TI-TRNG: Technology Independent True Random Number Generator

Md. Tauhidur Rahman, Kan Xiao, Domenic Forte, Xuhei Zhang, Jerry Shi, and Mohammad Tehranipoor
ECE Department, University of Connecticut
{tauhid, kanxiao, forte, xuz09001, zshi, tehrani}@engr.uconn.edu

ABSTRACT

True random number generators (TRNGs) are needed for a variety of security applications and protocols. The quality (randomness) of TRNGs depends on sensitivity to random noise, environmental conditions, and aging. Random sources of noise improve TRNG quality. In older or more mature technologies, the random sources are limited resulting in low TRNG quality. Prior work has also shown that attackers can manipulate voltage supply and temperature to bias the TRNG output. In this paper, we propose bias detection mechanisms and a technology independent TRNG (TI-TRNG) architecture. The TI-TRNG enhances power supply noise for older technologies and uses a self-calibration mechanism that reduces bias in TRNG output due to aging and attacks. Experiment results on $130nm$, $90nm$, and $45nm$ FPGAs demonstrate the quality of random sequences from the TI-TRNG across aging and different environmental conditions.

Categories and Subject Descriptors

SEC1 [**Hardware Security**]: Device, circuit, and architecture techniques for security

General Terms

Security, Design

Keywords

True random number generator, Tunable ring oscillator, Attacks detection, Random supply noise

1. INTRODUCTION

A random number generator (RNG) is an important security block widely used in most cryptographic applications such as one time pads, session and temporary keys, nonce, seeds, challenges for authentication, zero knowledge protocols, hardware metering, generation of primes, secure communications, secured servers and processors, VPN access, and customer-facing web access [1-8]. A quality RNG generates statistically independent and unpredictable sequences of random numbers. Compromising an RNG often means compromising an entire system.

A true RNG (TRNG) translates random physical phenomena such as thermal noise, atmospheric noise, shot noise, radio noise, flicker noise, clock jitter, phase noise, noise in a compact memory etc. into random digits [1-8]. A TRNG must have uniform statistics; non-uniform statistics due to biased random sequences help attackers to guess the random numbers. Generally, random numbers are generated by comparing two symmetric devices which possess some process variation (PV) and random inner noise. Variations in the inherent process, operating temperature, V_{DD}, and aging may bias the TRNG output by introducing large asymmetry between them. An attacker might exploit the dependence of the TRNG on supply voltage and temperature to intentionally bias the TRNG [9, 10]. In addition, the randomness of a TRNG is affected by limited PV when inner random noise source cannot provide enough source of entropy alone. Older and more mature technologies possess less PV and provides less randomness. The randomness of TRNG in such technologies can become even worse under environmental variations and different aging mechanisms. This presents a major issue for military and space applications which must typically rely on older technology nodes due to reliability concerns.

Most of the TRNGs presented in literature discuss the randomness due to process and inner random noise in lower technology nodes [1, 3, 11, 12]. Many challenges not fully addressed in prior work include: (i) insufficient PV in older and more mature technologies, (ii) robustness to aging, and (iii) potential attacks that exploit environmental conditions to bias a TRNG.

In this paper, we propose innovative techniques that overcomes each of the above challenges. In order to provide high quality entropy, extra power supply noise is induced by internal circuit switching in order to compensate for the insufficient PV in older technology nodes. It also overcomes the entropy loss due to aging by supplying continuous random noise. A self-compensation mechanism is proposed to improve randomness of TRNG across time, operating conditions, and attacks. In addition, a bias monitoring unit is proposed to detect attacks (manipulation of environmental conditions). We call our overall approach Technology Independent TRNG (TI-TRNG) because it automatically overcomes attacks and lack of randomness regardless of the technology used. Results from the NIST test suite [13] show

DAC '14, June 01 - 05 2014, San Francisco, CA, USA
Copyright 2014 ACM 978-1-4503-2730-5/14/06..$15.00
http://dx.doi.org/10.1145/2593069.2593236 .

that the proposed TI-TRNG possesses sufficient randomness over adverse environmental conditions, aging, and several technology nodes (old, mature, current) while only costing $\sim 14.62\%$ area overhead compared to a conventional TRNG.

The rest of the paper is organized as follows: In Section 2, we discuss the background of TRNG and the motivation of our proposed TI-TRNG. Attack/bias detection techniques are addressed in Section 3. In Section 4, we discuss the architecture and working principle of the proposed TI-TRNG. The experimental results and analyses are shown in Section 5. We conclude the paper in Section 6.

2. BACKGROUND AND MOTIVATION

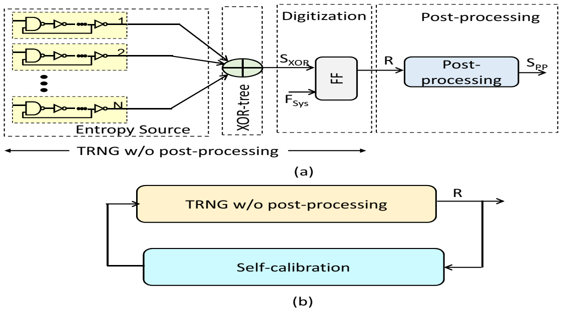


Figure 1: (a) Conventional RO-based TRNG with post-processing (b) TRNG with calibration.

A basic TRNG consists of a source of randomness (entropy) and a randomness extractor. Metastability of logic cells and timing jitter of Ring Oscillators (ROs) are the most common sources of entropy for a TRNG [1-9]. Simplicity of implementation and entropy collection have made the RO-based TRNG most popular. Figure 1(a) shows a conventional RO-based TRNG where the RO outputs are combined by a XOR-tree. The output of the XOR-tree, S_{XOR}, is sampled constantly by a synchronous D flip-flop driven by the system clock to convert RO jitter into a random digital sequence. Jitter in this case represents the deviation from ideal RO behavior caused by random process variation and temporal variations such as random physical noise, environmental variations, and aging. Without jitter, the ROs will possess almost identical phase and the output of XOR-tree would be almost constant which is undesirable. Among the variations, only the random physical noise (thermal noise, atmospheric noise, shot noise, radio noise, flicker noise, etc.) improves the randomness/entropy of the bitstream output by the TRNG. Large process variations can improve TRNG quality by increasing susceptibility to this random noise. The remaining variations reduce the randomness of the bitstreams by deterministically biasing the jitter.

Figure 2 illustrates how the output of XOR-tree, S_{XOR}, and bitstreams, R, is affected by process variations, environmental conditions and aging. In the figure, $'1'$ and $'0'$ represent deterministic output; **c** represents a critical bit that favors either 1 or $'0'$ with high probability; and 'r' represents random output (equal probability of $'1'$ and $'0'$). The gray areas of the waveforms represent jitter in S_{XOR} and uncertainty in the output R. Let us assume the S_{XOR} is sampled at points S1, S3, and S5 shown in the figure. At nominal conditions (Figure 2(a)), the output corresponding to these sampling points is **'1cr'**. The $'1'$ at the beginning is a result of large process variations that cause one of the ROs to be much faster than the other. The **c** will largely favor a $'0'$ at the output while the last bit may be either $'0'$ or $'1'$ due to jitter. Figure 2(b) illustrates the impact of decreasing V_{DD}, increasing operating temperature, or aging. Each case causes a decrease in the frequency of the ROs resulting in a deterministic output sequence **'100'**. Figure 2(c) highlights the impact of increasing V_{DD} and/or decreasing temperature. Both increase the RO frequency resulting in another deterministic bitstream **'101'**. Figures 2(b,c) illustrate how altering the voltage supply and operating temperature or aging the chip can reduce the randomness in the TRNG output. An attacker can exploit this deterministic behavior by manipulating the device's operating environment in order to break the security provided by the TRNG.

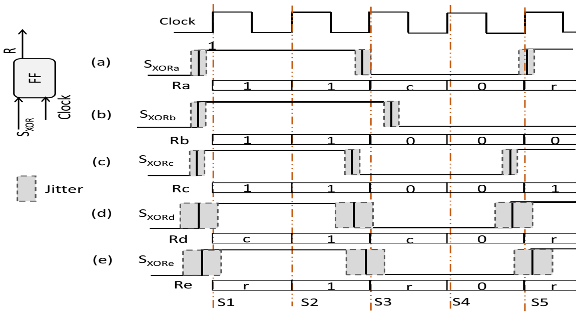


Figure 2: Deterministic behavior for (a) conventional RO-based TRNG, (b) decreasing frequency due to aging and environmental variations, (c) increasing frequency due to operating conditions. Overcoming bias by (d) widening jitter and (e) adjusting delay of RO (tunable RO).

As touched upon above, the entropy source has an interesting relationship with respect to PV. On the one hand, large process variations can make one RO much faster than another. On the other hand, PV plays a dominant role in the random jitter experienced by the ROs (gray areas shown in Figure 2). Random dopant fluctuation, sub-wavelength lithography, rapid-thermal anneal, high-stress capping layers etc. are the main sources of process variations [14, 15]. Number of dopant atoms have been decreasing with semiconductor scaling. Hence, in older technology nodes, PV is limited due to large number of dopant atoms resulting in less jitter and therefore lower quality TRNG bitstreams.

There are several major issues highlighted by the above discussion. First, attackers can bias TRNG output by manipulation of the voltage supply and operating temperature. Second, unavoidable IC aging will reduce the randomness of TRNG bitstreams. Third, mature technologies may not be sensitive enough to random noise resulting in low-quality TRNG. There have been two basic approaches for addressing the above issues: post-processing and calibration. Both are illustrated in Figure 1. In Figure 1(a), post-processing mechanisms such as XOR function or hash function for privacy amplification is applied to make the TRNG output more random [16]. This can improve upon all the above areas, but comes at the cost of high area and power overheads [16]. Calibration techniques, shown in Figure 1(b), use feedback from the TRNG output to remove deterministic effects. Both coarse grain and fine grain calibration techniques have been proposed in [2] for metastable based

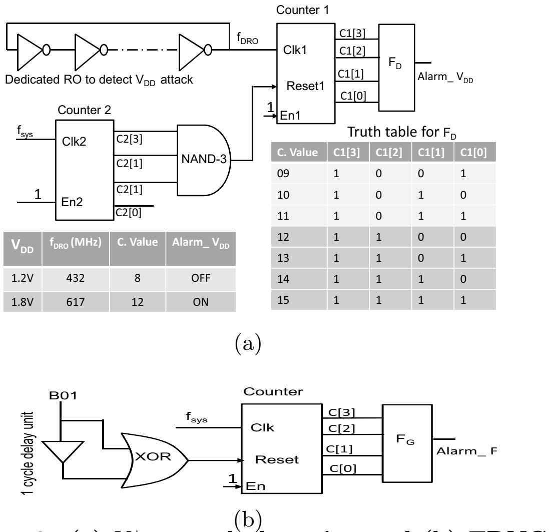
TRNG [1, 2]. There is no work on calibration technique for conventional RO-based TRNG in literature.

In this paper, we propose a new TRNG architecture that is an instance of the calibration approach shown in Figure 1(b). First, we propose two supplementary TRNG modules that detect if the environmental conditions are being manipulated by an attacker. If there is a bias present in the TRNG output (due to increase/decrease of V_{DD} or temperature), the modules will alert the system. The shortcoming with the detection approach on its own is that it cannot compensate for attacks, aging, and/or process variation that bias the TRNG output. Thus, we also propose a new calibration-based architecture for RO-TRNG that improves the quality (randomness) of TRNG output in two ways: (1) by increasing the sensitivity of the TRNG circuitry to random noise; (2) by automatically detecting any bias (using the modules discussed above) and then tuning the TRNG circuitry to "un-bias" the TRNG output. We call our approach Technology Independent TRNG (TI-TRNG) because it automatically overcomes attacks and lack of randomness regardless of the technology used.

3. ATTACK/BIAS DETECTION

As discussed in Section 2, an attacker might attempt to bias a TRNG by manipulating V_{DD} or temperature. In order to explicitly detect such attacks and raise a red flag, we propose two attack detection mechanisms. The first technique detects the attack based on raising V_{DD}. Attackers can increase the power supply to a certain value, $V^{\star}_{DD}$, in order to decrease the sensitivity of inner random noise and bias the TRNG output. In practice, the $V^{\star}_{DD}$ value resulting in deterministic output can be determined by observing the NIST p-value at different values of V_{DD}. Figure 3(a) shows the mechanism to detect the $V^{\star}_{DD}$ attack which consists of four components: (i) A dedicated RO; (ii) A small counter (counter-1) that counts the oscillations of the RO; (iii) another counter (counter-2) that resets counter-1 every so often; (iv) A logical function F_D that compares the count with the count we expect from an unbiased RO. Put simply the frequency of a dedicated RO, f_{DRO}, is monitored by counter-1. If V_{DD} is increased beyond $V^{\star}_{DD}$, the Ro's frequency will increase resulting in higher counter value. We pre-compute a threshold f_{thres} that corresponds to the RO frequency when the voltage supply is set to $V^{\star}_{DD}$. If counter-1's value exceeds f_{thres}, then the output of $F_D = and(C1[3], C1[2]) = 1$ and an alarm bit $Alarm\_Vdd$ is raised.

Our second approach detects more general attempts to bias the output (e.g., by manipulating temperature) by directly monitoring the TRNG output. Specifically, if the total number of consecutive zeros (ones) exceeds a threshold T_{bias}, then the TRNG is biased towards zero (ones). Our detection scheme is shown in Figure 3(b). It consists of a consecutive bit detector, a counter, and a detection calculation F_G. The consecutive bit detector compares the current and prior bit of the TRNG output. If they are the same, it assumes there is a small bias. In this case, the scheme increments the counter (which counts how many consecutive bits are zero or one). If the consecutive bits differ, the counter is reset. The function F_G is a logical function that computes if the counter value exceeds the threshold T_{bias} and turns on $Alarm\_F$.



V_{DD}	f_{DRO} (MHz)	C. Value	Alarm_ V_{DD}
1.2V	432	8	OFF
1.8V	617	12	ON

Truth table for F_D

C. Value	C1[3]	C1[2]	C1[1]	C1[0]
09	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

(a)

(b)

Figure 3: (a) $V^{\star}_{DD}$ attack detection and (b) TRNG failure detection.

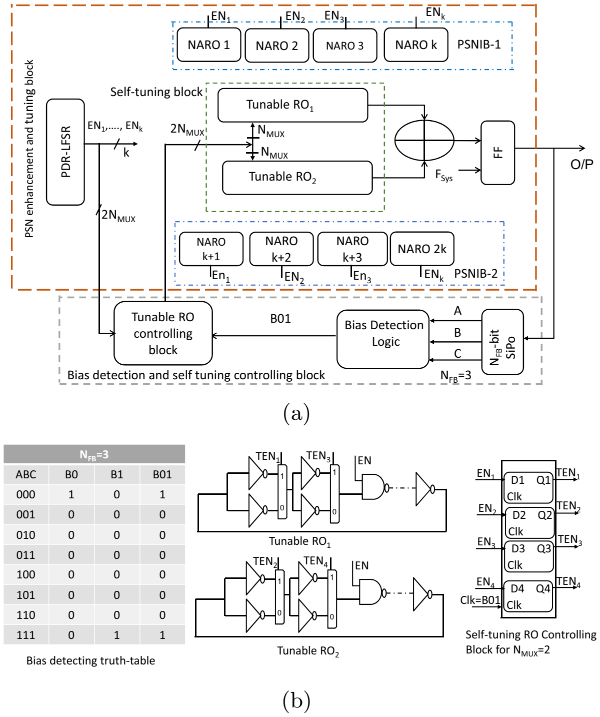
4. TECHNOLOGY INDEPENDENT TRNG

The detection mechanisms discussed in the prior section cannot be used to improve the TRNG quality; they are used to detect attacks only. In this section, we propose a new technology independent TRNG (TI-TRNG) that exploits power supply noise (PSN), induced randomly to the circuit, together with a self-calibration mechanism to increase randomness and reduce bias in bitstreams. We call it "technology-independent" because of its effectiveness in generating true random number for new as well as old technology nodes. The basic ideas are illustrated in Figures 2(d,e). The additional PSN widens the jitter and can partially reduce the impact of low PV, attacks based on V_{DD} and temperature manipulation, and aging. For example, in Figure 2(d), the output becomes **'ccr'** which is better than the deterministic **'101'** shown in Figure 2(c). Self-calibration is used to "unbias" the RO outputs by systematically tuning the RO oscillation frequencies based on feedback from the TRNG output. For example, in Figure 2(e), the RO frequencies are increased to reduce the impact of low V_{DD} and high temperature on the RO output resulting in **'rrr'** at the output. Together, the PSN and self-calibration provide a constant random source of entropy to combat against environment variations, aging and attacks based on V_{DD} and environment variations. Hence, an attacker will not be able to influence the TRNG output and the systems dependent on the TRNG will remain secure in the field even though it experience degradation due to NBTI and HCI [20, 21].

Figure 4(a) presents the proposed RO-based TI-TRNG. It consists of (1) ROs with tunable delay; (2) a linear feedback shift register (LFSR); (3) two power supply noise inducing blocks (PSNIB-1 and PSNIB-2) that are randomly enabled/disabled by the output of the LFSR; (3) one self-calibration controlling block that checks the bias from TRNG output and takes appropriate steps to remove bias by tuning the RO delay. Figure 4(b) shows the bias detection technique and the tuning mechanism. The detailed architecture and principles are discussed in the subsections below.

4.1 PSN enhancement and tuning block

In RO-based TRNG, jitter from RO is XORed and then sampled by a synchronous FF. Internal random noise with



N_{FB}=3			
ABC	B0	B1	B01
000	1	0	1
001	0	0	0
010	0	0	0
011	0	0	0
100	0	0	0
101	0	0	0
110	0	0	0
111	0	1	1

Bias detecting truth-table

Figure 4: (a) Proposed Technology Independent TRNG architecture and (b)detection and tuning mechanism.

PV brings more randomness. Power supply noise acts like substrate noise and enhances the jitter and hence the randomness. Besides, PSN widens the timing jitter and improves the filling rate which improves the source of entropy and randomness in TRNG output significantly [19]. In order to increase the random PSN, some smaller ROs have been introduced. These ROs are called noise augmenting ring oscillators (NAROs). The NAROs are kept close to the tunable ROs. The random noise enhancement unit consists of two symmetric PSNIB-1 and PSNIB-2 blocks that have equal number of symmetric NAROs.

However, constant switching might create deterministic noise and would bias the number generator. To randomize the switching and hence the PSN, a partial dynamic LFSR (PD-LFSR), [17], is used to enable the NAROs which updates the seed dynamically. The seed is updated by taking feedback from TRNG output. Random numbers generated by PD-LFSR activate or deactivate the NAROs and make the switching random. Mirror NAROs, NARO 1 and NARO k+1, are turned ON or OFF at the same time.

The delay of logic gates is directly impacted by power supply noise. In modern compact IC, transition in one gate affects the power supply of other gates in that network with close proximity [18]. [19] provides a mathematical expression about how PSN introduces jitter in an inverter. It has also shown in [18] that more ring oscillators (NARO in our design) introduce more PSN. So, in older and matured technology nodes, where PV is not sufficient alone, NAROs provide the enhanced random PSN to improve the quality of TRNG bitstreams by widening the jitter.

Entropy is wasted during extraction when jitter from two ROs overlap (aka. interlocking). Instead of conventional RO, a tunable RO is proposed to avoid interlocking. Increase in the speed difference between two RO's due to PV might bias the TRNG output. The tuning mechanism tunes the tunable RO in order to remove any bias, as illustrated in Figure 2(e).

4.2 Bias detection and self-calibration

In order to remove/reduce post-processing implementation cost, a self-calibration technique is proposed to maintain a minimum level of security as randomness and entropy of a TRNG are affected by adverse operating environments, aging in the field, or any intentional attack. Figures 4(a) and (b) together show the complete bias detection and calibration mechanism in order to remove bias.

An N_{FB}-bit series to parallel (SiPo) register is used to store N_{FB} consecutive bits from the random TRNG output bitstreams as shown in Figure 4(b). The N_{FB} consecutive bits can be biased to $'0'$ or $'1'$ or can be unbiased. Depending on the value of N_{FB} a logic function, B01, is required to detect the bias as shown in Figure 4(b). For example, B01=$\bar{A}\bar{B}\bar{C}$+ABC is used to detect whether the consecutive three bits are biased or not for $N_{FB} = 3$. $B0 = 1$ or $B1 = 1$ mean the consecutive N_{FB} bits are biased to $'0'$ or $'1'$, respectively. $B01 =' 1'$ means the consecutive N_{FB} bits are biased to either $'0'$ or $'1'$.

Bias detector detects the bias and changes the delay path of two tunable ROs to avoid interlocking and large speed difference between them. Figure 4(b) shows the tunable RO which consists of odd number of inverters and N_{MUX} MUXs between some of them. The MUXs in the tunable RO are controlled by the same PD-LFSR which controls the NAROs. The total number of MUXs, N_{MUX}, is an important factor to control the effectiveness of tuning. Put simply, the more MUXs we have, the greater our ability to tune the RO oscillation frequencies. However, this comes at a tradeoff with area overheads.

The righthand side of Figure 4(b) also illustrates the controlling block which adjusts the delay of two tunable ROs in order to remove the bias. MUXs in the tunable RO are controlled by a simple tuning control unit. The controlling unit consists of $2N_{MUX}$ latches which are controlled by $B01$ from the bias detection unit. Any $2N_{MUX}$ bit from PD-LFSR output are passed through tuning control unit. B01 is used as clock of each latches. A latch is transparent if its clock is high ($B01 =' 1'$) and holds its previous state if the clock is low ($B01 =' 0'$). For $B01 =' 1'$ (i.e., bias detected), the MUXs are controlled by latches' present states and change the delay path until bias is removed. When there is no bias ($B01 =' 0'$), latches hold their previous states to maintain a high quality TRNG bitstream.

5. SIMULATION RESULTS AND ANALYSIS

In order to verify the effectiveness of the proposed TI-TRNG, we implemented it along with a conventional RO-based TRNG [4] in three FPGAs from newer to older technology nodes: Spartan-6 (45nm), Spartan-3E (90nm), and Virtex II (130nm). The parameters of the TRNGs were as follows: (i) we used N_{FB} equal to 4 for the bias detection mechanism; (ii) an 8-bit PD-LFSR was used for both the conventional TRNG and TI-TRNG; (iii) 20 ROs were used to implement the conventional TRNG and 16 NAROs and 2 tunable ROs were used for TI-TRNG. Aside from collecting results from different FPGAs, we also conducted experiments at different operating conditions (which can also

Table 1: NIST Test Results at different operating conditions for SPARTAN-3E (90nm)

Op. V_{DD}	1.2V						1.4V		1.8V	
NIST Tests	0°C		25°C		70°C		25°C			
	Con.	TI	Con.	TI	Con.	TI	Con.	TI	Con.	TI
Freq.	0.02	0.54	0.28	0.74	0.09	0.67	0.18	0.64	**0.00**	0.44
Block Freq.	0.10	0.69	0.46	0.86	0.14	0.67	0.23	0.78	0.13	0.42
Cum. Sums	0.17	0.65	0.52	0.84	0.19	0.71	0.25	0.73	0.10	0.44
Runs	0.13	0.58	0.22	0.61	0.07	0.45	0.15	0.54	**0.00**	0.41
Longest Run	0.14	0.49	0.49	0.53	0.24	0.41	0.41	0.41	0.10	0.27
Rank	0.05	0.55	0.25	0.67	0.07	0.49	0.03	0.43	**0.00**	0.28
FFT	0.09	0.49	0.20	0.6	0.09	0.38	0.07	0.32	0.01	0.21
Overlap. Template (OT)	**0.00**	0.41	0.11	0.48	0.08	0.31	0.08	0.33	**0.00**	0.24
Non-OT	0.05	0.48	0.24	0.67	0.12	0.41	0.18	0.52	**0.00**	0.34
Serial	**0.00**	0.39	0.29	0.41	**0.00**	0.29	0.11	0.34	0.03	0.25
Lin. Complexity (LC)	0.03	0.46	0.27	0.51	0.07	0.37	0.22	0.43	0.04	0.30
Average	0.07	0.52	0.30	0.62	0.11	0.5	0.17	0.49	0.041	0.33

Table 2: NIST Test Results for different V_{DD}s for SPARTAN-6 (45nm) **and Virtex-II** (130nm) **at** 25°**C.**

Technology	SPARTAN-6 (45nm)				VIRTEX-II (130nm)			
Op. cond.	1.2V		1.8V		1.2V		1.8V	
	Con.	TI	Con.	TI	Con.	TI	Con.	TI
Frequency	0.37	0.77	**0.00**	0.51	0.11	0.38	**0.00**	0.41
Block Frequency	0.54	0.89	0.16	0.46	0.27	0.43	0.11	0.31
Cumulative Sums	0.49	0.82	0.12	0.56	0.31	0.56	**0.00**	0.20
Runs	0.23	0.64	**0.00**	0.46	0.11	0.43	0.01	0.22
Longest Run	0.52	0.56	0.11	0.31	0.19	0.37	0.05	0.21
Rank	0.29	0.71	**0.00**	0.29	0.21	0.42	**0.00**	0.25
FFT	0.27	0.77	0.06	0.22	**0.00**	0.41	**0.00**	0.22
OT	0.19	0.55	**0.00**	0.34	0.10	0.30	**0.00**	0.24
Non-OT	0.21	0.65	**0.00**	0.30	**0.00**	0.37	**0.00**	0.30
Serial	0.33	0.49	0.10	0.29	0.17	0.23	**0.00**	0.19
LC	0.30	0.57	0.10	0.30	0.13	0.27	**0.00**	0.21
Average	0.34	0.67	0.06	0.37	0.16	0.37	0.01	0.25

represent potential attacks). Temperature and V_{DD} were varied from 0 °C to 70 °C and from 1.2V to 1.8V respectively. We also aged the FPGAs by applying stress at excess supply voltage (1.8V) and excess temperature (100°C) for 5 hours using a thermal stream system. In all our experiments, we collected output sequences containing 5 million bits from each TRNG and then measured the randomness of the TRNG bitstreams by using the NIST Test Suite [13], a common standard in the literature. Each NIST test provides a p-value for the bit sequences. A TRNG passes a test if the associated p-value exceeds a certain confidence level, 0.01 [13].

We started by experimenting with the SPARTAN-3E FPGA (90nm) at different operating conditions. Table 1 shows the NIST p-value for both conventional RO-based TRNG (Con.) and proposed TI-TRNG (TI). Column 1 shows the different tests run and reported by the NIST Test Suite. Columns 2-7 shows the results when V_{DD}=1.2V. Under this condition, conventional TRNG and TI-TRNG (TI) were tested under three different temperature points, T=0°C, 25°C, and 70°C, respectively. More results were collected when V_{DD}=1.4V and 1.8V, both when T=25°C. The last row shows the percentage change of the average p-value under different conditions with respect to $p - value_{@1.2V,25°C}$. Bold entries in the table indicate instances where the NIST test was failed (p-value less than 0.01) by the TRNG. At nominal conditions ($1.2V, 25°C$), both TRNGs pass the NIST tests. However, the average p-value for the conventional TRNG is only .30 while the the TI-TRNG's is .62. The enhanced PSN increases the randomness in the TI-TRNG so this results is expected. At T=0°C and 70°C, the TRNG output is biased which reduces the randomness of the bitstreams. While the p-values of the conventional TRNG and TI-TRNG both decrease as a result, the % change is much smaller for TI-TRNG due to the bias detection and tuning mechanisms that compensate for the bias. As the voltage increases to 1.4 and 1.8 volts, the ROs which are XORed in the TRNGs experience different rates of changing speed due to V_{DD} variation which reduces randomness by causing phase interlocking or large speed difference between them. Once again, the rate of decrease in p-values for the proposed TI-TRNG is much less than the conventional TRNG. At 1.4 volts, both TRNGs still pass all NIST tests, but at 1.8 volts there is too much bias in conventional TRNG and it fails 5 out of 11 NIST tests. The TI-TRNG passes all NIST tests because of the NAROs in its design that provide a much stronger entropy source that nullifies the effect of frequency shift from increase in V_{DD}.

We performed similar experiments on newer and older technology nodes. Specifically, results at nominal conditions and 1.8 volts are shown in Table 2 for SPARTAN-6 (45nm) and Virtex-II (130nm) FPGAs. Since the SPARTAN-6 is a newer technology node, the TRNGs have larger process variations and should be more sensitive to power supply noise at nominal conditions. This is supported by the average p-value in the last row of Table 2 which is greater than in the SPARTAN-3E FPGA's average shown in Table 1. The exact opposite is true for the Virtex-II (130nm) which is the more mature technology and therefore has less PV. Both TRNGs implemented in Virtex-II have lower p-values on average compared to SPARTAN-6 (45nm) and SPARTAN-3E (90nm). The conventional TRNG even fails two tests at nominal conditions due to lack of entropy at 130nm. The TI-TRNG, on the other hand, does not fail because of the additional random noise provided by the NAROs. As the supply voltage increases, the conventional TRNG design fails 5 out of 11 and 8 out of 11 NIST tests for the SPARTAN-6 and Virtex-II respectively. It makes sense that the performance of the Virtex-II is worse because of the lower process variations at 130nm. The TI-TRNG still passes all NIST tests for the Virtex-II even at 1.8 volts.

Due to NBTI, the threshold voltage increases over time which also alters the frequency of the ROs in the TRNGs and results in less randomness. Agings results for both conventional TRNG and TI-TRNG are presented in Table 3. Comparing Tables 1-3 shows that the average p-value goes down significantly due to aging for a conventional TRNG (41% for Spartan-6, 60% for Spartan-3E, and 75% for Virtex II). Because of calibration technique and PSN enhanced design, TI-TRNG passes the NIST tests for aging when conventional TRNG fails. The TI-TRNG passes aging test for all three technologies and the decrease in p-value is much lower than conventional TRNG (24% for Spartan-6, 41% for Spartan-3E, and 35% for Virtex II). Hence, we conclude that the enhanced PSN and tuning of ROs provide strong randomness over aging in the TRNG.

Our last set of results show the area associated with conventional TRNG, TI-TRNG, and the conventional TRNG combined with our attack detection modules. There are two parameters in our TI-TRNG that might influence the area overheads: N_{FB} and N_{MUX}. N_{FB} is the number of consecutive bits used in the bias detection and calibration. N_{MUX}

Table 3: Aging analysis for different technologies

NIST Tests	Spartan-6 ($45nm$)		Spartan-3E ($90nm$)		Virtex-II ($130nm$)	
	Con.	TI	Con.	TI	Con.	TI
Freq.	0.12	0.56	0.12	0.48	**0.00**	0.27
Block Freq.	0.31	0.51	0.17	0.43	0.09	0.29
Cum. Sums	0.34	0.49	0.22	0.41	0.11	0.31
Runs	0.19	0.41	0.11	0.37	**0.00**	0.27
Longest Run	0.27	0.44	0.21	0.39	0.09	0.29
Rank	0.19	0.60	0.14	0.51	0.07	0.34
FFT	0.19	0.44	**0.00**	0.36	**0.00**	0.27
OT	0.18	0.43	0.08	0.31	**0.00**	0.21
Non-OT	0.24	0.67	0.16	0.51	0.10	0.29
Serial	**0.00**	0.56	**0.00**	0.31	**0.00**	0.19
LC	0.14	0.47	0.11	0.33	**0.00**	0.19
Average	0.20	0.51	0.12	0.40	0.04	0.24

Table 4: Area in # FPGA slices for conventional TRNG, conventional TRNG with attack detector, and TI-TRNG at different N_{FB} and N_{MUX} in Spartan-3E.

		N_{MUX}			Con.	Con. + Attack Detection
		2	4	8	-	
N_{FB}	3	360	386	431	342	360
	4	366	392	437	342	360
	12	375	401	446	342	360

represents the number of MUXs in the tunable ROs. The area in number of slices for each TRNG is shown in Table 4. As the N_{FB} and N_{MUX} increase, the area required by the TI-TRNG also increases. In the worst and best cases, the area overhead (compared to conventional TRNG) was 5.26% and 30.99%. Note that the results in Tables 1-tab:aging are given for $N_{FB} = N_{MUX} = 4$ which results in only a 14.62% area overhead. Finally, for the attack detection combined with conventional TRNG, the area overhead was 5.26%.

6. CONCLUSION

In this work, we have proposed a technology independent TRNG which provides high quality TRNG output across environmental conditions, attacks, and device aging. The experiment results show that the proposed TI-TRNG also exhibits sufficient randomness for matured and older technology which may be used in military applications. The bias detection unit and possible attack detection have increase the strength of security with a minor tradeoff in device area. In future work, we shall extend the proposed TI-TRNG to other non-RO based TRNGs.

7. ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation under grant CNS-0844995 and a grant from Comcast.

8. REFERENCES

[1] M. Majzoobi et al., “FPGA-Based True Random Number Generation Using Circuit Metastability with Adaptive Feedback Control,” Cryptographic Hardware and Embedded Systems, pp. 17-31, 2011.

[2] S. Srinivasan, et. al, “2.4GHz 7mW all-digital PVT-variation tolerant true random number generator in 45nm CMOS,” VLSI Circuit, pp. 203-204, 2010.

[3] V. Fischer et al., “True random number generator embedded in reconfigurable hardware,” Cryptographic Hardware and Embedded Systems, pp. 415-430, 2002.

[4] B. Sunar et al., “A Provably Secure True Random Number Generator with Built-In Tolerance to Active Attacks,” IEEE Trans. Comp., vol. 58, pp. 109-119, 2007.

[5] C. Tokunaga et al.,“A True Random Number Generator with a Metastability-Based Quality Control,” Proc. IEEE Int. Solid-State Circuits Conf., Digest of Technical Papers, 2007.

[6] M. Bucci et al., “A high-speed oscillator-based truly random number source for cryptographic applications on smart card IC,” IEEE Trans. Comput., vol. 52, pp. 403-409, 2003.

[7] D. Schellekens, et al., “FPGA vendor agnostic TRNG,” in Proc. 16th Int. IEEE Conf. Field Programmable Logic and Applications, pp. 139-144, 2006.

[8] B. Sunar, “True Random Number Generators for Cryptography,” In: Cryptographic Engineering. Springer, Heidelberg, 2009.

[9] A. Rukhin et al., “Improving the Robustness of Ring Oscillator TRNGs,” J. ACM Trans. Reconfigurable Technol. Syst., pp. 1-30, 2010.

[10] V. Fischer, “A Closer Look at Security in Random Number Generators Design,” Third International Workshop, COSADE, pp. 167-182, 2012.

[11] D. Holcomb, W. Burleson, and K. Fu, “Power-Up SRAM State as an Identifying Fingerprint and Source of True Random Numbers,” IEEE Transactions on Computers, vol. 58, pp. 1198-1210, 2009.

[12] S. Srinivasan et al., “A 4 Gbps 0.57 pJ/bit process-voltage-temperature variation tolerant all digital true random number generator in 45 nm CMOS,” in 22nd IEEE Int. Conf. VLSI Design, 2009.

[13] A. Rukhin et al., “A statistical test suite for random and pseudorandom number generators for statistical applications,” NIST Special Publication in Computer Security, pp. 800-822, 2001.

[14] S. Borkar, “Designing Reliable Systems from Unreliable Components: The Challenges of Transistor Variability Degradation,” IEEE Micro, vol. 25, no. 6, pp. 10-16, 2005.

[15] K. Kuhn et al., “Process technology variation,” IEEE Trans. Elec. Devices, vol. 58, no. 8, pp.2197 -2208, 2011.

[16] S. Kwok et al., “A Comparison of Post-Processing Techniques for Biased Random Number Generators,” International Workshop, WISTP, pp. 175-190, 2011.

[17] C. Krishna et al., “Achieving High Encoding Efficiency With Partial Dynamic LFSR Reseeding,” ACM Trans. Des. Auto. Elec. Syst., vol. 9, no. 4, pp. 500-516, 2004.

[18] X. Zhang et al., “Detection of trojans using a combined ring oscillator network and off-chip transient power analysis,” J. Emerg. Technol. Comp. Sys., pp. 1-20, 2013.

[19] A. Strak et al., “Analysis of timing jitter in inverters induced by power-supply noise,” Proc. IEEE Int. Conf. Design Test Integr. Syst. Nano. Tech., pp. 52-56, 2006.

[20] S. Krishnappa et al., “Incorporating Effects of Process, Voltage, and Temperature Variation in BTI Model for Circuit Design,” IEEE Latin American Symp. on Circuits and Syst., pp. 236-239, 2010.

[21] W. Wang et al., “The impact of NBTI effect on combinational circuit: modeling, simulation, and analysis,” IEEE TVLSI, vol. 18, no. 2, pp. 173-183, 2010.