Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue5, May , 3609- 3618

Research Article

Efficient Hamming Code Encoder & Decoder Using True Random Number Generator

G. Sai Vineeth[1] Dr. L. Padma Sree[2]

Abstract

In this paper, True random number generator (TRNG) based Hamming code encoder and decoder circuit is designed based on Gate Diffusion Input (GDI) logic to achieve error free transmission and reception in digital data communication. It is better to generate the random numbers by using the TRNG frameworks. In the existing, several works are focused on implementation of TRNG by using CMOS technology. But they consume the greater number of transistors, thus area, power consumption and delay were increasing. To overcome this problem, the TRNG based Hamming Encoder-Decoder is developed with the GDI technology. The GDI-TRNG will be functioned based on the switchable ring oscillators (SCRO. The simulations are implemented by using the H-spice software, the quantitative analysis of area, power and delay shows the GDI outperforms compared to the state of art approaches.

Keywords: SCRO, TRNG, GDI

1. INTRODUCTION

A random number generator is designed to generate a sequence of numbers without any specific pattern. In the current era the VLSI based RNG applications [1] are increased day-by-day rapidly in the human lives due to their vast applications in the different types of domains like energy metering system, medical applications-commercial applications, industrial machinery monitoring, SCADA applications, etc. A secure system requires random numbers at various stages [2], such as random key generation, initialization vectors, random nonce, etc. Randomness in the number generated by a random number generator is essential to ensure its privacy, anonymity and unpredictability. The security of most cryptographic algorithms using random number generator is based on the assumption that it is impossible to predict the random sequence by an unauthorized user. True random number generator and pseudo random number generator are the two basic types of random number generators. A TRNG [3] uses some physical phenomenon while a pseudo random number generator uses a mathematical function, initial state and a seed to generate long random sequence of numbers. Therefore, the cryptographic strength of a pseudo random number generator depends on the security of the initial state and the seed. The same initial state and seed causes pseudo-random generator to generate the same sequence. In order to improve security, the random number generator must be designed in such a way that it is impossible to predict the initial state and the seed of the generated sequence.

[1]M. tech Scholar, ECE Dept, VNRVJIET Hyderabad, India
[2]Professor, ECE Dept, VNRVJIET Hyderabad, India

Data communication the most important aspect in our daily life. Encoding based Cryptography is technique used for providing secure communication of data between the one end and another end. These cryptography algorithms are basically used for military and business purpose.
The secret data messages or data calls of the government higher officials, politicians may be trapped by the intermediate hackers to involve malpractices against government movements, thus these secret data messages should be made more secure. Data is the continuous streaming data. It consists of huge number of bits. Data signal consists of both negative and positive bit values. Data signal is defined as a special kind of message signal, but it can process different properties than conventional message signal. Data signal is a narrow band and message is a broader band signal. Data signal is represented in two forms like analog and digital form. In order to provide the secure communication between one end to another end, encryption of data messages is the only way to secure communication from trapping. To overcome these drawbacks, the GDI method is implemented as follows:

- All the basic gates and fundamental building blocks are implemented such as SCRO, DFF and CB4 by using the both CMOS and GDI technology.
- Then, by utilizing the concept of SCRO with beat frequency detection the GDI TRNG generated with high probability based random numbers.
- Finally, the output of the TRNG was applied as the input to the GDI Hamming encoder-decoder logic for providing the more secured communication with fewer errors.

  Rest of the paper is organized as follows: chapter 2 deals with the detailed analysis of the various state of art approaches with their drawbacks. Chapter 3 deals the implementation details of GDI method with detailed operation. Chapter 4 deals with the simulations of GDI method and comparison with the state of art approaches. Finally, chapter 5 deals with the conclusion and possible future studies.

2. LITERATURE SURVEY

There are several journals discussed regarding Viterbi decoder [11] realization in VLSI environment. In [12] authors, discussed about a VLSI method for area examination for a hard-soft decision-based decoder. They portray any calculations that be measured using new ACS unit. In [13] authors have projected a low-area consumed decoder with the Transpose-algorithm for trellis approach and trellis modulation. Then, a low-power method is GDI since a decoder has soaring power dissipation in trellis methods [14]. In addition to power dissipation, authors have studied on speed and delay evaluation in [15] and hardware efficiency and area utilization in [16]. All these literatures are focused to develop TRNG communication system by using conventional CMOS technology based basic gates, as the trellis method is the major approach for decoding, for reducing the number of paths and path delays, the reversible logic is preferable. As it consumes low quantum cost, low area, less power consumption and fewer delays compared to other literatures.
In [17] authors give a new design method for TRNG decoder with no input carry with one ancillary input bit. Authors have examined new QR carry adder designs with no ancillary input bit gives improved delay. The reversible decoder in the existing literature is evaluation by garbage outputs, total RL used, QC and delay.
In [18] authors have described concepts of convolutional decoder. Also, designed and implemented high cost, efficient, fault tolerant, reversible decoder. In this more garbage outputs were compensated with fewer operations. The author concluded that the decoder performs all the logical operations better than existing methods not arithmetic operations.

In [19] addressed a concept that a TRNG function can be reversible if every vector produces equal number of outputs. In this the High-speed turbo decoder design is presented by making use of control signals and RNG multiplier units. With this design author has found that the GDI design is more effective as per garbage outputs and constant inputs are considered.
In [20] authors described a novel nonprogrammable logic gate and verified its implementation in parallel decoders design using reversible multipliers. With this work author has found that the delay and are of decoder using TRNG should be maintain low.

3. PROPOSED METHODOLOGY

3.1 Gate diffusion input (GDI)

A new technology derived is known as GDI, which reduces power consumption compared to that of CMOS technology. It also helps in the reduction of transistor count and area of the circuit and similarly reduces the complexity of circuit. GDI consists of G, P, N. The AND gate in GDI method requires only 2-transistors, the GDI cell as shown in table 2 has 3-input pins known as P, N & G. Using GDI we reduced the transistors as shown above for basic gates of and, or requires about 4 transistors for each gate in the CMOS technology. We reduced half of its count by using GDI methodology.

N	P	G	Out	Function
'0'	B	A	$\overline{A}B$	F1
B	'1'	A	$\overline{A}+B$	F2
'1'	B	A	$A+B$	OR
B	'0'	A	AB	AND
C	B	A	$\overline{A}B+AC$	MUX
'0'	'1'	A	$\overline{A}$	NOT

Table 1: GDI basic cell

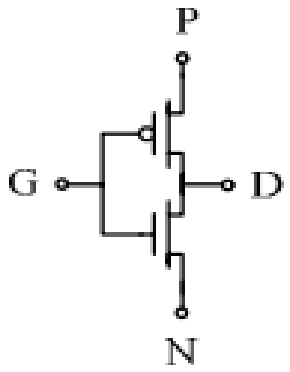


Figure 1: GDI Cell Structure

3.2 TRNG Frameworks

The block diagram of TRNG based hamming encoder-decoder method is presented in the Figure 2 respectively. The detailed architectures of the TRNG presented in Figure 3 and Figure 4 respectively. Here, the figure 3 dissipates the block diagram of TRNG, whereas the Figure 4

indicates the circuit level diagram. Each and every block in the Figure 4 was implemented by using the priorities of GDI cells. It means SCRO, 2 to 1 Multiplexer, Phase detector, Reset-Set (RS) latch and 4-Bit binary counter (CB4) were developed by using the GDI gates.

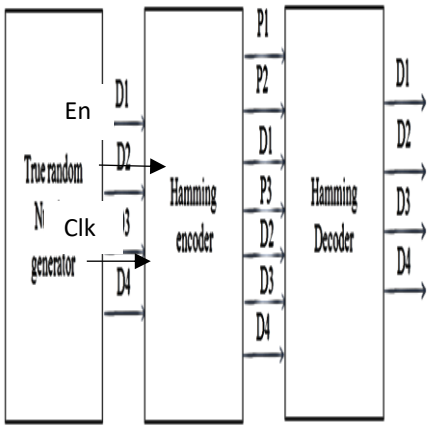


Figure 2: **Block-diagram**

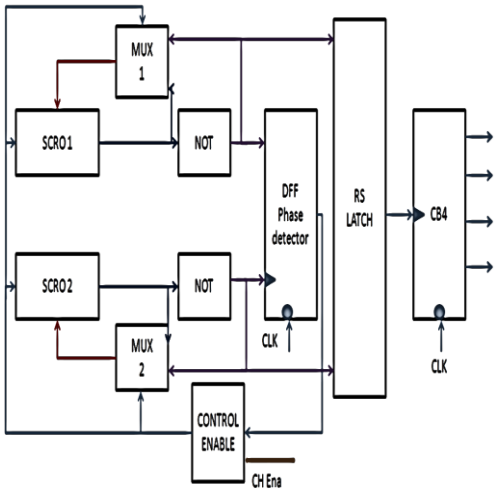


Figure 3: TRNG block diagram

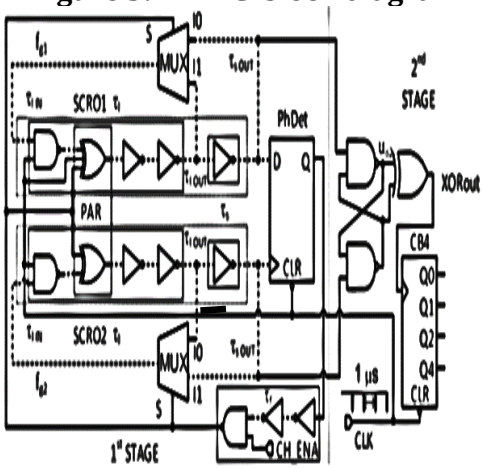


Figure 4: TRNG circuit level diagram

The detailed procedure of the TRNG method as follows:

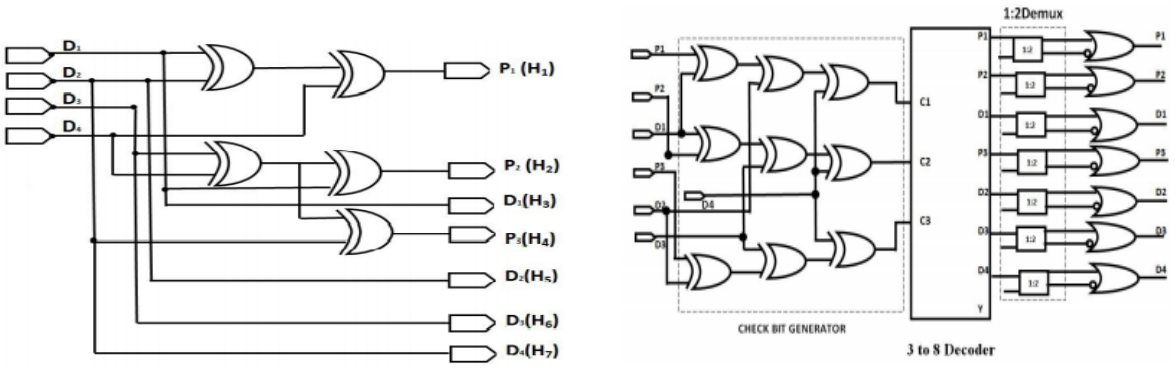
By initializing, the Chanel enable (CH ENA) to active low, the circuit starts functioning. The CH ENA input and phase detector output is applied as inputs to the control enable block. Control enable output will be applied as the selection input to the two individual multiplexers MUX1 and MUX2 respectively. Here, the multiplexer is used to selection of optimal path of SCRO. The control enable is also applied as one of the inputs to the SCRO to its XOR gate respectively. The major functionality of the TRNG will be depended on the beat frequency of SCRO. Here, SCRO is a digital oscillator, which is used to generate the various frequencies. The frequency of SCRO1 is not same as the frequency of SCRO2. If the both frequencies are same then there is less probability of generation of random sequences. For this purpose, the 3 input XOR gate is used in the SCRO1 block, whereas 2-input XOR gate is used in the SCRO2 block respectively. In order to calculate the phase differences between two frequencies, the outputs of both SCRO will be applied as inputs to the D-Flip Flop. Here, D-FF is used as the phase detector, and calculates the phase difference between two signals respectively. SCRO1 output will be applied as data input and SCRO2 output will be applied as the clock input to the DFF. Thus, with respect to the SCRO2 clock triggering, SCRO1 data will be monitored and results the output as phase detected output. The controlled SCRO 1 and SCRO 2 clock output frequencies will be applied as inputs to the Set-Reset (RS) latch. It will trigger the output and generates the final enable signal for the CB4. The RS latch enable signal will be applied as input to the CB4, as CB4 is a counter it will generates the output sequences randomly. The randomization majorly dependent on the triggering of SR latch output set and reset conditions respectively.

3.3 Encoder-Decoder module

The TRNG output will be applied as input to the Hamming Encoder and decoder module. A finest structure for Hamming-Code Encoder & Decoder is shown in Figure 5 respectively. Initially, D1, D2, D3 and D4 are the input data sources, and corresponding outputs are Out0 to out 6. The parity symbols P1, P2, P3 are user defined, accordingly connections between XOR gates have done. Output encoded frame format is generated as follows:

Out= [P1, P2, D1, P3, D2, D3, D4];
P1=XOR (D1, D2, D4);
P2=XOR (D1, D3, D4);
P2=XOR (D2, D3, D4);

After successful completion of encoding operation, the encoded codeword's are transmitted into channel. Generally, the channel consists of lots of Gaussian noise, random noise and AWGN noise. The encoded codeword's will be affected by this noise, thus error will be added. Thus, the major task of the decoder is to remove the error from encoded data instead of decoding it. For this purpose, the PBC block in the Hamming decoder is useful to check the error status. It will monitor the every codeword and if error presents, it will identify the type of noise added then it will alert the 3 to 8 decoder blocks, if there is no error PBC block simply decodes the codeword's by using branch metrics. These branch metrics are formed by the multiple combination parity symbols with their low to high probabilities.



(a) (b)

Figure 5: **Hamming encoder and decoder**

After identifying the error and noise type status in PBC unit, it is necessity to locate the error in codeword. Thus, the error location identification will be done by the 3 to 8 decoder unit. Generally, the 3 to 8 decoder contains the trellis methodology to identify the error location. These compared coefficients will then apply to XOR gates for approximate addition.

D0	D1	D2	D3	P1	P2	P3
0	0	0	0	0	0	0
0	0	0	1	1	1	0
0	0	1	0	0	1	0
0	0	1	1	1	0	0
0	1	0	0	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1
0	1	1	1	0	0	1
1	0	0	0	1	1	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	0	1	1	0	1	0
1	1	0	0	0	1	1
1	1	0	1	1	0	1
1	1	1	0	0	0	1
1	1	1	1	1	1	1

Table 2: Encoding Matrix with parity bits

D0	D1	D2	D3	P1	P2	P3	P11	P22	P33	D00	D11	D22	D33
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	1	1	0	0	0	0	1
0	0	1	0	0	1	0	0	1	0	0	0	1	0
0	0	1	1	1	0	0	1	0	0	0	0	1	1
0	1	0	0	1	0	1	1	0	1	0	1	0	0
0	1	0	1	0	1	1	0	1	1	0	1	0	1
0	1	1	0	1	1	1	1	1	1	0	1	1	0
0	1	1	1	0	0	1	0	0	1	0	1	1	1

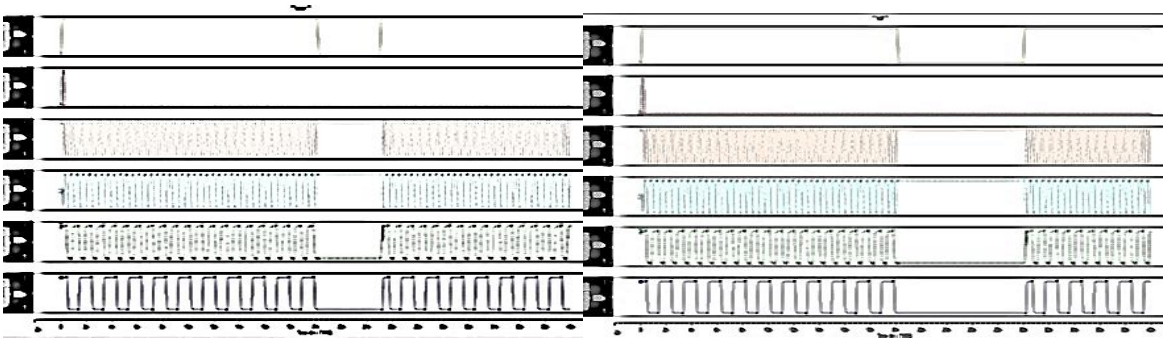
1	0	0	0	1	1	0	1	1	0	1	0	0	0
1	0	0	1	0	0	0	0	0	0	1	0	0	1
1	0	1	0	1	0	0	1	0	0	1	0	1	0
1	0	1	1	0	1	0	0	1	0	1	0	1	1
1	1	0	0	0	1	1	0	1	1	1	1	0	0
1	1	0	1	1	0	1	1	0	1	1	1	0	1
1	1	1	0	0	0	1	0	0	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 3: Decoding Matrix with parity bits

The final error location identification and error correction done in the error correction block, here each and every path will be monitored with respect to the approximation coefficients generated in 3 to 8 decoder blocks. Here prioritized error elimination metrics will be generated in each path, so they termed as path metrics also. By performing the bit modifications in these survived paths decoded error free data will generates.

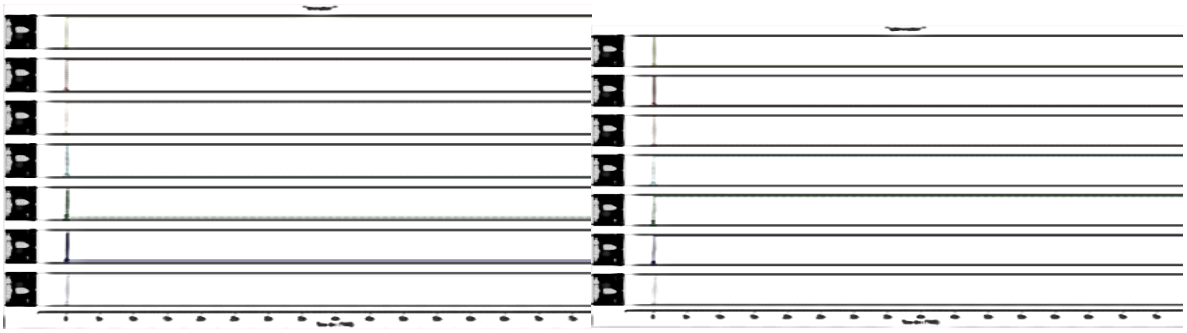
4. SIMULATION RESULTS

All the GDI designs have been designed using H-spice software this software tool provides the two categories of outputs named as simulation and synthesis. The simulation results give the detailed analysis of GDI design with respect to inputs, output byte level combinations. Through simulation analysis of accuracy of the encoding, decoding process estimated easily by applying the different combination inputs and by monitoring various outputs. Through the synthesis results the utilization of area with respect to the Transistor count will be achieved. And also, time summary with respect to various path delays will be obtained and power summary generated using the static and dynamic power consumed.



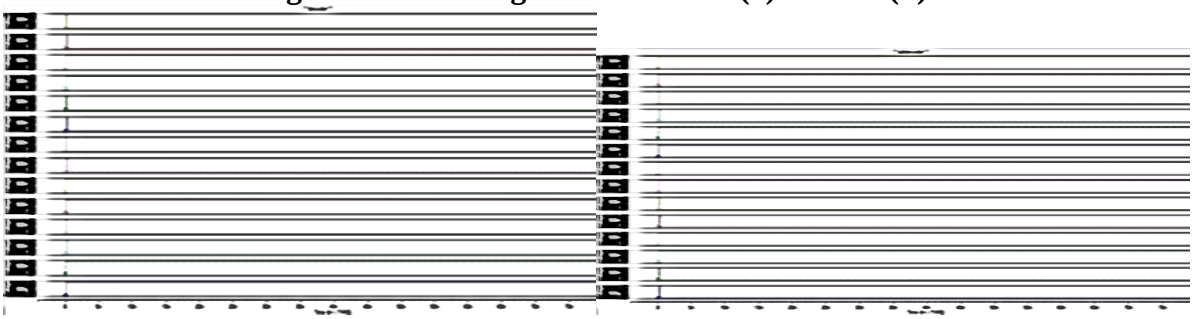
(a) (b)

Figure 6: **TRNG (a) CMOS (b) GDI**



(a) (b)

Figure 7: **Hamming-Code Encoder (a) CMOS (b) GDI**



(a) (b)

Figure 8: **Hamming-Code Decoder (a) CMOS (b) GDI**

From figure 6,7,8 it is observed that the CMOS method outputs are affected by the threshold noise levels, but the GDI outcomes of TRNG, encoder and decoder results in the better outcomes.

Technique	**Power**	**No. of Transistors**	**Delay**
	(a)	**TRNG**	
CMOS	**30.27 uw**	**288**	**31.754 us**
GDI	**0.8283uw**	**174**	**13.82 ns**
	(b)	**Encoder**	
CMOS	**41.39 nw**	**84**	**210.5 ps**
GDI	**0.106 nw**	**24**	**10.52 ps**
	(c)	**Decoder**	
CMOS	**14.46 uw**	**368**	**60.52 us**
GDI	**25.43 pw**	**130**	**36.40 ns**

Table 4: Comparison of CMOS and GDI technologies on GDI designs

From the Table 4, it is observed that the GDI method with GDI technology gives the better results compared to the standard CMOS technology for TRNG, encoder and decoder respectively.

5. CONCLUSION

This work was majorly focusing on implementation of TRNG by using the GDI based technology. For implementing the GDI-TRNG, the SCRO based mechanism has been adapted with the beat frequency-based detection concepts. Thus, the probabilities of occurrences of random numbers are increased and oscillations in the frequency also improved with the reduced error rates. Finally, the output of TRNG was applied as input to the Hamming encoder-decoder based mechanism for securely transmission of random numbers in the various communication channels. The simulation results by using H-spice software shows that the GDI technology gives the better outcomes compared to the state of art approaches. The work can be extended to implement the real time secured protocols, such as public key cryptography.

References:

[1]. Jiang, Hao, et al. "A novel true random number generator based on a stochastic diffusive memristor." Nature communications 8.1 (2017): 1-9.

[2]. Tuna, Murat, et al. "Hyperjerk multiscroll oscillators with megastability: analysis, FPGA implementation and a novel ANN-ring-based true random number generator." AEU-International Journal of Electronics and Communications 112 (2019): 152941.

[3]. Tuna, Murat, et al. "Hyperjerk multiscroll oscillators with megastability: analysis, FPGA implementation and a novel ANN-ring-based true random number generator." AEU-International Journal of Electronics and Communications 112 (2019): 152941.

[4]. Yu, Fei, et al. "A survey on true random number generators based on chaos." Discrete Dynamics in Nature and Society 2019 (2019).

[5]. Koyuncu, Ismail, et al. "Design, FPGA implementation and statistical analysis of chaos-ring based dual entropy core true random number generator." Analog Integrated Circuits and Signal Processing 102.2 (2020): 445-456.

[6]. Gaviria Rojas, William A., et al. "Solution-processed carbon nanotube true random number generator." Nano letters 17.8 (2017): 4976-4981.

[7]. Abutaleb, M. M. "A novel true random number generator based on QCA nanocomputing." Nano Communication Networks 17 (2018): 14-20.

[8]. Koyuncu, Ismail, and Ahmet Turan Özcerit. "The design and realization of a new high speed FPGA-based chaotic true random number generator." Computers & Electrical Engineering 58 (2017): 203-214.

[9]. Lee, Kyungroul, et al. "TRNG (True Random Number Generator) method using visible spectrum for secure communication on 5G network." IEEE Access 6 (2018): 12838-12847.

[10]. Qu, Yuanzhuo, et al. "A true random number generator based on parallel STT-MTJs." Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017. IEEE, 2017.

[11]. Karakaya, Barış, Vedat Çelik, and Arif Gülten. "Chaotic cellular neural network-based true random number generator." International Journal of Circuit Theory and Applications 45.11 (2017): 1885-1897.

[12]. Brown, James, et al. "A low-power and high-speed True Random Number Generator using generated RTN." 2018 IEEE Symposium on VLSI Technology. IEEE, 2018.

[13]. Satpathy, Sudhir K., et al. "An all-digital unified physically unclonable function and true random number generator featuring self-calibrating hierarchical Von Neumann extraction in 14-nm tri-gate CMOS." IEEE Journal of Solid-State Circuits 54.4 (2019): 1074-1085.

[14]. Lee, Hochul, et al. "Design of high-throughput and low-power true random number generator utilizing perpendicularly magnetized voltage-controlled magnetic tunnel junction." AIP Advances 7.5 (2017): 055934.
[15]. Jerry, Matthew, et al. "Stochastic insulator-to-metal phase transition-based true random number generator." IEEE Electron Device Letters 39.1 (2017): 139-142.
[16]. Kaya, Turgay. "A true random number generator based on a Chua and RO-PUF: design, implementation and statistical analysis." Analog Integrated Circuits and Signal Processing 102.2 (2020): 415-426.
[17]. Kim, Eunhwan, Minah Lee, and Jae-Joon Kim. "8.2 8Mb/s 28Mb/mJ robust true-random-number generator in 65nm CMOS based on differential ring oscillator with feedback resistors." 2017 IEEE International Solid-State Circuits Conference (ISSCC). IEEE, 2017.
[18]. Wieczorek, Piotr Zbigniew, and Krzysztof Gołofit. "True random number generator based on flip-flop resolve time instability boosted by random chaotic source." IEEE Transactions on Circuits and Systems I: Regular Papers 65.4 (2017): 1279-1292.
[19]. Arslan Tuncer, Seda, and Turgay Kaya. "True random number generation from bioelectrical and physical signals." Computational and mathematical methods in medicine 2018 (2018).
[20]. Song, Min, et al. "Power and area efficient stochastic artificial neural networks using spin–orbit torque-based true random number generator." Applied Physics Letters 118.5 (2021): 052401.