International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Design of Low-Cost Stochastic Number Generator Using TSPC Logic in 45nm Technology

Pavan PH[1], Lalitha S[2]

[1]Electronics and Communication Engineering, BMS College of Engineering, Bull Temple Road, Basavanagudi, Bengaluru, India
Email: *pavanph123[at]email.com*

[2]Electronics and Communication Engineering, BMS College of Engineering, Bull Temple Road, Basavanagudi, Bengaluru, India
Email: *lalithas.ece[at]bmsce.ac.in*

Abstract: ***An essential part of stochastic computing (SC) is the stochastic number generator (SNG). The SNG has an ability to change the binary numbers into stochastic bit streams. An SNG includes random number source (RNS) i.e., LFSR and a comparator. LFSR generates random bit sequences based on the tapings made in it. Using simulation outcomes, it describes how the weighted binary generator (WBG) which can be used to replace the comparator (CMP) component of SNG circuits to reduce SC correlation. The probability conversion circuit (PCC) is another name for WBG. The PCC converts the generated random numbers into a random bit stream, which has the desired chance of generating 1s. This project presents a comparative approach towards designing SNG in different logics in 45nm technology. Different logics are CMOS and TSPC. SNG is designed for every logic mentioned above. By comparing parameters like Area, Power consumption, Number of transistors used, working frequency, and concluding which logic based SNG is better.***

Keywords: Linear feedback shift register, Stochastic number generator, weighted binary generator, Random number source

1. Introduction

An essential part of modern computing, the Stochastic Number Generator (SNG) manages the unpredictability in a wide range of applications. SNGs have proven to be remarkably sophisticated and adaptable in the face of contemporary issues. The problem of predictability and vulnerability has been addressed using advanced cryptography algorithms and methods. The generated sequences are protected from decryption efforts and pattern recognition by cryptographically secure SNGs, which contain challenging mathematical operations and strict entropy sources to fend off even the most cunning attacks.

The development of unique algorithms that optimize the trade-off between speed and statistical validity has addressed the problem of balancing randomness with computational efficiency. A new era of fully indeterministic sequences is also promised by the introduction of quantum-based random number generators, which will take use of the inherent unpredictable nature of quantum processes.

Fundamentally, modern SNGs have met the problem by using state-of-the-art cryptographic concepts, algorithmic breakthroughs, and quantum-inspired techniques. This development gives them the ability to overcome the complex difficulties given by modern computing environments, safeguarding data transmission, improving simulations, and strengthening their position as suppliers of real randomness in an increasingly interconnected digital world.

A contemporary method for using logic circuits to perform computations is stochastic computing (SC). Then executing computation on deterministic binary numbers, Stochastic computing circuits are built to process random bit streams. Bit streams are used to represent the input and output, and the values are encoded as the chances that the bit streams will include 1s. Compared to deterministic binary computing, Numerous benefits of stochastic computing include reduced hardware complexity, which may lead to fault-tolerant computing and cost-effective computing circuits.

A. Stochastic Number Generator (SNG)

A Stochastic Number Generator (SNG) is a fundamental computing tool that generates numbers in sequences with statistical characteristics similar to randomness. SNGs are essential for replicating randomness in different applications since computers create numbers deterministically, in contrast to actual random processes. These techniques are used by these generators, which are also used in simulations and cryptography, to create sequences that appear random. However, because SNGs are deterministic and repeating when the initial conditions are known, the phrase "pseudo-random" is frequently employed. Effective SNG design requires balancing variables like speed and statistical quality as well as resolving security issues.

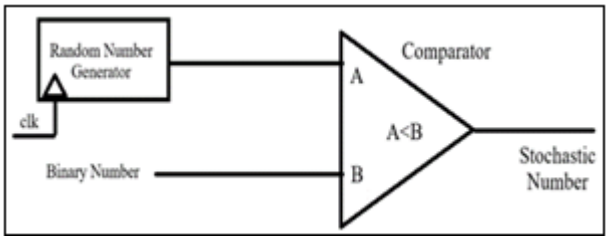

Figure 1: Stochastic number Generator

B. Linear Feedback Shift Register

A Shift register's input bits called in LFSR has a previous state that is linearly connected to it. The register's functioning is fated, and the streaming values generated is entirely defined by its current (or previous) state. A repeating cycle will occur due to the register's finite number of states. After 2n-1 cycles, the sequence repeats, and for a

given LFSR length, there may be more than one maximal tap sequence. Typically, this works with a clock pulse and an EXOR signal applied to each shift register's output. Typically, shift registers will be used as D flip flops.

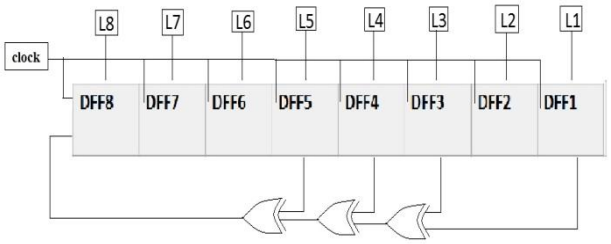

Figure 2: Linear Feedback Shift Register

The paper is organized as follows: Section II, depicts the literature reviews of Stochastic Number Generator. Section III describes the proposed Stochastic number generator using CMOS and TSPC logic. Section IV, the simulation results of proposed SNG. Section V concludes this paper.

2. Literature Survey

In this paper [1], the author has proposed a novel stochastic number generator architecture and proved that the resulting circuit can deliver independent stochastic numbers and improve the accuracy of the calculation results obtained using some recent conventional stochastic computing-based arithmetic circuits. This study is motivated by the increasingly important role of stochastic computing in various fields, such as the digital circuit design, where the stochastic number generators are responsible for a significant share of the hardware cost. Furthermore, the circuit constructed according to the proposed architecture delivers independent stochastic numbers and outperforms the circuits designed using conventional methods. In future studies, the author intends to evaluate the hardware cost of the proposed architecture [1].

A viable paradigm for achieving soft error-tolerant, low-complex digital circuits is stochastic computing (SC). SNG is a component of stochastic circuits that generates a stochastic number that corresponds to an input binary number. Traditional SNGs would take a long time to run since they use linear feedback shift registers (LFSRs) to create random numbers serially. This article suggests an asynchronous SNG that may generate stochastic numbers simultaneously by changing the given binary number into a modified unary number and permuting it using a bit permutation network. A technique for sharing a single LFSR among several SNGs has also been presented. Here, a parallel SNG that can produce all bits of a stochastic number simultaneously has been proposed. Parallel stochastic number generation has been accomplished using an omega-flip bit permutation (BP) network. A stochastic number's SCC is also enhanced by using BP networks because more permutations are possible for a given bit stream. Comparing the proposed SNG to the shared LFSR-based SNG, experimental research revealed improvements in average SCC and energy-delay-product of 28.57% and 4.32x, respectively. When complex applications with many SNGs are considered, a way of sharing a single LFSR among multiple SNGs has also been given, which offers benefits like reduced area. When compared to previous methods, using the suggested SNG in applications like multiplication, edge detection, and complex multiplication results in execution times that are up to 1000 times faster and area-delay products that are up to 9 times faster. Future study will look at ways to generate stochastic numbers using time domain methodologies and alternative logic philosophies in order to reduce the energy usage of the proposed SNG. [2].

A novel computing paradigm that uses stochastic bit streams is stochastic computing (SC). Due to the computing core's extremely low area and power requirements, it has recently attracted interest. To translate input binary integers to stochastic bit streams, SC uses stochastic number generators (SNGs). A comparator and an LFSR, which is the normal RNS, are components of a standard SNG. The main benefits of the SC core are negated because it requires significantly greater area and power. SNGs using novel nanoscale components like memristors and spintronic devices have been proposed as a solution to this issue. However, due to unforeseen differences in their fabrication methods and noise in their control signals, these devices frequently have significant inaccuracies in their output probabilities. The author offers a fresh approach for utilizing such tools to create a very precise SNG. It is based on an RNS that, under ideal (nominal) circumstances, creates uniformly distributed random numbers. Additionally, when the RNS is prone to mistakes, it has a revolutionary error-canceling probability conversion circuit (ECPCC) that ensures extremely high accuracy in the output probability under realistic circumstances. Maximally correlated stochastic streams can also be produced by an ECPCC, which is a valuable feature for some applications [4].

Cryptographic key generation is an important part of the secured communication system where the key that is generated has a major role to play in the strength of the security of the data that is transferred. To enhance the necessary strength of the key, the random number generated has to be highly secure. This is enhanced using a True Random Number Generation. Diffused Bit Generator (DBG) is an entropy source which is used to produce a sequence of random bits. It is composed of a LFSR and a Cellular Automata for increasing the randomness emanating from the DBG. The proposed LFSR has been designed using TSPC based D flip-flops and the XOR gates consisting of 6 transistors, which has enabled it to fulfill the objective of low power. The circuit implementation has been done in Cadence Virtuoso in the CMOS 180 nm technology and simulated in Cadence Spectre. [3].

Low-area circuits are provided by stochastic unary computing. The required stochastic number generators (SNGs) in these circuits, however, can reduce their overall gain in area, especially if multiple SNGs are needed. By distributing the permuted output of a single linear feedback shift register (LFSR) among multiple SNGs, the author suggests creating area-efficient SNGs. The suggested design generates stochastic bit streams with the least amount of stochastic computing correlation (SCC) with no additional hardware overhead. When a 10-bit LFSR is shared between two SNGs, the authors' approach results in stochastic bit streams with 67% less average SCC than the circular

shifting method given in earlier work. The author suggests an algorithm to create a set of m permutations (n > m > 2) with a minimum pairwise SCC for an n-bit LFSR in order to generalize our method. When n rises, the search space for permutations with an exact minimum SCC expands quickly, and for n > 9, it becomes infeasible to run a search algorithm utilizing pairwise SCC values that have been precisely determined. To quickly locate a group of permutations with SCC values close to the minimal one, the author suggests a similarity function that may be incorporated into the suggested search process. For a number of applications, the author examined the strategy. The findings demonstrate that it delivers lower mean-squared error (MSE) with the same (or even smaller) area as past work [6].

3. Proposed Method

As it has been seen that how the Stochastic number generator works and what exactly it consists of. Stochastic number generator occupies more area in the stochastic computing circuits. By knowing all these, attempt has been to improvise area consumption, a smaller number of transistor usage and the working frequency of the SNG. There are some of the logics in the VLSI design which are known for their unique characteristics. We decided to design SNG in 4 different logics such as CMOS and TSPC logic. Let us see the entire implementations of all the circuit.

Any SC circuit must have a stochastic number generator (SNG). To translate binary numbers into the corresponding random bit streams, a SC circuit employs SNGs. With probability of creating 1s equal to their corresponding binary integers, they produce random bit streams. SNGs are essential to a SC circuit's effectiveness.

There are many logics in VLSI designing Complementary Metal Oxide Semiconductor (CMOS) and True Single-Phase Clock (TSPC) to name a few. Here the question arises that which logic based SNG is better in all the parameters.

In response to the above question, the works in this paper are as follows: 1) The SNG is designed in different logic such as CMOS and TSPC. 2)Various parameters like Area, Power consumption, Number of transistors used, working frequency are compared and found that which logic SNG is better.The SNG's implemented till date were designed using CMOS Logic and there is no information on implementations on TSPC and many other different logics in VLSI. Stochastic number generators continue to be among the costliest components of stochastic circuits. The problems of designing and deploying cost effective SNGs that ensure a desired, system-wide level of accuracy or precision is far from solved.

Till date there was an approach on improvising SNG by changing the circuit of PCC. In this work we have improved SNG in different parameters by designing it in different logics in 45nm technology. Not only PCC, each and every component has been designed in different logic to observe the mentioned constraints.

By comparing various parameters of SNG in different logics following were observed: Reduction in area, Reduction in no. of transistors required, Improvement in the working frequency and reduction in cost of the SNG in few logics. Thus, concluding which SNG is better according to the specifications.

The main objective of the proposed method is as follows:

- To design a low-cost stochastic number generator by sharing the single LFSR with two WBG circuits.
- To design all the components of the SNG in different logics like CMOS and TSPC and observe for perfect output.
- To observe the schematics, outputs and layouts of different SNGs.
- To build the compact layouts of all the components used in the project.
- To compare the different logic SNGs in parameters like Area consumed, Power consumed, No. of transistors and working frequency, thus concluding which SNG is better according to the given constraints and requirements.

A. CMOS logic

In VLSI there are different technologies. Most common used logic is Complementary Metal Oxide Semiconductor (CMOS). This is basically a class of integrated circuits, and is used in a range of applications with digital logic circuits, such as microprocessors, microcontrollers, static RAM, etc. It is also used in applications with analogue circuits, such as in data converters, image sensors, etc. There are quite a few advantages that the CMOS technology has to offer. One of the main advantages that CMOS technology, which makes it the most commonly-used technology for digital circuits today, is the fact that it enables chips that are small in size to have features like high operating speeds and efficient usage of energy.

Besides, they have very low static power supply drain most of the time. Besides, devices using CMOS technology also have a high degree of noise immunity. CMOS circuits use a combination of p-type and n-type metal–oxide–semiconductor field effect transistor (MOSFETs) to implement logic gates and other digital circuits.

B. TSPC Logic

Originally proposed as a high-speed topology, the TSPC structure also consumes less power and occupies less area than other methods. The high-speed dynamic True Single-Phase Clock (TSPC) logic design style offer fully pipelined logic circuits using only one clock signal, which makes clock distribution simple and compact. The TSPC-cell consists of one N-block and one P-block each driven by single clock signal.

In standard TSPC logic style the implementation of simple gates like AND, OR, XOR increases the transistor count since each logic cell implementation use both N-block together with P-block to remove transparency between the pipelined stages, while logic function is implemented only with N-block; therefore, it makes the P-block redundant, since it performs no logic function other than latching. Although logic merging is possible, in standard TSPC cell by implementing logic functions with both the N-block and P-block, but such a logic merging is inefficient, since in practical cases inputs to a gate may appear after different

cycle delays, which may not allow effective logic merging.

As we saw in the existing methodology, the block diagram of LFSR remains same for here too. But the main difference is that the circuit inside it is different. There are changes in basic components inner circuits.

Proposed Weighted Binary generator(WBG):

Weighted binary generator (WBG) is a PCC introduced by Gupta and Kumaresan. A WBG with the probability precision k = 8 is shown in Fig. The AND gates in the first level take unbiased random bits L8…...L1 as inputs and produce the intermediate signals W8….W1.

W8=L8,W7=L8'&L7, W6=L8'&L7'&L6,
W5=L8'&L7'&L6'&L5, W4= L8'&L7'&L6'&L5'&L4,
W3= L8'&L7'&L6'&L5'&L4'&L3,
W2= L8'&L7'&L6'&L5'&L4'&L3'&L2,
W1=L8'&L7'&L6'&L5'&L4'&L3'&L2'&L1……(1)

Where & represents the logical AND. The AND gates in the second level take W8,….,W0 and the target bits x8,...,x0 as inputs and produce the intermediate signals. Finally, a tree of 7 two-input OR gates have been used. R2…R8 are the blocks of AND gates which we have made for references. Given from equation (1), the probability of Wi's are where i ranges from 8 to 1:-

P(W8)=1/2, P(W7)=1/(2^2), P(W6)=1/(2^3),P(W5)=1/(2^4), P(W4)=1/(2^5), P(W3)=1/(2^6), P(W2)=1/(2^7), P(W1)=1/(2^8)

Probability of out is given by: - P(out)= P(W8)x8+ P(W7)x7+ P(W6)x6+ P(W5)x5+ P(W4)x4+ P(W3)x3+ P(W2)x2+ P(W1)x1

P(out)=(1/2)x8+(1/2^2)x7+(1/2^3)x6+(1/2^4)x5+(1/2^5)x4 +(1/2^6)x3+(1/2^7)x2 +(1/2^8) x1

P(out)= X/(2^8), where X= (x8……x1). Thus, the WBG functions as PCC.

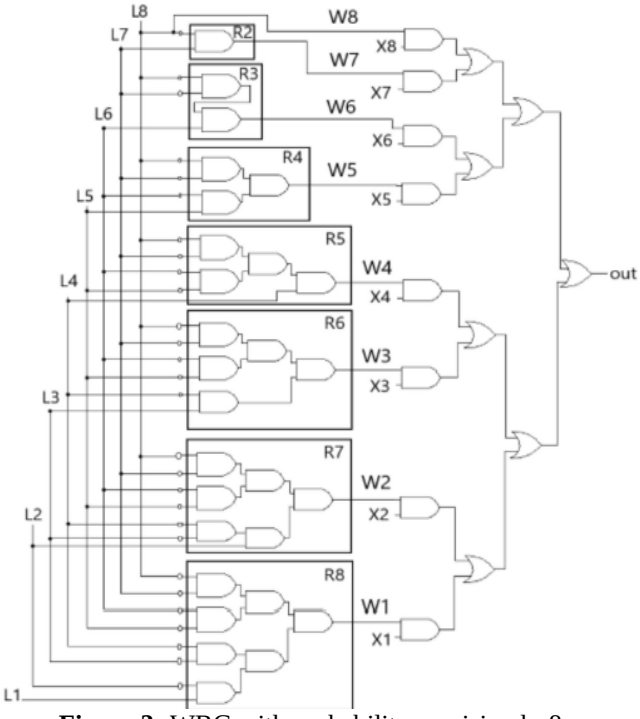

Figure 3: WBG with probability precision k=8

Proposed stochastic number generator (SNG)

Stochastic number generator (SNG) is an important component of a stochastic computing. It is the component which converts binary number into a stochastic bit stream. There are two main components in the stochastic number generator. SNG consists of a random number source (RNS) and a probability conversion circuit (PCC). RNS generates a random binary number distributed uniformly in the range [0,2k-1] i.e., for every [0,2k-1] bits the sequence repeats, where k is the bit width of the random binary number. The output of RNS can be taken as k unbiased bits having probability 0.5 for getting 1 and 0.5 for getting 0. We can consider linear feedback shift register as a random number source. In the above Fig we can observe that the random number source which provides random bit sequences has been shared between two PCC circuit just to make it of low cost as they saw in the paper.

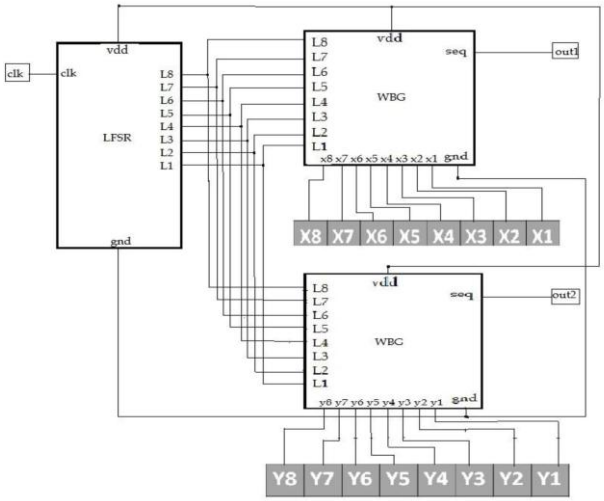

Figure 4: SNG containing shared LFSR

The output of LFSR (RNS) is fed to the PCC circuit i.e., weighted binary generator. WBG is fed with another sort of inputs, those are called as target bits. x8..x1 are the target bits given to the 1st WBG circuit, y8….y1 are the target bits given to the 2nd WBG circuit. The 2 outputs are taken from 2 WBG circuit. They have to initialize the outputs of the linear feedback shift register; this process is called as a SEED.

When the clock pulse is applied to the LFSR, as we saw in the previous part, LFSR starts to generate random bit sequences. When these bits are fed to the WBG circuit, with the help of these random bits WBG encodes the target bits.

Then, can observe the output of SNG which has number of bits which are high is equal to decimal equivalent of target bits out of 2k clock pulses. For example, if the target bits are 00000100. The decimal equivalent of the target bit is 4. In the output of an SNG, there will 4 bits which are high out of 256 bits.

4. Simulation Results and Discussion

The output of the SNG mainly depends upon the target bits given to the WBG circuit. The total number of ones in the output of SNG is the decimal equivalent of the target bits

given to the WBG circuit. In this project we have shared a single SNG with two WBGs and hence we will get two outputs. We have designed an 8-bit SNG the total output bits are of 256 bits (2^8). The decimal equivalent of the target bits is the number of ones of the 256 bits.

The figures 5, 6 and 7 below shows the CMOS based 8-bit LFSR which consists of D Flip flops, WBG and SNG respectively. The design is done in Cadence Virtuoso Software using 45nm Technology.

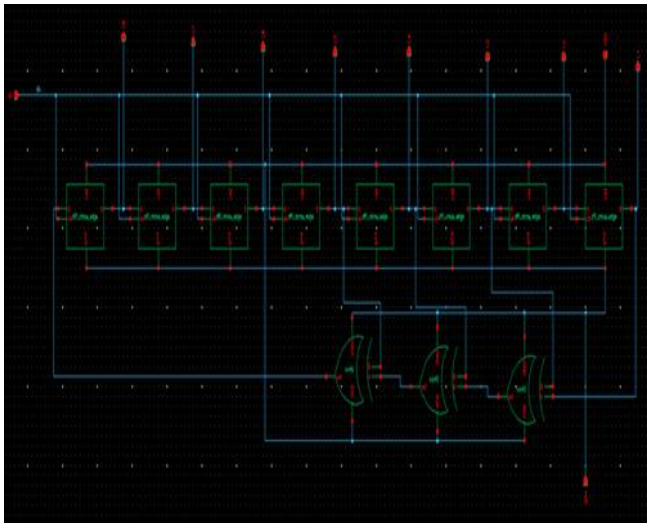

Figure 5: 8-bit LSFR using CMOS logic

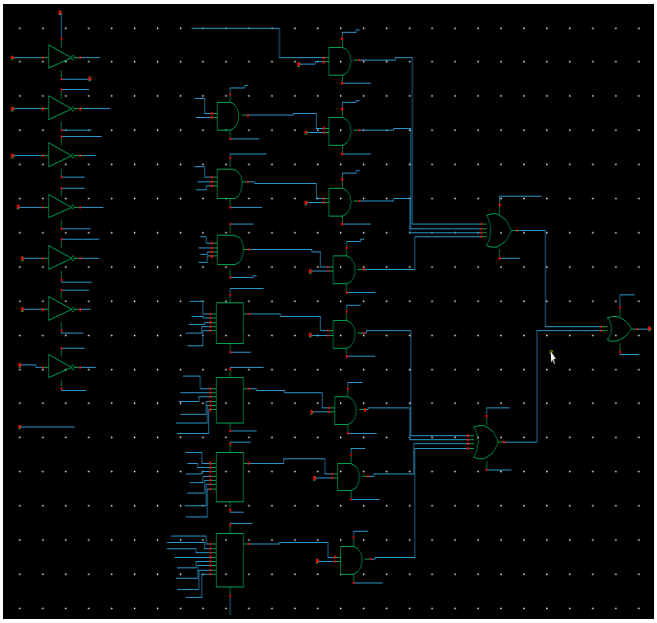

Figure 6: WBG using CMOS logic

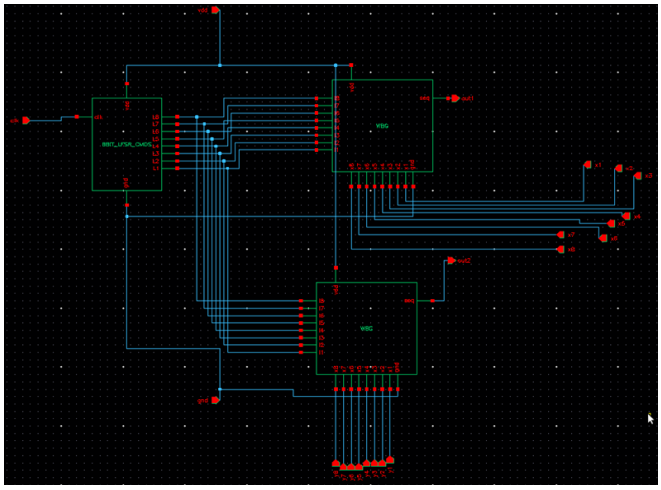

Figure 7: 8-bit SNG using CMOS logic

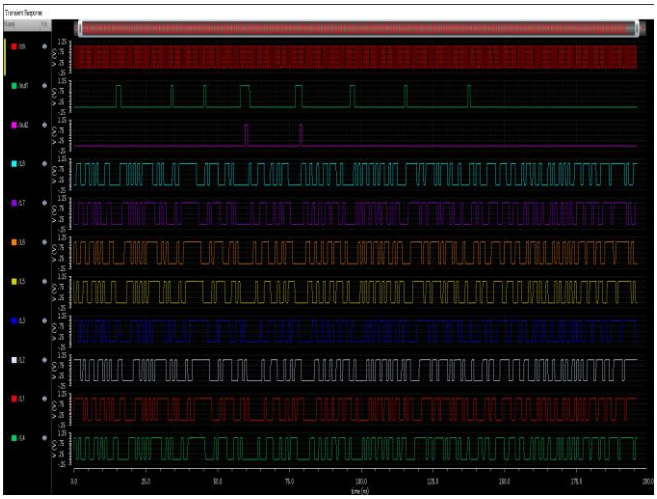

Figure 8: Output of SNG using CMOS logic

The figures 9, 10 and 11 below are the TSPC based 8-bit LFSR, WBG and SNG respectively which consists of D Flip flops, AND gates, OR gates and XOR gates. The design is done in Cadence Virtuoso Software using 45nm Technology.

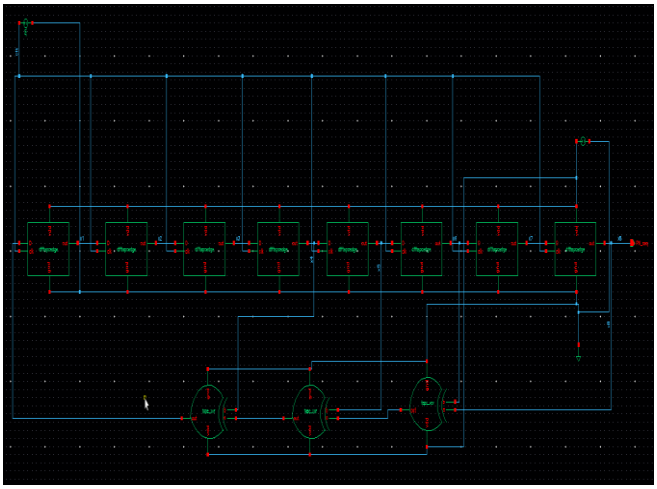

Figure 9: 8-bit LFSR using TSPC logic

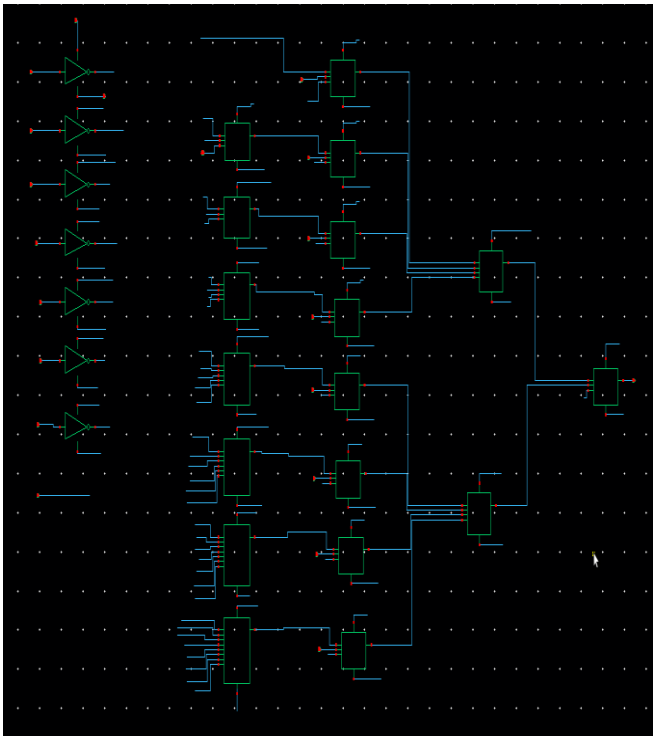

Figure 10: WBG using TSPC logic

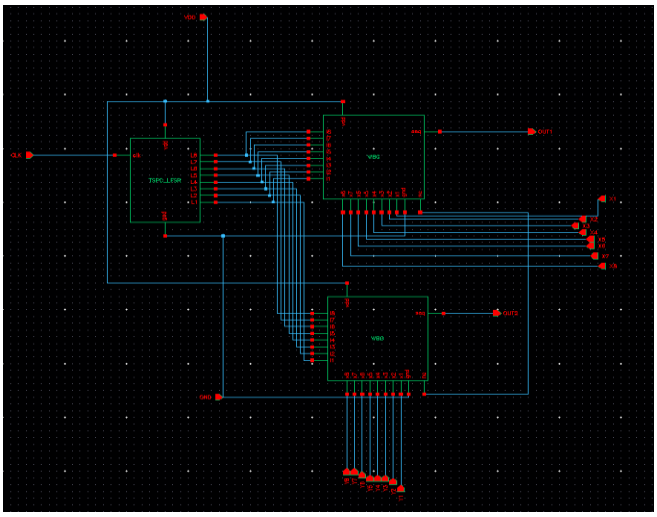

Figure 11: 8-bit SNG using TSPC logic

The figure 12 shows the output of the TSPC based SNG.

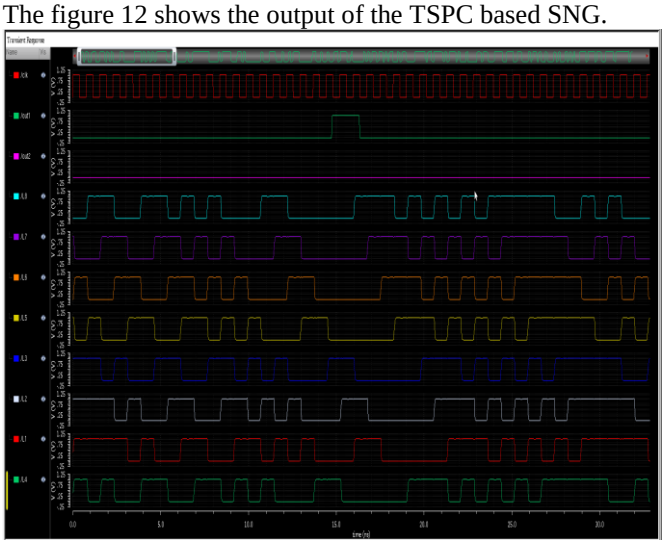

Figure 12: Output of 8-bit SNG using TSPC logic

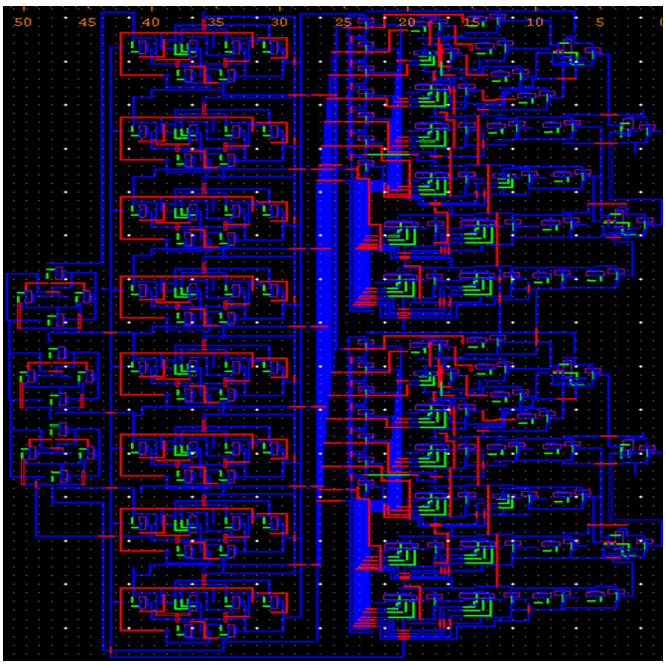

Figure 13: Layout of SNG using CMOS logic

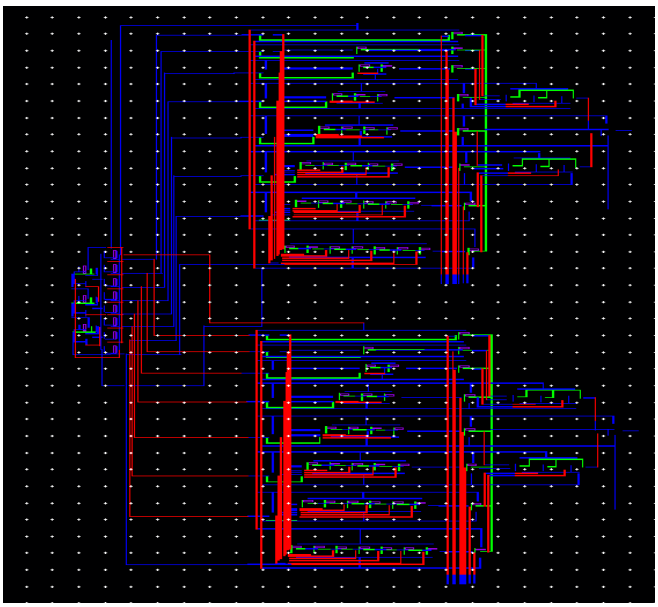

Figure 14: Layout of SNG using TSPC logic

5. Comparison of Results

Table 1: Comparison of results between CMOS and TSPC in 45nm Technology

Logics	CMOS	TSPC
No. of Transistors	**640**	736
Area (in pm^2)	38.85	**35.95**
Average power (in mW)	**8.736**	15.07
Working frequency (in GHz)	1.3	**2**

In the above table it can be observedthat the values of different parameters in both logics. Here the SNG in CMOS logic uses less no. of transistors and the average power consumed is also low with respect to TSPC logic. SNG in TSPC logic consumes less area compared to CMOS. Therefore, it can be told that logics can be used according to the specifications and requirements.

6. Conclusion and Future Works

In this work, the schematic has been implemented SNG in different parameters by designing it in CMOS and TSPC logics in 45nm technology. Not only PCC, each and every component has been designed in corresponding logic to observe the mentioned constraints and to declare which logic based SNG is better. Improvements like reduction in area, reduction in number of transistors used, reduction in total power consumed and increase in working frequency is obtained in few logics.

According to observations, it can conclude that, if a SNG has to be designed with less no. transistors and less average power consumption CMOS based SNG can be used The TSPC based SNG can be used when less area constraints are to be satisfied and for high-speed applications.

The future scope in this field can be designing the SNG in many other different logics and observing which works in the better way. Furthermore, we can decompose the WBG into two parts, the set of AND gates in the first level that produces the intermediate signals and the remaining set of gates that produce the final output. This decomposition can be implemented in many other logics and corresponding outputs can be observed an experiment can be done by mixing the logics of few circuits for better outputs.

References

[1] Tawada, Masashi, and Nozomu Togawa. "Designing stochastic number generators sharing a random number source based on the randomization function." In 2020 18th IEEE international new circuits and systems conference (NEWCAS), pp. 271-274. IEEE, 2020.

[2] Sehwag, Vikash, N. Prasad, and Indrajit Chakrabarti. "A parallel stochastic number generator with bit permutation networks." IEEE Transactions on Circuits and Systems II: Express Briefs 65, no. 2 (2017): 231-235.

[3] Bharadwaj, D. Aneesh, P. Anirvinnan, and B. S. Premanada. "A Low Power Diffused Bit Generator as a TRNG for Cryptographic Key Generation." In 2021 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), pp. 187-192. IEEE, 2021.

[4] Yang, Meng, John P. Hayes, Deliang Fan, and Weikang Qian. "Design of accurate stochastic number generators with noisy emerging devices for stochastic computing." In 2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 638-644. IEEE, 2017.

[5] Neugebauer, Florian, Ilia Polian, and John P. Hayes. "Building a better random number generator for stochastic computing." In 2017 Euromicro Conference on Digital System Design (DSD), pp. 1-8. IEEE, 2017.

[6] Salehi, Sayed Ahmad. "Low-cost stochastic number generators for stochastic computing." IEEE Transactions on Very Large Scale Integration (VLSI) Systems 28, no. 4 (2020): 992-1001.

[7] Ashwini, H., S. Rohith, and K. A. Sunitha. "Implementation of high speed and low power 5T-TSPC D flip-flop and its application." In 2016 International Conference on Communication and Signal Processing (ICCSP), pp. 0275-0279. IEEE, 2016.

[8] Bai, Yuxin, Yanwei Song, Mahdi Nazm Bojnordi, Alexander Shapiro, Eby G. Friedman, and Engin Ipek. "Back to the future: Current-mode processor in the era of deeply scaled CMOS." IEEE Transactions on Very Large Scale Integration (VLSI) Systems 24, no. 4 (2015): 1266-1279.

[9] Hossain, Md Sazzad, Mateus Bernardino Moreira, Francois Sandrez, Francois Rivet, Herve Lapuyade, and Yann Deval. "Low Power Frequency Dividers using TSPC logic in 28nm FDSOI Technology." In 2022 IEEE 13th Latin America Symposium on Circuits and System (LASCAS), pp. 1-4. IEEE, 2022.

[10] Mishra, Asirbad, Harshita Rai, Shilpi Birla, Neha Singh, and Neeraj Kumar Shukla. "Investigation of Timing Issues of True Single-Phase Clock Circuits for Nanodevices." In Intelligent Computing Techniques for Smart Energy Systems: Proceedings of ICTSES 2021, pp. 53-61. Singapore: Springer Nature Singapore, 2022.

[11] Bishnoi, Suman, Shubham Gupta, Deepak Bhatia, and Riyaz Ahmed. "45 nm CMOS-Based MTSPC DFF Design for High Frequency Operation." In Proceedings of the Third International Conference on Information Management and Machine Intelligence: ICIMMI 2021, pp. 409-416. Singapore: Springer Nature Singapore, 2022.

[12] Palumbo, Gaetano, and Giuseppe Scotti. "A multi-folded MCML for ultra-low-voltage high-performance in deeply scaled CMOS." IEEE Transactions on Circuits and Systems I: Regular Papers 67, no. 12 (2020): 4696-4706.

[13] Babu, A. Suresh, and B. Anand. "Modified dynamic current mode logic based LFSR for low power applications." Microprocessors and Microsystems 72 (2020): 102945.

[14] Rajkumar, K., P. Anuradha, RajeshwarraoArabelli, and J. Vasavi. "Design and Synthesis of Random Number Generator Using LFSR." In Smart Intelligent Computing and Applications, Volume 1: Proceedings of Fifth International Conference on Smart Computing and Informatics (SCI 2021), pp. 131-139. Singapore: Springer Nature Singapore, 2022.