Variation-resilient True Random Number Generators based on Magnetic Tunnel Junctions

by

Yuanzhuo Qu

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Integrated Circuits and Systems

Department of Electrical and Computer Engineering

University of Alberta

Abstract

In the Internet of Things (IoT) era, security has increasingly become a challenge, so encryption has been widely used to protect data. Random number generators (RNGs), as an essential part of cryptographic systems, are implemented in connected devices for information security. However, inadequate levels of encryption may put data at risk. To ensure a higher level of security for IoT applications, designs of CMOS-compatible true random number generators (TRNGs) are needed instead of conventional pseudo-random number generators.

In this thesis work, the stochastic behavior of spin transfer torque magnetic tunnel junctions (STT-MTJs) is exploited as the source of randomness. However, the randomness quality of the sequences generated from a basic generator with a single MTJ is undermined by fabrication variations in MTJs and PVT (process, voltage and temperature) variations in circuits. To overcome the variation challenges, three variation-resilient TRNG designs based on STT-MTJs are proposed in this thesis work. The first design utilizes a parallel structure with multiple devices to minimize the variation effects, the second design leverages the symmetry of an MTJ pair to take advantage of two identical distributions, and the third design compensates for the probability inaccuracy caused by the variations using a two-step switching process. All three designs can generate high-quality random sequences without using complicated post-processing or real-time feedback circuits. Moreover, general flawed random sources and quality improvement circuits are discussed to provide effective solutions for improving the randomness quality of the random sequences.

The National Institute of Standards and Technology (NIST) statistical test suite is used to evaluate the randomness quality of the generated sequences for the encryption keys in the Transport Layer Security or Secure Sockets Layer (TLS/SSL) cryptographic protocol. The circuit operations are validated in a 28-nm CMOS process by Monte Carlo simulation with a compact model of the MTJ. The randomness quality and hardware properties of the proposed designs are compared comprehensively with other RNGs from the literature. Each of the three TRNG designs requires fewer than 40 transistors and consumes less than 1 pJ for generating 1 random bit, with an operating frequency no lower than 50 MHz, showing the variation-resilience with efficient hardware, low energy and high speed.

Acknowledgements

I am grateful for every unforgettable moment I spent with so many nice people during my two years of graduate study. First of all, I would like to express my sincere gratitude to my supervisor, Dr. Jie Han, for his continuous support of my research. I really appreciate his diligence and motivation in work. My sincere thanks also go to Dr. Bruce F. Cockburn for his professional insights and valuable feedbacks. I would also like to express my thanks for all the contributions from my co-supervisor Dr. Witold Pedrycz, examining committee member Dr. Masum Hossain, and collaborators Dr. Yue Zhang and Dr. Weisheng Zhao. This work would not have been possible without financial support from the Natural Sciences and Engineering Research Council of Canada and Stanley G Jones, and the simulation tools from Canadian Microelectronics Corporation.

I appreciate my colleagues in the research group for their support and friendship: Honglan Jiang, Siting Liu, Yidong Liu, Mohammad Saeed Ansari, Oleg Oleynikov and Anqi Jing. I also wish to give thanks to my friends and mentors for their inspirations and guidance: Lang Liu, Leo Y. Li, Feixia Zhang, Ouyang Wu, Zhankun Xi, Wenhan Shen, Mengqi Fang, Zhiqi Xu, Xiaoxue Jiang, Yufeng Li, Zhuoxuan Shen, Lei Yang, Summer J.R.R. Cowley, Boon L. Chew, Kathy Gottlob and many else. My special thanks go to Lubin Qu, Zizhou Lao and Tianyang Liu, for their invaluable long-lasting friendship.

I owe my deepest gratitude to my family, for their unconditional love and support.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AP State	Anti-Parallel State
CDF	Cumulative Density Function
CMOS	Complementary Metal-Oxide-Semiconductor
CTG	Combined Tausworthe Generators
DC	Direct Current
FD-SOI	Fully Depleted Silicon-On-Insulator
IoT	Internet of Things
LFSR	Linear-Feedback Shift Registers
MTJ	Magnetic Tunnel Junction
NIST	National Institute of Standards and Technology
P State	Parallel State
PDF	Probability Density Function
PMA	Perpendicular Magnetic Anisotropy
PRNG	Pseudo-Random Number Generator
PVT	Process, Voltage and Temperature
QIC	Quality Improvement Circuit
RNG	Random Number Generator
RRAM	Resistive Random-Access Memory
SC	Stochastic Computation
SSL	Secure Sockets Layer
STT	Spin Transfer Torque
TG	Tausworthe Generators
TLS	Transport Layer Security

TMR	Tunnel Magnetoresistance Ratio
TRNG	True Random Number Generator

Chapter 1: Introduction

1.1 Background and Motivation

The Internet of Things (IoT) names an era of enormous data exchange in physically distributed networks of interconnected devices [1]. Due to the rapidly growing volume of valuable data transmitted over the Internet, data security has become an increasing concern. Therefore, data encryption needs to be implemented to prevent unauthorized parties from accessing the data during storage and transmission. Inadequate levels of encryption may put data at risk and lead to privacy, property or even physical losses [2] [3], and consequently strong on-chip encryption methods are needed to ensure a high level of security for IoT applications.

Random numbers are an essential part in an encryption algorithm. Two categories of random number generators (RNGs) are used: pseudo-random number generators (PRNGs) and true random number generators (TRNGs) [4]. Tausworthe generators and a specific implementation, linear-feedback shift registers (LFSRs), are typical examples of PRNGs [5]. The sequences generated from PRNGs are fully deterministic but their statistical properties make them look random. The generation algorithms make the sequences fully predictable and periodic, and the same sequence will be generated from the same random seed [6]. Thus, there are interests in replacing PRNGs in cryptographic applications because of the predictability.

In contrast with PRNGs, TRNGs generate numbers with true randomness that originates from nondeterministic physical phenomena [7]. Some types of random physical events,

such as the chaotic behavior in semiconductor lasers [8] [9], can produce random bitstreams extremely fast with high quality (e.g., 480 Gbit/s is reported in [9]). However, on-chip applications require schemes that are scalable and compatible with CMOS technology. Moreover, energy consumption and the generation speed are important implementation criteria for mobile devices in the IoT era. Therefore, we seek TRNGs that can produce random sequences for cryptographic applications with CMOS compatibility, high statistical quality, low area cost and high energy-efficiency.

One major group of generators that does not involve non-CMOS devices are called all-digital TRNGs [10]. Designs leveraging metastability [11] and oscillator jitter [12] [13] tend to have relatively poor randomness, so complicated post-processing is usually needed, which increases circuit area and energy consumption. Oxide breakdown-based TRNGs can produce high-quality random numbers, but they have a relatively slow generation speed and high power consumption (e.g., only 11 kbit/s in [14] with a power of 2mW).

Some emerging nanoscale devices with stochastic behaviors, such as memristors [15] [16], resistive random-access memories (RRAMs) [17] and magnetic tunnel junctions (MTJs) [18] [19] [20] [21] [22] [23], can be implemented as TRNGs. MTJs with spin transfer torque (STT) switching have the advantages of high density, high endurance and compatibility with CMOS process, so they are promising candidates for TRNG designs. STT-MTJ based TRNGs are more power-efficient and have higher generation speed compared with memristor-based or RRAM-based TRNGs [24].

However, variations exist in the MTJ devices and also in the circuits. Due to limitations in fabrication and operation, there is a probability bias in the generated random sequences,

i.e., the frequency of 1's in the output binary bitstream is shifted away from the expected 50%. The sources and effects of the variations are explained in detail in Section 2.1.4. The basic TRNG design based on a single STT-MTJ device has to be post-processed or tracked in real time to ensure an acceptable level of randomness. Several designs using post-processing or real-time tracking circuits are reviewed in Section 1.2. These additional circuits will increase the hardware cost and energy consumption of the basic generator, and may introduce some other undesired behaviors. Therefore, this thesis work is focused on hardware-efficient TRNG designs based on MTJs that can provide random sequences with high variation-resilience.

1.2 Related Work

The emerging nano-devices whose stochastic behaviors are leveraged to design TRNGs include memristors, RRAM and MTJs. Despite the different details in the physical mechanisms, all types of devices have the following properties in common:

- There are (at least) two stable resistive states for a device.
- The state of the device is non-volatile.
- The switching of the states is probabilistic under certain conditions.

The designs based on these devices share some similarities. To produce one random bit, the targeted device is set to a certain state with the expected probability based on the probabilistic switching. Then it remains in its state because of the non-volatility, and the output is produced from the sensing of the state. However, there are problems with the quality caused by the insufficiency of the random entropy, so various ways are used to improve the randomness quality.

One of the possible solutions in the literature uses multiple generators to generate multiple uncorrelated bits, and then performs XOR operations among them [16] [22]. At least four MTJs and three XOR gates are needed to obtain one random bit in [22], which wastes generated bits and increases the hardware cost.

Another method is to post-process the original output, such as by using the von Neumann correction, which considers two non-overlapping bits at a time and only produces one bit if the two bits are not equal [17]. Therefore, it requires complicated digital circuits and the bit utilization rate is only 25% at most. Additionally, the generation of the processed bitstream depends on the original output, so it is not time-constant [25]. Note that some post-processing schemes are used in some of the proposed designs in this work, but they are much simpler and do not compromise the generation speed.

In addition, many TRNG designs include real-time feedback calibration circuits, in which the actual frequency of 1's in the output is calculated. Then the probability of the next bit(s) to be generated is adjusted according to the previous outputs in order to ensure an overall probability of 50% [19] [20] [21]. However, the calibration circuit is quite large as it involves counters, comparators and other circuit components. Moreover, the use of calibration circuits undermines the randomness, because the probability is always fluctuating to be either higher or lower than 50% according to the previous outputs.

1.3 Contributions of this Work

The main contributions of this thesis work are the novel designs of hardware-efficient TRNGs that aim at reducing the probability bias in the generated random sequences with the presence of variations. Contrary to the designs in the literature, none of the proposed

designs in this work require complicated circuits to ensure the high randomness quality in the output sequences. The designs were verified in simulation using the perpendicular magnetic anisotropy (PMA) STT-MTJ compact model [26] with ST Microelectronics' 28-nm fully depleted silicon-on-insulator (FD-SOI) CMOS technology [27].

Note that most of the RNG designs in the literature do not include comprehensive statistical tests. The correct function is usually claimed by only proving a 50% frequency of 1's in the output sequences, however, this condition is not sufficient to fully verify the randomness. Thus in this thesis work, the randomness quality of all generated sequences is validated appropriately using the National Institute of Standards and Technology (NIST) SP-800 statistical test suite [28].

Specifically, the contributions are summarized as follows:

- A review of recent work on TRNGs, especially those based on emerging nano-devices.
- A theoretical analysis of two categories of flawed random sources: one with a fixed bias and the other with a certain variation.
- Three different designs for TRNGs based on MTJs: the parallel design, the MTJ-pair design and the self-calibration design. The parallel design uses multiple devices to minimize the variation effects, the MTJ-pair design leverages the symmetry of two MTJs, and the self-calibration design compensates for the probability inaccuracy by a two-step switching process. Randomness quality evaluations using a statistical test suite are conducted on all designs. Work on the parallel design appeared as [29] in the *2017 Design, Automation and Test in Europe (DATE)* conference.

- A universally applicable quality improvement circuit: the tolerance levels of the probability bias/variation are provided as general guidelines for choosing the random source and the quality improvement circuit based on quality requirements.
- Comprehensive comparisons of the randomness quality and hardware properties of the proposed TRNG designs with other RNGs in the literature: the comparisons show that each of the three proposed designs has specific advantages; however, they are all variation-resilient and hardware-efficient.

1.4 Thesis Outline

This thesis is composed of five chapters:

- Following the Introduction, Chapter 2 provides background about the device structure and the stochastic behavior of MTJs, explains the existing problems caused by variations in the basic single-MTJ generator, introduces the statistical test suite for evaluating the randomness quality, and presents a theoretical analysis of two categories of general flawed random sources.
- In Chapter 3, the three TRNG designs based on MTJs are presented using theories, schematics and generating procedures. A quality improvement circuit is proposed to improve the randomness quality.
- In Chapter 4, the randomness quality and hardware properties of the proposed designs are simulated and evaluated, and then compared with several RNGs from the literature.
- Finally, Chapter 5 concludes the thesis work and provides suggestions for future research.

Chapter 2: Background

2.1 Magnetic Tunnel Junctions

2.1.1 MTJ Device Structure

An MTJ is a basic spintronic device that exploits the tunnel magnetoresistance effects. Figure 2.1 shows the structure of a typical MTJ, which has three layers: two relatively thick ferromagnetic layers (e.g., CoFeB) separated by one relatively thin tunneling barrier layer (e.g., MgO) [30]. One of the ferromagnetic layers is called the free layer for its switchable magnetization and the other one is called the pinned layer or fixed layer for its fixed magnetization. There are two stable states for an MTJ, i.e., the parallel (P) state and the anti-parallel (AP) state, determined by the relative magnetization of the two ferromagnetic layers. When the device is in the P state, it has a lower electrical resistance R_P, and when the device is in the AP state, it has a higher resistance R_{AP}. The MTJ will remain in its state unless a magnetic field or a current interferes with it, so it can be used for non-volatile memories. The tunnel magnetoresistance ratio (TMR), namely

$$TMR = \frac{R_{AP} - R_P}{R_P} \tag{2.1}$$

characterizes the relative resistance difference between the two states, which is typically around the range of 150% to 200% [31].

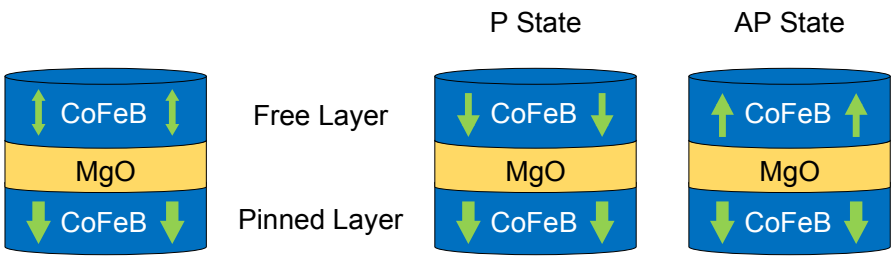

Figure 2.1 The structure of an MTJ and its two states

The MTJ used in this work has perpendicular magnetic anisotropy (PMA), which means that the magnetization of the ferromagnetic layers is perpendicular to the layer plane. This configuration has a better thermal stability and a lower critical current compared with the in-plane magnetic anisotropy MTJ [32].

2.1.2 MTJ Probabilistic Switching

An efficient way to set the state of an MTJ is to inject a current into it to produce an effect called spin transfer torque (STT) switching [33]. Figure 2.2 shows the STT switching, and the direction of the current determines the final state of the MTJ: the MTJ will be set to the AP state if the current is injected from the pinned layer side, and the MTJ will be set to the P state if the current is injected from the free layer side. During the STT switching process, the current (electrons) is spin-polarized when passing through the pinned layer, and the spin-polarized current will transfer sufficient spin-angular momentum to the magnetic moment in the free layer to switch its magnetization making it align with that of the current. STT switching needs a lower current density compared with the field-induced switching using a separate current to produce a magnetic field, so the STT-MTJ is both more scalable and more energy-efficient [34].

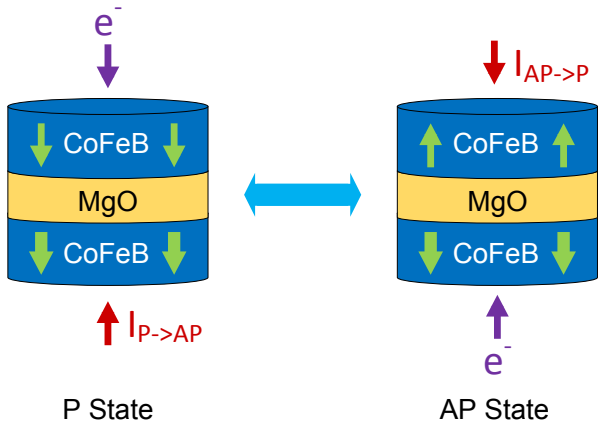

Figure 2.2 The STT switching of an MTJ between the two states

Due to thermal fluctuations of magnetization during STT switching, the time to complete the switching follows a statistical distribution. In fact, the switching is probabilistic given a fixed current and pulse duration. The relationship between the amplitude (I), duration (t) of the current pulse and the switching probability (P) can be expressed as follows:

$$P(I,t) = 1 - \exp\left(-\frac{t}{\tau}\right) \tag{2.2}$$

$$\tau(I) = \tau_0 \exp\left[\Delta\left(1 - \frac{I}{I_{c0}}\right)^2\right] \tag{2.3}$$

where τ is the mean switching time, τ_0 is the attempt time, I_{c0} is the critical switching current at 0 K and Δ is the thermal stability factor related to temperature [22].

Based on (2.2) and (2.3), when the current (I) and the pulse duration (t) are well controlled, a certain switching probability can be achieved. When a carefully controlled current pulse aiming for a certain switching probability is applied to an MTJ, the MTJ will end up in a certain state with the expected probability. By sensing the state of the MTJ, the intrinsic stochastic behavior can be exploited to generate random numbers.

2.1.3 Device Variations of the MTJ

In this work, a 28-nm PMA-STT-MTJ compact model [26] was used with 28-nm FD-SOI CMOS technology, and the hybrid MTJ/CMOS circuits were simulated in Cadence Virtuoso [35]. The values of the parameters set for the MTJ model are listed in Table 2.1.

The two resistance values R_P and R_{AP} of an MTJ are affected by several factors such as the dimensions of the device as well as other material properties. Due to the limitations in fabrication, especially the limited accuracy in the thickness of the three layers during thin film deposition, the resistances of the fabricated MTJs will vary from the nominal values

[36]. To consider this effect at the design stage, three parameters are extracted to represent the MTJ variations: the thickness of the tunneling barrier layer (t_{ox}), the thickness of the free layer (t_{sl}) and the TMR value. These parameters are assumed to follow Gaussian distributions with standard deviations of 3% of the expected value (Table 2.1) [37]. The resistance values are affected by the combined effects of these parameters.

Parameter	Description	Value
t_{ox}	Thickness of the MgO layer	0.85 nm
$\sigma_{t_{ox}}$	Standard deviation of t_{ox}	3% of 0.85 nm
t_{sl}	Thickness of the free layer	1.3 nm
$\sigma_{t_{sl}}$	Standard deviation of t_{sl}	3% of 1.3 nm
TMR	Tunnel magnetoresistance ratio	200%
σ_{TMR}	Standard deviation of TMR	3% of 200%
Area	MTJ dimensions	$28\ \mathrm{nm} \times 28\ \mathrm{nm} \times \pi/4$

Table 2.1 Parameters of the MTJs used in this work

Monte Carlo simulation is the main method to obtain the parameter distributions and the switching probabilities. Figure 2.3 is a DC simulation example of the MTJ model used in this work. The four hysteresis loops illustrate the resistances of four MTJs changing with the voltage applied directly to the MTJ devices (V_{dc}). The variation effects can be seen from the resistance differences in each hysteresis loop.

Figure 2.4 shows the distributions of the two resistance values for the MTJs, where 1000 Monte Carlo simulations were performed for each resistance state. The mean values of

R_P and R_{AP} are 8.1 kΩ and 23.7 kΩ, respectively, and the standard deviation is 6.3% of the mean. In TRNG designs, MTJ variations will affect the current in circuits and these variations can undermine the quality of the generated random numbers.

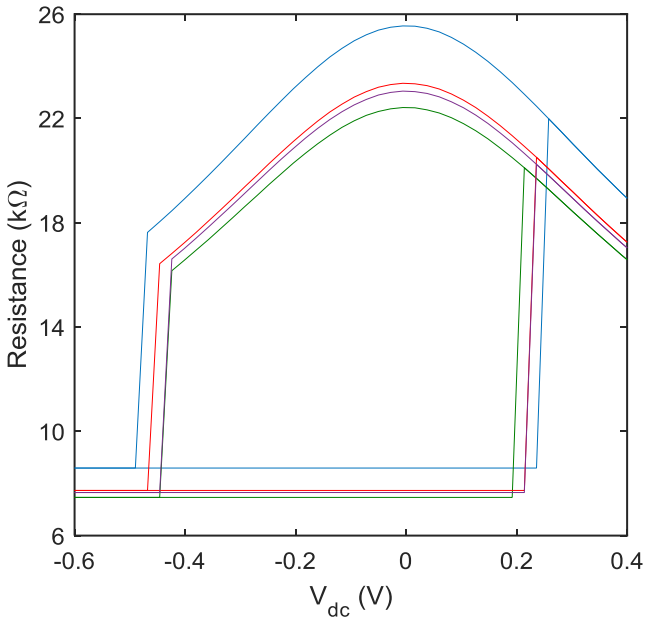

Figure 2.3 DC simulation for the 28-nm PMA-STT-MTJ

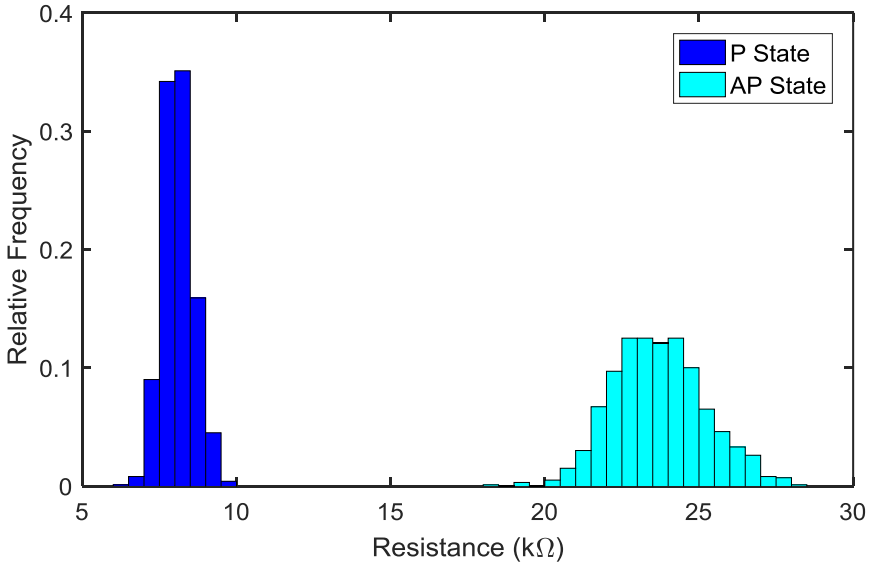

Figure 2.4 The resistance distributions of R_P and R_{AP} for the 28-nm PMA-STT-MTJ

2.1.4 Basic Generator with a Single MTJ

The MTJ switching probabilities were examined in an actual circuit according to the theory. Figure 2.5 shows a basic writing circuit for a single MTJ. The switching current is applied from a voltage source (V_{write}) and controlled by two NMOS access transistors. When the initial state is set to the P state, single MTJ switching probabilities under different voltages with 5-ns and 10-ns pulse durations are shown in Figure 2.6, where different voltages and pulse durations are seen to affect the MTJ switching probabilities. Moreover, when the initial state is set to the AP state and the pulse duration is fixed to 5 ns, similar results can be obtained and are shown in Figure 2.7. Each result is an average from 100 Monte Carlo simulations in Figure 2.6 and Figure 2.7.

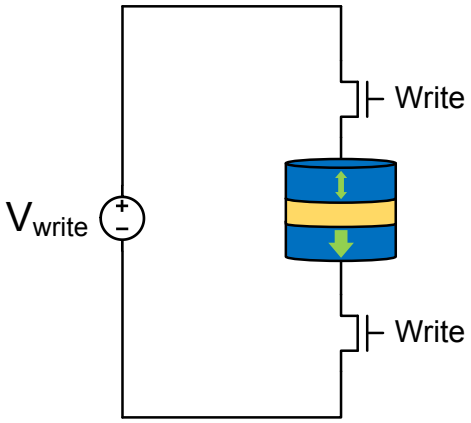

Figure 2.5 Basic writing circuit for a single MTJ

Note that the switching probabilities in Figure 2.6 and Figure 2.7 are only examples illustrating the trend. Due to the finite number of simulations, the exact values may vary a little in each round of simulations. In actual implementations, the actual voltage and pulse width applied in a given design should be chosen according to the specific circuit parameters to achieve the desired switching probability. Also, the discrepancy between these two figures is caused by the parameters in the CMOS part.

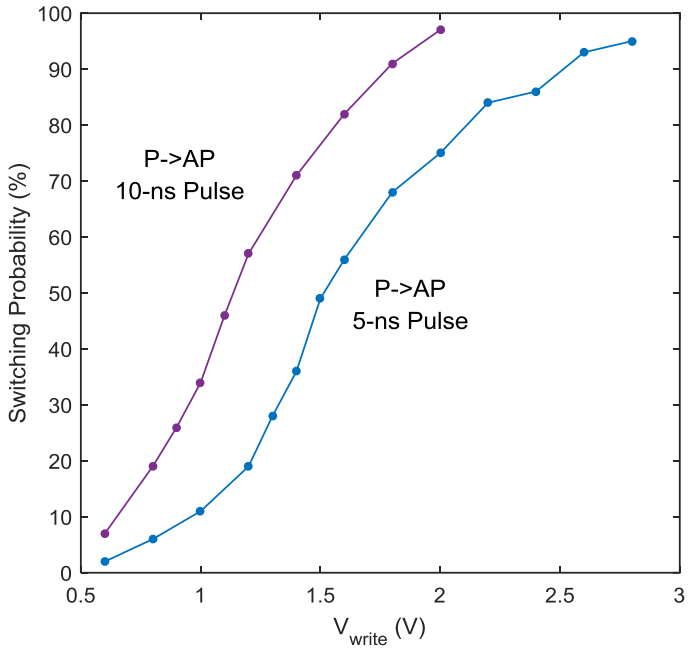

Figure 2.6 The switching probability under different voltages with 5-ns and 10-ns pulse durations

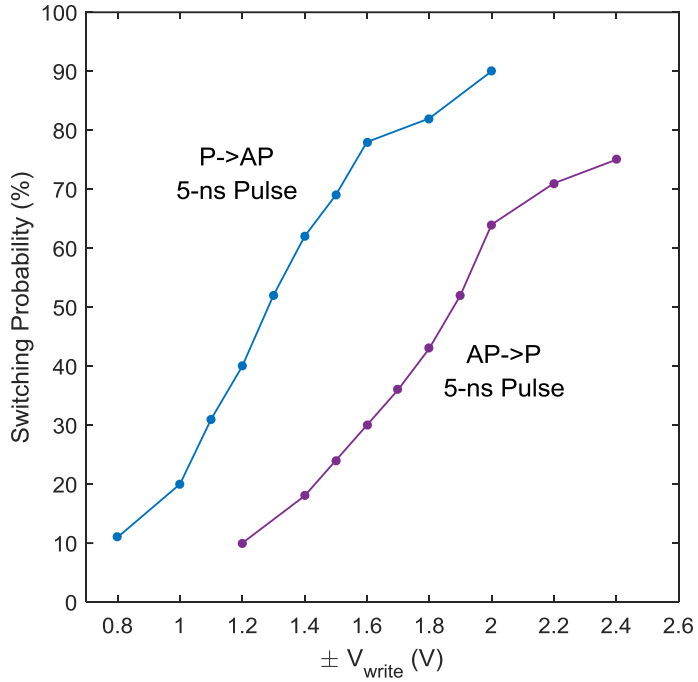

Figure 2.7 The switching probability under different voltages with both initial states

Since device variations exist in all MTJs, the resistances of the two states for any particular MTJ will differ a little from the nominal values. Therefore, the current going through an MTJ will differ and so will the switching probability, which will usually lead to a probability bias in the generated sequences. The MTJ fabrication variation will lead to a standard deviation of 3.14% in the actual probability from the ideal 50%. Therefore, using only one MTJ is not sufficient to generate practical random sequences because the probability varies from 40.58% to 59.42% over $\pm 3\sigma$.

The probability bias may also come from other sources, such as the PVT (process, voltage and temperature) variations in the circuit elements. For example, if the CMOS process parameters change to Fast or Slow from Typical, or the operating voltage varies from 0.9 (low voltage) to 1.1 (high voltage) times the nominal voltage, a basic TRNG based on a single MTJ switching will have a severe probability bias of more than $\pm$ 10% from the expected 50% (see Figure 2.8 and Figure 2.9, respectively). In these two figures, the proportion of MTJs that switch will converge to the switching probability with increasing numbers of simulation cycles.

As there are variations both in the MTJ devices and in the CMOS circuit operations, the basic generator with a single MTJ is subjected to the unacceptable probability bias in the output sequences. Therefore, other design methods are required to improve the randomness quality.

The VerilogA code of the MTJ model is in Appendix A for reference.

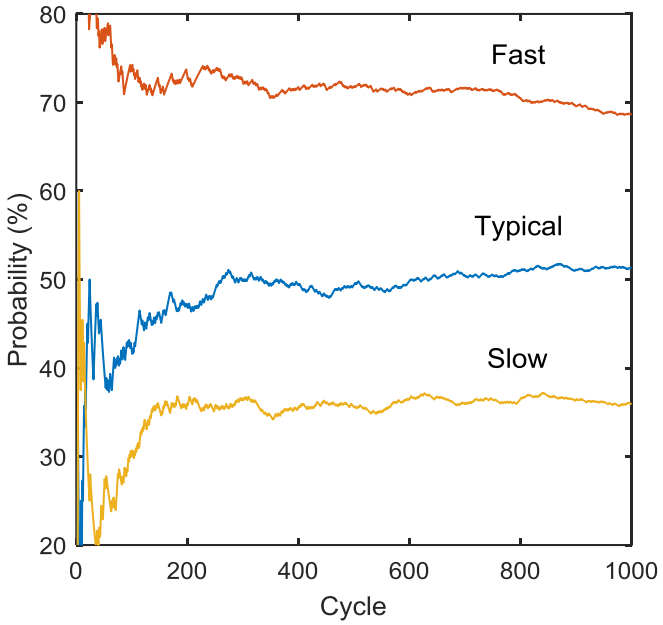

Figure 2.8 Single MTJ switching probability for different process parameters

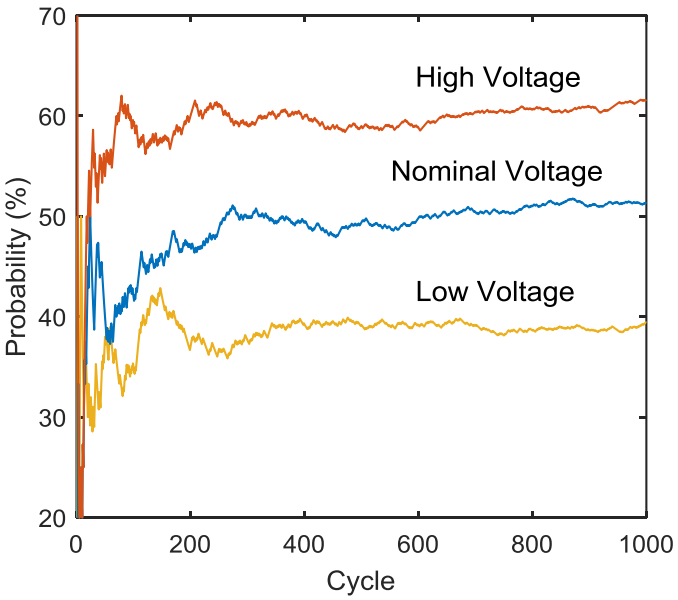

Figure 2.9 Single MTJ switching probability for different operating voltages

2.2 The Evaluation Methods for Randomness Quality

To test and compare the randomness quality, all sequences generated by the methods proposed in this thesis work went through the evaluation process described below:

The length of the random sequences was chosen to be 256 bits, because in cryptographic applications, such as Internet security, the typical key length is 256 bits for a Transport Layer Security or Secure Sockets Layer (TLS/SSL) cryptographic protocol [38].

The widely used statistical test suite National Institute of Standards and Technology (NIST) Special Publication 800-22 rev.1a [28] was applied to evaluate the quality of the random sequences. There are 15 types of tests in the suite in total, but 7 types with a total of 9 tests in the suite were selected to evaluate the sequences because other tests in the suite require millions of bits in a sequence. The selected tests were divided into two categories according to their relationship with frequency:

- Frequency-related tests
    - Frequency (Monobits) Test
    - Frequency Test within a Block
    - Cumulative Sums (2 tests)

Frequency-related tests examine whether a sequence has a reasonable portion of 1's and 0's as a whole or in any sub-sequences.

- Non-frequency tests
    - Runs
    - Longest Run of Ones in a Block
    - Approximate Entropy
    - Serial (2 tests)

Non-frequency tests evaluate a sequence in aspects other than frequency such as the presence of oscillations and reoccurring patterns. The detailed definitions and descriptions of the tests can be found in [28]. The values of the parameters set for the test suite are listed in Table 2.2.

Parameter	Value
Block length for the Frequency Test within a Block test	32
Block length for the Approximate Entropy test	2
Block length for the Serial tests	5
Significance level (α)	0.01
Number of bits in a sequence	256
Number of sequences in a test	1000

Table 2.2 Parameters of the statistical test suite used in this work

The significance of using a multi-test suite is that only aiming at 50% of 1's and 0's, as in many research, may lead to undesired results. For example, a sequence with alternating 1's and 0's (10101010…) will definitely pass all frequency-related tests, since it has perfect proportions of 1's and 0's in every part of the sequence. However, this sequence is very unlikely to be random. With the non-frequency tests, it is easy to exclude this sequence from the choices of good random sequences. First, there are too many runs (sub-sequences of consecutive 1's or 0's) in this sequence, or we could say the oscillation is too fast, which will cause the sequence to fail the tests of Runs and Longest Run. Second, when two overlapping bits are considered at a time, the patterns of "10" and "01" occur far

more frequently than the patterns of “00” and “11”, which will cause the sequence to fail the Serial tests [39].

Basically, all the tests are based on statistical hypothesis testing [40]. First, two hypotheses are made: the null hypothesis (the sequence under test is random) and the alternative hypothesis (the sequence under test is not random). Then, a significance level (α) is chosen, which is the probability that a random sequence is wrongly indicated as non-random. Next, a P-value is calculated based on the actual sequence. If the P-value is larger than or equal to α, the sequence is considered random with a confidence of $1 - \alpha$. After all sequences are processed for a certain test, finally, a confidence interval is used to determine whether the certain test is passed or not. If the pass rate for the certain test lies in the interval, then the corresponding test is passed.

To have a convincing conclusion, 1000 sequences were generated in every scenario: when the significance level is $\alpha = 0.01$ and the number of sequences tested is $m = 1000$, the confidence interval is $(1 - \alpha) \pm 3 \times \frac{\alpha(1-\alpha)}{m} = 0.99 \pm 0.0094392$. Therefore, the pass rate for any tests needs to be greater than or equal to 0.981 to satisfy acceptable randomness. In other words, at least 981 in 1000 sequences should pass the test. Note that to validate the randomness quality of a generator, all 9 tests for that generator must pass with all pass rates of no less than 0.981. All of the average pass rates in this work are for illustration and comparison purposes only, and should not be used as indicators for passing the tests.

2.3 Two Categories of Flawed Random Sources

Before the specific designs for MTJ-based TRNGs are proposed, some general analysis of random sources is conducted to better understand the probability bias/variation issues in TRNGs. A drawback of TRNGs based on nondeterministic physical phenomena is that the probability in the generated sequences is more sensitive to various factors, so they are often flawed to some extent. Generally, the flawed random sources are divided into two categories: a particular generator device under certain operating conditions lies in the first category, while the population of a group of devices before fabrication lies in the second category. The first category is called “random sources with a fixed bias”, and the second category is called “random sources with a certain variation”.

A fixed bias means that the probability that the random source produces is not exactly 50%. We define the bias δ as the difference between the actual probability and the ideal 50%. For example, a generator which produces 60% of 1’s or 40% of 1’s in the output sequences has a δ of 10%. The direction of the bias is of no significance since it can be converted by an inverter.

This category of random sources is usually a particular device after fabrication and under certain operating conditions. For example, an MTJ after fabrication will have fixed parameters, so it will have a fixed switching probability given a certain pulse in a certain temperature. However, due to the limited precision of all the parameters, the expected probability will be a fixed value yet not an accurate 50%. Therefore, the probability bias in the generated random sequences will be fixed, but it will only be known after learning about the fabrication results and other operating conditions.

On the contrary, for a general type of random sources, such as a special design based on MTJs, the distribution of the probabilities in the sequences from all the individual generators of the design can be predicted from the device properties and design parameters. However, the actual probability bias of a particular generator in that type cannot be known.

To analyze the variation quantitatively, we introduce a variation factor d. It is defined as the percentage of the standard deviation σ of a random source over its expected value μ:

$$\frac{\sigma}{\mu} = d\% \tag{2.4}$$

We always expect the probability of a random source to be 50%, or μ = 0.5, so $\sigma = 0.5 \times d\% = 0.5d\%$. The actual probability of that type of random sources varies from $50\% - 1.5d\%$ to $50\% + 1.5d\%$ over $\pm 3\sigma$. For example, a design for MTJs with probability variations of σ = 3.14% (d = 6.28) under certain fabrication process and operating conditions will have probabilities varying from 40.58% to 59.42% over $\pm 3\sigma$.

Chapter 3: True Random Number Generator Designs based on Magnetic Tunnel Junctions

3.1 The Parallel Design

The parallel design compensates for the device variation problem with multiple MTJ devices. Since the standard deviation of the average of N independent Gaussian-distributed random variables is

$$\sigma_{\frac{X_1+\cdots+X_N}{N}} = \frac{\sqrt{\sigma_1^2+\cdots+\sigma_N^2}}{N}\left(=\frac{\sigma_N}{\sqrt{N}}, if\ X_1=\cdots=X_N\right), \quad (3.1)$$

the random sequences generated by multiple MTJs will have smaller standard deviations (divided by $\sqrt{N}$) in the probability [41]. In other words, the parallel structure averages the biased probabilities of each single MTJ to obtain an overall probability closer to 50%.

3.1.1 Schematics and Generating Procedures

The schematic of the proposed parallel MTJ TRNG design is shown in Figure 3.1. Three MTJs are shown in the figure, but the actual number of MTJs used can be adjusted according to the requirements. Note that if only one MTJ is implemented, the schematic reduces to the basic generator.

For an array with N MTJs, the control signals are *Reset, Write* and *Read*$_n$ ($n = 1, 2, \cdots, N$). To produce N random bits, the circuit needs to go through $N + 2$ phases: 1) a reset phase, 2) a write phase and 3) N read phases. In each phase, the corresponding control signal is driven high while the others are held low. In the first two phases, all MTJs work simultaneously. In the read phases, one MTJ is sensed at a time.

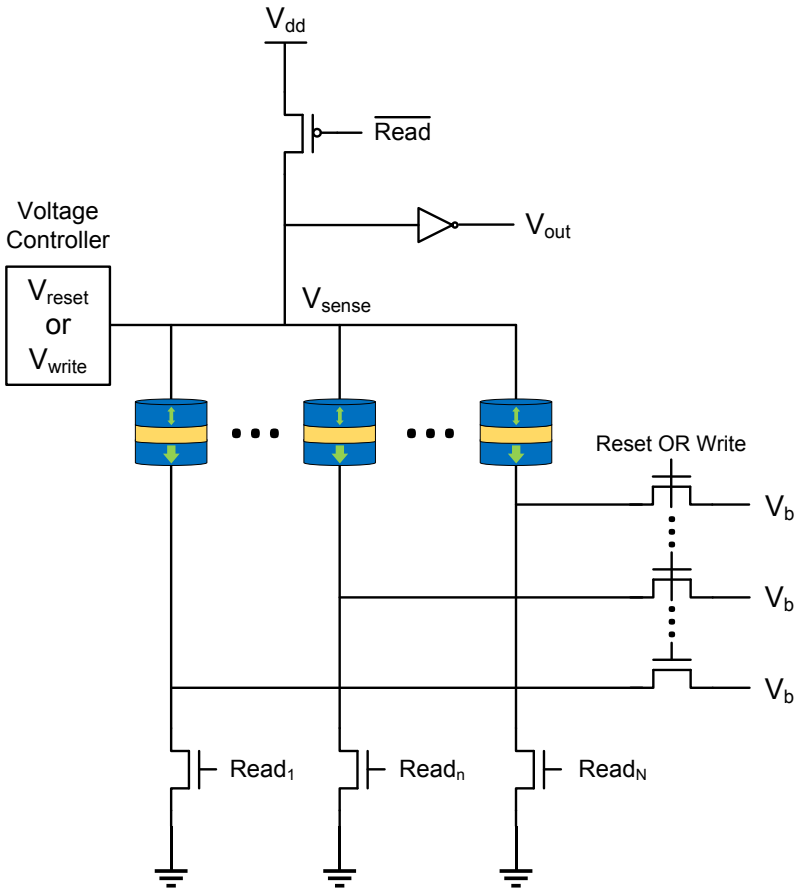

Figure 3.1 Proposed TRNG with multiple parallel MTJs

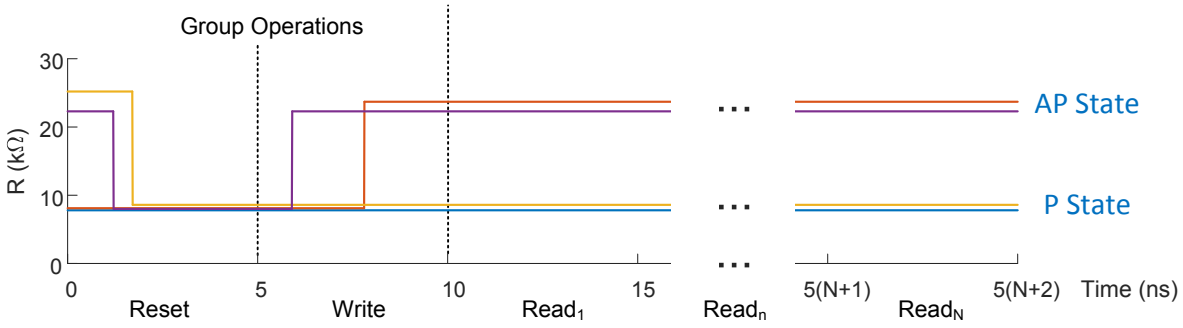

Figure 3.2 Timing diagram of the parallel design in one cycle of operation

Figure 3.2 shows a complete operation cycle for random number generation: each phase takes 5 ns so the whole cycle lasts $(N + 2) \times 5$ ns. The resistances of four MTJs are

plotted in the figure where the probabilistic switching and resistance variations can be seen. Here the $N + 2$ phases are explained in detail:

1) Group Reset:

   In the reset phase, *Reset* is high and other control signals are low. The voltage controller drives V_{reset}, and current flows from the free layer (top) to the pinned layer (bottom) until the MTJs which are in the AP state are switched to the P state. V_{reset} is higher enough than V_b to ensure an almost deterministic switching. At the end of the reset phase, all MTJs are in the P state waiting for the probabilistic switching in the write phase.

2) Group Write:

   In the write phase, *Write* is high and other control signals are low. The voltage controller drives V_{write}, which is lower than V_b to induce a switching current going from the pinned layer to the free layer. The voltages are selected to target a 50% switching probability in 5 ns for each MTJ. Since the MTJs are connected in parallel, the voltages across each MTJ and the corresponding transistors are the same. All MTJs are written simultaneously, but each MTJ switches independently. The voltage controller ensures that V_{write} is held steady despite MTJ switching. At the end of the write phase, an MTJ will change to the AP state if it switches; otherwise, it will remain in the P state.

3) Read:

   In the read phases, only one of the N $Read_n$'s is high, from $Read_1$ to $Read_N$, while all other signals are low. The current flows from V_{dd} to *GND* passing through only the selected MTJ. Depending on the resistance of that MTJ, the V_{sense} will differ

(the voltage controller is now off). The inverter (or some other kind of sense amplifier) will detect the difference and amplify it. Finally, the digital output at V_{out} will indicate the resistance state of the selected MTJ. After N cycles, the states of all the N MTJs are sensed.

The proposed parallel structure will not only produce random numbers with higher randomness quality but will also introduce other advantages compared with a single MTJ circuit. First, only one multiplexed sensing circuit is needed to read out all states of the N MTJs at V_{out}, which saves hardware. Also, all MTJs are reset and written simultaneously, which requires less time compared with using a single MTJ to obtain the same number of random bits. Since $(N + 2) \times 5$ ns are needed to produce N random bits, a generation speed of $\frac{N}{N+2} \times 200$ Mbit/s can be achieved. If N is large enough, the read phase will dominate the operation and the speed will be about ~200 Mbit/s.

3.2 The MTJ-pair Design

In the parallel design, the accuracy of the switching probability is subject to the actual voltage and duration of the pulse applied to the MTJs, and PVT corners (process, voltage and temperature). These global parameters will affect all MTJs in the circuit in the same way and to the same extent. In other words, each of the MTJs may produce random numbers with a probability biased to the same direction, either higher or lower than the expected 50%. In order to keep the probability precise, the pulses applied to the MTJs should be well controlled and the variations of the IC process should be insignificant.

However, instead of producing random numbers by controlling pulses carefully, we can leverage the symmetries of multiple MTJs in the circuit and compare two independent random variables (such as the switching times of two MTJs), which follow the same distribution, to obtain a 50% probability. As long as the two random variables are equally affected by the variations in the circuit, their distributions will be the same all the time. When the two independent variables follow identical distributions, the probability that the first variable is smaller than the second one is 50%, since there is equal probability that either variable is smaller than the other one because of the symmetry. The detailed mathematical proof of the theory appears in Appendix B.

The MTJ-pair design relies on the fact that the switching times of the two MTJs follow identical distributions, thus it can produce random sequences with high variation-resilience in the presence of all major variations. An additional advantage of the MTJ-pair design is that the correlation problem of the MTJs is not a drawback anymore. Instead, a higher correlation will have improvements on the randomness quality, which will be discussed in Section 3.2.3.

3.2.1 Schematics and Generating Procedures

Figure 3.3 shows the schematic of the proposed design. The core part of the design includes two MTJs with the same parameters connected in series to produce one random bit. The principle idea is that both of the MTJs have equal probability of switching first, because the distributions of the switching time for each MTJ are independent and almost identical, and the probability that the switching time of the first MTJ is shorter than the second MTJ will be 50%.

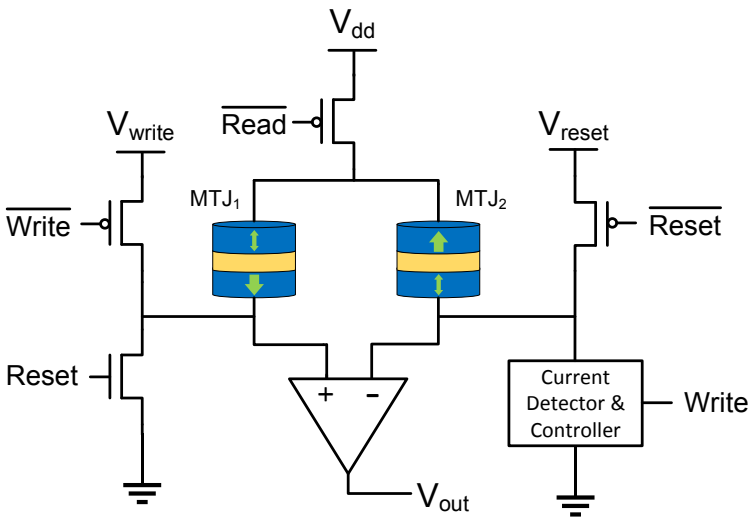

Figure 3.3 Proposed TRNG with symmetric MTJ-pair

The design works because of the following:

1) The two MTJs are connected in series, so the currents going through them are identical.
2) The parameters of the two MTJs are very similar to each other, so the two MTJs have the same properties such as the critical current and thermal stability factor.
3) The STT switching scheme ensures that the two MTJs switch individually and there's no correlation between them during the switching process.

However, it is impossible to know which MTJ switched first after the process if both of them switched. An alternative way is only allowing one of them to switch at a time. A current detector and controller is introduced to ensure only one of the two MTJs switches at a time in a vast majority of the cases.

To produce random numbers, the circuit needs to go through three phases: 1) a reset phase, 2) a write phase and 3) a read phase. Each phase takes 5 ns so the whole cycle

lasts 15 ns. One of the three control signals *Reset, Write* and *Read* is driven high while the others are held low in each phase correspondingly. Here the 3 phases are explained in detail:

1) Reset:

   In the reset phase, *Reset* is high and other control signals are low. MTJ_1 and MTJ_2 in Figure 3.3 are in series. The current flows from the free layer to the pinned layer for each MTJ until the MTJs which are in the AP state are switched to the P state. V_{reset} is high enough to ensure an almost deterministic switching. At the end of the reset phase, both MTJs are in the P state waiting for the probabilistic switching in the write phase.

2) Write:

   In the write phase, *Write* is high and other control signals are low. MTJ_1 and MTJ_2 are still in series, as well as the current detector and controller. V_{write} induces a switching current going from the pinned layer to the free layer. Once any one of the two MTJs switches to the AP state, the current in the path decreases suddenly since the resistance of the AP state is higher than that of the P state and the voltage remains the same. The current detector and controller responds to this change and cut off the circuit path immediately. Once the circuit is cut off, there's no current going through the MTJs and the write phase comes to an end, so the MTJ that didn't switch will not switch anymore. In this case, one MTJ will be in the P state and the other one will be in the AP state.

   However, it takes a small amount of time for the current detector and controller to cut off the circuit after the current changes, which cannot be completely ignored. If

the second MTJ happens to switch just after the first one switching in the rare case, both MTJs will be in the AP state.

Another rare case is when neither of the MTJs switches. Since the actual switching time follows a Gaussian distribution, but the pulse only lasts a finite period of time, there is the chance that neither MTJ switches before the pulse ends. If neither MTJ switches, both of them will remain in the initial P state.

In conclusion, there are actually three cases that might happen in the write phase:

- Case 1: Only one MTJ switches and the two MTJs end up in different states.
- Case 2: Both MTJs switch.
- Case 3: Neither MTJ switches.

Case 1 is common while cases 2 and 3 are rare.

3) Read

In the read phase, *Read* is high and other control signals are low. The current branches to the two MTJs and the path that has the MTJ with a higher resistance will have a lower current flowing through, and vice versa. A current comparator is used to determine the relative magnitude of the currents. Finally, the digital output at V_{out} will indicate the relative resistance of the MTJs. If V_{out} is low, then there is a lower current in the left path, which means that MTJ_1 has the higher resistance. If V_{out} is high, it means that MTJ_2 has the higher resistance.

For the cases that might happen in the write phase, the output is given slightly differently. When case 1 happens, the MTJs are in different states. The MTJ in the

AP state must have a higher resistance than the one in the P state (see Figure 2.4). Therefore, the output reveals which MTJ switched: if MTJ_1 switched, V_{out} is low. If MTJ_2 switched, V_{out} is high. When case 2 or case 3 happens, the MTJs are in the same state. However, the resistances of them are slightly different due to inevitable fabrication variations. The output will still reflect the relative resistance of the two MTJs: if the resistance of the MTJ_1 is higher, V_{out} is low; otherwise, V_{out} is high.

3.2.2 Discussion and Evaluation of Circuit Operations

Since the proposed design is based on the equal probability that either MTJ will switch first, we have to ensure that the probability of the rare cases 2 and 3 happening is small enough to ensure correct function.

3.2.2.1 The Current Detector and Controller

The delay of the current detector and controller should be short enough to prevent the second MTJ from switching as much as possible. The delay of the current detector and controller is defined as the time interval between when the first MTJ switches and when the circuit is cut off. The shorter the delay is, the less the probability that case 2 will happen. In our proposed design shown in Figure 3.4, the detector is based on a current mirror which can duplicate the current in the path using only two transistors. The current mirror can also duplicate the current by a certain proportion to save energy. The controller is based on a current-voltage converter and an amplifier, which converts the duplicated current to a digital voltage signal. The amplifier then regulates the voltage and provides an output. The I-V converter can be simply implemented by a resistor, and the amplifier

can be as simple as an inverter. Therefore, the change of current in the path is converted into the change of a digital control signal, and the signal is sent to cut off the circuit.

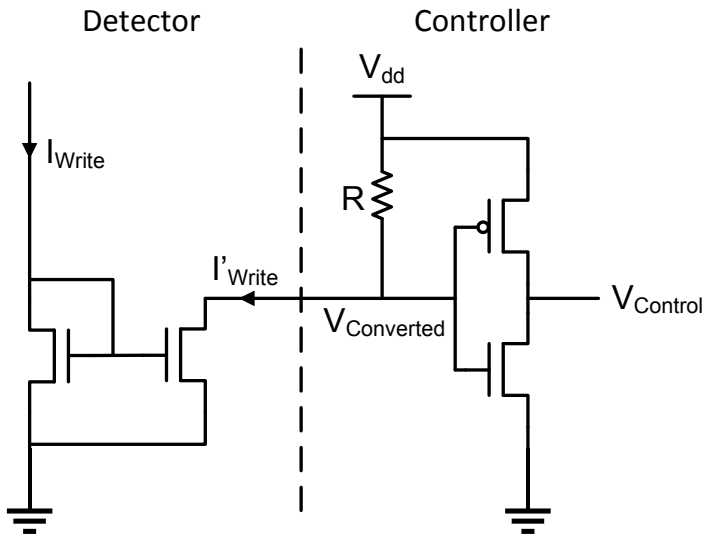

Figure 3.4 Proposed schematics of the current detector and controller

The simulation results show that the delay for the current detector and controller circuit described above is approximately 19.9 ps. Therefore, if the second MTJ happened to switch in less than 20 ps after the first one switched before the circuit is cut off, then both MTJs will end up in the AP state.

The probability of case 2 happening can be calculated theoretically as follows: the actual switching time can be taken to be a Gaussian distribution with a mean of μ = 2.72 ns and a standard deviation of σ = 1.28 ns, as shown in Figure 3.5. The distribution of the switching interval, which is the difference of the two independent Gaussian distributions, is also Gaussian. Since the two distributions are identical, the difference of the two distributions has a mean of $\mu' = \mu - \mu = 0$ and a standard deviation of $\sigma' = \sqrt{\sigma^2 + \sigma^2}$ = 1.81 ns. Therefore, the probability that the switching interval lies between $\pm$ 20 ps is 0.88%.

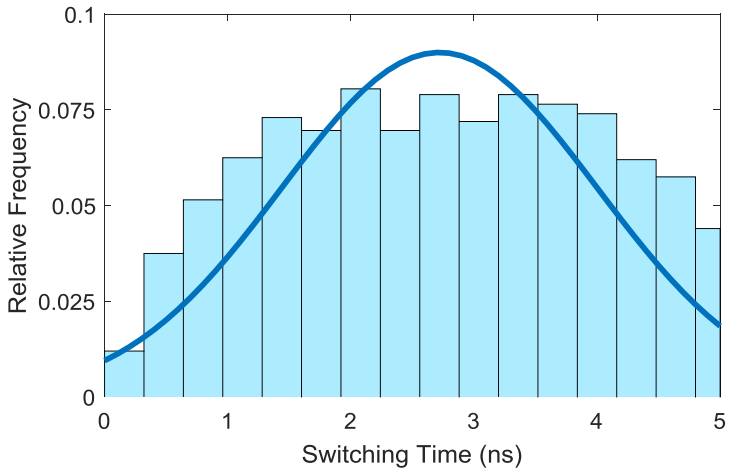

Figure 3.5 The distribution of the actual switching time for an MTJ

3.2.2.2 The Switching Pulse Width

Another issue is that neither MTJ might switch since the pulse only lasts a finite period of time but the actual switching time is Gaussian distributed. Although increasing the duration of the pulse can reduce the probability that case 3 happens, the generation speed and the power consumption are also concerns. A moderate pulse length of 5 ns makes this undesirable case a rare event while maintaining fast operation. The probability that one MTJ will not switch in 5 ns is 3.77% (Figure 3.5). Since the switching times for the two MTJs are independent, the probability that neither of them switches is approximately $(3.77\%)^2$ = 0.142% in theory.

The simulation results verified the calculation by showing an approximately 0.9% probability of case 2 happening, and a less than 0.2% probability of case 3 happening. The total probability that the two rare cases 2 and 3 happen is approximately 1%.

3.2.3 Discussion of Correlation Issues

Due to fabrication limitations, the parameters of the two MTJs are slightly different, which affects the probability that each of the three cases happens. The critical switching current is proportional to the size of the free layer,

$$I_{c0} = k \cdot t_{sl} \cdot Area\,, \tag{3.2}$$

so the MTJ with a smaller size has a smaller critical current. The series connection of the two MTJs ensures that the currents (I) flowing through them are the same, and according to (2.3), the smaller MTJ has a shorter mean switching time and thus is more likely to switch first.

For example, if MTJ_1 in Figure 3.3 is slightly smaller than MTJ_2, then MTJ_1 is more likely to switch first, and the probability that V_{out} is low is slightly higher than the probability that V_{out} is high. The difference of the two MTJs leads to a probability bias that will undermine the quality of the random sequences.

However, the correlation in the MTJs actually helps to relieve this problem. Due to the correlations in the fabrication process, some parameters, such as the dimensions, of MTJs fabricated close to each other will be similar, and this leads to correlations in the mean switching time of the two MTJs. As the theory suggests, when two independent variables have identical distributions, the probability that one variable is smaller than the other is 50%. The more similar the two distributions are, the closer the probability will be towards 50%. Therefore, the more correlations the two MTJs have, the more similar the distributions of the switching time will be.

To analyze the correlation, a simplified mathematical model can be built assuming that the mean switching time τ in (2.3) for the two MTJs follows a multivariate Gaussian distribution impacted by the fabrication effects. The distribution is determined by the mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ [41].

$$\boldsymbol{X} \sim \boldsymbol{\mathcal{N}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{3.3}$$

$$\boldsymbol{X} = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}, \boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix} \tag{3.4}$$

We aim for a pair of MTJs with the same parameters, so the expected value of mean switching time μ is the same for both MTJs, as well as the standard deviation σ.

$$\begin{gathered} \mu_1 = \mu_2 = \mu, \boldsymbol{\mu} = \begin{pmatrix} \mu \\ \mu \end{pmatrix}, \\ \sigma_1 = \sigma_2 = \sigma, \sigma_1^2 = \sigma_1\sigma_2 = \sigma_2^2, \boldsymbol{\Sigma} = \sigma^2 \cdot \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \end{gathered} \tag{3.5}$$

Note that the “expected value of mean switching time” here is the mean value of the mean switching time of multiple devices, which is determined by the process parameters and design objectives before fabrication. The standard deviation of the mean switching time is relatively small (see Figure 3.6). While the “mean switching time” is the mean value of the actual switching time of a certain device in multiple switching processes, which is determined by the material parameters and the size dimensions after fabrication. The standard deviation of the switching time is relatively large (see Figure 3.5). In conclusion, the first distribution is for a set of devices, while the second distribution is for one device and is slightly different for each device.

Finally, there are only three independent parameters, namely the expected value of mean switching time μ, the standard deviation of mean switching time σ, and the correlation coefficient ρ:

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu \\ \mu \end{pmatrix}, \sigma^2 \cdot \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right) \tag{3.6}$$

Under the simulation conditions, the expected value of mean switching time is μ = 2.72 ns, and the MTJs have a variation of 6.28% with respect to the expected value, which makes the standard deviation $\sigma = 6.28\% \times$ 2.72 ns.

The population correlation coefficient ρ reflects the correlations between the two MTJs. Since the correlation is non-negative, $0 \leq \rho \leq 1$. When $\rho = 0$, there's no correlation. And when $\rho = 1$, the two MTJs are identical. The correlation coefficient mainly depends on the limited accuracy during the fabrication process. For analytical purpose, we simulated different levels of correlation with $0 \leq \rho \leq 1$.

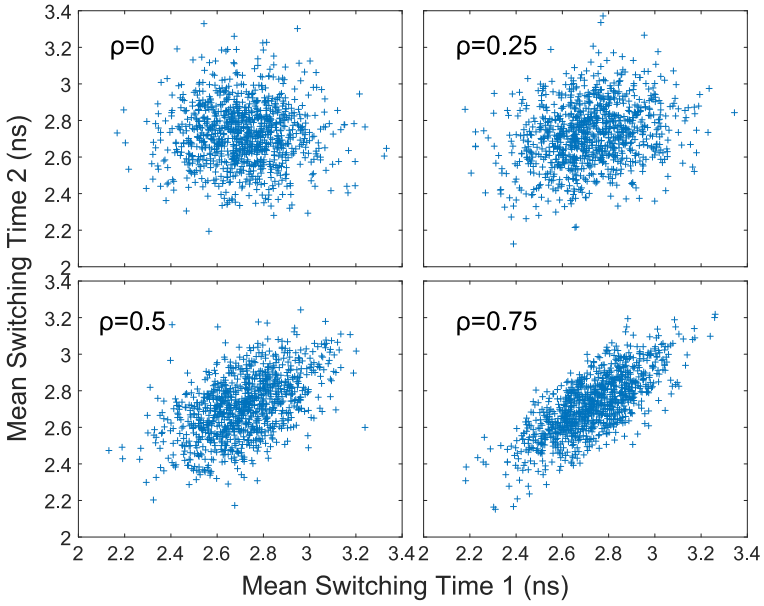

Figure 3.6 The mean switching time for two MTJs with different correlation coefficients

After all three parameters are set, the mean switching time for the two MTJs can be generated according to the multivariate Gaussian distribution. The random number generation will then be conducted based on the known distributions of the switching time of the two MTJs. The subfigures in Figure 3.6 illustrate the correlation in two switching times for ρ = 0, 0.25, 0.5 and 0.75, respectively, with 1000 samples for each case.

We have to point out that the correlations exist only during the fabrication process. After fabrication, the distributions of the switching time of both MTJs are determined. During each switching process, there are no correlations between the two MTJs since they switch individually, nor are there correlations in the time domain since the switching is based on independent quantum effects.

3.3 The Self-calibration Design

The basic idea of the self-calibration design is to suppress the switching probability variation by a self-calibration from the two-step switching process. In the basic generator, the probabilistic switching process is only conducted once, leading to a large probability bias which undermines the quality of the randomness. Here a two-step switching process shown in Figure 3.7 is proposed to decrease the bias.

For a particular device, the switching probability p is biased by a coefficient c due to device variations, so the actual switching probability is approximately $c \cdot p$. In the pre-write step, the MTJ is switched towards the AP state from the initial P state with a probability set to p_1. While in the calibration step, the MTJ is switched in the opposite direction towards the P state with a probability set to p_2. However, if the MTJ did not switch during the first step,

it will definitely remain in the P state during the second step. The actual switching probabilities in the presence of bias are approximately $c \cdot p_1$ and $c \cdot p_2$, respectively.

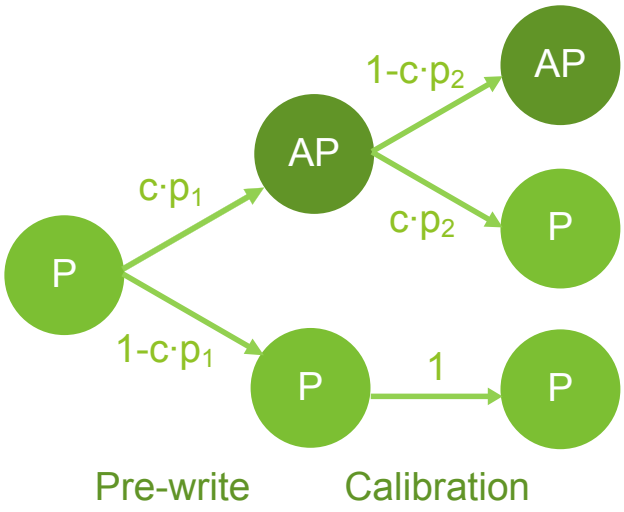

Figure 3.7 State transition diagram of the self-calibration design

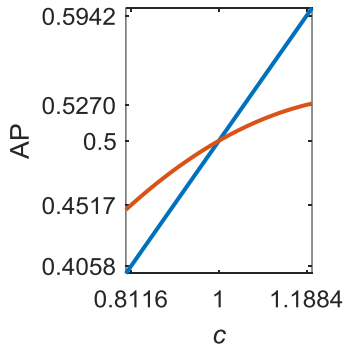

Figure 3.8 Suppressed probability variation by the two-step self-calibration

When $p_1 = 0.5$, the method reduces to the conventional one-step switching. In this case, $p_2 = 0$ and $AP = 0.5c$ (the straight line in Figure 3.8), which is linearly related to the bias coefficient c, and the probability variation lies in the range of $(0.4058,0.5942)$. In the proposed design, we choose $p_1 = 0.8$, and the detailed calculations for choosing the

parameters can be found in Appendix C. In this case, $p_2 = 0.375$ and $AP = -0.3c^2 + 0.8c$ (the curved line in Figure 3.8), and the probability variation reduced to the range of $(0.4517, 0.5270)$. Figure 3.8 illustrates the more than half suppression of the probability variation brought by the proposed method.

In conclusion, the self-calibration method leverages the calibration step to compensate for the probability inaccuracy occurred in the pre-write step without using other devices. Moreover, this method is also applicable to many other types of devices as long as they have controllable stochastic behavior in both state transitions, and the transition probabilities of both directions are affected by the same coefficient.

3.3.1 Schematics and Generating Procedures

Based on the discussions above, the schematic for the proposed self-calibration method was designed as shown in Figure 3.9. To produce random numbers, the circuit needs to go through four phases: 1) a reset phase, 2) a pre-write phase, 3) a calibrate phase and 4) a read phase. Each phase takes 5 ns so the whole cycle lasts 20 ns. There are two control signals: *OP* and *Read.* In the first three phases, *OP* is driven high and *Read* is held low. In the read phase, *Read* is driven high and *OP* is held low. Here the 4 phases are described in detail:

1) Reset:

   In the reset phase, V_{reset} is applied to the circuit. The current flows from the free layer to the pinned layer. V_{reset} is higher enough than V_b to ensure an almost deterministic switching to the P state if it ended in the AP state in the previous cycle. At the end of the reset phase, the MTJ is in the P state waiting for the probabilistic switching in the following two phases.

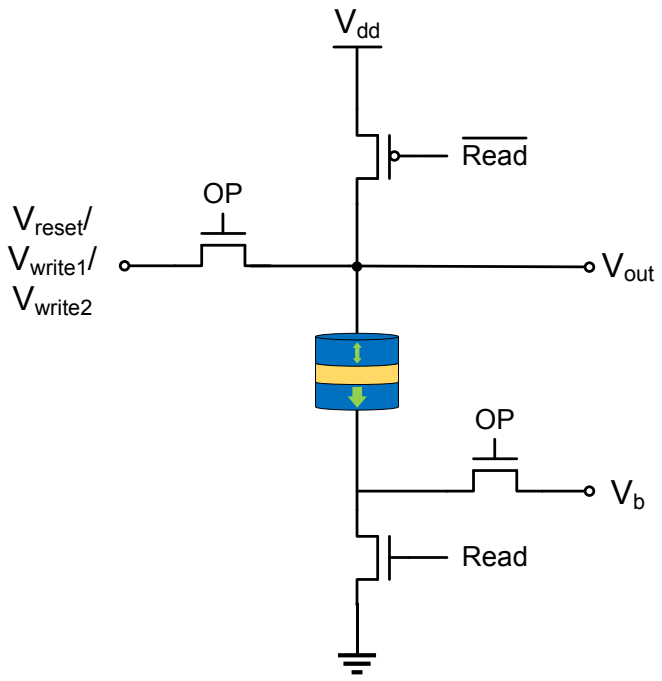

Figure 3.9 Proposed TRNG for the self-calibration design

2) Pre-write:

In the pre-write phase, V_{write1} is applied to the circuit, and it is lower than V_b to induce a switching current going from the pinned layer to the free layer. The voltages target a switching probability of p_1 in 5 ns for the MTJ. At the end of the pre-write phase, the MTJ will change to the AP state if it switched; otherwise, it will remain in the P state.

3) Calibrate:

In the calibrate phase, V_{write2} is applied to the circuit, and it is higher than V_b to induce a reverse switching current. If the MTJ didn't switch in the pre-write phase, it will still remain in the P state during this phase. If it switched to the AP state in the pre-write phase, the voltages target a switching probability of p_2 in 5 ns for the MTJ. At the end of the calibrate phase, the MTJ will be in the AP state if it switched

only in the pre-write phase; or, it will be in the P state if it switched in both of the pre-write phase and the calibrate phase, or neither of the two phases.

4) Read:

   In the read phase, the current flows from V_{dd} to *GND* passing through the MTJ. Depending on the resistance state of the MTJ, V_{out} will differ. Then a sense amplifier can observe the difference and amplify it to produce a digital output. Note that the sense amplifier can be implemented in various ways and is not included in the schematic.

After a cycle of four phases, one random bit with self-calibration is generated. Note that, if the calibrate phase is omitted, the schematic in Figure 3.9 becomes the basic generator.

Waveforms of some key nodes and the sizes of some key transistors in the three schematics of the TRNG designs can be found in Appendix D.

3.4 The Quality Improvement Circuits

A frequency of 50% 1's in random sequences is always desired. However, true random sources are often flawed to some extent as stated in Section 2.3. One of the simple ways to regulate frequency is using XOR (or XNOR) gates to process the biased sequences from a true random source with some unbiased sequences from a deterministic source [42] [43]. The true random source provides the nondeterministic property while the deterministic source ensures an unbiased frequency of 1's. Therefore, the probability bias issue can be mitigated by regulating the frequency of 1's occurred in the sequences closer to 50%, while keeping the true randomness.

Using probabilistic logic, the theory of using XOR gates to improve the quality of random sequences in terms of frequency can be given [5]. If the inputs are independent, Boolean function $C = A \,\mathrm{XOR}\, B = \bar{A}B + A\bar{B}$ corresponds to $c = (1-a)\cdot b + a\cdot(1-b)$ where $a = P(A=1)$, $b = P(B=1)$ and $c = P(C=1)$. Suppose A is the sequence from the true random source with a probability bias δ, so $a = 0.5 + \delta$. Then suppose B is the sequence from a deterministic source used for improvement. We then have

$$c = \big(1-(0.5+\delta)\big)\cdot b + (0.5+\delta)\cdot(1-b) = 0.5 + (1-2b)\delta. \tag{3.7}$$

Since $0 < b < 1$, then $-1 < 1-2b < 1$, and finally $0.5-\delta < c < 0.5+\delta$. Therefore, the quality of the random sequences is improved from $a = 0.5+\delta$ to $0.5-\delta < c < 0.5+\delta$, even though the sequence from the deterministic source may be unbiased to some extent.

If the sequence from the deterministic source has a probability of exactly 0.5, then the result will be the best since $c = 0.5$ when $b = 0.5$. Moreover, the non-frequency-related properties are not compromised since the true randomness from the random source is still kept in the process. We can see that they are actually improved in the following discussions.

The simplest sequence generator with nearly equal proportions of 1's and 0's is a 1-bit counter, and it can be simply implemented by a flip-flop. The output of a 1-bit counter is a sequence of alternating 1's and 0's (Figure 3.10). For an XOR gate, a 1 at one input will let the other input become its logical complement at the output, while a 0 at one input will let the other input remain its value at the output. So in short, the function of the XOR gate with a 1-bit counter is flipping every other bit in the original sequence. Intuitively, it can make the frequency of 1's in a biased sequence turn closer to 50% and break sub-

sequences of consecutive 1's or 0's, which improves the randomness quality of both frequency-related and non-frequency-related properties.

Any sequence generator producing approximately 50% of 1's and 50% of 0's in the sequence can be implemented. Therefore, we propose a quality improvement circuit (QIC) as shown in Figure 3.10. One of the inputs of the XOR gate is the original sequence from a TRNG (Seq In), and the other input is from the sequence generator. The output of the XOR gate is the sequence with improved randomness quality (Seq Out). The generator can be chosen with the consideration of the quality requirements and hardware cost. Examples include 1-bit counters, 2-bit counters and 4-bit LFSRs.

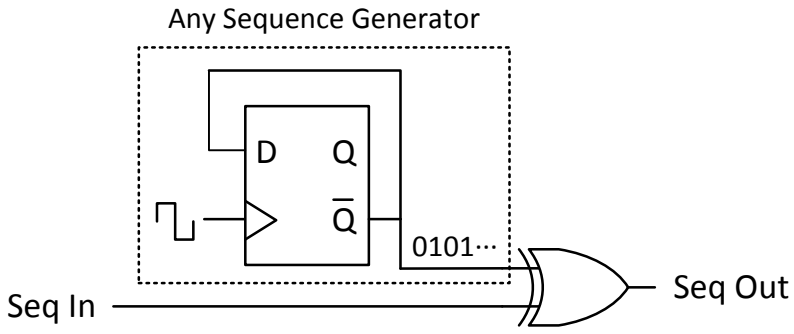

Figure 3.10 The quality improvement circuit for random number generators

The QIC can be applied to both categories of flawed random sources discussed in Section 2.3 and any random number generators. It is a general method to improve the quality of random sequences without complicated circuits.

Chapter 4: Simulation, Evaluation and Comparisons

To evaluate the proposed designs, random sequences were generated for the Transport Layer Security or Secure Sockets Layer (TLS/SSL) cryptographic protocol, and were evaluated using the National Institute of Standards and Technology (NIST) statistical test suite.

4.1 On the Parallel Design

As explained in Section 2.2, 1000 sequences with 256 bits in each are needed for the test suite. For each N value, the proposed generation procedure was repeated $\frac{256}{N}$ times, and each MTJ was used $\frac{256}{N}$ times to generate $\frac{256}{N}$ random bits, where N is the number of MTJs in the array. N was chosen to be 1, 2, 4, 8, etc., which are exact devisors of 256, so as to avoid wasting generated bits. After one sequence of 256 bits is generated, a new set of N MTJs is used to generate the next sequence. Altogether 1000 sequences were generated for each N value.

The sequences were evaluated using the test suite and the results are listed in Table 4.1. The values in bold denote that the generator fails in a particular test (the same for all tables showing statistical quality pass rates). However, note that, one generator must pass all tests to prove its functionality and the average values in the last three rows are for comparison purposes only. Here “MTJN” denotes N parallel MTJs used in the proposed parallel design. When using at least 16 MTJs, the pass rates for all tests are no less than 0.981, which means that the corresponding generators can pass all 9 randomness tests.

Generator		MTJ1	MTJ2	MTJ4	MTJ8	MTJ16	MTJ32	MTJ64
Freq-related	Frequency Test	**0.913**	**0.966**	**0.979**	0.981	0.985	0.989	0.991
	Block Frequency	**0.969**	0.991	0.987	0.987	0.989	0.991	0.997
	Cumulative Sums (1)	**0.924**	**0.968**	**0.976**	**0.979**	0.990	0.986	0.990
	Cumulative Sums (2)	**0.925**	**0.965**	**0.978**	0.985	0.987	0.986	0.991
Non-freq	Runs	0.984	0.990	0.986	0.990	0.992	0.991	0.992
	Longest Run	**0.974**	0.985	0.987	0.988	0.996	0.995	0.993
	Approximate Entropy	**0.961**	**0.973**	**0.979**	0.984	0.985	0.989	0.994
	Serial (1)	**0.968**	**0.980**	0.983	0.982	0.989	0.990	0.992
	Serial (2)	0.988	0.986	0.993	0.986	0.988	0.994	0.990
Average of frequency-related		0.933	0.973	0.980	0.983	0.988	0.988	0.992
Average of non-frequency		0.975	0.983	0.986	0.986	0.990	0.992	0.992
Average of all tests		0.956	0.978	0.983	0.985	0.989	0.990	0.992

Table 4.1 Statistical quality pass rates of the parallel design with different numbers of MTJs

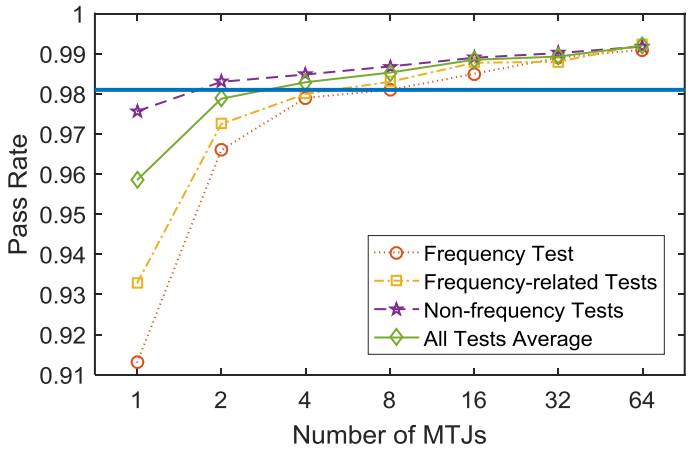

Figure 4.1 Statistical quality pass rates of the parallel design with different numbers of MTJs

Therefore, it was shown by the statistical test suite that using at least 16 MTJs in this proposed TRNG can generate high-quality 256-bit random sequences. The four curves in Figure 4.1 show the pass rate trends for different categories of tests, and illustrate the increasing quality of the generators for different categories of tests with the increasing

number of MTJs used. The bold horizontal line is the threshold of 0.981 for passing the tests (the same for all figures showing statistical quality pass rates).

In addition, the Tausworthe generators (TGs) and LFSRs [44] [45] were tested for comparison purposes and the results are listed in Table 4.2. “TG28” is the Tausworthe generator with a period of 2^{28}-1, “LFSR52” is a linear-feedback shift register with a period of 2^{52}-1, and “CTG88” and “CTG113” are combined Tausworthe generators with periods of nearly 2^{88} and 2^{113}, respectively. The results show that the simple Tausworthe generator with a period of 2^{28}-1 and the LFSR with a period of 2^{52}-1 have relatively poor randomness quality. However, with the more complex combined Tausworthe generators (CTGs), the statistical quality is improved. The comparison results are shown in Figure 4.2. Using 16 MTJs in the parallel design can produce random sequences with a similar randomness quality as using either of the CTGs, while using 32 and 64 MTJs will lead to better results. However, the test suite can only evaluate the statistical properties of the random sequences. The advantages of a TRNG over a PRNG are not shown from the numerical results: the MTJ-based generators generate true random numbers and are inherently better for cryptographic applications.

As a trade-off between quality, speed and area, using 16 MTJs is sufficient to satisfy basic quality concerns while providing a fast generation speed. 32 or more MTJs can be implemented in applications that require a higher security level where a better quality or a faster speed is needed. However, more hardware resources are required as the number of MTJs increases.

Generator		TG28	LFSR52	CTG88	CTG113
Freq-related	Frequency Test	**0.954**	**0.955**	0.983	0.988
	Block Frequency	**0.960**	**0.969**	0.988	0.994
	Cumulative Sums (1)	**0.954**	**0.952**	0.987	0.987
	Cumulative Sums (2)	**0.953**	**0.942**	0.982	0.991
Non-freq	Runs	**0.973**	**0.950**	0.994	0.994
	Longest Run	**0.964**	**0.917**	0.992	0.988
	Approximate Entropy	**0.965**	**0.893**	0.989	0.990
	Serial (1)	**0.930**	**0.772**	0.988	0.985
	Serial (2)	**0.951**	**0.871**	0.994	0.988
Average of frequency-related		0.955	0.955	0.985	0.990
Average of non-frequency		0.881	0.957	0.991	0.989
Average of all tests		0.973	0.956	0.989	0.989

Table 4.2 Statistical quality pass rates of some PRNGs

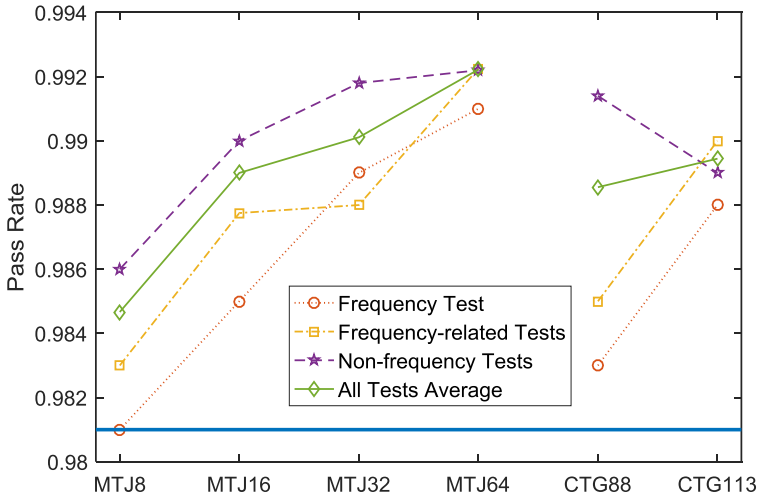

Figure 4.2 Comparisons of the randomness quality between the MTJ-based TRNGs and the combined Tausworthe generators

4.2 On the MTJ-pair Design

Again, 1000 sequences with 256 bits in each are needed for evaluations. For each correlation coefficient ρ, the proposed generation procedure was repeated 256 times to obtain a 256-bit sequence. After one sequence is generated, a new pair of MTJs is used to generate the next sequence. Altogether 1000 sequences were generated for each ρ value.

The sequences were evaluated using the test suite and the results are listed in Table 4.3. Figure 4.3 shows the pass rate trends for different categories of tests. The four curves illustrate the quality improvement of the generators with an increasing large correlation coefficient, showing that this design is especially suitable for MTJs with highly correlated

Generator, with the ρ of		0	0.25	0.5	0.75	0.875	0.9	0.925	0.95	0.999
Freq-related	Frequency Test	**0.838**	**0.895**	**0.924**	**0.962**	**0.970**	**0.978**	0.981	0.983	0.992
	Block Frequency	**0.940**	**0.955**	**0.965**	0.983	0.986	0.988	0.988	0.992	0.990
	Cumulative Sums (1)	**0.852**	**0.896**	**0.935**	**0.966**	**0.972**	**0.977**	0.988	0.987	0.991
	Cumulative Sums (2)	**0.848**	**0.895**	**0.931**	**0.965**	**0.975**	**0.979**	0.985	0.987	0.991
Non-freq	Runs	**0.956**	**0.979**	0.984	0.991	0.990	0.993	0.992	0.989	0.992
	Longest Run	**0.931**	**0.955**	**0.973**	0.983	0.986	0.985	0.985	0.994	0.991
	Approximate Entropy	**0.901**	**0.942**	**0.950**	**0.970**	0.988	0.983	0.987	0.993	0.992
	Serial (1)	**0.935**	**0.967**	**0.974**	0.984	0.987	0.985	0.987	0.995	0.993
	Serial (2)	0.984	0.986	0.990	0.991	0.989	0.990	0.991	0.996	0.990
Average of frequency-related		0.870	0.910	0.939	0.969	0.976	0.981	0.986	0.987	0.991
Average of non-frequency		0.941	0.966	0.974	0.984	0.988	0.987	0.988	0.993	0.992
Average of all tests		0.909	0.941	0.958	0.977	0.983	0.984	0.987	0.991	0.991

Table 4.3 Statistical quality pass rates of the MTJ-pair design with different correlation coefficients

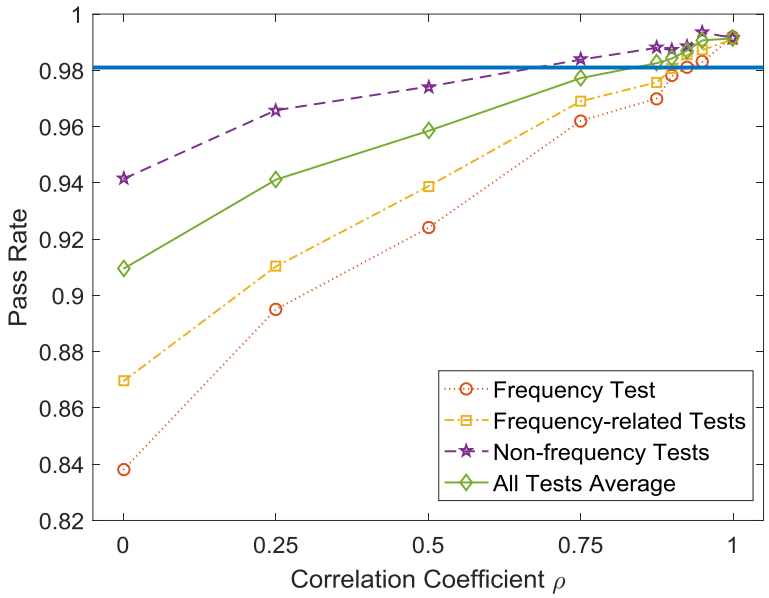

Figure 4.3 Statistical quality pass rates of the MTJ-pair design with different correlation coefficients

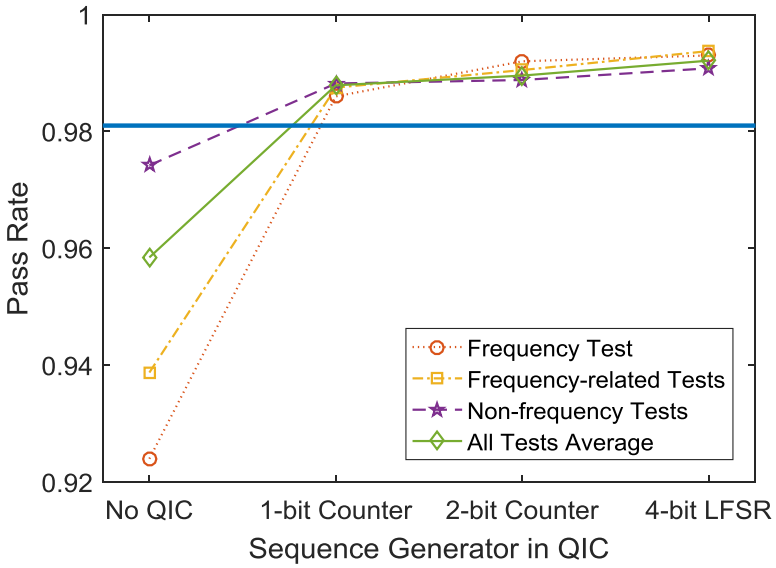

Figure 4.4 Statistical quality pass rates of the MTJ-pair design with different QICs

physical properties. Actually, all 9 tests are passed when $\rho \geq 0.925$. However, if the correlation of the MTJs is less significant and some tests fail, the quality improvement circuit (QIC) proposed in Section 3.4 can be added.

For example, when $\rho = 0.5$, the sequences fail the frequency test with a pass rate of 0.924. Moreover, none of the frequency-related tests are passed and only two of the non-frequency tests are passed. However, with the implementation of the QIC, the randomness quality improves significantly, as shown in Figure 4.4. Even combined with the simplest 1-bit counter, the output sequences can pass all 9 tests. The use of a 2-bit counter or a 4-bit LFSR will improve the quality even more, although the additional quality improvement is relatively small.

	PVT corners	Mean switching time μ (ns)
i.	FF, high voltage (1.1x, 825 mV), 0 °C	2.66 (lowest)
ii.	TT, nominal voltage (750 mV), 27 °C	2.72 (nominal)
iii.	SS, low voltage (0.9x, 675 mV), 70 °C	2.88 (highest)

Table 4.4 PVT corner test for mean switching time

To test the variation-resilience of the design, experiments were conducted with different combinations of the process, voltage and temperature. First, the mean switching times with different PVT corners were tested. The simulation results are shown in Table 4.4. Next, with the correlation coefficient ρ set to 0.95, the random sequences are generated under these PVT corners. Figure 4.5 shows that with all combinations of the process, voltage and temperature, the generated sequences can pass all tests with similar pass

rates. Therefore, it is confirmed that the PVT corners have only minor effects on the randomness quality, and the proposed design has an intrinsic resistance to all major variations in the circuit.

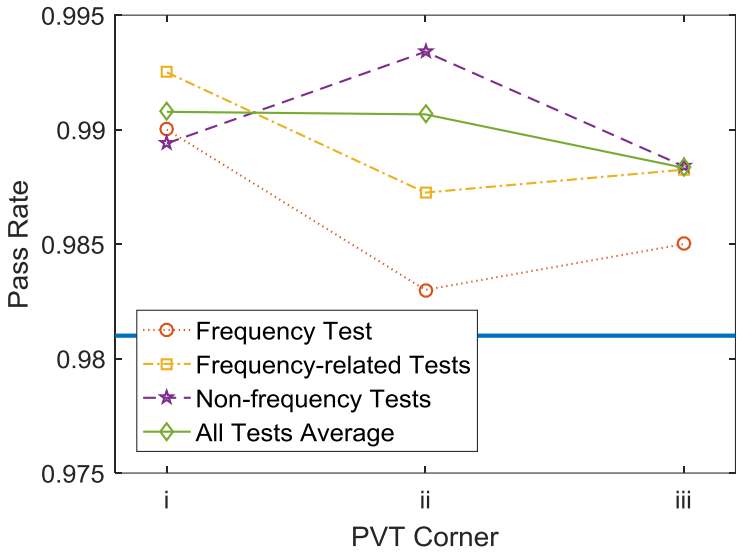

Figure 4.5 Statistical quality pass rates of the MTJ-pair design with different PVT corners

If the CMOS process parameters change to Fast or Slow or the operating voltage varies from 0.9 to 1.1 times the nominal voltage, a TRNG based on a single MTJ switching will have a probability bias of more than $\pm$ 10% (see Figure 2.8 and Figure 2.9), which will severely undermine the randomness quality. Compared with other TRNG designs based on a single MTJ switching, the main advantage of the MTJ-pair design is its resistance to variations. Since all variations will affect both MTJs in the circuit to almost the same extent, the difference between the parameters of the two MTJs will still be small. The random number generation depends on the similarity of the statistical distribution of the two MTJs instead of the actual value of a certain parameter, so the quality of the generated sequences will remain unimpaired (as long as the variation is moderate keeping the mean switching time within the expected range).

The related test results discussed above are listed in Table 4.5.

Generator		$\rho = 0.5$ with 1-bit counter	$\rho = 0.5$ with 2-bit counter	$\rho = 0.5$ with 4-bit LFSR	$\rho = 0.95$ with corner i	$\rho = 0.95$ with corner iii
Freq-related	Frequency Test	0.986	0.992	0.993	0.990	0.985
	Block Frequency	0.990	0.987	0.994	0.995	0.993
	Cumulative Sums (1)	0.988	0.990	0.994	0.993	0.987
	Cumulative Sums (2)	0.986	0.993	0.994	0.992	0.988
Non-freq	Runs	0.987	0.985	0.991	0.986	0.984
	Longest Run	0.989	0.992	0.994	0.991	0.992
	Approximate Entropy	0.987	0.988	0.991	0.992	0.991
	Serial (1)	0.992	0.986	0.988	0.991	0.985
	Serial (2)	0.986	0.993	0.990	0.987	0.990
Average of frequency-related		0.988	0.991	0.994	0.993	0.988
Average of non-frequency		0.988	0.989	0.991	0.989	0.988
Average of all tests		0.988	0.990	0.992	0.991	0.988

Table 4.5 Statistical quality pass rates of the MTJ-pair design with various parameters

4.3 On the Self-calibration Design

Once more, 1000 sequences with 256 bits in each are needed for evaluations. The generation procedure was repeated 256 times to obtain a 256-bit sequence. After one sequence is generated, a new MTJ device is used to generate the next sequence. Altogether 1000 sequences were generated for each randomness test.

Figure 4.6 shows the pass rate for different categories of tests, and the results imply that the random sequences generated from the proposed two-step self-calibration design can pass all randomness tests without any post-processing because of the limited probability variations. However, the basic generator can hardly pass any tests due to the excess

device fabrication variations. Moreover, the parallel designs from Section 3.1 are compared in Figure 4.6 as well. The self-calibration design has a similar quality as using 16 MTJs in the parallel design, but uses far fewer transistors, so it can greatly save hardware.

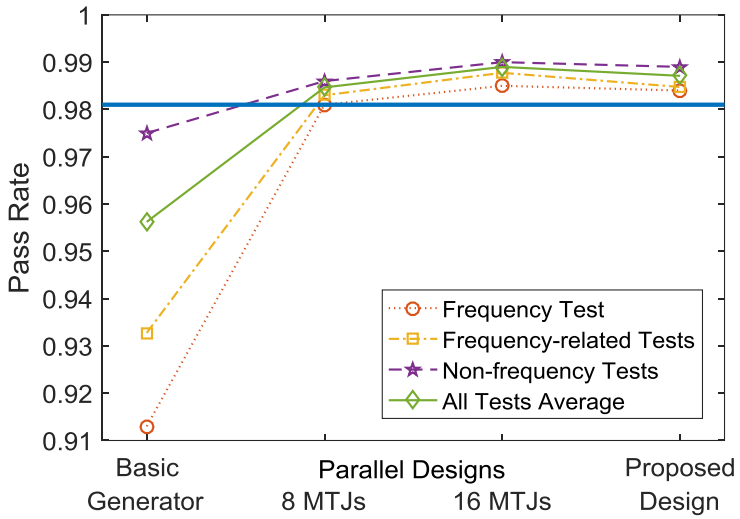

Figure 4.6 Statistical quality pass rates of the self-calibration design in comparison with other designs

The different process, voltages and temperatures will affect the switching probability as well. Therefore, it is worthwhile to examine the variation-resilience of the designs. The same PVT corners as in Table 4.4 were selected and the sequences were generated by the basic generator and the proposed self-calibration TRNG. After processed by the quality improvement circuit with a 1-bit counter, the randomness quality was tested. Figure 4.7 shows the advantage of the proposed design: when the QIC with a 1-bit counter is applied, the proposed design can pass all tests in every process corner, so it is insensitive to all major variations in the circuit. Note that corner ii is actually the nominal corner and its results are already shown in Figure 4.6.

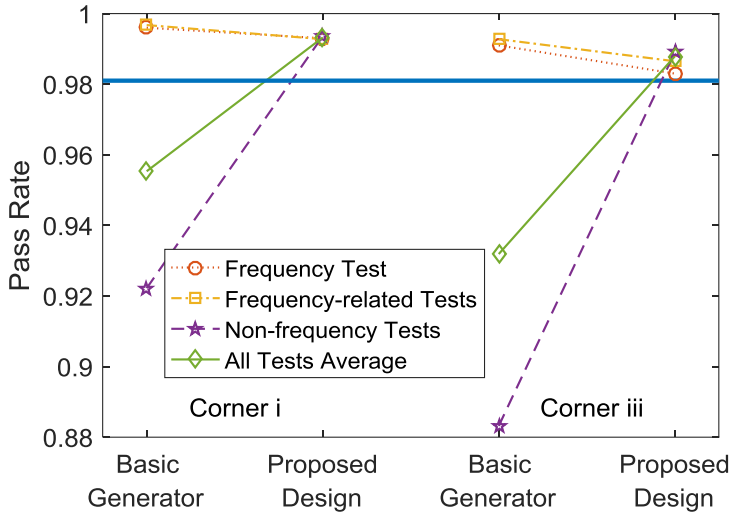

Figure 4.7 Statistical quality pass rates of the self-calibration design with different PVT corners

The related test results discussed above are listed in Table 4.6.

Generator		proposed, corner ii, no QIC	basic, corner i, with 1-bit counter	proposed, corner i, with 1-bit counter	basic, corner iii, with 1-bit counter	proposed, corner iii, with 1-bit counter
Freq-related	Frequency Test	0.984	0.996	0.993	0.991	0.983
	Block Frequency	0.984	0.998	0.993	0.995	0.994
	Cumulative Sums (1)	0.983	0.996	0.993	0.994	0.988
	Cumulative Sums (2)	0.988	0.997	0.992	0.991	0.981
Non-freq	Runs	0.988	**0.895**	0.993	**0.855**	0.987
	Longest Run	0.994	**0.970**	0.994	**0.973**	0.986
	Approximate Entropy	0.988	**0.919**	0.993	**0.861**	0.991
	Serial (1)	0.987	**0.874**	0.996	**0.789**	0.991
	Serial (2)	0.988	**0.953**	0.992	**0.938**	0.990
Average of frequency-related		0.985	0.997	0.993	0.993	0.987
Average of non-frequency		0.989	0.922	0.994	0.883	0.989
Average of all tests		0.987	0.955	0.993	0.932	0.988

Table 4.6 Statistical quality pass rates of the self-calibration design and the basic generator

In conclusion, when using the self-calibration design and the simplest QIC together, high-quality random sequences for cryptographic protocols can be generated with high variation-resilience. The resilience comes from the compensation in the self-calibration: if a higher than normal probability occurs in the pre-write phase, the probability to make the device back into the initial state will also be higher than normal in the calibration phase, and vice versa, which offsets the variations.

4.4 On the General Flawed Random Sources

To provide a general guideline for choosing the sequence generator in the QIC, the two categories of flawed random sources with different sequence generators in the QIC were implemented and the output sequences were evaluated by the test suite. The randomness quality of the various combinations of random sources and QICs can be referred to when implementing a TRNG according to quality requirements.

First, random sources with a fixed bias were implemented and different probability biases were introduced. Without any QICs, it is shown that a mere 0.5% probability bias can be tolerated, which means that only sequences with a probability of 49.5% to 50.5% can pass all tests. However, adding a simple 1-bit counter and an XOR gate makes the probability bias tolerance increase by 10 times from 0.5% to 5%. With more complicated QIC used, the tolerance level is further raised to 15% with a 2-bit counter, or 18% with a 4-bit LFSR (Column 2 in Table 4.7).

Second, for random sources with a certain variation, similar generations and tests were done with different combinations of variation levels and QICs. The results show that only when $d \leq 3$ can the output sequences pass all tests, or the standard deviation should not

exceed 1.5% if no QIC is implemented. This result verifies that a generator based on a single MTJ switching is not sufficient to pass all tests ($d = 6.28 > 3$). Similarly, QICs with different sequence generators are added to the random sources. A 1-bit counter can allow a d value up to 10, while a 4-bit LFSR can allow a d value up to 21. The tolerance of d value is expanded by 3 to 7 times (Column 3 in Table 4.7).

Generator in QIC	Tolerance for fixed bias (δ)	Tolerance for certain variation (d)
None	0.5%	3
1-bit Counter	5%	10
2-bit Counter	15%	18
4-bit LFSR	18%	21

Table 4.7 Tolerance levels for different sequence generators in QICs

For random sources with a high bias or variation, the randomness quality improvement introduced by the QIC can be better understood from illustrations. Figure 4.8 shows the results of the random sources with fixed biases: solid lines are for 10% bias and broken lines are for 20% bias. Without QICs, neither of the two scenarios can pass the tests and the pass rates are very low, so the results with no QIC are omitted in the figure to avoid disproportionality. With a 10% bias, a 1-bit counter is not sufficient to make all tests pass since the maximum tolerance level is 5% for the 1-bit counter, however, it does bring the pass rates very close to the threshold. A 2-bit counter will make all tests pass since the capability for it is 15%. A 4-bit LFSR will not improve the randomness quality further since a 2-bit counter has already made the sequences very good in terms of randomness. For 20% bias, none of the 3 QICs will improve the sequences to the level of passing the tests, though the randomness quality is improved significantly. Note that in Figure 4.8, it seems

that the one with a 4-bit LFSR is sufficient to pass the tests, but actually, one of the non-frequency tests fails.

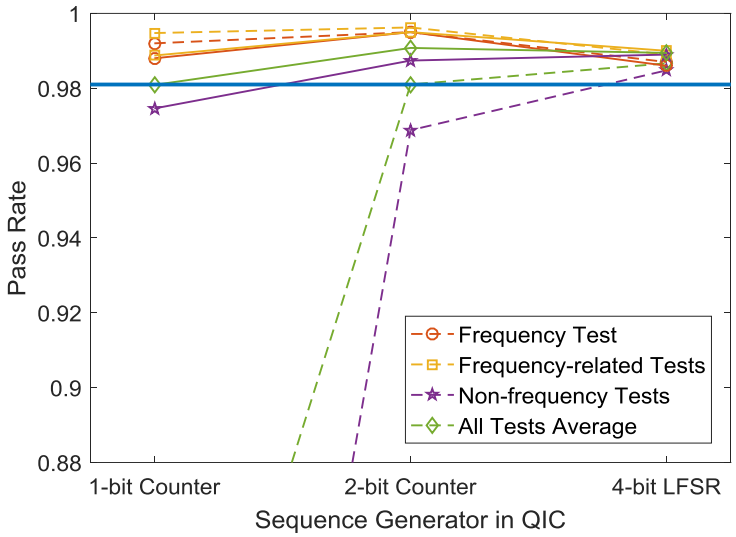

Figure 4.8 Statistical quality pass rates for random sources with fixed biases using different QICs

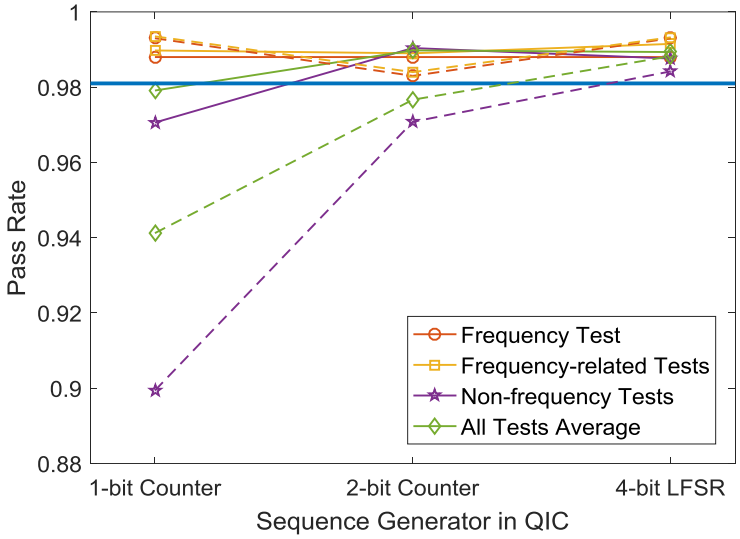

Figure 4.9 Statistical quality pass rates for random sources with certain variations using different QICs

For random sources with certain variations, the analysis is essentially the same. The solid lines are for d = 15 and broken lines are for d = 24 in Figure 4.9, which is very similar to Figure 4.8. Actually, all kinds of QICs can greatly improve the randomness quality, but whenever the quality is improved to a very high level, the additional improvement from a more complex QIC is minor.

The average pass rates of the random sources mentioned above are summarized in Table 4.8. Values in bold denote that the particular combination of random source and QIC passes all tests. According to Table 4.7 and Table 4.8, for generators or designs which cannot pass certain randomness tests, if the sources are flawed within the tolerance level, the QIC can make the sequences pass the tests. If the sources are highly flawed, the quality can at least be significantly improved. On the other hand, for the sources that can already pass the tests, the QIC can also be used to make the sequences of a higher randomness quality, but not with a major improvement.

Sequence Generator	δ = 10%	δ = 20%	d = 15	d = 24
None	58.4%	7.3%	79.2%	63.8%
1-bit Counter	98.1%	76.3%	97.9%	94.1%
2-bit Counter	**99.1%**	98.1%	**99.0%**	97.7%

Table 4.8 Average pass rates of some flawed random sources

The tolerance level analysis provides a guideline for choosing the sequence generator in the QIC: after knowing the bias/variation of a random source, the sequence generator in the QIC should be chosen in order to make sure the quality requirements are met while avoiding unnecessary hardware costs.

4.5 Comparisons of the Proposed Designs and Some Other Generators

As validated in Sections 4.1 to 4.3, each of the three proposed MTJ-based TRNGs can generate high-quality random numbers for cryptographic protocols. The advantages of the designs in terms of variation-resilience are also discussed. To have an overall idea of the hardware properties and other main characteristics, this section provides comprehensive comparisons of the proposed designs with other random number generators from the literature.

The trade-offs in terms of randomness quality and the number of transistors used are shown in Figure 4.10. Note that the MTJs are fabricated above all metal layers without occupying additional chip area in the integrated circuit, so the number of transistors is a good representation of the area.

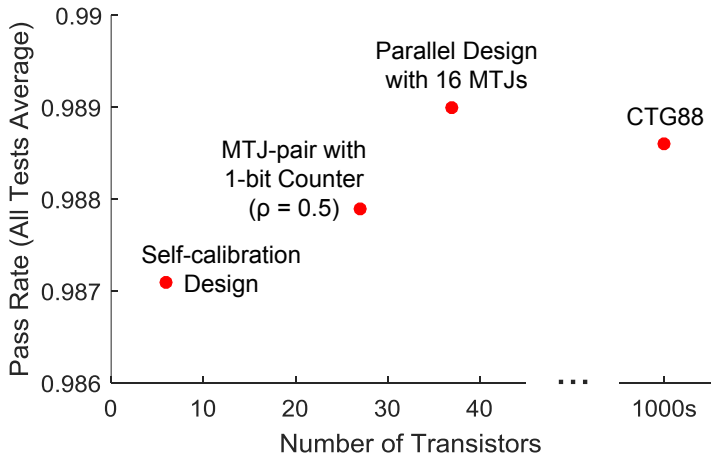

Figure 4.10 Comparison of the RNGs in terms of randomness quality and hardware cost

As shown in Figure 4.10, all of the three proposed designs are hardware-efficient, using fewer than 40 transistors. The self-calibration design uses the least number of transistors while the parallel design produces random numbers with the highest quality. Actually, it will save more hardware for the symmetric MTJ-pair design if the correlation coefficient is high enough to omit the quality improvement circuit. Moreover, when compared with other random number generators, all three proposed designs are very compact without compromising the randomness quality. For example, some comparable PRNGs, such as the CTGs, require much more hardware resources, since they contain hundreds of shift registers and other cells that add up to thousands of transistors. Other MTJ-based TRNGs with complicated post-processing or real-time tracking circuits also have much more hardware overhead.

The hardware simulation results for the proposed designs are summarized in Table 4.9 and are compared with those in [19], which includes a probability-locked loop. It is shown that, in addition to the high hardware-efficiency, all the proposed designs are energy-efficient (less than 1 pJ/bit) with a high generation speed (tens of MHz).

Generator	Parallel Design (with 16 MTJs)	MTJ-pair Design	Self-calibration Design	[19]
Technology	28 nm	28 nm	28 nm	90 nm
Frequency	177.8 MHz	66.7 MHz	50 MHz	66.7 MHz
Area Estimation	7.64 μm^2	3.84 μm^2	2.82 μm^2	Large
Energy	0.64 pJ/bit	0.81 pJ/bit	0.92 pJ/bit	Unknown
Statistical Tests	Passed	Passed	Passed	Not reported

Table 4.9 Performance comparisons of the RNGs

Other than the quantifiable comparison results provided above, some other main characteristics of the three proposed designs include:

- The parallel design:

  The parallel design has the highest throughput among the three designs. And both the throughput and the randomness quality can be adjusted according to the requirements by choosing the proper number of parallel MTJs. However, it is relatively large compared with the other two designs.

- The MTJ-pair design:

  The MTJ-pair design is especially suitable for MTJ devices with high correlations. Also, it can maintain a good behavior under various PVT corners so it is suitable for circuits with significant variations. Moreover, it has the simplest control among the three designs. The only disadvantage of it is that the probability of the random sequences that it can produce is fixed to 50% because of its special schematic, however, it does not matter for a common TRNG.

- The self-calibration design:

  The self-calibration design can also behave well under all PVT corners so it is also suitable for circuits with significant variations. It uses the least hardware among the three designs. However, it has a relatively low throughput compared with the other two designs.

In conclusion, each of the three proposed designs has its own specific characteristics which are summarized in Table 4.10; however, the common advantages for all of them are the high variation-resilience and high hardware-efficiency.

Generator	Advantages	Disadvantages
Parallel Design	Highest throughput Adjustable quality	Relatively large in area
MTJ-pair Design	Best for MTJs with high correlations Simplest control High resilience against circuit variations	Only for 50% frequency of 1's
Self-calibration Design	Highest hardware efficiency High resilience against circuit variations	Relatively slow

Table 4.10 Main characteristics of the proposed designs

Chapter 5: Conclusions and Future Work

5.1 Conclusions

This thesis work focuses on variation-resilient TRNGs based on MTJs for on-chip Internet cryptographic protocols. MTJ device fabrication variations and circuit variations, i.e., process, voltage and temperature, cause probability biases in the generated random sequences and undermine the randomness quality. To obtain high-quality random numbers, the basic generator with a single MTJ is not sufficient. Therefore, the objective is to design novel MTJ-based TRNGs with higher variation-resilience. In contrast to other work, designs in this work do not involve complicated circuits to ensure a high level of randomness, thus saving hardware and energy.

Three designs of TRNGs based on MTJs are proposed in this thesis work. All of them are both hardware-efficient and variation-resilient. The parallel design uses an array structure to minimize fabrication variation effects by averaging the biased probabilities of each single MTJ. The MTJ-pair design leverages the symmetry of two MTJs fabricated close to each other to obtain the 50% probability directly from the two identical distributions. The self-calibration design uses a two-step switching process to compensate for any probability inaccuracy occurring in the conventional one-step switching.

Apart from the main contributions of the three TRNG designs, an analysis of two categories of general flawed random sources, i.e., one with a fixed bias and the other with a certain variation, is conducted. Simple but universally applicable quality improvement

circuits are discussed and a general guideline for choosing the sequence generator in the QIC is provided as a possible solution for improving randomness.

All designs in this work are validated in a 28-nm CMOS process by Monte Carlo simulation with a compact model of the MTJ. It is verified by the statistical test suite that all three designs can produce high-quality random sequences for cryptography applications. The variation-resilience is verified by conducting the experiments with different PVT corners. Each of the designs is found to have specific advantages: the parallel design has an adjustable structure and the fastest speed; the MTJ-pair design is suitable for MTJs with high correlations and circuits with high variations; and the self-calibration design uses the least area and also works well with high variations. Comprehensive comparisons show that the designs save significant hardware compared with PRNGs and other MTJ-based RNGs in the literature. Hardware simulations confirm that all designs have high hardware-efficiency (using fewer than 40 transistors), high energy-efficiency (consuming lower than 1 pJ for generating 1 random bit), and high generation speeds (with frequencies of no lower than 50 MHz).

5.2 Future Work

Some possible directions for future work are as follows:

- The proposed TRNG designs could be examined more thoroughly by including more statistical tests in the suite. In that case, much longer random sequences, with millions of bits in each sequence, would be needed for those tests.
- The three TRNG designs could be combined somehow to obtain better generators; for example, the two-step switching method might be applied to the parallel

structure. The combined designs could be evaluated and compared with the three proposed designs.

- Some other simple post-processing circuits could be designed and implemented to improve the randomness quality of the sequences which fail certain tests. The reason that some tests fail needs to be analyzed in order to design circuits with the specific purpose in mind.
- The generators could be used and evaluated in the context of stochastic computation (SC). Some applications in SC might be found to have a better performance when using TRNGs instead of PRNGs.
- Some other properties or structures of spin-based devices might be leveraged to generate random numbers, such as the precessional switching [46] and complementary polarizer MTJs [47].

Bibliography

[1] A. Botta, W. de Donato, V. Persico and A. Pescapé, "Integration of Cloud Computing and Internet of Things: A survey," *Future Generation Computer Systems,* vol. 56, pp. 684-700, 2016.

[2] R. Weber, "Internet of Things – New Security and Privacy Challenges," *Computer Law & Security Review,* vol. 26, no. 1, pp. 23-30, 2010.

[3] H. Suo, J. Wan, C. Zou and J. Liu, "Security in the Internet of Things: A Review," *2012 International Conference on Computer Science and Electronics Engineering,* 2012.

[4] A. Menezes, P. van Oorschot and S. Vanstone, Handbook of Applied Cryptography, Boca Raton, FL: CRC Press, 1997.

[5] J. Han, H. Chen, J. Liang, P. Zhu, Z. Yang and F. Lombardi, "A Stochastic Computational Approach for Accurate and Efficient Reliability Evaluation," *IEEE Trans. Computers,* vol. 63, no. 6, pp. 1336-1350, 2014.

[6] A. Alaghi and J. Hayes, "Survey of Stochastic Computing," *ACM Trans. Embedded Computing Systems,* vol. 12, no. 2, pp. 1-19, 2013.

[7] P. Hellekalek, "Good Random Number Generators are (not so) Easy to Find," *Mathematics and Computers in Simulation,* vol. 46, no. 5-6, pp. 485-505, 1998.

[8] A. Uchida, K. Amano, M. Inoue, K. Hirano, S. Naito, H. Someya, I. Oowada, T. Kurashige, M. Shiki, S. Yoshimori, K. Yoshimura and P. Davis, "Fast Physical Random Bit Generation with Chaotic Semiconductor Lasers," *Nature Photonics,* vol. 2, no. 12, pp. 728-732, 2008.

[9] N. Oliver, M. Soriano, D. Sukow and I. Fischer, "Fast Random Bit Generation Using a Chaotic Laser: Approaching the Information Theoretic Limit," *IEEE J. Quantum Electronics,* vol. 49, no. 11, pp. 910-918, 2013.

[10] K. Yang, D. Blaauw and D. Sylvester, "An All-Digital Edge Racing True Random Number Generator Robust Against PVT Variations," *IEEE J. Solid-State Circuits,* vol. 51, no. 4, pp. 1022-1031, 2016.

[11] S. Mathew, S. Srinivasan, M. Anders, H. Kaul, S. Hsu, F. Sheikh, A. Agarwal, S. Satpathy and R. Krishnamurthy, "2.4 Gbps, 7 mW All-Digital PVT-Variation Tolerant True Random Number Generator for 45 nm CMOS High-Performance Microprocessors," *IEEE J. Solid-State Circuits,* vol. 47, no. 11, pp. 2807-2821, 2012.

[12] K. Yang, D. Fick, M. Henry, Y. Lee, D. Blaauw and D. Sylvester, "A 23Mb/s 23pJ/b Fully Synthesized True-random-number Generator in 28nm and 65nm CMOS," *2014 IEEE International Solid-State Circuits Conference (ISSCC),* 2014.

[13] M. Bucci, L. Germani, R. Luzzi, A. Trifiletti and M. Varanonuovo, "A High-speed Oscillator-based Truly Random Number Source for Cryptographic Applications on a Smartcard IC," *IEEE Trans. Computers,* vol. 52, no. 4, pp. 403-409, 2003.

[14] N. Liu, N. Pinckney, S. Hanson, D. Sylvester and D. Blaauw, "A True Random Number Generator Using Time-dependent Dielectric Breakdown," *IEEE Symp. VLSI Circuits,* pp. 203-204, 2010.

[15] P. Knag, W. Lu and Z. Zhang, "A Native Stochastic Computing Architecture Enabled by Memristors," *IEEE Trans. Nanotechnology,* vol. 13, no. 2, pp. 283-293, 2014.

[16] Y. Wang, W. Wen, H. Li and M. Hu, "A Novel True Random Number Generator Design Leveraging Emerging Memristor Technology," *The 25th Great Lakes Symp. VLSI (GLSVLSI),* 2015.

[17] S. Balatti, S. Ambrogio, R. Carboni, V. Milo, Z. Wang, A. Calderoni, N. Ramaswamy and D. Ielmini, "Physical Unbiased Generation of Random Numbers with Coupled Resistive Switching Devices," *IEEE Trans. Electron Devices,* vol. 63, no. 5, pp. 2029-2035, 2016.

[18] N. Onizawa, D. Katagiri, W. Gross and T. Hanyu, "Analog-to-Stochastic Converter Using Magnetic Tunnel Junction Devices for Vision Chips," *IEEE Trans. Nanotechnology,* vol. 15, no. 5, pp. 705-714, 2016.

[19] S. Oosawa, T. Konishi, N. Onizawa and T. Hanyu, "Design of an STT-MTJ based True Random Number Generator Using Digitally Controlled Probability-locked Loop," *2015 IEEE 13th International New Circuits and Systems Conference (NEWCAS),* 2015.

[20] Y. Wang, H. Cai, L. Naviner, J. Klein, J. Yang and W. Zhao, "A Novel Circuit Design of True Random Number Generator Using Magnetic Tunnel Junction," *2016 IEEE/ACM International Symp. Nanoscale Architectures (NANOARCH),* 2016.

[21] W. H. Choi, Y. Lv, J. Kim, A. Deshpande, G. Kang, J.-P. Wang and C. Kim, "A Magnetic Tunnel Junction based True Random Number Generator with Conditional Perturb and Real-time Output Probability Tracking," *2014 IEEE International Electron Devices Meeting,* 2014.

[22] A. Fukushima, T. Seki, K. Yakushiji, H. Kubota, H. Imamura, S. Yuasa and K. Ando, "Spin Dice: A Scalable Truly Random Number Generator based on Spintronics," *Appl. Phys. Express,* vol. 7, no. 8, p. 083001, 2014.

[23] E. Vatajelu, G. Di Natale and P. Prinetto, "STT-MTJ-based TRNG with on-the-fly temperature/current variation compensation," *2016 IEEE 22nd International Symposium on On-Line Testing and Robust System Design (IOLTS),* 2016.

[24] N. Rizzo, F. B. Mancoff, R. Whig, K. Smith, K. Nagel, T. Andre, P. G. Mather, S. Aggarwal, J. M. Slaughter, D. Mitchell and S. Tehrani, "Toggle and Spin Torque: MRAM at Everspin Technologies," *Proc. Non-Volatile Memories Workshop,* 2010.

[25] W. Gaviria Rojas, J. McMorrow, M. Geier, Q. Tang, C. Kim, T. Marks and M. Hersam, "Solution-Processed Carbon Nanotube True Random Number Generator," *Nano Letters,* vol. 17, no. 8, pp. 4976-4981, 2017.

[26] Y. Zhang, B. Yan, W. Kang, Y. Cheng, J. Klein, Y. Zhang, Y. Chen and W. Zhao, "Compact Model of Subvolume MTJ and Its Design Application at Nanoscale Technology Nodes," *IEEE Trans. Electron Devices,* vol. 62, no. 6, pp. 2048-2055, 2015.

[27] H. Cai, Y. Wang, L. Naviner and W. Zhao, "Robust Ultra-Low Power Non-Volatile Logic-in-Memory Circuits in FD-SOI Technology," *IEEE Trans. Circuits and Systems I: Regular Papers,* vol. 64, no. 4, pp. 847-857, 2017.

[28] National Institute of Standards and Technology, "Special Publication 800-22 rev.1a," 2010. [Online]. Available: http://csrc.nist.gov/groups/ST/toolkit/rng/documents/SP800-22rev1a.pdf.

[29] Y. Qu, J. Han, B. Cockburn, Y. Zhang, W. Zhao and W. Pedrycz, "A True Random Number Generator based on Parallel STT-MTJs," *2017 Design, Automation and Test in Europe (DATE),* 2017.

[30] Y. Wang, H. Cai, L. Naviner, Y. Zhang, X. Zhao, E. Deng, J. Klein and W. Zhao, "Compact Model of Dielectric Breakdown in Spin-Transfer Torque Magnetic Tunnel Junction," *IEEE Trans. Electron Devices,* vol. 63, no. 4, pp. 1762-1767, 2016.

[31] W. Kang, Z. Li, J. Klein, Y. Chen, Y. Zhang, D. Ravelosona, C. Chappert and W. Zhao, "Variation-Tolerant and Disturbance-Free Sensing Circuit for Deep Nanometer STT-MRAM," *IEEE Trans. Nanotechnology,* vol. 13, no. 6, pp. 1088-1092, 2014.

[32] D. Zhang, L. Zeng, Y. Qu, Y. Zhang, M. Wang, W. Zhao, T. Tang and Y. Wang, "Energy-efficient Neuromorphic Computation based on Compound Spin Synapse with Stochastic Learning," *2015 IEEE International Symp. Circuits and Systems (ISCAS),* 2015.

[33] R. Sbiaa, H. Meng and S. Piramanayagam, "Materials with Perpendicular Magnetic Anisotropy for Magnetic Random Access Memory," *physica status solidi (RRL) - Rapid Research Letters,* vol. 5, no. 12, pp. 413-419, 2011.

[34] S. Ikeda, K. Miura, H. Yamamoto, K. Mizunuma, H. Gan, M. Endo, S. Kanai, J. Hayakawa, F. Matsukura and H. Ohno, "A Perpendicular-anisotropy CoFeB–MgO Magnetic Tunnel Junction," *Nature Materials,* vol. 9, no. 9, pp. 721-724, 2010.

[35] P. Magarshack, "Breakthrough Technologies and Teference Designs for New IoT Applications," *2015 Symp. VLSI Circuits,* 2015.

[36] M. Martin, B. Dlubak, R. Weatherup, H. Yang, C. Deranlot, K. Bouzehouane, F. Petroff, A. Anane, S. Hofmann, J. Robertson, A. Fert and P. Seneor, "Sub-nanometer Atomic Layer Deposition for Spintronics in Magnetic Tunnel Junctions Based on Graphene Spin-Filtering Membranes," *ACS Nano,* vol. 8, no. 8, pp. 7890-7895, 2014.

[37] Y. Zhang, W. Zhao, Y. Lakys, J. Klein, J. Kim, D. Ravelosona and C. Chappert, "Compact Modeling of Perpendicular-Anisotropy CoFeB/MgO Magnetic Tunnel Junctions," *IEEE Trans. Electron Devices,* vol. 59, no. 3, pp. 819-826, 2012.

[38] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol version 1.2.," 2008. [Online]. Available: http://tools.ietf.org/html/rfc5246.

[39] A. Yamaguchi, T. Seo and K. Yoshikawa, "On the Pass Rate of NIST Statistical Test Suite for Randomness," *JSIAM Letters,* vol. 2, pp. 123-126, 2010.

[40] K. Koch, Parameter Estimation and Hypothesis Testing in Linear Models, Berlin: Springer, 2011.

[41] R. Johnson and D. Wichern, Applied Multivariate Statistical Analysis. Vol. 4., New Jersey: Prentice-Hall, 2014.

[42] D. Eastlake, S. Crocker and J. Schiller, "Randomness Recommendations for Security," 1994. [Online]. Available: http://tools.ietf.org/html/rfc1750.

[43] R. B. Davies, "Exclusive OR (XOR) and Hardware Random Number Generators," Tech. Rep., 2002. [Online]. Available: http://www.robertnz.net/pdf/xor2.pdf.

[44] A. Alimohammad, S. Fard, B. Cockburn and C. Schlegel, "On the Efficiency and Accuracy of Hybrid Pseudo-random Number Generators for FPGA-based Simulations," *2008 IEEE International Symp. Parallel and Distributed Processing,* 2008.

[45] P. L'Ecuyer, "Maximally Equidistributed Combined Tausworthe Generators," *Mathematics of Computation,* vol. 65, no. 213, pp. 203-214, 1996.

[46] N. Rangarajan, A. Parthasarathy and S. Rakheja, "A Spin-based True Random Number Generator Exploiting the Stochastic Precessional Switching of Nanomagnets," *J. Applied Physics,* vol. 121, no. 22, p. 223905, 2017.

[47] X. Fong, M. Chen and K. Roy, "Generating true random numbers using on-chip complementary polarizer spin-transfer torque magnetic tunnel junctions," *72nd Device Research Conference,* 2014.

Appendix A: VerilogA Code for the MTJ Model

```
/* Copyright @ 2015 Institut d'Electronique Fondamentale, CNRS UMR 8622, University of Paris-Sud 11, 91405 Orsay, France
The terms under which the software and associated documentation (the Software) is provided are as the following:
The Software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability,
fitness for a particular purpose and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other
liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the Software or the use or other dealings
in the Software.
The authors or copyright holders grants, free of charge, to any users the right to modify, copy, and redistribute the Software, both within the
user's organization and externally, subject to the following restrictions:
1. The users agree not to charge for the code itself but may charge for additions, extensions, or support.
2. In any product based on the Software, the users agree to acknowledge the Research Group that developed the software. This acknowledgment shall
appear in the product documentation.
3. The users agree to obey all U.S. Government restrictions governing redistribution or export of the software.
4. The users agree to reproduce any copyright notice which appears on the software on any copy or modification of such made available to others.
Agreed to by You WANG, Yue Zhang, Weisheng Zhao, Jaques-Olivier Klein, Thibaut Devolder, Dafine Ravelosona and Claude Chappert 22 February 2016*/

//Title: Compact model of Perpendicular Magnetic Anistropy (PMA) MTJ integrating dielectric breakdown effect
//Version: Beta.5 breakdown
//Date:22 February 2016
//Language: VerilogA

/*----------------------------------------
Property: IEF, UMR8622, Univ.Paris Sud-CNRS
Authors: You WANG, Yue ZHANG, Weisheng ZHAO, Yahya Lakys, Dafine Ravelosona, Jacques-Olivier Klein and Claude Chappert

In this model, it takes into account the static, dynamic and stochastic behavoirs of PMA MTJ nanopillar

1.MTJ resistance calculation based on brinkman model
2.TMR dependence on the bias voltage
3.Spin polarity calculation model for magnetic tunnel junction
4.Critical current calculation
5.Dynamic model (>critical current, also sun's model)
6.Stochastic model
7.Resistance variation
8.Temperature evaluation
9.Breakdown voltage
10.Lifetime (Time to failure)
11.Breakdown probability
12.Temperature dependent parameters
```

```
The parameters are from the prototypes of Univ. Tohuku

-----------------------------------------------*/
`resetall
`include "constants.vams"
`include "disciplines.vams"
`define explimit 85.0
`define exp(x) exp(min(max((x),-`explimit),`explimit))
`define sqrt(x) pow( (x), 0.5)

//Shape definition
`define rec 1
`define ellip 2
`define circle 3

/*-----------------------------------------
Electrical Constants
-----------------------------------------------*/

/*----------Elementary Charge---------------*/
`define e 1.6e-19
/*----------Bohr Magnetron Costant----------*/
`define ub 9.27e-28
/*----------Boltzmann Constant------------- */
`define Kb 1.38e-23
/*----------Electron Mass------------- */
`define m 9.10e-31
/*----------Euler's constant---------------*/
`define C 0.577


module Model(T1,T2,Ttrans,Temp,Break);

inout T1, T2;
electrical T1, T2;
electrical n1,n2;    //virtual terminals of RC circuit for time modelisation for temperature


/*----------Ttrans=store the state of the MTJ with time influence, non-volatile way------------ */
/*----------Temp=store the temperature------------- */
inout Ttrans,Temp;
electrical Ttrans,Temp;
inout Break;
electrical Break;
```

```
/*-----------------------------------------
MTJ Technology Parameters
(Corresponds to the HITACHI MTJ Process)
-----------------------------------------*/

/*----------Gilbert Damping Coefficient---------------*/
parameter real  alpha=0.027;
/*----------GyroMagnetic Constant in Hz/Oe---------------*/
parameter real  gamma=1.76e7;
/*----------Electron Polarization Percentage % ---------------*/
parameter real  P=0.52;
/*----------Out of plane Magnetic Anisotropy in Oersteds---------------*/
parameter real  Hk0=1433;
/*----------Saturation Field in the Free Layer in Oersteds---------------*/
parameter real  Ms0=15800;
/*----------The Energy Barrier Height for MgO in electron-volt---------------*/
parameter real  PhiBas=0.4;
/*----------Voltage bias when the TMR(real) is 1/2TMR(0) in Volt---------------*/
parameter real  Vh=0.5;          //experimental value with MgO barrier


/*-----------------------------------------
Device Parameters
(Corresponds to the HITACHI 240 x 80 MTJ)
-----------------------------------------*/
/*----------Height of the Free Layer  in nm---------------*/
parameter real  tsl=1.3e-9 from[0.7e-9:3.0e-9];
/*----------Length in nm---------------*/
parameter real  a=40e-9;
/*----------Width in nm---------------*/
parameter real  b=40e-9;
/*----------Radius in nm---------------*/
parameter real  r=20e-9;
/*----------Height of the Oxide Barrier in nm---------------*/
parameter real  tox=8.5e-10 from[8e-10:15e-10];
/*----------TMR(0) with Zero Volt Bias Voltage ---------------*/
parameter real  TMR=0.7;

/*----------Shape of MTJ---------------*/
parameter real   SHAPE=2  from[1:3];  //SQUARE

/*----------Neel-Brown model parameter ---------------*/
parameter real   tau0=8.7e-10;   //experiental value, prototype Hitachi 2007m with CoFe layer
```

```
/*----------Error probability Ps=1-Pr(t) ----------------*/
parameter real   Ps=0.999999;

/*----------Threshold for Neel-Brown model----------------*/
parameter real brown_threshold=0.0;

/*----------MTJ State Parameters----------------*/
/*----------Initial state of the MTJ, 0 = parallele, 1 = anti-parallele----*/
parameter integer       PAP=1 from[0:1];
/*----------Room temperature in Kelvin----------------*/
parameter real          T= 300;//$random % 50 +323;

/*----------Resistance area product in ohmum2----------------*/
parameter real          RA=5 from[5:15];

/*----------Parameters of RC circuit for time modelisation for temperature---------------*/
/*----------Heat capacity per unit volume in J/m3*K----------------*/
parameter real          Cv= 2.74e6  from[2.735e6:2.7805e6];
/*----------Thermal conductivity of the thermal barrier(MgO) in W/m*K----------------*/
parameter real          lam= 84.897 from [84.8912:84.9449];//
/*----------Total thickness of MTJ nanopillar in nm----------------*/
parameter real          thick_s= 3.355e-8;//
/*----------RC circuit for time modelisation for temperature---------------*/
parameter real resistor=100e6; //R=100M
parameter real coeff_tau=12; //Coefficient to increase tau_th

real capacitor;   //virtual capacitor
real tau_th;        //characteristic heating/cooling time
real temp;    //real temperature of MTJ
real temp_init;   //temperature initialised
real R;         //resistance of MTJ

/*----------Parameters for real TMR ratio---------------*/
parameter real S=1.5;
parameter real Em0=1.936e-20;    //121 meV
parameter real epsilon=0.305;           //1/3.279;
parameter real Q=0.025;
parameter real Ec=4.32e-23;                    //0.27e-3*1.6e-19;
real Ms, Hk, Beta;

/*---------Parameters for stochastic behaviors---------------*/
parameter integer STO=0 from[0:2];           //choice of stochastic dynamic, 0 no stochastic, 1 random exponential distribution,2 random gauss
distribution
```

```
parameter integer RV=0 from[0:2];         //choice of stochastic static intrinsically, 0 no stochastic, 1 random uniform distribution,2 random gauss
distribution

parameter integer Temp_var=0 from[0:1];     //choice of time modelisation for temperature

parameter real DEV_tox=0.03;        //choice of standard deviation of stochastic static gauss distribution for tox when RV=2
parameter real DEV_tsl=0.03;        //choice of standard deviation of stochastic static gauss distribution for tsl when RV=2
parameter real DEV_TMR=0.03;        //choice of standard deviation of stochastic static gauss distribution for TMRwhen RV=2

parameter real STO_dev=0.03;        //choice of standard deviation of stochastic dynamic gauss distribution when STO=2

//variables

//Polaristion constant for the two states of STT-MTJ
real PolaP;      //Polarization state parallel of STT-MTJ
real PolaAP;     //Polarization state anti-parallel of STT-MTJ

real surface;    //Surface of MTJ

//Critical current density for the two states of STT-MTJ
real gp;         //Critical current density for P state
real gap;        //Critical current density for AP state

real Em,EE;      //Variable of the Slonczewski model

//TMR real value for the two states of STT-MTJ
real TMRR;       //TMR real value for P state
real TMRRT;      //TMR real value for AP state

//Resistance of MTJ
real Ro;//Resistance of MTJ when bias voltage = 0V
real Rap;        //Resistance value for AP state
real Rp;//Resistance value for P state

//Voltage of MTJ
real Vb;//V(T1,T2)
real Vc;//V(T2,T1)

real Id;//Current of MTJ

//critial current for the two states of STT-MTJ
real IcAP;       //Critial current for AP state
real IcP;        //Critial current for P state

real ix;//Current used to store the state of the MTJ
```

```
real tau;       //Probability parameter

real FA; //Factor for calculating the resistance based on RA

integer seed;   //Used to initialize the random number generator

//Stochastic effects
real durationstatic,duration;    //time needed to be sure that the switching is effected

real toxreal;   //real thickness of oxide layer
real tslreal;   //real thickness of free layer
real TMRreal;   //real TMR
(*cds_inherited_parameter*)parameter real seedin = 0;  //generation of a real random value of seed for random distribution function modified 20140223
(*cds_inherited_parameter*)parameter real seed1 = 0;  //generation of a real random value of seed for breakdown
//probability modified 20150416
integer seed2;
/*----------switching delay----------------*/
real P_APt;
real AP_Pt;
real NP_APt,NAP_Pt;

/*----------breakdown voltage----------------*/
real Vbp_p,Vbp_n,Vbap_p, Vbap_n;

/*----------parameters for calculating the lifetime----------------*/

parameter real acc=1.53e-8;   //acceleration parameter   1.53e-8
parameter real H=0.8e-19;   //activation energy parameter 0.8e-19
parameter real beta=1.5;   //shape parameter 1.5

/*----------parameters for calculating the breakdown probability----------------*/
real possibilite;
real F;
real xF;
real TF;

real break;   //breakdown has already occured or not

analog begin

        if (SHAPE==1)
                begin
                surface=a*b;    //SQUARE
                end
```

```
                else if (SHAPE==2)
                begin
                surface=`M_PI*a*b/4;     //ELLIPSE
                end
                else
                begin
                surface=`M_PI*r*r;    //ROUND
        end

        Vc=V(T2,T1);     //potential between T2 and T1
        Vb=V(T1,T2);     //potential between T2 and T1
//initial conditions
@(initial_step)
begin
        //H=1.082e-11*toxreal+6.996e-20;
        break=0;       //Breakdown doesn't occur at the beginning of simulation
        seed=1000000000*seedin;  //initialization of seed modified 20140516

        seed2=100000000*seed1;

        FA=3322.53/RA;  //initialization of resistance factor according to RA product

        if (RV==1)
                begin
                        //real thinkness of oxide layer, free layer and real TMR considering the random distribution(uniform distribution)
                        toxreal=$rdist_uniform(seed,(tox-tox*DEV_tox),(tox+tox*DEV_tox));
                        tslreal=$rdist_uniform(seed,(tsl-tsl*DEV_tsl),(tsl+tsl*DEV_tsl));
                        TMRreal=$rdist_uniform(seed,(TMR-TMR*DEV_TMR),(TMR+TMR*DEV_TMR));
                end
        else if (RV==2)
                begin
                        //real thinkness of oxide layer, free layer and real TMR considering the random distribution(gauss distribution)
                        toxreal=abs($rdist_normal(seed,tox,tox*DEV_tox/3));
                        tslreal=abs($rdist_normal(seed,tsl,tsl*DEV_tsl/3));
                        TMRreal=abs($rdist_normal(seed,TMR,TMR*DEV_TMR/3));

                end
        else
                begin
                        toxreal=tox;
                        tslreal=tsl;
                        TMRreal=TMR;
                end
        temp=T;                     //parameters for temperature
        temp_init=T;
```

```
        tau_th= Cv*thick_s / (lam/thick_s);

        capacitor=coeff_tau*tau_th/resistor;     //tau_th=resistor*capacitor

        Ro=(toxreal*1.0e10/(FA*`sqrt(PhiBas)*surface*1.0e12))*exp(1.025*toxreal*1.0e10*`sqrt(PhiBas));    //resistance

        Vbp_p=toxreal*7.6e8+0.202;    //breakdown voltage of parallel, positive bias
        Vbp_n=toxreal*8.3e8+0.206;    //parallel, negative bias
        Vbap_p=toxreal*8.3e8+0.436;  //antiparallel, positive bias
        Vbap_n=toxreal*8e8+0.32;     // antiparallel, negative bias

        Em=Ms*tslreal*surface*Hk/2;       //parameters for calculating switching delay
        duration=0.0;
        P_APt=1000000000;
        AP_Pt=1000000000;
        NP_APt=1000000000;
        NAP_Pt=1000000000;
        if(analysis("dc"))               //States inititialisation
            begin
              ix=PAP;
            end
        else
            begin
              ix=-PAP;
            end
end

if(Temp_var==0)
   begin
      temp=temp_init;        //temperature is constant
   end
else
   begin
      temp=V(Temp);          //temperature actualisation
   end

Ms=18342*(1-(temp/1120)*sqrt(temp/1120));
Hk=-3*temp+2333;
//Hk=1433;
//Ms=15800;
Em=Ms*tslreal*surface*Hk/2;
EE=Em/(`Kb*temp*40*`M_PI);  //result of E/kbT
Beta=S*`Kb*temp/(Em0*epsilon);
```

```
/*----calculation of real current------*/

TMRR=1/(1+Vb*Vb/(Vh*Vh))*((TMRreal+1)/(1+2*Q*Beta*log(`Kb*temp/Ec))-1);  //real TMR ratio

Rp=Ro;
Rap=Rp*(1+TMRR);

 if(break==1)
     begin
        R=10;
     end
 else if(break==0&&ix==0)
     begin
        R=Rp;
     end
 else
     begin
        R=Rap;
     end
 Id=Vb/R;

/*----calculation of rcritical current------*/

PolaP=`sqrt(TMRR*(TMRR+2))/(2*(TMRR+1));        //Polarization state parallel
gp=alpha*gamma*`e*Ms*tslreal*Hk/(40*`M_PI*(`ub*PolaP)); //Critical current density
IcP=gp*surface;         // Critical current for P state

PolaAP=`sqrt(TMRR*(TMRR+2))/(2*(TMRR+1));       //Polarization state anti parallel
gap=alpha*gamma*`e*Ms*tslreal*Hk/(40*`M_PI*(`ub*PolaAP));       //Critical current density
IcAP=gap*surface;       // Critical current for AP state

     /*------Counter of time when real current is higher than critical current */
          @(above(Id-IcP,+1))
          begin
                P_APt = $abstime;
                NP_APt=1000000000;
          end

          @(above(-Id-IcAP,+1))
          begin
                AP_Pt = $abstime;
                NAP_Pt=1000000000;
          end
          @(above(Vb-brown_threshold,+1))
          begin
```

```
                NP_APt = $abstime;
                AP_Pt=1000000000;
                NAP_Pt=1000000000;
        end

        @(above(Vc-brown_threshold,+1))
        begin
                NAP_Pt = $abstime;
                P_APt=1000000000;
                NP_APt=1000000000;
        end

if(analysis("dc"))   //dc analysis

        begin
                if(ix==0)       //Case which the magnetizations of the two layers are parallel
                        begin
                             if(Vb>=Vbp_p||Vb<=-Vbp_n)
                               begin
                                   R=10;
                               end
                             else
                               begin
                                 if(Vb>=(IcP*Rp))
                                        begin
                                                ix=1.0;

                                        end
                               end
                        end
                else
                        begin
                             if(Vb>=Vbap_p||Vb<=-Vbap_n)
                                begin
                                    R=10;
                                end
                             else

                               begin
                                  if(Vc>=(IcAP*Rap))
                                        begin
                                                ix=0.0;
                                        end

                                end
```

```
                        end


                V(Ttrans)<+ix;
                Id=Vb/R;
                I(T1,T2)<+Id;    //Actualisation of the current of MTJ with the value calculated

                /*$display("Vbp_p=",Vbp_p);        //visualise the breakdown voltages
                $display("Vbp_n=",Vbp_n);
                $display("Vbap_p=",Vbap_p);
                $display("Vbap_n=",Vbap_n);
                $display("Id=",Id);*/
        end
else                        //transient analysis
        begin

       if(break==0)    // breakdown hasn't occured
        begin
             if(Vb>=Vbp_p||Vb<=-Vbp_n||Vb>=Vbap_p||Vb<=-Vbap_n)
                 begin
                      break=1;
                 end


              possibilite=$rdist_uniform(seed2,0,1);   //a probability between 0 and 1

              TF= exp(H/(`Kb*T)-acc*abs(Vb)/toxreal);    //lifetime of breakdown
              if($abstime<=1e-8)
                  begin
                       xF=beta*(log(1e-20)-ln(TF));  //If abstract time is too small, the value is defined to avoid bug
                  end
              else
                  begin
                       xF=beta*(log($abstime-1e-8)-ln(TF)+log(exp(1)));    //weibull distribution
                  end
              F=1-exp(-exp(xF));     //probability of breakdown

              if(F>=possibilite)
                  begin
                       break=1;    // If the random probability is inferior to the breakdown probability, breakdown occurs
                  end
              else
                  begin
                       break=0;
                  end
```

```
            if(STO==1||STO==2)    //considering the stochastic behaviors
                begin

                if(ix==0)     //Case which the magnetizations of the two layers are parallel
                     begin
                          if(Vb>=IcP*Rp)
                              begin   //Current higher than critical current,STT-MTJ dynamic behavior : Sun model
                                      //Time needed to be sure that the switching is effected

durationstatic=(`C+ln(`M_PI*`M_PI*(Em/(`Kb*temp*40*`M_PI))/4))*`e*1000*Ms*surface*tslreal*(1+P*P)/(4*`M_PI*2*`ub*P*10000*abs(Id-IcP));


                                      if(STO==1)
                                         begin
                                              duration=abs($rdist_exponential(seed, durationstatic));     //stochastic           effect(exponential
distribution)
                                         end
                                      else if(STO==2)
                                         begin

                                              duration=abs($rdist_normal(seed,durationstatic,durationstatic*STO_dev/3));   //stochastic
effect(gauss distribution)
                                         end
                                      else
                                         begin
                                              duration=durationstatic;
                                         end

                                      if(duration<=($abstime-P_APt))
                                         begin   //Switching of the free layer always occurs
                                                 ix=-1.0;   //change the current state of MTJ

                                         end
                                      else
                                         begin
                                                 ix=0.0;
                                         end
                                  end

                              else
                                  begin  //Current smaller than critical current

                                       ix=0.0;   //save the current state of MTJ,STT-MTJ dynamic behavior : Neel-Brown model
                                       tau=tau0*exp(Em*(1-abs(Id/IcP))/(`Kb*temp*40*`M_PI));
```

```
                                    if(Vb>brown_threshold)
                                       begin
                                            if (Vb<0.8*IcP*Rp)
                                                 begin

                                                      if(STO==1)
                                                             begin
                                                                  duration=abs($rdist_exponential(seed, tau)); //stochastic effect
                                                             end
                                                      else if(STO==2)
                                                             begin

                                                                  duration=abs($rdist_normal(seed,tau,tau*STO_dev/3));  //stochastic
effect(gauss distribution)
                                                             end
                                                      else
                                                             begin
                                                                  duration=tau;
                                                             end
                                                    if (($abstime-NP_APt) >= duration)
                                                             begin
                                                                   ix=-1.0;   //change the current state of MTJ

                                                             end
                                                      else
                                                             begin
                                                                   ix=0.0;
                                                             end

                                                 end
                                        end

                                end

                    end  //end of parallel state


            else        //Case which the magnetizations of the two layers are antiparallel
               begin
                 if(Vc>=(IcAP*Rap))
                    begin       //Current higher than critical current,STT-MTJ dynamic behavior : Sun model


       durationstatic=(`C+ln(`M_PI*`M_PI*(Em/(`Kb*temp*40*`M_PI))/4))*`e*1000*Ms*surface*tslreal*(1+P*P)/(4*`M_PI*2*`ub*P*10000*abs(-Id-IcAP));
```

```
                        //time needed to be sure that the switching is effected
                        if(STO==1)
                                begin
                                        duration=abs($rdist_exponential(seed, durationstatic));  //stochastic effect
                                end
                        else if(STO==2)
                                 begin
                                        duration=abs($rdist_normal(seed,durationstatic,durationstatic*STO_dev/3.0));      //stochastic
effect(gauss distribution)
                                 end
                        else
                                 begin
                                        duration=durationstatic;
                                 end
                        if(duration<=($abstime-AP_Pt))
                                 begin  //Switching of the free layer always occurs
                                        ix=0.0;   //change the current state of MTJ

                                end
                        else
                                 begin
                                        ix=-1.0;
                                 end
                  end
                else
                  begin   //Current smaller than critical current,STT-MTJ dynamic behavior : Neel-Brown model
                        tau=tau0*exp(Em*(1-abs(Id/IcAP))/(`Kb*temp*40*`M_PI));

                        if(Vc>brown_threshold)
                             begin
                                  if (Vc<0.8*IcAP*Rap)
                                        begin
                                             if(STO==1)
                                                   begin
                                                        duration=abs($rdist_exponential(seed, tau));     //stochastic effect
                                                   end
                                             else if(STO==2)
                                                   begin
                                                        duration=abs($rdist_normal(seed,tau,tau*STO_dev/3));     //stochastic         effect(gauss
distribution)
                                                   end
                                             else
                                                   begin
                                                        duration=tau;
                                                   end
```

```
                                        if (duration<=($abstime-NAP_Pt))
                                              begin
                                                   ix=0.0;   //change the current state of MTJ

                                             end
                                        else
                                              begin
                                                   ix=-1.0;
                                              end
                                   end

                              end
                    end



              end   // end of antiparallel state

           end    //end of module with consideration of stochastic behaviors

      else                          //without consideration of stochastic behaviors
             begin

                if(ix==0) //Case which the magnetizations of the two layers are parallel
                     begin
                         if(Vb>=IcP*Rp)  //Current higher than critical current,  STT-MTJ dynamic behavior : Sun model
                                 begin
                                        //Time needed to be sure that the switching is effected

durationstatic=(`C+ln(`M_PI*`M_PI*(Em/(`Kb*temp*40*`M_PI))/4))*`e*1000*Ms*surface*tslreal*(1+P*P)/(4*`M_PI*2*`ub*P*10000*abs(Id-IcP));
                                    duration=durationstatic;
                                    if(duration<=($abstime-P_APt))
                                         begin//Switching of the free layer always occurs
                                              ix=-1.0;   //change the current state of MTJ

                                         end
                                    else
                                         begin
                                              ix=0.0;
                                         end

                                end
                          else
                                begin  //Current smaller than critical current
                                      tau=tau0*exp(Em*(1-abs(Id/IcP))/(`Kb*temp*40*`M_PI));
```

```
                                    if(Vb>brown_threshold)
                                        begin
                                             if (Vb<0.8*IcP*Rp)
                                                    begin
                                                           duration=tau;
                                                           if (($abstime-NP_APt) >= duration)
                                                                begin
                                                                    ix=-1.0;   //change the current state of MTJ

                                                                end
                                                           else
                                                                begin
                                                                    ix=0.0;
                                                           end
                                                    end
                                             end
                                  end
                            end

                  else    //Case which the magnetizations of the two layers are antiparallel

                        begin
                              if(Vc>=(IcAP*Rap))
                                     begin//Current higher than critical current,STT-MTJ dynamic behavior : Sun model

durationstatic=(`C+ln(`M_PI*`M_PI*(Em/(`Kb*temp*40*`M_PI))/4))*`e*1000*Ms*surface*tslreal*(1+P*P)/(4*`M_PI*2*`ub*P*10000*abs(-Id-IcAP));

                                           duration=durationstatic;  //time needed to be sure that the switching is effected
                                           if(duration<=($abstime-AP_Pt))
                                                 begin   //Switching of the free layer always occurs
                                                        ix=0.0;   //change the current state of MTJ
                                                 end
                                           else
                                                 begin
                                                        ix=-1.0;
                                                 end
                                     end
                               else
                                     begin       //Current smaller than critical current,STT-MTJ dynamic behavior : Neel-Brown model
                                      tau=tau0*exp(Em*(1-abs(Id/IcAP))/(`Kb*temp*40*`M_PI));
                                      if(Vc>brown_threshold)
                                           begin
                                               if (Vc<0.8*IcAP*Rap)
                                                    begin
```

```
                                                                duration=tau;
                                                                if (duration<=($abstime-NAP_Pt))
                                                                      begin
                                                                              ix=0.0;   //change the current state of MTJ

                                                                      end
                                                                              else
                                                                      begin
                                                                              ix=-1.0;
                                                                      end
                                                        end
                                        end
                                end

                    end

                end



        end
        I(Ttrans)<+ transition(ix,0,1e-12,1e-12); //Ttrans has the same function than x but it includes the time effects

        I(T1,T2)<+Id;     //Actualisation of the current of MTJ with the value calculated

       end    //end of transient analysis
            if(Temp_var==1)
                 begin
                      V(n1) <+ ( V(T1,T2)*V(T1,T2) )/ ( R*surface*lam/(thick_s ));   //Definition of the maximum incrase of temperature

                      I(n1,n2) <+ V(n1,n2) / resistor;     // RC circuit definition,RC circuit parallel
                      I(n2) <+ capacitor * (ddt(V(n2)));

                      V(Temp) <+  V(n2) + temp_init;
                 end
            else
                begin
                      V(Temp) <+  temp_init;

                end
           V(Break) <+ break;
    end  //end of analog begin

endmodule
```

Appendix B: Mathematical Proof of the Theory in Section 3.2

Theory:

If two independent variables follow identical distributions, the probability that the first variable is smaller than the second one is 50%.

Proof:

Suppose X_1 and X_2 are two independent random variables drawn from the same distribution X $(X_1 = X_2 = X)$. The probability density function (PDF) of the random variables is $f(x)$, while the cumulative distribution function (CDF) is $F(x)$. The minimum and maximum possible values of X are $\min(X)$ and $\max(X)$, respectively. By definition,

$$F(\min(X)) = 0, F(\max(X)) = 1 \tag{B.1}$$

The probability that X_1 is less than X_2 is

$$\begin{aligned} P(X_1 < X_2) &= \int_{a=\min(X)}^{\max(X)} \left[f(a) \cdot \int_{b=a}^{\max(X)} f(b)\, db \right] da \\ &= \int_{a=\min(X)}^{\max(X)} \left[f(a) \cdot F(b) |_{b=a}^{\max(X)} \right] da \\ &= \int_{a=\min(X)}^{\max(X)} \left[f(a) \cdot \left(1 - F(a)\right) \right] da \\ &= \left[F(a) - \frac{1}{2} F^2(a) \right] |_{a=\min(X)}^{\max(X)} \\ &= \frac{1}{2} \end{aligned} \tag{B.2}$$

Therefore, it is proved that the result is fixed to 0.5 and is irrelevant to the actual distribution of the random variables X_1 and X_2, as long as they follow identical distributions.

Appendix C: Calculations for the Theory in Section 3.3

Based on the diagram shown in Figure 3.7, after the two switching steps, the probabilities that the MTJ is in the P state and the AP state are

$$\begin{aligned} P &= 1 - c \cdot p_1 + c^2 \cdot p_1 \cdot p_2 \\ AP &= c \cdot p_1 - c^2 \cdot p_1 \cdot p_2. \end{aligned} \tag{C.1}$$

For the MTJs used in this paper, $0.8116 < c < 1.1884$ when 3σ is considered, which aligns with the probability variation from 40.58% to 59.42%. Moreover, the different process corners will further bias the actual switching probability beyond $c \cdot p$.

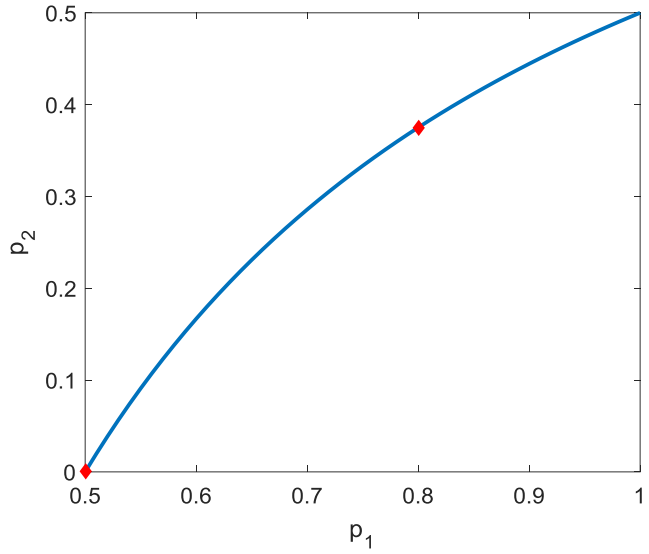

Figure C.1 Relationship between the two switching probabilities

Aiming at a 50% probability, we let $P = AP = 0.5$ and $c = 1$ in (C.1), and then we get

$$\begin{aligned} p_2 &= \frac{p_1 - 0.5}{p_1} \ (0.5 < p_1 < 1) \\ AP &= (-p_1 + 0.5)c^2 + p_1 c. \end{aligned} \tag{C.2}$$

The relationship between the two switching probabilities p_1 and p_2 are shown in Figure C.1. In the basic generator, $p_1 = 0.5$, and $AP = 0.5c$, which is linearly related to the bias coefficient c (lower point in Figure C.1). We aim to limit the variation by making the switching probability less related to c, so we minimize the derivative of AP with respect to c at $c = 1$. From (C.2), we have

$$\frac{dAP}{dc}|_{c=1} = 1 - p_1. \tag{C.3}$$

A higher value of p_1 helps to minimize it. However, considering the fact that the probability cannot exceed 1, so $c \cdot p_1 < 1$ and $p_1 < 0.84$. We choose $p_1 = 0.8$ to make sure that the calculated switching probability is no more than 1 in all process corners (upper point in Figure C.1). Finally, we have $p_2 = 0.375$ and $AP = -0.3c^2 + 0.8c$.

Appendix D: Key Waveforms and Transistor Parameters of the TRNG Designs

Figure D.1 shows the waveforms at V_{sense} and V_{out} in the read phases of the parallel design described in Section 3.1. Note that the actual values of V_{sense} are reflecting the resistive state of the MTJ being sensed, so they are random but within certain corresponding ranges. The waveform at V_{out} for the read phase of the self-calibration design described in Section 3.3 is similar to V_{sense} during any of the 5-ns sections in Figure D.1.

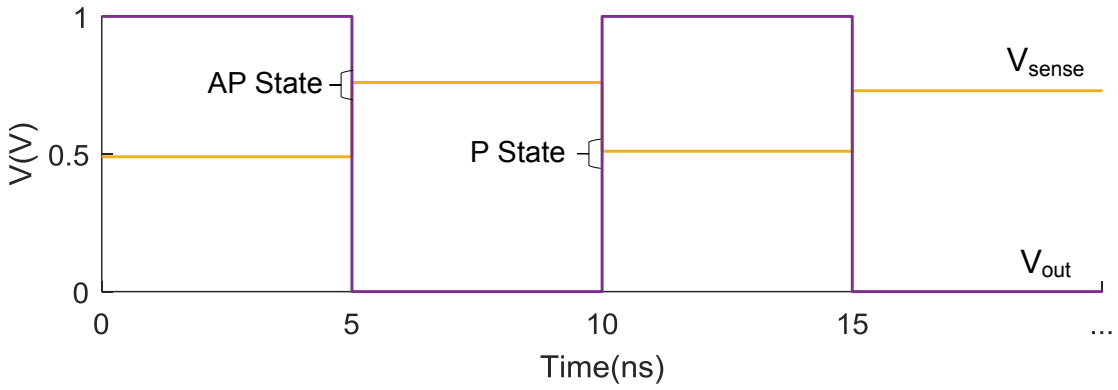

Figure D.1 Waveforms of selected nodes in Figure 3.1

All the transistors working as switches controlled by control signals in Figure 3.1, Figure 3.3 and Figure 3.9 have minimal dimensions. The only differences are the transistors for the inverter in Figure 3.1. They have different sizes than others because they convert an analog signal into a digital signal. Note that the transistor sizes are for the 28-nm FD-SOI CMOS technology from ST Microelectronics. The sizes of the transistors are summarized in Table D.1.

Dimensions (nm)	For switches		For the inverter	
	PMOS	NMOS	PMOS	NMOS
Width	160	80	480	80
Length	30	30	30	30

Table D.1 Transistor sizes