelberg, 2011, pp. 17–32.

[7]. H. Hata and S. Ichikawa, “FPGA Implementation of Metastability-Based True Random Number Generator,” IEICE Transactions on Information and Systems, vol. E95.D, no. 2, pp. 426–436, 2012.

[8]. A. P. Johnson, R. S. Chakraborty, and D. Mukhopadhyay, “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, April 2017.

[9]. 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, June 2016.

[10]. A. Beirami and H. Nejati, “A Framework for Investigating the Performance of Chaotic-Map Truly Random Number Generators,” IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 60, no. 7, pp. 446–450, July 2013.

[11]. J. von Neumann, “Various techniques used in connection with random digits,” in Monte Carlo Method. National Bureau of Standards Applied Mathematics Series, 12, 1951, pp. 36–38. [12] 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, Jan 2007. [13] M. Dichtl and J. D. Golic, “High- Speed True Random Number Genera- ´ tion with Logic Gates Only,” in Cryptographic Hardware and Embedded Systems - CHES 2007. Springer Berlin Heidelberg, 2007, pp. 45–62.

[12]. Harshitha G; Kishore E J; Manoj R; Priyanka R Devarmani; Praveen Kumar Y G; M Z Kurian, “Gate-Diffusion Input based Linear Feedback Shift Register : A Review,” in 2092 .

[13]. K. Wold and C. H. Tan, “Analysis and Enhancement of Random Number Generator in FPGA Based on Oscillator Rings,” in Int. Conf. on Reconfigurable Computing and FPGAs, Dec 2008, pp. 385–390.

[14]. O. Petura, U. Mureddu, N. Bochard, V. Fischer, and L. Bossuet, “A survey of ais 20/31 compliant trng cores suitable for fpga devices,” in 2016 26th International Conference on Field Programmable Logic and Applications (FPL), Aug 2016, pp. 1–10.

[15]. N. Bochard, F. Bernard, V. Fischer, and B. Valtchanov, “TrueRandomness and PseudoRandomness in Ring Oscillator-Based True Random Number Generators,” Int. J. Reconfig. Comp., vol. 2010, pp. 879 281:1–879 281:13, 2010.