Enhancing Noise Sensitivity of Embedded SRAMs for Robust True Random Number Generation in SoCs

M. Tauhidur Rahman*, Domenic Forte*, Xiaoxiao Wang†, and Mark Tehranipoor*
*Dept. of Electrical and Computer Engineering, University of Florida Gainesville, FL
Dept. of Electrical and Computer Engineering, Beihang University, China
Email: rahman.tauhid@ufl.edu*, dforte@ece.ufl.edu*, wangxiaoxiao@buaa.edu.cn†, and tehranipoor@ece.ufl.edu*

Abstract **True random number generators (TRNGs) play an important role in trusted execution and communication for modern system on chips (SoCs). Building a TRNG in today's SoCs is complex and often challenging because it must have uniform statistical characteristics at any operating condition and workload, and in hostile environments over the entire system lifetime. The startup outputs of SRAM cells, another vital component in SoCs, have been used to generate random numbers and/or unique keys. However, the quality of existing SRAM-based TRNGs is limited due to limited amount of entropy which also can be manipulated by operating voltage or temperature. Another disadvantage of the existing SRAM-based TRNG is that the system requires power off and on to obtain random numbers which hampers the system performance. In this paper, we propose a noise sensitive embedded SRAM (NS-SRAM) based TRNG that reliably provides unpredictable random numbers at high rates regardless of the operating conditions. We design a noise sensitive SRAM and propose a technique that utilizes the existing power-management scheme to obtain random numbers. We evaluate the quality of NS-SRAM based TRNGs for 90*nm*, 45*nm*, and 32*nm* technology nodes. The proposed NS-SRAM based TRNG is $\sim 275X$ faster and $\sim 432X$ more area efficient (excluding post-processing overhead) than existing SRAM-based TRNGs.**

I. Introduction

Modern SoCs consist of a large number of sensitive assets that need to be protected from unauthorized access and malicious attacks or communications [2]. Multiple design blocks may be affected by the security policies because of the involvement of subtle interactions among hardware, firmware, OS kernel, and applications [1], [2]. It requires end-to-end, layered security beginning at the device level to protect a device/system from all of these potential attacks. Cryptographic hardware unit is used to protect critical assets of a SoC from many form of attacks. It also assures that the code running in the board is authentic [1], [2], [4]. The major components of a modern cryptographic hardware unit include Physical Unclonable Functions (PUFs), TRNG, accelerator of cryptographic algorithms (i.e., hardware implementation of cryptograpich algorithms such as RSA, AES, SHA, etc.) [1], [2], [4]. PUFs are used for authentication and key generation [5]. TRNGs are used to generate a key to be used in cryptographic algorithms for secure communication and privacy [5].

Security and privacy (i.e., the effectiveness of cryptographic hardware unit) rely on encryption, *whose effectiveness depends on the quality of random numbers*. A TRNG translates a physical entropy source such as thermal noise, atmospheric noise, shot noise, radio noise, flicker noise, chaos etc. into a nondeterministic random bitstream. TRNG is widely used in any real world security and trusted infrastructures. Besides algorithm, protocol, and key management system of cryptographic hardware unit, random numbers also play an important role for IP/IC piracy protection, anti-counterfeit mechanisms, countermeasures against side channel attacks, generating session keys, one time pads, random seeds, nonces, etc. [1], [3], [11] [14], [21]. Many of such applications require lightweight implementation, with necessary tuners and processing blocks that control the TRNG quality and throughput throughout its lifetime.

An SoC usually includes microprocessor, memory and peripherals. SRAM, a volatile memory, is used to store large quantities of information temporarily for high speed performance. It has been reported that more than 70% active area will be consumed by integrated memories by 2017, most of it being SRAM [1]. The start-up outputs of SRAM cells have been used to generate a random number or unique key by utilizing the unavoidable process variation during fabrication [5], [7], [8]. SRAM-based key/random number generation has been a popular choice for resource constrained hardware platforms due to a variety of reasons. Most notably, SRAM is commonly available in most of the SoCs and therefore does not require additional hardware. However, SRAM-based TRNG requires additional post-processing scheme for better quality of random numbers due the limited amount of entropy in a SRAM memory [5], [7], [8].

In this paper, we propose a noise sensitive embedded SRAM (NS-SRAM) based TRNG for modern SoC security. Our main contributions include:

- We modify regular 6T SRAM cells for higher quality random numbers by making them very sensitive to noise. The results show that NS-SRAM based TRNG (TRNG with the NS-SRAM cells) offers much more entropy than the regular 6T SRAM-based TRNG with much lower area overhead.
- We present a method of obtaining random numbers from dedicated NS-SRAM unit using power-saving scheme presented in [20].
- We show the robustness of our proposed TRNG in 90*nm*, 45*nm*, and 32*nm* technology nodes. Our proposed NS-SRAM based TRNG is 13*X* more random than conventional SRAM-based TRNG for 90*nm* technology node at room temperature.
- We also compare the area and throughput of our proposed NS-SRAM based TRNG with existing SRAM-based TRNGs. The results show that our proposed NS-SRAM based TRNG is $\sim 275X$ faster and $\sim 432X$ more area efficient than existing SRAM-based TRNG systems.

The rest of the paper is organized as follows. In Section II, we briefly discuss existing SRAM-based TRNGs' working principle and the impact of aging and environmental variations on TRNG outputs. We conclude Section II with the motivations of our proposed NS-SRAM based TRNG. In Section III, we present our proposed NS-SRAM based TRNG, where we describe the modification of SRAM cells and the method to obtain random numbers from dedicated NS-SRAM unit. We present the simulation results of our proposed design for different technology nodes to evaluate the technology independence of our proposed NS-SRAM based TRNG in Section IV. We also present possible attacks on the proposed NS-SRAM based TRNGs and the future directions of our work. We conclude our work in Section V.

II. BACKGROUND AND MOTIVATION

A. True Random Number Generators

TRNGs are hardware modules that produce random bits based on the outcome of unpredictable physical processes such as device's internal thermal noise, metastability, random telegraph noise (RTN), power supply noise, clock jitter, or even quantum phenomena [5], [7] [10]. TRNG exploits the source of physical entropy in order to translate it into a random bit-sequence (aka raw binary output). There have been different types of TRNGs in literature [21]. Metastability-based TRNGs offer good performance but need a controlling scheme to remove mismatch in devices. In addition, they are very sensitive to environmental variations [21]. Ring oscillator jitter-based TRNGs are simple, but have relatively low randomness. Besides, they are susceptible to high-voltage attack [21]. SRAM-based TRNG, DRAM-based TRNG, and Flash-based TRNG are popular memory-based TRNGs [5], [7] [10]. The start-up values of SRAM cells are used to generate a random number by utilizing the unavoidable manufacturing process variations and internal noise [5], [7], [8]. Two cross-coupled CMOS inverters and two access transistors of a regular 6T SRAM cell, Fig. 1 (a), are designed to be symmetric, match in size, etc., but random variations incurred during manufacturing will result in random mismatches. When powered-up, SRAM cells settle to a stable state either a '0' or a '1' depending on the mismatch of strength between cross-coupled inverters as well as the thermal and shot noise of the cell. A series of SRAM cells generate random numbers. On the other hand, embedded DRAM is denser than embedded SRAM but DRAM-based TRNG is much slower than SRAM-based TRNG because discharging DRAM memory takes large amount of time [10]. Flash memory is fast to read but slow to erase/write. Hence, SRAM is typically considered a better candidate, in terms of area and speed, to generate memory-based TRNGs.

B. Environmental Variations and Aging

Reliability and sensitivity to environmental conditions are the major bottlenecks for current and future CMOS technologies. Bias temperature instability (BTI) affects the strength of MOS transistors by increasing threshold voltage and reducing their drain current [15]. BTI degrades noise margins of a traditional 6-T SRAM cell. On the other hand, hot carrier injection (HCI) changes the threshold voltage of pMOS. Changing threshold voltage changes the strength between two cross-coupled inverters in a SRAM cell. The aging, due to BTI and HCI, increase the delay permanently [15]. Like aging, the robustness of TRNG is impacted by environmental variations [7], [9], [28]. Temperature variations impact threshold voltage, mobility, and thermal noise and thus the PUF outputs [7]. Voltage variation is responsible for delay variations and impact the PUF output [7], [9], [28].

C. Motivations

The motivation of our proposed NS-SRAM based TRNG is as follows.

Limited Entropy and Post-processing Scheme: The quality of hardware-based TRNGs varies from vendor to vendor and across technology nodes [21]. The quality of random numbers depends on entropy, a mathematical measure of randomness, disorder, and variability. The entropy present in SRAM's start-up pattern is limited. The worst-case entropy (i.e., min-entropy) of SRAM's start-up patterns is $\sim 3\%$ [5], [6]. Conditioning algorithm or post processing scheme is used to make the pattern truly random (i.e., with full entropy). A conditioning algorithm requires, in addition with its own area overhead, more than 1100 bytes of start-up patterns, with 3% min-entropy, to obtain a 256-bit true random number [5].

Data Dependency: The amount of uninitialized RAM available for random number generation is limited because same SRAM cells are used for data storage and random number generation [8]. Besides, entropy of start-up patterns might be affected by environmental variations, aging, electronic noise etc [1], [22] [24], [26]. The values stored in SRAM over long periods of time can influence start-up behavior [22] [24], [26]. Thus, a dedicated SRAM block is needed for TRNG.

Availability and Speed: In existing SRAM-based TRNG, the system requires power off and on or power-gating techniques (note that both techniques erase the stored SRAM data) to obtain random numbers which may be challenging because some systems ask for random numbers very frequently. To support the frequent and fast computation, we need a TRNG that makes the output available for the security block. Besides, speed is an important issue that can cause performance problem in Web, mail servers, etc. Post-processing schemes improve the randomness of a raw bitstream but with the cost of speed, area, and power.

Inspired by the above limitations and motivations, we propose a low-cost TRNG that's robust against environmental variations, aging, technology nodes, and different form of attacks in next section.

III. NS-SRAM BASED TRNG

In this section, we present the modification of regular 6T SRAM cells to make them more sensitive to noise. We also present the random number generation scheme without interrupting the other operations of a SoC using power-saving scheme proposed in [20].

A. Design of NS-SRAM

In a SRAM-based TRNG, random number is generated by examining the power-up state of a SRAM because the process variations and resultant preferences are truly random. Upon power up, SRAM cells settle to a stable state either a '0' or a '1' depending on the mismatch of strength between cross-coupled inverters as well as the thermal and shot noise of the cell. The quality of random numbers depends on the quality of entropy in physical source. Hardware-based TRNGs suffer deviation, due to environmental variations, workload, and aging, from ideal statistical properties such as unpredictability, randomness, etc [21].

In this work, our main goal is to achieve very unpredictable start-up values for statistically uniform random numbers at any operating condition, workload, in hostile environments, etc. over the entire system lifetime. The conventional SRAM cells are designed in such a way that they can tolerate significant amount of noise and consume less amount of area. The start-up values depend on the noise margin of the SRAM cells [25]. The main problem of existing SRAM-based TRNG is that most of the cells have stable zero or stable one as start-up value [22], [24], [26].

Fig. 1 (a) shows a regular 6T SRAM cell. In conventional SRAM, proper sizing of these transistors is required to achieve the desired cell area, and stability during reading, writing, and holding data. SRAM cell is powered up to '0' or 1' because cross-coupled inverters have different strength due to uncontrollable process variations and internal random noise. The robustness of a SRAM cell's start-up behavior depends on the amount of noise that a SRAM cell can tolerate. Static noise margin (SNM) is used to determine the robustness of a typical SRAM cell [25]. SNM is defined as the DC noise required to flip the cell's state. Noise margin impacts the start-up value of SRAM cell [25] and hence we need to modify the SRAM cell in such a way that the NS-SRAM cell

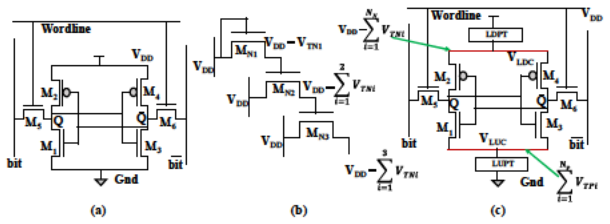

Figure 1: (a) Regular 6T SRAM cell (b) NS-SRAM cell: 6T SRAM cell with voltage level-controlling block to reduce static noise margin, and (c) LDPT: level-down converter with cascaded nMOS transistors. V_{TNi}, V_{TPi}, N_N, and N_P are threshold voltage of i^{th} nMOS PT, i^{th} pMOS PT, total number of nMOS PTs, and total number of nMOS PTs respectively.

can tolerate very low amount of noise during read, write, and hold operations. Read noise margin (RNM), write noise margin (WNM), and hold noise margin (HNM) are the major metrics to quantify the robustness of a SRAM cell. These three metrics are key factors to design a NS-SRAM cell that has very low noise tolerance. Our goal is to minimize these two metrics:

- **Read Noise Margin:** During read operation, the high bit-lines must not overpower the NMOSs M_1/M_3. In other words, the drain to source voltage (V_{ds}) of these transistors must not rise above the threshold voltage (V_{TN}) of M_1/M_3. This condition is defined as $V_Q < V_{TN}$. For robust read operation, I_{M_1} must be greater than or equal to I_{M_5}. To accelerate the read failure, which is our goal, we change the parameters, width mainly, such that $I_{M_1} < I_{M_5}$.
- **Hold Noise Margin:** The inability of the cell to store a given data is measured by hold noise margin. A reduction in HNM of a SRAM cell make it more noise sensitive.

Figure 1 (c) shows the proposed NS-SRAM cell which is more sensitive to noise compared to a regular 6T SRAM cell. Voltage level-up/down converting scheme is used to reduce the static noise margin and DRV significantly. A level-up converting scheme is inserted between power supply (i.e., V_{DD} node) and V_{LDC} node (sources of inverters' nMOSs of regular 6T SRAM). A level down converting scheme is inserted between ground (i.e., V_{SS} node) and V_{LUC} node (sources of inverters' nMOSs of regular 6T SRAM). Pass transistor (PT) based level-converting scheme is used to lower/raise the voltage level in order to make the SRAM cell more sensitive to noise. The advantage of using PTs over a voltage converter is that the lowering circuits are highly susceptible to V_{TH} variations. Fig. 1 (b) shows a level-down converting nMOS PTs (LDPT); note that pMOS PTs replace nMOS of Fig. 1 (b) for level-up converter LUPT. LDPT consists of cascaded nMOS pass transistors and it's placed between power supply and the source of inverter's pMOS of regular 6T SRAM cell (Fig. 1 (c)). LDPT is used to reduce the noise margin by lowering the supply voltage. Fig. 1 (b) shows the LDPT with three cascaded nMOS PTs. The output of LDPT depends on the number of nMOS pass transistors (N_N) and it's equal to $\sum_{n=1}^{N_N} V_{TNi}$, where V_{TNi} is threshold voltage of i^{th} nMOS transistor. On the other hand, LUPT is used to reduce the noise margin by raising the ground voltage. Note that, this voltage up/down can be performed using voltage regulator but it's very expensive and not suitable for area constrained SoCs. Besides, PTs used in LUPT and LDPT increase the noise entropy because of their fluctuations against environmental variations, aging, and internal noise.

Fig. 2 shows the SNM of our proposed NS-SRAM (Fig. 1 (c)) with different number of PTs. The results show that the SNM decreases as the number of PT increases in LDPT and LUPT.

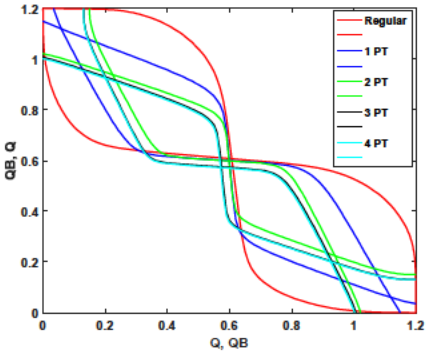

Figure 2: Noise margin decreases as the number of PTs increases in level-converter (PTM 90nm technology node).

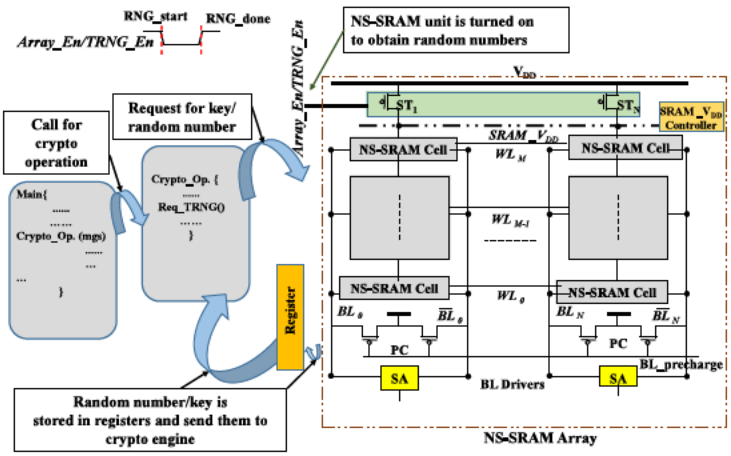

Figure 3: Obtaining random numbers from NS-SRAM based TRNG.

It can be concluded from Fig. 2 that our proposed NS-SRAM cell is a better candidate than regular SRAM for TRNG because the proposed NS-SRAM cell is more sensitive to noise due to significant reduction of noise margin. The NS-SRAM cell is more sensitive to noise during reading, writing, and holding data which is the opposite of what one would want for SRAM used to reliably store data. Thus, start-up value of the NS-SRAM cell produces uncorrelated output from measurement to measurement. The results (Section IV) show that the noise sensitivity of our proposed NS-SRAM cell increases with the number of cascaded PTs.

B. Random Number Generation Mechanism

The memory subsystem of an SoC includes several large SRAM blocks to support core and several small SRAM blocks to support accelerators such as fast image processing, fast video processing, crypto, network interface controller, scratchpad memory, FiFOs, SRAM tag cache to support high density DRAM, etc. [16] [19]. A small SRAM memory block can be as minimim as 1KB of size (memory array of 64X18) [19]. A program in a microprocessor calls for a crypto algorithm which asks for a random number/key to perform the crypto operations in order to protect unauthorized access for preventing malicious attacks or to encrypt the message that will be transmitted through public network channels. Fig. 3 shows our proposed NS-SRAM based TRNG architecture. The proposed NS-SRAM architecture is similar to the Intel architecture presented in [20]. The proposed NS-SRAM cells replace the regular 6T SRAM cells of Intel architecture [20] in order to obtain highly random numbers. We use Intel power-saving scheme/circuit [20] to obtain random numbers/keys from our proposed NS-SRAM array.

Power-gating is a technique used to reduce the leakage power significantly by inserting a transistor switch between a given block circuitry and voltage supply (and/or ground). The major power-gating techniques to reduce potential leakage include memory cell power-gating, wordline power-gating, bitline floating, and bitline I/O power-gating [20]. However, lowering voltage below *retention voltage* causes the SRAM cells to lose data. There are two popular power saving modes: i) the sleep (or standby) mode, where the power supply is reduced to minimum *retention voltage* and ii) the shut-down mode, where supply voltage is reduced to almost zero and SRAM cells lose all stored information [20]. During sleep mode, *SRAM_*V_{DD} controller must assure that the *SRAM_*V_{DD} is above or equal to *retention voltage*. *Retention voltage* of a regular 6T SRAM cell varies from technology to technology. However, NS-SRAM cells lose their data in both modes because of voltage drop by LDPT block (see Fig. 1). Hence, our proposed NS-SRAM cells lose the data during sleep mode.

Random numbers are produced after turning on NS-SRAM based TRNG block when a security module requests for random numbers by lowering the **TRNG_EN/Array_En** to LOW to power ON the NS-SRAM array in order to obtain start-up values. The generated random numbers can be loaded/saved to a register depending on the system requirements. A confirmation signal, which makes the **TRNG_EN/Array_En**='1', is sent to the controller when the random numbers are stored in the registers or used by the security module. Time between the request of random numbers and confirmation signal depends on wake-up latency and the stable start-up outputs from the proposed NS-SRAM cells which depend on the size of sleep transistors, bitline capacitance, bitline drivers, etc. (which is our future study). Note that, **TRNG_EN/Array_En**='1' during idle state of TRNG and **TRNG_EN/Array_En**='0' only when the NS-SRAM array produces random start-up values. Sleep mode has larger wake-up latency than shut-down mode. Depending on the frequency of required random numbers, the NS-SRAM TRNG can be in shut-down mode or sleep mode (NS-SRAM cells still lose the data in sleep mode) and cab be used to generate random numbers.

IV. Results and Discussion

We validated our proposed NS-SRAM based TRNG using three different PTM technology nodes (90nm, 45nm, and 32nm) through HSPICE simulations. We use minimum width for both nMOS and pMOS transistors of level-up/down converter. During simulation, $2-10\%$ (of V_{DD}) noise is incorporated to provide measurement and internal noise. In order to evaluate the robustness against temperature, T is varied from -75^0 C to 100^0 C with 25^0 C of increment. Supply voltage is varied $+25\%$ of nominal V_{DD} (worst case: increase in V_{DD} reduces the randomness) to evaluate the robustness of our proposed NS-SRAM based TRNG against power supply noise or V_{DD} attack. We use HSPICE MOSRA to age the TRNGs (both NS-SRAM based and conventional SRAM-based) for five years. To emulate the process variations, we incorporate $\pm5\%$ threshold voltage variation and $\pm5\%$ oxide thickness (t_{OX}) variation. The min-entropy for a regular SRAM-PUF is ~ 0.03 with these varaiations. We collect 2 billion of SRAM cells' start-up values for NIST test [27]. In our NS-SRAM cell, we use three cascaded PTs to evaluate the robustness of our proposed NS-SRAM based TRNG. We use 3 PTs in both LUPT and LDPT so that the differential sense amplifier can read the start-up values without any read access failure [25]

Impact of Temperature: Threshold voltage and mobility of a transistor are temperature dependent. Fig. 4 (a) shows that the SNM of our proposed NS-SRAM increases with temperature because

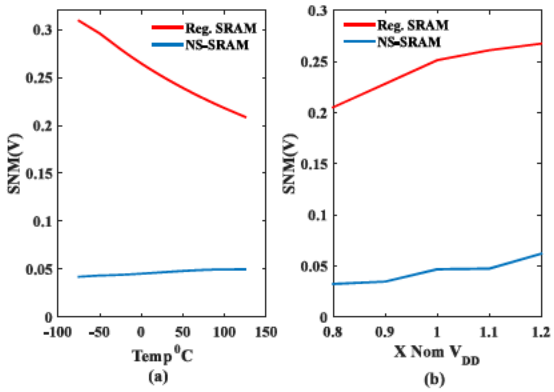

Figure 4: Effects of (a) temperature variations [-75 125] ^0C and (b) power supply variation ([-20% +50%] of nominal V_{DD}) (simulation: PTM 90nm technology node and no process variation).

the threshold voltage of cascaded PTs decreases with temperature (change of current due to change in mobility is less dominant than effective change in threshold voltage in our proposed design). On the other hand, the mobility effect of a regular SRAM dominates the threshold voltage effect because the drain voltage of pMOS and source voltage of nMOS do not change. In NS-SRAM, mobility dominates because the drain voltage of pMOS and source voltage of nMOS change. For the same reason, SNM of our proposed NS-SRAM cell decreases with temperature. This is completely opposite behavior of regular SRAM cell. In regular SRAM cell, SNM decreases as temperature increases and SNM increases as temperature decreases. The above observation in NS-SRAM cell is an advantage in terms of randomness unlike regular SRAM cell. In regular SRAM cell, change in SNM, from -75^o C from 100^o C, is 14% which is 8% for our proposed NS-SRAM cell.

Impact of V_{DD} Variation: Fig. 4(b) shows the impact of V_{DD} variations (from $\pm20\%$ of nominal V_{DD}). Increase in V_{DD} increases the static noise margin. The regular SRAM's SNM increases by 20% where the proposed NS-SRAM cell suffer 7% SNM increment. The reason is that in our proposed NS-SRAM cell, the cascaded PTs are designed with body-biasing. The threshold voltage of cascaded PTs of level up/down converter significantly controls the effectiveness of proposed NS-SRAM cell. Hence, V_{DD} attack is less effective for our proposed NS-SRAM based TRNG.

Impact of Aging: Aging effects shift the threshold voltage of a transistor permanently unlike temperature and voltage variations (temporal change). The NBTI effect increases the read failure of a SRAM due to the presence of weaker pMOS transistors. The SNM degradation due to aging over three years for a regular SRAM cell is $\sim 6\%$. On the other hand, the proposed NS-SRAM cell experiences increase in SNM by $\sim 8.2\%$ over three years. The NS-SRAM cell experiences more degradation than regular SRAM cell because the cascaded PTs also experience threshold voltage degradation. Thus, the proposed NS-SRAM based TRNG performs better against aging.

Randomness: NIST test suite [27] is used to evaluate the randomness of bit streams obtained from SRAM. There is a total of 15 NIST tests and different tests require a different minimum length of bitstreams. For example, rank test, linear complexity test, and overlapping template matching test require at least 38912, 10^6, and 10^6 bits long bitstreams respectively. On the other hand, frequency test, block frequency test, and runs test require a minimum of 100-bit long bitstream. Total 2 billion bits are collected from each measurement to satisfy the minimum required number of bits [27].

We change the temperature from 0^oC to 100^oC and increase

Table I: The Robustness of proposed NS-SRAM based TRNG against environmental variations for (90*nm*) PTM.

Op V_{DD}	1 2V						1 6V	
NIST Tests	0°C		25°C		100°C		100°C	
	Reg	p op	Reg	p op	Reg	p op	Reg	p op
F equency	0 210	0 816	0 290	0 816	0 410	0 739	0 275	0 739
Block F equency	0 015	0 739	0 015	0 739	0 018	0 657	**0.001**	0 554
Cumulat ve	0 011	0 554	0 011	0 616	0 021	0 554	**0.000**	0 484
Runs	0 004	0 484	0 029	0 484	0 035	0 419	**0.000**	0 383
Longest Run	0 008	0 383	0 015	0 419	0 011	0 383	**0.000**	0 334
Rank	**0.001**	0 419	0 010	0 383	0 018	0 334	**0.001**	0 334
FFT	**0.000**	0 474	**0.000**	0 419	0 008	0 419	**0.001**	0 383
OT	**0.000**	0 334	**0.000**	0 213	**0.001**	0 213	**0.000**	0 213
Non-OT	**0.000**	0 213	**0.001**	0 275	**0.001**	0 186	**0.001**	0 275
Se al	0 004	0 213	0 007	0 186	0 007	0 186	**0.001**	0 186
LC	0 015	0 719	0 015	0 616	0 021	0 616	0 011	0 616
Ave age	0 024	0 486	0 036	0 470	0 050	0 370	0 026	0 410

the voltage up to 25% of nominal V_{DD} in order to evaluate the robustness of our proposed NS-SRAM based TRNG against hostile environment. Figures 4(b) and 4(a) show that the start-up values of SRAM have direct impact on voltage and temperature variations. High voltage and high temperature is the worst case since both increases the noise margin (bad for TRNG). The results, in Table I, show that the proposed NS-SRAM based TRNG produces good quality of random numbers. The results show that the proposed NS-SRAM based TRNG is 13*X* more robust against extreme environmental conditions than SRAM-based TRNGs. The result shows that the proposed inspired-TRNG is robust, passes all NIST test (i.e., p-value>0.01), against extreme environmental conditions when existing SRAM-based TRNGs fail the test (marked in bold in Table I). The randomness of a conventional SRAM-based TRNG increases with temperature because of decrease in noise margin and increase in thermal noise. For NS-SRAM based TRNG, on the other hand, temperature reduces the noise margin but increases the thermal noise and thus NS-SRAM based TRNG is more robust than existing SRAM-based TRNGs. Increase in voltage decreases the noise sensitivity of both regular SRAM and NS-SRAM, but the percentage of noise sensitivity decreases more for a regular SRAM than the proposed NS-SRAM. Thus, NS-SRAM based TRNG is more robust than existing SRAM-based TRNGs.

Table II: NIST test results for different nominal V_{DD} and +20% of nominal V_{DD} for different PTM technology nodes at 50°C.

Technology	32*nm*		45*nm*	
Op Voltage (V_{DD})	1 0V	1 2V	1 0V	1 2V
F equency	0 911413	0 816537	0 816537	0 816537
Block F equency	0 455937	0 455937	0 419021	0 383827
Cumulat ve ‘	0 816537	0 657933	0 534146	0 455937
Runs	0 657933	0 616305	0 419021	0 419021
Longest Run	0 739918	0 455937	0 383827	0 350485
Rank	0 739918	0 534146	0 657933	0 657933
FFT	0 350485	0 455937	0 383827	0 275709
Ove lapp ng Templates (OT)	0 455937	0 455937	0 534146	0 419021
Non-OT	0 739918	0 616305	0 616305	0 512398
Se al	0 534146	0 534146	0 739918	0 657933
L nea Complex ty (LC)	0 739918	0 739918	0 534146	0 455937
Ave age	0 649278	0 576276	0 548984	0 491339

One of the most important features of a good TRNG is that it has to be technology independent. Table II shows that the implemented TRNG offers highly random bit-streams for 45*nm* and 32*nm* PTM technology nodes. In terms of randomness, 32*nm* offers better randomness than 45*nm* (or 90*nm*) technology node because lower technology nodes are more sensitive to noise.

Aging degrades the SNM for both regular and our proposed SRAM cells. Table III shows the randomness of our proposed NS-SRAM based TRNG. 32nm technology node has the lowest

Table III: Robustness of our proposed NS-SRAM based TRNG against aging for different technology nodes.

Tests	32*nm*	45*nm*	90*nm*
F equency	0 911413	0 816537	0 739918
Block F equency	0 401199	0 350485	0 350485
Cumulat ve	0 816537	0 657933	0 534146
Runs	0 739918	0 739918	0 455937
Longest Run	0 739918	0 455937	0 350485
Rank	0 739918	0 739918	0 657933
FFT	0 213309	0 350485	0 350485
OT	0 534146	0 455937	0 534146
Non-OT	0 739918	0 455937	0 616305
Se al	0 657933	0 739918	0 739918
LC	0 851383	0 739918	0 534146
Ave age	0 649421	0 576300	0 512398

threshold voltage. The degradation of low threshold voltage is higher than high threshold voltage. Thus, our proposed NS-SRAM based TRNG more robust at lower technology nodes. Our proposed SRAM-based TRNG offers better randomness because the proposed NS-SRAM degrades more than the regular SRAM cell.

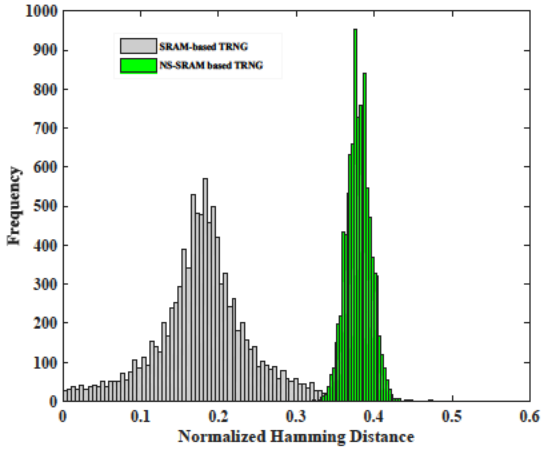

Figure 5: Correlation among start-up values from measurement to measurement (simulation: PTM 90nm technology node, 5% measurement noise, and total 1024 measurements).

Correlation among Start-up Values: The random numbers from an TRNG have to be different from measurement to measurement. Fig. 5 shows the correlation between start-up values of NS-SRAM based TRNG and SRAM-based TRNGs. The result shows that the proposed NS-SRAM based TRNG has better normalized hamming distance (average: 0.38) than SRAM-based TRNGs (average: 0.19); the ideal value is 0.5 (i.e, least correlation).

Table IV: Area and throughput comparison for 256-bit random numbers

TRNG type	M n-ent opy	Req cells	Speed (# eq cells/wo d s ze)
SRAM-based	0 03	8800	1375 cycles
NS-SRAM based	0 89	288 1 5	5 cycles
Imp ovement	∼30X	432X	∼275X

Area and Throughput: The required area depends on the quality of bi tstream and length of the output of a TRNG. The average min-entropy per bit (i.e., start-up value of a cell) for a regular SRAM-based TRNG is ~ 0.03 for 90*nm* technology node. It requires ~ 8800 SRAM cells (1 bit per SRAM cell) for a SHA-256 (post-processing or conditioning algorithm) to generate 256-bit random number [5], [6]. On the other hand, the average min-entropy per bit for our proposed NS-SRAM based TRNG is 0.8921 because of significant amount of reduction in noise margin. The randomness and min-entropy provided by the proposed NS-SRAM based TRNG

is good enough for most of the applications. Table IV shows the comparison of area and throughput between existing SRAM-based TRNG and our proposed NS-SRAM based TRNG. Note that we exclude speed/required area of post-processing scheme, row and column decoders to avoid comparison complexity. We need a total of 8800 cells for regular SRAM-based TRMG and a total of 288 NS-SRAM cells for NS-SRAM based TRNG. We use minimum sized nMOSs and pMOSs for LDPT and LUPT; hence each NS-SRAM is equivalent to 1.5 regular 6T-SRAM cells. The result shows that the proposed NS-SRAM based TRNG is $\sim 432X$ area efficient than a regular 6T-SRAM based TRNG. On the other hand, regular 6T-SRAM based TRNG and our proposed NS-SRAM based TRN require 1375 cycles and 5 cycles to generate 8800 and 288 start-up values respectively (we assume the word size is 64); an improvement of $\sim 275X$.

Limitations: The major limitation of our proposed NS-SRAM based TRNG is that the NS-SRAM cells cannot be used for other operations (e.g., storing data) because of very low noise margin.

Summary of Results Overall, we draw following major conclusions from these results for our NS-SRAM based TRNG:

- NS-SRAM based TRNG is much more robust than existing SRAM-based TRNG against environmental variations and aging.
- Correlation among start-up values of NS-SRAM cells from measurement to measurement is very low; much better than regular SRAM cells. Less correlation among measurement to measurement is essential property of a strong TRNG.
- The proposed NS-SRAM based TRNG is $\sim 275X$ is faster and $\sim 432X$ more area efficient than existing SRAM-based TRNGs.

Attacks: On-chip SRAM memory communicates with other blocks in the system by internal connection bus and does not have any data/address lines at device pins. Thus, NS-SRAM based TRNG is secure against physical attacks. The on-chip SRAM also resists SW attack because SRAM can be isolated from the SoC OS [2]. One can attack by elevating the power supply voltage and/or by increasing temperature since the robustness of TRNG is affected by environmental variations. High V_{DD} and high temperature based attacks are quite difficult to perform because most of the SoCs have sensor to detect these kind of attacks. Besides, we show that the robustness of our proposed NS-SRAM based TRNG against V_{DD} and high temperature attacks by elevating the voltage up to 20% of nominal V_{DD} and increasing the temperature up to 100^0C.

V. Conclusion

In this paper, we proposed a NS-SRAM based TRNG, for modern SoCs, that is more robust and random compared to existing SRAM-based TRNGs. In order to achieve high randomness, we modify the SRAM cells so that they are very sensitive to noise (i.e., have very low noise margin). We also propose a technique to generate random numbers from the NS-SRAM based TRNG using existing power-management scheme of modern SoCs without turning-off and turning on again (i.e., interrupting) the whole SRAM memory. In terms of area and speed, the SRAM inspired TRNG is superior to SRAM-based TRNGs because our proposed NS-SRAM cells possess more entropy than regular SRAM cells.

VI. Acknowledgment

This work was supported in part by the National Science Foundation (NSF) under grant CNS-1561023 and by the Semiconductor Research Corporation (SRC) under contracts 2572 and 2648.

References

] “True random number generators for a more secure IoT,” ava ab e http //www techdes gnfo ums com/pract ce/techn que/t ue random number generators for more secure systems

2] S Ray, Y J n and A Raychowdhury, “The Chang ng Comput ng Parad gm w th Internet of Th ngs A Tutor a Introduct on,” n IEEE Des gn & Test, vo 33, no 2, pp 76 96, 20 6

3] M Tauh dur Rahman, D Fo te, Q Sh , G K Contreras, M Tehran poor, “CSST Prevent ng d str but on of un censed and rejected ICs by unt usted foundry and assemb y, n IEEE Internat ona Sympos um on Defect and Fau t To erance n VLSI and Nanotechno ogy Systems (DFT), 20 4, pp 46 5

4] K N McG , “Trusted Mob e Dev ces Requ rements for a Mob e Trusted P atform Modu e ” www jhuap edu/techd gest/TD/td3202/32_02 McG pdf

5] V. Leest et al., “Efficient implementation of true random number generator based on SRAM PUFs ” C yptography and Secur ty from theo y to app cat ons, Spr nger Ver ag, Ber n, He de berg, 20 2

6] L , Dongfang et a , “PUFKEY A H gh Secur ty and H gh Throughput Hard ware True Random Number Generator for Sensor Networks Ed Leonhard M Re nd Sensors (Base , Sw tzer and), 20 5

7] D Ho comb, W P Bur eson and K Fu, “Power Up SRAM State as an Ident fy ng F ngerpr nt and Source of True Random Numbers,” n IEEE Transact ons on Computers, vo 58, no 9, pp 98 2 0, 2009

8] N Saxena and J Vor s, “We can remember t for you who esa e Imp cat ons of data remanence on the use of RAM for t ue random number generat on on RFID tags,” Proceed ngs of the Conference on RFID Secur ty (2009)

9] M T Rahman, K X ao, D Forte, X Zhang, J Sh and M Tehran poor, “TI TRNG Techno ogy ndependent true random number generator,” 20 4 5 st ACM/EDAC/IEEE Des gn Automat on Conference (DAC), 20 4, pp 6

0] F Tehran poor, W Yan and J Chandy, “ Robust Hardware T ue Random Number Generators us ng DRAM Remanence Effects ” n HOST, 20 6

] S Sr n vasan et a , “2 4GHz 7mW a d g ta PVT var at on to erant True Random Number Generator n 45nm CMOS,” 20 0 Sympos um on VLSI C rcu ts, Hono u u, HI, 20 0, pp 203 204

2] M Majzoob , F Koushanfar, and S Devadas, “FPGA based t ue random number generat on us ng c rcu t metastab ty w th adapt ve feedback contro ,” C yptograph c Hard and Embedded Systems, pp 7 32, 20

3] V Suresh and W Bur eson,“Entropy and Energy Bounds for Metastab ty Based TRNG w th L ghtwe ght Post Process ng,” n C rcu ts and Systems I Regu ar Papers, IEEE Transact ons on, vo 62, no 7, pp 785 793, Ju y 20 5

4] V F scher, “A c oser ook at secur ty n random number generators des gn,” n Construct ve S de Channe Ana ys s and Secure Des gn, ser Lecture Notes n Computer Sc ence, 20 2, vo 7275, pp 67 82

5] C Cak r, M Bhargava and K Ma , “6T SRAM and 3T DRAM data retent on and remanence character zat on n 65nm bu k CMOS,” Proceed ngs of the IEEE 20 2 Custom Integrated C rcu ts Conference, 20 2, pp 4

6] H. E. Sumbul et al., “A synthesis methodology for application-specific logic-n memory des gns,” DAC, 20 5, pp 6

7] C F Fajardo et a , “Buffer Integrated Cache A cost effect ve SRAM arch tecture for handhe d and embedded p atforms,” n DAC 20 , pp 966 97

8] C Huang and V Nagarajan, “ATCache Reduc ng DRAM Cache Latency v a a Sma SRAM Tag Cache”, PACT, 20 4

9] “SmartFus on2 SoC and IGLOO2 FPGA Fabr c,” m crosem

20] Y Wang et a , “A GHz 2 A/Mb Leakage SRAM Des gn n 65 nm U tra Low Power CMOS Techno ogy w th Integrated Leakage Reduct on for Mob e App cat ons,” n IEEE Journa of So d State C rcu ts, vo 43, no , pp 72 79, Jan 2008

2] K Yang, D B aauw and D Sy vester, “An A D g ta Edge Rac ng True Random Number Generator Robust Aga nst PVT Var at ons,” n IEEE Journa of So d State C rcu ts, vo 5 , no 4, pp 022 03 , 20 6

22] K X ao et a „ ”B t se ect on a gor thm su tab e for h gh vo ume product on of SRAM PUF,” Hardware Or ented Secur ty and T ust (HOST), 20 4 IEEE Internat ona Sympos um on, pp 0 06

23] R Maes and V van der Leest, “Counter ng the effects of s con ag ng on SRAM PUFs”, Proc IEEE Int Symp HW Or ented Secur Trust, pp 48 53

24] A Hosey, M T Rahman, K X ao, D Fo te and M Tehran poor, “Advanced Ana ys s of Ce Stab ty for Re ab e SRAM PUFs,” 20 4 IEEE 23rd As an Test Sympos um, 20 4, pp 348 353

25] M Cortez et a , “Mode ng SRAM Start Up Behav or for Phys ca Unc onab e Funct ons,” IEEE Symp Defect and Fau t To erance n VLSI, 20 2

26] Z Guo, M T Rahman, M M Tehran poor and D Forte, “A zero cost approach to detect recyc ed SoC ch ps us ng embedded SRAM,” 20 6 IEEE Internat ona Sympos um on Hardware Or ented Secur ty and Trust, 20 6, pp 9 96

27] A Rukh n, J Soto and J Nechvata , “A Stat st ca Test Su te for Random and Pseudorandom Number Generators for Crytograph c App cat ons,” Apr , 20 0

28] M T Rahman, D Fo te, J Fahrny and M Tehran poor, “ARO PUF An ag ng res stant r ng osc ator puf des gn,” Des gn Automat on and Test n Europe (DATE), 20 4

29] M Sad and M Tehran poor, ”Des gn of a Network of D g ta Sensor Macros for Extracting Power Supply Noise Profile in SoCs,” IEEE Transactions on very Large Sca e Integrat on (VLSI) Systems, vo 24, pp 702 7 4, 20 6