Beat Frequency Detector based High-Speed True Random Number Generators: Statistical Modeling and Analysis

Yingjie Lao, University of Minnesota
Qianying Tang, University of Minnesota
Chris H. Kim, University of Minnesota
Keshab K. Parhi, University of Minnesota

True random number generators (TRNGs) are crucial components for the security of cryptographic systems. In contrast to pseudo random number generators (PRNG), TRNGs provide higher security by extracting randomness from physical phenomena. In order to evaluate a TRNG, statistical properties of the circuit model and raw bitstream should be studied. In this paper, a model for the beat frequency detector based high-speed TRNG (BFD-TRNG) is proposed. The parameters of the model are extracted from the experimental data of a test chip. A statistical analysis of the proposed model is carried out to derive mean and variance of the counter values of the TRNG. Our statistical analysis results show that mean of the counter values is inversely proportional to the frequency difference of the two ring oscillators (ROSCs), while the dynamic range of the counter values increases linearly with standard deviation of environmental noise and decreases with increase of the frequency difference. Without the measurements from the test data, a model cannot be created; similarly without a model performance of a TRNG cannot be predicted. The key contribution of the proposed approach lies in fitting the model to measured data, and the ability to use the model to predict performance of BFD-TRNGs that have not been fabricated. Several novel alternate BFD-TRNG architectures are also proposed; these include parallel BFD, cascade BFD, and parallel-cascade BFD. These TRNGs are analyzed using the proposed model, and it is shown that the parallel BFD structure requires less area per bit, while the cascade BFD structure has a larger dynamic range while maintaining the same mean of the counter values as the original BFD-TRNG. It is shown that the $3.25M$ and $4M$ random bits can be obtained per counter value from parallel BFD and parallel-cascade BFD, respectively, where M counter values are computed in parallel. Furthermore, the statistical analysis results illustrate that the BFD-TRNGs have better randomness and less cost per bit than other existing ROSC-TRNG designs. For example, it is shown that the BFD-TRNGs accumulate 150% more jitter than the original two-oscillator TRNG, and parallel BFD-TRNGs require one-third power and one-half area for same number of random bits for a specified period.

Categories and Subject Descriptors: B.7.0 [**Hardware**]: General

General Terms: Security, Design, Performance

Additional Key Words and Phrases: Beat Frequency Detector, Hardware Security, Jitter, Post-Processing, Randomness, Ring Oscillator, Statistical Analysis, True Random Number Generator, Unbiasedness

ACM Reference Format:
Yingjie Lao, Qianying Tang, Chris H. Kim, and Keshab K. Parhi. 2015. Statistical Modeling and Analysis of Beat Frequency Detector based True Random Number Generators. *ACM J. Emerg. Technol. Comput. Syst.* V, N, Article A (January YYYY), 22 pages.
DOI : http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

The security of most cryptographic systems relies on unpredictability and irreproducibility of digital key-streams that are used for encryption and/or signing of confidential information. These key-streams are generated by random number generators (RNG), which can be further classified into two categories: true random number generators (TRNG) and pseudo random number generators (PRNG). The key difference between TRNG and PRNG lies in the entropy source component. A TRNG derives randomness from an analog physical process (electronic thermal noise, radioactive decay, etc.), while a PRNG relies on computational complexity, whose outputs are completely determined by the seed. TRNGs are used for authentication and encryption purposes in systems requiring a high level of security. On-chip TRNGs typically harvest randomness from a circuit that converts transistor level noise such as random telegraph noise (RTN), flicker noise and thermal noise [Brederlow et al. 2006; Holcomb et al. 2007; Tokunaga et al. 2008; Majzoobi et al. 2011; Srinivasan et al. 2010; Yang et al. 2014; Rahman et al. 2014] into a voltage or delay signal.

A source of randomness commonly used in FPGA and ASIC implementations of TRNGs is the unpredictability of signal propagation time across logic gates. This unpredictability is typically accumulated in so-called ring oscillators (ROSCs), consisting of a series of inverters or delay elements connected in a ring. The phase jitter of a ring oscillator is then extracted by another ring oscillator or by an external clock signal. Ring oscillators and the underlying physical phenomena have been

widely studied in the literature as building blocks for many on-chip TRNGs [Petrie and Connelly 1998; 2000; Epstein et al. 2003; Bock et al. 2004; Kohlbrenner and Gaj 2004; Sunar et al. 2007; Wold and Tan 2009; Valtchanov et al. 2009]. One major advantage of these TRNG designs is that no analog component is required, while conventional delay based TRNGs typically involve extensive analog components for amplifying the device noise [Bucci et al. 2003], which makes them less suitable for practical TRNG devices.

Evaluating TRNGs is a difficult task. Clearly, it should not be limited to testing the TRNG output bitstream. The physical characteristics of the source of randomness and the randomness extraction method determine the principal parameters of the generated bit stream: the bias of the output bit stream, correlation between subsequent bits, visible patterns, etc. While some of the non-randomness can be corrected by efficient post-processing, it is better if the generator inherently produces a good quality random bitstream. Furthermore, passing NIST [Rukhin et al. 2001] or DIEHARD [Marsaglia 1996] tests does not guarantee a TRNG, as these tests were originally designed to check the performance of PRNGs.

One important requirement in TRNG security evaluation is the existence of a mathematical model of the physical noise source and the statistical properties of the digitized noise derived from it [Killmann and Schindler 2001]. If a stochastic model of the physical randomness source is available, it can be used in combination with the raw signal to estimate the entropy and the bias depending on the random input variables and the TRNG principle. Therefore, in order to provide a proof of security for a TRNG, an analysis of the statistical property of the underlying mathematical model is needed. However, creating a model of a TRNG is difficult as the model parameters are unknown. Thus, it is impossible to predict performance of new TRNG designs as their models cannot be created. On the other hand, it can be argued that TRNG performance can only be measured from fabricated chips. Therefore, how good a new TRNG design can only be determined by measurements from a fabricated design. This paper exploits the synergy between a model and the measurements of the real device. A new ROSC based BFD-TRNG was fabricated and tested [Tang et al. 2014]. Based on NIST tests, this TRNG was demonstrated to be an effective TRNG. This paper, for the first time, presents a model of this BFD-TRNG. The model parameters are derived by fitting the data measured from the fabricated device. Based on this created model, a rigorous analysis of the BFD-TRNG is presented. Furthermore, several new BFD-TRNG architectures are proposed and their performances are predicted based on the proposed model.

The rest of this paper is organized as follows. In Section 2, we review the high-speed BFD-TRNG design. Section 3 describes statistical modeling of the physical components in ROSC based TRNGs. In Section 4, we present a comprehensive statistical analysis for BFD-TRNGs. Motivated by our statistical analysis results, we propose a number of alternate BFD-TRNG architectures in Section 5. We summarize the performance comparisons between the BFD-TRNG designs and other existing ROSC based TRNGs in Section 6. Finally, Section 7 presents remarks, conclusions and future directions.

2. BEAT FREQUENCY DETECTOR BASED HIGH-SPEED TRNG

The oscillator sampling method extracts randomness from phase noise in free-running oscillators [Petrie and Connelly 1998; 2000; Kohlbrenner and Gaj 2004]. An example of this technique is shown in Fig. 1, where the output of a fast oscillator is sampled on the rising edge of a slower ring oscillator using a D flip-flop (DFF). Note that the design parameters for the inverters of the two ROSCs are not necessarily the same. The timing fluctuations of the edges of the slow signal relative to the fast oscillator is the source of the randomness in the ROSC based TRNG. Oscillator jitter causes uncertainty in the exact sample values, ideally producing a random bit for each sample. Additionally, randomness can be artificially enhanced by carefully selecting the ratio of the fast and slow oscillator frequencies. Periods of these oscillations vary from cycle to cycle causing jitter in the rising and falling edges. The goal is to sample the signal at a point in time that is in close proximity of a transition zone thereby making sampled value unpredictable. In order to accumulate sufficient jitter when the fast ring oscillator is sampled, a large ratio of the fast and slow oscillator frequencies is usually desired. Note that the slow oscillator can also be substituted by an external clock, such as in the IBM M-parallel structure [Liberty et al. 2013].

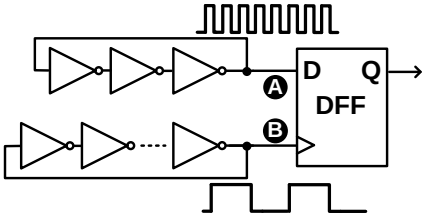

Fig. 1: Two-Oscillator TRNG.

Built on the prior work of ROSC based TRNGs, we have proposed a novel TRNG design to harvest randomness from jitter variation based on the beat frequency detector (BFD) [Tang et al. 2014]. A beat frequency detector captures the frequency difference between the two ROSCs [Kim et al. 2008] with a very high resolution, which was originally used to measure frequency degradation of digital circuits. As shown in Fig. 2, the ROSC A is continuously sampled by a ROSC B whose frequency is slightly different from ROSC A. The output of the DFF exhibits the beat frequency Δf, which is determined by the frequency difference of the two ROSCs. A counter measures the beat frequency with ROSC B as the clock. The counter output increments every ROSC period until it reaches the beat frequency interval after which the count is sampled and reset. The output count will fluctuate due to the random jitter in the circuit. The mean of the frequency difference of the two ROSCs is caused by manufacturing process variations, and can be further adjusted by trimming capacitors associated with the ring oscillators [Tang et al. 2014]. The average frequency tune resolution is 0.1%. The ROSC frequency decreases as we increase the load of each ROSC stage by enabling more MOS capacitors. For example, if we would like to increase the counter values, we can either enable additional capacitors on the fast ROSC or disable capacitors on slow ROSC to achieve the target count range. In our test chip data, the initial count measured from different chips ranges from 200 to 1000 when using the same trimming capacitor setting. Through extensive testing, we found that a count range of 200 to 500 provides a reasonable trade-off between speed and bit efficiency. A simple one-time calibration step shown in Fig. 3 can be used to guarantee that the initial count is in the desired range (200 to 500) across the different TRNG chips. This can be readily achieved within a few beat frequency periods using minimal hardware overhead during the initial startup. Fig. 4 shows the measured average count through a continuous 15 hour operation test. Without any real-time calibration, the TRNG generates a steady output across a long operation period. Under the presented setting, we can generate approximately 3.25 bits per sample by using first 3 least significant bits (LSBs) directly and processing the 4th LSB with the von Neumann corrector [Von Neumann 1951].

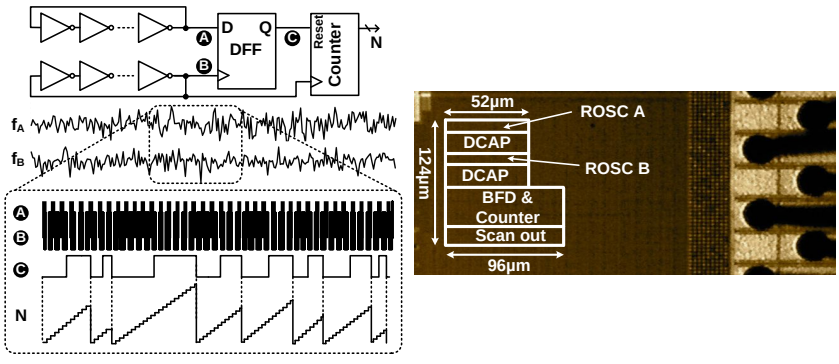

Fig. 2: BFD-TRNG: (a) basic principle, (b) die microphotograph in 65nm.

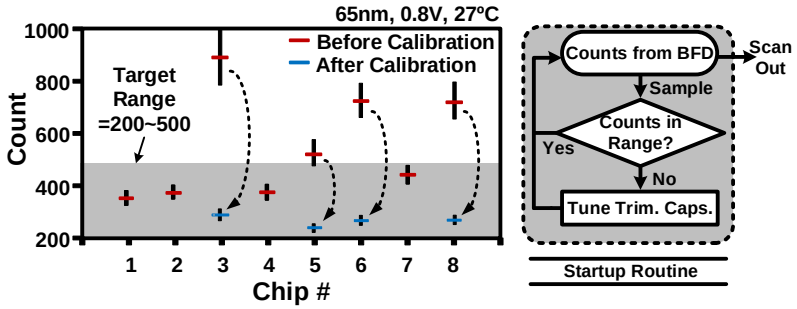

Fig. 3: One-time calibration of average count during start up.

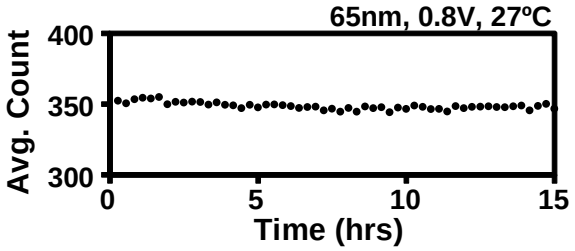

Fig. 4: Stability under continuous operation.

3. PHYSICAL COMPONENT MODELING OF ROSC TRNGS

As discussed above, the statistical tests such as NIST and DIEHARD are designed to check the performance of PRNGs. The core of a TRNG is its randomness source, which usually generates a time-continuous analog signal that is digitized by certain harvest mechanism. In order to validate a TRNG, characterization of the randomness source and the harvest mechanism are needed. In this section, we investigate the statistical properties of the BFD-TRNG.

The randomness source of the ring oscillator based TRNGs is the timing jitter in each ROSC, which is a stochastic phenomenon caused by internal random noise such as thermal, shot, and random telegraph noise in the transistors of a ring oscillator. Jitter can be considered as a short-term variation of a digital signal from their ideal position in time. The size of the jitter is determined by the properties of the hardware device and the operating environment. In these ROSC based TRNG designs, two or more oscillators are combined to produce a random bitstream. This jitter will create an accumulated phase drift in each ring so that the transition region in the sampling period is assumed to be unpredictable. In the literature, several studies of the jitter in ring oscillators have been presented [Petrie and Connelly 1996; Schindler 2003; Abcunas 2004; Coppock 2005; Abidi 2006; Baudet et al. 2011; Wold 2011]. More precisely, the jitter model should incorporate a Gaussian variable, flicker noise, and a coupling sinusoidal signals [Petrie and Connelly 2000]. However, existing works [Schindler 2003; Abcunas 2004] report that the durations between the transition times appear in many cases to be independent and identically distributed Gaussian, as it is the most dominant component. This allows us to create simple model for ROSC based TRNGs by characterizing the jitter as a Gaussian random variable with zero mean. Moreover, there are two major reasons that we do not consider Random Telegraph Noise (RTN) as the major random noise source: First, due to the averaging effect, the RTN induced jitter is much smaller than that on a single transistor. Second, the

occurrence of RTN with large amplitude and high frequency is rare [Brederlow et al. 2006; Tang et al. 2013].

With respect to the flicker noise, we know that the flicker noise will dominate in the low frequency domain while the Gaussian white noise dominates in the high frequency domain [Schindler 2003; Abcunas 2004]. However, the frequency of the ROSC in our current test chip is about 356 MHz. Therefore, the impact of flicker noise will be negligible at this frequency region. Since the counter values we obtained from our silicon results are in the range of $[200, 400]$, the frequency of counter output is around 1 MHz, which is still relatively high and is greater than the corner frequency of the flicker noise. Moreover, based on our silicon results, there is no sign that the flicker noise plays a significant role in our BFD-TRNG design. Our BFD-TRNG design could pass all NIST tests when the ROSC has a frequency of 356 MHz and the counter outputs are in the range of $[200, 400]$. Our silicon results show that the correlation of the 4 LSBs between two successive counter outputs and the correlation among the 4 LSBs of the same counter output are both very small. Our test results also show that the first 3 LSBs can be directly concatenated and streamed out without any post-processing, while the 4th LSB can also pass all the NIST tests after applying von Neumann correction. In conclusion, in our current test chip, the flicker noise does not play a significant role in the BFD-TRNG, as the frequency of the counter output is still relative high, which has also been confirmed by our silicon results. However, it is important to note that if the BFD-TRNG is operated in a frequency that is lower than the corner frequency in a future fabricated chip, then the flicker noise must be incorporated into the model, as the flicker noise could be a major contributor.

A ROSC consists of an odd number of inverters connected together in a ring configuration. This causes the output of the oscillator to change with a period of approximately $2kD$, where k is the number of inverters in a ROSC and D is the delay of a single inverter. If we consider the delay of each inverter as a Gaussian random variable $D_i \sim N(\mu_i, \sigma_i^2)$, a period of the ROSC can be written as

$$T = 2\sum_{i=1}^{k} Di \sim N(\mu, \sigma^2), \tag{1}$$

which is also a Gaussian random variable. For simplicity, we directly consider a period of the ROSC as a Gaussian random variable in this paper. Periods vary from cycle to cycle causing jitter in the rising and falling edges. Note that this model can incorporate different operating conditions (e.g., temperature, supply voltage) by modifying σ accordingly.

4. STATISTICAL ANALYSIS OF BFD-TRNG

Based on the illustrated model above, this section presents a comprehensive statistical analysis to help resolve some important BFD-TRNG design issues:

(a) How much of the frequency difference is required to produce sufficient random numbers?
(b) How many bits of the counter value can be used?
(c) How can the TRNG performance be further improved?

4.1. BFD Model

As shown in Fig. 2, the BFD-TRNG consists of two ROSCs whose frequencies are slightly different. The period of the two ROSCs can be modeled as $TA \sim N(\mu_A, \sigma_A^2)$ and $TB \sim N(\mu_B, \sigma_B^2)$, respectively. Note that the two ring oscillators are implemented identically; therefore their free-running frequencies are very close but not identical due to the process variation. To prevent injection locking phenomenon or any other unintended coupling between the two ROSCs, we separated the frequencies of the two ROSCs using trimming capacitors prior to the testing. Furthermore, the two ROSCs are oscillating and trimmed independently. Experimental data showed no signs of correlation between the ROSC frequencies [Tang et al. 2014]. An output will be generated once the beat frequency is obtained, i.e., the faster ROSC completes one more cycle than the slower ROSC. The output will be the number of cycles completed by ROSC B at this moment. Without loss of generality, we always assume ROSC A is faster than ROSC B in this paper, i.e., $\mu_A < \mu_B$. Since the inverters in ROSC A and ROSC B are almost equivalently designed with only slight frequency difference and operated under the same environmental condition, we can assume $\sigma = \sigma_A = \sigma_B$.

Therefore, the probability density function (pdf) of the counter value N can be expressed as

$$pdf(N) = \{\min N : \sum_{i=1}^{N} TA_i < \sum_{i=1}^{N-1} TB_i\} = \{\min N : TA_N < \sum_{i=1}^{N-1} (TB_i - TA_i)\}. \quad (2)$$

However, the N in Equation (2) does not have a standard probability density function. Instead, we perform Monte Carlo simulations to study the statistical properties of this model. Model parameters extracted from experimental measurements in [Tang et al. 2014] imply $\frac{\Delta\mu}{\mu_B} = \frac{\mu_B - \mu_A}{\mu_B} = 0.28\%$ and $\sigma_A = \sigma_B = 0.0006$. Note that in this paper, without loss of generality, we always assume the mean of the clock signal of the DFF to be 1 (i.e., $\mu_B = 1$). Therefore, $\Delta\mu = 0.0028$ and $\mu_A = 0.9972$ in this setup. The distribution of the counter values is shown in Fig. 5.

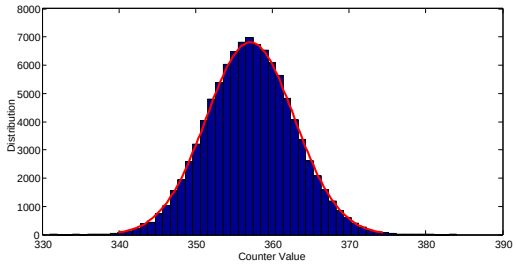

Fig. 5: Counter Value Distribution ($\Delta\mu = 0.28\%$, $\sigma = 0.0006$).

4.2. Effect of Counter Value

4.2.1. Mean of Counter Value. It can be seen from Fig. 5 that the mean of the counter values is close to $\frac{1}{\Delta\mu} = 357$. We repeat the simulation with $\Delta\mu = 0.4\%$ as shown in Fig. 6. The mean is also close to $\frac{1}{\Delta\mu} = 250$. Thus, we observe that the mean of counter values is inversely proportional to the value of $\Delta\mu$.

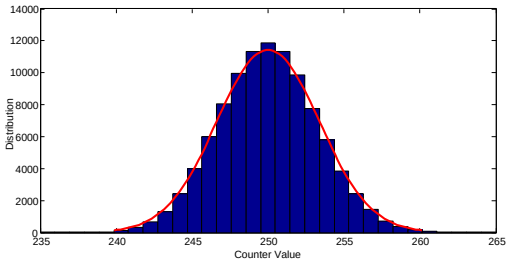

Fig. 6: Counter Value Distribution ($\Delta\mu = 0.4\%$, $\sigma = 0.0006$).

This property can also be derived by mathematically. Since Equation (2) does not have a closed-form expression, we consider a simpler case: TA and TB remain unchanged during one measurement time. In fact, this is the original function of a beat frequency detector, i.e., to measure the frequency difference of ROSC A and ROSC B. In this case, Equation (2) can be simplified to

$$N = \{\min N : |N - \frac{NTB}{TA}| \geq 1\}. \quad (3)$$

By solving the equation, we can get

$$N = \lceil \frac{TB}{|TB - TA|} \rceil. \tag{4}$$

There is one trivial observation that $|TB-TA|$ cannot be very small; otherwise, the counter value will be very large (i.e., only one counter value can be obtained in very large number of cycles). If we consider TA and TB as the average periods of N cycles, these can be characterized as Gaussian random variables $\frac{\sum_{i=1}^{N} TA_i}{N} \sim N(1 - \Delta\mu, \frac{\sigma^2}{N})$ and $\frac{\sum_{i=1}^{N} TB_i}{N} \sim N(1, \frac{\sigma^2}{N})$, respectively. Thus,

$$\frac{TB}{TB - TA} \sim \frac{N(1, \frac{\sigma^2}{N})}{N(\Delta\mu, \frac{2\sigma^2}{N})}, \tag{5}$$

which can be described as a ratio of two Gaussian random variables. We can approximate the expected value of $\frac{TB}{TB-TA}$ by a second order Taylor expansion [Stuart and Ord 1994]:

$$\begin{aligned} E(\frac{TB}{TB - TA}) \approx & \frac{E(TB)}{E(TB - TA)} - \frac{Cov(TB, TB - TA)}{E^2(TB - TA)} + \frac{Var(TB - TA)E(TB)}{E^3(TB - TA)} \\ = & \frac{1}{\Delta\mu} - \frac{\frac{\sigma^2}{N}}{\Delta\mu^2} + \frac{\frac{2\sigma^2}{N}}{\Delta\mu^3} \\ = & \frac{1}{\Delta\mu} + \frac{1}{N}\frac{\sigma^2}{\Delta\mu^2}(\frac{2}{\Delta\mu} - 1). \end{aligned} \tag{6}$$

Based on the assumption that TA and TB are uncorrelated, we can obtain that $Cov(TB, TB - TA) = Var(TB) = \sigma^2$. For the parameters $\Delta\mu = 0.28\%$, $\sigma = 0.0006$, the above equation will equal to

$$\frac{1}{\Delta\mu} + \frac{1}{N}\frac{\sigma^2}{\Delta\mu^2}(\frac{2}{\Delta\mu} - 1) = 357.14 + \frac{32.75}{N}. \tag{7}$$

It can be seen from Fig. 5 that $330 < N < 390$. Consequently, the second term of Equation (7) (i.e., $\frac{32.75}{N}$) is less than 0.1. Consequently, the mean of counter values is approximately $\frac{1}{\Delta\mu} \approx 357$ in this case.

Typically, the second term of Equation (7) will be relatively small compared to $\frac{1}{\Delta\mu}$, since $\Delta\mu$ is a very small number and $\frac{\sigma^2}{\Delta\mu^2}$ is generally less than 1.

As a result, we can conclude

$$E(N) \approx \frac{1}{\Delta\mu}. \tag{8}$$

Thus, in contrast to the original function of the BFD, i.e., to measure the slight frequency difference of two signals, we should set up an appropriate frequency difference for the two ROSCs. In other words, trimming capacitors should be used to set an appropriate $\Delta\mu$ for the two ROSCs, instead of equalizing their frequencies or making the frequency differences as small as possible. Therefore, we mainly harvest randomness from the jitter noise instead of the metastability in our design, as the random numbers are generated from the delay difference variations of the two ROSCs, instead of from sampling one ROSC with another ROSC. Note that our current test chips provide a frequency trimming resolution of 0.1%, this can be further reduced with more trimming capacitor bank controls.

4.2.2. Dynamic Range of Counter Value. The randomness of the BFD-TRNG comes from the counter values. If the dynamic range is larger, we could use more bits of the counter value as random numbers. Therefore, it is important to examine the statistics of the dynamic range of the counter values. We attempt to use a Gaussian distribution to fit the counter values from experimental measurements and consider $6\sigma_G$ as the dynamic range based on the fitting result $N(\mu_G, \sigma_G^2)$.

An example is shown in Fig. 5, with parameters $\Delta\mu = 0.28\%$, $\sigma = 0.0006$. It is shown that the distribution of the counter values is close to a Gaussian distribution. In this case, the dynamic range $6\sigma_G$ is equal to 34.6.

We repeat the simulation for different parameters as shown in Fig. 6 and Fig. 7, whose dynamic ranges are 20.3 and 68.9, respectively. The relationship between the dynamic range of counter values and $\Delta\mu$ is shown in Fig. 8, where the mean of counter values is equal to $\frac{1}{\Delta\mu}$. It can be seen that the dynamic range of counter values will increase with the increase of σ, while it will decrease with the increase of $\Delta\mu$. This observation is also conformed from our measured chip data [Tang et al. 2014]. Moreover, it can be seen that only slight change of $\Delta\mu$ will affect the counter values significantly.

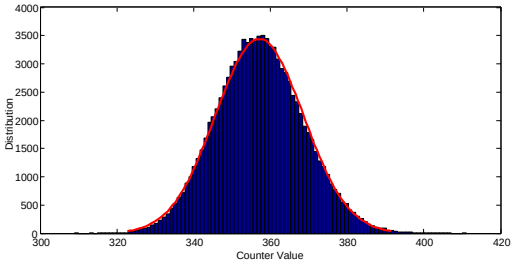

Fig. 7: Counter Value Distribution ($\Delta\mu = 0.28\%$, $\sigma = 0.0012$).

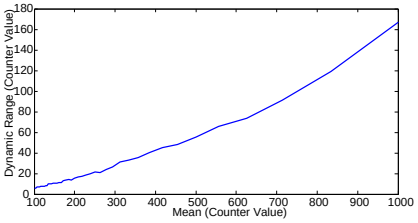

Fig. 8: The relationship between the dynamic range of counter values and $\Delta\mu$ ($\sigma = 0.0006\%$).

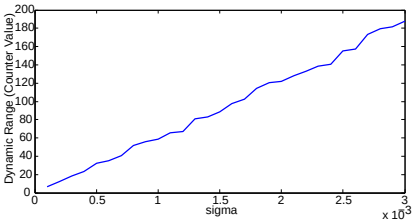

Fig. 9: The relationship between the dynamic range of counter values and σ ($\Delta\mu = 0.28\%$).

We can also examine the variance of counter values by a second order Taylor expansion [Stuart and Ord 1994]:

$$\begin{aligned} Var(\frac{TB}{TB-TA}) \approx & \frac{E^2(TB)}{E^2(TB-TA)}(\frac{Var(TB)}{E^2(TB)} - 2\frac{Cov(TB, TB-TA)}{E(TB)E(TB-TA)} + \frac{2Var(TB-TA)}{E^2(TB-TA)}) \\ = & \frac{1}{\Delta\mu^2}(\frac{\sigma^2}{N} - \frac{2\sigma^2}{N\Delta\mu} + \frac{2\sigma^2}{N\Delta\mu^2}) \\ = & \frac{\sigma^2}{N\Delta\mu^2}(1 - \frac{2}{\Delta\mu} + \frac{2}{\Delta\mu^2}). \end{aligned} \tag{9}$$

Since we have demonstrated that the expected value of N is approximately $\frac{1}{\Delta\mu}$ and $\Delta\mu$ is a very small value, we can further approximate the above equation as

$$\begin{aligned} Var(\frac{TB}{TB-TA}) \approx & \frac{\sigma^2}{N\Delta\mu^2}(1 - \frac{2}{\Delta\mu} + \frac{2}{\Delta\mu^2}) \\ \approx & \frac{\sigma^2}{\Delta\mu}\left(2(\frac{1}{\Delta\mu} - \frac{1}{2})^2 + \frac{1}{2}\right) \\ \approx & \frac{2\sigma^2}{\Delta\mu^3}. \end{aligned} \tag{10}$$

Consequently, we can obtain

$$\sigma_G \approx \sqrt{Var(\frac{TB}{TB-TA})} \approx \frac{\sqrt{2}\sigma}{\Delta\mu^{\frac{3}{2}}}. \tag{11}$$

Thus, we can conclude that the dynamic range of counter values increases linearly with σ and is inversely proportional to $\Delta\mu^{\frac{3}{2}}$.

4.3. Bounds on Bias of Each Bit

According to the NIST test, the probability of "1" occurrence, $p1$, should satisfy $49.91\% \leq p1 \leq 50.09\%$ to pass the frequency test of NIST tests [Rukhin et al. 2001]. For any bit of the counter values, if the probability of "1" occurrence is within the acceptance range $[49.91\%, 50.09\%]$, then this bit might be used as random numbers directly. Otherwise, certain techniques are required to post-process the bit. Note that we only consider the biasedness metric in our simulation, as other metrics are completely dependent on the performance of the employed pseudo random number generator for simulation. The $p1$ of each bit for the distribution in Fig. 5 is presented in the second row of Table I. Since the counter values are less than 512 in our setup, we only need to consider the first 9 LSBs. It can be seen that the first 3 LSBs are within the acceptance range. In fact, our chip experimental results show that all of the first 3 LSBs can pass the NIST test individually or collectively after serializing them.

Table I: The Probability of "1" Occurrence $p1$ for Each Bit

Distribution	$p1(b_8)$	$p1(b_7)$	$p1(b_6)$	$p1(b_5)$	$p1(b_4)$	$p1(b_3)$	$p1(b_2)$	$p1(b_1)$	$p1(b_0)$
Fig. 5	1	0	1	0.8384	0.1979	0.4564	0.5003	0.4991	0.4995
Fig. 10	0.5173	0.4827	0.4827	0.4827	0.4838	0.4991	0.4998	0.5007	0.5005

We also present the value of $p1$ of each bit for the distribution as shown in Fig. 10 with parameters $\Delta\mu = 0.391\%$ and $\sigma = 0.0009$ in Table I. The dynamic range $6\sigma_G$ is 30.2 which is less than the dynamic range of the distribution in Fig. 5. However, the first 4 LSBs are within the acceptance range. Furthermore, the higher bits are also less biased compared to counter values in Fig. 5. Therefore, the counter values in Fig. 10 have better randomness than the counter values in Fig. 5, even though the dynamic range of the counter values in Fig. 10 is smaller.

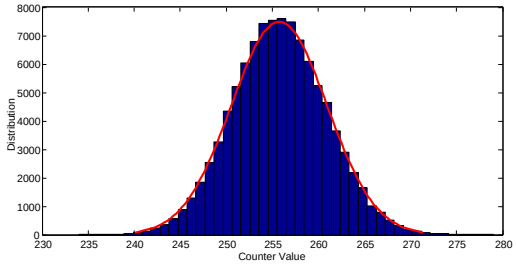

Fig. 10: Counter Value Distribution ($\Delta\mu = 0.391\%$ and $\sigma = 0.0009$).

As a result, we can conclude that the number of bits that we can use is not only dependent on the dynamic range, but also dependent on the mean of the counter values. For example, we consider the two cases as shown in Fig. 11. The counter values in the top and the bottom panels of Fig. 11 have the same dynamic range. However, the mean of counter values in the top panel is 15.5, which is just at the boundary of $b_3 = 0$ and $b_3 = 1$. As a result, the expected bias ε will be 0, where ε is defined as $\varepsilon = |0.5 - p1|$. For the counter values in the bottom panel whose mean is 19.5, $p1 = 0.11$. Thus, the bias is $|0.5 - p1| = 0.39$. This is because the mean of the counter values is in the middle of the region where $b_3 = 0$.

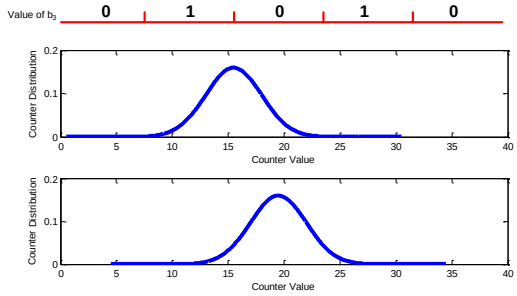

Fig. 11: The biasedness of b_3 for different counter values.

For a certain dynamic range of counter values, in the best case, $E(N) = 2^k m - 0.5$ for b_k (m is an integer), which leads to $\varepsilon = 0$. In the worst case, $E(N) = 2^k m + 2^{k-1} - 0.5$, which generates the largest bias ε. Ideally, we can extract more randomness from the counter values by carefully adjusting the mean. However, in order to ensure the quality of each bit when taking noise and operating environmental change into consideration, we have to consider the bias in the worst case. Table II presents the corresponding bias ε for each bit under different σ_G in the worst case.

It can be seen from Table II that the first LSB is guaranteed to be unbiased if the dynamic range of counter values is greater than $6\sigma_G = 12$. Furthermore, if the dynamic range is greater than 30, the first 3 LSBs might be used as random numbers without any post-processing. As a result, we have to ensure at least a dynamic range of 30 for the BFD-TRNG [Tang et al. 2014], if we output the first 3 LSBs directly. In addition, we cannot use b_4 to b_8 directly while the dynamic range is less than 90. Since the dynamic range increases with the increase of the mean of the counter values, we have to set an appropriate $\Delta\mu$ to attain sufficient randomness of the TRNG design.

Fig. 12 shows the relationship between the number of bits that we can output directly and the dynamic range. The number of unbiased bits can be expressed by fitting the curve perfectly as

$$\text{number of unbiased bits} = \lfloor log_2 \frac{40}{23} + log_2 \sigma_G \rfloor = \lfloor log_2 \frac{40}{23} + log_2 \sigma - \frac{3}{2} log_2 \Delta\mu + \frac{1}{2} \rfloor. \quad (12)$$

Table II: Bias ε for Each Bit under Different σ_G in the Worst Case

σ_G	$\varepsilon(b_8)$	$\varepsilon(b_7)$	$\varepsilon(b_6)$	$\varepsilon(b_5)$	$\varepsilon(b_4)$	$\varepsilon(b_3)$	$\varepsilon(b_2)$	$\varepsilon(b_1)$	$\varepsilon(b_0)$
1	0.5	0.5	0.5	0.5	0.5	0.4999	0.4545	0.1854	0.0046
2	0.5	0.5	0.5	0.5	0.4999	0.4545	0.1854	0.0046	0
3	0.5	0.5	0.5	0.5	0.4923	0.3176	0.0397	0	0
4	0.5	0.5	0.5	0.4999	0.4545	0.1854	0.0046	0	0
5	0.5	0.5	0.5	0.4986	0.3904	0.0926	0.0003	0	0
6	0.5	0.5	0.5	0.4923	0.3176	0.0397	0	0	0
7	0.5	0.5	0.5	0.4777	0.2475	0.0146	0	0	0
8	0.5	0.5	0.4999	0.4545	0.1854	0.0046	0	0	0
9	0.5	0.5	0.4996	0.4246	0.1336	0.0012	0	0	0
10	0.5	0.5	0.4986	0.3904	0.0926	0.0003	0	0	0
11	0.5	0.5	0.4964	0.3542	0.0618	0.0001	0	0	0
12	0.5	0.5	0.4923	0.3176	0.0397	0	0	0	0
13	0.5	0.5	0.4862	0.2816	0.0245	0	0	0	0
14	0.5	0.4999	0.4777	0.2475	0.0146	0	0	0	0
15	0.5	0.4999	0.4671	0.2153	0.0083	0	0	0	0

Therefore, we can conclude that the number of bits that we can use is logarithmically proportional to the dynamic range of the counter values.

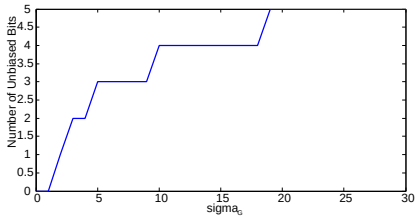

Fig. 12: The relation between the number of unbiased bits and the value of σ_G.

Summary
The observations are summarized below.

1. The mean of the counter values is inversely proportional to $\Delta\mu$.
2. The dynamic range of counter values is inversely proportional to $\Delta\mu^{\frac{3}{2}}$ and is proportional to σ.
3. The sampling rate is inversely proportional to the mean of counter values.
4. The number of unbiased bits is logarithmically proportional to the dynamic range of counter values; thus, it is inversely logarithmically proportional to $\Delta\mu^{\frac{3}{2}}$.
5. The sampling rate is proportional to $\Delta\mu$.

As we are only able to control the value of $\Delta\mu$, we should set an appropriate $\Delta\mu$ to achieve better throughput. Note that a higher σ can be obtained by appropriately sizing the transistor and choosing number of stage in the ROSCs, but this is out of the scope of this paper. For post-fabrication throughput optimization, we are able to achieve better throughput with the following two methods based on $\Delta\mu$ controlling.

(a) Use a higher $\Delta\mu$ (i.e., lower counter values), which could improve the sampling rate. But we may only be able to use limited number of bits from each counter value as random numbers.
(b) Use a lower $\Delta\mu$ (i.e., higher counter values). The sampling rate is reduced. But we can use more bits from the counter values as random numbers. Moreover, we can further post-process the higher bits of the counter values to generate more bits.

Based on the summary above, we can obtain the relationship between the throughput (i.e., rate $\times$ number of unbiased bits) of the BFD based design and parameters $\Delta\mu$, σ as below:

$$\text{throughput} \propto \Delta\mu\lfloor log_2\frac{40}{23} + log_2\sigma_G\rfloor \approx \Delta\mu\lfloor log_2\frac{40}{23} + log_2\sigma - \frac{3}{2}log_2\Delta\mu + \frac{1}{2}\rfloor. \quad (13)$$

If we consider Equation (13) without the floor function as

$$\text{throughput} \propto \Delta\mu(log_2\frac{40}{23} + log_2\sigma - \frac{3}{2}log_2\Delta\mu + \frac{1}{2}). \quad (14)$$

Equation (14) achieves maximum value at

$$\Delta\mu = 2^{(\frac{2}{3}log_2\frac{40\sqrt{2}\sigma}{23} - \frac{1}{ln2})}. \quad (15)$$

For example, the values of Equations (13) and (14) are shown in Fig. 13 when $\sigma = 0.0006$.

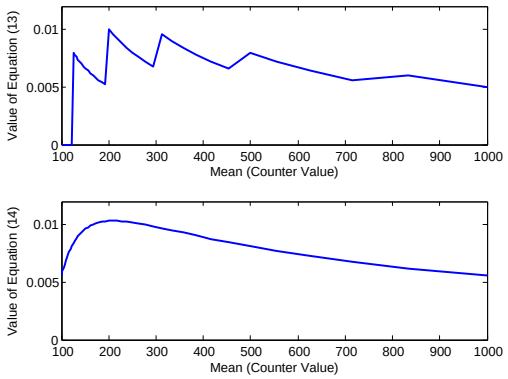

Fig. 13: Values of Equations (13) and (14) for different $\Delta\mu$'s ($\sigma = 0.0006$).

It can be seen from the top panel of Fig. 13 that the BFD-TRNG achieves the best throughput when $\Delta\mu = 0.0050$ (2 bits can be used as random numbers), while the maximum value of Equation (14) is achieved at $\Delta\mu = 2^{(\frac{2}{3}log_2\frac{40\sqrt{2}\sigma}{23} - \frac{1}{ln2})} = 0.004768$ as shown in the bottom panel. Therefore, in this case, we can adjust the $\Delta\mu$ such that the mean of the counter values is about 200 to achieve higher throughput. Generally, the $\Delta\mu$ should be adjusted according to the environmental noise σ based on Equation (15).

4.4. Post-Processing

As discussed above, the number of bits that can be treated as random numbers is determined by both the dynamic range and the mean of counter values. For those bits with some randomness that do not meet the NIST test requirement (i.e, $0.0009 < \varepsilon < 0.5$), post-processing techniques can be used to generate more random bits.

Post-processing techniques for TRNGs are used to ameliorate non-randomness in the raw bit-stream, which are basically compression functions that are applied to the raw bitstream before the output of the TRNG. Furthermore, post-processing techniques can improve the stability of a TRNG, as it is able to correct the raw bitstream if operating conditions change. In the BFD-TRNG design, if a certain bit starts out with a high bias, the post-processing step would transform the bitstream such that the bias becomes more acceptable. The two common techniques we consider in the post-processing step include block-wise XOR and Von Neumann corrector [Von Neumann 1951]. Other techniques, such as linear compression functions based on good linear codes can also be used for de-biasing [Sunar et al. 2007; Dichtl 2007; Lacharme 2008]. The comparison of block-wise XOR and Von Neumann corrector is illustrated in Table III, where XOR d corresponds to XOR operation with a block size of d.

Table III: Comparison of Block-Wise XOR and Von Neumann Corrector (The Bits are Assumed to Be Independent)

	XOR d	Von Neumann
Rate	$\frac{1}{d}$	$\frac{1}{4}-\varepsilon^2$
bias	$2^{d-1}\varepsilon^d$	0

Each of these post-processing techniques has its pros and cons. Using Von Neumann corrector will produce perfect correction with 0 bias but throughput is reduced to less than 25% of its original. XOR may achieve better throughput with a small d. However, we have to ensure that $2^{d-1}\varepsilon^d$ is within the acceptance range. For example, the compression rate is 50% when $d = 2$. However, the bias is only improved from ε to $2\varepsilon^2$.

Table IV presents the number of bits that we can generate for different σ_G in the worst case. Each value in Table IV represents how many bits can be used as random numbers for a certain counter value. For example, if a bit has a bias $\varepsilon < 0.0009$, the bit can produce 1 random bit per sample. However, if the bias ε exceeds the threshold, this bit of the counter value can be used to generate $\frac{1}{d}$ random bit per sample by using block-wise XOR or $\frac{1}{4}-\varepsilon^2$ bit per sample by the Von Neumann corrector. Entropy is essentially an upper bound on the number of bits that we can generate for a given dynamic range. Note that the total entropy can be obtained by the sum of the entropy for each individual bit, according to Table II.

Table IV: Number of bits per Sample for Different σ_G in the Worst Case

σ_G	Von Neumann	XOR	Entropy
1	0.51	0.64	2.17
2	1.51	1.64	3.17
3	2.41	2.4	3.75
4	2.51	2.64	4.17
5	3.34	3.29	4.49
6	3.41	3.4	4.75
7	3.46	3.61	4.97
8	3.51	3.64	5.17
9	3.55	3.7	5.34
10	4.34	4.29	5.49
11	4.37	4.3	5.62
12	4.41	4.4	5.75
13	4.43	4.42	5.86
14	4.46	4.61	5.97
15	4.49	4.63	6.07

It can be seen that the block-wise XOR and Von Neumann corrector have comparable performances for the BFD-TRNG. However, for the bit with a small bias, XOR will be more favored than Von Neumann corrector. As we always try to utilize the bits with smaller biases, block-wise XOR could outperform Von Neumann corrector in general. From the simulation results, besides the bits of counter values that we can output directly, we may only be able to use 2 more bits by post-processing, as the rate will be too low for the higher bits. According to Table II, for example, we can use the first 3 LSBs directly when $\sigma_G = 7$. If we only want to post-process the 4th LSB to generate more bits, we can generate 3.5 bits per sample by XOR or approximately 3.25 bits per sample by Von Neumann corrector. Moreover, the output of block-wise XOR is synchronous while the output of Von Neumann corrector is asynchronous. The challenge of using block-wise XOR is that we need to determine the value of d for each bit based on the dynamic range of the counter values. Although we use Von Neumann corrector in our work [Tang et al. 2014], future work will be directed towards using block-wise XOR to extract more randomness of the BFD-TRNG designs.

5. ALTERNATE BFD-TRNG ARCHITECTURES

Motivated by the statistical analysis results, in this section, we propose a number of alternate BFD-TRNG designs, which can further improve the performances.

5.1. Parallel Structure

Parallelizability is also a desired metric of a TRNG. In fact, the BFD-TRNG is notably easy to parallelize by adding as many extra ROSCs instead of 2 ROSCs to generate multiple outputs, as shown in Fig. 14, where the block of the Beat Frequency Counter is same as the counter used in the BFD-TRNG as shown in Fig. 2(a). Our experimental results show that the counter values of the two adjacent outputs are highly correlated. However, the advantage of the BFD-TRNG is that we can consider bits of the counter values individually. For example, the simulated correlation coefficients for a 4-parallel structure are presented in Table V. Note that the $\Delta\mu$ is assumed to be same for all the ROSC pairs. According to NIST test, only the bitstreams with correlation coefficients less than 0.073 can pass the test [Rukhin et al. 2001]. It can be seen that the correlation coefficients of the counter values of two adjacent outputs are large, i.e., around 0.5. However, the correlations of the first 4 LSBs from two adjacent outputs are very small, while the counter values and individual bits are not correlated for non-adjacent outputs. Therefore, it can be concluded that the first 4 LSBs from the outputs in Fig. 14 can still be used as random numbers. In general, we can generate $3.25M$ bits per sample by using $M + 1$ ROSCs and M DFFs, since we can use the first 3 LSBs directly and generate $\frac{1}{4}$ bits per count by postprocessing the 4th LSB with Von Neumann corrector.

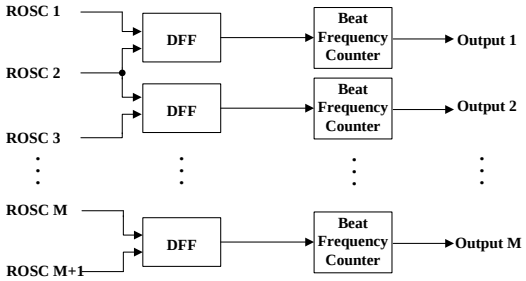

Fig. 14: An M-parallel TRNG structure.

Table V: Correlation Coefficients of Each Bit among the Outputs for a 4-Parallel Structure

	Correlation Coefficients					
	Output (1,2)	Output (2,3)	Output (3,4)	Output (1,3)	Output (2,4)	Output (1,4)
Counter Value	0.4967	0.4970	0.4974	0.0018	-0.0006	-0.0007
b_0	0.0005	0.0005	0.0008	-0.0003	0.0000	0.0009
b_1	0.0003	0.0008	0.0016	-0.0006	-0.0004	-0.0018
b_2	0.0010	0.0013	0.0005	0.0004	0.0001	-0.0016
b_3	0.0027	0.0049	0.0023	0.0014	-0.0006	-0.0004
b_4	0.1193	0.1177	0.1190	-0.0008	-0.0009	0.0002
b_5	0.2974	0.2966	0.2971	0.0006	-0.0013	0.0001
b_6	0.3023	0.2998	0.3001	0.0009	-0.0010	0.0001
b_7	0.3023	0.2998	0.3001	0.0009	-0.0010	0.0001
b_8	0.3023	0.2998	0.3001	0.0009	-0.0010	0.0001

5.2. Cascade Structure

A novel cascade structure which could achieve better randomness is shown in Fig. 15. Note that ROSC B also connects to the clock signal of the counter (not shown in Fig. 15). The dynamic range of this cascade structure is higher than that of the original BFD-TRNG. Fig. 16 shows the counter value distributions of the original BFD-TRNG and the cascade structure. Note that in our simulation, we set the $|f_A - f_B| - |f_B - f_C| = 0.28\%$ to maintain the same mean of the counter values. In other words, the frequencies of the 2 DFFs in the first stage are not very close to each other.

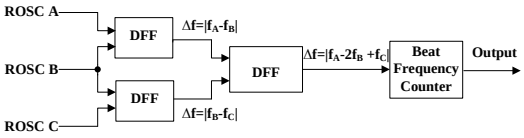

Fig. 15: A cascade TRNG structure.

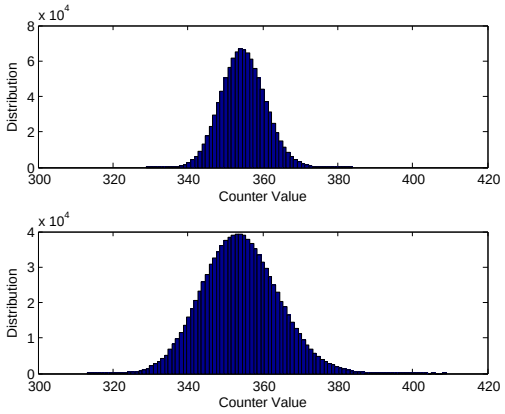

Fig. 16: Counter values of the original BFD-TRNG (top figure) and the cascade structure (bottom figure).

The dynamic range is increased from 34.6 to 64.3 by adopting the cascade structure. As a result, we can also use the 4th LSB directly according to Table II, while maintaining the same mean of the counter values. Therefore, the randomness is improved by using the cascade structure. Alternatively, we can reduce the mean of the counter values to increase the sampling rate, while maintaining considerable randomness. The cascade structure will be extremely useful when the environmental noise is small. The BFD-TRNG may even be adaptively configured between 3 ROSCs and 2 ROSCs. For example, if the noise is relatively small, we could use 3 ROSCs as shown in Fig. 15 to increase the dynamic range of counter values; otherwise, we could use 2 ROSCs as shown in Fig. 14 to output 2 bitstreams of random numbers. Furthermore, this cascade structure provides higher flexibility for adjusting trimming capacitors associated with the ring oscillators.

5.3. Parallel-Cascade Structure

We can also parallelize the cascade structure, which leads to the so-called parallel-cascade structure of the BFD-TRNG. For example, a 4-parallel-cascade structure is shown in Fig. 17. Two adjacent outputs share two ring oscillators, while the outputs that are separated from one output share one ring oscillator. We also need to examine the correlation coefficients of the outputs to ensure the bits that are used as random numbers are not correlated. The simulated correlation coefficients for the counter values and the individual bits among different outputs are presented in Table VI. Note that we set all the Δf's in Fig. 17 as 0.0028. It can be seen that the correlation coefficients for the

first 4 LSBs between any two of the outputs are still very small and satisfy the NIST criteria. The correlations of the counter values and higher bits (i.e., b_4 to b_8) between two adjacent outputs are large, while the correlation coefficients of the counter values and higher bits between the outputs 1 and 3 or the outputs 2 and 4 are smaller but still exceed the threshold (i.e., 0.0073). The outputs 1 and 4 are not correlated for both the counter values and the individual bits, since they do not share any ring oscillator. Therefore, an M-parallel cascade structure can generate $4M$ bits per sample by using $(M + 2)$ ROSCs and $(2M + 1)$ DFFs. Note that in order to generate 4 bits from each output, we need to ensure the frequencies are either descending or ascending from the first ROSC to the last ROSC, as $\Delta f = ||f_A - f_B| - |f_B - f_C|| = |f_A - 2f_B + f_C|$ only if $f_A > f_B > f_C$ or $f_A < f_B < f_C$. Otherwise, Δf will equal to $|f_A - f_C|$, which leads to the same dynamic range as the original BFD-TRNG. As a result, only 3.25 bits can be obtained from each output in this case. However, there is a problem when M is large that the frequency difference between the first ROSC and the last ROSC will be fairly large if we want to generate 4 bits from each output, which may exceed the capability of the trimming capacitors. Therefore, the frequencies need not necessarily be set as either descending or ascending from the first ROSC to the last ROSC, which leads to a parallel-cascade structure where some of the outputs can generate 4 bits each and the others can generate 3.25 bits each. The performance is still improved.

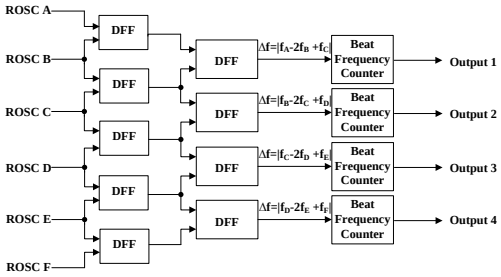

Fig. 17: A 4-parallel-cascade structure.

Table VI: Correlation Coefficients of Each Bit among the Outputs for a 4-Parallel-Cascade Structure

	Correlation Coefficients					
	Output (1,2)	Output (2,3)	Output (3,4)	Output (1,3)	Output (2,4)	Output (1,4)
Counter Value	0.6640	0.6634	0.6650	0.1666	0.1664	-0.0027
b_0	0.0002	-0.0004	-0.0003	0.0001	0.0008	0.0000
b_1	-0.0009	0.0012	0.0005	-0.0014	0.0005	-0.0003
b_2	-0.0007	0.0011	-0.0007	-0.0006	0.0010	-0.0002
b_3	-0.0029	-0.0015	-0.0023	0.0005	0.0006	-0.0004
b_4	0.0377	0.0382	0.0388	0.0197	0.0181	-0.0001
b_5	0.3539	0.3534	0.3542	0.1032	0.1013	0.0009
b_6	0.1771	0.1673	0.1701	0.0163	0.0189	-0.0015
b_7	0.1817	0.1721	0.1760	0.0145	0.0177	-0.0011
b_8	0.1817	0.1721	0.1760	0.0145	0.0177	-0.0011

6. COMPARISON WITH OTHER EXISTING ROSC BASED TRNGS

Furthermore, by adopting the proposed statistical model, we could also analyze prior ring oscillator based TRNG designs. In this section, we present the performance comparisons of the BFD-TRNG with other existing ring oscillator based TRNGs.

6.1. Two-Oscillator TRNG

The most comprehensive model of a two-oscillator TRNG is presented in [Baudet et al. 2011]. In this section, we analyze the two-oscillator TRNG as shown in Fig. 1 based on our simple model, i.e., assume a Gaussian random variable for the period of a ring oscillator. As discussed in Section 2, the frequency ratio between the two ROSCs plays a very important role in the randomness of the output. Experimental results have shown that the randomness is the worst when the fast oscillator frequency is an integer multiple of half the slow oscillator frequency [Petrie and Connelly 1996]. In practice, the ratio is often carefully selected to achieve better randomness [Kohlbrenner and Gaj 2004].

However, even if the TRNG design was originally designed to operate at a suitable oscillator frequency/sampling frequency ratio, a change in environmental conditions or worse adversarial influences may shift the frequency ratio to a weak operating point. It is claimed that the amount of accumulated jitter $6\sigma_{acc}$ should be at least six times as large as the period of the fast oscillator to attain sufficient randomness [Balachandran and Barnett 2008]:

$$6\sigma_{acc} \geq 6\mu_A, \tag{16}$$

where $\sigma_{acc}^2 \approx \sigma_B^2 + L\sigma_A^2$, since the randomness is generated from the timing fluctuations of the edges of the slow signal relative to the fast oscillator. Let L represent the number of periods ROSC A is completed before it is sampled. If we assume the design parameters for the inverters of the two ROSCs are the same, σ_B^2 will equal $L\sigma_A^2$, since the two ROSCs accumulated approximately with the same amount of jitter. Consequently, the value of L can be calculated as:

$$L \geq \frac{\mu_A^2}{2\sigma_A^2}. \tag{17}$$

In order to ensure sufficient randomness, a large frequency ratio is required. For example, L should be greater than 1 million when $\sigma_A = 0.0006$. However, in the application of two-oscillator TRNG, the value of σ_A is usually much larger. Frequency dividers can also help to achieve a large frequency ratio [Bucci and Luzzi 2008; Fischer et al. 2008]. Furthermore, a smaller ratio is sufficient to pass the NIST test in practice (i.e., NIST test is not that strict, compared to the statistical analysis). For example, experimental results [Amaki et al. 2013] show that the period of a 7-stage ring oscillator implemented with a 65 nm CMOS process is $220ps$ from circuit simulation; thus, $220 \times 6 = 1320ps$ of jitter is required. On the other hand, the jitter amount of a 251-stage ring oscillator with 64-frequency dividers is measured as $100ps$, which is much smaller than the necessary value. Moreover, the results in [Liberty et al. 2013] demonstrate that at least a ratio of 500 is required to achieve sufficient randomness to pass the NIST test.

6.2. ROSC TRNG with XOR Tree

A ROSC TRNG with XOR tree has been proposed in [Sunar et al. 2007], which does not require large frequency separation of the fast and slow ring oscillators. The outputs from the oscillator rings are XOR-ed together and sampled with a DFF. A series of ring oscillators are combined to compensate for the imbalance between the number of zeros and ones in the random signal. In this structure, the jitter is accumulated spatially instead of temporarily. The TRNG structure is shown in Fig. 18.

A stochastic approach of this TRNG is presented in [Sunar et al. 2007]. It shows that in order to increase the entropy of the generated binary raw signal and to make the generator provably secure, large number of ROSCs needs to be employed. Experimental results show that the outputs of at least 114 supposedly independent ROSCs are XOR-ed and sampled using a reference clock with a fixed frequency can pass the NIST test. Only a small frequency ratio of 5 to 20 is required (e.g., approximately 6 in [Sunar et al. 2007]).

However, some weakness of this TRNG design has been pointed out in [Dichtl and Golić 2007]. The main concern is that the XOR-tree and the sampling D flip-flop cannot handle the high number of transitions from the oscillator rings. With many oscillator rings in parallel, the number of transitions during a sampling period will be too high to meet the setup/hold-time requirements. Experimental results show that approximately 50% of the transitions get lost [Rozic and Verbauwhede

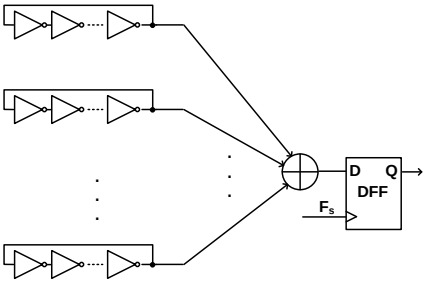

Fig. 18: ROSC TRNG with XOR Tree.

2009]. To cope with the problem with many transitions in the sampling period, an enhanced TRNG based on the ROSCs has been proposed in [Wold and Tan 2009] by adding an extra DFF after each ring oscillator before the XOR gate Fig. 19. This TRNG design can generate desirable raw bitstream with a significantly reduced number of ROSCs. Its outputs can pass the NIST and DIEHARD tests without postprocessing.

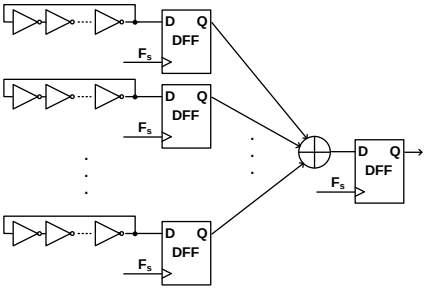

Fig. 19: Enhanced ROSC TRNG with XOR Tree.

The mathematical models for the ROSC TRNG with XOR tree as shown in Fig. 18 and the enhanced structure as shown in Fig. 19 are the same [Bochard et al. 2009]. Similar to the two-oscillator based TRNG, the variance of the accumulated jitter of the ROSC TRNG with XOR tree can be expressed as

$$\sigma_{acc}^2 \approx \sigma_B^2 + ML\sigma_A^2, \tag{18}$$

where M is the number of ROSCs in parallel and L is the frequency ratio.

The number of ROSCs can be reduced by using the enhanced ROSC TRNG with XOR tree [Wold and Tan 2009]. Experimental results in [Wold and Tan 2009] show that 50 ROSCs in parallel are required to achieve sufficient randomness to pass the NIST test. However, this TRNG design is still not very efficient, since most of the ROSCs in this structure do not improve the entropy of random numbers if their transition regions are not sampled.

6.3. Comparison

There are a number of advantages of the BFD-TRNG designs. First of all, the random numbers of the BFD-TRNG are generated from counter values, which is a better harvest mechanism that can utilize more of the entropy. The bits per sample can be increased by post-processing or appropriately adjusting the counter values, while other existing ROSC based TRNGs are only able to generate maximum 1 bit per sample. Moreover, we could also choose to post-process with the counter values instead of individual bits.

Furthermore, other existing ROSC based TRNGs are sampled continuously. If the accumulated jitter is not sufficient between consecutive samplings, these samples will be correlated. However,

for the BFD-TRNG, the counter will be reset after collecting the data. As a result, the correlation between consecutive samples is reduced.

We continue to compare their performances according to evaluation metrics as below.

6.3.1. Randomness. In fact, the BFD-TRNG can be considered as a faster ROSC B that is sampled by a slower ROSC with frequency $|f_A - f_B|$. Therefore, the variance of the accumulated jitter between two consecutive samplings is

$$\sigma_{acc}^2 \approx 2^2\sigma_B^2 + L\sigma_A^2, \tag{19}$$

where L is equal to the counter value N in this case. This is similar to the sum of the jitter in ROSC A and two times of the jitter in ROSC B. If we still assume the clock signal is generated from a slower ROSC and the design parameters for the inverters in the two ROSCs are the same (i.e., $\sigma_B^2 = L\sigma_A^2$), the value of σ_{acc}^2 for the BFD-TRNG is

$$\sigma_{acc}^2 \approx 5L\sigma_A^2. \tag{20}$$

Similarly, the cascade structure as shown in Fig. 15 can be considered as a faster ROSC B which is sampled by a slower ROSC with frequency $|f_A - 2f_B + f_C|$. In this case, the accumulated jitter will be the sum of the jitter in ROSC A, the jitter in ROSC C, and three times of the jitter in ROSC B. As a result, the value of σ_{acc}^2 for the cascade structure will be

$$\sigma_{acc}^2 \approx (1 + 1 + 3^2)L\sigma_A^2 = 11L\sigma_A^2. \tag{21}$$

The value of σ_{acc}^2 for each TRNG design is summarized in Table VII.

Table VII: Comparison of σ_{acc}^2 for Different ROSC based TRNG Designs

	σ_{acc}^2	σ_{acc}^2 per ROSC
Two-Oscillator TRNG (Fig. 1)	$2L\sigma_A^2$	$L\sigma_A^2$
ROSC TRNG with XOR tree (Fig. 18, Fig. 19)	$(M+1)L\sigma_A^2$	$L\sigma_A^2$
BFD-TRNG (Fig. 2)	$5L\sigma_A^2$	$2.5L\sigma_A^2$
M-parallel BFD-TRNG (Fig. 14)	$5ML\sigma_A^2$	$\frac{5M}{M+1}L\sigma_A^2$
Cascade BFD-TRNG (Fig. 15)	$11L\sigma_A^2$	$3.67L\sigma_A^2$
M-parallel Cascade BFD-TRNG (Fig. 17)	$11ML\sigma_A^2$	$\frac{11M}{M+2}L\sigma_A^2$

It can be seen that the BFD-TRNG has greater σ_{acc}^2 per ROSC than prior ROSC based TRNGs, which could lead to better randomness, as it accumulates a larger amount of jitter before it is sampled. Moreover, it can be seen that the σ_{acc}^2 of BFD-TRNG is 150% higher than the σ_{acc}^2 of two-oscillator TRNG. The parallel, cascade, and parallel-cascade structures of the BFD-TRNG can further improve the randomness. Note that the σ_{acc}^2 is just a rough estimate of the randomness when the TRNG is sampled.

6.3.2. Cost. We summarize the performance of different ROSC based TRNG designs in Table VIII. We measure the area and power consumptions for the 7-stage ROSC, DFF, and 10-bit counter from the test chip in 65nm, as shown in Table IX. Consequently, the cost comparisons (only considering the components) for different ROSC based TRNGs are presented in Table X. It can be seen that the BFD-TRNGs can generate more bits per sample. Furthermore, the BFD-TRNGs have less cost per bit in general, compared to prior ROSC based TRNG designs. We can further improve the performance by setting an appropriate $\Delta\mu$ as discussed in Section 4. Moreover, the parallel and the parallel-cascade structures of the BFD-TRNG can further reduce the cost per bit, as only one extra ROSC is required for each extra output. When M is large, the costs of the parallel and the parallel-cascade structures will be significantly less than prior existing ROSC based TRNG designs.

We now compare the area and power performance of the M-parallel BFD-TRNG and the 64-parallel IBM TRNG in [Liberty et al. 2013]. Since the M-parallel BFD generates $3.25M$ bits per count, for 64 parallel bits, $M = 64/3.25 \approx 20$. With $M = 64$ for IBM TRNG and $M = 20$ for BFD-TRNG, the (power)(sample period)/bit products for the two designs are given by 522.2125 and 182.2308, respectively. The (area)(sample period)/bit products for the two designs are give by 534.0625 and 285.0000, respectively. Thus, we conclude that the M-parallel BFD-TRNG has approximately 3 times power advantage and 2 times area advantage for a specified number of bits per same period, compared to the IBM TRNGs. Similar calculations show that the power and area consumptions of M-parallel cascade BFD-TRNG are only 30.9% and 45.4% of the IBM TRNG, respectively. However, we caution that the M-parallel and M-parallel-cascade BFD-TRNG results are not based on actual measurements, but are predicted from models.

Table VIII: Summary of Different ROSC based TRNG Designs

	# Bits per Sample	Sample Period	Component
Two-Oscillator TRNG (Fig. 1)	1	> 500	2 ROSCs, 1 DFF
M-parallel Two-Oscillator TRNG ([Liberty et al. 2013])	M	> 500	$(M+1)$ ROSCs, M DFFs
ROSC TRNG with XOR tree (Fig. 18)	1	$5 \sim 20$	115 ROSCs, 1 DFF†
Enhanced ROSC TRNG with XOR tree (Fig. 19)	1	$5 \sim 20$	51 ROSCs, 50 DFFs†
BFD-TRNG (Fig. 2)	3.25	500	2 ROSCs, 1 DFF, 1 Counter
M-parallel BFD-TRNG (Fig. 14)	$3.25M$	500	$(M+1)$ ROSCs, M DFFs, M Counters
Cascade BFD-TRNG (Fig. 15)	4	500	3 ROSCs, 3 DFFs, 1 Counters
M-parallel Cascade BFD-TRNG (Fig. 17)	$4M$	500	$(M+2)$ ROSCs, $(2M+1)$ DFFs, M Counters

† the cost of XOR is negligible

Table IX: Area and Power Consumptions for ROSC based TRNG Components

	Power	Normalized Power	Area	Normalized Area
ROSC	21.19μ	1	$40 \times 10\mu^2$	1
DFF	0.61μ	0.0288	$3 \times 7\mu^2$	0.0525
Counter	2.24μ	0.1057	$30 \times 10\mu^2$	0.75

Table X: Cost for Different ROSC based TRNG Designs

	Total Power	(Power)(Sample Period)/Bit	Total Area	(Area)(Sample Period)/Bit
Two-Oscillator TRNG (Fig. 1)	2.0288	> 1014.4	2.0525	> 1026.25
M-parallel Two-Oscillator TRNG ([Liberty et al. 2013])	$1.0288M + 1$	$> 514.4 + 500/M$	$1.0525M + 1$	$> 526.25 + 500/M$
ROSC TRNG with XOR tree (Fig. 18)	115.0288	$575.144 \sim 2300.576$	115.0525	$575.2525 \sim 2301.05$
Enhanced ROSC TRNG with XOR tree (Fig. 19)	52.44	$262.2 \sim 1048.8$	52.625	$263.125 \sim 1052.5$
BFD-TRNG (Fig. 2)	2.1345	328.3846	2.8025	431.1538
M-parallel BFD-TRNG (Fig. 14)	$1.1345M + 1$	$174.5385 + 153.8461/M$	$1.8025M + 1$	$277.3077 + 153.8461/M$
Cascade BFD-TRNG (Fig. 15)	3.1921	399.0125	3.9075	488.4375
M-parallel Cascade BFD-TRNG (Fig. 17)	$1.1633M + 2.0288$	$145.4125 + 253.6/M$	$1.855M + 2.0525$	$231.85 + 256.5625/M$

7. CONCLUSION AND FUTURE WORK

This paper has presented a comprehensive statistical analysis for the high-speed BFD-TRNG. The relationships of period difference of the two ROSCs, environmental noise and the counter values have been investigated. Furthermore, how the counter values affect the number of random bits per sample that we can use has also been examined. We have concluded that an appropriate frequency difference of the two ROSCs should be set based on the environmental noise to achieve higher throughput. Other aspects of the BFD-TRNG design, such as post-processing techniques, have also been explored. Based on statistical analysis results, we have proposed several alternate BFD-TRNG designs, which include the parallel structure, the cascade structure, and the parallel-cascade structure. These novel structures could achieve improved performances. Comparisons of the BFD-TRNG with other existing ROSC based TRNGs have also been conducted. We have shown that the BFD-TRNG designs have better performances from both the randomness and the cost perspectives. Future work will be directed towards improving the BFD-TRNG design by utilizing our statistical analysis results, which would also include the aspects of transistor sizing and trimming capacitors selection. Novel TRNG designs need to be fabricated and tested. Our statistical analysis can also be verified

by new silicon data. Moreover, since we are unable to include the analysis of flicker noise in this paper as it is not possible for us to collect test data at frequencies below the corner frequency of flicker noise in our current high-speed BFD-TRNG chip, we leave the analysis as a future work. The current model can be refined and improved in future efforts by embedding models of flicker noise from data collected from future fabricated chips.

8. ACKNOWLEDGMENT

The authors are grateful to all anonymous reviewers for numerous constructive comments.

This research has been supported in part by the National Science Foundation under grant number CNS-1441639 and the Semiconductor Research Corporation under contract number 2014-TS-2560.

REFERENCES

Brian J Abcunas. 2004. *Evaluation of random number generators on FPGAs*. Ph.D. Dissertation. Worcester Polytechnic Institue.

Asad A Abidi. 2006. Phase noise and jitter in CMOS ring oscillators. *IEEE Journal of Solid-State Circuits* 41, 8 (2006), 1803–1816.

Takehiko Amaki, Masanori Hashimoto, and Takao Onoye. 2013. Jitter amplifier for oscillator-based true random number generator. *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences* 96, 3 (2013), 684–696.

Ganesh K Balachandran and Raymond E Barnett. 2008. A 440-nA true random number generator for passive RFID tags. *IEEE Transactions on Circuits and Systems I: Regular Papers* 55, 11 (2008), 3723–3732.

Mathieu Baudet, David Lubicz, Julien Micolod, and André Tassiaux. 2011. On the security of oscillator-based random number generators. *Journal of cryptology* 24, 2 (2011), 398–425.

Nathalie Bochard, Florent Bernard, and Viktor Fischer. 2009. Observing the randomness in RO-based TRNG. In *Proceedings of International Conference on Reconfigurable Computing and FPGAs*. 237–242.

Holger Bock, Marco Bucci, and Raimondo Luzzi. 2004. An offset-compensated oscillator-based random bit source for security applications. In *Cryptographic Hardware and Embedded Systems (CHES)*. 268–281.

Ralf Brederlow, Ramesh Prakash, Christian Paulus, and Roland Thewes. 2006. A low-power true random number generator using random telegraph noise of single oxide-traps. In *Proceedings of IEEE International Solid-State Circuits Conference*. 1666–1675.

Marco Bucci, Lucia Germani, Raimondo Luzzi, Alessandro Trifiletti, and Mario Varanonuovo. 2003. A high-speed oscillator-based truly random number source for cryptographic applications on a smart card IC. *IEEE Trans. Comput.* 52, 4 (2003), 403–409.

Marco Bucci and Raimondo Luzzi. 2008. Fully digital random bit generators for cryptographic applications. *IEEE Transactions on Circuits and Systems I: Regular Papers* 55, 3 (2008), 861–875.

Wayne R Coppock. 2005. *A mathematical and physical analysis of circuit jitter with application to cryptographic random bit generation*. Ph.D. Dissertation. Worcester Polytechnic Institue.

Markus Dichtl. 2007. Bad and good ways of post-processing biased physical random numbers. In *Proceedings of Fast Software Encryption*. 137–152.

Markus Dichtl and Jovan Dj Golić. 2007. *High-speed true random number generation with logic gates only*. Springer.

Michael Epstein, Laszlo Hars, Raymond Krasinski, Martin Rosner, and Hao Zheng. 2003. Design and implementation of a true random number generator based on digital circuit artifacts. In *Cryptographic Hardware and Embedded Systems (CHES)*. 152–165.

Viktor Fischer, Florent Bernard, Nathalie Bochard, and Michal Varchola. 2008. Enhancing security of ring oscillator-based TRNG implemented in FPGA. In *Proceedings of International Conference on Field Programmable Logic and Applications*. 245–250.

Daniel E Holcomb, Wayne P Burleson, and Kevin Fu. 2007. Initial SRAM state as a fingerprint and source of true random numbers for RFID tags. In *Proceedings of the Conference on RFID Security*, Vol. 7.

Wolfgang Killmann and Werner Schindler. 2001. *AIS 31: Functionality Classes and Evaluation Methodology for True (Physical) Random Number Generators, Version 3.1*. Bundesamt fur Sicherheit in der Informationstechnik (BSI).

Tae-Hyoung Kim, Randy Persaud, and Chris H Kim. 2008. Silicon odometer: An on-chip reliability monitor for measuring frequency degradation of digital circuits. *IEEE Journal of Solid-State Circuits* 43, 4 (2008), 874–880.

Paul Kohlbrenner and Kris Gaj. 2004. An embedded true random number generator for FPGAs. In *Proceedings of the 2004 ACM/SIGDA 12th international symposium on Field Programmable Gate Arrays*. 71–78.

Patrick Lacharme. 2008. Post-processing functions for a biased physical random number generator. In *Proceedings of Fast Software Encryption*. 334–342.

JS Liberty, A Barrera, DW Boerstler, TB Chadwick, SR Cottier, HP Hofstee, JA Rosser, and ML Tsai. 2013. True hardware random number generation implemented in the 32-nm SOI POWER7+ processor. *IBM Journal of Research and Development* 57, 6 (2013), 4–1.

Mehrdad Majzoobi, Farinaz Koushanfar, and Srinivas Devadas. 2011. FPGA-based true random number generation using circuit metastability with adaptive feedback control. In *Cryptographic Hardware and Embedded Systems (CHES)*. 17–32.

George Marsaglia. 1996. DIEHARD: a battery of tests of randomness. *See http://stat. fsu. edu/ geo/diehard. html* (1996).

Craig S Petrie and J Alvin Connelly. 1996. Modeling and simulation of oscillator-based random number generators. In *Proceedings of IEEE International Symposium on Circuits and Systems*, Vol. 4. 324–327.

Craig S Petrie and J Alvin Connelly. 1998. A noise-based random bit generator IC for applications in cryptography. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, Vol. 2. 197–200.

Craig S Petrie and J Alvin Connelly. 2000. A noise-based IC random number generator for applications in cryptography. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 47, 5 (2000), 615–621.

Md Tauhidur Rahman, Kan Xiao, Domenic Forte, Xuhei Zhang, Jerry Shi, and Mohammad Tehranipoor. 2014. TI-TRNG: Technology Independent True Random Number Generator. In *Proceedings of the The 51st Annual Design Automation Conference on Design Automation Conference*. 1–6.

Vladimir Rozic and Ingrid Verbauwhede. 2009. Random numbers generation: investigation of narrowtransitions suppression on FPGA. In *Proceedings of International Conference on Field Programmable Logic and Applications*. 699–702.

Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, and Elaine Barker. 2001. *A statistical test suite for random and pseudorandom number generators for cryptographic applications*. Technical Report. DTIC Document.

Werner Schindler. 2003. A stochastical model and its analysis for a physical random number generator presented at CHES 2002. In *Cryptography and Coding*. 276–289.

Suresh Srinivasan, Sanu Mathew, Rajaraman Ramanarayanan, Farhana Sheikh, Mark Anders, Himanshu Kaul, Vasantha Erraguntla, Ram Krishnamurthy, and Greg Taylor. 2010. 2.4 GHz 7mW all-digital PVT-variation tolerant True Random Number Generator in 45nm CMOS. In *Proceedings of IEEE Symposium on VLSI Circuits (VLSIC)*. 203–204.

Alan Stuart and J Keith Ord. 1994. Kendalls advanced theory of statistics. Vol. I. Distribution theory. *Arnold, London* (1994).

Berk Sunar, William J Martin, and Douglas R Stinson. 2007. A provably secure true random number generator with built-in tolerance to active attacks. *IEEE Trans. Comput.* 56, 1 (2007), 109–119.

Qianying Tang, Bongjin Kim, Yingjie Lao, Keshab K. Parhi, and Chris H. Kim. 2014. True Random Number Generator Circuits Based on Single- and Multi-Phase Beat Frequency Detection. In *Proceedings of IEEE Customs Integrated Circuits Conference*.

Qianying Tang, Xiaofei Wang, John Keane, and Chris H Kim. 2013. RTN induced frequency shift measurements using a ring oscillator based circuit. In *Symposium on VLSI Technology*. T188–T189.

Carlos Tokunaga, David Blaauw, and Trevor Mudge. 2008. True random number generator with a metastability-based quality control. *IEEE Journal of Solid-State Circuits* 43, 1 (2008), 78–85.

Boyan Valtchanov, Viktor Fischer, Alain Aubert, and Florent Bernard. 2009. TRNG based on the coherent sampling. In *CryptArchi*.

John Von Neumann. 1951. Various techniques used in connection with random digits. *Applied Math Series* 12, 36-38 (1951), 1.

Knut Wold. 2011. *Security properties of a class of true random number generators in programmable logic*. Ph.D. Dissertation. Gjovik University College.

Knut Wold and Chik How Tan. 2009. Analysis and enhancement of random number generator in FPGA based on oscillator rings. *International Journal of Reconfigurable Computing* 2009 (2009), 4.

Kaiyuan Yang, David Fick, Michael B Henry, Yoonmyung Lee, David Blaauw, and Dennis Sylvester. 2014. A 23Mb/s 23pJ/b fully synthesized true-random-number generator in 28nm and 65nm CMOS. In *Proceedings of IEEE International Solid-State Circuits Conference Digest of Technical Papers*. 280–281.