Military Institute of Science and Technology

DEPARTMENT OF ELECTRICAL ELECTRONIC AND COMMUNICATION ENGINEERING

COURSE: EECE 458

PROJECT: Comparative Study of Stochastic Number Generator Using Modified NLFSR and WBG in 90nm Technology.

Faculty Members:

Lt. Col. Tawfiq Amin
Lec. Anika Tasnim Tapti
Lec. Gourab Datta
Lec. Muhammed Zubair Rahman

Submitted By:

Name	Roll
Md. Shahriar Abid Swapnil	202216058
Mubashwira Nawar	202216066

Comparative Study of Stochastic Number Generator Using Modified NLFSR and WBG in 90nm Technology.

Md. Shahriar Abid Swapnil[1], Mubashwira Nawar[2]
Department of Electrical, Electronic and Communication Engineering
Military Institute of Science and Technology, Dhaka, Bangladesh
[1]shahriarabid54@gmail.com, [2]nawarnuha.023@gmail.com

***Abstract*—Stochastic Computing (SC) is a hardware-efficient paradigm that relies on Stochastic Number Generators (SNGs) for probabilistic data representation. This work presents the design and analysis of an efficient SNG architecture based on a Modified Non-Linear Feedback Shift Register (NLFSR) with bipartite clock-gating, integrated with Weighted Binary Generator (WBG)-based Probability Conversion Circuits (PCCs). To reduce hardware duplication, a shared-RNS configuration with multiple WBGs is also explored.Simulation in Cadence 90nm process technology demonstrates the trade-offs between power and delay. For the 4-bit design, the modified NLFSR + WBG achieves 14.32 µW and 0.168 ns, while the dual-WBG configuration increases power to 32.23 µW with negligible delay benefit. For the 8-bit case, the single-WBG design consumes 33 µW at 0.352 ns, whereas the dual-WBG shared design reduces delay to 0.286 ns at the cost of 49.11 µW.Results show that single-WBG architectures are optimal for low-power, resource-constrained applications, while dual-WBG designs are suitable for high-speed requirements. Practical implementations include lightweight cryptography, IoT security modules, neural network accelerators, and sensor interfaces, where energy efficiency and fault tolerance are critical.**

***Index Terms*—Stochastic Computing, Stochastic Number Generator (SNG), Non-Linear Feedback Shift Register (NLFSR), Bipartite Clock-Gating, Weighted Binary Generator (WBG), Probability Conversion Circuit (PCC), IoT Security, Neural Network Accelerators.**

I. Introduction

In the era of modern computing and contemporary circuits, Stochastic Computing (SC) has emerged as an eminent and promising computational paradigm due to its distinctive method of data processing and representation. The nature of Stochastic Computing is probabilistic. In contrast to deterministic traditional binary computing, which operates on fixed binary numbers, Stochastic Computing (SC) operates with random bit streams referred to as Stochastic Numbers that encode information probabilistically. Data is represented by the sequences of 0's and 1's, which are random in nature. Stochastic computing is employed in diverse domains as it facilitates cost-effective, hardware-efficient designs. Notably, its application is highly seen in the domains of Neural Networks, spectral transforms, digital filtering, and image processing, which are crucial operations. In the recent decade data security is of paramount significance, Stochastic computing enhances cryptographic systems by encoding data as probabilities,hence resulting efficient mechanisms for secure data encryption and decryption. A stochastic number (SN) encodes a real number x in the range [0,1] as a sequence of bits (0s and 1s), where the probability of a 1 appearing in the stream equals the value of the number. For instance, 0.25 can be represented by the following sequences: 0010, 1000, 00010001, 001001001000 etc. Therefore, the value is dictated depending on the ratio of number of 1's to the total bit sequence rather than the position of 1's that is independent of its placement.The main advantage of stochastic computing is that it enables the implementation of complex arithmetic functions by means of standard logic gates as well as reduction in hardware complexity and high fault tolerance during computing.

In Stochastic Computing, random bit streams that are to be computed is mainly generated with the aid of a Stochastic Number Generator (SNG). The SNG comprises of two parts: a Random Number Source (RNS) and a Probability Conversion Circuit (PCC) respectively. Random Number Source (RNS) is the fundamental block in this aspect as it is responsible for producing uniformly distributed and uncorrelated bit sequences. In conventional designs, Linear Feedback Shift Registers (LFSR) are used as RNS. An n-bit LFSR can generate $2^n - 1$ random sequences only. However there is a possibility of limited sequence length and exclusion of certain states, which reduces randomness quality.to achieve all 2^n random sequences from an n-bit RNS, Non-linear Feedback Shift Registers (NLFSRs) are becoming popular.

The Probability Conversion Circuit (PCC) serves as a crucial interface between stochastic representation and deterministic binary inputs. It mainly operates by comparing the binary input with the bit stream generated by the RNS, converting each combination into a single bit output such that the probability of 1's in the final bit stream is equal to the probability of 1's in the corresponding binary input given to the PCC.

This paper deals with the challenge of reducing the power and area consumed by SNGs in SC circuits. In response to this challenge, this work presents a comprehensive design and analysis of an efficient SNG architecture that combines two key strategies: first, the implementation of a modified Non-Linear Feedback Shift Register (NLFSR) using a bipartite clock-gating technique to serve as a low-power Random

Number Source (RNS); and second, the sharing of this single, efficient RNS across multiple Weighted Binary Generator (WBG)-based Probability Conversion Circuits (PCCs) to minimize hardware duplication.The proposed 4-bit and 8-bit SNG designs were implemented and analyzed IN Cadence 90nm process technology.

II. Methodology

A. Stochastic Number Generator and PCC

The Stochastic Number Generator (SNG) is a fundamental component of Stochastic Computing (SC), for generating random bit streams where the probability of a '1' encodes a specific value. A generic SNG structure is illustrated in Fig. 1. It is primarily composed of two key subsystems: a Random Number Source (RNS) and a Probability Conversion Circuit (PCC).

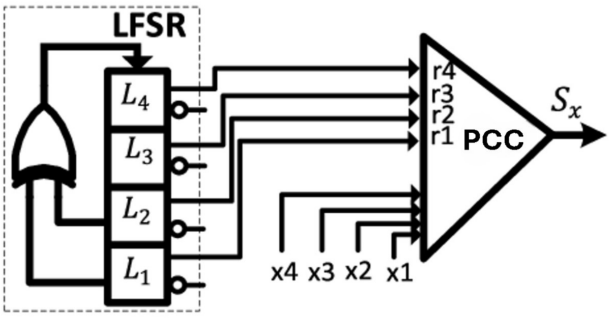

Fig. 1. Basic Structure of SNG using LFSR.

1) Random Number Source (RNS): The RNS generates the pseudo-random bit sequences that form the basis of stochastic computation. While Cellular Automata (CA) can produce high-quality random numbers, their hardware complexity, stemming from serially connected cells each containing a flip-flop and combinational logic, often makes them impractical for area-constrained SC designs. Consequently, Feedback Shift Registers (FSRs) are predominantly favored due to their lower complexity and higher operational speed.

FSRs consist of a cascade of flip-flops with a feedback path from the later stages to the input. The nature of the feedback function categorizes FSRs into two primary types:

- **Linear Feedback Shift Register (LFSR) :** The feedback path is implemented using a linear function, typically XOR operations. A primary limitation of an n-bit LFSR is that it sequences through only $2^n - 1$ states, excluding the all-zero state, before repeating. This periodicity can be a constraint in applications requiring the full range of binary combinations.
- **Non-Linear Feedback Shift Register (NLFSR):** The feedback path incorporates non-linear operations, such as AND or OR gates, in addition to XOR. Fig. 2 shows an example of an NLFSR.This crucial modification allows an n-bit NLFSR to traverse all 2^n possible states, including the all-zero state, providing a longer and more complete sequence of random numbers compared to its linear counterpart.

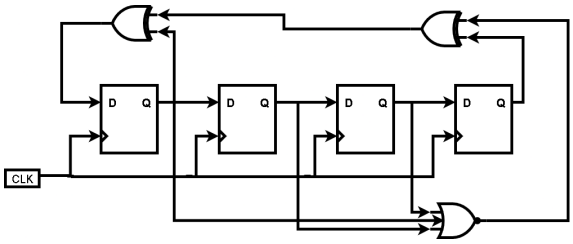

Fig. 2. Nonlinear Feedback Shift Register.

Therfore, if LFSR is used in the SNG, then $2^n - 1$ bits are produced. On the other hand, an NLFSR produces all combinations including all 0s, and thus produces 2^n combinations in succession before the combinations get repeated and when used in the SNG the output has 2^n bits.

2) Probability Conversion Circuit (PCC): The PCC translates the random bit stream from the RNS into a new stochastic stream with a user-defined probability. It takes two inputs: the random bitstream from the RNS and a deterministic binary input bit stream representing the desired probability. It has two n-bit inputs, one from the RNS and other a pre-determined user input, and one output. There are two types of PCC: Weighted Binary Generator (WBG) and Comparator (CMP) respectively.The comparator produces 1 if binary input is greater than or equal to the LFSR's output and 0 otherwise. While WBG produces output same as the bit in binary input whose position is equal to the leading 1's position in input from LFSR and 0 if input from LFSR is 0 .

3) Comparator (CMP): The CMP, illustrated in Fig. 3, functions by directly comparing the binary value of the RNS output against the target probability value.

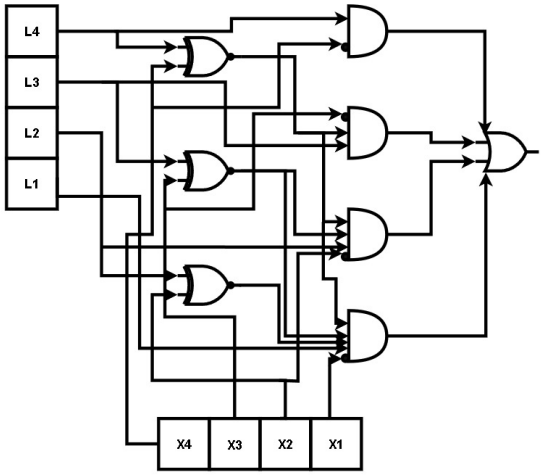

Fig. 3. Comparator Diagram.

$$\text{Output} = \begin{cases} 1 & \text{if } (Binary\ Input) \geq (RNS\ Input) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

4) Weighted Binary Generator (WBG): The WBG, shown in Fig. 4, operates by using the RNS output not as a number to be compared, but as a pointer or index.WBG produces output same as the bit in binary input whose position is equal to the leading 1's position in input from LFSR, and 0 if the LFSR input is 0. The circuit identifies the position of the leading '1' in the RNS bitstream. If no '1' is present (i.e., the RNS output is zero), the output is '0'. Otherwise, the output is the bit in the target probability value that resides at the located index position. Let B represent the *Binary Input* and L represent the *RNS Input* in Table 1.

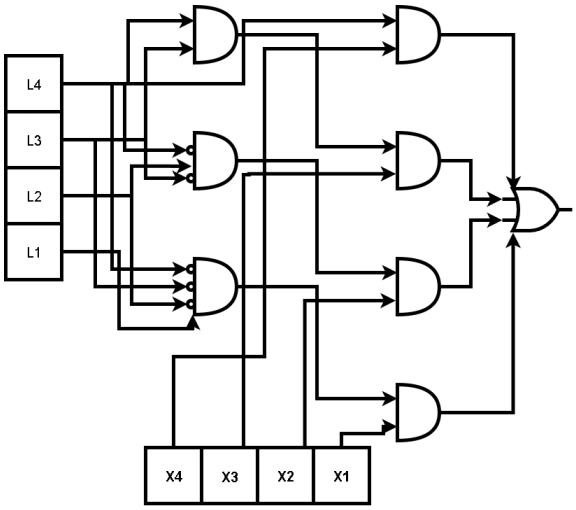

Fig. 4. WBG Diagram.

TABLE I
WBG OPERATION EXAMPLE ($B = 13$ (1101))

L (RNS)	Leading '1'	Output
0100	2	1
0010	3	0
0001	4	1
0000	–	0

$$\text{Output} = \begin{cases} B[k] & \text{if the } k^{th} \text{ bit of } L \text{ is the leading '1'} \\ 0 & \text{if } L = 0 \end{cases} \quad (2)$$

Stochastic Computing Correlation (SCC) serves as a quantitative measure of the correlation between two stochastic bit streams. Elevated correlation between streams generated by SNGs adversely impacts computational accuracy. If the correlation between two-bit streams generated by SNG is more, then the accuracy of the computation is affected. Let s_x and s_y be the random bit streams generated by binary numbers x and y respectively, then SCC is given by

$$SCC(s_x, s_y) = \begin{cases} \dfrac{\partial(s_x, s_y)}{D_1} & \text{if } \partial(s_x, s_y) \geq 0 \\ \dfrac{\partial(s_x, s_y)}{D_2} & \text{otherwise} \end{cases}$$

where
$D_1 = \min\left(P(s_x), P(s_y)\right) - P(s_x)P(s_y)$
$D_2 = P(s_x)P(s_y) - \max\left(P(s_x) + P(s_y) - 1, 0\right)$

where $P(s_x)$ and $P(s_y)$ are probabilities of 1's in bit stream s_x and s_y and $\partial(s_x, s_y) = P(s_x \wedge s_y) - P(s_x)P(s_y)$ where $\wedge$ denotes bitwise AND operation of s_x and s_y. SCC varies between -1 and +1. A value of 0 is desired as SCC which means that there is no correlation. If all corresponding bits of two-bit streams are equal, then SCC is 1 and it becomes -1 if they are complemented. The average SCC between two SNGs is calculated by

$$SCC_{\text{avg}}(SNG_1, SNG_2) = \frac{1}{2^{2n}} \sum_{i=0}^{2^n-1} \sum_{j=0}^{2^n-1} |SCC(s_i, s'_j)|$$

where s_i, s'_j represent the bit streams generated by SNG1 and SNG2 respectively for all possible input combinations.

III. PROPOSED MODEL

This section details the proposed Random Number Source (RNS) and Stochastic Number Generator (SNG) architectures. The modified RNS design is described for both 4-bit and 8-bit configurations.

A. Proposed RNS Design

While Linear Feedback Shift Registers (LFSRs) generate only $2^n - 1$ states, Non-Linear Feedback Shift Registers (NLFSRs) produce the complete set of 2^n states, thereby improving sequence quality and stochastic computation accuracy.

A modified NLFSR architecture employing a bipartite clock-gating technique is employed. The n-bit register is partitioned into two segments of size $n/2$. The first segment operates on every clock cycle, while the second segment is activated only when specific conditions from the first segment are met.The second part does not work for each clock cycle which reduces the number of switching activities in the generated random sequences.

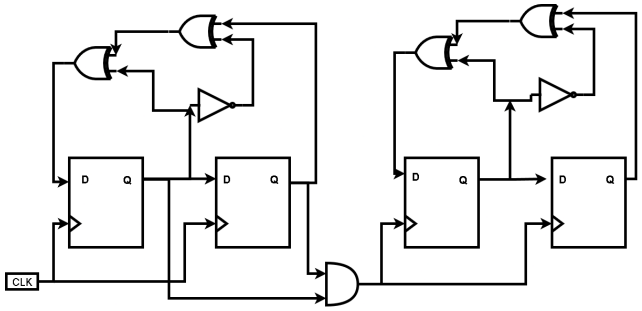

Fig. 5. Schematic of modified 4-bit NLFSR using positive-edge triggered D flip-flops.

The proposed modified NLFSR is implemented using positive-edge triggered D flip-flops, as shown in Fig. 5. Each part internally has nonlinear feedback and thus maintaining non-linearity internally. The first part receives normal external clock while the second part's clock is the output of AND

operation of outputs of the first two flip flops. In other words, last two flip-flops work only when the first two flip flops output is 1. By doing so the number of switching activities, in the generation of all the random sequences, are reduced.The effectiveness of the proposed bipartite clock-gating technique is demonstrated through switching activity analysis of the modified 4-bit NLFSR. The conditional activation of the second flip-flop pair significantly reduces unnecessary transitions, thereby lowering dynamic power consumption. Table II presents the switching characteristics The generated sequence of the 4-bit Non-Linear Feedback Shift Register (NLFSR) is shown in Table II. In addition to the raw NLFSR output states, the corresponding decimal values are computed based on the non-standard *8-7-2-1 weighted code*, where each bit is assigned a distinct positional weight.

TABLE II
SWITCHING ACTIVITY OF MODIFIED 4-BIT NLFSR SEQUENCE

Clock Cycle	4-bit NLFSR State	Weighted Decimal Value
1	0000	0
2	1000	8
3	1110	17
4	0110	9
5	0010	2
6	1010	10
7	1111	18
8	0111	10
9	0011	3
10	1011	11
11	1101	16
12	0101	8
13	0001	1
14	1001	9
15	1100	15
16	0100	7

The weighted decimal value is calculated as $Value = 8b_3 + 7b_2 + 2b_1 + 1b_0$, where b_3 is the MSB and b_0 is the LSB of the 4-bit NLFSR output.

B. Proposed SNG Design

The proposed Stochastic Number Generator (SNG) integrates the modified NLFSR as the Random Number Source (RNS) with a Weighted Binary Generator (WBG) as the Probability Conversion Circuit (PCC). This architectural choice is motivated by the superior efficiency of WBG compared to Comparator-based PCC (CMP) in terms of Stochastic Computing Correlation (SCC), area, and power consumption.

Our analysis confirms these advantages, demonstrating that a 4-bit WBG exhibits significantly lower area and power compared to an equivalent 4-bit CMP implementation. This makes the WBG the optimal choice for low-power stochastic computing applications.

The SNG operates by transforming the output of the modified NLFSR into weighted binary values, which are then used in conjunction with the deterministic binary input to generate the final stochastic bit stream. For applications requiring multiple parallel stochastic streams, a resource-sharing architecture employed where a single modified NLFSR is shared among multiple WBG units. This approach maximizes hardware utilization while minimizing area and power duplication.

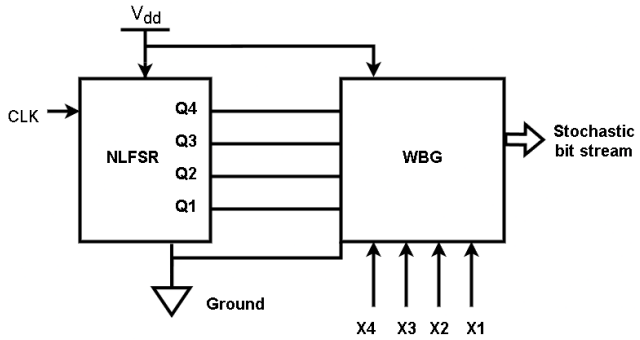

Fig. 6. Architecture of 4-bit modified NLFSR with single WBG

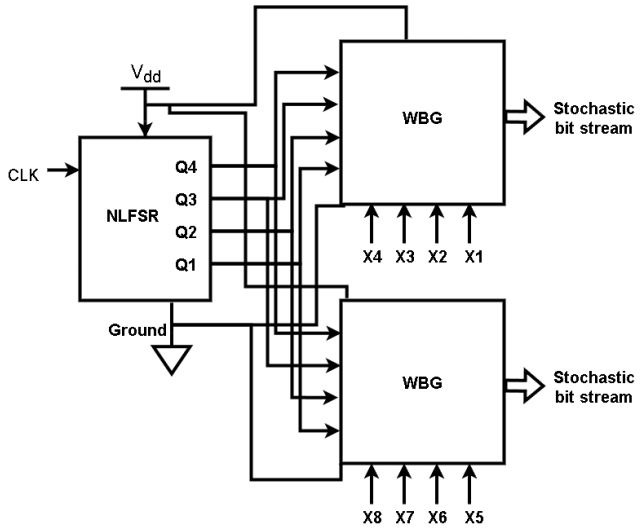

Fig. 7. Architecture of 4-bit modified NLFSR with two shared WBGs

Performance evaluation of the complete SNG demonstrates significant efficiency gains. The integrated design achieves optimal power and area efficiency without compromising performance, while the shared architecture particularly excels in multi-stream applications, providing an optimal balance between hardware resources and computational throughput.

IV. RESULTS AND PERFORMANCE ANALYSIS

For performance evaluation and comparison, the designs were synthesized using the Cadence Virtuoso with 90 nm technology and analyzed for power, area, and delay metrics.

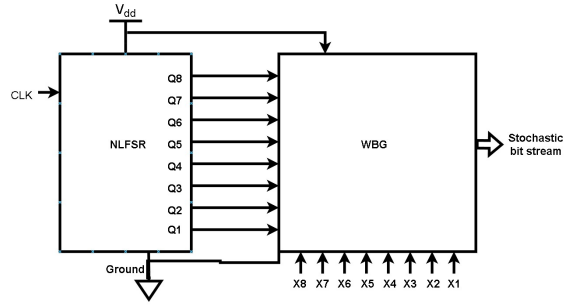

Fig. 8. Architecture of 8-bit modified NLFSR with single WBG

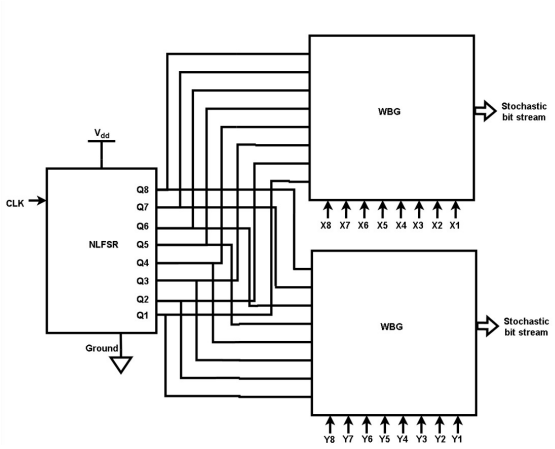

Fig. 9. Architecture of 8-bit modified NLFSR with two shared WBGs

A. Analysis of NLFSR Output Switching Activity

The proposed modified NLFSR demonstrates significantly reduced switching activity compared to conventional designs. Figure 10 shows the switching activity graph implementation.

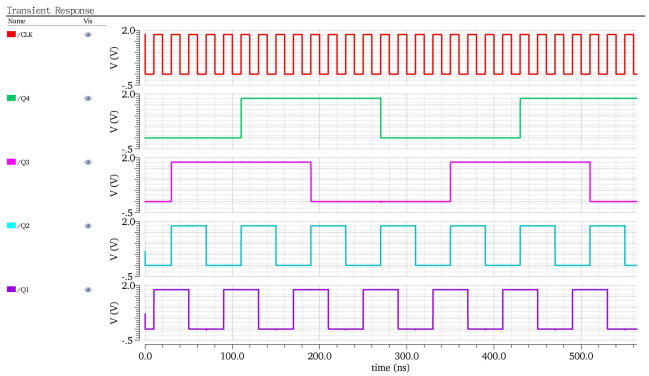

Fig. 10. Switching activity comparison between standard NLFSR and proposed modified NLFSR

At every positive edge of the clock, the outputs of the Modified 4-bit NLFSR ($Q_4Q_3Q_2Q_1$) change according to the nonlinear feedback function.

B. Comparison of 4-bit WBG and Comparator

A comprehensive comparison between 4-bit Weighted Binary Generator (WBG) and Comparator (CMP) implementations was conducted. The results demonstrate the significant efficiency advantage of the WBG architecture over the conventional comparator approach.

TABLE III
PERFORMANCE COMPARISON OF 4-BIT PCC IMPLEMENTATIONS

Parameter	4-bit WBG	4-bit Comparator
Area (μm^2)	406.86	1727.57
Power (μW)	4.91	6.34

Table III shows that the WBG uses 76.4% less area, 22.5% less power than the Comparator. The WBG achieves this through its simpler architecture, which requires fewer logic gates. In contrast, the Comparator has a comparatively complex arithmetic circuit. This results in higher gate count, larger area, and increased power consumption . The WBG's design makes it clearly superior for low-power stochastic computing applications. Both designs underwent Design Rule Check (DRC) and Layout Versus Schematic (LVS) checks. The DRC verification confirmed that both layouts comply with the 90-nm technology design rules. The LVS verification successfully matched the extracted netlist from the layout with the original schematic, confirming correct connectivity and device parameters. The WBG layout demonstrates significantly better area utilization and more compact routing compared to the Comparator, directly contributing to its better performance metrics.

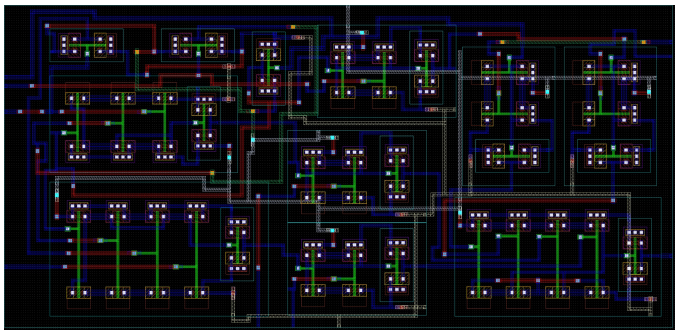

Fig. 11. Layout of 4-bit WBG implementation

C. Performance Analysis of Integrated SNG Designs

The complete SNG implementations, integrating the proposed modified NLFSR with WBG-based PCCs, were evaluated for both 4-bit and 8-bit configurations. Table IV and V presents the power and delay measurements for standalone and shared configurations.

The 4-bit shared configuration consumes 55.56% more power than the standalone version while maintaining nearly identical delay (0.170 ns vs. 0.168 ns).

The 8-bit single SNG consumes 33.00 μW with 0.352 ns delay. Remarkably, the shared configuration achieves 49.11 μW power consumption with reduced delay of 0.286 ns.

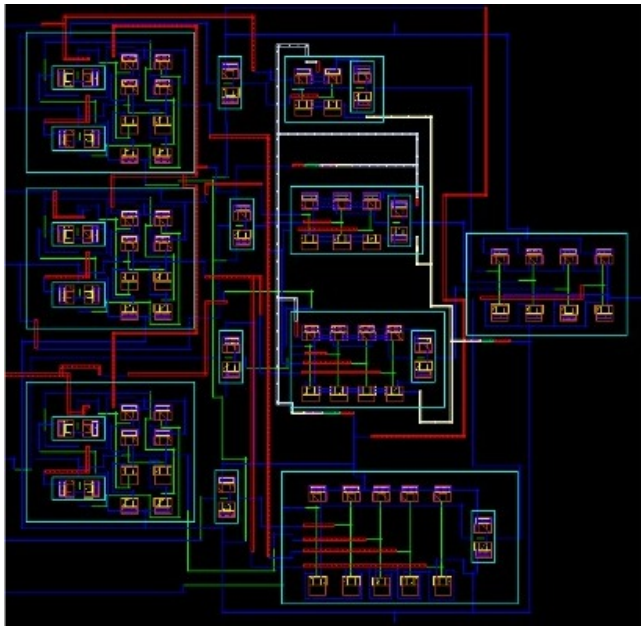

Fig. 12. Layout of 4-bit Comparator implementation

TABLE IV
PERFORMANCE OF 4-BIT SNG IMPLEMENTATIONS

Configuration	Power (μW)	Delay (ns)
4-bit SNG (1xNLFSR + 1xWBG)	14.32	0.168
4-bit Shared SNG (1xNLFSR + 2xWBG)	32.23	0.170

TABLE V
PERFORMANCE OF 8-BIT SNG IMPLEMENTATIONS

Configuration	Power (μW)	Delay (ns)
8-bit SNG (1xNLFSR + 1xWBG)	33.00	0.352
8-bit Shared SNG (1xNLFSR + 2xWBG)	49.11	0.286

While the shared SNG architecture increases total power consumption compared to a single SNG, this tradeoff is justified by its significant advantages in multi-stream applications. The higher total power—32.23 μW for the shared 4-bit design versus 14.32 μW for a single SNG—reflects the energy cost of generating two independent stochastic streams simultaneously.

In addition the 8-bit version, the shared system not only supports two streams with a moderate power increase (33.00 μW to 49.11 μW) but also delay from 0.352 ns to 0.286 ns

The principal advantage of this architecture lies in its ability to generate multiple high-quality stochastic bit streams using a single, shared modified NLFSR. This approach eliminates the substantial area and power overhead that would be incurred by duplicating the entire Random Number Source (RNS) for each stream. The design is therefore well-suited for several key application domains:

- **Parallel Stochastic Computing Systems:** Applications requiring numerous simultaneous stochastic operations, such as complex polynomial evaluation or multi-input decoding, benefit greatly. The shared RNS efficiently supplies randomness to all parallel computational units.
- **Low-Area Embedded Designs:** In resource-constrained environments like IoT devices or edge AI accelerators, hardware reuse is paramount. This architecture maximizes functional throughput per unit area, making it ideal for these settings.

The tradeoff of a higher total power budget is consequently justified by significant gains in functional density,efficiency, and, notably in the 8-bit configuration, enhanced performance. This renders the proposed shared architecture a superior and highly suitable choice for modern parallel processing applications, where multiple stochastic operations must be performed concurrently without proportional increases in latency or silicon area.

V. CONCLUSION

This paper presented a comparative study of SNG architectures using Modified NLFSRs, WBGs, and Comparators in 4-bit and 8-bit configurations. Results show that WBG-based PCCs provide better randomness than comparators while consuming comparatively lesser power. The single-WBG architecture proved most power-efficient (14.32 μW at 4-bit, 33 μW at 8-bit), while the dual-WBG shared designs achieved lower delay (0.286 ns at 8-bit) at the cost of higher power. These findings highlight a clear power–delay trade-off: single-WBG designs are suitable for low-power IoT and security applications, whereas dual-WBG configurations favor high-speed domains.Future extensions of this work may include scaling to higher bit-widths, FPGA/ASIC prototyping, statistical randomness validation, and robustness testing under process-voltage-temperature (PVT) variations. Additionally, exploring hybrid architectures combining multiple SNGs, optimization for ultra-low-power IoT devices, and integration with stochastic computing accelerators for neural networks or cryptographic applications can further enhance the practicality and performance of the proposed designs, enabling broader deployment in real-world SC systems.

REFERENCES

[1] B. Brown and H. Card, "Stochastic neural computation. I. Computational elements," *IEEE Transactions on Computers*, vol. 50, no. 9, pp. 891–905, Sep. 2001.

[2] H. Ichihara, T. Sugino, S. Ishii, T. Iwagaki, and T. Inoue, "Compact and accurate digital filters based on stochastic computing," *IEEE Transactions on Emerging Topics in Circuits and Systems*, vol. 7, no. 1, pp. 31–43, Jan. 2019.

[3] S. R. Prasad, A. Siripagada, S. Selvaraj, and N. Mohankumar, "Random seeding LFSR-based TRNG for hardware security applications," *Studies in Computational Intelligence*, vol. 771, pp. 427–434, 2019.

[4] P. P. Mahapatra and S. Agrawal, "RSA cryptosystem with modified Montgomery modular multiplier," in *Proc. IEEE Int. Conf. Computational Intelligence and Computing Research (ICCIC)*, pp. 1–6, 2017.

[5] R. S. Durga, C. K. Rashmika, O. N. V. Madhumitha, D. G. Suvetha, B. Tanmai, and N. Mohankumar, "Design and synthesis of LFSR based random number generator," in *Proc. 3rd Int. Conf. Smart Systems and Inventive Technology (ICSSIT)*, pp. 438–442, 2020.

[6] S. Mohajer, Z. Wang, K. Bazargan, M. Riedel, D. Lilja, and S. Faraji, "Parallel computing using stochastic circuits and deterministic shuffling networks," U.S. Patent 11,018,689, Apr. 2019.

[7] N. Haridas and M. N. Devi, "Efficient linear feedback shift register design for pseudo exhaustive test generation in BIST," in *Proc. 3rd Int. Conf. Electronics Computer Technology (ICECT)*, vol. 1, pp. 350–354, 2011.

[8] V. N. Teja and E. Prabhu, "Test pattern generation using NLFSR for detecting single stuck-at faults," in *Proc. IEEE Int. Conf. Communication and Signal Processing (ICCSP)*, pp. 716–720, 2019.

[9] N. Krishna, V. Murugappan, R. Harish, M. Midhun, and E. Prabhu, "Design of a novel reversible NLFSR," in *Proc. IEEE Int. Conf. Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2279–2283, 2017.

[10] P. K. Gupta and R. Kumaresan, "Binary multiplication with PN sequences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 4, pp. 603–606, Apr. 1988.