END-TRUE: Emerging Nanotechnology-Based Double-Throughput True Random Number Generator

Shubham Rai, Nishant Gupta, Abhiroop Bhattacharjee, Ansh Rupani, Michael Raitza, Jens Trommer, Thomas Mikolajick, Akash Kumar

▶ **To cite this version:**

Shubham Rai, Nishant Gupta, Abhiroop Bhattacharjee, Ansh Rupani, Michael Raitza, et al.. END-TRUE: Emerging Nanotechnology-Based Double-Throughput True Random Number Generator. 29th IFIP/IEEE International Conference on Very Large Scale Integration - System on a Chip (VLSI-SoC), Oct 2021, Singapore, Singapore. pp.175-203, 10.1007/978-3-031-16818-5_9 . hal-04419566

HAL Id: hal-04419566
https://inria.hal.science/hal-04419566v1

Submitted on 26 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

END-TRUE: Emerging Nanotechnology-based Double-Throughput True Random Number Generator

Shubham Rai[1], Nishant Gupta[1], Abhiroop Bhattacharjee[1], Ansh Rupani[1], Michael Raitza[1], Jens Trommer[2], Thomas Mikolajick[2], Akash Kumar[1]

[1] Chair of Processor Design, Technische Universität Dresden, Germany
[2] NaMLab gGmbH, Dresden, Germany

1 Abstract

True Random Number Generators (TRNGs) are essential primitives in any cryptographic system. They provide the foundation to secure authorization and authentication. This work proposes a generator that exploits the metastability effect of cross-coupled logic gates, as found in SR latches. Based on emerging reconfigurable transistor technology, a random number generator design has been proposed that doubles the throughput, compared to a similar standard CMOS design, by exploiting transistor-level reconfiguration. The proposed design is superior in terms of the number of transistors per block, power consumption and in critical path delay with respect to its CMOS counterpart. Random Number bit sequence are generated by operating the given design at three operating frequencies of 10 MHz, 100 MHz and 200 MHz. Firstly, the Shannon entropy for the generated bit sequence is measured, and then the generated bit sequence are subjected to statistical evaluation using the NIST benchmark suite. The P' values for the NIST benchmarks is above the accepted threshold, which underlines the assumption that the designed circuit produces the random numbers based on the metastability effect.

Keywords: Reconfigurable Field Effect Transistor (RFET), True Random Number Generator (TRNG), Metastability, NIST benchmark suite, Von-Neumann extraction

2 Introduction

Hardware security is an area of prime concern in the current era of *Internet of Things* (IoT) owing to the growing security threats and adversarial vulnerabilities to embedded devices [1]. To this end, reliable and secure hardware primitives are required to be interfaced with low-cost and resource-constrained embedded devices for secure communication, identification or authorization and privacy protection. These security primitives can manage digital keys, perform

encryption and decryption for digital signatures, strong authentication and various other cryptographic functions [33]. Generating secrets is a cornerstone of cryptographic applications [39, 11, 19, 1], whose randomness, as a measure of unpredictability, is the defining property of a secure secret key.

Random Number Generators (RNGs) produce uniformly distributed sets of numbers for secret keys. There are two kinds of RNGs: True Random Number Generators (TRNGs) and Deterministic Random Number Generators (DRNGs). Hardware-based TRNGs extract the noise from chaotic physical processes in the form of an unpredictable sequence of bits (e.g., thermal noise, flicker noise, clock-jitter, metastable states, power supply fluctuations) [19, 41, 1]. On the other hand, DRNGs use one or more inputs known as 'seeds' and are used in generating '*pseudo random numbers.*' These numbers are generated using deterministic algorithms and satisfy every requirement posed by random numbers. However, the only drawback is that these sequences can easily be retraced if the seed is known. Hence, to make DRNGs truly random, the seeds must be generated from some TRNG. A TRNG comprises of three main components: 1) an entropy source, like the one proposed in this work. It is used to generate unpredictable and independent values. 2) A conditioning component, which is usually optional. It is used to reduce the bias in the random outputs. 3) A health test, which checks for the failure of the entropy source [61].

There are various places where a TRNG can be used, including providing security against existing adversarial attacks, such as spoofing and cloning, because of their ability to generate unique secret keys [1]. They are also used as initialization vectors, random masks, challenges and nonces in side-channel attack countermeasures [15]. To guarantee a high level of random output, a compromise in terms of speed, power and area is generally considered [29]. However, TRNGs which are to be installed in embedded devices have certain criterion to fulfil. They need to be area and power efficient and should utilize existing hardware elements such as logic gates for random number generation [9, 48, 61]. Hence, it is imperative to explore emerging nanotechnology-based solutions.

Recent developments in emerging technologies have opened up new avenues to bring security from the technology side within the hardware system [42, 2, 68, 49]. Various emerging technologies have been explored in works such as [6, 41, 66, 36] which cater for random number generations with low power requirements.

Runtime reconfigurable technologies form an interesting class of such emerging devices. Transistors based on these technologies (such as silicon [18, 7] or germanium [63, 56] nanowires) show electrical symmetry and can be reconfigured between p- and n-type behavior at runtime. Due to their transistor-level reconfiguration, devices made of such nanotechnologies are often termed as *Reconfigurable FETs* (RFETs). RFETs can encapsulate more logic and functionality into a smaller area and are able to achieve reduced power consumption and higher speed during their operation [51, 35, 75, 45, 41]. Due to their extended functionality, they have shown great potential for hardware security applications, particularly in the domain of logic locking and layout camouflaging [4, 3, 51, 6, 2, 46, 51].

In this work, a TRNG design based on RFETs that is referred as *Emerging Nanotechnology - based Double - Throughput True Random Number Generator* (END-TRUE) is proposed. It introduces a *bistable* device/circuit responsible for generating random numbers. The occurrence of a *metastable* state in any bistable circuit is unavoidable. There are three static equilibrium points in the proposed circuit, two of which are stable (bistable) and known as accurate output states, while the third point is known to be the metastable state. The metastable state, in simple terms, can be defined as a state where the output of the circuit is unpredictable. The output can settle down to either of the two stable states (the bistable states). The proposed TRNG consists of a metastable RFET-based SR latch and a dual-edge triggered *True-Single Phased Clock D-Flip Flop* (TSPC DFF). Conventionally, each random bit generated using metastability-based TRNGs is a result of two cross-coupled elements entering into a metastable state at the rising (or falling) edge of an input clock signal. This implies that each clock period translates to one random bit at the output, thereby making the throughput of the TRNG equal to the input clock frequency. However, in this design, the property of transistor-level reconfigurability allows to have both cross-coupled NAND and NOR operations in a single clock cycle, which are triggered into metastable states at the rising and falling clock edges respectively, thereby generating two random bits per clock cycle. However, when the design is fabricated, it introduces some device variations which influences the outcome of the TRNG. As this mismatch increases, the 50-50 probability of resolving the output to two different stable states gets unbalanced [40].

The proposed TRNG is able to generate random numbers achieving double-throughput over a similar architecture of a TRNG using CMOS technology (which would need additional hardware components for reconfiguration). This work also shows that the END-TRUE is more efficient in terms of area (60% saving in number of transistors), delay (77.3% reduction) and power consumption (94.5% lower leakage power and 70.7% lower dynamic power) over its CMOS counterpart.

The present work is an extension of an earlier work [1]. Compared to the previous work, this work contains a detailed experimental evaluation and discusses PVT robustness in the case of RFETs, which is integral to any TRNG design.

Contributions: Major contributions of this work are as follows:

- Use of a Verilog-A model from a predictive RFETs design kit [14], to propose a *Minority*-based SR latch which allows reconfiguration between a NAND and NOR-based SR latch. This is essential to achieve double throughput and forms the core for the proposed TRNG.
- An improved design for a reconfigurable dual edge-triggered TSPC D-flip flop using RFETs based on TSPC logic. This allows random number generation at both edges of the clock.
- It has been demonstrated the runtime reconfigurability of RFETs can be exploited to design the double-throughput TRNG (END-TRUE) using less

hardware than its CMOS counterpart. The proposed design is better in terms of transistor count, power consumption and critical path delay.
– It has been further shown that the raw random bit sequence obtained from the TRNG, upon post-processing using Von-Neumann extraction have sufficient entropy to pass the statistical tests.

Experimental evaluation over NIST benchmark suite [50] at three different frequencies – 10 MHz, 100 MHz and 200MHz, demonstrate that the proposed END-TRUE returns raw bit sequence with high values of Shannon entropy.

The remainder of the chapter is organised as follows: Section 3 presents the fundamentals of RFET device operation and RFET-based circuits. It also describes various kinds of TRNGs with an emphasis on metastability-based TRNGs. Section 4 deals with the circuit design of the reconfigurable dual edge-triggered D-flip flop followed by the proposed design. Section 5 begins with the simulation results using the proposed design and presents a comparison between the END-TRUE and its equivalent CMOS counterpart on the basis of number of transistors, power consumption and critical path delay. It culminates with the results of various statistical tests carried out on the raw bitstreams generated from END-TRUE. Section 6 involves analysis of the test results along with assessment of the impact of post-processing (using Von-Neumann extraction) on the raw bit sequences from the END-TRUE. Some consideration on the impact of process, voltage and temperature (PVT) variations on the TRNG functionality are given. Finally, Section 7 presents the conclusions to this chapter.

3 Background

3.1 Reconfigurable FETs

Reconfigurable transistor functionality has been demonstrated on a variety of materials such as 1D silicon [18, 7] or germanium nanowires [63, 56], carbon nanotubes [73], graphene nanoribbons [16], or by planar 2-D devices based on materials such as $MoTe_2$ [37] or graphene p-n junctions [58]. More recently also the integration into an existing fully depleted silicon on insulator 22nm technology was demonstrated [53]. This chapter focuses on nanowire-based RFETs since it is one of the most actively researched emerging technologies having Verilog-A models [14] as well a first physical synthesis flow [44] available.

Reconfigurable nanowire-based transistors, unlike conventional CMOS based transistors, feature two types of gate terminals, a *Program Gate* (PG) and a *Control Gate* (CG). The PG is used to reconfigure the channel between p-type and n-type by selectively suppressing the injection of one type of charge carrier. Whereas the CG receives a voltage input to the FET and modulates the flow of the other type of carrier [35, 75, 45]. This is shown in (Fig. 1). Fig. 1a shows how an RFET logically encapsulates both PMOS and NMOS together [43]. The electrical symmetry in I-V characteristics for nanowire transistors for both p- and n-type behavior can be seen in Fig. 1d [48]. This electrical symmetry is necessary while realizing complementary circuits.

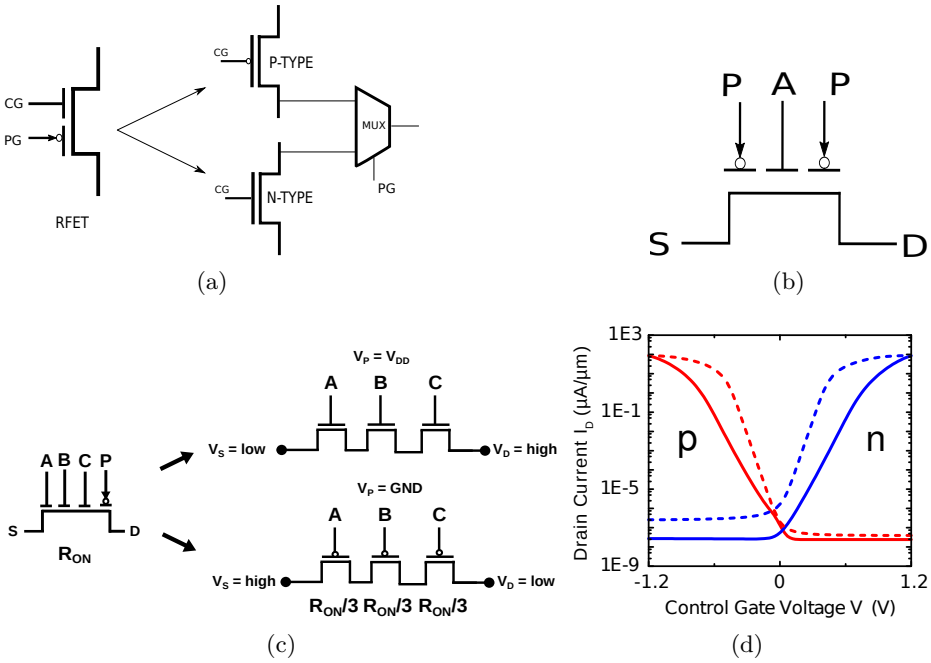

Fig. 1. (a) Transistor level-equivalent model of the RFET showing how it encapsulates both p- and n-type behavior. The runtime-reconfigurability is represented by the MUX.; (b) All-around *Three-Independent Gate FET* (TIGFET) with the control gate (A) and program gates (P) marked [7]; (c) multi-gate RFET with inputs A, B and C, and a program signal P [45]. It shows how the channel resistance is reduced as compared to a series of conventional CMOS transistors; (d) Ambipolar transfer characteristics for SiNW RFET [48]. Bold lines corresponds to the high-V_t operation and dotted lines corresponds to the low-V_t operation.

As shown in Fig. 2, to switch the device into an n-type FET, V_{PG} should be larger than 0, whereas to convert the device into p-type FET, V_{PG} should be below 0. When V_{CG} is equal to 0, then both the FETs are switched off because of the barrier induced by the opposing potential of CG and PG. On the other hand, when V_{CG} is above 0, it allows the conduction of current through tunneling in an n-type FET, and when V_{CG} is below 0, p-type FET switches ON. A small amount of thermionic emission also contributes to the overall I_{DS} current flowing through the RFET. This tunneling and thermionic current is possible because of the strong band bending which takes place at the source contact leading to injection of electrons and holes from the metal to the semiconductor through the thinned barrier respectively.

RFET can also exist is multi-independent gated form. The authors in [75, 18] have shown that multi-independent-gate RFETs allow merging of two or more series transistors in CMOS technology into a single RFET as shown in Fig. 1c. RFETs having two or more inputs on a single channel in a wired-AND [54] con-

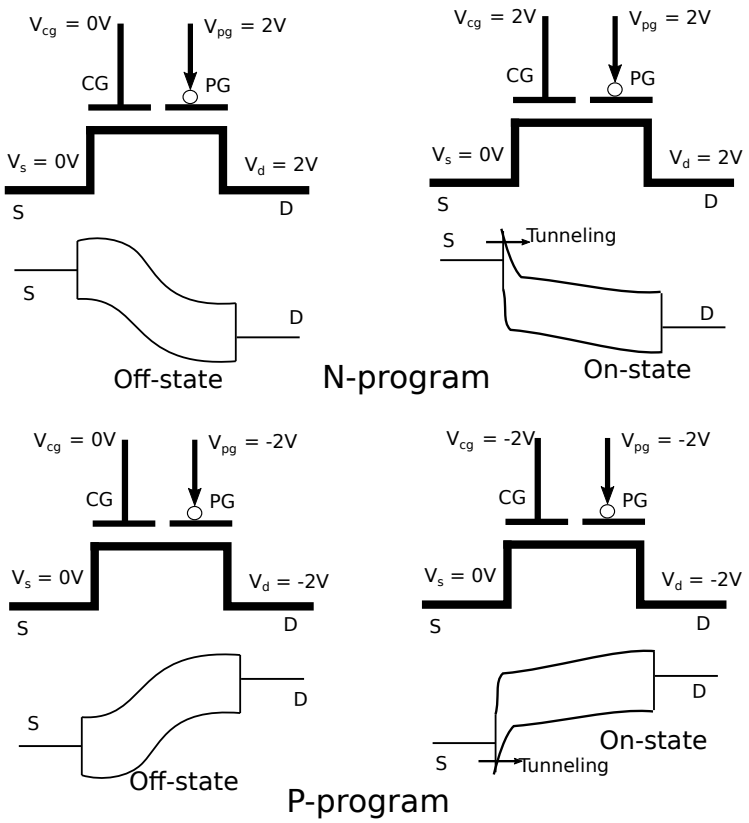

Fig. 2. Working principle of a reconfigurable FET [47]

figuration operates with a virtually lower channel resistance per input (Fig. 1c) thereby dramatically reducing transistor count in digital circuits as well as the parasitics and delays. This is due to the presence of Schottky barrier in the on-state of the RFETs [62]. This helps to design compact digital circuits with added functionalities [45].

3.2 RFET-based logic gates

In this work, an RFET variant with *Three Independent Gates* has been used to design digital circuits. The devices is thus often called TIGFET in literature. A typical layout having a single-input configuration with CG (input A) in the middle of the channel and the program signal (P) is used to alter between p-channel (P = '0') and n-channel (P = '1') behaviour is shown in Fig. 1b. Fig. 3a shows an extension of the TIGFETs for two inputs. If one chooses to operate a TIGFET as shown in Fig. 3a, then there are 8 possible configurations in which it could be operated. Out of these possible 8 configurations, only 6 are useful

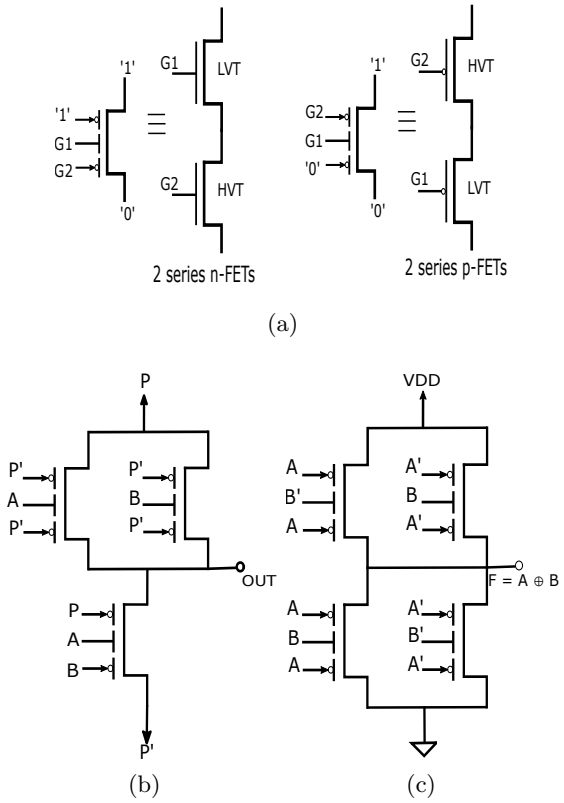

Fig. 3. (a) N-MOS and P-MOS transistor level equivalent models for TIGFETs in dual-threshold voltage configuration; (b)A configurable MIN gate behaving as a NAND gate when P = '1' and NOR gate when P = '0' (c) XOR gate

to us. These 6 configurations are (assume, $V_{DS} = V_{DD}$): 1) ON state: When the V_{PGS} (voltage of program gate at the source) = V_{PGD} (voltage of program gate at the drain) = V_{CG} (voltage of control gate present in the middle). When this condition is applied on the RFET, one of the Schottky barriers is very thin, allowing the tunneling of the majority charge carriers. 2) OFF state: This state occurs when $V_{PGD} = V_{PGS}$ and V_{CG} has opposite biasing to the polarity gates. In this state, there is still a small number of charge carriers that can cross the barrier. 3) Low-leakage OFF state: This state occurs when V_{PGS} = S and V_{PGD} = D. This condition creates a thick enough barrier across the channel, which is sufficient enough to prevent the tunneling at both ends. The unused states must be avoided while implementing the RFET in the circuit. This could be done by fixing the $V_{PGS} = 0$ ($V_{PGD} = 1$) for p-RFET (n-RFET) as shown in the Fig. 3a or by using $V_{PGD} = V_{PGS}$ [75].

The authors in [75] have shown that it is a dual-threshold voltage configuration wherein, input $G1$ has lower threshold voltage (**LVT**) and input $G2$ has higher threshold voltage (**HVT**) (corresponding to lower leakage current). This feature of configuring threshold in RFETs is used later for the TSPC-based DFF as discussed in Section 4.3. The dual-threshold feature helps in improving the leakage power of the circuit. Leakage power is a critical issue in the present-day circuits. It comprises around 30% - 50% of the current SoC power consumption. Low V_t devices are used in the paths that are critical, in order to meet the timing constraints while high V_t devices which have low leakage are used in slack paths [75].

Note - In the world of CMOS technology, implementation of multi-V_t devices is a bit difficult task. CMOS devices require an extra technological step which increases the cost of fabrication and affect the regularity of the layout. Another method which can help in decreasing the leakage power in the CMOS technology is by employing adaptive body biasing. But it adds a separate overhead in terms of area consumption by additional circuits and routing resources, thus RFET are supposed to shown an advantage over standard CMOS technology in this regard as well.

Fig. 3b presents the configuration of a configurable MIN gate that is shown to behave both as a two-input NAND gate (P = '1') and NOR gate (P = '0') [45]. Switching between NAND and NOR occurs because of the interchange in the functionality between the pull-up and the pull-down part of the circuit. This happens when the value of P is modified. In RFET technology, NAND and NOR gates, thus, can be built with equal performance owing to the electrical symmetry of the underlying devices. This helps in simplifying timing constraints [45]. Similarly, an RFET-based 2-input XOR is shown in Fig. 3c [7].

3.3 Types of TRNGs

True Random Number Generators (TRNGs) produce unpredictable numbers that originate from some stochastic physical phenomenon [34]. There have been various works on TRNGs in CMOS technology. There are three kinds of TRNG architectures - TRNGs that attribute a logic value to noise (aka. noise-based TRNGs), TRNGs that attribute a time value to noise (aka. jitter-based TRNGs) and metastability-based TRNGs that exploit the random outcome of transient metastable behavior [1].

Noise-based TRNG Noise-based TRNGs involve direct amplification of a noise source (e.g., thermal noise), followed by quantization or digitization using a comparator to produce random output [21, 5]. However, noise-based TRNGs are difficult to interface with highly dense digital ASICs owing to the presence of thermal analog sensors and amplifiers [13, 1].

Jitter-based TRNG Jitter-based TRNGs amplify the frequency noise, called jitter, of voltage-controlled oscillators (VCOs) [65, 57, 71], free-running oscillators (FROs) [13] and ring oscillators. Ring-oscillator based TRNGs have been

shown to have resilience to temperature fluctuations [65, 57]. However, factors that affect oscillator-based TRNGs include high power consumption, aging and frequency injection attacks that can result in loss of entropy [13, 31]

Metastability-based TRNG Metastability-based TRNGs use metastable circuits [25, 60, 20, 64, 32, 17, 61] to generate random numbers. They use cross-coupled elements (for ex. cross-coupled NAND gates in SR latch or cross-coupled inverters) to amplify random noise and generate random bits. The cross-coupled elements are biased precisely to attain a metastable state, which is eventually resolved to a stable random state. Any kind of unbalance/asymmetry in the circuit would cause the output of the TRNG to be biased [13]. Metastability-based TRNGs are well-suited for interfacing with embedded devices due to their small-scale and low power consumption and they have been shown to be robust against temperature and supply voltage variations [61].

4 Design of the TRNG using RFETs

The potential of using transistor-level reconfiguration in RFETs to develop compact and power-efficient circuits with less parasitics motivates to employ them for the END-TRUE design. The aim of this work is to double the throughput of generation of random binary sequences by exploiting the feature of runtime reconfigurability in *Minority* (MIN) gates based on RFETs. The present section details about the components of the proposed metastability-based TRNG.

4.1 Metastability in SR latch

The TRNGs employ the metastable state attained by cross-coupled elements as a source of randomness. Fig. 4 shows a NAND gate based SR latch unit which initially rests in a ground state when the input clock (A) has a value of '0', i.e. the outputs (B and C) of the unit are '1'. At the rising clock edge of the input clock, the output of the latch begins to race and temporarily enters into a metastable state. However, due to the random noise, the metastability is resolved and the latch eventually generates a random bit sampled using a positive-edge triggered D-flip flop at the output. Again, at the falling clock edge, the output of the latch resets itself to its ground state and the phenomenon is repeated with each clock cycle. Hence, the throughput of the TRNG is equal to the input clock frequency. The raw bit sequence generated at the output for each clock cycle would only be unbiased or perfectly random if driving capabilities of the two NAND gates are same.

4.2 Minority gate-based SR latch for END-TRUE

In the present work, the metastability-based TRNG is designed using reconfigurable MIN gates (Fig. 3b). Fig. 5 shows a single SR latch unit consisting of

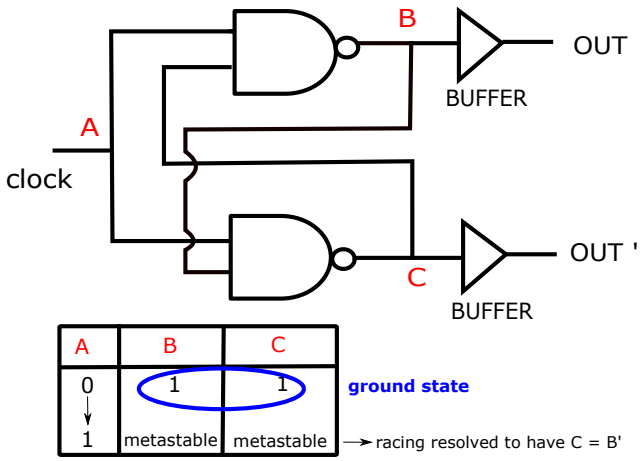

Fig. 4. An SR latch unit for the TRNG in [17] with two cross-coupled NAND gates and two buffers

two cross-coupled MIN gates and two buffers. Two clock signals ($clk\_Program$ and clk_IN) with the same time period T are fed into the unit, clk_IN being a time-delayed version of $clk\_Program$, delayed by t_d satisfying the condition $t_d < T/2$. In the first half-period of $clk\_Program$ ($clk\_Program =$ '1'), the MIN gates behave as NAND gates and the rising edge of the clk_IN signal occurs (when $clk\_IN =$ '0'), the outputs of both the gates are '1' (ground state). Post the transition in clk_IN signal, the outputs begin to race and temporarily enter into metastability. However, owing to the random noise, the output 'OUT' stabilises in order to generate a random bit ('0' or '1'). Similarly, in the second half-period of $clk\_Program$ ($clk\_Program =$ '0'), the MIN gates behave as NOR gates and the falling edge of the clk_IN signal occurs. This time in the ground state the outputs of both the gates are '0' and metastability is attained at the '1'$\rightarrow$ '0' transition of clk_IN signal, which eventually results in another random bit. Thus, in one complete clock cycle, two random bits are generated implying that the throughput of the SR latch unit is twice the input clock frequency.

4.3 Dual edge-triggered TSPC-based D-flip flop

In this section, a compact design of a dual-edge triggered TSPC-based D-flip flop using RFETs (Fig. 6) is proposed that is employed in the TRNG design. At the transistor level, flip flops require the clock signal directly and inverted. This poses challenges to the clock-tree synthesis [59]. However, TSPC-based D-flip flops require only a single clock signal [74]. This, along with the dynamic logic of TSPC-based design, leads to compactness and faster response [23]. Thus, TSPC-based D-flip flop can be used for high speed applications efficiently [59].

The authors in [59] proposed a design of a positive edge-triggered TSPC-based D-flip flop using RFETs that has been shown to have a reduced transistor

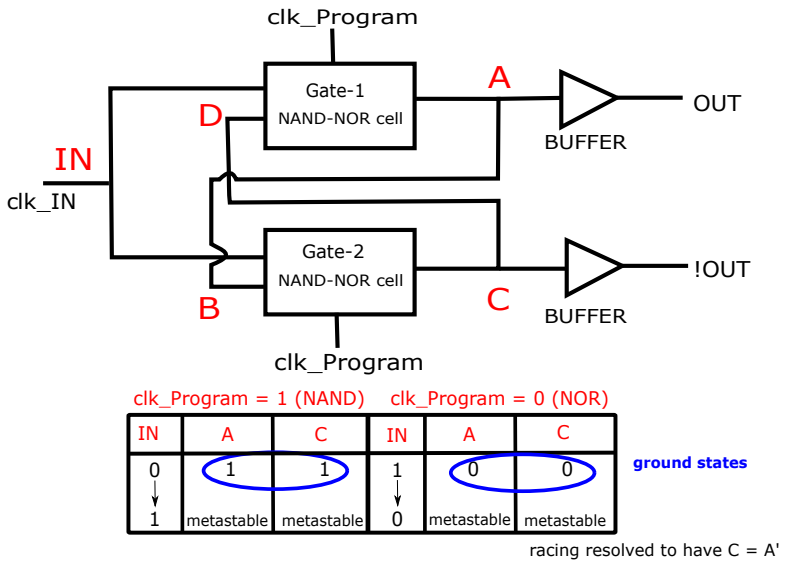

Fig. 5. An SR latch unit for the END-TRUE

count and area compared to its CMOS counterpart [74]. Furthermore, since in the design of the flip flop, each pull-up and pull-down path consists of a single transistor, the parasitics are further reduced, thereby improving the speed of the flip flop [59, 75].

This work exploits the runtime reconfigurability feature of RFETs to make the TSPC-based D-flip flop proposed in [59] dual-edge triggered. This can be done by using a program signal (P) instead of the power-rails as shown in Fig. 6. If P = '1', the upper four transistors encircled in red provide the pull-up path while the lower four transistors encircled in blue provide the pull-down path. In this case, the flip flop samples data at the rising edge of the clock and hence, behaves as a positive edge-triggered flip flop. Conversely, if P = '0', the pull-up and pull-down paths get interchanged and the flip flop samples data at the falling clock-edge. This way it behaves as a negative edge-triggered flip flop. Dual-threshold voltage design style as shown in Fig. 3a has been adopted (for three transistors encircled in purple) to make the design compact and reduce leakage power consumption.

Thus, the same circuit of the flip flop can be reconfigured into both positive and negative edge-triggered functionalities based on the program signal during runtime. However, the same TSPC-based design of a D-flip flop in CMOS technology [74] cannot be reconfigured as both positive and negative edge-triggered and it also uses a higher number of transistors (11 transistors) with respect to the proposed design in this work using RFETs (8 transistors). To the best of the author's knowledge, none of the earlier works have explored a TRNG design

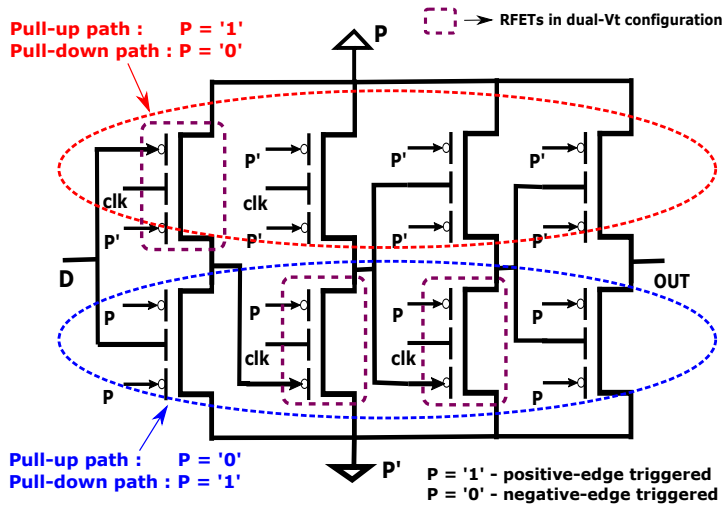

Fig. 6. A configurable dual edge-triggered D-flip flop based on TSPC logic style

using device-level reconfigurability offered by reconfigurable emerging nanotechnologies.

4.4 XOR-ing the outputs

For the purpose of simulation, the outputs of the two SR latch units are XOR-ed and its results are fed into a dual edge-triggered TSPC-based D-flip flop (Fig. 6). Compact implementation of MIN gates (Fig. 3b), XOR gate (Fig. 3c) and the proposed TSPC-based D-flip flop has been carried out using dual-threshold-voltage design style that makes the design area-efficient with improved speed and reduced leakage power consumption [75].

It has been mathematically proven in [66, 8] that by XOR-ing outputs from multiple TRNGs (in this case, the SR latch units), the randomness (entropy) of the resultant output sequence can be increased and the TRNG becomes more robust against PVT variations.

Let the i^{th} TRNG produce a probabilistic output signal (bitstream) for which probability of obtaining bit '1' is equal to p_i. In the ideal scenario, the value of p_i should be equal to 0.5 for an unbiased binary sequence (perfectly random). Let the probability deviation from the ideal value be defined as $\alpha_i = |0.5 - p_i|$, $\alpha_i \epsilon [0, 0.5]$. If two such probabilistic output signals are combined by the XOR gate, then the resultant probabilistic signal can be given as -

$$p_{XOR} = p_1(1-p_2) + p_2(1-p_1) = 0.5 \pm 2\alpha_1\alpha_2 \quad (1)$$

For n such probabilistic signals, equation (1) becomes-

$$p_{XOR} = 0.5 \pm 2^{n-1}\alpha_1\alpha_2...\alpha_n \quad (2)$$

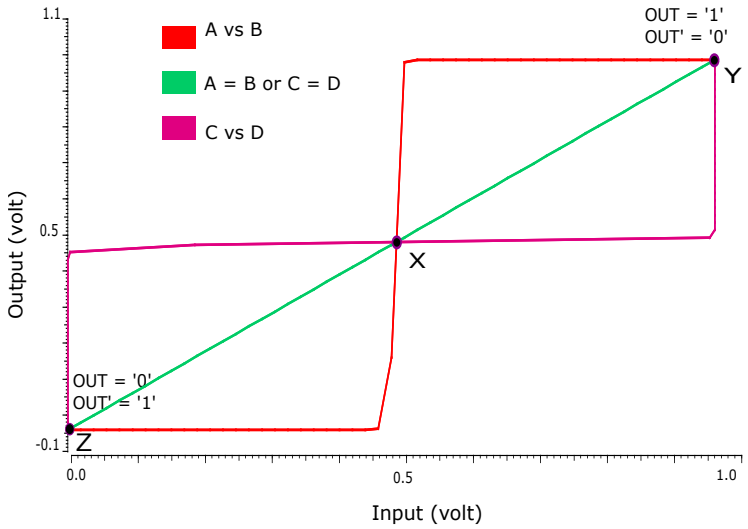

Fig. 7. Butterfly curve in the Voltage-Transfer Characteristic (VTC) for the SR latch unit of the END-TRUE

The deviation of the resultant output signal from the ideal value is, therefore, given as-

$$\alpha_{XOR} = 2^{n-1}\alpha_1\alpha_2...\alpha_n \tag{3}$$

with $\alpha_{XOR} \epsilon [0, 0.5]$. Smaller the value of α_{XOR}, higher the randomness in the output signal. Furthermore, $\alpha_{XOR} \leq min\{\alpha_1, \alpha_2, ..., \alpha_n\}$. This implies that by XOR-ing the output signals of multiple TRNGs, randomness (entropy) in the resulting signal can be improved.

4.5 Analysis of Randomness

The cross-coupled MIN gates in the SR latch unit (Fig. 5) are analogous to two cross-coupled inverters (such as in an SRAM cell) that are powered-ON when the input clock makes a '0' → '1' transition for *clk_Program* = '1' or when it makes a '1' → '0' transition for *clk_Program* = '0'. '*B*' and '*D*' are respectively the inputs to *Gate-2* and *Gate-1* while, '*A*' and '*C*' are respectively the outputs of *Gate-1* and *Gate-2*. The corresponding butterfly-curve in the *Voltage-Transfer Characteristic* (VTC) for the SR latch unit is shown in Fig. 7. It can be clearly seen that point '*X*', which is the point of metastability, lies on the identity line (shown in green). This means that both stable states demarcated by points '*Y*' and '*Z*' are equally preferred, when the MIN gates in the latch have similar drive capabilities. Eventually, the latch attains either state '*Y*' or '*Z*' due to noise, thereby producing a random bit at the output (OUT). Thus the SR latch unit in END-TRUE generates a random bit when triggered into a metastable state.

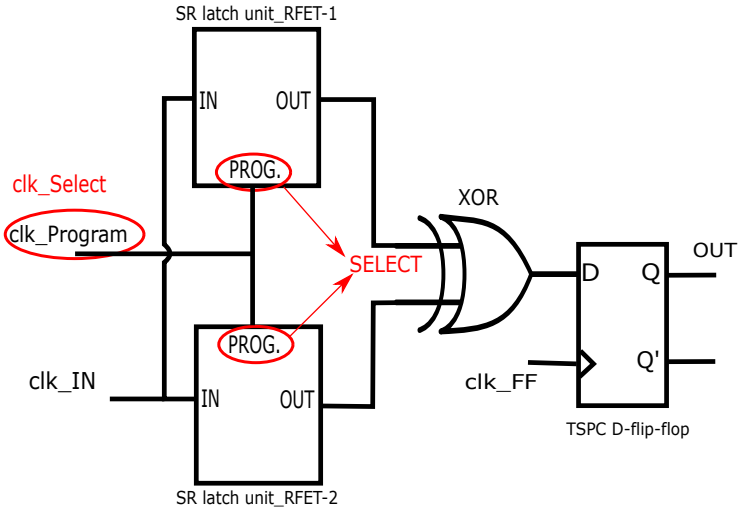

Fig. 8. The simulation model for the END-TRUE consisting of two SR latch units, an XOR gate in DG configuration and the configurable dual edge-triggered D-flip flop **(The signals and nodes in the equivalent CMOS model have been marked in red)**

Fig. 8 shows the complete circuit for the END-TRUE based on RFETs with all the components. In the next sub-section END-TRUEhas been compared with an equivalent CMOS-based design.

4.6 Comparison with CMOS-based design

Table 1. A comparison between RFET-based SR latch unit and its CMOS equivalent on no. of transistors

RFET SR latch	**No. of Transistors**	**CMOS SR latch**	**No. of Transistors**
MIN gate	6	NAND gate	8
Buffer	8	NOR gate	8
–	–	Buffer	16
–	–	2 x 1 MUX	14
TOTAL (RFET)	**14**	**TOTAL (CMOS)**	**46**

It has been shown that the transistor-level reconfigurability in RFETs helps to double random bit generation rate per clock cycle in case of the END-TRUE,

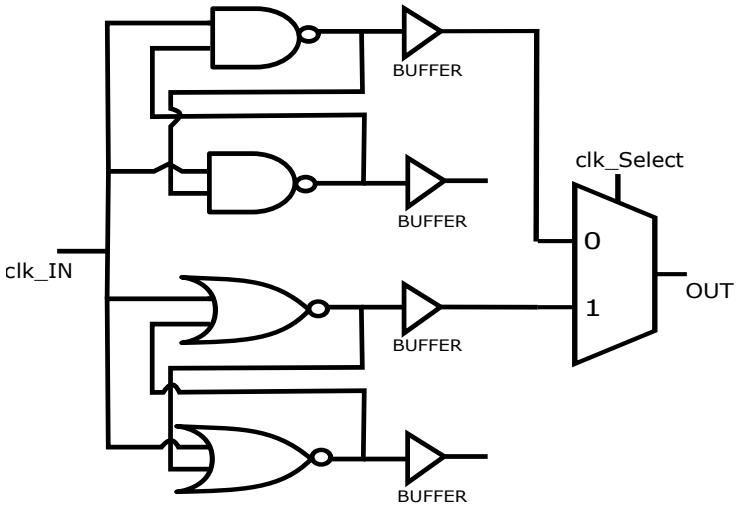

Fig. 9. An SR latch unit for the CMOS equivalent of the END-TRUE

thereby achieving double-throughput. For the same functionality to be implemented in CMOS technology, the SR latch unit consists of two cross-coupled NAND gates, two cross-coupled NOR gates, four buffers and one 2×1 MUX as shown in Fig. 9. In this case *clk_Select* is fed into the select line of the multiplexer and *clk_Select* and *clk_IN* are clock signals having the same time period T, the latter being a t_d time-delayed version of the former satisfying the condition $t_d < T/2$. In one clock cycle of *clk_Select*, two random bits are generated at the output 'OUT' - one corresponding to the metastability of NAND-based SR latch in one half cycle (*clk_Select* = '0') and the other corresponding to the metastability of NOR-based SR latch in the other half cycle (*clk_Select* = '1'). This way a throughput equal to twice the input clock frequency is obtained at the cost of additional hardware and greater number of transistors with respect to the RFET-based implementation.

A tabular comparison of the number of transistors for implementation of the RFET based SR latch unit and its CMOS equivalent is presented in Table 1. It can be observed that there is 69.6% saving in transistor count by using RFET technology.

5 Experiments

5.1 Experimental setup

The simulation of the END-TRUE has been carried out in Cadence Virtuoso. The Verilog-A model for the RFET in three-independent gate configuration (TIGFET) from [14] was used during the circuit-level simulations. This model

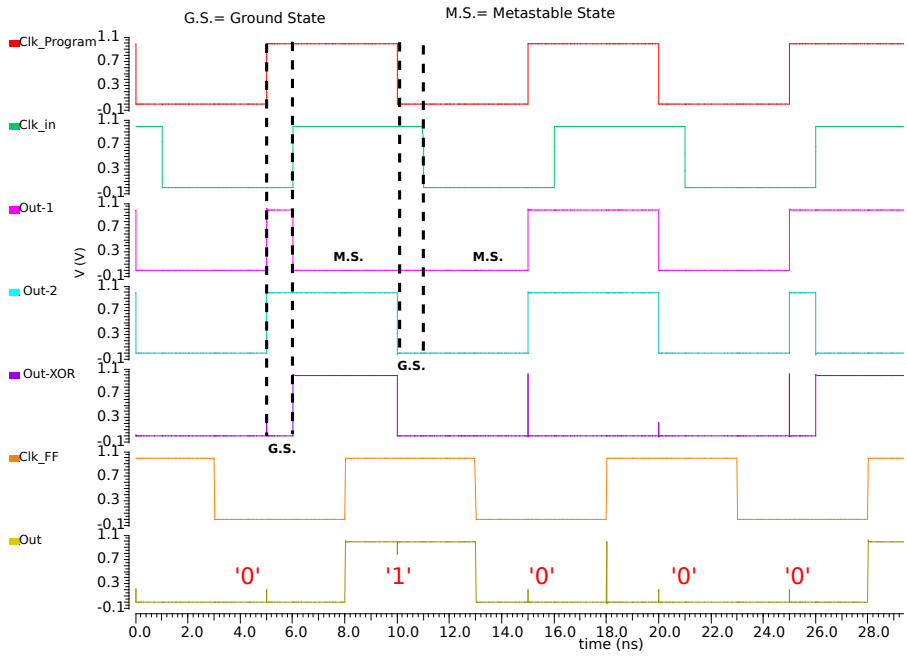

Fig. 10. Transient waveforms on operating the END-TRUE at 100 MHz clock frequency with the ground states and metastable states marked for a clock cycle. Smaller glitches in the *Out* signal appear since the same *clk_Program* is also fed to configure the proposed TSPC D-flip flop as negative or positive edge-triggered. Larger glitches appear during the switching transience of the D-flip flop during data sampling.

has been adapted to incorporate flicker and white noise parameters. Furthermore, according to the current-drive capability of vertically-stacked SiNW technology, it has been assumed that there are four nanowires per stack of TIGFET [14, 35, 75]. It is to be noted that the main focus in this work is to demonstrate how transistor-level reconfigurability can be used for random number generation at increased throughput and hence [14] has been used for the experimental simulations. Verilog-A models for other ambipolar devices with different performance parameters and model characteristics are orthogonal to this work and can be used as well.

5.2 Simulation and results

For the circuit shown in Fig. 8, the transient waveforms for the input and output signals are shown in Fig. 10. All the analyses have been done for a supply voltage of 1.0 V. Here, all the clock signals *viz.*, *clk_Program*, *clk_IN* and *clk_FF* operate at a frequency of 100 MHz. Also, *clk_IN* and *clk_FF* are time-delayed versions of *clk_Program*, delayed by $1ns$ and $3ns$ respectively. A transient analysis has been performed using the embedded transient noise feature in *Virtuoso*

Spectre Circuit Simulator and obtain random bits at the 'OUT' node after every 5 ns. The ground states (**G.S.**) and metastable states (**M.S.**) attained by the TRNG in a clock cycle have been marked in Fig. 10. It can be noted that the throughput is equal to 200 Mbps which is twice the input clock frequency of 100 MHz.

In the corresponding CMOS-based implementation (designed for double-throughput) of the END-TRUE, operating at supply voltage of 1.0 V, PTM 16 nm low power based CMOS model has been used for the simulation of the MOSFETs [70]. In this case, clk_Select and clk_IN signals operate at 100 MHz whereas, the clk_FF signal, that samples data entering into the positive-edge triggered D-flip flop, operates at 200 MHz (Fig. 8). Random bits having a throughput of 200 Mbps has been obtained at the 'OUT' node.

The above procedure is repeated for a higher clock frequency of 200 MHz and a lower frequency of 10 MHz in case of the END-TRUE and the output raw bit sequences are recorded.

5.3 Comparison with the equivalent CMOS-based TRNG

Table 2 and Table 3 present a comparison between the simulated END-TRUE and its CMOS counterpart (both for double-throughput). It can be seen that there is 60% saving in the number of transistors by employing an RFET based design. Furthermore, Table 3 shows a comparison between one SR latch unit for the END-TRUE and its CMOS counterpart on the basis of power consumption and delay operating at a clock frequency of 100 MHz. A 94.5% reduction in leakage power, 70.7% reduction in dynamic power and 77.3% reduction in critical path delay has been observed in case of the SR latch unit based on RFETs with respect to its CMOS equivalent.

Table 2. Comparison of the number of transistors, to realize END-TRUE and its CMOS equivalent.

No. of Transistors	**RFET model**	**CMOS model**
SR latch unit	14	46
2-input XOR	4	8
TSPC-based D-flip flop	8	11
TOTAL	**26**	**65**

5.4 Statistical evaluation of the generated bit sequence

By performing a transient analysis in the *Spectre* simulator, 110,000 bits has been generated as output from the END-TRUE for the clock frequencies of 10 MHz, 100 MHz and 200 MHz respectively. The statistical tests are performed on the random bits generated.

Table 3. Comparison of power consumption and delay of an RFET-based SR latch unit and its CMOS equivalent.

SR latch unit	Leakage power (nW)	Dynamic power (nW)	Delay (ps)
END-TRUE	16.85	79.65	206
CMOS equivalent	308.25	271.5	909

In order to carry out thorough statistical evaluation and owing to the complexity of the simulations due to large number of parameters, high precision and a bulk of simulated and stored data points, two types of analysis are carried out– Firstly, 110 sequences of 1000 bits each are formed from the overall 110,000 bits for each frequency of operation and are subjected to various statistical evaluations. This is required to evaluate the randomness in smaller chunks of the bit patterns. Secondly, statistical analysis is performed by consolidating all the 110 sequences, thereby forming a 110,000-bits long sequence each for the clock frequencies of 10 MHz, 100 MHz and 200 MHz. This is necessary to carry out evaluation for the complete sequence.

For the simulation model as proposed in Section 4.2, that is used to extract entropy from the END-TRUE, the output bit sequence can be assumed to be *i.i.d.* (independent and identically distributed). It is because before the occurrence of a metastability event, either at the rising or at the falling clock edge, the output node 'OUT' of the TRNG attains a ground state in which it resets itself before generating another random bit. Hence, the model does not involve correlation between two consecutive bits generated at 'OUT' due to the metastability event.

Shannon entropy as a measure for randomness Entropy is defined as the average amount of information produced by a stochastic source of data [66]. The amount of randomness in the outcome of an experiment can be measured using a metric called Shannon entropy. For an *i.i.d.* binary sequence that takes values from a finite set $X\{0,1\}$ with a probability distribution function $p : X \rightarrow [0,1]$, the Shannon entropy per bit (H) is given as:

$$H = -\Sigma_{x_i \epsilon X}\ p(x_i) \log_2 p(x_i) \tag{4}$$

For an uniformly distributed (unbiased) sequence of bits for which $p(0) = p(1) = 0.5$, the Shannon entropy per bit is equal to 1.0 which is the maximum value. If the output bit sequence from the TRNG is strongly biased, *i.e.* one bit appears more frequently than the other, then the Shannon entropy deviates significantly from its maximum value, indicating that the given bit sequence is less random and more deterministic.

Fig. 11 shows the variation in the Shannon entropy per bit for each 1000 bit-long output sequence when the END-TRUE operates at various clock frequencies. It can be clearly observed that the Shannon entropy for most of the

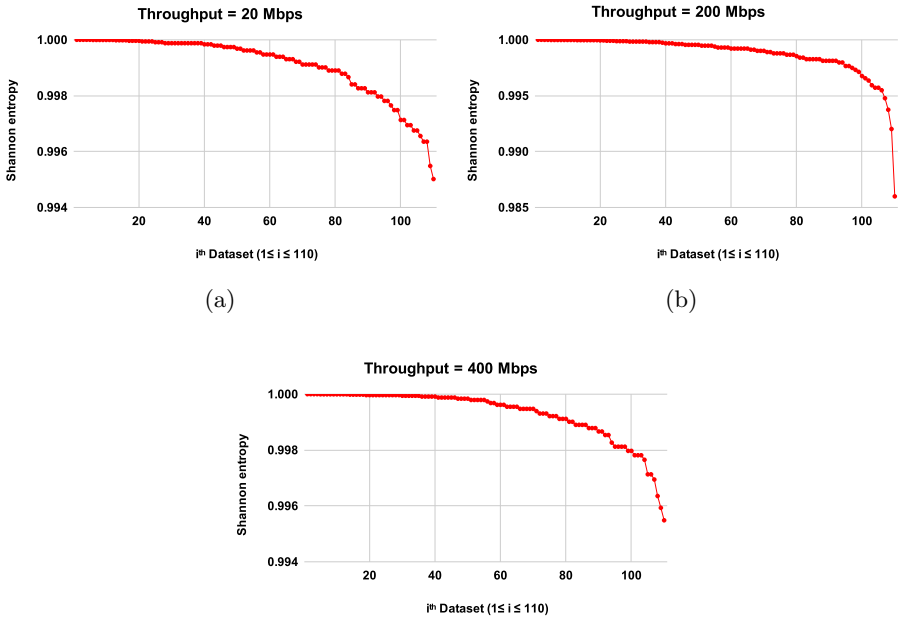

Fig. 11. Plots showing variation in Shannon entropy with the 110 datasets of 1000 bits each for frequencies- (a) 10 MHz, (b) 100 MHz, and (c) 200 MHz

sequences is very close to 1.0, indicating that the raw bit sequences are uniformly distributed. This, however, does not imply deeming the bit sequence as random. It is to be noted that uniform distribution of bits is a necessary but not a sufficient condition to assess randomness and hence evaluation over NIST benchmark suite has been carried out in this work.

NIST benchmark suite (SP800-22 rev. 1a) for statistical evaluation of randomness [50] The *National Institute of Standards and Technology* (NIST) test suite is used to evaluate the randomness of the binary sequences generated at the output of a TRNG. This benchmark suite is commonly used to evaluate both hardware and software-based RNGs and indicates whether the bitstream is likely to come from a uniform i.i.d. [50, 24]. The test suite consists of several benchmarks that are run on all the binary sequences and a value, *p-value* corresponding to each sequence per benchmark is generated. An RNG is said to pass a benchmark if the p-value is greater than a particular threshold, which is termed as a 'success' for that specific benchmark. Subsequently, for each benchmark in the suite, two metrics are defined namely, *success rate* and *P'-value*. The success rate for a benchmark is the proportion of the binary sequences passing the benchmark, while the P'-value quantifies the uniformity in the distribution of all

the p-values for a benchmark in the suite. The P'-value is a number between 0 and 1. An RNG is said to pass a benchmark if the success rate and the P'-value are greater than a threshold [50]. Only those benchmark are performed from the suite which can be run on the generated number of bits (110,000).[3]

Table 4. Results of the NIST benchmark suite for the END-TRUEusing 110 sequences of 1000 bits each. The threshold for P'-value is 0.0001 and for success rate is 105/110 = 0.954 [24, 41]. **(Failed benchmark results have been highlighted in red)**

	10 MHz		**100 MHz**		**200 MHz**	
benchmark name	**Success rate**	**P'-value**	**Success rate**	**P'-value**	**Success rate**	**P'-value**
Monobit Frequency	0.991	0.2238	0.964	0.0052	1.0	0.7757
Block frequency	0.982	0.0004	0.936	3.19E-12	0.918	1.08E-10
Runs	1.0	0.7399	1.0	0.6276	0.973	0.3807
Longest run	1.0	0.5159	1.0	0.5899	1.0	0.1431
DFT	0.991	7.99E-18	0.973	5.04E-14	0.991	2.25E-20
Overlap template matching	0.964	0.0004	0.945	0.0020	0.964	2.02E-07
Non-overlap template matching	0.991	0.0064	1.0	0.3218	1.0	0.6655
Cumulative sum - 1	0.991	0.1359	0.973	0.0002	1.0	0.5526
Cumulative sum - 2	1.0	0.2820	0.945	6.66E-05	1.0	0.5159
Serial - 1	0.991	0.7216	0.954	0.0011	0.973	0.0027
Serial - 2	0.982	0.3654	0.991	0.9230	0.973	0.1506
Approximate entropy	0.991	0.8421	0.964	0.0597	0.982	0.0083
Binary matrix rank	1.0	0.2949	0.991	0.0885	0.982	0.2346

6 Results and Discussion

Table 4 shows the the NIST benchmark results (success rates and P'-values) for the 110 raw bit sequences generated from the END-TRUE operating at clock frequencies of 10 MHz, 100 MHz and 200 MHz. Table 5 shows the the NIST benchmark results (p-values) and Shannon entropies for the 110,000 bits-long binary sequence each for the clock frequencies of 10 MHz, 100 MHz and 200 MHz.

It can be observed that when the TRNG operates at a lower frequency of 10 MHz, the success-rate for raw output binary sequence passes all the NIST benchmarks as shown in Table 4. At higher values of operating frequency or throughput, only a few benchmarks (in this case, *Block frequency*, *DFT*, *Overlap template matching* and *Cumulative sum - 2* benchmarks) fail from the perspective of success rate and/or P'-value. However, the observed success rates for the failed benchmarks are still more than 90% for all the binary sequences tested. Moreover, even without post-processing the raw output bit sequences, it can be

[3] This is because the remaining benchmarks in the suite (*Maurer's Universal statistical, Linear, Radom excursion* tests) require more than 10^7 bits for evaluation and it would amount to an unfeasible time duration to generate the bits using simulation [41] for a TCAD-based verilog-A model for RFETs [14]

Table 5. P-values for the NIST benchmarks with the binary sequences of 110,000 bits from the END-TRUE taken altogether (without post-processing). The threshold for P-value is 0.01 for a benchmark to pass [50]. **(Failed benchmark results have been highlighted in red)**

benchmark name	**10 MHz**	**100 MHz**	**200 MHz**
Monobit Frequency	0.59	0.21	0.51
Block frequency	0.01	7.32E-05	0.57
Runs	6.15E-08	1.33E-10	6.78E-07
Longest run	0.11	0.68	0.51
DFT	0.72	0.27	0.23
Overlap template matching	0.02	0.46	0.02
Non-overlap template matching	0.42	0.03	0.32
Cumulative sum - 1	0.39	0.25	0.35
Cumulative sum - 2	0.64	0.12	0.67
Serial - 1	6.44E-07	1.23E-17	1.76E-17
Serial - 2	0.08	0.01	0.02
Approximate entropy	9.19E-07	5.69E-17	5.10E-17
Binary matrix rank	0.68	0.35	0.06
Shannon Entropy	**0.9999980685**	**0.9999896832**	**0.9999972186**

Table 6. P-values for the NIST benchmarks after Von-Neumann post-processing of the raw binary sequence of 110,000 bits from the END-TRUE. The threshold for P-value is 0.01 for a benchmark to pass [50]. **(Failed benchmark results have been highlighted in red)**

Test name	**10 MHz**	**100 MHz**	**200 MHz**
Monobit Frequency	0.95	0.46	0.63
Block frequency	0.11	0.75	0.47
Runs	0.87	0.00025	0.04
Longest run	0.39	0.91	0.25
DFT	0.46	0.16	0.26
Overlap template matching	0.25	0.19	0.03
Non-overlap template matching	0.81	0.16	0.998
Cumulative sum - 1	0.44	0.20	0.74
Cumulative sum - 2	0.39	0.72	0.81
Serial - 1	0.59	0.02	0.22
Serial - 2	0.50	0.68	0.28
Approximate entropy	0.60	0.02	0.24
Binary matrix rank	0.31	0.15	0.87
Shannon Entropy	**0.9999999011**	**0.9999854835**	**0.9999937737**

observed that the *Monobit Frequency* benchmark is passed for both higher and lower frequencies of operation. It implies that the number of '0's and '1's produced by the TRNG are approximately equal as would be expected for a truly random sequence [50]. It is important to note here that the *Monobit Frequency* benchmark is compulsory to pass as other subsequent benchmarks in the NIST suite depend on it [50].

6.1 Post-processing unit of a TRNG

On fabrication, it is quite plausible that if our proposed metastability-based TRNG generates a raw binary sequence at higher speed (higher throughput), then the sequence has a statistical weakness resulting in a skewed (biased) distribution of '0's and '1's [12, 1]. Statistical weaknesses may also arise from PVT variations that hamper the source of entropy among other factors. Thus, typically every metastability-based TRNG has an integration of two units, *viz.* a physical source of entropy (in this case, the SR latch units generating the raw binary sequences) and a post-processing unit that transforms the raw binary sequences with statistical weaknesses into a sequence which is computationally tedious to differentiate from a purely random sequence [12, 61].

A very commonly used post-processing technique is the Von-Neumann extraction. This method acts on raw bit streams with statistical weaknesses and outputs a uniformly distributed and uncorrelated bit stream independent from the input raw stream however, at the cost of reduced throughput [38]. In this algorithm, the raw bit stream is grouped into non-overlapping pairs of consecutive bits. For each pair, in case both the bits are equal then the pair is discarded, otherwise, the first bit in the pair is taken to be the output. Thus, this algorithm essentially uses two input bits to produce either zero or one output bit. This algorithm is employed to post-process the raw binary sequences consisting of the entire set of 110,000 bits from the END-TRUE operating at frequencies of 10 MHz, 100 MHz and 200 MHz. Subsequently, all the NIST benchmarks are run on the processed output sequences and the corresponding p-values are recorded in Table 6.

On comparing the data shown in Table 5 and Table 6, a significant improvement in the NIST benchmark results after Von-Neumann processing can be observed, for all the three frequencies of operation. After Von-Neumann processing of the 110,000 bits-long sequence, almost all the NIST benchmarks have been shown to pass.

XOR-ing the outputs of multiple SR latch units to increase entropy: It has been mentioned in Section 5 that to have a feasible runtime for simulations, the two outputs of SR latch units has been XOR-ed to generate the output binary sequence of the END-TRUE. However, as discussed in section 3, the outputs of multiple SR latch units can be XOR-ed to further increase the entropy of the output sequence of the END-TRUE and make the device more robust against PVT variations [66]. For the CMOS-based ASIC implementation of the TRNG proposed in [61], it has been shown that XOR-ing the outputs of 256 SR latches can generate a random sequence with sufficient entropy that is able to pass all

the benchmarks in the NIST suite without post-processing. This methodology can also be employed for the END-TRUE to obtain a bitstream having sufficient randomness to pass all the benchmarks in the suite. Thus, the optimal number of SR latch units for the END-TRUE that must be XOR-ed to obtain a random sequence with entropy high enough to pass all the statistical tests without post-processing is to be determined when physically implemented in hardware.

6.2 Considerations on PVT Variations

TRNGs should preferably generate high-quality random numbers even in the case of an adverse surrounding environment. Unfortunately, TRNGs based on CMOS and other charge-based technologies are sensitive to the variations of process conditions, supply voltage, and temperature (PVT). An attackers can try to evade the secure device by intentionally providing such PVT variations externally. For example, they can reduce the supply voltage or can put the embedded device in a freezing environment and thus deteriorating the quality of random numbers hence degrading the security of the device [61]. An RFET being CMOS compatible charge-based device is not spared from these variations and hence can compromise the overall functioning of the TRNG if not optimized properly. They may also cause an asymmetry in the electrical characteristics, thus compromising the overall functioning of the reconfigurable circuit. Thus, PVT variations may cause biases in the output of a TRNG. Therefore, to develop secure hardware, one should ideally run simulations corresponding to all the mentioned variations on TRNG and verify the quality of randomness by using NIST benchmarks.

Unfortunately some the limitations posed by the version of the Verilog-A RFET table model used in this work does not allow us to provide quantitative data for our design. To study the effect of voltage variations in the TRNG output, one has to simulate the circuit with the supply voltage variation of V_{DD} $\pm$ 10% and collect up to 110,000 data points, and subsequently perform NIST SP 800-22 tests on those data points. The current version of the model used in this work is too slow and takes considerable resources and time to collect this high amount data points. Further, no temperature dependencies or process variation parameters are integrated into the model. Thus in this work, a qualitative discussion of the PVT impact on our TRNG application based on some results from literature has been done. The conclusions given in this section have to be verified by extensive simulations once better models are available.

Process Variations: The impact of process variation is an important effect that should be considered while designing a robust circuit. They occur because of the manufacturing conditions like temperature, concentration levels, etc. These conditions although extremely well controlled in modern CMOS processes, still have some unavoidable variations, which manifests as slight variations of device parameters. It is an unavoidable variation and bound to exist. The main contributors to the process variations are: Line Edge Roughness (LER) [22, 67], Gate

Edge Roughness (GER) [76], Work Function Variation (WFV) [26, 28], and Random Dopant Fluctuations (RDF) [28]. Threshold voltage and different parasitic capacitance's are two of most the significant parameters which get affected in classical CMOS [52]. Beeing a dopant-free technology, RFETs are expected to show better performance in terms of RDF. Still, to develop a reliable circuit using RFETs, process variation estimation is crucial. It is imperative to calculate reliable process corner information and timing variations [27]. Combining GER, LER, and WFV RFETs have been found to be more vulnerable to process fluctuations overall than CMOS devices [27]. However, this effect can be mainly attributed to the high impact of WFV, which is the main contributor in terms of process variations for RFET technology reasoned in the metallic source and drain electrodes. Thus, to yield a good TRNG, special attention must be given to a highly controlled work function, while mass manufacturing the RFETs [27]. Considering that most silicon nanowire based RFETs are based on sharp $NiSi_2$/Si junctions with a quasi-epitaxial relation between both materials, it is reasonable to assume that this metric can be achieved [69].

Voltage Variations: Supply voltage variations are also a crucial parameter that is needed to be taken care of. It has been stated that the circuit must at least support a variation of $V_{DD} \pm 10\%$ to be known as a reliable device [52]. In terms of RFET designs, it is mainly crucial, that the symmetric behavior between p- and n-type operation is not lost by the voltage variations. Such an attack scenario was first tested in [10] on various NAND/NOR logic gate design variants. In this work, it was described that increasing V_{DD} lead to no particular differences in the propagation delay values of both configurations. If the nominal operation voltage has been instead decreased the relative difference between individual NAND and NOR rise and fall times increased, but stayed always below 10% difference. A V_{DD} reduction of more than 33% of the initial supply voltage resulted in malefaction of the circuit.

Temperature Variations: The final parameter to be considered for RFET based security solutions are temperature variations. It is well known that in CMOS technology, temperature variation impacts the I-V characteristics of a transistor. It affects the thermal voltage V_T in subthreshold conduction, transistor threshold voltage V_t as well as the μ, due to higher number of scattering events at higher temperatures [52]. Thus, typically the off-state currents of CMOS devices is increased, while the on-state is decreased, leading to lower on/off ratio at higher temperatures. This is different for RFET devices, which rely on the thermionic field-emission based injection of carries over the Schottky junctions at source and drain. With increase in temperature, more carriers are injected, overshadowing the effect of the lower effective channel mobility. As a result, both, on- as well as off-current increase with increasing temperature. It is conceivable, that this behavior makes RFET based circuit solutions more stable with respect to temperature variations than their CMOS based counterpart.

7 Conclusions

In the present work a metastability-based TRNG design has been proposed using emerging reconfigurable nanotechnology. This is referred as *Emerging Nanotechnology -based Double -Throughput True Random Number Generator* (END-TRUE). The transistor-level ambipolarity in RFETs allows us to duplicate cross-coupled SR latches and hence random bits can be sampled at both the edges of the clock. The END-TRUE generates a random bit at each half cycle of the input clock, thereby a throughput of twice the input clock frequency is obtained. This enables the dual edge-triggered D-flip flop operate at the same clock frequency as the input clock signal to the TRNG. Using runtime reconfigurability, the TRNG is shown to use less hardware, be compact in terms of transistor count per block (60% saving in the transistor count), consume less power (94.5% saving in leakage power and 70.7% saving in dynamic power) and has a lower critical path delay (77.3% reduction in delay) with respect to its equivalent CMOS counterpart. Statistical evaluations show that the generated bitstream using our proposed END-TRUE has high values of Shannon entropy as well as successfully passes the NIST benchmark suite (except one) upon post-processing. The technique of post-processing is used regardless to mitigate the effects of process variation [1].

The present work demonstrates a viable circuit implementation for emerging reconfigurable nanotechnology which is a key component in hardware security. Silicon or germanium nanowire-based RFETs follow similar CMOS-like top-down fabrication process [55, 30] and come in stacked nanowire geometry [72] and hence are commercially feasible and can complement CMOS technology. While in the present work, a specific application has been demonstrated, it is expected that with better device models, better evaluation can be carried out.

References

[1] Swarup Bhunia and Mark Tehranipoor. “Chapter 12 - Hardware Security Primitives”. In: *Hardware Security*. Ed. by Swarup Bhunia and Mark Tehranipoor. Morgan Kaufmann, 2019, pp. 311 –345.

[2] Y. Bi et al. “Enhancing hardware security with emerging transistor technologies”. In: *2016 International Great Lakes Symposium on VLSI (GLSVLSI)*. 2016, pp. 305–310.

[3] Y. Bi et al. “Enhancing hardware security with emerging transistor technologies”. In: *International Great Lakes Symposium on VLSI (GLSVLSI)*. 2016.

[4] Y. Bi et al. “Leveraging Emerging Technology for Hardware Security - Case Study on Silicon Nanowire FETs and Graphene SymFETs”. In: *2014 IEEE 23rd Asian Test Symposium*. 2014, pp. 342–347.

[5] M. Bucci et al. “A high-speed oscillator-based truly random number source for cryptographic applications on a smart card IC”. In: *IEEE Transactions on Computers* 52.4 (2003), pp. 403–409.

[6] A. Chen et al. “Using emerging technologies for hardware security beyond PUFs”. In: *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*. 2016, pp. 1544–1549.
[7] M. De Marchi et al. “Polarity control in double-gate, gate-all-around vertically stacked silicon nanowire FETs”. In: *2012 International Electron Devices Meeting*. 2012, pp. 8.4.1–8.4.4.
[8] *Exclusive OR (XOR) and hardware random number generators.*
[9] N. Fujieda, M. Takeda, and S. Ichikawa. “An Analysis of DCM-based True Random Number Generator”. In: *IEEE Transactions on Circuits and Systems II: Express Briefs* (2019), pp. 1–1.
[10] Giulio Galderisi, Thomas Mikolajick, and Jens Trommer. “Reconfigurable Field Effect Transistors Design Solutions for Delay-Invariant Logic Gates”. In: *IEEE Embedded Systems Letters* (2022).
[11] Blaise Gassend et al. “Silicon physical random functions”. In: Jan. 2002, pp. 148–160.
[12] J. D. J. Golic. “New Methods for Digital Generation and Postprocessing of Random Data”. In: *IEEE Transactions on Computers* 55.10 (2006), pp. 1217–1229.
[13] L. Gong et al. “True Random Number Generators Using Electrical Noise”. In: *IEEE Access* 7 (2019), pp. 125796–125805.
[14] G. Gore et al. “A Predictive Process Design Kit for Three-Independent-Gate Field-Effect Transistors”. In: *2019 IFIP/IEEE 27th International Conference on Very Large Scale Integration (VLSI-SoC)*. 2019, pp. 172–177.
[15] Patrick Haddad et al. “A Physical Approach for Stochastic Modeling of TERO-Based TRNG”. In: *Cryptographic Hardware and Embedded Systems – CHES 2015*. Ed. by Tim Güneysu and Helena Handschuh. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 357–372.
[16] Naoki Harada et al. “A polarity-controllable graphene inverter”. In: 2010.
[17] Hisashi Hata and Shuichi Ichikawa. “FPGA Implementation of Metastability-Based True Random Number Generator”. In: *IEICE Transactions on Information and Systems* 95.2 (Jan. 2012), pp. 426–436.
[18] André Heinzig et al. “Reconfigurable Silicon Nanowire Transistors”. In: *Nano letters* 12 (Nov. 2011), pp. 119–24.
[19] D. E. Holcomb, W. P. Burleson, and K. Fu. “Power-Up SRAM State as an Identifying Fingerprint and Source of True Random Numbers”. In: *IEEE Transactions on Computers* 58.9 (2009), pp. 1198–1210.
[20] J. Holleman et al. “A 3 μW CMOS True Random Number Generator With Adaptive Floating-Gate Offset Cancellation”. In: *IEEE Journal of Solid-State Circuits* 43.5 (2008), pp. 1324–1336.
[21] W. T. Holman, J. A. Connelly, and A. B. Dowlatabadi. “An integrated analog/digital random noise source”. In: *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 44.6 (1997), pp. 521–528.

[22] Xiaobo Jiang et al. “Investigations on Line-Edge Roughness (LER) and Line-Width Roughness (LWR) in Nanoscale CMOS Technology: Part I–Modeling and Simulation Method”. In: *IEEE Transactions on Electron Devices* 60.11 (2013), pp. 3669–3675.

[23] Jiren Yuan and C. Svensson. “New single-clock CMOS latches and flipflops with improved speed and power savings”. In: *IEEE Journal of Solid-State Circuits* 32.1 (1997), pp. 62–69.

[24] Song-Ju Kim, Ken Umeno, and Akio Hasegawa. “Corrections of the NIST Statistical Test Suite for Randomness”. In: 103 (Feb. 2004).

[25] D. J. Kinniment and E. G. Chester. “Design of an on-chip random number generator using metastability”. In: *Proceedings of the 28th European Solid-State Circuits Conference*. 2002, pp. 595–598.

[26] Kyul Ko et al. “Compact Model Strategy of Metal-Gate Work-Function Variation for Ultrascaled FinFET and Vertical GAA FETs”. In: *IEEE Transactions on Electron Devices* 66.3 (2019), pp. 1613–1616.

[27] Xianglong Li et al. “Impact of Process Fluctuations on Reconfigurable Silicon Nanowire Transistor”. In: *IEEE Transactions on Electron Devices* 68.2 (2021), pp. 885–891.

[28] Yiming Li et al. “Process variation effect, metal-gate work-function fluctuation and random dopant fluctuation of 10-nm gate-all-around silicon nanowire MOSFET devices”. In: *2015 IEEE International Electron Devices Meeting (IEDM)*. 2015, pp. 34.4.1–34.4.4.

[29] N. Liu et al. “A true random number generator using time-dependent dielectric breakdown”. In: *2011 Symposium on VLSI Circuits - Digest of Technical Papers*. 2011, pp. 216–217.

[30] M. De Marchi et al. “Top-down fabrication of gate-all-around vertically stacked silicon nanowire fets with controllable polarity”. In: *IEEE Transactions on Nanotechnology* 13.6 (2014), pp. 1029–1038.

[31] A. Theodore Markettos and Simon W. Moore. “The Frequency Injection Attack on Ring-Oscillator-Based True Random Number Generators”. In: *Cryptographic Hardware and Embedded Systems - CHES 2009*. Ed. by Christophe Clavier and Kris Gaj. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 317–331.

[32] S. K. Mathew et al. “2.4 Gbps, 7 mW All-Digital PVT-Variation Tolerant True Random Number Generator for 45 nm CMOS High-Performance Microprocessors”. In: *IEEE Journal of Solid-State Circuits* 47.11 (2012), pp. 2807–2821.

[33] Stathis Mavrovouniotis and Mick Ganley. “Hardware security modules”. In: *Secure Smart Embedded Devices, Platforms and Applications*. Springer, 2014, pp. 383–405.

[34] Alfred J Menezes et al. *Handbook of applied cryptography*. CRC press, 1996.

[35] Thomas Mikolajick et al. “The RFET - A reconfigurable nanowire transistor and its application to novel electronic circuits and systems”. In: *Semiconductor Science and Technology* 32 (Dec. 2016).

[36] Halid Mulaosmanovic, Thomas Mikolajick, and Stefan Slesazeck. "Random Number Generation Based on Ferroelectric Switching". In: *IEEE Electron Device Letters* 39.1 (2018), pp. 135–138.

[37] Shu Nakaharai et al. "Electrostatically Reversible Polarity of Ambipolar - MoTe2 Transistors". In: *ACS Nano* 9.6 (2015). PMID: 25988597, pp. 5976–5983. eprint: `https://doi.org/10.1021/acsnano.5b00736`.

[38] John von Neumann. "Various Techniques Used in Connection with Random Digits". In: *Monte Carlo Method*. Ed. by A. S. Householder, G. E. Forsythe, and H. H. Germond. Vol. 12. National Bureau of Standards Applied Mathematics Series. Washington, DC: US Government Printing Office, 1951. Chap. 13, pp. 36–38.

[39] Ravikanth Pappu et al. "Physical One-Way Functions". In: *Science* 297.5589 (2002), pp. 2026–2030. eprint: `https://science.sciencemag.org/content/297/5589/2026.full.pdf`.

[40] Rachael J. Parker. "Entropy justification for metastability based nondeterministic random bit generator". In: *2017 IEEE 2nd International Verification and Security Workshop (IVSW)*. 2017, pp. 25–30.

[41] B. Perach and s. kvatinsky. "An Asynchronous and Low-Power True Random Number Generator Using STT-MTJ". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 27.11 (2019), pp. 2473–2484.

[42] F. Rahman et al. "Security Beyond CMOS: Fundamentals, Applications, and Roadmap". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 25.12 (2017), pp. 3420–3433.

[43] S. Rai, M. Raitza, and A. Kumar. "Technology mapping flow for emerging reconfigurable silicon nanowire transistors". In: *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*. 2018, pp. 767–772.

[44] S. Rai et al. "A physical synthesis flow for early technology evaluation of silicon nanowire based reconfigurable FETs". In: *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*. 2018, pp. 605–608.

[45] S. Rai et al. "Designing Efficient Circuits Based on Runtime-Reconfigurable Field-Effect Transistors". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 27.3 (2019), pp. 560–572.

[46] S. Rai et al. "Hardware Watermarking Using Polymorphic Inverter Designs Based On Reconfigurable Nanotechnologies". In: *ISVLSI*. 2019.

[47] Shubham Rai et al. "Security Promises and Vulnerabilities in Emerging Reconfigurable Nanotechnology-Based Circuits". In: *IEEE Transactions on Emerging Topics in Computing* (2020), pp. 1–1.

[48] Michael Raitza et al. "RAW 2014: Random Number Generators on FPGAs". In: *ACM Trans. Reconfigurable Technol. Syst.* 9.2 (Dec. 2015).

[49] J. Rajendran et al. In: *Proceedings of the IEEE* 103.5 (2015), pp. 829–849.

[50] Andrew Rukhin et al. "NIST Special Publication 800-22: A Statistical Test Suite for the Validation of Random Number Generators and Pseudo Random Number Generators for Cryptographic Applications". In: *NIST Special Publication 800-22* (Apr. 2010).

[51] A. Rupani, S. Rai, and A. Kumar. "Exploiting Emerging Reconfigurable Technologies for Secure Devices". In: *Euromicro DSD*. 2019.

[52] Adel S. Sedra and Kenneth C. Smith. *Microelectronic Circuits*. fifth. Oxford University Press, 2004.

[53] V Sessi et al. "Back-Bias Reconfigurable Field Effect Transistor: A Flexible Add-On Functionality for 22 nm FDSOI". In: *2021 Silicon Nanoelectronics Workshop (SNW)*. IEEE. 2021, pp. 1–2.

[54] M. Simon et al. "A wired-AND transistor: Polarity controllable FET with multiple inputs". In: *2018 76th Device Research Conference (DRC)*. 2018, pp. 1–2.

[55] M. Simon et al. "Bringing reconfigurable nanowire FETs to a logic circuits compatible process platform". In: *2016 IEEE Nanotechnology Materials and Devices Conference (NMDC)*. 2016, pp. 1–3.

[56] Masiar Sistani et al. "Nanometer-scale Ge-based adaptable transistors providing programmable negative differential resistance enabling multivalued logic". In: *ACS nano* 15.11 (2021), pp. 18135–18141.

[57] B. Sunar, W. J. Martin, and D. R. Stinson. "A Provably Secure True Random Number Generator with Built-In Tolerance to Active Attacks". In: *IEEE Transactions on Computers* 56.1 (2007), pp. 109–119.

[58] S. Tanachutiwat et al. "Reconfigurable multi-function logic based on graphene p-n junctions". In: *Design Automation Conference*. 2010, pp. 883–888.

[59] X. Tang et al. "TSPC Flip-Flop circuit design with three-independent-gate silicon nanowire FETs". In: *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*. 2014, pp. 1660–1663.

[60] C. Tokunaga, D. Blaauw, and T. Mudge. "True Random Number Generator with a Metastability-Based Quality Control". In: *2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*. 2007, pp. 404–611.

[61] Naoya Torii et al. "ASIC implementation of random number generators using SR latches and its evaluation". In: *EURASIP Journal on Information Security* 2016.1 (2016), p. 10.

[62] J. Trommer et al. "Reconfigurable nanowire transistors with multiple independent gates for efficient and programmable combinational circuits". In: *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*. 2016, pp. 169–174.

[63] Jens Trommer et al. "Enabling energy efficiency and polarity control in germanium nanowire transistors by individually gated nanojunctions". In: *ACS nano* 11.2 (2017), pp. 1704–1711.

[64] Michal Varchola and Milos Drutarovsky. "New High Entropy Element for FPGA Based True Random Number Generators". In: *Cryptographic Hardware and Embedded Systems, CHES 2010*. Ed. by Stefan Mangard and François-Xavier Standaert. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 351–365.

[65] Ihor Vasyltsov et al. "Fast Digital TRNG Based on Metastable Ring Oscillator". In: Aug. 2008, pp. 164–180.

[66] E. I. Vatajelu and G. Di Natale. "High-Entropy STT-MTJ-Based TRNG". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 27.2 (2019), pp. 491–495.

[67] Runsheng Wang et al. "Investigations on Line-Edge Roughness (LER) and Line-Width Roughness (LWR) in Nanoscale CMOS Technology: Part II–Experimental Results and Impacts on Device Variability". In: *IEEE Transactions on Electron Devices* 60.11 (2013), pp. 3676–3682.

[68] Y. Wang et al. "A novel circuit design of true random number generator using magnetic tunnel junction". In: *2016 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*. 2016, pp. 123–128.

[69] Walter M Weber et al. "Silicon to nickel-silicide axial nanowire heterostructures for high performance electronics". In: *physica status solidi (b)* 244.11 (2007), pp. 4170–4175.

[70] Wei Zhao and Yu Cao. "New generation of predictive technology model for sub-45nm design exploration". In: *7th International Symposium on Quality Electronic Design (ISQED'06)*. 2006, 6 pp.–590.

[71] K. Wold and C. H. Tan. "Analysis and Enhancement of Random Number Generator in FPGA Based on Oscillator Rings". In: *2008 International Conference on Reconfigurable Computing and FPGAs*. 2008, pp. 385–390.

[72] Peide Ye, Thomas Ernst, and Mukesh V Khare. "The last silicon transistor: Nanosheet devices could be the final evolutionary step for Moore's Law". In: *IEEE Spectrum* 56.8 (2019), pp. 30–35.

[73] Yu-Ming Lin et al. "High-performance carbon nanotube field-effect transistor with tunable polarities". In: *IEEE Transactions on Nanotechnology* 4.5 (2005), pp. 481–489.

[74] J. Yuan and C. Svensson. "High-speed CMOS circuit technique". In: *IEEE Journal of Solid-State Circuits* 24.1 (1989), pp. 62–70.

[75] J. Zhang et al. "Configurable Circuits Featuring Dual-Threshold-Voltage Design With Three-Independent-Gate Silicon Nanowire FETs". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 61.10 (2014), pp. 2851–2861.

[76] Zhe Zhang et al. "Extraction of Process Variation Parameters in FinFET Technology Based on Compact Modeling and Characterization". In: *IEEE Transactions on Electron Devices* 65.3 (2018), pp. 847–854.