Design and Evaluation of a PVT Variation-Resistant TRNG Circuit

Bikash Poudel* and Arslan Munir†
Department of Computer Science
Kansas State University, Manhattan, KS, USA
Email: *bpoudel@ksu.edu, †amunir@ksu.edu

***Abstract*—On-chip true random number generators (TRNGs) can be designed by using traditional CMOS technology or by using more recent nanoscale devices and technologies. In CMOS technology, TRNGs are designed by harnessing random physical variations (e.g., thermal/supply/telegraph noise, jitter, latch metastability, etc.). Since CMOS technologies are slowly saturating in development, more recently, nanoscale devices and technologies, such as memristor, magnetic tunnel junction, carbon nanotubes, graphene, etc., are being used to design TRNG circuits. An ideal TRNG circuit is expected to generate a sequence of random bits with very high bit-entropy and zero correlation among the generated bitstreams. However, increasing variations in the fabrication process and the sensitivity of transistors to operating conditions (e.g., voltage, and temperature (PVT)) have a significant impact on bit-entropy of TRNGs designed in deep nanometer technologies. Furthermore, PVT variations can be exploited by an adversary as effective tools to attack TRNGs. To mitigate these issues, we propose three probabilistic circuits: probability booster, probability dropper, and probability stabilizer. We use these circuits to build our proposed TRNG circuit. We also propose a stochastic model of our proposed TRNG circuit. To validate our proposed model, we have simulated our TRNG circuit using PSpice simulator using $65nm$ and $28nm$ processes. Results reveal that our proposed TRNG can generate random numbers with bit-entropy that always lies in the range $[0.998, 1]$ at a data rate of 16 Mbps and is robust against PVT variations. Using the NIST SP 800-22 test suite for randomness, we demonstrate that the output of the proposed TRNG circuit is statistically random with 99% confidence levels.**

***Index Terms*—TRNG, thermal noise, PVT variations, probabilistic switches, PCMOS, stochastic switching circuits**

I. Introduction and Motivation

True random numbers are needed in many areas such as Monte-Carlo computation, randomized algorithm, cryptography, and hardware security. Some of the existing TRNGs include ring oscillator (RO) based TRNG which uses jitter in output signal of RO; metastability-based TRNG which leverages metastable circuit elements like cross-couple inverters or SRAM cells; phase-locked loop (PLL) based TRNG that extracts intrinsic randomness from low-jitter clock signals synthesized by on-chip analog PLL circuits; embedded analog-to-digital converter based TRNG designed by exploiting the non-linear signal processing and chaos; and thermal noise-based TRNGs [1]. One common disadvantage of all of these TRNG circuits is that they suffer from high correlation between the generated bits thus requiring a complicated post-processing of the generated bits to provide necessary bit-entropy. Moreover, these methods are rendered inefficient due to their sensitivity to deterministic operational variations (e.g., voltage and temperature (PVT)) that deteriorates the quality of the output random bitstreams [1]–[3]. Also, as these TRNGs are designed in the form of Giga- and Tera-scale integrated circuits (ICs) using deep nanometer technologies, there are multiple factors that affect the delay, power, and reliability of TRNGs [3]. Process variation is one factor that affects delay and power of a transistor. This variation generates deviation in the randomness of the same TRNG circuit implemented in multiple ICs which are manufactured using same process technology by the same foundry. Apart from the fabrication process variation, some factors (e.g., short channel effects and aging) that vary dynamically during functioning of the chip also affect the behavior of transistors and thus, the behavior of TRNGs. The cumulative effect of these indelible anomalies is deviation of probability associated with each bit generated by a TRNG from the ideal value of 0.5, thus, lowering the bit-entropy below 1.

Various methods exist in the literature to make the TRNG circuits resistant to these variations. In [4], the authors proposed a post-Si tuning technique that uses additional transistors to compensate for mismatch of the devices. In [5], the authors present a circuit calibration method that compensates the device mismatch due to process variation. In [3], the authors suggested sub-V_{dd} precharge and hybrid self-calibrating techniques for metastability based TRNGs. The principle advantage of these methodologies is that these approaches do not increase the circuit size of the TRNG. However, these techniques are too specific to a particular type of variation and applies to a specific TRNG circuit only and hence, are not scalable to other TRNG circuits. This situation is further exacerbated by the fact that every year new TRNGs are introduced in the research community that work on completely different principles and have entirely different circuits. Therefore, a generic technique to make these TRNGs resistant to PVT variations is imperative. There exist some generic solutions for PVT variations called algorithmic post-processing techniques. These techniques use either XOR corrector or Von Neumann corrector [3]. The potential disadvantage of XOR corrector is that as the device mismatch increases, there is minimal improvement in the bit-entropy obtained through the XOR corrector. The Von Neumann corrector substantially decreases the data rate of the TRNG.

All of the above mentioned TRNGs are designed and implemented using preexisting CMOS technology which is slowly saturating in development. Moreover, new attack models and vulnerabilities are continually emerging and cannot be adequately addressed by current CMOS technology. Furthermore, these noise- and metastability-based TRNG circuits perform poorly under PVT variations. To resolve these issues, a new class of TRNGs has emerged which is based on more recent nanoscale devices and technologies such as memristor, magnetic-tunnel junction (MTJ) (Spin-Dice), carbon nanotubes, graphene, and phase-change memory. Compared to conventional TRNGs that require amplifiers or comparators with high complexity, the use of memristors (or MTJ) to design TRNG significantly reduces the design cost [6]. Nevertheless, this new class of TRNG has its own demerits. The memristor-based TRNG shows promise but its entropy is expected to deteriorate under temperature fluctuations [7]. The Spin-Dice and the complementary polarizer MTJ-based TRNG use the spin-torque effect to excite damping switching in a perpendicular magnetic anisotropy nanomagnet, thereby generating a random bit-stream. However, the 50% switching probability of nanomagnets for the damping mechanism occurs over a very narrow range of input spin current. As such, the TRNG implementations will have diminished robustness over process and temperature variations [7]. New architectures of TRNGs based on memristor and MTJ have been introduced to alleviate the effect of PVT variations [7]. To sum up, although these emerging nanoscale devices and technologies provide improvements in speed and performance over the conventional CMOS technologies, other equally important security issues, such as anti-tamper, counterfeit detection/avoidance, side-channel attacks, and reverse engineering, have not been fully assessed in the context of these emerging nanoscale devices used to design TRNG circuits [6].

To overcome these challenges posed by PVT variations and to make the TRNG resistant to noninvasive fault attacks based on various parameter tweaking, we propose a generic circuit-level solution for robust TRNG design. The proposed technique is resistant to PVT variations and provides high bit-entropy. This generic architecture can be used with any entropy source to build a TRNG circuit that is resistant to PVT variations. In this work, we have used a probabilistic switch (pswitch) as a source of entropy to build a robust TRNG using our proposed approach as a ***case study***. The reason for using pswitch as a case study is that we can control the probability of correctness (refer to Section II) of the pswitch by controlling the amount of thermal noise being injected into the CMOS inverter. This provides us the opportunity to analyze the worst-case behavior of our proposed TRNG. We provide an analytical model of the proposed approach and validate our model by circuit simulation using PSpice circuit simulator and Cadence Virtuoso Layout tool. Simulation results indicate that the TRNG circuits designed using our proposed approach are resistant to PVT variations and provides high bit-entropy ($\in [0.998, 1]$). Hence, these TRNG circuits can be used in applications that demand high reliability such as automotive, military, aerospace, unmanned aerial vehicle (UAV), etc. However, the TRNG circuits designed using our proposed approach result in area and power overhead, and may not be suitable for lightweight applications. The major contributions of this work are summarized below:

- We propose a generic approach to build a reliable TRNG using unreliable entropy sources (e.g., probabilistic switches, memristors, MTJs). Our approach is based on three proposed probabilistic circuits: probability booster, probability dropper, and probability stabilizer.
- We present an analytical stochastic model of the proposed probability stabilizer circuit.
- We have performed functional verification of the proposed TRNG circuit using PSpice circuit simulator for $65nm$ and $28nm$ processes. Furthermore, we have analyzed the robustness of our proposed circuit against the PVT variations. Results indicate that our proposed TRNG is robust against PVT variations and is also resistant to noninvasive fault attacks that involve changing operating temperature or supply voltage.

II. Probabilistic Switch

We start with introducing the basic concepts of a probabilistic switch (pswitch) because we have used pswitch as the source of entropy in our proposed TRNG design approach.

A. Basic Concept

For completeness, we introduce the concept of probabilistic switching and probabilistic complementary-metal-oxide semiconductor (PCMOS) inverter realization of a pswitch (refer [8]–[11] for more details). A CMOS inverter is a digital gate that realizes a compliment function. The behavior of a CMOS inverter is deterministic because the *probability* that the CMOS inverter produces HIGH (or LOW) output given LOW (or HIGH) input is 1. We will call this *probability* as the *probability of correctness* and we will use this term throughout the paper to refer to the probability that a pswitch behaves correctly as a CMOS inverter. A pswitch is realized using a CMOS inverter circuit coupled with thermal noise as shown in Fig. 1. This CMOS inverter coupled with thermal noise is PCMOS inverter [8]. For a probabilistic inverter (or pswitch), the output-to-input relationship is stochastic in nature. From now on, we will use the term pswitch to represent the probabilistic inverter. In our pswitch design, we will only consider the thermal noise coupled to the input of the CMOS inverter. We will use power supply noise as a source of variation of power supply voltage to study the effect of supply voltage fluctuation on the probabilistic behavior of pswitch and that of our proposed TRNG.

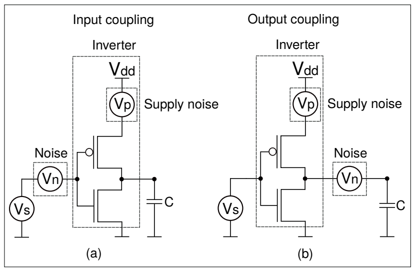


Fig. 1: *Constructing a pswitch by coupling thermal noise to a deterministic CMOS inverter: (a) Noise is coupled to the input of inverter, (b) Noise is coupled to the output of inverter. (adapted from [8])*

Mathematically, pswitch can be represented as the sum of deterministic switch and entropy source (e.g., thermal noise). The pswitch acts as a one-input Boolean function $\mathcal{F}$ that

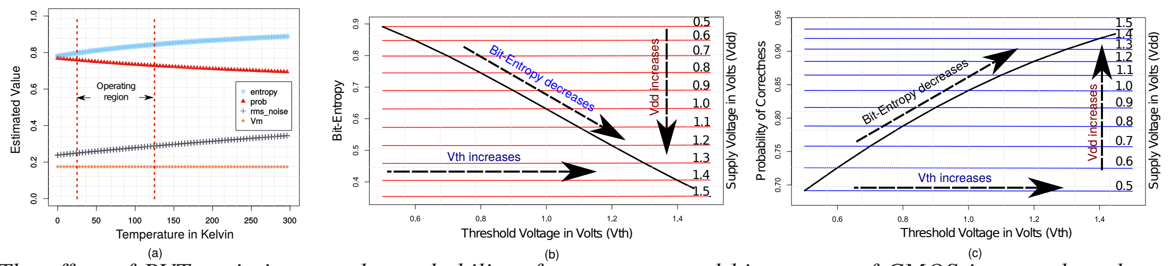


Fig. 2: *The effect of PVT variations on the probability of correctness and bit-entropy of CMOS inverter based pswitch: (a) Effect of temperature variation, (b) Effect of V_{th} and V_{dd} variation on bit-entropy, (c) Effect of V_{th} and V_{dd} variation on probability of correctness.*

maps input in $\{0, 1\}$ space to the random output in $\{0, 1\}$ space viz., $\mathcal{F} : \{0, 1\} \rightarrow \{0, 1\}$. If we denote $x_{bin}(t_0)$ as the one-bit binary input at time t_0 and $y_{bin}(t_1)$ as output of a pswitch at some later time t_1, the input-output relation can be summarized as: $y_{bin}(t_1) = \mathcal{F}(x_{bin}(t_0))$, with probability $p \in (0.5, 1)$ and $y_{bin}(t_1) = \mathcal{F}'(x_{bin}(t_0))$, with probability $1 - p$, where, p is the probability of correctness of pswitch. Since, pswitch is a probabilistic inverter, a correct functioning pswitch generates an output that is compliment of input. So, the probability of correctness can be defined as conditional probability that pswitch generates correct output for a given input. So, if V_{out} is output voltage of pswitch and V_{in} is input voltage of pswitch, then p can be written as Prob(V_{out} = HIGH | V_{in} = LOW) or Prob(V_{out} = LOW | V_{in} = HIGH). For a pswitch to act as a random number generator, the value of p must be in close proximity to the ideal value of 0.5.

A PCMOS inverter transforms a continuous-time input voltage into a continuous-time output voltage. However, a pswitch is required to produce digital output values. This can be accomplished by digitizing the continuous-time output signal of the PCMOS inverter at some sampling frequency f_s. The continuous-time output signal V_{out} is digitized using the following relation: the digital output $Y(t) = 1$, if $V_{out} \geq V_m$ and $Y(t) = 0$, otherwise. Here, V_m is the midpoint voltage of the CMOS inverter whose value is affected by PVT variations [12].

B. Analytical modeling of the pswitch

We build pswitch by coupling thermal noise to the input of the conventional CMOS inverter. To model the pswitch, we need to have a model to characterize the thermal noise. We follow the approach of [8], [13] to model thermal noise, where, noise is considered to be a random process characterized by a Gaussian distribution with zero mean and a standard deviation of σ_{th} where, σ_{th} is the root mean square (RMS) value of the noise. Furthermore, we consider thermal noise to be a white noise which has a constant power spectral density in the frequency of interest.

In modeling the probability of correctness of an inverter coupled with thermal noise, we follow the approach of [8]. The probability of correctness is modeled as shown in Eq. 1.

$$p = \frac{1}{2} + \frac{1}{4}erf(\frac{V_m}{\sqrt{2}\sigma_{th}}) + \frac{1}{4}erf(\frac{V_{dd} - V_m}{\sqrt{2}\sigma_{th}}) \quad (1)$$

where, V_{dd} is supply voltage, V_m is midpoint voltage, σ_{th} is RMS value of thermal noise, $erf(x)$ is the error function. From [9], the bound for the probability of correctness is found to be $p < 1 - 0.28e^{-1.275\frac{V_{dd}^2}{8\sigma^2}}$. In the following, we study the effect of PVT variations on the probability of correctness of our pswitch.

Effect of Temperature Variation: The V_{th} and RMS value of thermal noise change with the operating temperature [14]. To study the variation of the probability of correctness and bit-entropy with temperature, we have run a simulation using the model of Eq. 1, the result of which are depicted in Fig. 2(a) for $28nm$ process. As indicated by the graph, the probability of correctness moves towards 0.5 and bit-entropy moves towards 1 as the temperature increases. This is because the RMS value of noise increases with the increase in temperature and the pswitch behaves more like a thermal noise. At the normal room temperature, the probability of correctness is in the range $[0.7 - 0.8]$ which results in reduced entropy value to lie in the range $[0.8 - 0.9]$. This concludes that at normal room temperature pswitch is not a good random number generator and we need a higher RMS value of noise to make the probability of correctness go towards the ideal value of 0.5. To compute bit-entropy (h), we have used the relation, $h = -log_2(p) \cdot p - log_2(1 - p) \cdot (1 - p)$. The values of V_{dd} and V_{th} used in the calculation are shown in Table I.

TABLE I: *Simulation parameters for our pswitch in TSMC $65nm$ and $28nm$ process.*

Technology	TSMC 65nm	TSMC 28nm
Inverter fan-out	4	4
Load capacitance (fF)	17.20	9.44
Nominal Vdd (V)	0.9 – 1.2	0.8 – 1.1
(W/L)$_{pmos}$	0.35μm/0.07μm	0.20μm/0.024μm
(W/L)$_{nmos}$	0.21μm/0.07μm	0.12μm/0.024μm
RMS value of V_{th} noise, σ_{th} (V)	0.05 – 0.5	0.05 – 0.5
RMS value of V_{dd} noise, σ_p (V)	0.05 – 0.5	0.05 – 0.5

Effect of Threshold Voltage and Supply Voltage Variations: The variation in V_{th} and V_{dd} [9] causes variation in the probability of correctness of a pswitch. Empirical evidence suggests that the variations in V_{th} can be modeled by a Gaussian distribution [9]. So, we model the V_{th} variations using a Gaussian distribution with 0 mean and standard deviation equal to 10% of the nominal V_{th}. Power supply variation is modeled using power supply noise. For power supply noise, we assume it is normally distributed with $3\sigma_p$ equal to 10% of the the nominal V_{dd} [9] where σ_p is the RMS value of V_{dd} noise. We illustrate the effect of variation in V_{dd} and V_{th} on the bit-entropy and the probability of correctness of a pswitch implemented using $28nm$ technology in Fig. 2(b)

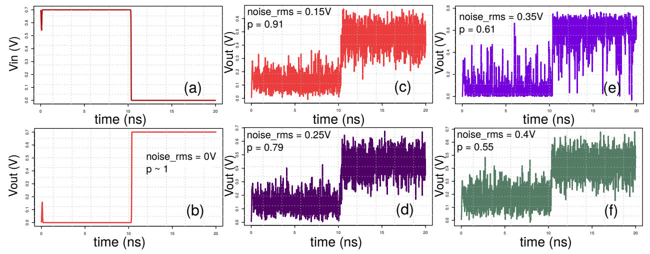


Fig. 3: *The variation of probability of correctness of pswitch with respect to σ_{th}.*

and (c), respectively. This figure is generated by running the model in Eq. 1. Fig. 2 (b) and (c) show that the variation in V_{th} does not have a significant effect on bit-entropy and probability of correctness. However, the bit-entropy (Fig. 2(b)) decreases with the increase in V_{dd} while the probability of correctness (Fig. 2(c)) increases with the increase in V_{dd}.

C. *Validation of Analytical Model of Pswitch*

To validate the analytical model of pswitch, we have performed circuit simulations in PSpice using models of inverters realized using Taiwan Semiconductor Manufacturing Company (TSMC) $65nm$ and $28nm$ processes. The simulation parameters are summarized in Table I. In our simulations, we inject thermal noise into the PSpice "netlist" in the form of a piecewise linear (PWL) voltage source. We derive the data points for the PWL source from a Gaussian distribution of random numbers generated by RStudio (coded in R). The noise pulse has identical rise and fall time and it is held constant for some amount of time. The noise is sampled at the beginning of the switching step, and it is held constant during switching. The sampling period of noise (T_{sn}) and that of the output voltage (T_{so}) are equal and are larger than the switching time (T_{sw}) of the inverter [8], [9]. The probability of correctness (p) of the pswitch is determined by sampling the output voltage at sampling frequency of ($1/T_{so}$), and is computed as follows,

$$p = \frac{\#\, of\ correct\ simulation\ points\ at\ sampling\ rate\ 1/T_{so}}{total\ \#\, of\ simulation\ points\ at\ sampling\ rate\ 1/T_{so}} \quad (2)$$

where the total number of samples is $18,000$.

We choose the sampling frequency ($1/T_{so}$) to be 12 MHz for our pswitch of $65nm$ process and 16 MHz for $28nm$ process. We obtain output samples by running the simulation of the pswitch circuit in PSpice and plot the graph (Fig. 3) of the output voltage generated by our pswitch. As shown in Fig. 3, the probability of correctness for small thermal noise is around 0.9 while the probability of correctness moves towards 0.5 as we increase the RMS value of thermal noise. We determine that the thermal noise with RMS value of $0.4\,V$ is a suitable choice which result in the probability of correctness equal to 0.55 for $65nm$ process. We have also analyzed the effect of variation of temperature and supply voltage on the probability of correctness of the circuit. Our simulation findings concur with the analytical findings presented in Section II-B.

III. High-Level Architecture of Proposed TRNG

Fig. 4 shows the high-level architecture of our proposed TRNG circuit. The main component of our TRNG is the probability stabilizer circuit which is made up of XOR gate, probability booster, and probability dropper circuits. The detailed architecture is discussed in Section IV. The stabilizer circuit takes source voltage as input. The source voltage can have a value from {HIGH, LOW} set. The stabilizer circuit then generates a corresponding output value which is again from {HIGH, LOW} set with a probability of correctness p_s. The output of the stabilizer is fed to a D-flipflop (DFF) which samples the output at a sampling frequency f_s = 12 MHz for $65nm$ process and $f_s = 16$ MHz for $28nm$ process.

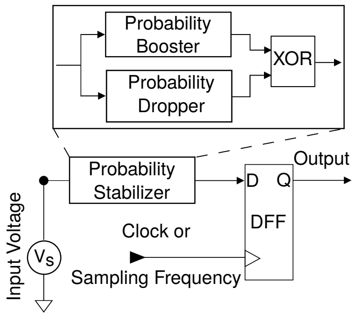


Fig. 4: *The block diagram of proposed TRNG.*

IV. Probability Booster, Dropper, and Stabilizer Circuits and their Analytical Model

In this section, first we briefly go through basic series, parallel, and XOR circuits formed by connecting pswitches. We then discuss designing probability booster, dropper, and stabilizer circuits. Note that the use of pswitch in our work is just a case study. We can replace pswitch by any unreliable entropy source, such as memrister, spin-dice, etc., to design a reliable TRNG circuit.

A. *Synthesizing Stochastic Switching Circuits using Pswitches*

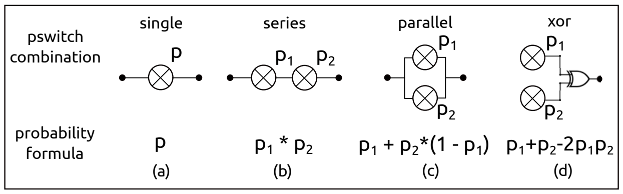
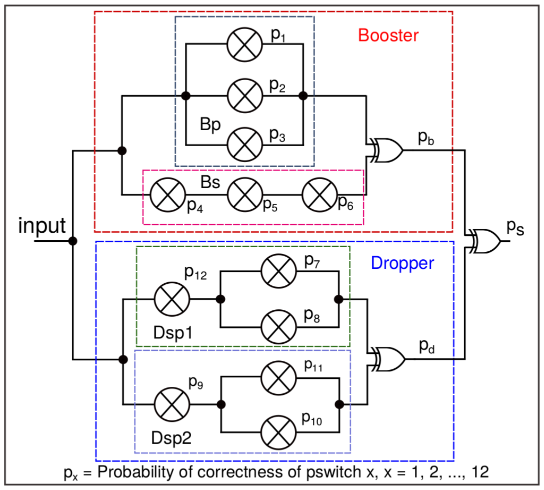


Fig. 5: *The symbolic representation of a pswitch and its series, parallel, and XOR connections.*

Series and Parallel Circuits: Let us consider two independent pswitches with the probability of correctness of p_1 and p_2, respectively. The series connection (Fig. 5(b)) of these two pswitches can produce correct output if both of the pswitches produce the correct output. Therefore, the probability that the series connection of pswitches produces correct output is the product of p_1 and p_2 i.e., $p_s = p_1 \cdot p_2$. If the two pswitches are connected in parallel (Fig. 5(c)), then the probability analysis is easier to perform considering the incorrect operation of the circuit. The parallel circuit produces incorrect output if any one of the pswitches produces incorrect output. So, the probability that a parallel connection of pswitches produces incorrect output is $p_{p_{ic}} = (1 - p_1) \cdot (1 - p_2)$. Therefore, the probability that the parallel connection of pswitches produce correct output is, $p_p = 1 - p_{p_{ic}} = p_1 + p_2 \cdot (1 - p_1)$ [15].

XOR Circuit: Let us consider the circuit in Fig. 5(d) where the output of the two pswitches are fed into an XOR gate. Let, X and Y represent the independent random bits generated by the pswitches. Since pswitches are Bernoulli processes with



Booster
Bp
Bs
input
Dropper
Dsp1
Dsp2
p_1 p_2 p_3 p_4 p_5 p_6 p_7 p_8 p_9 p_{10} p_{11} p_{12}
p_b
p_d
p_s
p_x = Probability of correctness of pswitch x, x = 1, 2, ..., 12

Fig. 6: *Internal architecture of the proposed probability stabilizer circuit. Here, we have used pswitches as the basic building block of the probability stabilizer.*

probability of correctness p_1 and p_2, E[X] = p_1 and E[Y] = p_2. From the derivation of expected value of Beta-Bernoulli process in Section IV-C, we have E[X $\oplus$ Y] = E[X] + E[Y] - 2 E[X]·E[Y]. This can be simplified in terms of probability of correctness as, $p_{xor} = p_1 + p_2 - 2p_1p_2$.

B. Internal Architecture of Probability Stabilizer Circuit

Fig. 6 shows the internal architecture of our proposed probability stabilizer circuit. As a case study we have used pswitch as a source of entropy. Each pswitch has a probability of correctness p_i where, $i \in \{1, 2, ..., N\}$ and N is the total number of pswitches in the circuit.

Observation 1: *Given a set of* 6 *pswitches with probability of correctness as* $\{p_1, p_2, ..., p_6\} \in [0.1, 0.9]$ *arranged in series-parallel connection as shown in Fig. 6 as* `Booster`*, output probability of correctness (p_b) of this series-parallel connection of pswitches always lies in interval* $[0.45, 0.99]$*. This series-parallel combination of pswitches is named probability booster circuit.*

In the following, we derive an expression for the probability of correctness of the probability booster circuit. Using the formula derived in Section IV-A, we can write the following equations for our probability booster circuit of Fig. 6, $B_p = p_1 + p_2 + p_3 - (p_1\,p_2 + p_2\,p_3 + p_3\,p_1) + p_1\,p_2\,p_3,\ B_s = p_4\,p_5\,p_6$, and $p_b = B_p + B_s - 2\,B_p\,B_s$. Let us assume that the probability of correctness of all pswitches are same and is equal to p. Then, using above expressions, the probability of correctness of the probability booster circuit is $p_b \approx 3p$. Here, the series-parallel connection of pswitches with probability p produces a circuit that has the probability of correctness approximately equal to $3p$. Hence, the circuit boosts the probability of correctness.

Observation 2: *Given a set of* 6 *pswitches with probability of correctness as* $\{p_7, p_8, ..., p_{12}\} \in [0.1, 0.9]$ *arranged in series-parallel connection as shown in Fig. 6 as* `Dropper`*, output probability of correctness (p_d) of this series-parallel connection of pswitches always lies in interval* $[0.18, 0.50]$*. This series-parallel combination of pswitches is named probability dropper circuit.*

Now, we derive an expression for the probability of correctness of the probability dropper circuit. Using the formula that we derived in Section IV-A, we can write following equations for our probability dropper circuit of Fig. 6, $D_{sp1} = p_7\,p_{12} + p_8\,p_{12} - p_7\,p_8\,p_{12},\ D_{sp2} = p_9\,p_{10} + p_9\,p_{11} - p_9\,p_{10}\,p_{11}$, and $p_d = D_{sp1} + D_{sp2} - 2\,D_{sp1}\,D_{sp2}$. Let us assume that the probability of correctness of all of the pswitches are same and is equal to p. Then, using above expressions probability of correctness of probability dropper circuit is $p_d \approx p^2$. Here, the series-parallel connection of pswitches with probability p produces a circuit that has the probability of correctness approximately equal to p^2 which is less than p. Thus, the circuit drops the probability of correctness.

Observation 3: *Given a set of pswitches with probability of correctness as* $\{p_1, p_2, ..., p_{12}\} \in [0.1, 0.9]$ *arranged in series-parallel connection as shown in Fig. 6, output probability of correctness (p_s) of this series-parallel connection of pswitches always lies in interval* $[0.45, 0.55]$*. This series-parallel combination of pswitches is named probability stabilizer circuit.*

To elucidate this observation, we derive an expression for the probability of correctness of the probability stabilizer circuit. The probability of correctness of the stabilizer circuit is the XOR of the probability of correctness of booster (p_b) and dropper circuit (p_d) and can be expressed as $p_s = p_d + p_b - 2\,p_d\,p_b$. However, it is not apparent that p_s always lies in $[0.45, 0.55]$. Hence, in the following, we perform worst and average case analysis of the probability stabilizer circuit.

C. Analyzing the Worst-Case Behavior of Proposed Probability Stabilizer Circuit

From Section II, it is apparent that the probability of correctness p of a pswitch strictly lies in the interval $[0.5, 1)$ and varies with PVT variations. To represent the stochastic behavior of pswitch, in this section, we model our pswitch as a *Beta-Bernoulli process* where the variation of p is captured by *Beta distribution* and the process of generating a random bit from $\{0, 1\}$ space is captured by *Bernoulli distribution*. We choose Beta-Bernoulli distribution because with this distribution we can model the worst-case behavior of our pswitch where the probability of correctness of our pswitch changes with time because of PVT variations. Here, we first derive the probability density function of Beta-Bernoulli process and then discuss the findings of the worst-case analysis.

Let the probability of getting correct output from a pswitch be P. Then, if we denote the output of pswitch by a random variable X, X follows a Bernoulli distribution with probability P. In real world implementation of a pswitch using PCMOS or memristor, the probability P is affected by supply voltage variation, manufacturing process variation, temperature variation, and the variation in thermal noise. Hence, we consider the probability P as a random variable following Beta distribution, $P \sim Beta(a, b)$, with parameters a and b. Then, the probability density function g of the random variable P is $g(p) = \frac{1}{\beta(a,b)} p^{a-1}(1-p)^{b-1}\ p \in (0, 1)$, where, $\beta(a, b)$ is the beta function and is a normalizing constant for the probability density function $g(p)$. Therefore, given $P = p$, the sequence X is a Bernoulli trials sequence with

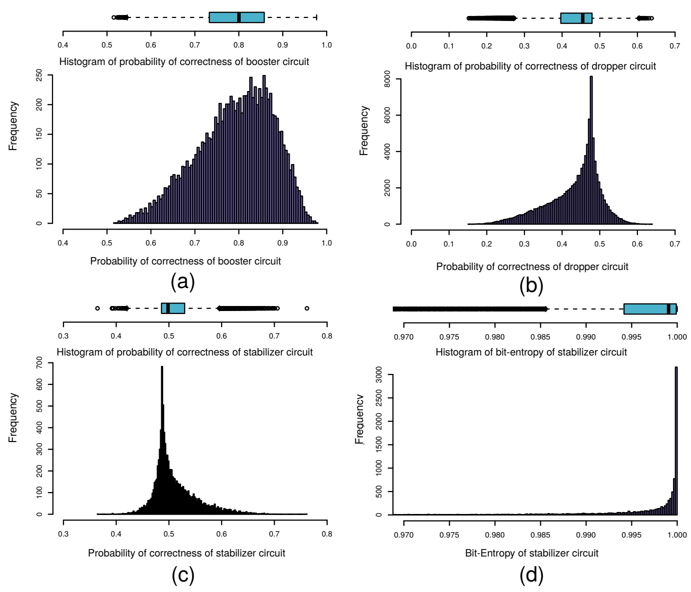


Fig. 7: *Histogram and box plot of the probability of correctness for booster, dropper, and stabilizer circuits and bit-entropy of probability stabilizer circuit obtained from our worst-case analysis.*

success parameter p where p is generated by a random source following Beta distribution. We will refer to X as the Beta-Bernoulli process with parameters a and b. The distribution of X is found to be $\mathcal{P}(X_i = k|P) = \frac{a^k b^{1-k}}{a+b}$ with mean equal to $E[X_i] = a/(a+b) = \mathcal{P}(X_i = 1|P) = \mathcal{P}(X_i = 0|P)$ and variance equal to $Var(X_i) = a/(a+b) \cdot b/(a+b)$.

The expressions for probability booster, dropper, and stabilizer circuits derived in Section IV-B serve as our analytical model. To picture the nature of probability booster, dropper, and stabilizer circuits graphically, we have implemented this model in RStudio using R and have generated the graphs shown in Fig. 7. Each pswitch of the circuit in Fig. 6 is modeled as a Beta-Bernoulli process where the probability of correctness of the pswitch follows Beta distribution and pswitch acts as a Bernoulli process generating samples from $\{0, 1\}$ binary space. The parameters for the Beta distribution are selected randomly for all of the pswitches such that different pswitches will have entirely different probability distributions.

In our experiment, we have used $100,000$ samples to compute the probability of correctness of booster, dropper, and stabilizer circuits and have averaged the results from 30 experiments. The probability of correctness values taken by booster, dropper, and stabilizer circuits are shown in Fig. 7. As shown in Fig. 7(a), the probability of correctness of booster circuit always lies in the interval $[0.6, 0.95]$ with a median value of 0.8. This is in accordance with the requirement of the booster circuit which generates a high value of probability of correctness. Fig. 7(b) depicts the probability of correctness of our dropper circuit. The probability of correctness values are spread in the interval $[0.1, 0.6]$ with a median of 0.46 which verifies that our dropper circuit can generate small values of probability. From Fig. 7(c), the probability of correctness of the stabilizer circuit (p_s) has a distribution centred at 0.48 which is illustrated by the median value in the box plot above the histogram. The box plot is small or compact in size which indicates that the values of p_s are concentrated around the 0.5 value. The histogram shows the spread of p_s which lies in interval $[0.4, 0.7]$. However, the majority (97%) of the values lies in the interval $[0.47, 0.58]$. This is a desirable property of a TRNG circuit because the probability of correctness around 0.5 gives better bit-entropy and thus better randomness. Fig. 7(d) shows the plot of bit-entropy of the stabilizer circuit. The bit-entropy of the stabilizer circuit has a median value of 0.997 which is very close to the ideal value of 1 and almost all of the bit-entropy values lies within the interval $[0.97, 1]$.

The box plot above the histogram in Fig. 7 shows many outliers which represent the combination of probabilities of pswitches that could hardly occur in a practical circuit, for example, some pswitches having a probability of correctness in the range $[0.18, 0.33]$ and others pswitches having a probability of correctness in the range $[0.90, 0.94]$. This combination of probabilities will not occur in a practical circuit because the stabilizer circuit is small and all of the components experience the same kind of operational conditions and external interference.

D. Analyzing the Average-Case Behavior of Proposed Probability Stabilizer Circuit

Worst-case analysis has many outliers due to non-practical values of pswitches. To verify that such cases would not occur in a practical circuit, we perform average-case behavior analysis. Here, to model a practical pswitch, we use the analytical model of pswitch introduced in Section II-B (Eq. 1) with the parameters listed in Table I for both the CMOS $65nm$ and $28nm$ processes. To generate a different probability of correctness for a different pswitch, we use a different RMS value of thermal noise.

Results reveal that for the dropper circuit, the probability of correctness always lies in the interval $[0.18, 0.5]$ with a median value of 0.35. The probability of correctness for the booster circuit always lies in the interval $[0.45, 1]$ with a median value 0.73. For the stabilizer circuit, the probability of correctness is crowded around the median value of 0.52 and it spreads inside the interval $[0.45, 0.55]$. The majority of bit-entropy values of the stabilizer circuit lie in the interval $[0.998, 1]$. This verifies that our probability stabilizer generates random bits with very high bit-entropy. Furthermore, results reveal that the box plot in average case analysis has few outliers. This is because we have used the analytical model of pswitch to generate the values of probability of correctness. The values generated by the model are always realistic, and our model for probability stabilizer works well for these realistic values.

E. Validation of Analytical Model of Probability Stabilizer and Proposed TRNG Circuit

To validate our analytical model, we have implemented the probability stabilizer circuit in PSpice for $65nm$ and $28nm$ processes. We have used the schematic of pswitch as described in Section II-C with one modification. The modification is that we have created pswitches that have different thermal noise coupled in their inputs such that different pswitches have different values of probability of correctness. We have done

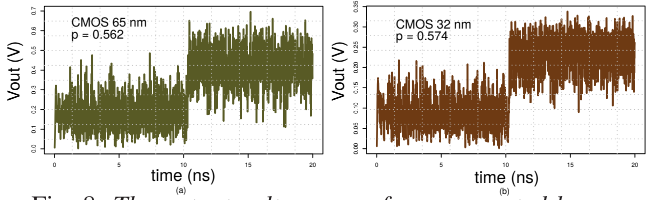


Fig. 8: *The output voltage waveform generated by our proposed probability stabilizer circuit implemented using* $65nm$ *and* $28nm$ *processes.*

this to verify our analytical model (Section IV-C and IV-D) and to show that our TRNG circuit can have pswitch (or entropy source) with any value of probability of correctness and can still produce output bitstream with very high bit-entropy. The simulation parameters are listed in Table I and simulation procedure is the same as the one described in Section II-C. To carry out functional verification of final TRNG circuit, we set the RMS value of noise to be zero for all of the pswitches and check the correctness in input-output behavior. After the functional verification, we have implemented the schematic of our probability stabilizer in $65nm$ and $28nm$ processes using Cadence Virtuoso Layout tool. Fig. 9 shows the layout of TRNG circuit implemented using TSMC $28nm$ process.

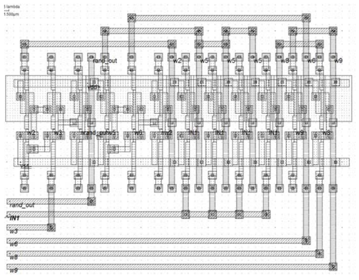


Fig. 9: *Layout of TRNG circuit implemented using TSMC* $28nm$ *process in Cadence Virtuoso Layout Tool.*

Functional Verification of Stabilizer Circuit: Fig. 8 shows the output waveform of the stabilizer circuit. We have sampled this output at the sampling frequency (f_s) of 12 MHz for $65nm$ and 16 MHz for $28nm$ process. We have used Eq. 2 to compute the probability of correctness of our stabilizer circuit. The probability of correctness for the stabilizer implemented using $65nm$ and $28nm$ processes are calculated to be 0.562 and 0.574, respectively. This verifies the claim that we made in Section IV about our stabilizer circuit. The bit-entropy of the circuit simulated for $65nm$ is around 0.972 and for $28nm$ is around 0.984. This result corroborates with the result from our analytical findings in Section IV. The data rate of the TRNG designed using $65nm$ process is observed to be around 10 Mbps and the data rate for TRNG designed using $28nm$ process is around 16 Mbps.

Resistance Against Temperature Variation: Fig. 10 shows that as the temperature increases, the probability of correctness of booster and dropper circuits increases. However, the probability of correctness of the stabilizer circuit stays in the neighborhood of 0.5. The bit-entropy of the booster circuit keeps on decreasing with temperature because the probability of correctness for the booster circuit increases with temperature. For the dropper circuit, as its probability of correctness moves towards 0.5 value with rise in temperature, its bit-entropy moves towards the ideal value of 1. For the stabilizer circuit, the bit-entropy is around the ideal value of 1.

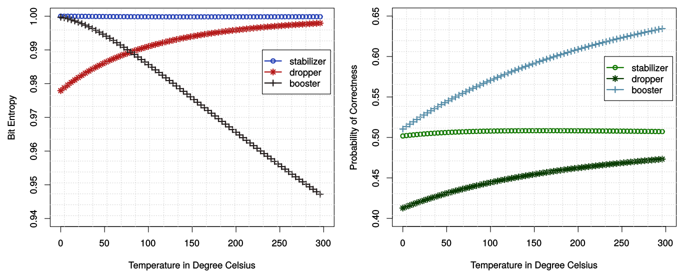


Fig. 10: *Effect of temperature variation in probability stabilizer circuit.*

Resistance Against Supply Voltage Variation: Fig. 11 and Fig. 12 depict the effect of variation of V_{dd} on the probability of correctness and bit-entropy of booster, dropper, and stabilizer circuits implemented using $28nm$ process. The variation in V_{dd} is modeled as V_{dd} noise with RMS value σ_p and we assume that V_{dd} noise has normal distribution with zero mean and σ_p standard deviation where $3\sigma_p$ equals to 10% of nominal V_{dd}. To analyze the worst case behavior, we use different nominal value of V_{dd} and different σ_p for different pswitch. The range of values of σ_p and V_{dd} are shown in Table I. In our experiment, the value of V_{dd} varies in the interval $[0.47, 1.81]$ for $65nm$ process and in the interval $[0.42, 1.43]$ for $28nm$ process. As shown in Fig. 11(a), the probability of correctness for the booster circuit spreads above 0.5, and for the dropper circuit, the probability of correctness spreads below 0.5 for various values of V_{dd} (refer Table I). Fig. 11(b) is a zoomed version of Fig. 11(a) where we show the fluctuation in the of probability of correctness of the stabilizer circuit with variation in V_{dd}. The figure confirms that the probability of correctness lies well within 0.5. Note that these figures do not show x-axis because we have 12 different types of variations in V_{dd} values for 12 different pswitches which need a $12D$ plot to show the trend of probability of correctness (or bit-entropy) with respect to V_{dd} variation. Hence, we have just plotted the probability of correctness and bit-entropy as points in $2D$ space.

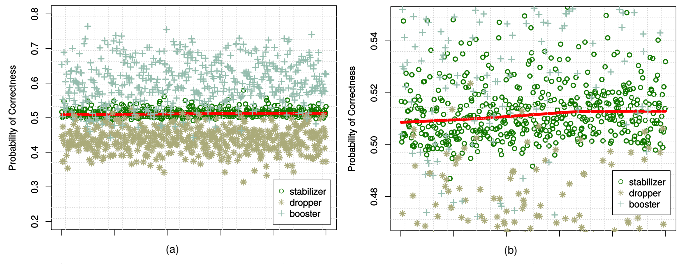


Fig. 11: *(a) Scatter plot for variation of probability of correctness of probability booster, dropper, and stabilizer circuits for* $28nm$ *process due to* V_{dd} *variation, (b) The probability of correctness variation plot in (a) zoomed around* 0.5 *value.*

Fig. 12 depicts the spread of bit-entropy values for the probability booster, dropper, and stabilizer circuits. The bit-entropy of the booster and dropper circuits are affected by the variation of V_{dd}. However, the bit-entropy of the stabilizer circuit remains in the neighborhood of 1.

NIST Randomness Test: To evaluate the randomness of our TRNG, we have tested bitstreams of a million bits using

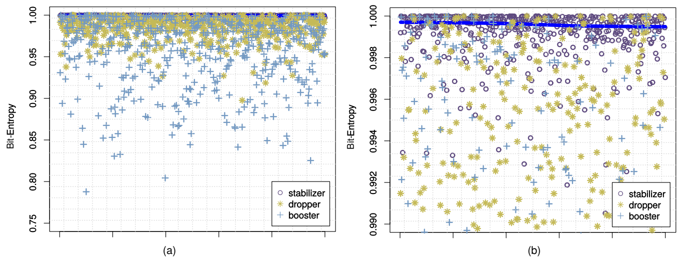


Fig. 12: *(a) Scatter plot for variation of bit-entropy of booster, dropper, and stabilizer circuits for $28nm$ process due to V_{dd} variation, (b) The bit-entropy variation plot in (a) zoomed around 0.5 value.*

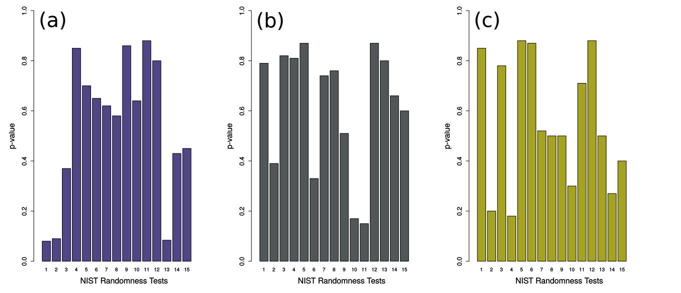


Fig. 13: *NIST test suite results for the proposed TRNG. The numbers 1 to 15 respectively represents frequency, block frequency, runs, longest runs of 1s, binary matrix rank, DFT, non-overlapping template, overlapping template, Maurer's universal, linear complexity, serial, approximate entropy, cumulative sums, random excursions, and random excursion variant tests, (a) 298K, nominal process, (b) 373K, +20% Δp, (c) 373K, -20% Δp. Here, Δp represents process variation.*

NIST's SP 800-22 statistical test suite for the validation of our TRNG circuit. We generate bitstreams by sampling the analog output of our TRNG circuit. Further, we perform this test for bitstreams generated from various process and temperature corners to appraise the PVT performance of our TRNG. We have tested a total of 10 binary sequences for each scenario. We observe that the generated bitstreams pass all the NIST tests for the various process and temperature corners (Fig. 13), where a pass is deemed to be a p-value ≥ 0.01, implying 99% confidence levels.

V. Conclusion

In this paper, we propose a generic architecture for TRNG that is robust against the PVT variations and is resistant to noninvasive fault attacks such as varying the supply voltage and operating temperature. The proposed architecture of the TRNG is a generic architecture in the sense that it can be used with any unreliable entropy source to design a reliable TRNG. The fundamental component of our TRNG is a probability stabilizer circuit which is made up of probability dropper and probability booster circuits. The simulation results and empirical data from the analytical model shows that the bit-entropy of our proposed TRNG always lies strictly in the interval $[0.998, 1]$ for both $65nm$ and $28nm$ processes. As a case study, we have analyzed our proposed TRNG architecture using a pswitch as a source of entropy. From our analytical modeling, we can claim that our proposed TRNG circuit can be used with any other entropy sources such as memristor, magnetic tunnel junction, etc., without loss of probabilistic behavior of the final TRNG circuit. Using the NIST SP 800-22 test suite for randomness, we demonstrate that the output of our proposed TRNG circuit is statistically random with 99% confidence levels.

Acknowledgments: This work was supported by the National Science Foundation (NSF) (NSF-CNS-1743490). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

[1] M. Stipcevic and C. K. Koc. (2017, Jan) True random number generators. [Online]. Available: http://cs.ucsb.edu/~koc/cren/docs/w06/trng.pdf

[2] H. Nejati, A. Beirami, and W. H. Ali, "Discrete-time chaotic-map truly random number generators: design, implementation, and variability analysis of the zigzag map," *Analog Integrated Circuits and Signal Processing*, vol. 73, no. 1, pp. 363–374, 2012.

[3] V. B. Suresh. (2012, Feb) On-chip true random number generation in nanometer cmos. [Online]. Available: http://scholarworks.umass.edu/cgi/viewcontent.cgi?article=1872&context=theses

[4] X. li, B. Taylor, Y. T. Chien, and L. T. Pileggi, "Adaptive post-silicon tuning for analog circuits: Concept, analysis and optimization," in *IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, USA, Dec 2007.

[5] B. Datta and W. Burleson, "Calibration of on-chip thermal sensors using process monitoring circuits," in *2010 11th International Symposium on Quality Electronic Design (ISQED)*, San Jose, CA, USA, March 2010, pp. 461–467.

[6] F. Rahman, B. Shakya, X. Xu, D. Forte, and M. Tehranipoor, "Security beyond cmos: Fundamentals, applications, and roadmap," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. PP, no. 99, pp. 1–14, 2017.

[7] N. Rangarajana, A. Parthasarathy, and S. Rakheja, "A spin-based true random number generator exploiting the stochastic precessional switching of nanomagnets," *Journal of Applied Physics*, vol. 121, no. 22, 2017.

[8] P. Korkmaz, B. E. S. Akgul, K. V. Palem, and L. N. Chakrapani, "Advocating noise as an agent for ultra-low energy computing: Probabilistic complementary metaloxidesemiconductor devices and their characteristics," *Japanese Journal of Applied Physics*, vol. 45, no. 45, pp. 3307–3316, 2006.

[9] P. Korkmaz, B. E. S. Akgul, and K. V. Palem, "Energy, performance, and probability tradeoffs for energy-efficient probabilistic cmos circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, no. 8, pp. 2249–2262, Sept 2008.

[10] B. E. S. Akgul, L. N. Chakrapani, P. Korkmaz, and K. V. Palem, "Probabilistic cmos technology: A survey and future directions," in *IFIP International Conference on Very Large Scale Integration*, Nice, France, Oct 2006, pp. 1–6.

[11] S. Cheemalavagu, P. Korkmaz, K. V. Palem, B. E. S. Akgul, and L. N. Chakrapani, "A probabilistic cmos switch and its realization by exploiting noise," in *Proc. of the IFIP international*, NY, USA, 2005.

[12] J. P. Uyemura, *CMOS Logic Circuit Design*. Kluwer Academic, Boston, 1999.

[13] S. Pant, D. Blaauw, V. Zolotov, S. Sundareswaran, and R. Panda, "A stochastic approach to power grid analysis," in *Proc. of 41st Design Automation Conference (DAC)*, San Diego, CA, USA, July 2004, pp. 171–176.

[14] N. Paydavoshi, T. Morshed, D. Lu, W. Yang, M. Dunga, X. Xi, J. He, W. Liu, Kanyu, M. Cao, X. Jin, J. Ou, M. Chan, A. Niknejad, and C. Hu. (2017, Jan) Bsim4v4.8.0 mosfet model - user's manual. [Online]. Available: http://ngspice.sourceforge.net/external-documents/models/BSIM480_Manual.pdf

[15] D. Wilhelm and J. Bruck, "Stochastic switching circuit synthesis," in *IEEE International Symposium on Information Theory*, Toronto, Ontario Canada, July 2008, pp. 1388–1392.