Waseda University Doctoral Dissertation

Study on Low Energy True Random Number Generator with Latch-based Core and von Neumann-based Post-Processing for Hardware Security

Ruilin ZHANG

Graduate School of Information, Production and Systems
Waseda University

May 2022

Abstract

Information security has become a critical issue in the IoT era due to a large number of edge devices communicating with servers and each other in an environment where human involvement is weak. That requires embedded cryptographic systems to protect confidential data. And true random number generator (TRNG) is an essential component of it. TRNG utilizes random physical phenomenon to generate unpredictable true random numbers, which can be used as secure keys, nonce, and so on.

There have been some requirements in TRNG. First, for a cryptographic application, TRNG must have high randomness output under PVT (process, voltage, and temperature) variations. Robustness against intentional attacks is also required. Second, the energy-constrained IoT devices motivate low energy TRNG. Therefore, this work targets to design low energy TRNG with high randomness and robustness.

Typically, a TRNG consists of a TRNG core and a post-processing block. The TRNG core utilizes physical noise and generates raw n bits with defects. Then the post-processing block is applied to remove these defects and generate m bits with high randomness. The extraction efficiency (ExE) is defined as m/n. And the total energy of TRNG (E_{TRNG}) is calculated as: $E_{\mathrm{CORE}}/ExE + E_{\mathrm{POST}}$, where E_{CORE} and E_{POST} are the energy of TRNG core and post-processing, respectively. Therefore, in order to design a low energy TRNG, both low E_{CORE} and E_{POST}, and high ExE are required.

For low energy and high ExE post-processing, N-bits von Neumann (VN_N) is a candidate. It can potentially reduce the dynamic power by processing N-bits at the same time. As the N value increases, its ExE is increased. But the mapping table complexity increases exponentially (2^N), which has been an obstacle for hardware implementation.

In the field of TRNG core, latch-based TRNG potentially provides low energy solution, thanks to the simple structure and only one-time voltage transition for data generation. However, it suffers mismatch-induced entropy drop, which requires complex calibration and feedback circuits. Or 256 latches are needed for obtaining 1-bit output. Both of them consumes a lot of energy.

Based on these considerations, in this dissertation, high ExE and low energy post-processing based on VN_N approach, and feedback control free low energy latch-based TRNG core are studied.

As for the post-processing, the mapping table complexity in VN_N is solved in three levels: At the algorithm level, a waiting strategy is proposed. By gathering two newly introduced waiting flags and generating output bits, high *ExE* with a small N value is achieved. At the architecture level, a Hamming weight-based structure is proposed to reconstruct the large table using smaller tables based on Hamming weight. At the logic level, an input-symbol-based code assignment using input codes as outputs is proposed for logic reduction. An 8-bit von Neumann with waiting (VN_8W, 62.21% *ExE*) is designed and confirmed in real chip.

As for the TRNG core, a low energy and high robustness latch-based TRNG core is proposed. It removes the calibration and feedback circuits by two novel methods: mismatch self-compensation and random noise enhancement. The mismatch self-compensation is realized by placing the initial state point close to the metastable point using newly added gate capacitance. In contrast with the conventional fixed initial state point, the proposed initial state point follows the metastable point, which changes position in response to mismatch variations. For noise enhancement, damped oscillation using large resistor is applied for the first time.

Finally, VN_8W is combined with the latch-based TRNG core to build a total TRNG. The performance of the total TRNG is verified by real chip.

The dissertation contains five Chapters as follows:

Chapter 1 briefly introduces the security issues in the IoT era and the random number generators in hardware security.

Chapter 2 shows the TRNG design requirement with introduction of previous works on post-processing and TRNG core. Then, the motivation and concept of this research are presented.

Chapter 3 describes the proposed N-bit von Neumann post-processing with high *ExE* and low energy. First, a light-weight 4-bit von Neumann (VN_4) using input-symbol based code assignment is presented. The logic complexity is roughly reduced to 2/16 times than the conventional code assignment. Then, the concept of waiting strategy is shown. VN_4 with waiting (VN_4W) achieved 46.88% *ExE*, which is higher than conventional 6-bit von Neumann (VN_6, 41.67% *ExE*). Targeting more than 50% *ExE*, hardware implementation of VN_8W with 62.21% *ExE* is presented. Using the Hamming weight-based structure, the conventional 2^8 complexity mapping table is rebuilt

with two identical 4 Bits Logic (2^4 complexity) and an 8 Bits Logic (5^2 complexity). Therefore, the mapping table complexity is roughly improved 4.5 times.

Fabricated in 130-nm CMOS, combined with the clock gating technique, VN_8W achieves low energy of 0.18 pJ/bit at 0.45 V, 1 MHz. Compared with previous work based on iterated von Neumann, it achieves more than 20% energy reduction at identical supply voltage.

Chapter 4 shows the proposed low energy latch-based TRNG. The TRNG core features mismatch self-compensation and random noise enhancement. First, the proposed entropy source latch circuit is described. Then, the mismatch self-compensation is introduced. It is achieved by placing the initial state point close to the metastable point, and 63.3% mismatch is self-compensated by newly added gate capacitance. The noise enhancement in damped oscillation mode by introducing RC delay in the feedback loop of each inverter is presented. The noise is 3 times enhanced by the newly added large resistor. As a result, the TRNG core exhibits 6σ robustness against process variations with only 4 entropy source latches. This is 1/64 times smaller than the conventional work (256 latches).

The total TRNG, including VN_8W, is verified with chips fabricated in 130-nm CMOS. It operates across a wide voltage (0.3–1.0 V) and temperature (–20°C–100°C) range. Cryptographic-grade randomness is verified by NIST SP 800-22 and 800-90B IID tests. Power noise injection resilience is also demonstrated. An aging test reveals the equivalent 11-year life of the TRNG. It achieves the state-of-the-art minimum energy consumption of 0.186 pJ/bit at 0.3 V. These measurement results demonstrate that the TRNG has high randomness under PVT variations and low energy consumption, suitable for energy-constrained IoT devices.

Chapter 5 concludes the dissertation.

Contents

List of Tables

List of Figures

Abbreviations and Symbols

γ	Voltage Exponent Factor
α	Significance Level
σ_n	Standard Deviation (Root Mean Square) of the Noise Voltage
$\Phi(x)$	CDF of the Normal Distribution
ACF	Autocorrelation Factor
ADC	Analog-to-Digital Converter
AES	Advanced Encryption Standard
CDF	Cumulative Distribution Function
CMOS	Complementary Metal-Oxide Semiconductor
CI	Confident Interval
d	Mismatch between Pair Inverters
e	Bias
E_a	Activation Energy
ES	Entropy Source
ExE	Extraction Efficiency
F	Feature Size
H	Shannon Entropy
H_∞	Min-entropy
i.i.d	Independent and Identically Distributed
IoT	Internet of Things
IVN	Iterated von Neumann
k	Boltzmann's constant
LFSR	Linear Feedback Shift Registers

NIST	National Institute of Standards and Technology
TAF	Thermal Acceleration Factor
T_{op}	Operating Temperature
T_{stress}	Stressed Temperature
TRNG	True Random Number Generator
p	Probability of "1"
q	Probability of "0"
PRNG	Pseudo Random Number Generator
PPMA	Privacy Preserving Mutual Authentication
PUF	Physically Unclonable Function
PVT	Process, Voltage, and Temperature
$P-value$	Test Result in NIST
PSD	Power Spectral Density
RO	Ring Oscillators
SHA	Secure Hash Algorithm
SRAM	Static Random Access Memory
XOR	Exclusive-OR
VN	von Neumann
VN_N	N-bit von Neumann
VN_8W	8-bit VN with Waiting Strategy
VN_4W	4-bit VN with Waiting Strategy
VAF	Voltage Acceleration Factor
V_{DD}	Supply Voltage
V_{op}	Operating Voltage
V_{stress}	Stressed Voltage
Vth	Threshold Voltage

Chapter 1

Background

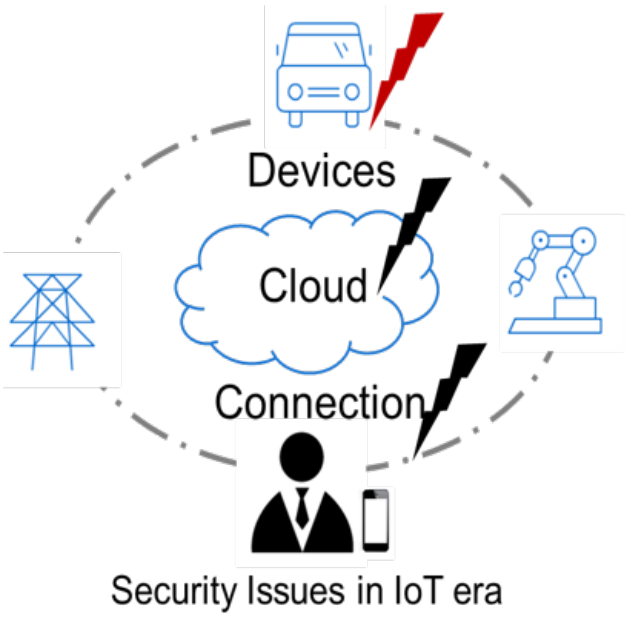


FIGURE 1.1: Security issues in IoT era.

1.1 Hardware Security Requirement in IoT era

Our life is becoming convenience with the help of the internet of things (IoTs). Its applications include smart vehicles, smart home, healthy care, production management, etc. As shown in Figure 1.1, many devices are wireless connected with each other. The user can remotely control the devices by just operating a tablet or mobile phone [1]. However, it also introduces many security crises. These devices are exposed to an environment where human involvement is weak. Thus, to protect the confidential data, the embedded hardware security [2],[3] is required.

Furthermore, the IoT devices are usually energy-constrained [4, 5, 6]. That is because the IoT devices are powered by independent battery or energy harvester. The lifetime of battery is limited. The power harvested from ambient is sometime unstable. Therefore, low energy hardware security solution is required. Furthermore, low operating voltage is preferred for improving the lifetime of the battery [7, 8].

1.2 Random Number Generators in Hardware Security

Embedded cryptography systems are wildly used to protect the security of IoT devices. High randomness random numbers (RNs) are the foundation of cryptography systems. They are used as the authentication keys, session IDs, nonce, etc. Basically, the RNs can be generated through two ways: by pseudo random number generator or true random number generator.

1.2.1 Pseudo Random Number Generator

Pseudo random number generator (PRNG) is also known as deterministic random bit generator (DRGB). It uses a algorithm to generate the "random looking" bitstreams. According to the security level, basically, there are two kinds of PRNGs [9]. One is non-cryptograpgically PRNGs, such as, linear congruential generator, XORShift generator, and permuted congruential generator. These PRNGs have advantage of simple structure. However, they lack of the prediction resistance, due to the linear properties of the algorithms. Therefore, they are unsuitable for cryptographic applications.

The other is cryptographically PRNGs. The NIST SP 800-90A[10] compliant PRNGs are commonly used: Hash_DRGB and HMAC_DRGB based on hash functions; CTR-_DRGB based on block ciphers. The RNs are generated based on current internal state of the algorithm. Therefore, even when the attacker knows the previous bits, it is hard to predict the future output bits. For example, if the internal states has 128 bits, the probability to obtain the correct output is $1/2^{128}$. However, the internal state are instantiate by a seed. If the seed is stolen or can be manipulated by an attacker, the internal state is determined, as is the output. As a solution, the seed is required to be true random numbers (TRNs) generated by a true random number generator (TRNG). In addition, reseed function is needed to periodically update the seed and generate a new internal state. Thus, the security strength of these PRNGs are determined by the randomness of the TRNs.

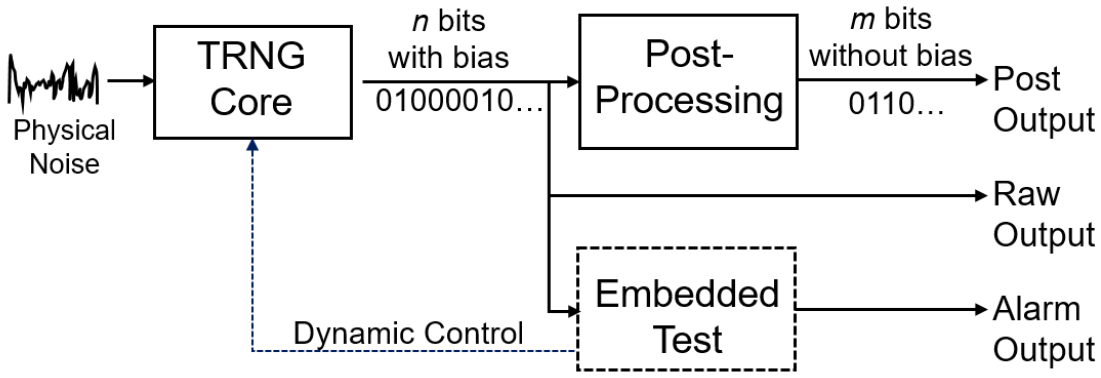


FIGURE 1.2: Block diagram of the TRNG used in a cryptographic system.

The cryptographically PRNGs guarantee high statistical quality outputs with high throughput efficiency. However, the circuits are complex and power-consumed due to a lot of complex mathematical operations and several cycles for internal state updates. That makes them unsuitable for being used in resource-constrained IoT devices.

1.2.2 True Random Number Generator

On the other hand, by utilizing uncontrollable physical phenomena, such as, thermal noise, true random number generator, TRNG can generate unpredictable RNs, serving as the root source of hardware security. However, due to the process, intentional and unintentional environmental variations, the operation of TRNGs are not always stable. The generated RNs are often suffered from bias, correlations, and other imperfection statistics. The poor quality of RNs can't be directly used in cryptographic system. In this case, post-processing techniques are needed to de-bias or to extract full entropy bitstream.

Figure 1.2 shows the block diagram of a TRNG used in cryptographic system [11]. Basically, it consists of TRNG core block, post-processing block, and embedded test block. The TRNG core harvests the physical noise and generates n bits raw data with some bias. Then, the raw data is fed into a post-processing block, which is a kind of data compression technique. After that, m bits without bias data is generated. The extraction efficiency (ExE) of the post-processing is defined as m/n. In addition, the raw data

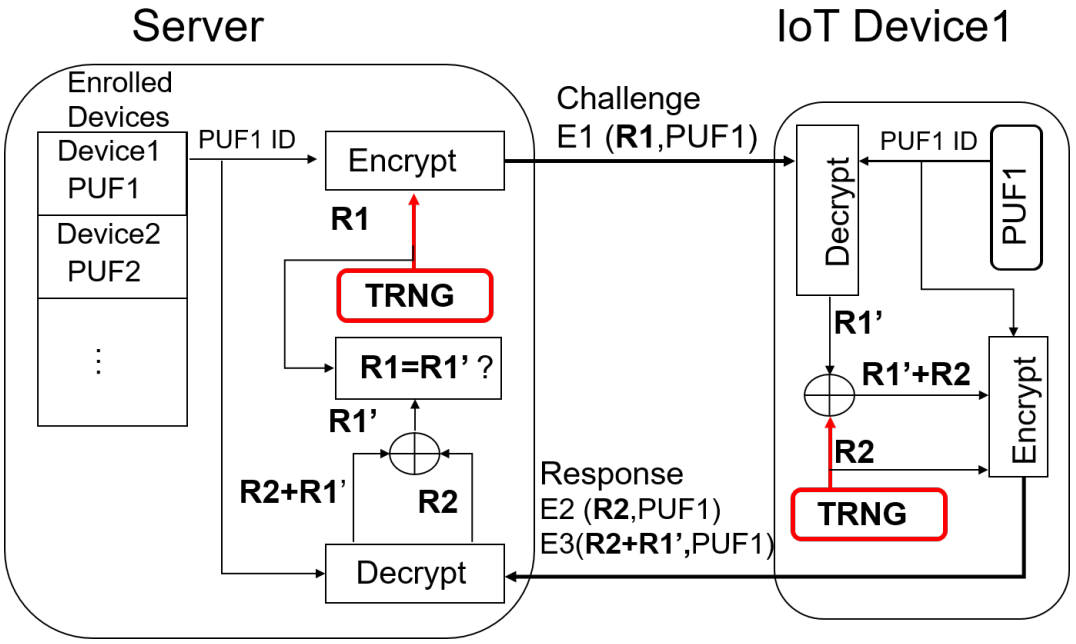


FIGURE 1.3: TRNG application in privacy preserving mutual authentication protocol.

may be fed into embedded test block, which is a kind of hardware implementation of statistical tests. It may generate a alarm signal to indicate the poor quality of the raw data or send feedback signal to control the operation of the TRNG core. Due to the hardware complexity, the embedded test block is usually optionally.

Targeting for being used in resource-constrained IoT devices, the TRNG not only needs to maintain high randomness under diverse working environments, but also needs to be low energy.

1.3 True Random Number Generators in Authentication in IoT Devices

TRNGs provide security keys, nonce, session IDs in cryptographic system. This section shows an implementation example of the TRNGs in authentication system. Figure 1.3 shows the concept of a privacy preserving mutual authentication (PPMA) protocol [12]. On the server side, there are stored enrolled PUF (physically unclonable function) IDs, encrypt and decrypt engines, and a TRNG. On the IoT device side, there are a PUF

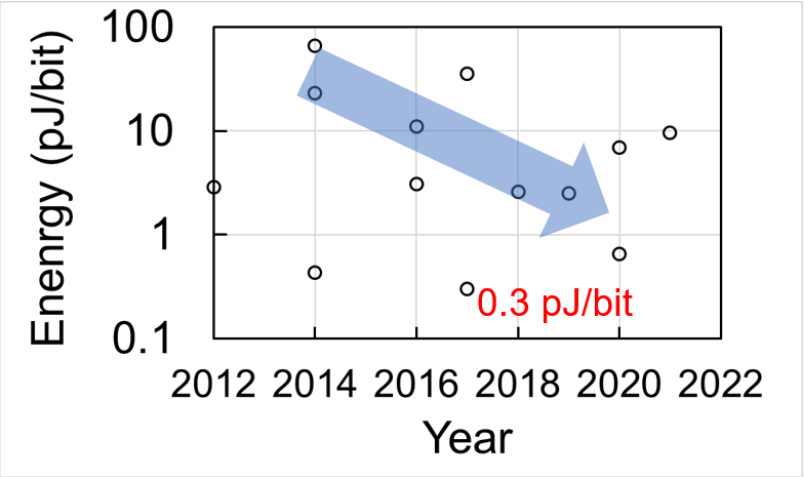


FIGURE 1.4: Energy consumption of TRNG in recent years.

block, encrypt and decrypt engines, and a TRNG. In the authentication phase, the TRNG on the serve side generates a random bitstream R1. Then, R1 is encrypt by PUF1 ID. The encrypt value is sent to the IoT device as challenge. After receiving the challenge, the IoT deceive decrypts the challenge with the PUF1 ID generating on the fly. The obtained R1' is XORed with a random bitstream R2 generated by TRNG on the device side. After that, two kinds of encrypt sequences are sent to server as responses. On the server side, PUF1 ID is used to decrypt the responses. Finally, if R1 equals to R1', the device passes the authentication.

In the conventional authentication system, the TRNG is only embedded on the server-side. And therefore, only the server can authenticate the IoT device. In the PPMA scheme, the IoT device also can authenticate the server thanks to the embedded TRNG on the device side.

1.4 Current State of True Random Number Generators

Figure 1.4 shows the current state of state-of-the-art TRNGs in the aspect of energy consumption. The minimum energy consumption is 0.3 pJ/bit. Therefore, to be used in energy-constrained IoT devices, there is still design space for further low energy solution. This is also the target of this research.

Chapter 2

Preliminaries

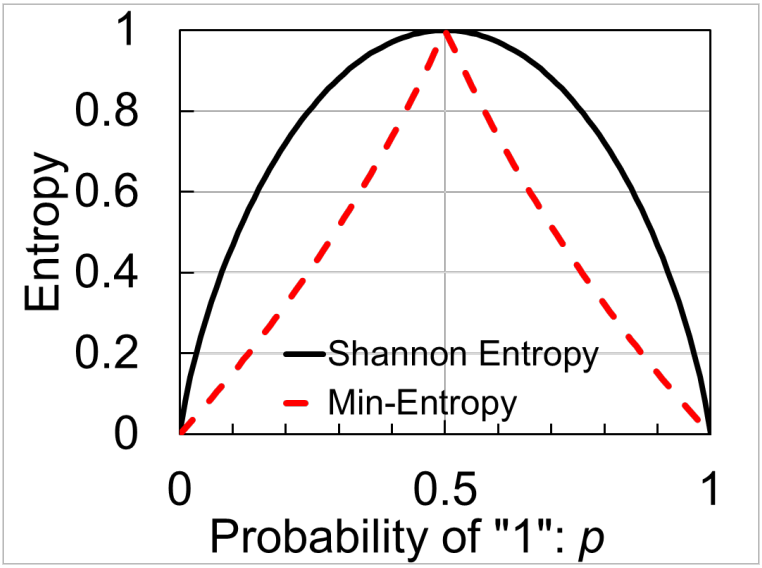


FIGURE 2.1: Entropy versus probability of "1".

2.1 TRNG Requirement

To design a TRNG suitable for IoT devices, there are many requirements: high randomness is the first priority. It requires high robustness against environmental variations and intentional attacks. Long-term reliability is also needed. Then, according to the research target, the performance aspects of energy, throughput, and area cost also should be considered. The details for each consideration are shown in the following sub-Sections.

2.1.1 Randomness Metrics

In this section, the relationship between the probability of "1", bias, and entropy in the TRNG output are introduced. After that, the autocorrelation calculation process is presented. The randomness checks based on statistical analysis suites developed by NIST are introduced later.

2.1.1.1 Bias and Entropy

High Randomness is the first priority of a TRNG. The probability of output "1", p, $(0 \leq p \leq 1)$ is 0.5 in ideal. In reality, p is not always 0.5. To quantify how far the real p

is away from the ideal value, bias e is used, as shown below.

$$e = |p - 0.5|. \tag{2.1}$$

e takes the value ranging from 0 to 0.5. If $e = 0$, it means no bias or $p = 0.5$. If $e = 0.5$, it stands for $p = 1$ or $p = 0$. The extraction efficiency of several kinds of post-processing techniques is inversely proportional to the bias. That means, the *ExE* is high/Low when input bias is small/large.

For the entropy estimation, Shannon Entropy is a well-known method, as shown in Equation (2.2). It has a symmetrical distribution and achieves the maximum value at $p = 0.5$, as depicted in Figure 2.1.

$$H = p * log_2(\frac{1}{p}) + (1 - p) * log_2(\frac{1}{1 - p}). \tag{2.2}$$

On the other hand, by solely utilizing the most likely probabilities, p or $1 - p$, the min-entropy is obtained:

$$H_\infty = -log_2(max(p, (1 - p))). \tag{2.3}$$

Min-entropy is always no greater than Shannon entropy (see Figure 2.1). It is usually used in the cryptographic field as a lower bound entropy requirement of keys. For example, For a 256 bits secret key, the required min-entropy is 0.998 per bit and the related p should locates between 0.42 and 0.58 [2].

2.1.1.2 Autocorrelation

Bits correlation degrades the randomness of the raw data. That means, if the bitstream is self-correlated, the Shannon entropy or min-entropy calculated by p is overestimated. For example, if the sequence is "0101010101...", although both Shannon entropy and min-entropy is 1, the sequence can be predicted. Therefore, an autocorrelation factor (ACF) check for adjacent bits/lags is required.

Pearson correlation for two bitstreams X, Y is a well known formula, as shown in Equation (2.4).

$$R_{X,Y} = \frac{Cov(X,Y)}{\sigma_X \sigma_Y}, \tag{2.4}$$

where $Cov(X,Y)$ is the covariance of X and Y. σ_X or σ_Y is the standard deviations of X and Y, respectively.

For single bitstream X: X_0,X_1,...X_t,..,X_{t+k},...,X_N, the lag-k ACF factor, is a Pearson correlation coefficient between X_t and X_{t+k}. Therefore, by replacing X with X_t, Y with X_{t+k}, σ_Y with σ_X, the ACF function is obtained as shown below.

$$\begin{aligned} R_{X_t,X_{t+k}} &= \frac{Cov(X_t, X_{t+k})}{\sigma_X^2} \\ &= \frac{\sum_{t=0}^{t=N-k}(X_t-\mu)(X_{t+k}-\mu)}{\sum_{t=0}^{t=N}(X_t-\mu)^2}, \end{aligned} \tag{2.5}$$

where μ is the mean value of X. If the ACFs are always near to zero and locate within 95% confidence boundary, it indicates that the bitstream does not have the correlation problem.

2.1.1.3 NIST SP 800-22 Check

The national institute of standards and technology (NIST) SP 800-22 statistical test suite [13] is generally used to test the randomness of a TRNG. It includes 15 statistical tests. The randomness of an input bitstream is verified from multiple mathematical aspects. Passing 15 tests is the basic requirement of a TRNG.

Each test result is formulated into a $P-value$. Only when $P-value > \alpha$, the test is passed. α is the significance level. Typically, α is set to 0.01, which indicates a confidence of 99%. In other words, if 100 sequences are tested, 1 sequence would expect to be failed the test.

The recommendation input size of the bitstream for 15 tests is summarized in Table 2.1. Typically, the bitstream length is 1M (1,000,000) in order to be checked by all tests.

TABLE 2.1: Recommendation of input bitstream size in NIST SP 800-22

#	Test Terms	Input Minimum Size
1	Frequency	100
2	Block Frequency	100
3	Runs	100
4	Longest Runs	128(block = 8)
5	Rank	38912(Matix is 32*32)
6	FFT	1000
7	Non-Overlapping Template	72
8	Overlapping Template	1M
9	Universal	387840
10	Linear Complexity	1M
11	Serial	1M
12	Approximate Entropy	100
13	Cumulative Sums	100
14	Random Excursions	1M
15	Random Excursions Variant	1M

Besides, for each test term, multiple bitstreams can be tested. The pass ratio is defined as n/m, where m is the total bitstream count, n is the bitstream count with $P-value \geq 0.01$. The acceptable pass ratio using the confidence interval is shown in below:

$$accptable\ range = \hat{p} \pm 3\sqrt{\frac{\hat{p} * (1-\hat{p})}{m}} \tag{2.6}$$

where, $\hat{p} = 1-\alpha$, $\alpha = 0.01$. Thus, the minimum n/m should no less than the lower bound, as:

$$n/m \geq \hat{p} - 3\sqrt{\frac{\hat{p}(1-\hat{p})}{m}} \tag{2.7}$$

The acceptable pass ratio for different m values are summarized in Table 2.2. When m is less than 10, no failure is required. When $10 \leq m \leq 31$, one failure is tolerated. When $32 \leq m \leq 63$, two failures are acceptable. When m is greater than 100, four failures are tolerated.

TABLE 2.2: Acceptable pass ratio in NIST SP 800-22

m	Acceptable n/m	Calculated Passed n	Acceptable Failures $m-n$
1	0.691503769	0.691503769	0
4	0.840751884	3.363007538	0
9	0.890501256	8.014511307	0
10	0.895607204	8.956072037	1
12	0.90383156	10.84597872	1
16	0.915375942	14.64601508	1
31	0.936388494	29.02804332	1
32	0.937232823	29.99145033	2
63	0.95239301	60.00075962	2
64	0.952687971	60.97203015	3
99	0.96	95.04	3
100	0.960150377	96.01503769	3
101	0.960298515	96.99015	4

2.1.1.4 NIST SP 800-90B IID Check

In order to guide how to design and test the entropy source used for cryptographic applications, NIST developed another recommendation, named NIST SP 800-90B [14]. It includes entropy source validation, health tests, IID check, and estimating min-entropy. The entropy source validation is divided into IID track and Non-IID track. The IID track is used for the entropy source, which generates *i.i.d* (independent and identically distributed) bits. Note that, the entropy source in a TRNG core with an arithmetic post-processing block, indicates the post-processing block output.

Figure 2.2 presents the flow chart of entropy source estimation strategy for IID track. First, 1,000,000 consecutive binary bits are required in the data collection process. Then the IID check are implemented. It consists of Permutation tests and additional Chi-square statistical tests. At the initial of the IID check, a min-entropy estimating, H_{original} is obtained based on the most common value estimate, H_{∞}_estimate as shown in below:

$$H_{\infty}\text{\_estimate} = -\log_2 p_u, \tag{2.8}$$

where p_u is the probability of the most common value, which can be calculated by Equation (2.9):

$$p_u = min(1, p + 2.576\sqrt{\frac{p(1-p)}{N-1}}), \tag{2.9}$$

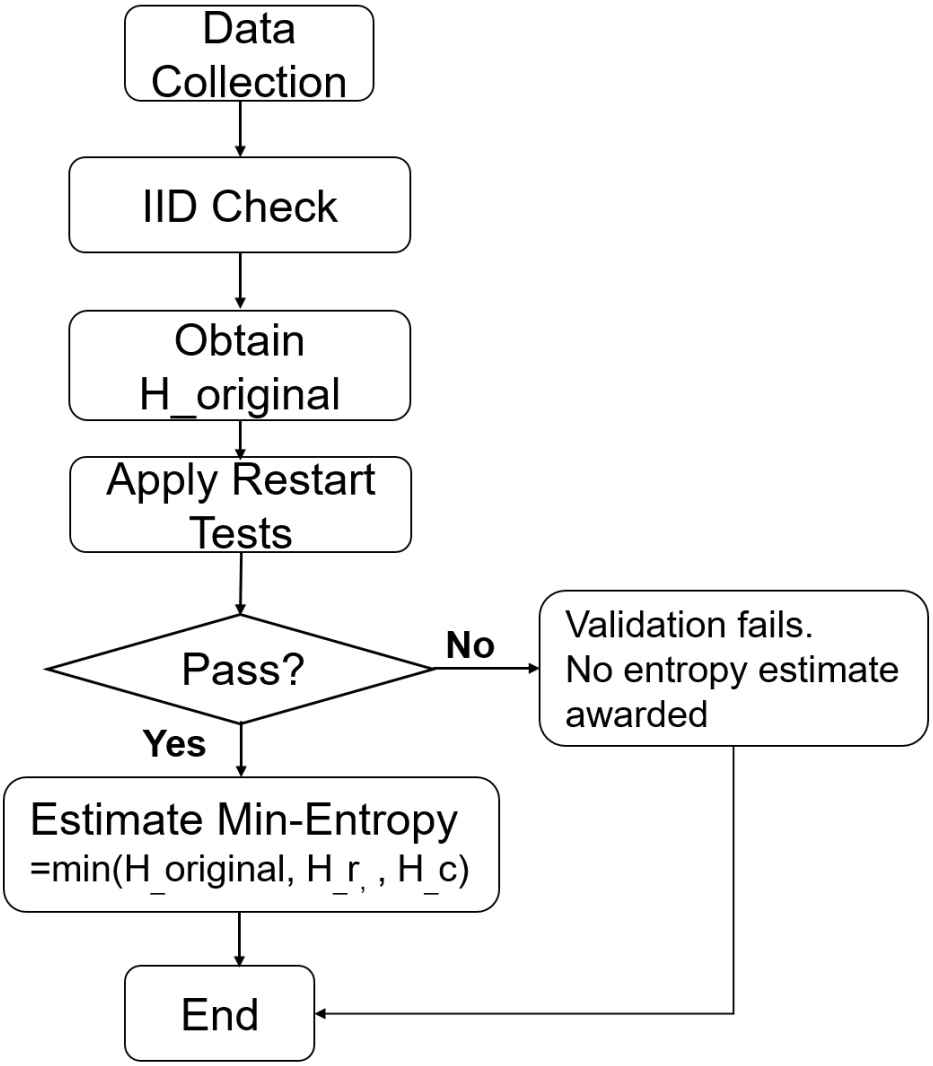


FIGURE 2.2: Entropy estimation strategy for IID track.

where p is the probability of "1" in the tested input bitstream. N is the bitstream length. 2.576 corresponds to 99% confidence interval.

After that, considering the correlation between sequences after restarts might leads to a overestimate of the H_{original}, restart tests are implemented. The input bitstream is created by restarting the entropy source 1000 times and each time collecting 1000 consecutive bits and combining them into one string. It is initialized into a 1000 by 1000 matrix during the restart tests. Then, the min-entropy estimate for row (H_r) and column (H_c) datasets using Equation (2.8) are obtained. If $min(H_r, H_c) < 0.5 * H_{\text{original}}$, restart tests are failed. No entropy estimate is awarded. Otherwise, the final estimate min-entropy is $min(H_r, H_c, H_{\text{original}})$.

2.1.2 Robustness Against PVT variations

In order to be used in IoT devices, which have a diverse operating environment, TRNGs must have high robustness against processes (P), voltages (V), and temperatures (T) variations. For a latch-based TRNG, it should have more than 6σ robustness against random mismatch variations.

2.1.3 Long-Term Reliability

For real life applications, the TRNG must have reliability against wear-out. An accelerating aging test [15], [16] is used to verify the long-term effect. The total estimated acceleration factor is a product of the thermal acceleration factor (TAF) and voltage acceleration factor (VAF), as shown in equation 2.10 and 2.11, respectively.

$$TAF = e^{\frac{Ea}{k}(\frac{1}{T_{op}} - \frac{1}{T_{stress}})}, \tag{2.10}$$

$$VAF = e^{\gamma\ (V_{stress} - V_{op})}, \tag{2.11}$$

where E_a is the activation energy with a typical value of $E_a = 0.5$ eV; k (8.62×10^{-5}) is Boltzmann's constant; T_{op}/V_{op} is the nominal operating temperature/voltage; T_{stress}/V_{stress} is the stressed temperature/voltage; γ is the voltage exponent factor. Thus, if the randomness of TRNG does not get worse after x hours of aging test, the effective life is $TAF \times\ VAF \times\ x/(365 \times\ 24)$ years.

2.1.4 Attack Tolerance

To steal the confidential data, the attacker may perform physical attacks to destroy the normal operation of the embedded cryptography systems or record the leakage information by side-channel attacks. The TRNG must provide resistance to physical attacks or remain high randomness even under attacks. Figure 2.3 shows the power noise injection circuit. A sine wave noise with noise peak to peak (Vpp) is coupled into the supply

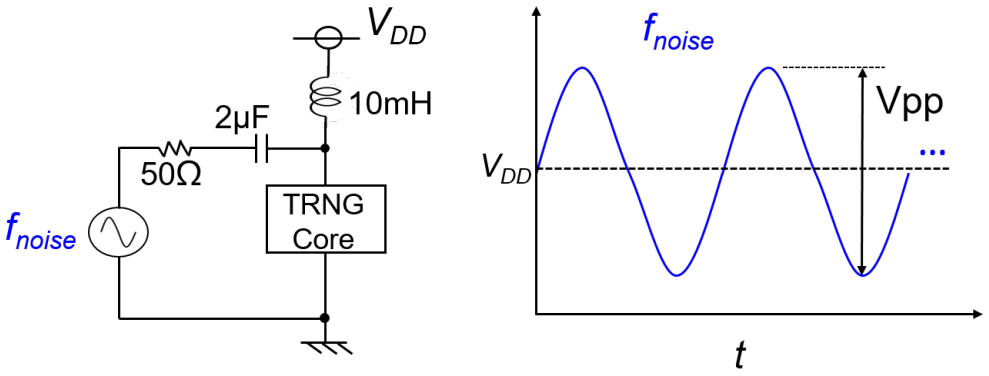


FIGURE 2.3: Power noise injection attack circuit.

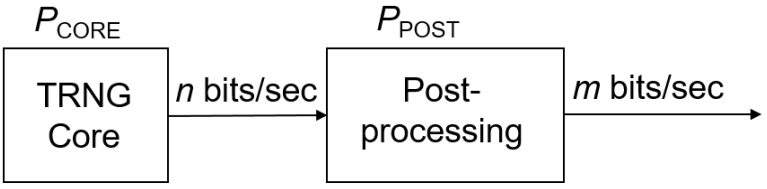


FIGURE 2.4: Power consumption and throughput of a TRNG.

voltage of TRNG core. The ring oscillator based TRNG are reported to be affected by the noise injection attack [17].

2.1.5 Area and Energy Efficiency

IoT devices are typically area- and energy-constrained. The total area and energy consumption of a TRNG consists of the TRNG core block and post-processing block. Considering different CMOS technologies (feature size, F), area normalized with the feature size is used.

$$Area\ in\ feature\ size(F^2) = \frac{Area_{\text{TRNG Core}} + Area_{\text{Post-Processing}}}{feature\ size^2}. \tag{2.12}$$

As for the energy efficiency, Figure 2.4 shows the power consumption and throughput of a TRNG. The power consumption of a TRNG core block and a post-processing block

are denoted as P_{CORE} and P_{POST}, respectively. The related energy consumption is defined as E_{CORE} and E_{POST}, respectively. Their throughput are n bits/sec and m bits/sec, respectively. Given the extraction efficiency, ExE of m/n, the energy consumption of the TRNG can be calculated as:

$$\begin{aligned} E_{\text{TRNG}} &= \frac{P_{\text{CORE}}}{n} + \frac{P_{\text{POST}}}{m} \\ &= \frac{P_{\text{CORE}} * m/n}{n * m/n} + \frac{P_{\text{POST}}}{m} \\ &= \frac{E_{\text{CORE}}}{ExE} + E_{\text{POST}} \ (pJ/bit) \end{aligned} \tag{2.13}$$

Therefore, in order to design a low energy TRNG, low energy consumption in TRNG core and post-processing block and high ExE are needed.

2.2 Previous Post-Processing Techniques

The post-processing techniques can be divided into two categories: cryptographic post-processing techniques and arithmetic post-processing techniques [11]. They are described in the following sections.

2.2.1 Cryptographic Post-Processing Techniques

The cryptographic algorithms such as hashing algorithm-based SHA, block cipher-based AES can be used to mask the defects of raw TRNG data. That is achieved by compressing the raw bitstream into the fixed-length bitstream. However, in order to ensure the quality of the output, the raw data musts meet the minimum entropy requirement. That may requires additional circuits for quantifying the entropy. Furthermore, these kinds of methods are rather complex and not suitable for resource-constrained IoT devices. Some simplified versions, such as strong blenders in [18], BIW in [19], and PRESENT in [20] also can remove the bias and correlation in the raw bitstream. But, these methods are still not power efficient and need long latency. The details of cryptographic post-processing design are out of the scope of this dissertation.

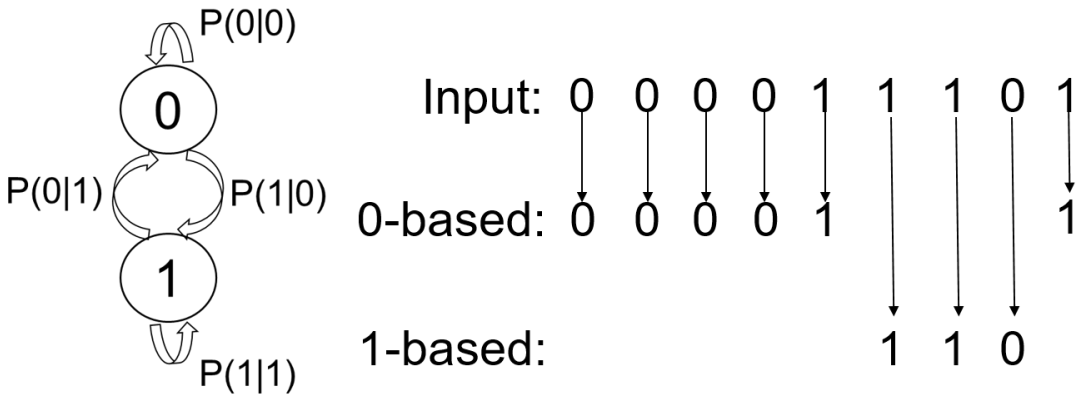


FIGURE 2.5: Two state Markov chain.

2.2.2 Arithmetic Post-Processing Techniques

Arithmetic post-processing techniques provide compact and power-efficient solutions, thanks to the simple arithmetic method and structure. Markov chains [21] and linear feedback shift register [22, 23] based post-processing techniques are often used for decorrelation. The XOR structure is the simplest one with limited de-bias ability [24]. Linear correctors can improve the de-bais ability, but it still can't achieve zero bias [25, 26, 27]. Von Neumann method provides zero bias output with low hardware cost [28]. However, its maximum *ExE* is 25%, which is not suitable for low-energy TRNG. Iterated von Neumann [29] and N-bit von Neumann [30] methods are proposed to improve the conventional von Neumann. The detailed introductions of each post-processing technique are shown in the following.

2.2.2.1 Markov Chain

Markov chain [21] is used to remove the correlations between adjacent bits. It is based on the assumption that the probability of current state only depends on the previous states. In other words, the states have identical previous states are independently with each other.

Figure 2.5 shows the simplest 2 (2^1) state Markov chain, in which the next state only depends on the current state. The transition probabilities of state-0 to state-0, state-0 to

state-1, state-1 to state-1, and state-1 to state-0 are denoted as P(0|0), P(1|0), P(1|1), and P(0|1), respectively. P(0|0)+P(1|0) =1. P(0|1)+P(1|1) =1. Its decorrelation function is as follows: The current bit is sent into state-0 based bitstream if the previous bit is zero. And it is sent into state-1 based bitstream if the previous bit is one. Therefore, the probabilities of one and zero in state-0 based bitstream are P(1|0) and P(0|0), respectively. Because of the identical knowledge of previous state-0, one and zero are independent with each other in the state-0 based bitstream. It is the same story in state-1 based bitstream. The decorrelation is achieved in this way.

The basic 2 state Markov chain can be easily extended to 2^n state Markov chain. In this case, the next state only depends on the previous n states. However, in hardware implementation, it needs 2^n memories to store n-state based bitstreams.

2.2.2.2 Linear Feedback Shift Register

Linear feedback shift register (LFSR) is a N-bit shift register. Its' internal states are a polynomial function of previous N-states. It has a $2^N - 1$ period, except for all zero state. For example, 4-bit LFSR and 16-bit LFSR have 15 ($2^4 - 1$) and 65535 ($2^{16} - 1$) periods, respectively. LFSR can be used as a pseudo-random number, post-processing technique, etc. In this part, its usage as a post-processing technique is presented.

Figure 2.6 shows an example of 4 bits LFSR post-processing. Its polynomial feedback function is summarized in Equation (2.14). The TRNG raw data DIN is fed into the feedback loop by inserting a new XOR function of XOR_2. Thus, the feedback value to x^1 is obtained by XORing x^4, x^3, and the TRNG raw data. The updated x^4 is obtained as post-processed output.

LFSR post-processing technique is easy to implement in hardware. Its output also has high statistical quality. However, the entropy per bit can't be improved. Because, no raw bits are discarded during the process. Therefore, LFSR are usually used as a decorrelation function, such as 4-bit LFSR in [22], 16-bit LFSR in [23]. In addition,

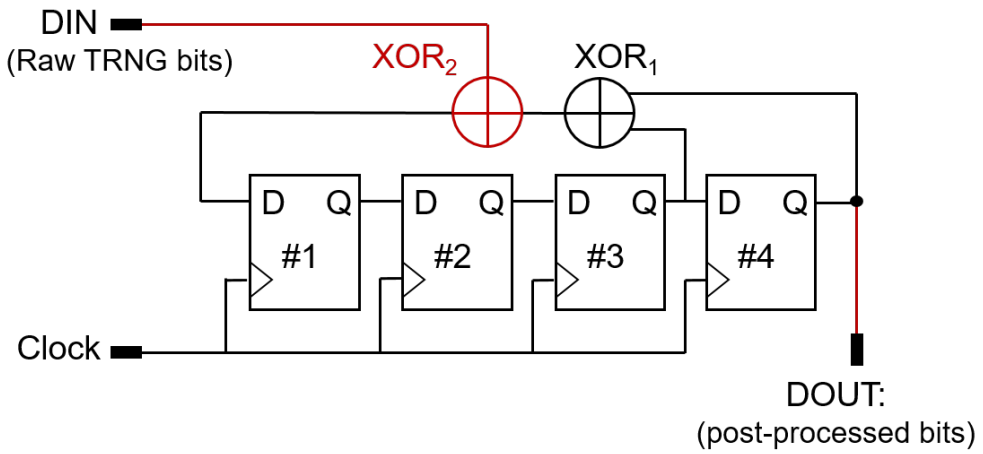


FIGURE 2.6: 4-bit Linear feedback shift register.

the long latency in LFSR consumes large dynamic power, which is unsuitable for low-power TRNG design.

$$\text{LFSR\_4}: \; x^4 + x^3 + 1. \tag{2.14}$$

LFSR also can be used as a post-processing block in TRNG design. In this case, the initial seed is updated every cycle by raw bits. Since the entropy only can be improved by data compressing, thus, LFSR can only improve the correlation in TRNG output.

2.2.2.3 Linear Corrector

The theory of linear corrector for a biased TRNG output is proposed by Lacharme in 2008 [25]. It is based on the error correcting code. It processes n bits and generates m bits. The de-bias ability is verified by:

$$e' = 2^{d-1} e^d, \tag{2.15}$$

where e' is the bias value in new generated m bits, e is bias in the raw n bits, d is the minimum distance of the linear code. For example, the 16 bits to 8 bits linear corrector implemented by Dichtl achieved $e' = 2^4 e^5$ with 50% extraction efficiency [26]. It is much better than $2e^2$ by using 2-bit XOR post-processing. However, the hardware

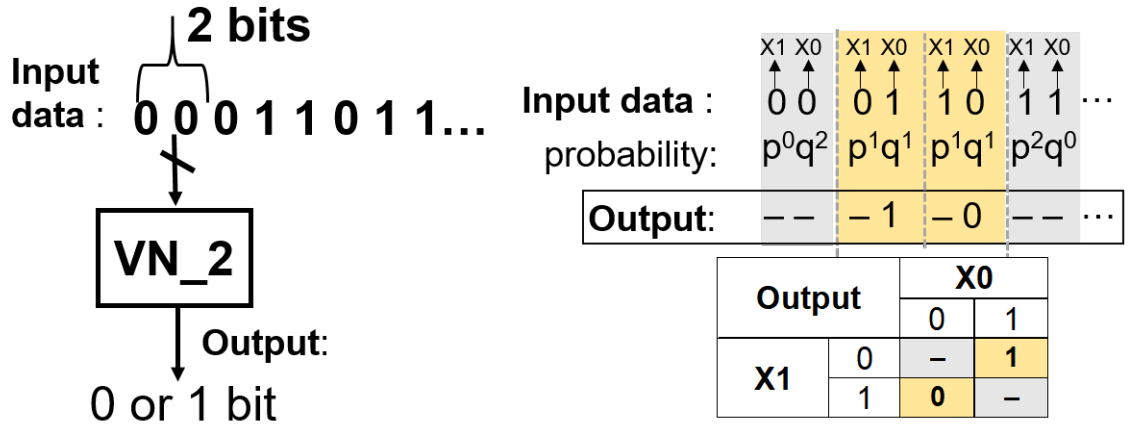


FIGURE 2.7: Structure of conventional von Neumann method.

efficiency is not high. Thus, the work in [27] proposes various BCH codes with $n = 255$. However, it still can't achieve zero bias output.

2.2.2.4 von Neumann Method

The conventional von Neumann (VN_2) method is a well-known post-processing technique. It processes two bits each time and can generate zero-bias bit when the input raw data is independent, identically distributed (*i.i.d.*) [28]. Figure 2.7 illustrates it's working fashion: when the two input bits X1, X0 is "0,1" or "1,0", X0 is saved and the output=X0, is "1" or "0"; When X1=X0, the two bits are discarded. Based on the *i.i.d.* assumption, the probability of X1 ≠ X0 is p^1q^1, where p and q are probability of ones and zeros in input data. Therefore, the probability of Y="0" or Y="1" is identical. Zero-bias output is achieved in this way.

VN_2 has the merit of lightweight. However, its maximum *ExE* is only 25% when the input bias is zero. In addition, its *ExE* is negatively proportional to the input bias, as shown in Equation (2.16) and Figure 2.8. Low *ExE* post-processing technique is not suitable for a low TRNG. To improve the *ExE*, basically, there are two directions: iterated von Neumann method and N-bit von Neumann method.

$$ExE_{\text{VN\_2}} = 0.25 - e^2. \tag{2.16}$$

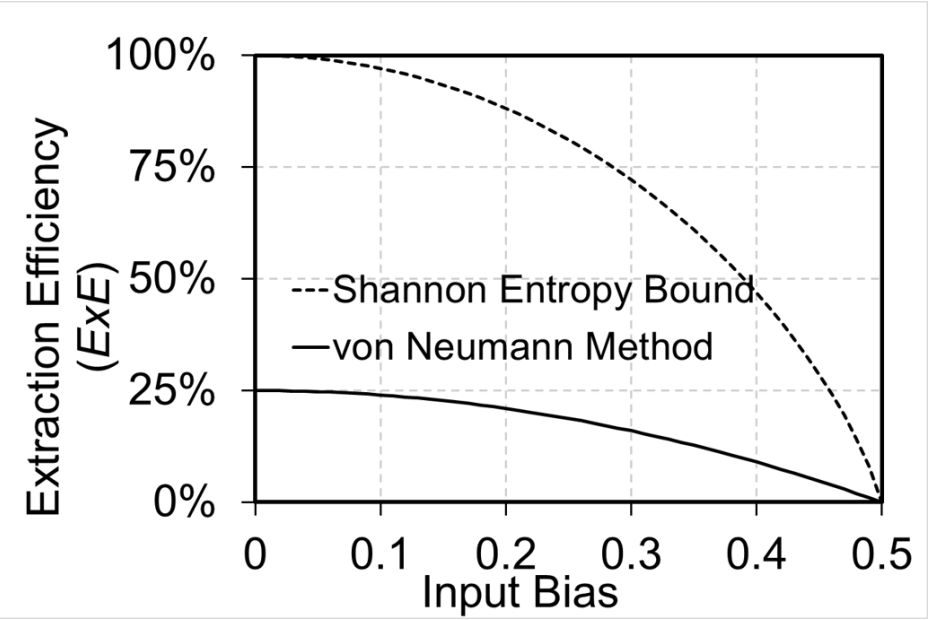


FIGURE 2.8: Extraction efficiency versus bias for conventional von Neumann method.

2.2.2.5 Iterated von Neumann Method

The iterated von Neumann method (IVN) is first proposed by Peres in [29]. Figure 2.9 shows the architecture of IVN. It reuses the discarded information in conventional von Neumann method by introducing XOR and R function. Each module of IVN structure includes three sequences: VN_2 is identical with conventional von Neumann, it generates without bias bits, which can be directly output. XOR sequence indicates the input information is discarded (denoted by '0') or not discarded (denoted by "1"). The R sequence indicates the discard information: "1" when the input is "11", "0" when the input is "0". XOR sequence (bias = $2e^2$) and R sequence (bias=$2e/(1+4e^2)$) are fed into next modules, respectively, for generating new VN_2 bitstreams. Through N times of iterations, the *ExE* can approach Shannon entropy bound.

However, because R sequence has large bias, that leads to a low *ExE* in R sequence. Thus, this kind of structure is not hardware efficient. Therefore, the work in [31] proposes an incomplete tree-based structure under hardware constraints. Figure 2.10 shows an iterated von Neumann with 7 modules (IVN_7) based on the incomplete binary tree

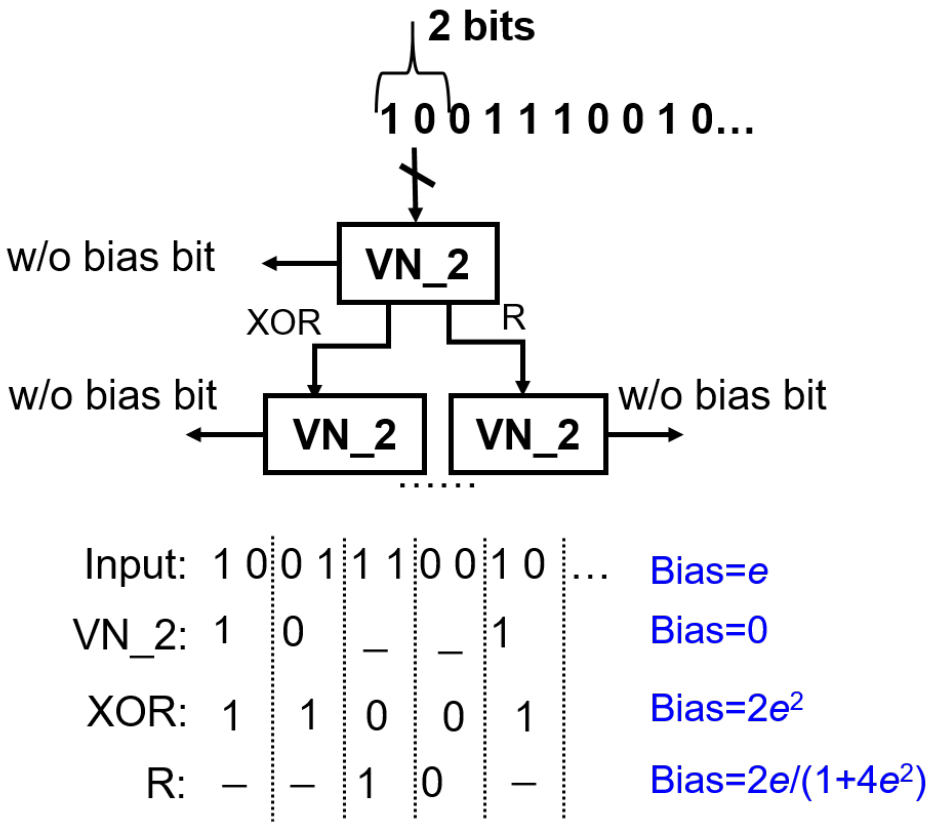


FIGURE 2.9: Structure of iterated von Neumann method.

structure. The work in [23] implemented an iterated von Neumann with 16 modules (IVN_16) for achieving more than 78% *ExE*. In addition, Markov chain is used to decorrelate the raw data before using the IVN_16. An hierarchical IVN was proposed to combine multi-entropy sources, targeting for high throughput [12].

In summary, IVN structure has the merit of regular structure and therefore fewer design efforts. However, the sequential data transition in every cycle in each processing module increases the dynamic power.

2.2.2.6 N-bit von Neumann Method

The concept of N-bit von Neumann post-processing (VN_N) is proposed by Elias in 1972 [30]. It processes N bits simultaneously and generates without bias bits, as shown in Figure. 2.11. By increasing the value of N, the *ExE* can approach Shannon entropy bound. Its work fashion is summarized below. Given an raw bitstream followed

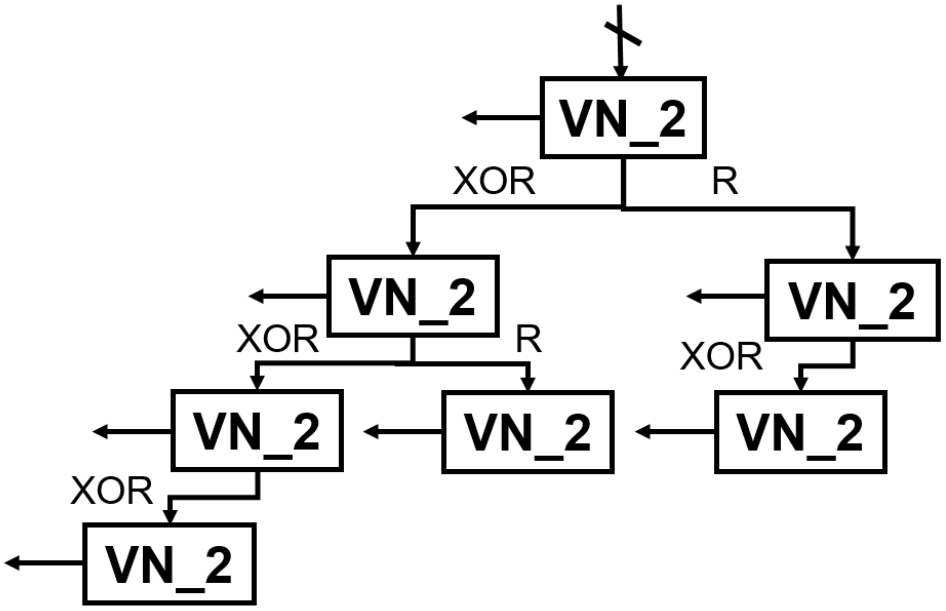


FIGURE 2.10: Structure of iterated von Neumann with 7 modules (IVN_7) based on incomplete binary tree.

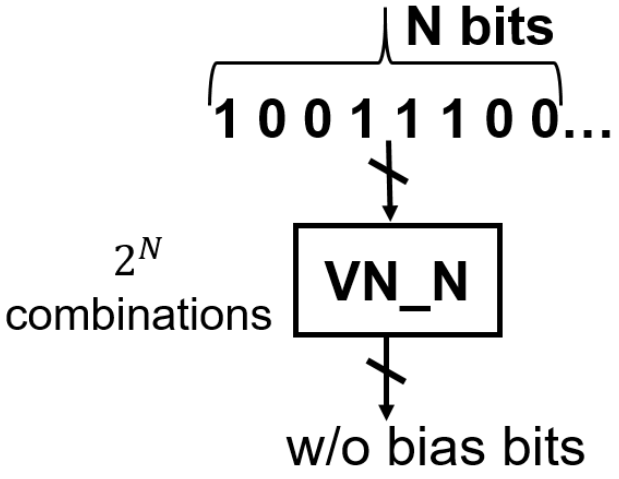


FIGURE 2.11: Structure of N-bit von Neumann method.

i.i.d., the probability of ones and zeros are defined as p and q, respectively. Each consecutive N bits has 2^N combinations and can be divided into $N+1$ groups, according to Hamming weight k, $0 \leq k \leq N$. Each group, e.g., g_k, has group probability of $Prob.(g_k) = p^k q^{N-k} \#g_k$, where $\#g_k$ is the total group member count: $\#g_k = C_N^k$. Same group members occur with the identical probability of $\frac{1}{\#g_k}$. Thus, the Shannon entropy

TABLE 2.3: Theoretical VN_N

Group	Ones Probability	$\#g_k$	Entropy
g_0	q^N	1	0
g_1	p^1q^{N-1}	N	$\log_2 N$
⋮	⋮	⋮	⋮
g_k	p^kq^{N-k}	C_N^k	$\log_2(C_N^k)$
⋮	⋮	⋮	⋮
g_N	p^N	1	0
total	1	2^N	$\sum_{k=0}^{N} p^kq^{N-k}C_N^k \log_2(C_N^k)$

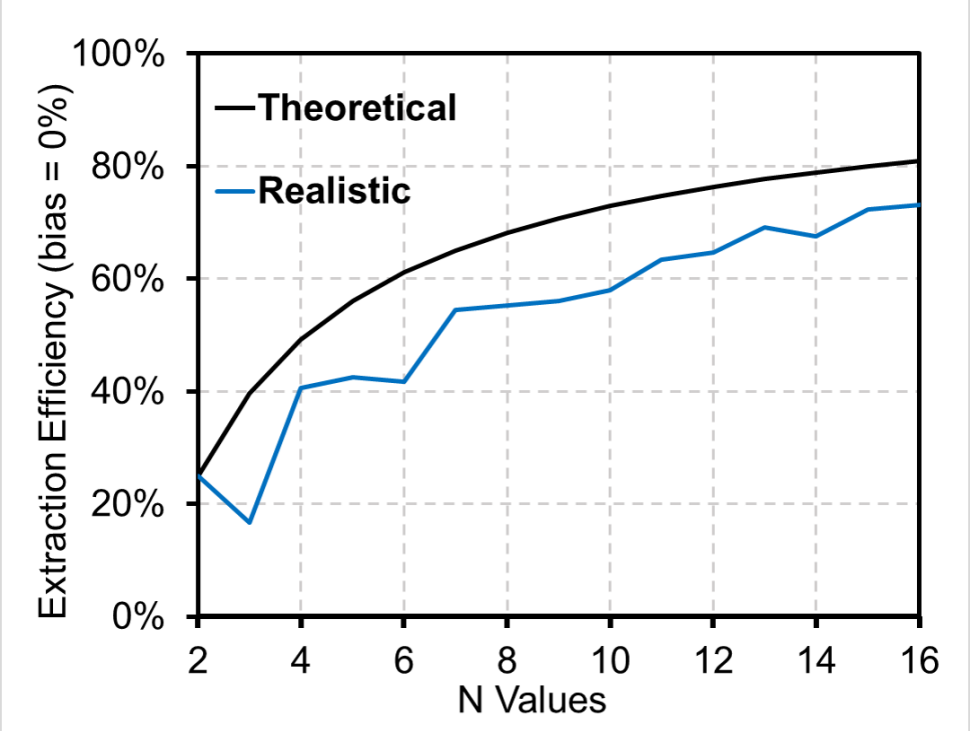


FIGURE 2.12: Extraction efficiency of VN_N.

of the N bits code in this group is:

$$H_{g_k} = -\sum \frac{1}{\#g_k} \log_2(\frac{1}{\#g_k}) = \log_2(\#g_k) = \log_2(C_N^k). \tag{2.17}$$

H_{g_k} is the theoretical random bits count extracted from this group. Table 2.3 summarized the theoretical assigned bits count in each group. By assigning each group with $\log_2(C_N^k)$ bits, the output bits of ones and zeros are balanced. In this way, VN_N

achieved zero bias output. Its *ExE* in theoretical is expressed as:

$$\begin{aligned} ExE\_\text{Theoretical} &= \frac{\sum_{k=0}^{N} Prob.(g_k) H_{g_k}}{N} \\ &= \frac{\sum_{k=0}^{N} p^k q^{N-k} C_N^k \log_2(C_N^k)}{N}. \end{aligned} \tag{2.18}$$

However, $\log_2(\#g_k)$ is not always an integer, while we can only assign integer bits in realistic. In this case, $\#g_k$ is decomposed into the following form:

$$\#g_k = a_{kn}2^n + a_{kn-1}2^{n-1} + \cdots + a_0 2^0 = \sum_{j=0}^{n} a_{kj}2^j, \tag{2.19}$$

where $a_{kj} \in \{0,1\}$ $(0 \leq j < n)$, and $a_{kn} = 1$. Thereby, its ExE in realistic is:

$$ExE\_\text{Realistic} = \frac{\sum_{k=0}^{N} \sum_{j=0}^{n} p^k q^{N-k} a_{kj} 2^j j}{N}. \tag{2.20}$$

As a result, there is an *ExE* drop between theoretical and realistic values, as shown in Figure 2.12. In addition, as N increases, *ExE* becomes saturated. Therefore, an appropriate N value should be selected under the target of *ExE*.

Compared with IVN structure, VN_N can potentially reduce the dynamic power by processing N bits at one cycle time. However, the mapping table complexity in VN_N increases exponentially (2^N). In order to be applied in low-energy TRNG core, low-cost hardware implementations are needed.

2.3 Previous TRNG Designs

Many kinds of physical phenomena have been used as entropy source for TRNG. For example, telegraph noise in [32], Si nanodevices noise in [33], SiN MOSFET noise in [34], and soft gate oxide breakdown noise in [35]. These phenomena exhibit large noise magnitude. However, these noises are directly harvested by amplifiers or counters, leading to a lot of area and power consumption.

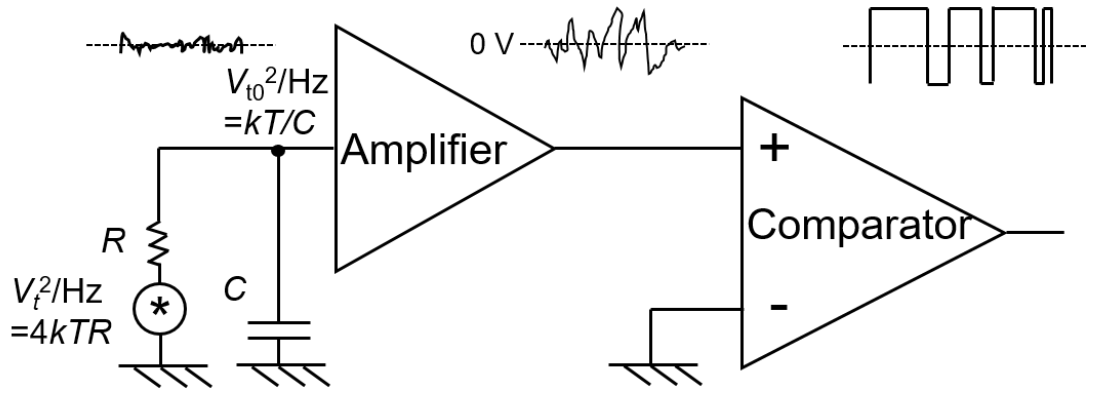


FIGURE 2.13: Topology of direct noise amplification-based TRNG.

On the other hand, thermal noise has been widely used in TRNG. Although the noise magnitude is smaller than the above mentioned noise, it can be directly amplified by amplifiers [36, 37, 38] or by inverters using multi-time amplification [39]. In general, thermal noise is harvested indirectly: jitter in ring oscillators (ROs) [40, 41] or in SRAM leakage current [42], chaotic map based on ADC structure [24, 43, 44], metastability in latches [12, 19, 45, 46, 47, 48, 49] or in sense-amp [23]. The merits and demerits of these TRNGs are described in the following Sections.

2.3.1 Direct Noise Amplification-based TRNGs

Figure 2.13 presents a simplified topology of the direct noise amplification-based TRNG in [36]. It consists of an noise seed, amplifier and a comparator. A larger resistor R with thermal noise voltage V_t serves as the noise seed. The noise power is proportional to R: $V_t^2/Hz = 4kTR$, where k is the Boltzmann's constant, T is absolute temperature. C is the equivalent input capacitance of the amplifier. Therefore, the power of input noise is $V_{t0}^2/Hz = kT/C$, when considering the RC low pass filter effect. The input noise bandwidth defined by 3dB cutoff frequency is negative proportional to the RC product: f=1/(2πRC).

The input noise is further amplified by a high gain amplifier and digitalized by a comparator. It can generate random bits based on the following assumptions: Assume the average noise voltage is 0 V, and the probability that the noise is above and below the

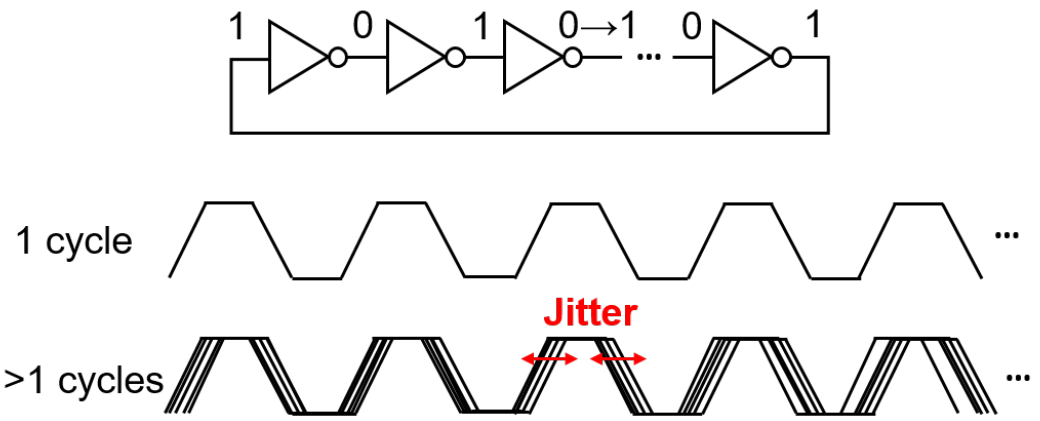


FIGURE 2.14: Concept of ring oscillator based TRNG.

average value is equal. Therefore, after the operation of the comparator, it can output ones or zeros without bias.

However, in order to reduce the un-uniform noise distribution-induced entropy drop, the amplifier needs to be designed with high gain and high bandwidth. This not only increases power consumption but also limits the output noise bandwidth (3.2 MHz). Besides, the systematic bias in the comparator needs to be solved for balanced zeros and ones output. To achieve a lower power consumption and high randomness design, the work in [38] combines the direct noise amplification with a chaos map then followed by a ring oscillator. The prototype TRNG is implemented in 2 μm CMOS with 3.9 mW power consumption at 1 MHz. It is still large for application in power-constrained IoT devices.

2.3.2 ROs-based TRNGs

Figure 2.14 shows the concept of a ring oscillator (RO) based TRNG. When power is on, the inverters begin to oscillate. Clock uncertainty or jitter is accumulated by many cycles of operation due to the noise in the circuit. Therefore, random output can be obtained by sampling the circuit at enough jitter state. ROs based TRNG exploits design simplicity. However, it requires a long period of sampling interval to guarantee the good quality of the jitter.

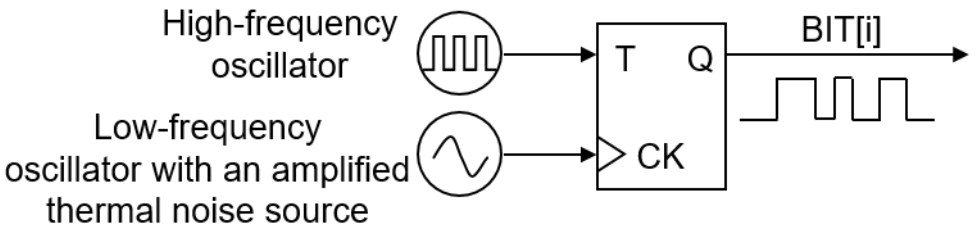


FIGURE 2.15: Architecture of ring oscillator based TRNG.

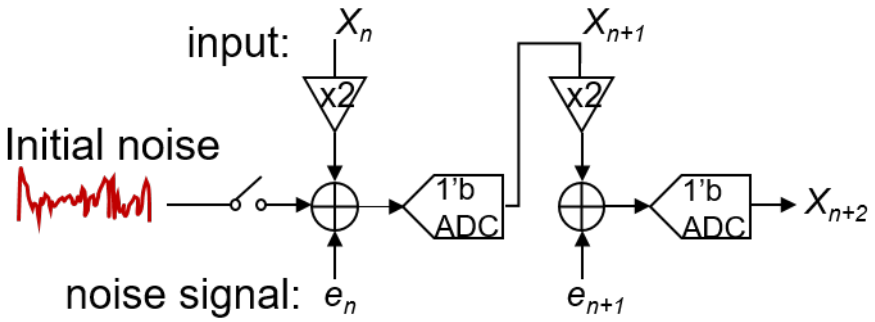


FIGURE 2.16: Bernoulli shift map based on ADC structure.

In the previous work in [40], as shown in Figure 2.15, a slow-frequency oscillator is used to sample a high-frequency oscillator based on a T flip-plop. The low frequency oscillator has an amplified thermal noise source. This enhances the randomness of the output. However, ROs-based TRNG is reported to be vulnerable to power supply noise injection attacks [17]. The improvements include increasing the ROs number, XORing multiple ROs, applying new jitter amplification methods, etc. However, the robustness against process, voltage, and temperature (PVT) is remained unsolved. Thus, the work presented in [41], proposed an all-digital edge racing TRNG with robustness against PVT variations and power injection attack. However, many inverters in RO still consumes a lot of power and area.

2.3.3 Chaotic Map ADC-based TRNGs

Chaotic map is one kind of analog machine. The chaotic map TRNG utilizes the non-reversible and being susceptible to noise characteristics of the analog circuit to generate

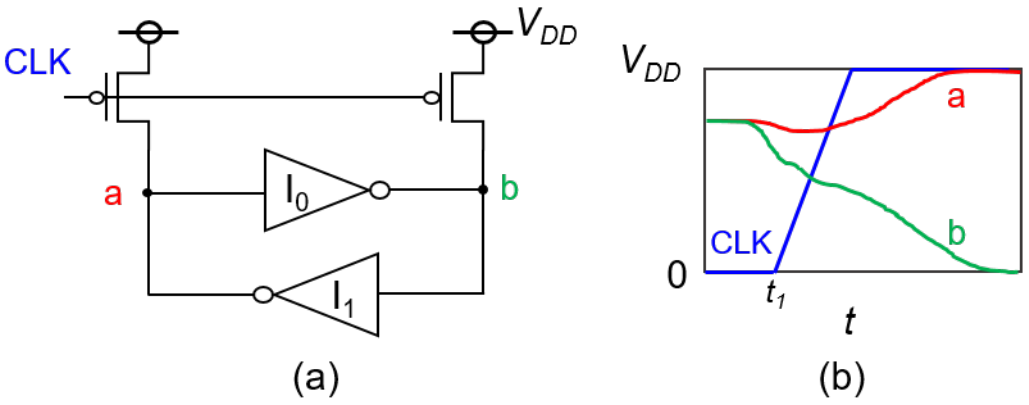


FIGURE 2.17: Latch based TRNG. (a) Basic structure. (b) Voltage transfer curves.

long-term unpredictable random numbers. To realize the chaotic map, Bernoulli shift map using a 1'b analog to digital converter (ADC) is a common approach, as shown in Figure 2.16. The initial noise is utilized to generate the initial state X_0. X_n is the arbitrary input of the system. e_n is the Gaussian noise signal at time n, which can continuously modify the internal state of this system. The basic mathematical principles can be summarized using Equation 2.21 and 2.22. Long-term unpredictable and non-periodic TRNs are obtained in this way. The implementations used 1.5-bit ADC are presented in [43] and [44]. To further reduce the structure complexity, the work presented in [24] used a successive approximation register (SAR) ADC structure and achieved a low energy consumption of 0.3 pJ/bit.

$$f(X_n) = \begin{cases} 2X_n + e_n, & 0 \leq X_n < 1/2 \\ 2X_n - 1 + e_n, & 1/2 \leq X_n < 1 \end{cases} \tag{2.21}$$

$$X_{n+1} = f(X_n) = aX_n + b = f^n(X_0), \quad (X_n \in R) \tag{2.22}$$

2.3.4 Latch-based TRNGs

Latch-based TRNGs are a good candidate for a resource limited application, thanks to its simple structure and low power consumption because of the fact that the data generation process is only one-time transition. The basic structure of a latch-based

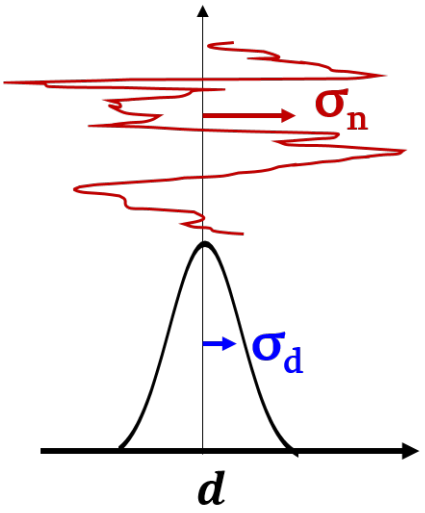


FIGURE 2.18: Mismatch-to-noise ratio.

TRNG is shown in Figure 2.17(a). It consists of a cross coupled inverter pair and two pMOSs to control the operation. First, by setting CLK = "0", the outputs of these two inverters a and b are set to "1". Then, when the CLK signal changes from "0" to "1", a and b first enter into metastable region. According to the differential thermal noise value in a and b side, the final resolution will be either a = "1", b = "0" or a ="0", b = "1", as shown in Figure 2.17(b). If there is no mismatch between two inverters, the probability of ones are expected to be 50%.

However, latch structures suffer from process variations induced mismatch. The final resolution will be biased ones or zeros. To solve it, the work in [47] proposed to use calibration and feedback control loop circuits. However, it consumes a lot of area. A capacitance-based charge-pump in [45], [46] is used. Again, this kind of analog method consumes a lot of area and power. 256 latches are used to generate one output bit in work in [48].

Above mentioned methods aim to reduce the mismatch or effective mismatch with high power consumption. However, the randomness of a latch output is evaluated by the mismatch-to-noise ratio (d/σ_n) [50], as shown in Figure 2.18. d stands for the mismatch. σ_n stands for the root mean square of the noise voltage. A smaller ratio is better for randomness. Therefore, efficient solutions to improve σ_n and reduce d are required for a low energy latch-based TRNG.

TABLE 2.4: Comparison of previous post-processing techniques and target in this dissertation

		Pros	Cons
1	Markov Chain	De-correlation	Memory cost
2	Linear Feedback Shift Register(LFSR)	De-correlation	Power cost
3	Exclusive-OR(XOR)	Low cost Simplest structure	Not zero bias
4	Linear Corrector	Improved de-bias	Power cost Hardware cost Not zero bias
5	von Neumann(VN)	Zero bias Simple structure Low power	Maximum *ExE* = 25% Need *i.i.d.* input
5.1	Iterated VN(IVN)	Higher *ExE* Regular structure	Dynamic power cost
5.2	N-bit VN(VN_N)	Higher *ExE* Low dynamic power	Mapping table complexity *ExE* drop
Target	Improved VN_N	1. Waiting strategy: high *ExE* with smaller N 2. Hamming weight-based structure: small mapping table 3. Input-symbol code assignment: simple logic	

2.4 Motivation and Concept of This Research

In this research, my goal is to design a low energy and high robust TRNG. It consists of a high *ExE* and low energy post-processing block and high robust TRNG core block.

For the post-processing block, the pros and cons of prior arts are summarized in Table 2.4. Among these techniques, N-bit VN is a good candidate for low energy post-processing block. The reason is: first, it can potentially reduce the dynamic power by processing N bits at the same time; then, its *ExE* can be improved by choosing large N value. However, its mapping table increases exponentially (2^N). And there is a *ExE* drop between the theoretical and realistic value.

Based on the above considerations, an improved VN_N post-processing technique is introduced, as shown in Chapter 3. It is based on a conference paper in VLSI-DAT 2018 [51] and a journal article in IEICE 2021 [52]. I propose an improved N-bit von Neumann post-processing featuring high *ExE* and low energy. To solve the *ExE* drop and mapping table complexity problems, at the algorithm level, I propose a waiting strategy. It improves *ExE* by using smaller N values. At the architectural level, I propose a Hamming weight mapping-based hierarchical structure to reconstruct the large

mapping table using smaller tables. This hierarchical structure also enables the removal of the correlations in the raw bitstream. At the logic level, an input symbol-based code assignment is proposed for simple logic.

An 8-bit von Neumann with waiting (VN_8W) is designed and fabricated in 130-nm CMOS. It achieves 62.21% *ExE*, which is 2.49 times larger than the conventional von Neumann method. The de-bias and de-correlation abilities of VN_8W are verified by NIST SP 800-22 and NIST SP 800-90B tests. It achieves low energy of 0.18 pJ/bit at 0.45 V, 1 MHz, which is more than 20% smaller than the optimized IVN_7 with 59.38% *ExE* at same supply voltage. In summary, VN_8W is suitable for sub-pJ TRNG cores.

For the TRNG core block, the pros and cons of previous types are summarized in Table 2.4. Among these types, Latch-based TRNG has the merits of low power and simple logic. This is because of the latch's simple structure and the output bit generation process, which only needs one-time voltage transition. However, the mismatch problem degenerates the randomness of latch-based TRNG output. Complex calibration circuit and feedback control loop increases the design complexity, power and hardware cost of latch-based TRNG.

To overcome these problems, a calibration and feedback-control free latch based TRNG is introduced, as shown in Chapter 4. It is based on a conference paper in VLSI 2021 [49] and a journal article in JSSC 2022 [53]. I propose a latch-based TRNG core featuring mismatch self compensation and noise enhancement. The mismatch self-compensation is achieved by setting the initial state point close to the metastable operation point using newly added gate capacitance. For noise enhancement, damped oscillation using large resistor is applied for the first time.

Finally, the TRNG core block and VN_N post-processing block (VN_8W) are combined to build a whole TRNG, as shown in Figure 2.19. The prototype TRNG is fabricated in 130-nm CMOS. It operates across a wide voltage (0.3–1.0 V) and temperature (–20°C–100°C) range. It has resilience against power noise injection attacks. An accelerating aging test demonstrated the equivalent 11-year life reliability of the TRNG.

TABLE 2.5: Comparison of previous TRNG core and target in this dissertation

		Pros	Cons
1	Direct noise amplification	High noise magnitude	Area cost Power cost Systematic bias
2	Ring oscillator	Simple structure Commercial products: (Apple TRNG, Arm TRNG)	Power cost Power noise injection attacks
3	Chaotic map ADC-based	Low power by reuse ADC	Complex structure
4	Latch-based	Simple logic Low power	Mismatch problem Calibration circuit
Target	Improved Latch-based	1. Mismatch self-compensation: reduce mismatch 2. Noise enhancement: improve the noise	

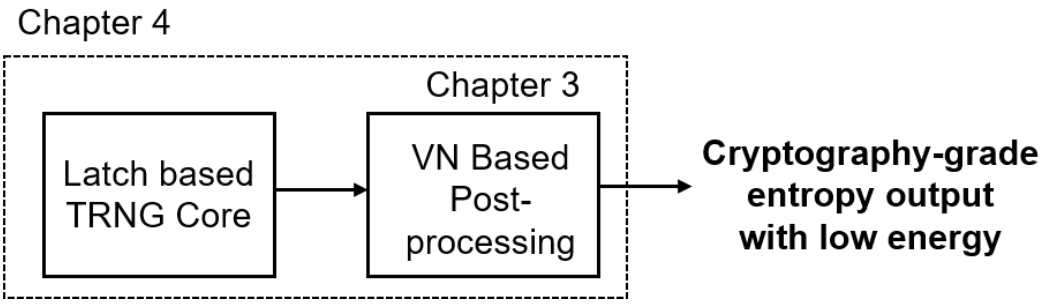


FIGURE 2.19: Concept of this research.

NIST SP 800-22 and NIST SP 800-90B tests verified the cryptography-grade randomness of the TRNG. It achieves the state-of-the-art minimum energy of 0.186 pJ/bit at 0.3 V, suitable for energy-constrained IoT devices.

Chapter 3

VN_N based Post-Processing Technique Having High Extraction Efficiency and Low Energy

3.1 Introduction

In this chapter, an improved N-bit von Neumann-based post-processing technique is presented. In order to solve the extraction efficiency (*ExE*) drop problem, a waiting strategy is proposed. By assigning waiting flags and generating new otput after gathering two waiting flags, it improves the *ExE* with much smaller N values. For example, 4 bit von Neumann with waiting (VN_4W, where W indicates waiting) achieves 46.88% *ExE*, which is 1.125 times larger than the conventional 6 bit von Neumann (VN_6). In addition, to tackle the mapping table complexity problem, a Hamming weight-based hierarchical structure is proposed. It reconstructs the large table with smaller tables based on Hamming weight. Furthermore, an input-symbol-based code assignment is proposed for simple logic. In order to confirm these techniques, an 8 bit von Neumann with waiting (VN_8W, 62.21% *ExE*) is designed and fabricated in 130-nm CMOS. It achieves low energy of 0.18 pJ/bit at 0.45 V, 1 MHz, which is suitable for low energy TRNG core design.

The remainder of this Chapter is organized as follows: First, the design and hardware implementation of an simple 4-bit von Neumann post-processing using input-symbol-based code assignment is introduced. Second, the theory analysis and implementation flow chart of waiting strategy are presented. Third, a hierarchical VN_8W with efficient hardware implementation solutions is introduced. Using the Hamming weight-based structure, the conventional 2^8 complexity mapping table is rebuilt with two identical 4 Bits Logic (2^4 complexity) and an 8 Bits Logic (5^2 complexity). The input-symbol-based code assignment is applied in mapping tables for reducing the hardware cost. Fourth, the experiment results are presented. The extraction efficiency is verified by real bitstream. The de-bias and de-correlation abilities of VN_8W are verified by NIST SP 800-22 and NIST SP 800-90B IID tests. Compared with IVN_7, VN_8W achieves more than 20% energy reduction at same supply voltage. Finally, a conclusion is drawn.

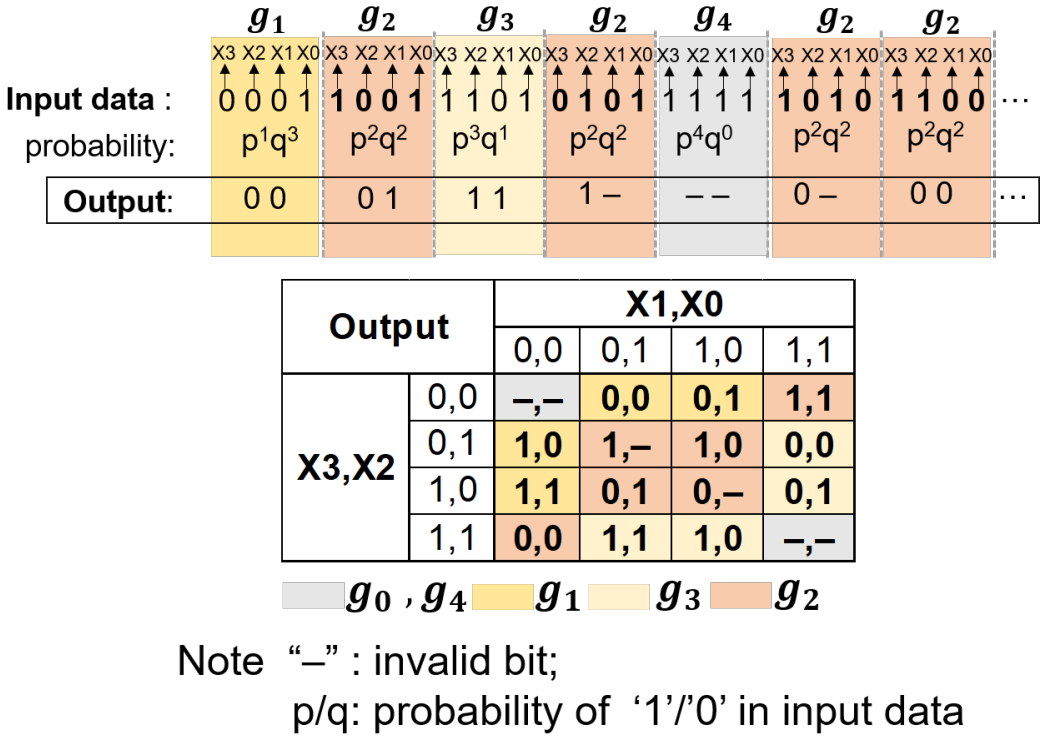


FIGURE 3.1: Code assignment in VN_4.

3.2 4-bit von Neumann with Input-Symbol-based Code Assignment

As mentioned in Chapter 2, VN_N post-processing has a mapping table complexity problem (2^N), an appropriate N value should be selected under hardware constraints. As shown in Figure 2.12, the *ExE* gap is smallest at N =4. Therefore, in order to design a simple but efficient post-processing, 4 bits von Neumann (VN_4) is implemented.

Figure 3.1 summarized the code assignment of VN_4. For consecutive 4 bits input (X3,X2,X1,X0), it has 2^4 combinations. According to the total number of ones count or Hamming weight, these combinations are divided into 5 groups: g_0 to g_4. Each group has $\#g_k (= C_4^k)$ members, $0 \leq k \leq 4$. These members occur with identical probability of $\frac{1}{\#g_k}$. Thus, each member can be assigned with different $\log_2(\#g_k)$ bits for achieving balanced ones and zeros output. As depicted in Figure 3.1, members of the same

DOUT		X1,X0			
		0,0	0,1	1,0	1,1
X3,X2	0,0	0,0	0,1	1,0	0,0
	0,1	1,1	0,1	1,0	1,1
	1,0	0,0	0,1	1,0	0,0
	1,1	1,1	0,1	1,0	1,1

Note: blue-colored symbol is invalid

(a)

DVALID		X1,X0			
		0,0	0,1	1,0	1,1
X3,X2	0,0	0,0	1,1	1,1	1,1
	0,1	1,1	1,0	1,1	1,1
	1,0	1,1	1,1	1,0	1,1
	1,1	1,1	1,1	1,1	0,0

(b)

FIGURE 3.2: Mapping tables of VN_4. (a) DOUT mapping table. (b) DVALID mapping table.

DOUT		X1,X0			
		0,0	0,1	1,0	1,1
X3,X2	0,0	X2,X2	X1,X0	X1,X0	X2,X2
	0,1	X2,X2	X1,X0	X1,X0	X2,X2
	1,0	X2,X2	X1,X0	X1,X0	X2,X2
	1,1	X2,X2	X1,X0	X1,X0	X2,X2

DOUT = (X1 == X0) ? {X2,X2} : {X1,X0}

FIGURE 3.3: Input-symbol-based code assignment for DOUT of VN_4.

group are marked with the same color. For example, group g_1 has $C_4^1 = 4$ input patterns: "0001","0010","0100","1000". The four patterns occur with same probability, therefore, $\log_2(C_4^1) = 2$ bits with four different patterns ("00","01","10","11") can be assigned to each member. As for group g_2, since $\log_2(C_4^2) = 2.58$ is not an integer, in realistic, $\#g_2$ is decomposed into $2^2 + 2^1$. As a result, 2 bits with four different patterns are assigned to four members and 1 bit with two different patterns are assigned to two members.

Note that, the code assignment for members of identical group is not fixed. For example,

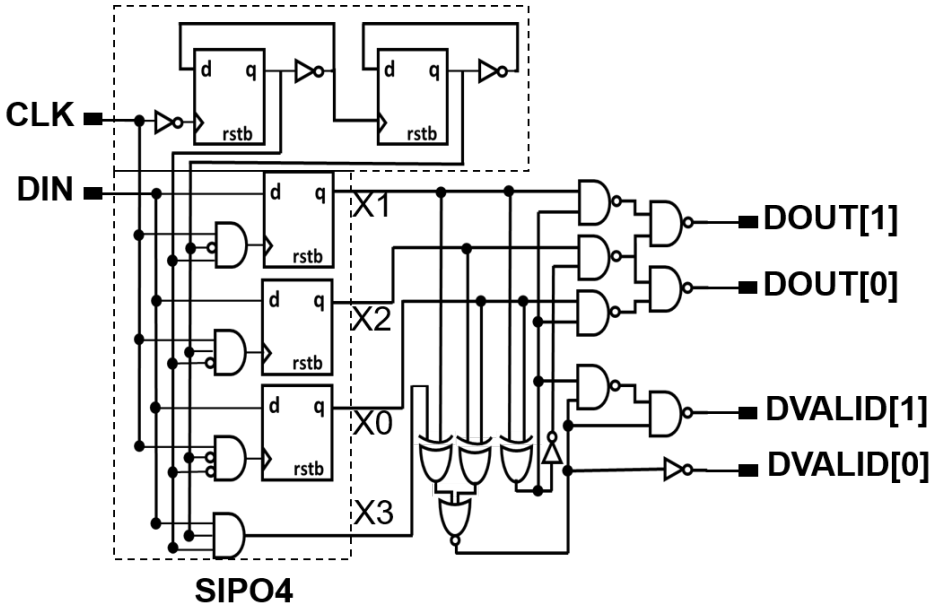


FIGURE 3.4: Logic structure of VN_4.

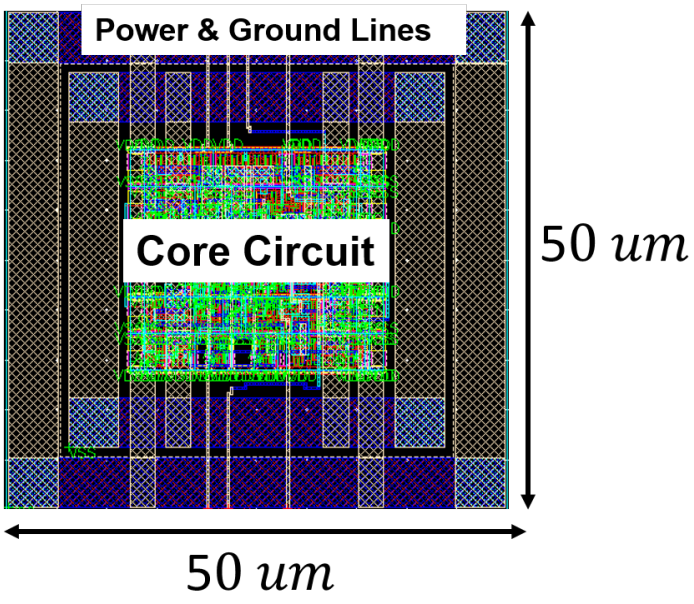


FIGURE 3.5: Layout image of VN_4.

in g_1, "00" has 4 choices, "01" has 3 choice, "10" has 2 choice, "11" has 1 choice. And totally, there are 24 different code assignments. Target for an efficient hardware implementation, the code assignments in Figure 3.1 are rearranged. And the mapping table consists of two tables: direct output bit table DOUT and valid mapping table DVALID, as shown in Figure 3.2(a) and (b). DVALID indicates if the value of DOUT is valid

("1") or discarded ("0"). With the help of DVALID mapping table, the 16 code assignment cases in DOUT can be summarized with only two patterns as distinguished by different colors. The two patterns are related to the input symbols, therefore, I propose to use the input-symbol-based code assignment for simple logic, as shown in Figure 3.3. By using Verilog HDL assignment, DOUT can be simply expressed as: DOUT = (X1 == X0) ? {X2,X2} : {X1,X0}. The other advantages of this proposal will be shown in Chapter 3.4.

Figure 3.4 shows the logic structure of VN_4. The input signals are CLK (for clock), DIN (for raw bits input), and RSTB (for resetting the registers when RSTB = 0). The output signals are DOUT[1:0], DVALID[1:0]. An serial-in parallel-out logic SIPO4 is used to read the raw input bits and gather 4-bits before processing by the mapping logic. The mapping table logic is implemented using combinational logic gates for low hardware costs.

For the functional verification, VN_4 is designed through ASIC design flow and fabricated into 130-nm CMOS. Figure 3.5 shows the layout image of VN_4. Including the power and ground line, VN_4 occupies a total area of 2500 μm^2. The core circuit occupies an area of 575 μm^2, which is 66 equivalent gates (GEs).

3.3 Waiting Strategy

To improve the *ExE* drop and achieve higher *ExE* with smaller N value in VN_N, a waiting strategy is proposed in this work [51], [52]. The general implementation flow chart is summarized in Figure. 3.6. For consecutive N bits, there are totally N + 1 groups: g_k, $0 \leq k \leq N$. Each group has $\#g_k$ ($=C_N^k$) members. If $\log_2(\#g_k)$ is an integer, assign $\log_2(\#g_k)$ bits for directly output. If not, $\#g_k$ is factorized as follows:

$$\#g_k = 2^m \times b, \tag{3.1}$$

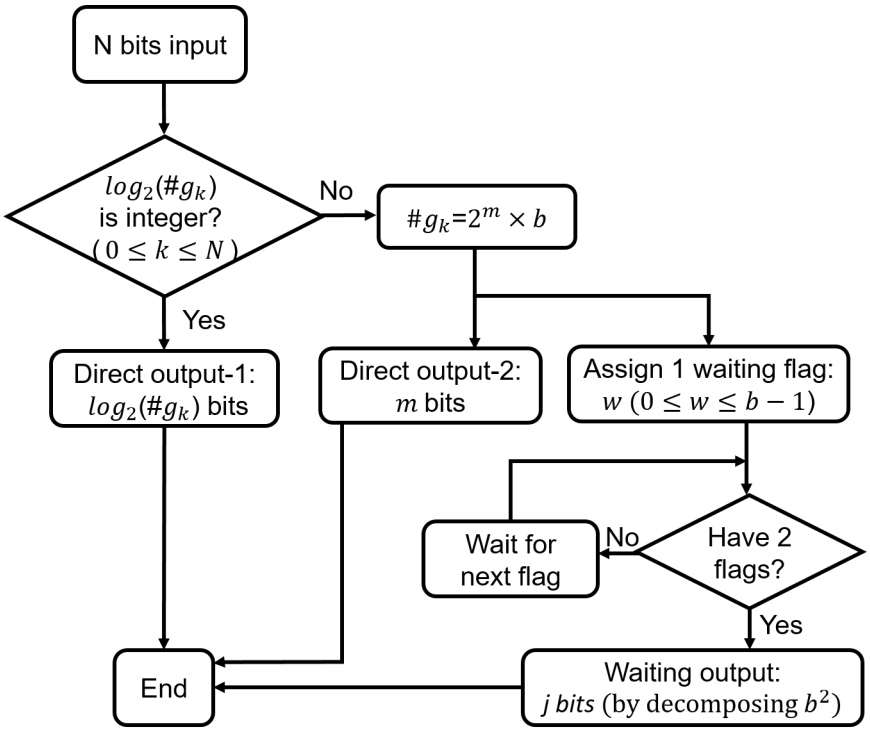


FIGURE 3.6: Waiting strategy flowchart.

$$b^2 = \sum_{j=0}^{n} A_{kj} 2^j, \tag{3.2}$$

where m is the directly output bits number, b is the base of waiting flag (0 to $b-1$). The waiting output bits are obtained by decomposing b^2 when two waiting flags are gathered, where $A_{kj} \in \{0,1\}(0 \le j < n)$, and $A_{kn} = 1$.

For example, Figure 3.7 compares the code assignment in VN_4 and VN_4 with waiting (VN_4W). The only difference occurs in group g_2. In VN_4 code assignment, $\#g_2$ is decomposed into $2^2 + 2^1$. As a result, four members are assigned 2 bits and two members are assigned 1 bit. In VN_4W, by waiting strategy, $\#g_2$ is factorized as 2×3. One direct output bit ("0" or "1") and a waiting flag ("*0*" or "*1*" or "*2*") are assigned to each member. When two waiting flags are gathered, 3^2 is decomposed into 2^3+2^0, 3 bits or 0 bit are generated. For all 16 input combinations, the mapping table is summarized in Figure 3.8. As a result, the average assigned bits count is 2.33 $[1 + (3 \times 8)/(9 \times 2)]$, which is close to the theoretical value of 2.58 (= $\log_2(6)$) and is 1.4 times larger than the realistic assignment in VN_4.

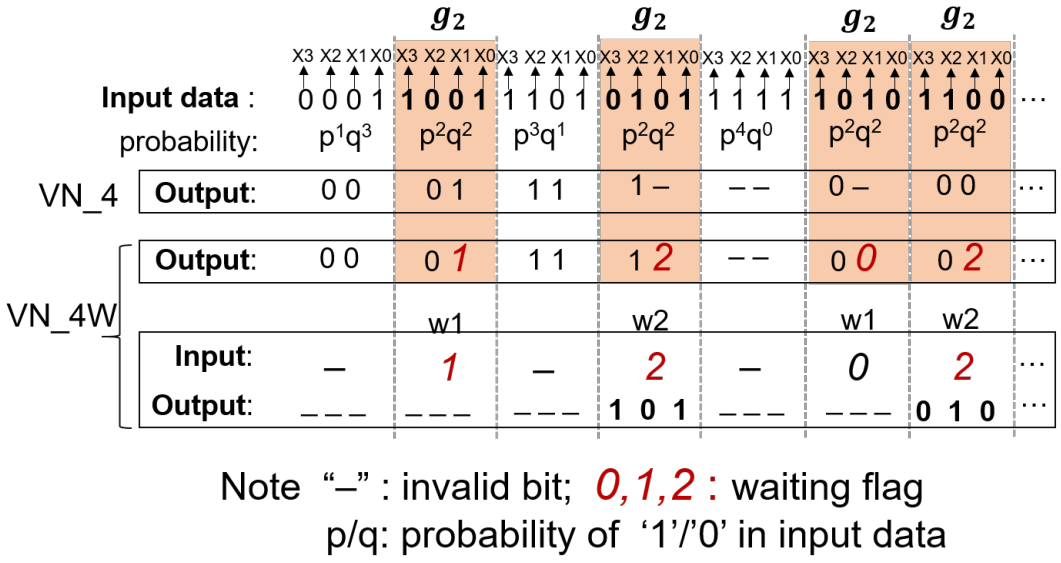


FIGURE 3.7: Code assignment of VN_4 and VN_4W.

For any different N values, the *ExE* by waiting strategy is shown in the following equation:

$$ExE\_\text{With waiting} = \frac{\sum_{k=0}^{N} p^k q^{N-k} C_N^k (m + \frac{\sum_{j=0}^{n} A_{kj} 2^j j}{2b^2})}{N}. \tag{3.3}$$

The red line in Figure. 3.9 indicates the *ExE* with waiting strategy. Comparing with the realistic value (denoted by the blue line), waiting strategy improves the *ExE* more close to the theoretical value (denoted by black line). This means higher *ExE* with smaller N value is achieved by waiting strategy. For example, VN_4W achieves 46.88% *ExE*, which is 1.154 times and 1.125 times larger than the conventional VN_4 (40.63% *ExE*) and VN_6 (41.67% *ExE*), respectively. VN_8W achieves 62.21% *ExE*, which approaches VN_12 with 64.63% *ExE*. Also, same with VN_N, as N values increases, the *ExE* becomes saturated. An appropriate N value should be selected for energy-efficient hardware implementation.

Output		X1,X0			
		0,0	0,1	1,0	1,1
X3,X2	0,0	–,–	0,0	0,1	1,*0*
	0,1	1,0	1,*2*	1,*1*	0,0
	1,0	1,1	0,*1*	0,*0*	0,1
	1,1	0,*2*	1,1	1,0	–,–

g_0 , g_4 g_1 g_3 g_2

Note: '–' is invalid; the **black-colored** digit is valid output bit; the **red-colored** digit is waiting flag bit;

(a)

Output		w2		
		0	*1*	*2*
w1	*0*	0,0,0	0,0,1	0,1,0
	1	0,1,1	1,0,0	1,0,1
	2	1,1,0	1,1,1	–,–,–

$3^2 = 2^3 + 2^0$

(b)

FIGURE 3.8: VN_4W bit assignments: (a) directly output mapping table. (b) waiting mapping table.

3.4 Hierarchical 8-Bit von Neumann with Waiting Strategy

As mentioned above, the *ExE* becomes saturated as increasing the N values. Furthermore, the hardware implementation complexity grows exponentially (2^N). Thus, for a target of more than 50% *ExE*, 8-bit von Neumann with waiting strategy (VN_8W) with efficient hardware implementation solutions is presented in this work.

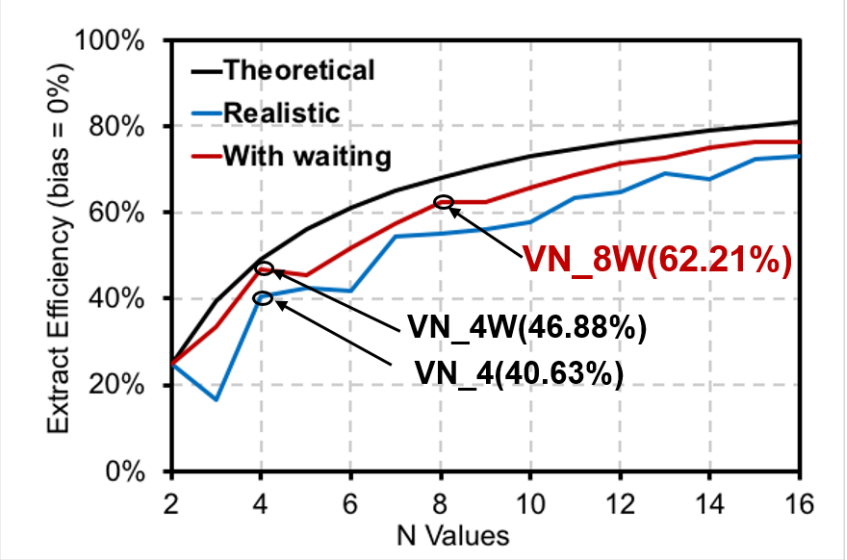


FIGURE 3.9: Extraction efficiency versus N values.

3.4.1 VN_8W Bit Assignment Strategy

The overall bit assignment of VN_8W is summarized in Table 3.1. All members in 8-bit are divided into nine groups: g_0 to g_8. Members in g_1 and g_7 are assigned $\log_2(C_8^1)$=$\log_2(C_8^7)$=3 direct output bits; Members in g_2 (g_6) and g_3 (g_5) are assigned 2 or 3 directly output bits with a waiting flag (base-7). For bit assignment in g_4, there are four ways: a) assign 6 , 2 , or 1 direct output bits; b) assign one direct output bit and one waiting flag (base-35); c) assign one direct output bit and two waiting flags (5-base and 7-base); d) assign three or one direct output bits and one waiting flag (base-7). The average assigned bits count are 5.63, 5.71, 5.05, and 4.89, respectively. Method b) yields the largest *ExE*. However, 35-base waiting flag requires a large hardware area. Method c) and d) have smaller *ExE*. Method a) has comparable high *ExE* without waiting logic. Thus, it is adopted in this design. As a result, the maximum *ExE* of VN_8W is 62.21%.

3.4.2 Hierarchical Structure

To overcome the 2^N hardware complexity, I propose a hierarchical structure, as shown in Figure 3.10. It consists of two 4 Bits Logic, one 8 Bits Logic, and one Waiting Logic

TABLE 3.1: VN_8W bit assignments strategy

Group	Total members	Direct output	Waiting flag*
g_0, g_8	$1 = 2^0$	0 bit	no
g_1, g_7	$8 = 2^3$	3 bits	no
g_2, g_6	$28 = 2^2 \times 7$	2 bits	yes: base-7
g_3, g_5	$56 = 2^3 \times 7$	3 bits	yes: base-7
g_4	a) $70 = 2^6 + 2^2 + 2^1$; Adopted	6, 2, 1 bits	no
	b) $70 = 2^1 \times 35$; Rejected	1 bit	yes: base-35
	c) $70 = 2^1 \times 5 \times 7$; Rejected	1 bit	yes: base-5, base-7
	d) $70 = (2^3 + 2^1) \times 7$; Rejected	3, 1 bits	yes: base-7

* Two waiting flags construct a waiting logic: $7^2 = 49 = 2^5 + 2^4 + 2^0$.

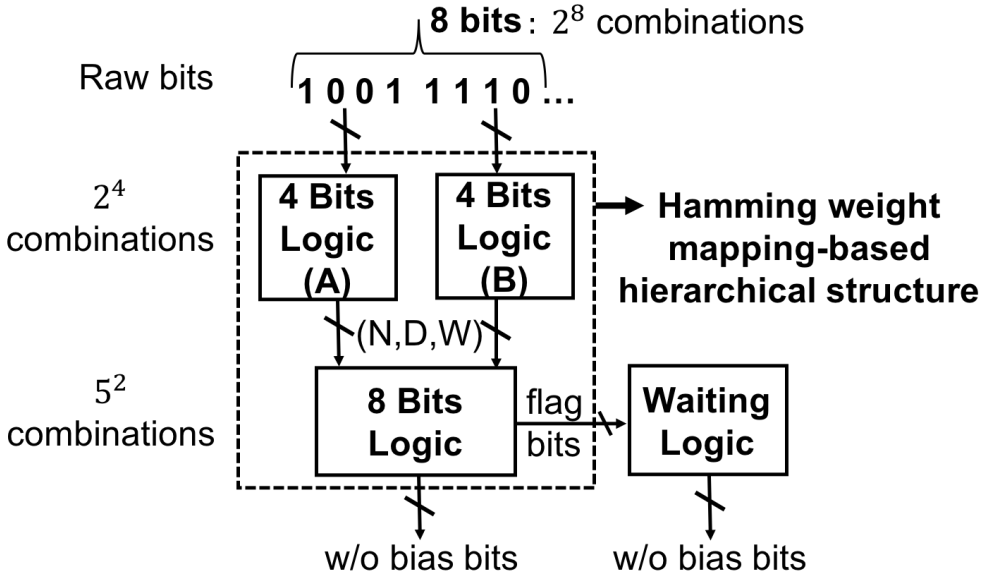


FIGURE 3.10: Hierarchical 8-bit von Neumann with waiting strategy.

block. The 8-bit inputs are divided into two 4-bit and sent to the 4 Bits Logic A and B, respectively. The function of 4 Bits Logic is an extended version of VN_4W. The generated intermediate variables N (denoted for Hamming weight of 4-bit), D (directly output bits of VN_4W), and W (waiting flags of VN_4W) are fed into 8 Bits Logic block. The 8 Bits Logic is constructed based on the Hamming weight NA and NB generated from the two 4 Bits Logic. Therefore, the table size of 8 Bits Logic is 5^2 due to NA (NB) ranging from 0 to 4. As a result, the large mapping table with 2^8 complexity is reconstructed into two 2^4-size smaller tables and one 5^2 table. Design details for each block are described below.

(a)

D		X1,X0			
		0,0	0,1	1,0	1,1
X3,X2	0,0	–,–	0,1	1,0	–,0
	0,1	1,1	–,1	–,0	1,1
	1,0	0,0	–,1	–,0	0,0
	1,1	–,1	0,1	1,0	–,–

g_0 , g_4 g_1 g_3 g_2

Note: '–' is invalid, it can be replaced by the blue-colored symbol

(b)

D		X1,X0			
		0,0	0,1	1,0	1,1
X3,X2	0,0	X2,X2	X1,X0	X1,X0	X2,X2
	0,1	X2,X2	X1,X0	X1,X0	X2,X2
	1,0	X2,X2	X1,X0	X1,X0	X2,X2
	1,1	X2,X2	X1,X0	X1,X0	X2,X2

D = (X1 == X0) ? {X2,X2} : {X1,X0}

FIGURE 3.11: D mapping table in the 4 Bits Logic: (a) actual binary codes; (b) input-symbol-based codes.

3.4.2.1 4 Bits Logic

The 4 Bits Logic processes 4-bit input and generates Hamming weight N, output bits D, and waiting flags W. A bit adder is used to generate N. D and W are generated according to the mapping tables. Instead of using the real code assignments, as shown in Figures 3.11(a) and 3.12(a), I propose to use the input-symbol-based code assignment, as shown in the Figures 3.11(b) and 3.12(b). It simplifies the assigned code cases and thereby reduces the hardware cost. For example, 16 code assignment cases in Figures. 3.11(a) are reduced two cases, as shown in Figure 3.11(b). Different code assignments are distinguished by colors. The blue colored digits are invalid. As a result, the code assignment can be easily described by the Verilog HDL codes:

D = (X1 == X0) ? {X2,X2} : {X1,X0};

W = (X3 == X2) ? {0,0} : {X3,X2};

(a)

W		X1,X0			
		0,0	0,1	1,0	1,1
X3,X2	0,0	–,–	–,–	–,–	0,0
	0,1	–,–	0,1	0,1	–,–
	1,0	–,–	1,0	1,0	–,–
	1,1	0,0	–,–	–,–	–,–

g_0 , g_4 g_1 g_3 g_2

Note: '–' is invalid, it can be replaced by the blue-colored symbol

(b)

W		X1,X0			
		0,0	0,1	1,0	1,1
X3,X2	0,0	0,0	0,0	0,0	0,0
	0,1	X3,X2	X3,X2	X3,X2	X3,X2
	1,0	X3,X2	X3,X2	X3,X2	X3,X2
	1,1	0,0	0,0	0,0	0,0

W = (X3 == X2) ? {0,0} : {X3,X2}

FIGURE 3.12: W mapping table in the 4 Bits Logic: (a) actual binary codes; (b) input-symbol-based codes.

Bit Assignments		NA (NA2,NA1,NA0)					
		0,0,0 <1>	0,0,1 <4>	0,1,0 <6>	0,1,1 <4>	1,0,0 <1>	
NB (NB2, NB1, NB0)	0,0,0 <1>	0 bit <1>	3 bits <4>	2 bits + w <6>	3 bits + w <4>	1 bit <1>	g_4
	0,0,1 <4>	3 bits <4>	2 bits + w <16>	3 bits + w <24>	6 bits <16>	3 bits + w <4>	g_5
	0,1,0 <6>	2 bits +w <6>	3 bits + w <24>	6 or 2 bits <32+4>	3 bits + w <24>	2 bits + w <6>	g_6
	0,1,1 <4>	3 bits + w <4>	6 bits <16>	3 bits + w <24>	2 bits + w <16>	3 bits <4>	g_7
	1,0,0 <1>	1 bit <1>	3 bits + w <4>	2 bits + w <6>	3 bits <4>	0 bit <1>	g_8

Note: +w means assigning the waiting flag.

$\#g_0 = \#g_8 = C_8^0 = 1$

$\#g_1 = \#g_7 = C_8^1 = 8 = 4 + 4$

$\#g_2 = \#g_6 = C_8^2 = 28 = 6 + 16 + 6$

$\#g_3 = \#g_5 = C_8^3 = 56 = 4 + 24 + 24 + 4$

$\#g_4 = C_8^4 = 70 = 1 + 16 + 36 + 16 + 1$

FIGURE 3.13: Bit assignments strategy in the 8 Bits Logic.

DOUT		NA (NA2,NA1,NA0)				
		0,0,0 <1>	0,0,1 <4>	0,1,0 <6>	0,1,1 <4>	1,0,0 <1>
NB (NB2, NB1, NB0)	0,0,0 <1>	NB0,DA1,DA0, DB1,DB0,NB1	NB0,DA1,DA0, DB1,DB0,NB1	NB0,DA1,DA0, DB1,DB0,NB1	NB0,DA1,DA0, DB1,DB0,NB1	NB0,DA1,DA0, DB1,DB0,NB2
	0,0,1 <4>	NB0,DA1,DA0, DB1,DB0,NB1	NB0,DA1,DA0, DB1,DB0,NB1	NB0,DA1,DA0, DB1,DB0,NB1	NB0,DA1,DA0, DB1,DB0,NB1	NB0,DA1,DA0, DB1,DB0,NB1
	0,1,0 <6>	NB0,DA1,DA0, DB1,DB0,NB1	NB0,DA1,DA0, DB1,DB0,NB1		NB0,DA1,DA0, DB1,DB0,NB1	NB0,DA1,DA0, DB1,DB0,NB1
	0,1,1 <4>	NB0,DA1,DA0, DB1,DB0,NB1	NB0,DA1,DA0, DB1,DB0,NB1	NB0,DA1,DA0, DB1,DB0,NB1	NB0,DA1,DA0, DB1,DB0,NB1	NB0,DA1,DA0, DB1,DB0,NB1
	1,0,0 <1>	NB0,DA1,DA0, DB1,DB0,NB2	NB0,DA1,DA0, DB1,DB0,NB1	NB0,DA1,DA0, DB1,DB0,NB1	NB0,DA1,DA0, DB1,DB0,NB1	NB0,DA1,DA0, DB1,DB0,NB1

Note: blue-colored bit is invalid.

		WA (WA1,WA0)		
		0,0	0,1	1,0
WB (WB1,WB0)	0,0	NB0,WA0,DA0 WB0,DB0,NB1	NB0,WA0,DA0 WB0,DB0,NB1	NB0,WA0,DA0 WB0,DB0,NB0
	0,1	NB0,WA0,DA0 WB0,DB0,NB1	NB0,WA0,DA0 WB0,DB0,NB1	NB0,WA0,DA0 WB0,DB0,NB0
	1,0	NB0,WB1,DA0 WA0,DB0,NB0	NB0,WB1,DA0 WA0,DB0,NB0	NB0,WA0,DA0 WB0,DB0,NB0

FIGURE 3.14: DOUT mapping table in the 8 Bits Logic.

3.4.2.2 8 Bits Logic

Figure 3.13 summarizes the bit assignment strategy in 8 Bits Logic. The input variables are Hamming weight NA[2:0] and NB[2:0] generated by the 4 Bits Logic A and B, respectively. Each input cell has two values: the upper one denotes the binary form of the Hamming weight, the lower one denotes the number of group members. For example, NA = "0,0,1" denotes for g_1 group, and $\#g_1$ is 4 and thereby denoted by $< 4 >$. In the output cells, the upper line shows the directly output bit count and the waiting flag assignments, which denoted by '+ w'. Members in same group are colored the same. This table covers all 2^8 cases shown in Table 3.1 with only 5^2-size. The intermediate variables DA or DB, WA or WB, and NA or NB are used to built the input-symbol-based bits assignment table. Figure 3.14 shows the direct output table, DOUT. The related valid bits mapping table, DVALID is shown in Figure 3.15. Waiting Flag code assignment table, DWAIT is presented in Figure 3.16. Each symbol may represent "0" or "1" with same probability, according to the Hamming weight. For example, NA =

DVALID		NA				
		0,0,0 <1>	0,0,1 <4>	0,1,0 <6>	0,1,1 <4>	1,0,0 <1>
NB (NB2, NB1, NB0)	0,0,0 <1>	**0,0,0, 0,0,0**	**1,1,1, 0,0,0**	**0,0,1, 0,0,1**	**1,1,1, 0,0,0**	**0,0,0, 0,0,1**
	0,0,1 <4>	**1,0,0, 1,1,0**	**0,1,1, 0,0,0**	**1,0,0, 1,1,0**	**1,1,1, 1,1,1**	**1,0,0, 1,1,0**
	0,1,0 <6>	**0,0,0, 0,1,1**	**1,1,1, 0,0,0**		**1,1,1, 0,0,0**	**0,0,0, 0,1,1**
	0,1,1 <4>	**1,0,0, 1,1,0**	**1,1,1, 1,1,1**	**1,0,0, 1,1,0**	**0,1,1, 0,0,0**	**1,0,0, 1,1,0**
	1,0,0 <1>	**0,0,0, 0,0,1**	**1,1,1, 0,0,0**	**0,0,1, 0,0,1**	**1,1,1, 0,0,0**	**0,0,0, 0,0,0**

		WA		
		0,0	0,1	1,0
WB (WB1,WB0)	0,0	**1,1,1, 1,1,1**	**1,1,1, 1,1,1**	**1,1,1, 1,1,1**
	0,1	**1,1,1, 1,1,1**	**1,1,1, 1,1,1**	**1,1,1, 1,1,1**
	1,0	**1,1,1, 1,1,1**	**1,1,1, 1,1,1**	**0,0,1, 0,1,0**

FIGURE 3.15: DVALID mapping table in the 8 Bits Logic.

DWAIT		NA (NA2,NA1,NA0)				
		0,0,0 <1>	0,0,1 <4>	0,1,0 <6>	0,1,1 <4>	1,0,0 <1>
NB (NB2, NB1, NB0)	0,0,0 <1>	**0,DA1,DA0**	**0,DA1,DA0**	**1,WA1,WA0**	**0,DA1,DA0**	**0,DA1,DA0**
	0,0,1 <4>	**0,DB1,DB0**	**0,DB1,DB0**	**0,DB1,DB0**	**0,DB1,DB0**	**0,DB1,DB0**
	0,1,0 <6>	**1,WB1,WB0**	**1,WB1,WB0**	**1,WB1,WB0**	**1,WB1,WB0**	**1,WB1,WB0**
	0,1,1 <4>	**0,DB1,DB0**	**0,DB1,DB0**	**0,DB1,DB0**	**0,DB1,DB0**	**0,DB1,DB0**
	1,0,0 <1>	**0,DA1,DA0**	**0,DA1,DA0**	**1,WA1,WA0**	**0,DA1,DA0**	**0,DA1,DA0**

Note: **blue-colored** bit is invalid.

FIGURE 3.16: DWAIT mapping table in the 8 Bits Logic.

"0,0,1" (which denotes g_1 in 4-bit), "DA1, DA0" indicates all four combinations in 2 bits. Different code assignments are distinguished by colors. As can be seen, DOUT table can be realized by only five patterns with the help of DVALID mapping table. Likewise, the waiting flag DWAIT can be described by only four patterns. In addition, the sum of NA and NB is used to judge the valid DWAIT. Only when the results equal to 2, 3 ,5 ,6 in decimal, the DWAIT is valid.

DOUT_WAIT		DWAIT_2 (W2,W1,W0)						
		0,0,0	0,0,1	0,1,0	0,1,1	1,0,0	1,0,1	1,1,0
DWAIT_1 (W5, W4, W3)	0,0,0	W1,W0,W5, W4,W3 <28>				W1,W0,W5, W4,W3 <14>		W4,W3,W2, W2,W2 <4>
	0,0,1							
	0,1,0							
	0,1,1							
	1,0,0							W4,W3,W2, W2,W2 <2>
	1,0,1							
	1,1,0							W4,W3,W2, W2,W2 <1>

$7^2 = 49 = 2^5 + 2^4 + 2^0 = (28 + 4) + (14 + 2) + 1$

Note: **blue-colored** bit is invalid.

FIGURE 3.17: DOUT_WAIT mapping table in the Waiting Logic.

DVALID_WAIT		DWAIT_2						
		0,0,0	0,0,1	0,1,0	0,1,1	1,0,0	1,0,1	1,1,0
DWAIT_1 (W5, W4, W3)	0,0,0	1,1,1, 1,1				0,1,1, 1,1		1,1,1, 1,1
	0,0,1							
	0,1,0							
	0,1,1							0,1,1, 1,1
	1,0,0							
	1,0,1							
	1,1,0							0,0,0, 0,0

FIGURE 3.18: DVALID_WAIT mapping table in the Waiting Logic.

3.4.2.3 Waiting Logic

To reduce the hardware cost, the Waiting Logic is shared for all base-7 waiting flags in group g_2, g_6, g_3, g_5. The waiting output code assignment table, DOUT_WAIT is shown in Figure 3.17. The related valid output mapping table, DVALID_WAIT, is shown in Figure 3.18. 7^2 is decomposed into $2^5 + 2^4 + 2^0$. Thus, 32 input cases are assigned 5 bits, 16 input cases are assigned 4 bits, and 1 input case is assigned 0 bit. In conjunction with the DVALID_WAIT mapping table, DOUT_WAIT can be described by only two patterns.

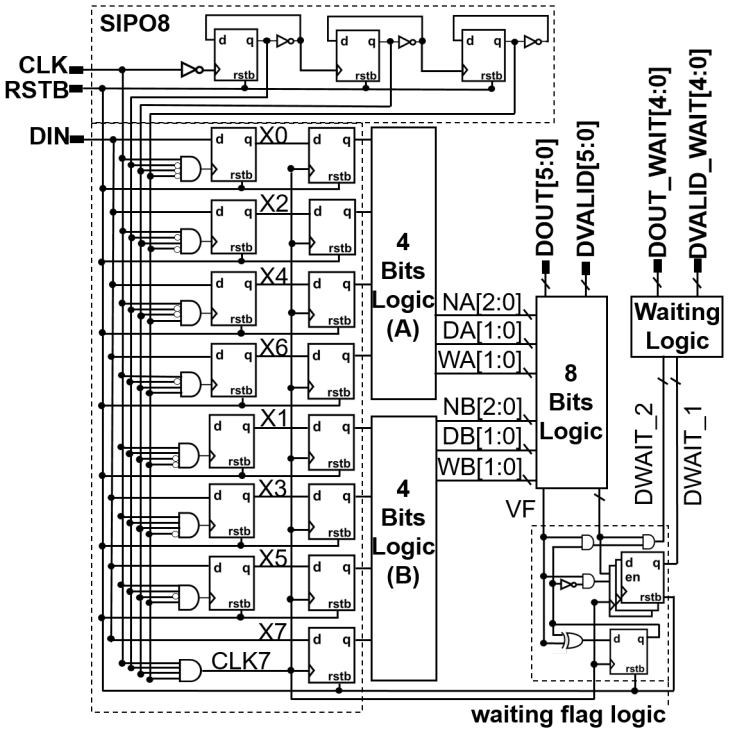


FIGURE 3.19: Schematic of VN_8W.

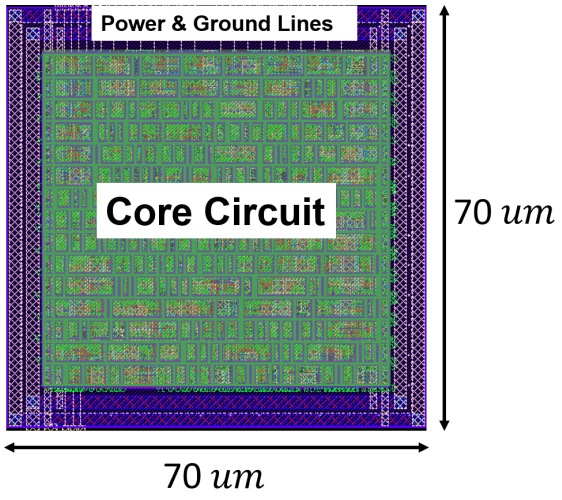


FIGURE 3.20: Layout image of VN_8W.

3.4.3 VN_8W Schematic

Figure 3.19 shows the schematic of VN_8W. The input signals are CLK (clock signal), RSTB (reset signal), DIN (raw input bit signal). The output signals include

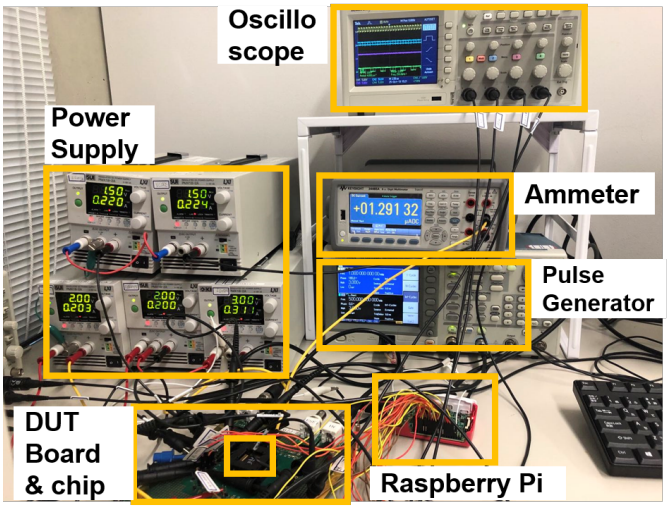


FIGURE 3.21: Measurement setup.

DOUT[5:0] (for direct output), DVALID[5:0], DOUT_WAIT[4:0] (for waiting output), and DVALID_WAIT[4:0]. Two identical 4 Bits Logic, 8 Bits Logic, and Waiting Logic form the mapping table logic of VN_8W. A waiting flag logic is used to store the valid waiting flag bits. In which, the waiting flag is valid only when VF = "1". The serial-in-parallel-out part SIPO8 is used to gather 8-bit input. It should be noted that the mapping logic is triggered every 8 clock cycles through gated clock CLK7. As a result, the dynamic power is reduced.

Furthermore, the hierarchical structure enables a scramble function of input bits without additional hardware overhead. In this design, the 8 bits input are separated into odd and even parts and fed into the 4 Bits Logic A and B, respectively. The most significant lag-1 correlation factor can be improved by this structure. Meanwhile, the bit correlation is further masked by the 8 Bits Logic. The decorrelation results will be discussed in Chapter 3.5.

3.5 Experiment Results

To verify the effectiveness, VN_8W is designed and fabricated in 130-nm CMOS. The layout image is shown in Figure 3.20. The total area including power and ground line is 4900 μm^2 . The area of core circuit is 2583 μm^2 (381 GEs). For comparison, VN_4 with

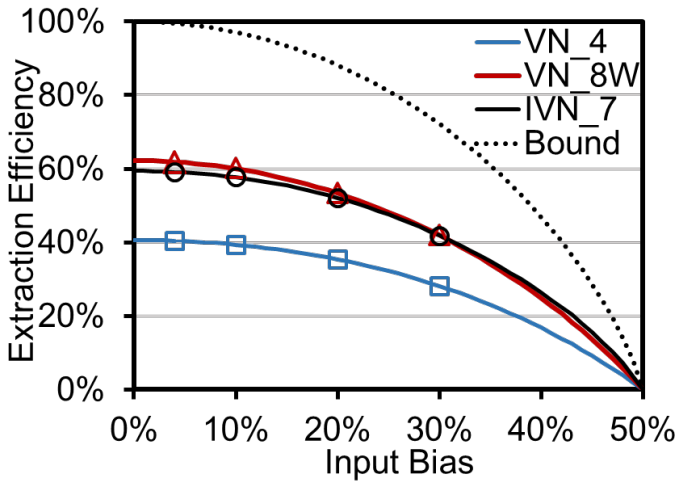


FIGURE 3.22: Extraction efficiency versus input bias(solid lines for the expected values; the hollow points for the measured values).

40.63% *ExE* and IVN_7 (Shown in Figure 2.10) with 59.38% *ExE* are also fabricated in 130-nm CMOS using identical automatic design rule as VN_8W. The core circuits of VN_4 and IVN_7 cost 66 GEs and 204 GEs, respectively.

Figure 3.21 shows the basic measurement setup. The silicon chip is put into a DUT (Device Under Test) board. Other equipment is connected with the DUT board: power supply, oscilloscope (for observing the waveform), ammeter (for current measurement), pulse generator (for generating clock signal), and Raspberry Pi (for reading data and generating control signal). The final results are stored in the memory of Raspberry Pi.

3.5.1 Extraction Efficiency and Randomness Check for Biased Only Data

One raw bitstream with 4% bias generated by the TRNG in [39] and three bitstreams with 10%, 20%, and 30% bias are used as the test data. Each bitstream has 3,000,000 bits. The *ExE* curves are summarized in Figure 3.22. It is indicated that the measured *ExE*s meet the theoretical values well. Meanwhile, each post-processed 1,000,000-bit

is tested by NIST SP 800-22 test suites with $P-value \geq 0.01$ as passed condition. The results are summarized in Table 3.2. VN_4 and VN_8W post-processed data passed all terms. Only IVN_7 failed at Runs test at bias = 4%. The related NIST SP 800-90B IID results are summarized in Table 3.3. Also, the 4% bias raw TRNG data after IVN_7 post-processing failed the chi square independence and IID permutation tests.

TABLE 3.2: NIST SP 800-22 test results for the data with bias after post-processing.

	VN_4		VN_8W		IVN_7	
	Ave. P-Value	Passed	Ave. P-Value	Passed	Ave. P-Value	Passed
Frequency	0.73	4/4	0.46	4/4	0.57	4/4
Block Frequency	0.49	4/4	0.28	4/4	0.21	4/4
Runs	0.64	4/4	0.58	4/4	0.59	3/4 [a]
Longest Runs	0.59	4/4	0.52	4/4	0.50	4/4
Rank	0.23	4/4	0.31	4/4	0.50	4/4
FFT	0.16	4/4	0.51	4/4	0.60	4/4
Non-Overlapping Template	0.50	4/4	0.51	4/4	0.50	4/4
Overlapping Template	0.60	4/4	0.51	4/4	0.47	4/4
Universal	0.45	4/4	0.45	4/4	0.68	4/4
Linear Complexity	0.43	4/4	0.72	4/4	0.42	4/4
Serial	0.58	4/4	0.50	4/4	0.58	4/4
Approximate Entropy	0.69	4/4	0.48	4/4	0.59	4/4
Cumulative Sums	0.64	4/4	0.49	4/4	0.61	4/4
Random Excursions	0.41	4/4	0.45	4/4	0.52	4/4
Random Excursions Variant	0.38	4/4	0.48	4/4	0.40	4/4

[a] Failed at raw data with 4% bias.

TABLE 3.3: NIST SP 800-90B IID test results for bias data after post-processing.

	VN_4	VN_8W	IVN_7
	Passed	Passed	Passed
Chi Square Independence	4/4	4/4	3/4[a]
Chi Square Goodness-of-fit	4/4	4/4	4/4
IID Permutation Tests	4/4	4/4	3/4[a]
Min-Entropy Estimate	Ave. = 0.996	Ave. = 0.995	Ave. = 0.995

[a] Failed at 4% raw data.

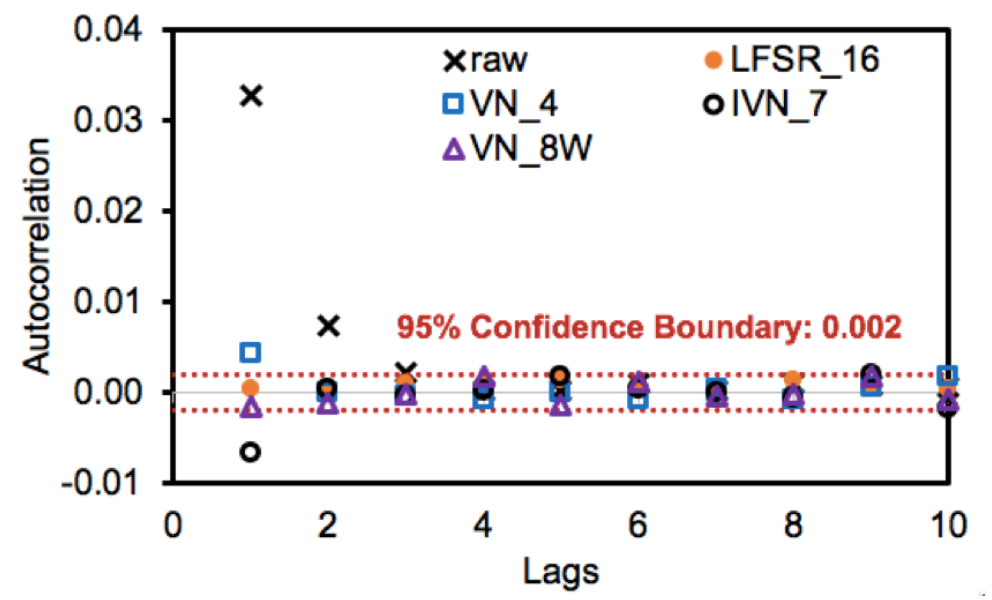


FIGURE 3.23: Autocorrelation of 1M bits with 100 lags.

3.5.2 Autocorrelation and Randomness Check for Correlated Data

To verify the post-processing techniques' decorrelation abilities, a 3,000,000 raw bits is generated according to ARIMA [54] model. Autocorrelation check results for each 1,000,000 length bitstream of raw data and post-processed data are summarized in Figure 3.23. The raw data has large lag-1 (0.033) and lag-2 (0.007) autocorrelation factors. After lag-3, the autocorrelation factors are within 95% confidence boundary. By using VN_4 and IVN_7, the lag-1 factors are reduced but remain above or below the confidence boundary. By using VN_8W and LFSR_16, both lag-1 and lag-2 factors are within the confidence boundary. NIST SP 800-22 and NIST SP 800-90B IID tests results are summarized in Tables 3.4 and 3.5, respectively. By using VN_8W post-processing, the data passed all test terms. It verified the decorrelation and de-bias ability of VN_8W.

TABLE 3.4: NIST SP 800-22 test results of correlated raw data and post-processed data

	Raw		VN_4		IVN_7		LFSR_16		VN_8W	
	P Val.	Pass?	P Val.	Pass?	P Val.	Pass?	P Val.	Pass?	P Val.	Pass?
Frequency	0.54	Yes	0.94	Yes	0.58	Yes	0.25	Yes	1.00	Yes
Block Frequency	0.00	**No**	0.11	Yes	0.55	Yes	0.33	Yes	0.29	Yes
Runs	0.00	**No**	0.00	**No**	0.00	**No**	0.64	Yes	0.13	Yes
Longest Runs	0.01	Yes	0.11	Yes	0.47	Yes	0.05	Yes	0.05	Yes
Binary Matrix Rank	0.23	Yes	0.55	Yes	0.31	Yes	0.06	Yes	0.79	Yes
FFT	0.51	Yes	0.80	Yes	0.99	Yes	0.59	Yes	0.96	Yes
Non-Overlapping Template	0.15	Yes	0.48	Yes	0.46	Yes	0.50	Yes	0.43	Yes
Overlapping Template	0.00	**No**	0.85	Yes	0.76	Yes	0.85	Yes	0.20	Yes
Universal Statistical	0.01	Yes	0.51	Yes	0.82	Yes	0.30	Yes	0.72	Yes
Linear Complexity	0.23	Yes	0.42	Yes	0.55	Yes	0.79	Yes	0.92	Yes
Serial	0.08	Yes	0.88	Yes	0.44	Yes	0.60	Yes	0.38	Yes
Approximate Entropy	0.00	**No**	0.54	Yes	0.95	Yes	0.36	Yes	0.73	Yes
Cumulative Sums	0.81	Yes	0.46	Yes	0.81	Yes	0.40	Yes	0.71	Yes
Random Excursions	0.34	Yes	0.59	Yes	0.52	Yes	0.00	**No**	0.66	Yes
Random Excursions Variant	0.73	Yes	0.51	Yes	0.21	Yes	0.00	**No**	0.70	Yes

TABLE 3.5: NIST SP 800-90B IID test results for correlated data after post-processing.

	Raw	VN_4	IVN_7	LFSR_16	VN_8W
Chi square independence	Failed	Pass	Passed	Passed	Passed
Chi square goodness of fit	Failed	Pass	Passed	Passed	Passed
IID permutation tests	Failed	Fail	Failed	Passed	Passed
Min-Entropy	0.995	0.996	0.995	0.995	0.996

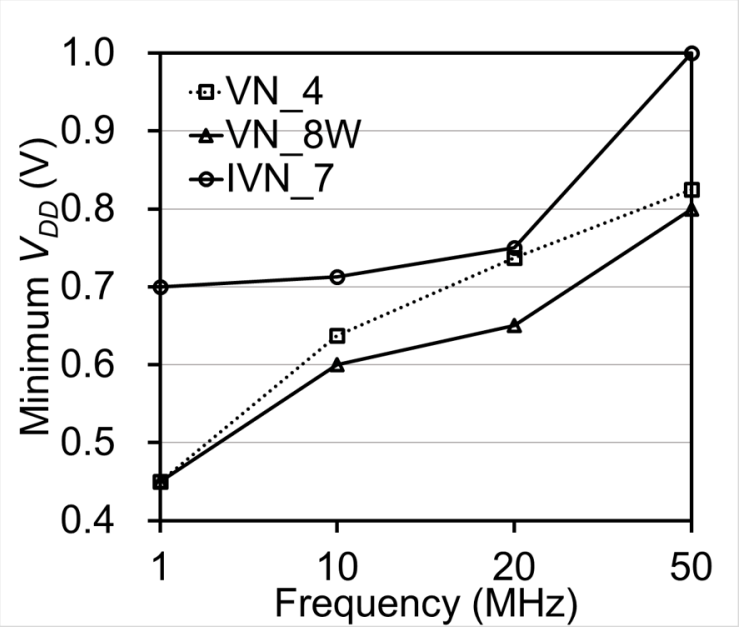


FIGURE 3.24: Minimum operating voltage versus maximum frequency.

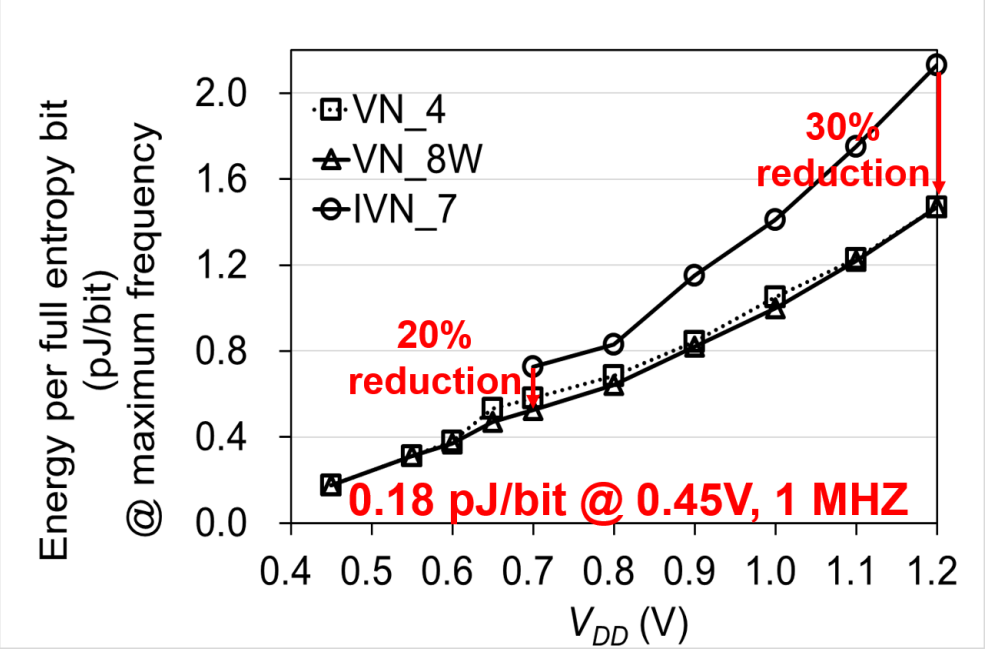


FIGURE 3.25: Energy consumption.

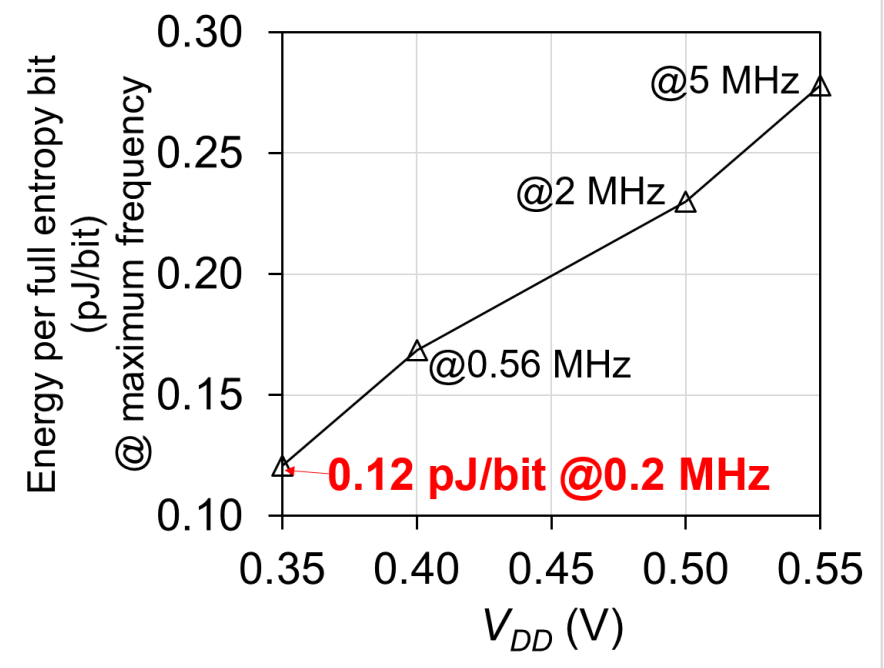


FIGURE 3.26: Low energy consumption measurement results of VN_8W.

3.5.3 Power and Energy Consumption

Figure 3.24 illustrates the average results of minimum supply voltage versus frequency for four chips under room temperature. It can be seen that, at 1 MHz, VN_4 and VN_8W operate at 0.45 V, while, IVN_7 needs 0.7 V. With the increase of the frequency, VN_8W always can operate at lower supply voltage, compared with IVN_7. It is benefited from the gated clock design. The mapping logic is activated every 8 clock cycles, thus, the delay issue is not significant in VN_8W.

The energy consumption of post-processing techniques (E_{VN}) are calculated as follows:

$$E_{\mathrm{VN}} = \frac{Power}{Frequency \times ExE}. \tag{3.4}$$

Figure 3.25 summarizes the energy consumption results. At 0.45 V, 1 MHz, VN_8W and VN_4 achieved low energy of 0.18 pJ/bit. VN_4's energy consumption is slightly larger than or approach VN_8W's energy consumption. In the voltage range of 0.7 to 1.2 V, VN_8W achieved 20% to 30% energy reduction when compared with IVN_7.

Considering a TRNG system, which consists of a TRNG core and a post-processing technique. VN_8W has the maximum *ExE* and the smallest energy consumption, thus, it is the best post-processing technique candidate among the three von Neumann-based methods.

Targeting for application in energy-constrained IoT devices even with low supply voltage, the low supply measurement results of VN_8W is shown in Figure 3.26. VN_8W achieved minimum energy of 0.12 pJ/bit at 0.35 V, 0.2 MHz.

3.5.4 Comparisons with Previous Works

The comparisons with previous works including iterated von Neumann and finite field theory-based &Block cipher are shown in Table 3.6. Among all methods, VN_4 costs the smallest GEs. VN_8W achieves the minimum energy consumption with moderate *ExE* when compared the IVN based methods in work [31], [23], and [12]. When compared with the finite field theory-based and block cipher-based methods in [19], [18], [20], VN8W also has the decorrelation ability. Meanwhile, VN_8W costs much smaller GEs than these cryptographic-based post-processing techniques. It indicated that VN_8W is suitable for the area and energy-constrained TRNGs in IoT devices.

TABLE 3.6: Comparison with prior post-processing techniques

	N-bit von Neumann		Iterated von Neumann			Finite Field Theory-based & Block Cipher		
	This work		[31] HOST 2016	[23] SSCL 2018	[12] JSSC 2019	[19] JSSC 2016	[18] TCAS-II 2018	[20] TCAS-I 2015
	VN_4	**VN_8W**	IVN_7	Markov + IVN_16 + LFSR	Hierarchical VN	Decorrelators + BIW	Strong Blenders	PRESENT
Process Technology	**130-nm CMOS**	**130-nm CMOS**	130-nm[a] CMOS	65-nm CMOS	ARRIA FPGA	14-nm CMOS	45-nm[b] NanGate	32-nm[b] CMOS PTM
Gate Equivalent (GE)	**66**	**381**	204[a]	N/A	750[c]	586	166.3 ~13K[d]	1171
Max. Extraction Efficiency	**40.63%**	**62.21%**	59.38%	≈ 78%[e]	43.75%	12.5%	4%~20%[d]	50%/80%[d]
De-Biasing	**Yes**	**Yes**	Yes	Yes	Yes	Yes	Yes	Yes
De-Correlation	**No**	**Yes**	No	Yes	No	Yes	Yes	Yes
Energy per full-entropy bit (pJ/bit)	**0.18 @0.45 V**	**0.18 @0.45 V**	0.73[a] @0.70 V	2.58[f] @0.53 V	N/A	9 @0.75 V	N/A	1.0~2.5[f] @0.9 V

[a] Implemented in this work. The previous work does not show this data. The GE count is based on the final layout circuit.
[b] Only for Synthesis/Simulation.
[c] Gate counts
[d] Including several versions.
[e] Read from Fig. 6(a) in [23].
[f] Including TRNG cores.

3.6 Conclusion

This Chapter presented an energy-efficient von Neumann-based post-processing technique, targeting for low energy consumption TRNG. The extraction efficiency drop and mapping table complexity (2^N) problems in conventional N-bit von Neumann method are solved in the following three levels.

At the algorithm level, a waiting strategy is proposed to improve the extraction efficiency to approach the Shannon entropy bound. Compared with the conventional N-bit von Neumann method, waiting strategy enables high extraction efficiency with smaller N values. For example, VN_4W achieves 46.88% *ExE*, which is much higher than VN_6 with 41.67% *ExE*. VN_8W achieves 62.21% *ExE*, which is close to 64.63% *ExE* of VN_12. In addition, small N values cost less hardware area in mapping table. Thereby, high *ExE* can be achieved with less hardware cost.

At the architectural level, a Hamming weight based hierarchical structure is proposed. The large mapping table with 2^N complexity is now reconstructed to smaller tables according to the Hamming weight. This hierarchical structure also enables the decorrelation function, while the conventional von Neumann-based methods only have a de-bias function.

At the logic level, an input-symbol-based code assignment is proposed for simple logic. For example, 16 output cases can be reduced to 2 output cases using the input symbol as output codes.

Targeting more than 50% *ExE* , VN_8W with 62.21% *ExE* is designed and fabricated in 130-nm CMOS. Its total area including the power and ground line is 4900 μm^2. Its de-bias and de-correlate abilities are verified by NIST SP 800-22 and NIST SP 800-90B IID tests. Measured at 0.45 V, 1 MHz, VN_8W achieves low energy of 0.18 pJ/bit at 0.45 V, which is suitable for low energy TRNG designs.

Chapter 4

Latch-Based True Random Number Generator Featuring Mismatch Compensation and Random Noise Enhancement

4.1 Introduction

In this Chapter, a latch-based TRNG is presented. The TRNG core features mismatch self compensation and random noise enhancement. The complex calibration and feedback control circuit in the conventional work [47] is removed. By XORing only four entropy source latch circuits, the TRNG core achieves high randomness against PVT variations, while 256 latches are needed in the previous work [48]. Combined with the VN_8W post-processing technique shown in Chapter 3, the total TRNG with high randomness and robustness achieves an ultra-low energy of 0.186 pJ/bit at 0.3 V.

The mismatch self compensation is achieved by putting the initial state point close to the metastable operation point M. This is because M is always located in the watershed line of "1" and "0" regardless of the process variations induced mismatches. Thereby, the initial state start from M can obtain balanced "1" and "0". It is realized by adding gate capacitance into the inverters of the latch.

The random noise enhancement is achieved by adding an resistive-capacitive (RC) delay in each inverter's feedback loop. A damped oscillation system is constructed through this feedback loop. The small noise seed is amplified into a large magnitude noise with a random phase. A large resistor is newly added to the gate side of each inverter for about 3 times noise enhancement.

The remainder of this Chapter is organized as follows: First, the entropy source latch circuit is introduced. Second, the theory of mismatch compensation and the capacitor size selection are described. Third, the noise enhancement is described by using parameters of damped oscillation waveform. After that, the mismatch-to-noise ratio improvement by using XOR functions is introduced. Then, the full entropy extraction structure using VN_8W is shown. After that, the experimental results based on 130-nm CMOS implementation are presented. The TRNG achieves low energy of 0.186 pJ/bit with high randomness and robustness, suitable for energy-constrained IoT devices. Finally, the conclusion is drawn.

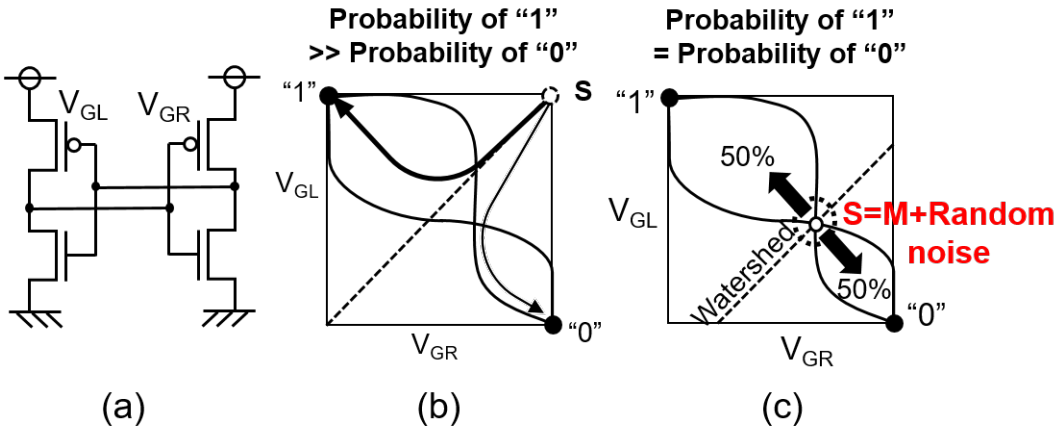


FIGURE 4.1: Concept of mismatch compensation. (a) Inverter pair. (b) Conventional method. (c) This approach.

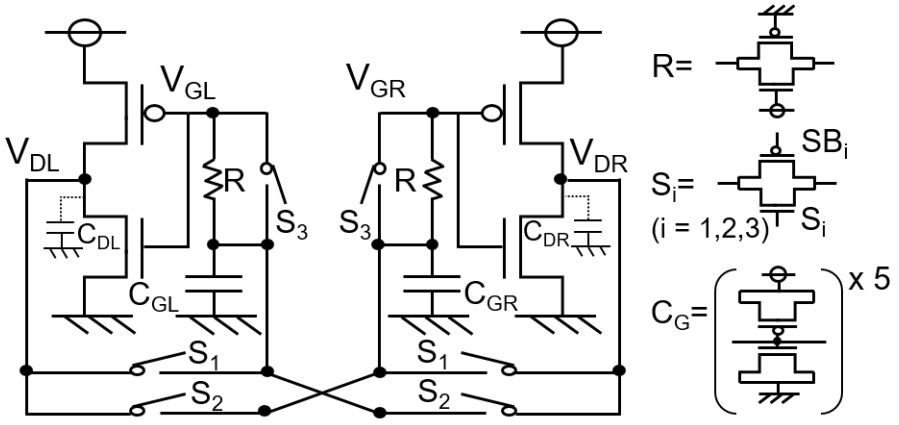


FIGURE 4.2: Entropy source (ES) latch circuit.

4.2 TRNG Entropy Source Latch Circuit

The basic structure of a latch-based TRNG is shown in Figure 4.1(a). It consists of two CMOS inverters, which are cross-coupled connected. In ideally, its voltage transfer curves (VTCs) are symmetric. However, due to the mismatch variations, the VTCs are asymmetric in reality. Therefore, if the pre-charged state (V_{GL} = V_{GR}="1") are used as the initial state of the TRNG, the final resolutions will be biased one or zero, as shown in Figure 4.1(b). Therefore, I propose to put the initial state on the cross-point or the metastable point, M, as shown in Figure 4.1(c). M is always located in the watershed line of final resolution of "1" or "0" even under mismatch variations. Thereby, the initial state start from M can obtain balanced "1" and "0". Furthermore, random noise

TABLE 4.1: Transistor size of ES latch circuit

	Both pMOS and nMOS	
	Width (μm)	Length (μm)
Latch Inverter	Wmin	0.5
R	Wmin	5
C_G (1/5)	0.6	0.2
S_1–S_3	Wmin	0.13

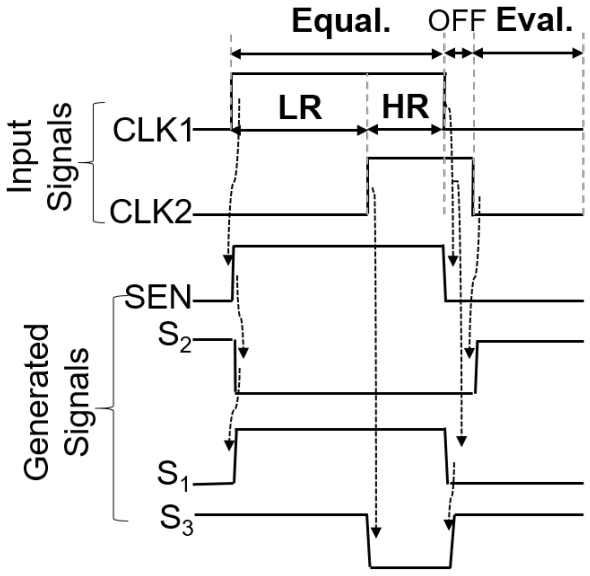
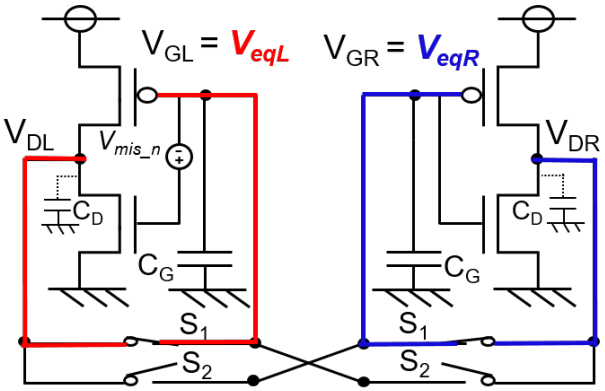


FIGURE 4.3: Control signal waveform.

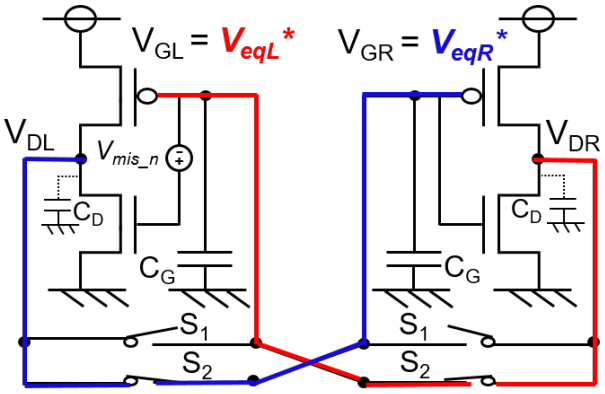
is enhanced on M. High randomness TRNG is achieved without calibration or feedback control circuits.

The schematic of entropy source (ES) latch circuit is depicted in Figure 4.2. In addition to the basic latch structure, a resister (R), gate capacitor (C_G), and three switches (S_1–S_3) are added in each side inverter. The details of transistor size are shown in Table 4.1. R is constructed by the transmission gate with 10 times longer gate length than the size of latch inverter. Considering the clock feedthrough induced randomness drop, the switches are designed with equal sizes of nMOS and pMOS pairs. Besides, the common-mode noise by the clock feedthrough is canceled by the differential operation of the latch. The whole circuit is designed to be symmetric. Still, there may have residual nonidealities, which can be improved by noise enhancement and by taking XOR of four entropy sources, as shown in Chapter 4.5.

Basically, the ES latch includes two operation phases: the equalization phase (S_1 on, S_2



(a)



(b)

FIGURE 4.4: Two steps of mismatch compensation. (a) Equalization. (b) Evaluation.

off) and the evaluation phase (S_1 off, S_2 on). The switch signals are generated from a clock driver with two inputs: CLK1 and CLK2. Figure 4.3 shows the waveform. Each switch signal includes a complementary pair. The SEN signal is used in a sense latch circuit based on a strong-arm latch to read the V_{GL} and V_{GR} differences for the final output. An "OFF" phase is set between the equalization and evaluation phase to prevent their overlapping.

4.3 Mismatch Compensation

The mismatch compensation is achieved with the help of C_G. Figure 4.4 illustrates the two steps of the mismatch compensation:

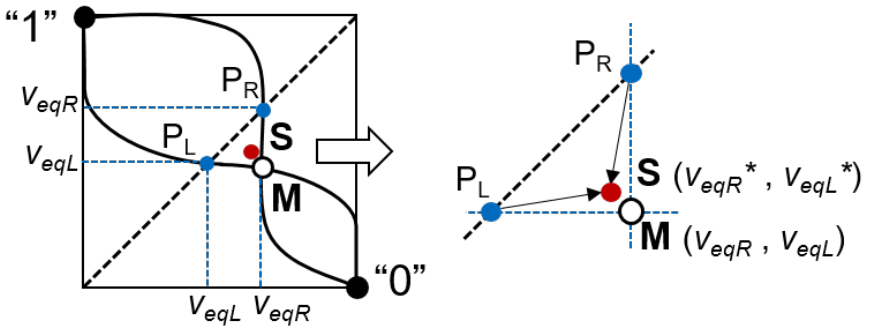


FIGURE 4.5: Position of the initial state S.

Step 1, in equalization phase, the gate and drain voltages are equalized and stored at C_G in each inverter side. The equalized voltages are denoted as V_{eqL} and V_{eqR}, respectively. Due to mismatch, they are not equal, as shown in Figure 4.5.

Step 2, at the beginning of the evaluation phase, the inverters are connected in cross-coupled state. Owing to the voltage stored in the capacitors, the initial state S (V_{eqR}*, V_{eqL}*) are set close to M, as shown in Figure 4.4(b) and Figure 4.5.

In the ideal case of infinity inverter gain, the coordinate of M is (V_{eqR}, V_{eqL}). Parasitic C_D can affect the position of S through charge redistribution. But it can be solved by setting $C_G \gg C_D$. The compensation efficiency (η_{com}) defined in Equation (4.1) is expressed by the function of C_D/C_G in Equation (4.2). The mismatch is self-compensated in this way.

$$\eta_{com} \triangleq \frac{V_{eqL}{}^* - \mathrm{V_{eqR}}^*}{V_{eqL} - V_{eqR}} \tag{4.1}$$

$$\eta_{com} \triangleq \frac{1-\frac{C_D}{C_G}}{1+\frac{C_D}{C_G}} \approx 1 - \frac{2C_D}{C_G} (\text{when } \mathrm{C_G} \gg \mathrm{C_D}). \tag{4.2}$$

Results of the transient simulation without noise for the circuit with 20-mV nMOS artificial mismatch is shown in Figure 4.6. As C_G increases from 5 to 40 fF, S is set closer to M. The comparison between simulation and theoretical values in Equation (4.2) are exhibited in Figure 4.7.

Considering the size of C_G, there is a tradeoff between the mismatch compensation efficiency and noise filtering effect. First, the mismatch (d) is inversely proportional to

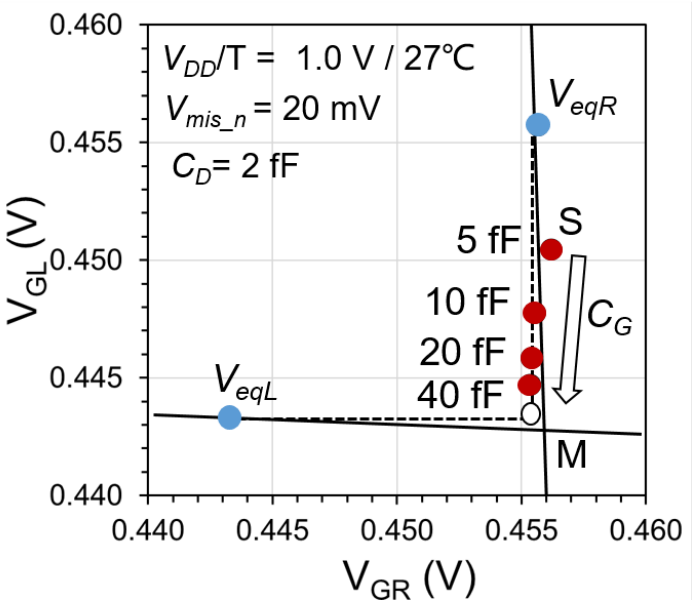


FIGURE 4.6: Position of S as a function of C_G.

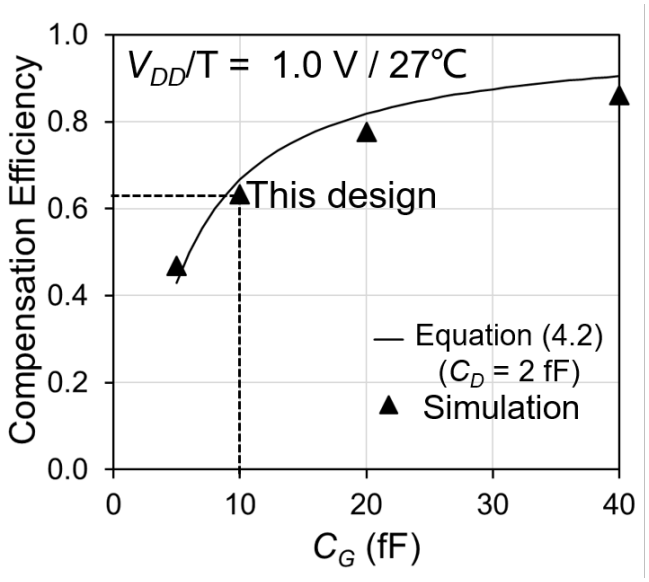


FIGURE 4.7: Mismatch compensation ratio versus C_G.

C_G, according to $d \propto (1-\eta_{com}) \propto \frac{1}{C_G}$. Then, the thermal noise voltage σ_n is inversely proportional to $\sqrt{C_G}$, according to $\sigma_n^2 \propto \frac{kT}{C}$. From the latch output model, the output is evaluated by the mismatch-to-noise ratio ($\frac{d}{\sigma_n}$) [50]. The smaller the ratio, the better the randomness. As a result, $\frac{d}{\sigma_n} \propto \frac{1}{\sqrt{C_G}}$. Thus, by using larger C_G, the randomness is improved.

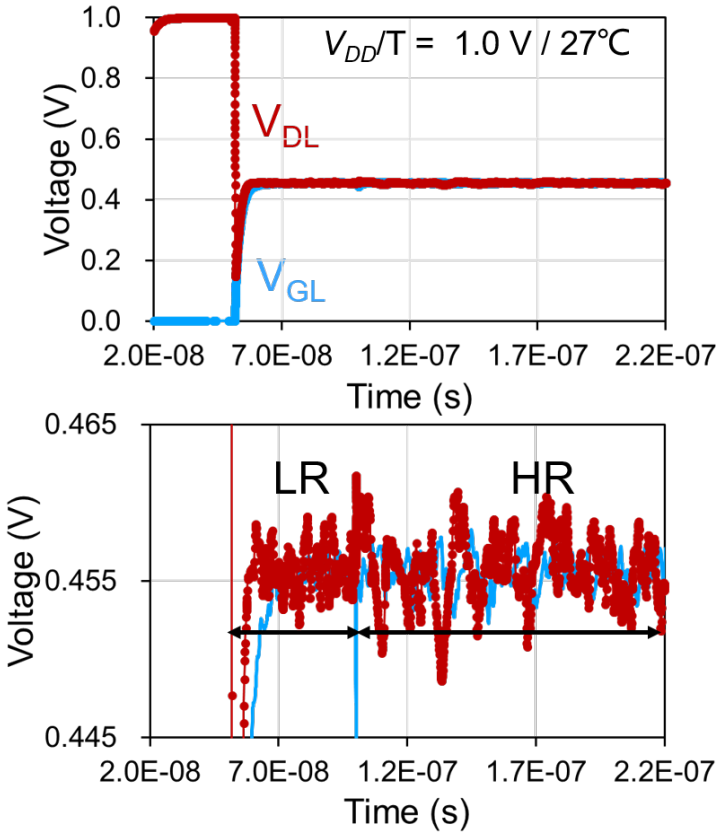


FIGURE 4.8: Simulated noise waveform in equalization phase.

On the other hand, larger C_G consumes a lot of power. Therefore, the C_G is set to 10 fF with 63.3% compensation efficiency, as shown in Figure 4.7.

4.4 Noise Enhancement

As depicted in Figure 4.2, the equalization phase is divided into low resistance LR phase and high resistance HR phase by setting S_3 on and off. The drain and gate voltages are quickly equalized in the LR phase, as shown in Figure 4.8. Note that the LR duration should be larger than the time in which the differences of the average drain and gate voltages from the equilibrium state become smaller enough than the noise voltage. In the HR phase, a damped oscillation is introduced in the feedback loop by the resistance and capacitance induced RC delay. In which, the main part of C is the input capacitance of the inverter. The small noise seed in the LR phase is amplified into larger magnitude noise with random phase.

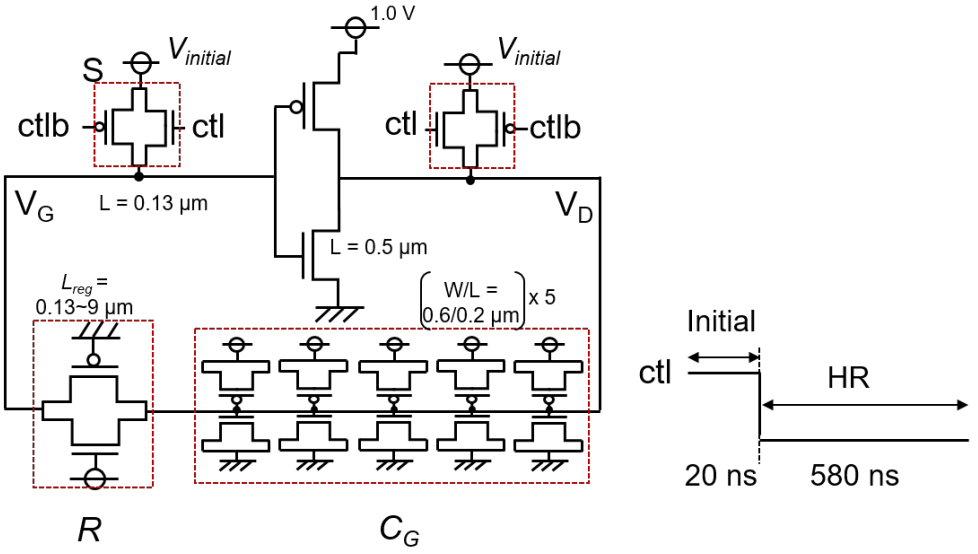


FIGURE 4.9: Simplified half ES latch circuit for damped oscillation system simulation.

To quantify the noise enhancement by the damped oscillation, a half ES latch circuit with initial voltage setting switches, S, is used, as shown in Figure 4.9. It should be noted that, the initial voltages of V_G and V_D are intentionally set to $V_{initial}$, which is far away from the equilibrium state V_{eq}, to observe the damped oscillation parameters. In the real operation, by using LR phase for equilibrium initial state, the system skips the damped oscillation period in Figure 4.10 and goes into the thermal noise dominate state immediately at the beginning of the HR phase, as shown in Figure 4.8.

Monte-Carlo noise simulation with 100-runs is done with the resister gate length L_{reg} ranging from 0.13 to 9 μm. Figure 4.10 shows an example when L_{reg} = 5 μm. Because of the non-equilibrium initial state, V_D shows a large swing at first, then exhibits damped oscillation. The damping ratio ζ are obtained from these amplitudes, according to $ln(A_1/A_2) = 2\pi\zeta/\sqrt{1-\zeta^2}$ [55]. After that, the system goes into a random noise dominant state. The random noise is quantified by noise voltage peak to peak (Vpp) and RMS of noise voltage (σ_n). Figure 4.11 summarizes the random noise and damping ratio under L_{reg} variations. When L_{reg} reaches to 2 μm, the system state changes from overdamped ($\zeta \geq 1$, no oscillation) to underdamped ($0 < \zeta < 1$, damped oscillation). When L_{reg} reaches to 9 μm, the system goes into a full oscillation state. The damped oscillation is used for noise enhancement. As L_{reg} increases, Vpp and σ_n increase. Vpp

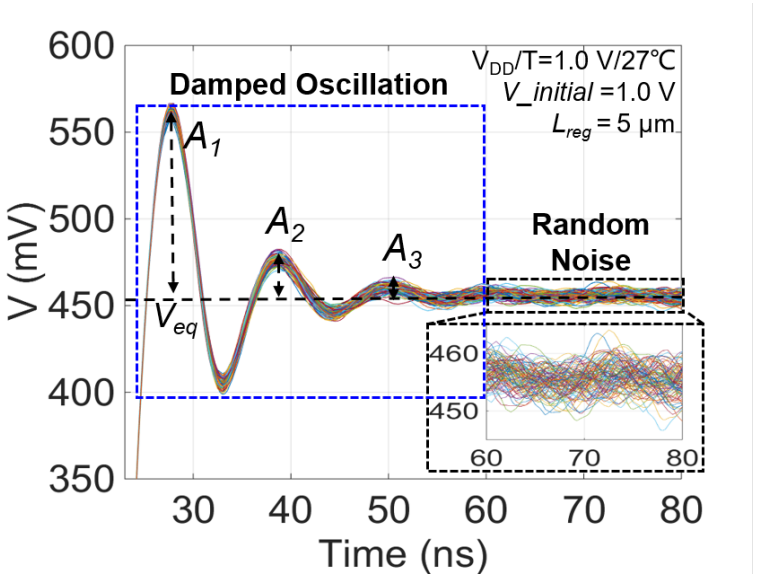


FIGURE 4.10: Monte-Carlo noise simulation of 100 runs under non-equilibrium initial voltage.

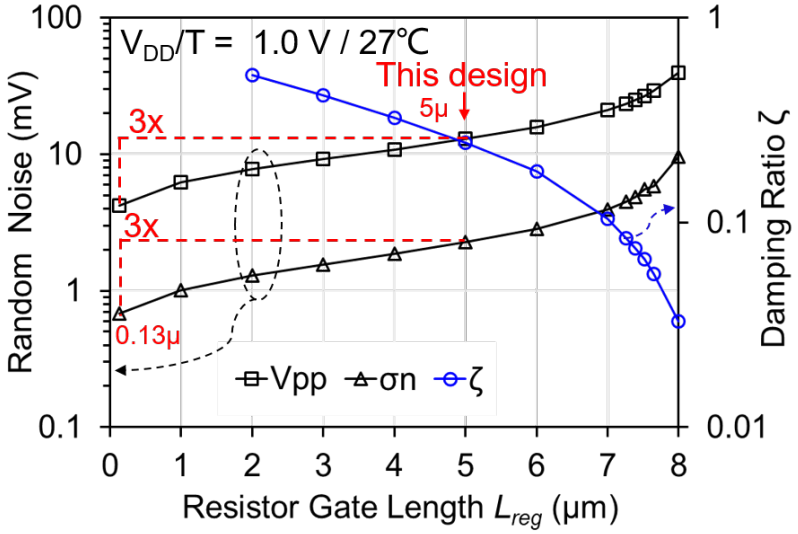


FIGURE 4.11: Random noise and damping constant versus resister gate length.

is always 5 to 6 times of σ_n. To investigate the relationship between the noise σ_n and damping ratio a (=A_2/A_1), a gain factor (GF) extracted from an analogous to the sum of geometric progression is introduced, as shown in equation (4.3):

$$GF = \frac{1}{1-a}. \tag{4.3}$$

σ_n and GF has a linear relationship as depicted in Figure 4.12. This indicates that

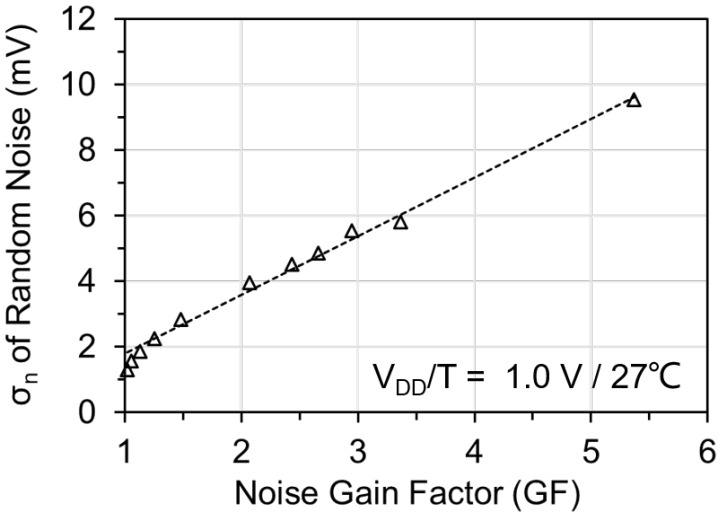


FIGURE 4.12: RMS noise voltage versus noise gain factor.

TABLE 4.2: Summary of the damped oscillation parameters and random noise under voltage and temperature variations when L_{reg} = 5 μm

V_{DD}/Temperature (V/°C)	ζ	Vpp (mV)	σ_n (mV)	GF
1.0/27	0.25	12.9	2.26	1.25
1.0/–20	0.24	12.1	2.13	1.27
1.0/100	0.24	14.2	2.51	1.27
0.5/27	0.50	11.2	2.23	1.03
0.5/–20	0.58	9.1	1.96	1.01
0.5/100	0.40	13.6	2.57	1.07

the σ_n is enhanced by GF. The summary of the damped oscillation parameters and random noise under voltage and temperature (VT) variations are shown in Table 4.2. It is indicated that the stable random noise across VT variations.

The power spectral density (PSD) analysis of the enhanced noise when L_{reg} = 0.13, 5, and 7 μm are shown in Figure 4.13. When L_{reg} = 0.13 μm, the system is overdamped and the PSD is flat until the cutoff frequency. When L_{reg} = 5 and 7 μm, the system is in damped oscillation state. Noise is shaped with the damped oscillation. In the low-frequency region, the PSD is flat. It increases as L_{reg} increases. Because the noise power is proportional to resister value: $4kTR$. In the middle-frequency range, a mild peak in resonance with damped oscillation appears. This part dominates the total noise power. In the high-frequency range, L_{reg} has little effect. The bandwidth defined by 3 dB below the peak of PSD is f_L= 63.2 MHz and f_H = 96 MHz, f_L= 58.7 MHz and f_H

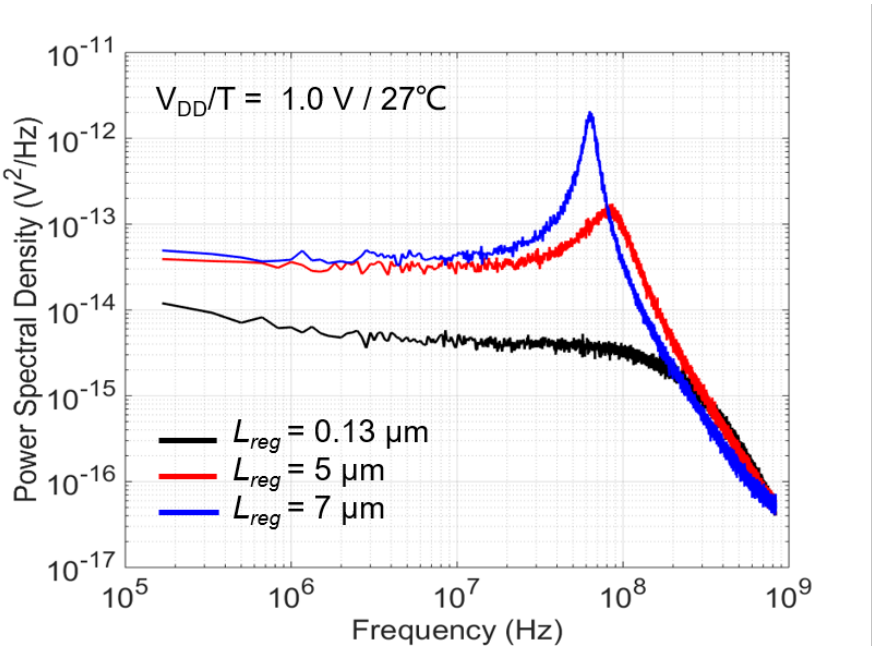


FIGURE 4.13: Power spectral density of random noise in HR phase.

= 66.7 MHz when L_{reg} = 5 μm and 7 μm, respectively. Their f_H/f_L ratios of 1.52 and 1.14 are much larger than that of full oscillation (< 1.01 at L_{reg} = 9 μm). Therefore, even if the phases are aligned at beginning due to the non-equilibrium initial state, they become random after waiting several cycles, as shown in Figure 4.10.

L_{reg} is set to 5 μm in this deign. The equivalent value is 0.43 MΩ at 1.0 V/27°C during the HR phase. It achieved 2.26 mV σ_n and 12.9 mV Vpp of random noise in simulation. Both noise Vpp and σ_n are improved by 3 times when compared with 0.13 μm L_{reg}.

4.5 Mismatch-to-Noise Ratio Improvement by XOR

The process of generating an output bit in a latch-based TRNG can be modeled as a differential buffer [56]. According to the model, the two inputs are mismatch voltage d, noise voltage V_n. The output is high if $d \geq V_n$ and is low when $d < V_n$. The mismatch d is dependent on the process variations, while the noise V_n can be modeled as a Gaussian distribution, followed $\mathcal{N}(0, \sigma_n^2)$. By normalizing d with σ_n, the relationship between the probability of "1" p and the mismatch-to-noise ratio ($\frac{d}{\sigma_n}$) can be expressed in the

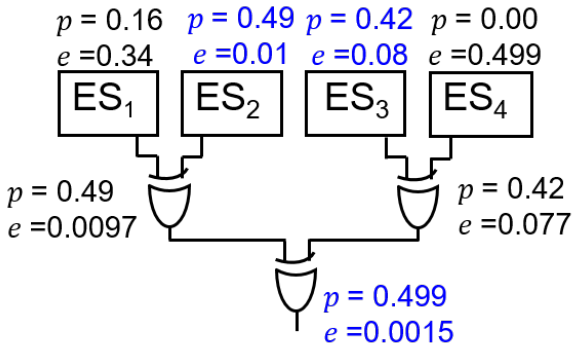


FIGURE 4.14: XORing of four entropy sources.

following Equation [50]:

$$p = \Phi(\frac{d}{\sigma_n}), \tag{4.4}$$

where $\Phi(x)$ is the cumulative distribution function (CDF) of the normal distribution $\mathcal{N}(0,1)$, thus, $\frac{d}{\sigma_n}$ can be obtained by the inverse CDF of p:

$$\frac{d}{\sigma_n} = \Phi^{-1}(p) \tag{4.5}$$

Exclusive-OR(XOR) function is a light-weight post-processing technique used for de-biasing. The bias e is calculated from p. Thus by XORing N bitstreams from multiple entropy sources (ESs), the final p can be expressed as:

$$p_{N-bitXOR} = 0.5 \pm 2^{N-1} \prod_{i=1}^{N} e_i, \tag{4.6}$$

$$e_i = |p_i - 0.5|, \tag{4.7}$$

where e_i and p_i are bias and p in ES_i, respectively. $p_{N-bitXOR}$ is dominated by the smallest e_i. Thus, as long as one ES performs well, the final results is close to the ideal value. Figure 4.14 shows an example of the effectiveness of 4-bit XOR. By increasing the number of N in N-bit XOR, $p_{N-bitXOR}$ can get more close to 0.5. However, N-bit XOR consumes a lot of power and area.

For a high randomness TRNG output, smaller $\frac{d}{\sigma_n}$ is better. As shown in Equation (4.5), the mismatch-to-noise ratio is an inverse CDF function of p. Thus, the target of $\frac{d}{\sigma_n}$ can

TABLE 4.3: Stochastic calculation results

		Input	Calculated Results		
		Raw ES	2-bit XOR	4-bit XOR	8-bit XOR
	Bit Length	20000	10000	5000	2500
Distribution 1	μ	0.1400	-0.0297	-0.0007	0.0000
	σ	0.1981	0.0339	0.0015	0.0000
Distribution 2	μ	0.3513	-0.1624	-0.0197	-0.0003
	σ	0.4998	0.1670	0.0332	0.0009
Distribution 3	μ	0.6945	-0.4696	-0.1417	-0.0145
	σ	0.9983	0.4386	0.1953	0.0345
Distribution 4	μ	1.0668	-0.7946	-0.3204	-0.0712
	σ	1.5099	0.6939	0.3773	0.1384
Distribution 5	μ	1.4120	-1.1055	-0.5075	-0.1442
	σ	1.9916	0.9321	0.5355	0.2170
Distribution 6	μ	1.6056	-1.2892	-0.6314	-0.2020
	σ	2.2836	1.0662	0.6528	0.2892

be achieved by choosing appropriate N-bit XOR under the limited hardware cost. In this work, the performance of 2-bit XOR, 4-bit XOR, and 8-bit XOR are verified based on stochastic calculations. The calculation process is summarized into four steps:
Step 1: generate a 20,000 random data of $\frac{d}{\sigma_n}$ of ES, each data follows the distribution of $\mathcal{N}(\mu_{ES}, \sigma_{ES}^2)$, σ_{ES} takes the value ranging from 0 to 2.5, $\mu_{ES} = 0.7\sigma_{ES}$;

Step 2: calculate the related p_{ES_i} according to Equations (4.4) and (4.7);

Step 3: divide the 20,000 data into 20,000/N groups, each group with N bits (N takes the value of 2, 4 and 8), and then derive $p_{N-bitXOR}$ by XORing the N bits of p_{ES_i} in each group, according to Equation (4.6) ;

Step 4: obtain the $\frac{d}{\sigma_n}$ for 2-bit XOR, 4-bit XOR, and 8-bit XOR, respectively;

The summary of μ and σ of $\frac{d}{\sigma_n}$ in raw data and calculated data are shown in Table 4.3. The μ and σ is reduced by increasing the number bits of XOR function. For example, in distribution 1, both μ and σ is reduced to zero by using 8-bit XOR. However, it needs eight entropy source circuit to generate one output. There is a tradeoff between the mismatch-to-noise ratio improvement and energy consumption.

Considering using VN_8W post-processing for zero bias output and targeting for more than 50% *ExE*, the required p before VN_8W should satisfy $0.27 \leq p \leq 0.73$ [52], as

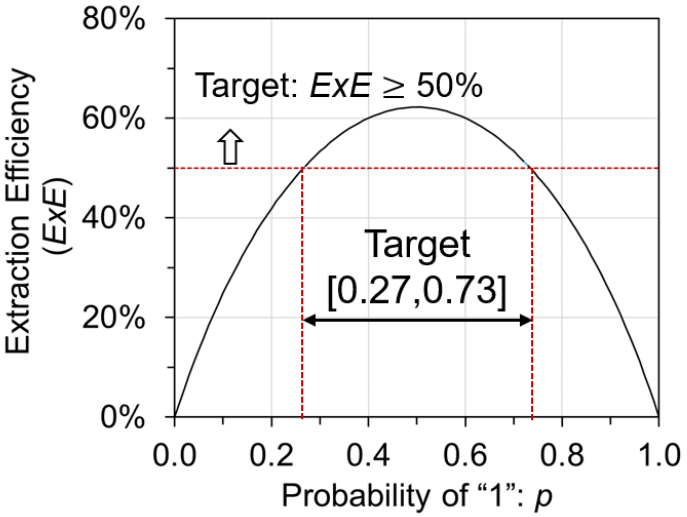


FIGURE 4.15: Extraction efficiency of VN_8W and target ranges.

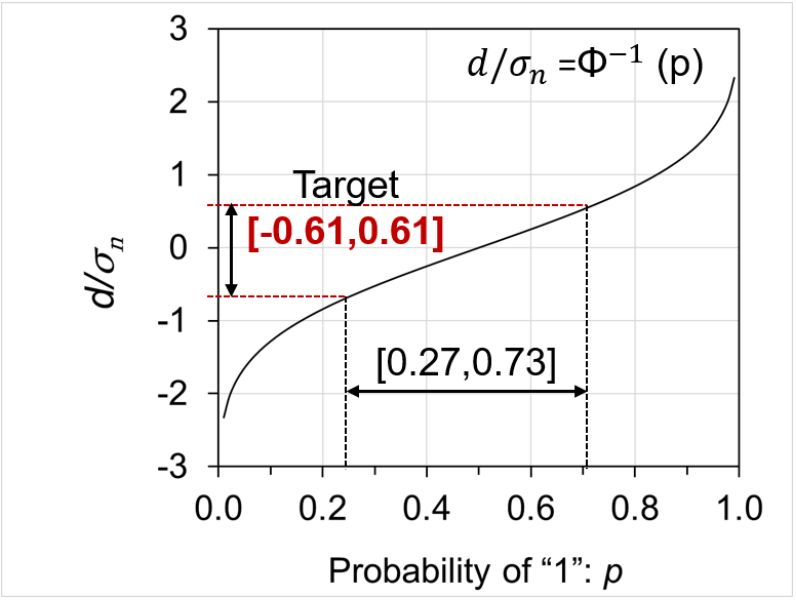


FIGURE 4.16: Mismatch-to-noise ratio versus probability of "1".

shown in Figure 4.15. The related $\frac{d}{\sigma_n}$ should be located within the interval of [–0.61, 0.61] (where $0.61 = \Phi^{-1}(0.73)$, $\Phi(x)$ is a CDF of a $\mathcal{N}(0,1)$), as presented in Figure 4.16.

When consider a 6σ variation, $\mu \pm 6\sigma$ should less than 0.61. Since μ is around 0, thus, the target σ is around 0.102. Assuming σ_n is constant, then the σ of $\frac{d}{\sigma_n}$ summarized in Table 4.3 can be described as $\frac{\sigma_d}{\sigma_n}$. Figure 4.17 summarized the $\frac{\sigma_d}{\sigma_n}$ after XOR function versus raw ES data. As can be seen from the figure, the σ requirement in raw ES data

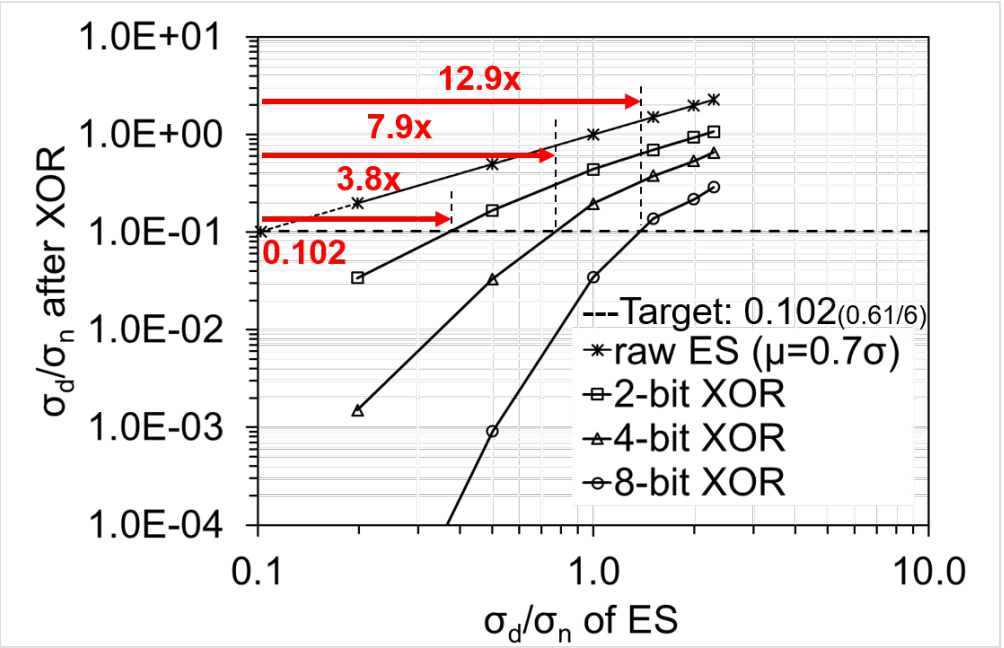


FIGURE 4.17: Standard deviation of mismatch-to-noise ratio: after XOR versus raw ES.

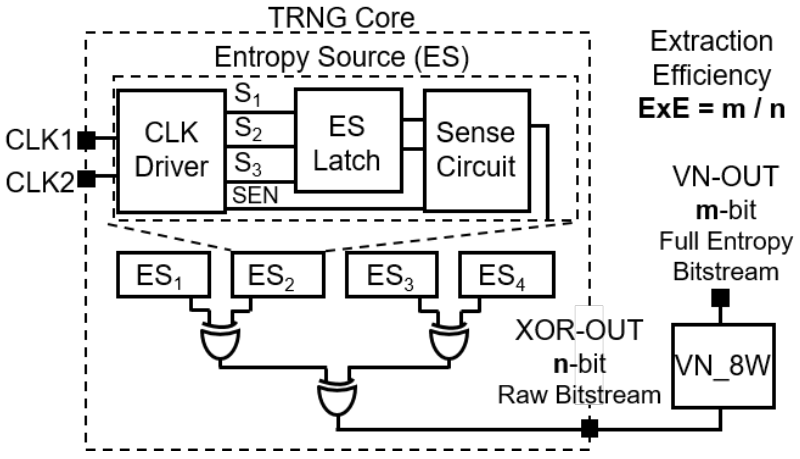


FIGURE 4.18: Full entropy extraction structure.

can be relaxed to 0.39 (3.8x), 0.81(7.9x), 1.32(12.9x) by 2-bit XOR, 4-bit XOR, and 8-bit XOR, respectively. 4-bit XOR is applied to the TRNG core design. Its performance is verified by measurement result, as shown in Chapter 4.7.2.

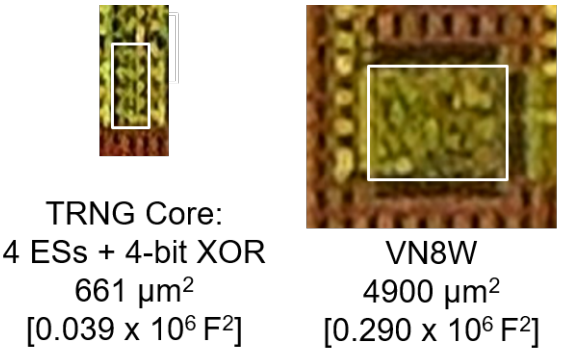


FIGURE 4.19: Die micrographs of TRNG core and VN8W.

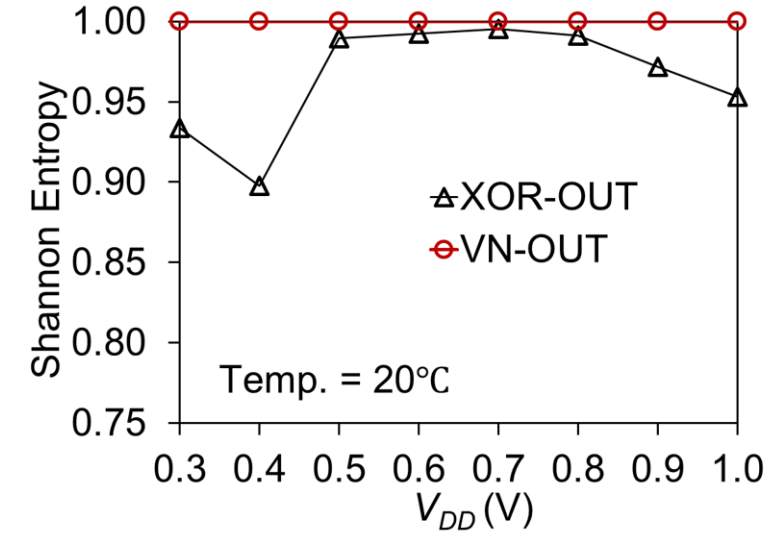
4.6 Full Entropy Extraction

To achieve a cryptographic-grade full entropy extraction, a total TRNG is proposed, as shown in Figure 4.18. It consists of a TRNG core block and VN_8W [52] post-processing block. The TRNG core block is built by four entropy sources (ESs) with a 4-bit XOR circuit. Each ES circuit includes a clock driver, an ES latch circuit, as shown in Figure 4.2, and a sense latch circuit.

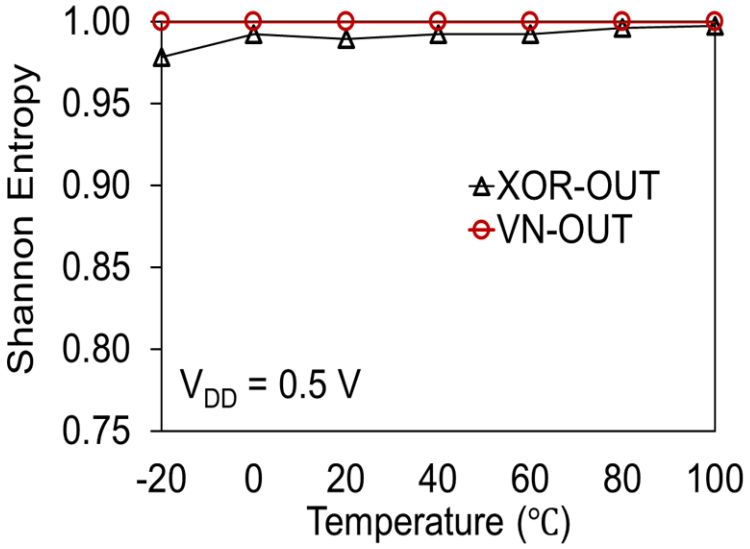
The 4-bit XOR output (XOR-OUT) provides a high to middle level randomness raw bitstream. The residual bias and correlations is removed by VN_8W for achieving a cryptographic level full entropy bitstream VN-OUT. The extraction efficiency (*ExE*) is defined as n bits XOR-OUT over m bits VN-OUT. VN_8W achieved 62.21% *ExE* at zero bias, which is 2.49x larger than the conventional von Neumann method. Although VN_8W has a larger area, it can bring higher throughput and energy efficiency to the TRNG core. By using energy efficient hardware implementations [52], the total energy can be minimized.

4.7 Experimental Results

The prototype of latch-based TRNG is implemented into 130-nm CMOS. Figure 4.19 shows the die micrographs of TRNG core and VN8W. The TRNG core, including four ESs and 4-bit XOR, occupied an area of 661 μm^2, which is 0.0391×10^6 F^2 when



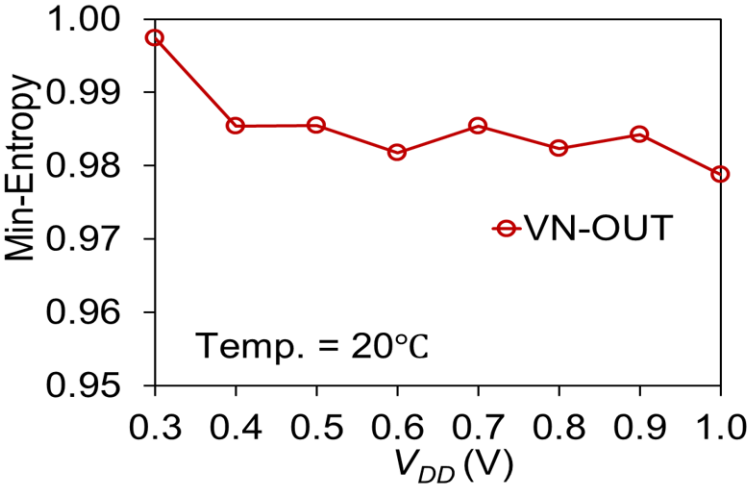
(a)



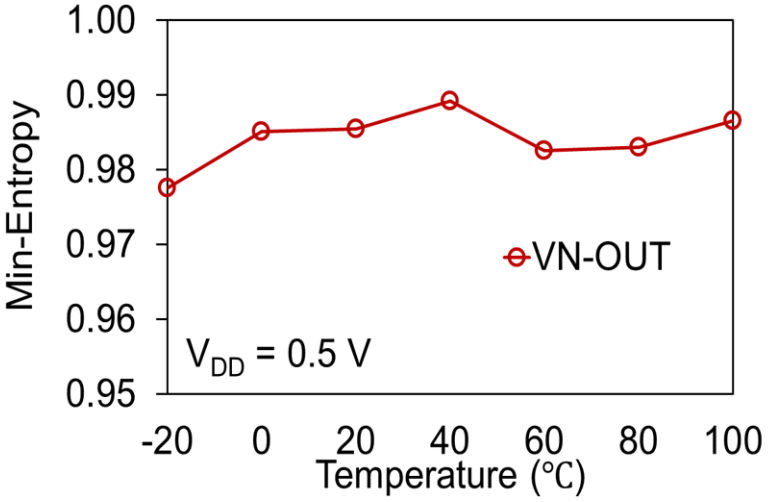
(b)

FIGURE 4.20: Shannon entropy under (a) voltage variations, (b) temperature variations.

normalized with the feature size F (130-nm in this case). VN8W occupied an area of 4900 µm^2 (0.2899×10^6 F^2). Although VN_8W dominates the total area, it can be improved using advanced technologies.



(a)



(b)

FIGURE 4.21: Min-entropy under (a) voltage variations, (b) temperature variations.

4.7.1 Randomness Verification and Autocorrelation Check

To verify the randomness, six chips with 12 TRNGs are measured at 0.3–1.0 V/20°C and –20°C–100°C/0.5 V. Shannon entropy results of XOR-OUT (denoted by the black line) and VN-OUT (denoted by the red line) under voltage and temperature variations are shown in Figure 4.20 (a) and (b), respectively. Shannon entropy of XOR-OUT is greater than 0.90, excluding the point at 0.4 V/20°C, which is 0.898. Shannon entropy of VN-OUT is always close to 1. Shannon entropy is not sensitive when p is close to 0.5. Therefore, min-entropy of VN-OUT is redrawn and shown in Figure 4.21 (a) and

(b), respectively. Min-entropy of VN-OUT is greater than 0.978 for approximately 6k random bitstream, which is within the stochastic error and high enough for cryptography applications.

TABLE 4.4: NIST SP 800-22 test results for low voltage corner and temperature corners.

	V_{DD} = 0.3 V, T = 20°C		V_{DD} = 0.5 V, T = –20°C		V_{DD} = 0.5 V, T = 100°C	
	Ave. $P-value$	Passed	Ave. $P-value$	Passed	Ave. $P-value$	Passed
Frequency	0.22	11/12[a]	0.44	16/16	0.56	16/16
Block Frequency	0.42	12/12	0.56	16/16	0.49	16/16
Runs	0.40	11/12[a]	0.45	16/16	0.43	16/16
Longest Runs	0.54	12/12	0.53	16/16	0.57	16/16
Rank	0.32	12/12	0.52	16/16	0.51	16/16
FFT	0.61	12/12	0.33	16/16	0.64	16/16
Non-Overlapping Template	0.47	12/12	0.50	16/16	0.50	16/16
Overlapping Template	0.45	12/12	0.46	16/16	0.55	16/16
Universal	0.50	12/12	0.46	16/16	0.60	16/16
Linear Complexity	0.42	12/12	0.45	16/16	0.57	16/16
Serial	0.48	11/12[a]	0.42	16/16	0.39	16/16
Approximate Entropy	0.36	12/12	0.59	16/16	0.58	16/16
Cumulative Sums	0.20	11/12[a]	0.39	16/16	0.56	16/16
Random Excursions	0.41	7/7	0.40	6/6	0.48	8/8
Random Excursions Variant	0.44	7/7	0.28	6/6	0.45	8/8
	Input: Each 1M bits of 12 TRNGs from 6 chips		Input: Each Four 1M bits of 4 TRNGs from 2 chips			

[a] Acceptable pass ratio is 0.904 @ 12 bitstreams [13].

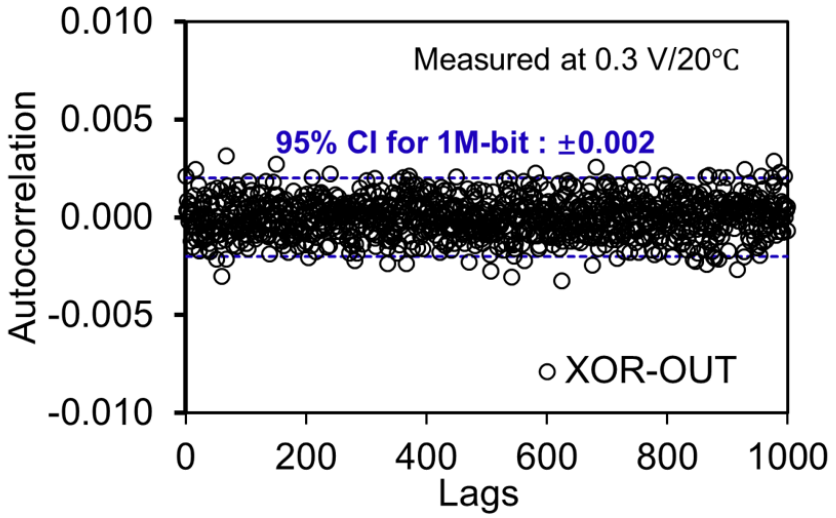


FIGURE 4.22: Autocorrelation check result.

TABLE 4.5: NIST SP 800-90B IID test results for low voltage corner and temperature corners.

	0.3 V/20°C	0.5 V/–20°C	0.5 V/100°C
	Passed	Passed	Passed
Chi Square Independence	2/2	2/2	2/2
Chi Square Goodness-of-fit	2/2	2/2	2/2
IID Permutation Tests	2/2	2/2	2/2
Restart Test	2/2	2/2	2/2
Min-Entropy Estimate	0.993/ 0.993	0.996/ 0.996	0.996/ 0.996
	Input: Each 1M bits of 2 TRNGs from 1 chip		

Bit correlation is a problem in many TRNG designs. In this research, a correlation problem is avoided by maintaining a sufficient LR phase time, i.e., fully equalizing the drain and gate voltages. Figure 4.22 shows the autocorrelation result of XOR-OUT with 1M-bit length measured at 0.3 V/20°C . Almost all factors are located within the 95% confidence interval, indicating nearly zero correlation. Besides, the zero correlation is further enhanced by VN_8W, thanks to the decorrealtion function design [52]. The randomness is checked by NIST SP 800-22 tests [13] with $P-value \geq 0.01$ for passing. The results are summarized in Table 4.4. For low voltage corner of 0.3 V/20°C, 12 bitstreams each with 1M bits generated by 12 TRNGs across 6 chips are tested. For temperature corners of 0.5 V/–20,100°C, 16 bitstreams each with 1M bits generated by

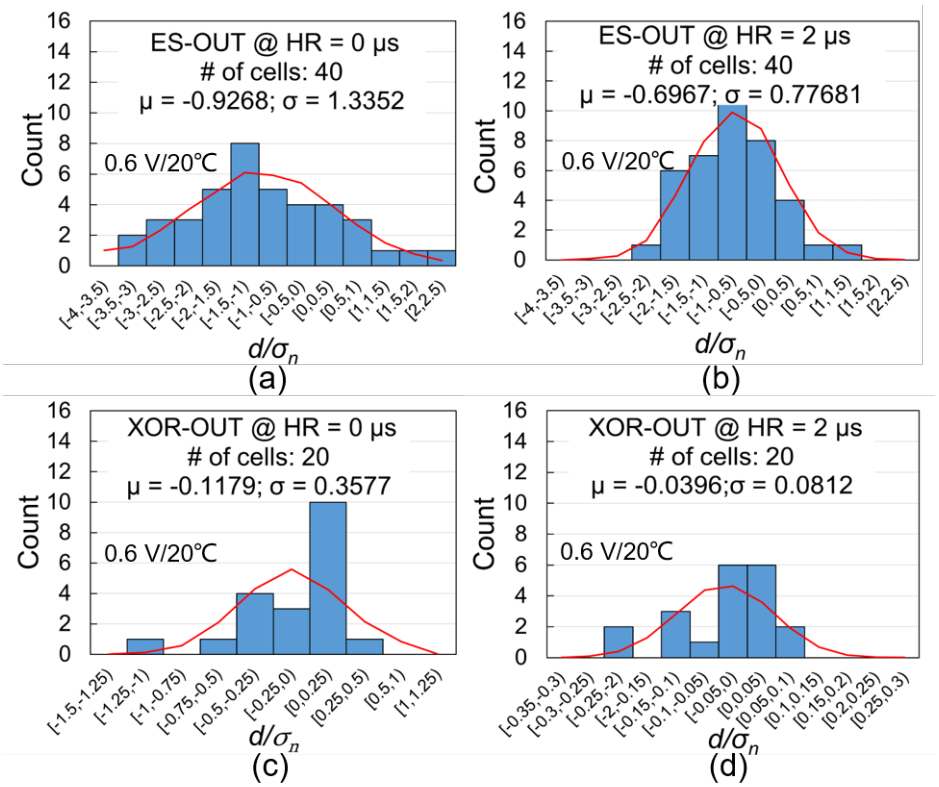


FIGURE 4.23: Measurement results of d/σ_n, (a) ES-OUT @ HR = 0 μs, (b) ES-OUT @ HR = 2 μs, (c) XOR-OUT @ HR = 0 μs, (d) XOR-OUT @ HR = 2 μs.

4 TRNGs across 2 chips are used. The results verified high randomness of the TRNG. The NIST SP 800-90B IID tests results are shown in Table 4.5.

4.7.2 Mismatch-to-noise Ratio Analysis

To verify the effectiveness of 4-bit XOR and noise enhancement by damped oscillation, ten chips with 20 XOR-OUT and 40 ES-OUT are measured at 0.6 V/room temperature under HR = 0 μs (without noise enhancement) and HR = 2 μs (with noise enhancement, 4x larger than the nominal condition for sufficient margin under chip variations), respectively.

The results are summarized in Figure 4.23. ES-OUT at HR = 0 μs has μ = –0.9268 and σ = 1.3352. The values are reduced near to half by adding the HR time (2 μs). The $\frac{d}{\sigma_n}$ ratio is further improved using 4-bit XOR post-processing. XOR-OUT achieved μ = –0.118 (0.127x of ES-OUT), σ = 0.358 (0.268x) @ HR = 0 μs, and μ = –0.0396

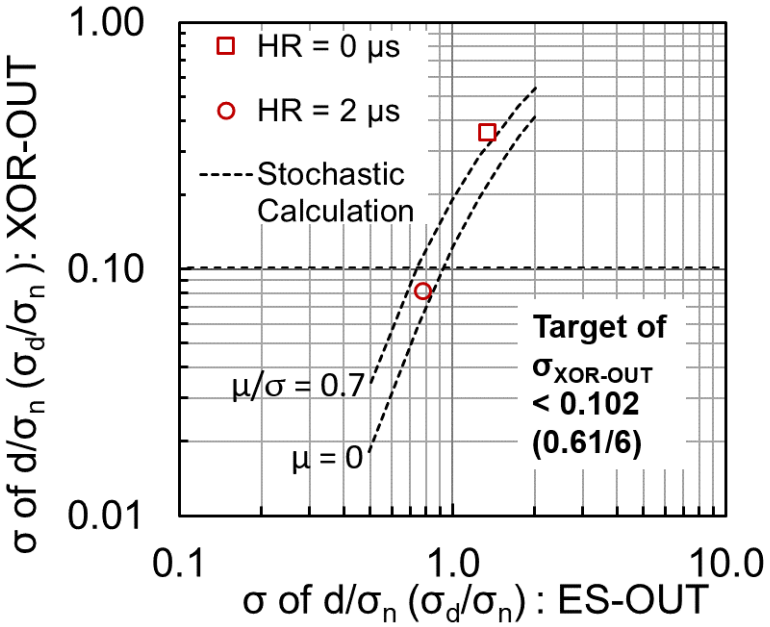


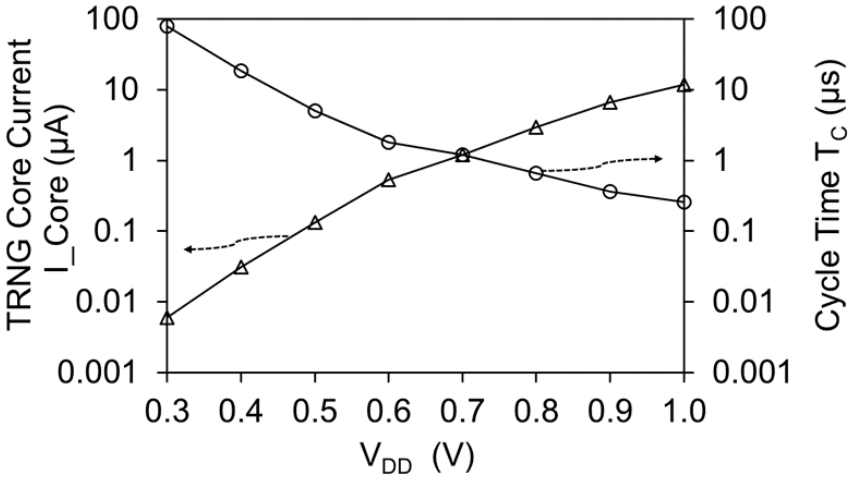
FIGURE 4.24: Standard deviation of mismatch-to-noise ratio: XOR-OUT versus ES-OUT.

(0.0568x), σ = 0.0812 (0.105x) @ HR = 2 μs. Targeting for at least 50% *ExE* after VN_8W post-processing and considering 6σ variations of mass production, as mentioned in Chapter 4.5, $\mu \pm 6\sigma$ should be within the target range of [-0.61, 0.61]. By using 4-bit XOR, XOR-OUT @ HR = 2 μs achieved $\mu \pm 6\sigma$ in [-0.5268, 0.4476]. It indicated 6σ robustness against random mismatch variations.

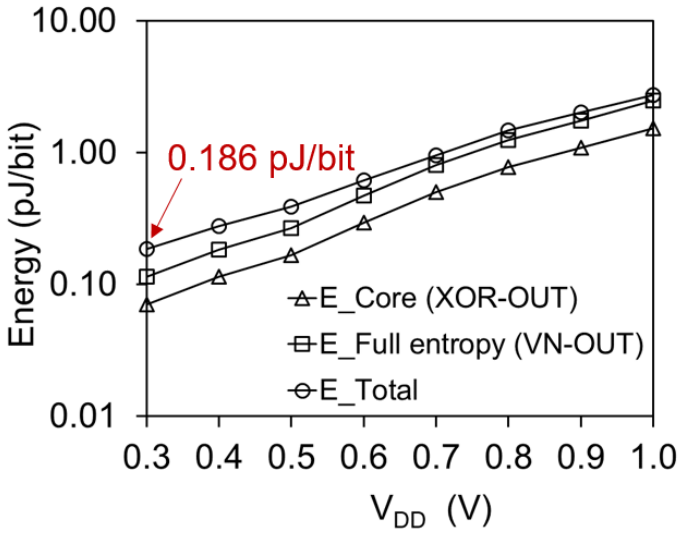
In addition, the measurement results are compared with the stochastic calculation results, as mentioned in Chapter 4.2. The results are summarized in Figure 4.24. The measurement results meet the stochastic calculation lines well, indicating good measurement accuracy.

4.7.3 Energy Consumption and Throughput

The measured operating current and cycle time of one TRNG chip is shown in Figure 4.25(a). About 50% of cycle time is used for LR phase time to ensure the equilibrium state before the HR phase starts. The related energy consumption of XOR-OUT,



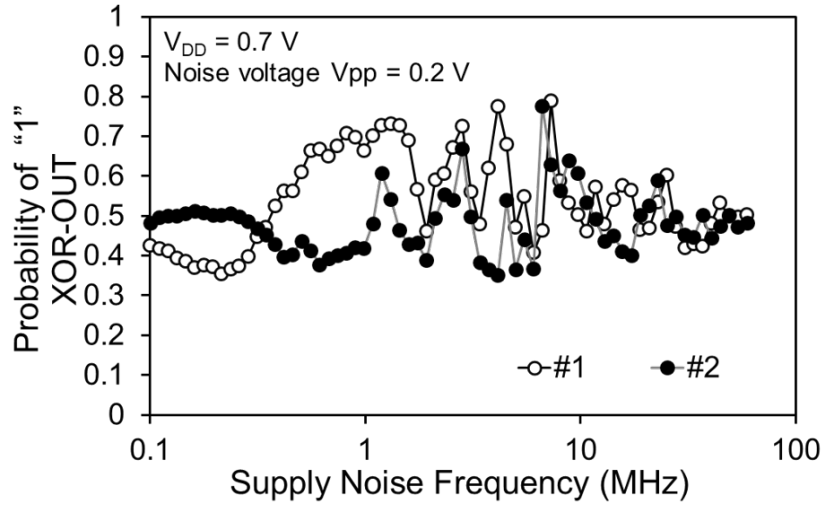
(a)



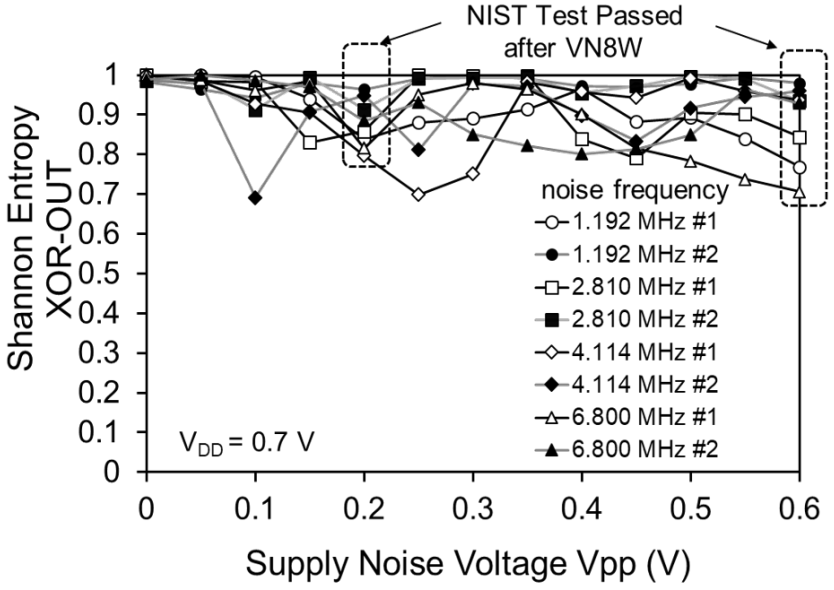
(b)

FIGURE 4.25: AC characteristics. (a) Current and cycle time. (b) Energy consumption.

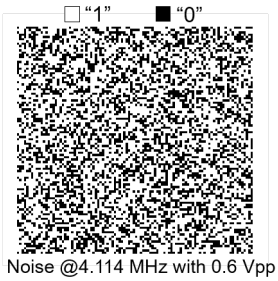
VN-OUT, and total energy are shown in Figure 4.25(b). The TRNG achieved the minimum energy of 0.186 pJ/bit at 0.3 V: 0.114 pJ/bit (VN-OUT) + 0.072 pJ/bit (VN_8W). Energy consumption of VN_8W is smaller than VN-OUT. If the conventional von Neumann is applied, the energy consumption of VN-OUT would exceed 0.28 (0.114×2.49) pJ/bit. The final throughput after VN_8W post-processing is 0.00787 Mb/s at 0.3 V and 2.39 Mb/s at 1.0 V.



(a)



(b)



(c)

FIGURE 4.26: Power noise injection attacks. (a) Supply noise frequency dependence. (b) Supply noise voltage Vpp dependence. (c) Bit map.

4.7.4 Power Injection Attack

The randomness of TRNG may be affected by noise injection attacks. RO-based TRNGs have been reported to a failure under power noise injection attack [17]. To verify the resilience to power noise injection attacks, 2 TRNGs (#1, #2) across one chip are measured. A sine wave noise is injected into the power line with frequency ranging from 0.1 to 59.335 MHz with 1.1x growth step. This covers the simulated noise bandwidth of f_L= 7.9 MHz and f_H = 18 MHz at 0.7 V.

Figure 4.26(a) shows the probability of ones (p) in XOR-OUT under noise frequency variations with 0.2 V noise Vpp. In the frequency range of 1 to 10 MHz, several peaks and bottoms are observed. But the p still within the range of 0.3 to 0.8. Four peak position frequencies of 1.192, 2.810, 4.114, 6.800 are further selected to measure the effect of noise Vpp variations of 0–0.6 V. The results are shown in Figure 4.26(b). The Shannon entropy drops a little at 0.2Vpp and 0.6Vpp. After VN_8W post-processing, the average min-entropy is 0.999 both at 0.2Vpp and 0.6Vpp. Randomness are verified by NIST SP 800-22 tests and NIST SP 800-90B IID tests, as shown in Tables 4.6 and 4.7. One 10k bit map under noise frequency of 4.114 MHz with 0.6 V Vpp is presented in Figure 4.26(c).

The tolerance against power noise injection attacks is summarized into two points. First, the noise bandwidth is wide enough, which is difficult to be resonance with the noise frequency. Second, the random Vth variations induced frequencies variations among 8 inverters in one TRNG core help to avoid a rapid entropy degradation.

4.7.5 Long-Term Reliability

Long-term effects may cause randomness drop in a TRNG. An accelerated aging test is applied to one TRNG chip to verify the long-term reliability. The chip is baked in 2.0 V/125°C for a long period and measured at 0.5–0.8 V/25°C. Figure 4.27 summarizes the average Shannon entropy of four single ES-OUT and two XOR-OUT. The Shannon entropy of ES-OUTs increased a little after 11 hours of aging and decreased a little

TABLE 4.6: NIST SP 800-22 test results under power noise injection attack

	V_{DD} = 0.7 V, T = 20°C Noise Vpp(V): 0.2, 0.6 Fre.(MHz):1.192–6.800	
	Ave. $P-value$	Passed
Frequency	0.52	16/16
Block Frequency	0.51	16/16
Runs	0.54	16/16
Longest Runs	0.46	16/16
Rank	0.42	16/16
FFT	0.56	16/16
Non-Overlapping Template	0.49	16/16
Overlapping Template	0.64	16/16
Universal	0.39	16/16
Linear Complexity	0.54	15/16[a]
Serial	0.44	16/16
Approximate Entropy	0.49	16/16
Cumulative Sums	0.60	16/16
Random Excursions	0.48	12/12
Random Excursions Variant	0.47	12/12
	Input: Each 1M bits of 2 TRNGs from 1 chip	

[a] Acceptable pass ratio is 0.915 @ 16 bitstreams [13].

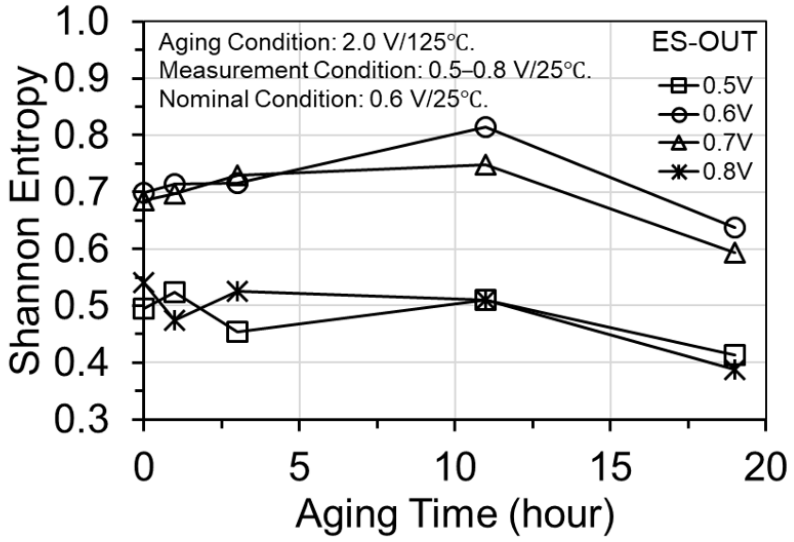
TABLE 4.7: NIST SP 800-90B IID test results under power noise injection attack.

	V_{DD} = 0.7 V, T = 20°C Noise Vpp(V): 0.2, 0.6			
	1.192 MHz	2.810 MHz	4.114 MHz	6.800 MHz
	Passed	Passed	Passed	Passed
Chi Square Independence	4/4	4/4	4/4	4/4
Chi Square Goodness-of-fit	4/4	4/4	4/4	4/4
IID Permutation Tests	4/4	4/4	4/4	4/4
Min-Entropy Estimate	0.994–0.996	0.994–0.995	0.994–0.996	0.995–0.996
	Input: Each 1M bits of 2 TRNGs from 1 chip			

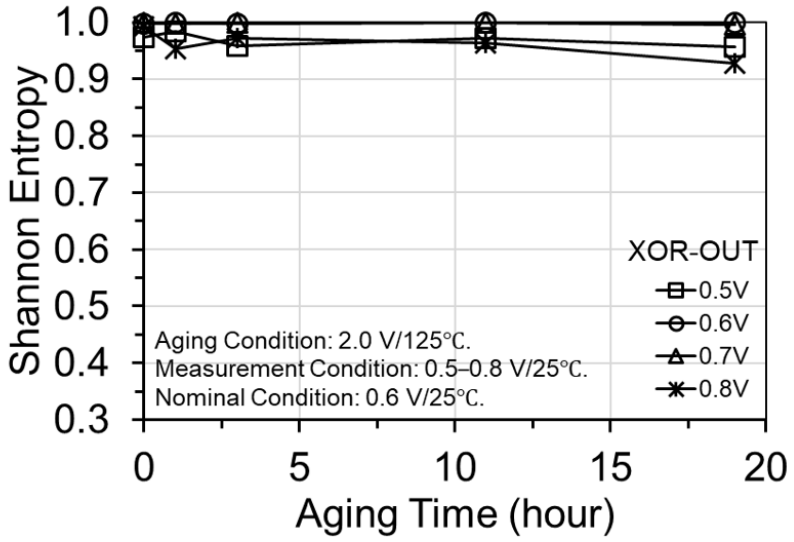
after 19 hours of aging. Shannon entropy of XOR-OUTs are always larger than 0.9, indicating an equivalent life of 11 years under 0.6 V/25°C.

4.7.6 Comparisons

The comparison with prior arts is shown in Table 4.8. The proposed TRNG occupies a comparable small core area and can operate across a wide temperature and voltage



(a)



(b)

FIGURE 4.27: Shannon entropy versus aging time for (a) single entropy source output: ES-OUT; (b) 4-bit XOR output: XOR-OUT.

range. Compared with previous designs in [42], [23]and [12], the TRNG's throughput is not high but it achieves the lowest energy consumption of 0.186 pJ/bit. Meanwhile, the robustness against power noise injection attack is verified. Furthermore, an accelerated aging test demonstrated an equivalent 11-year life of the TRNG.

TABLE 4.8: Comparison with prior works

	JSSC'2017 [24]	JSSC'2016 [41]	ISSCC'2021 [42]	SSCL'2018 [23]	JSSC'2019 [12]	**This work**
Technology (Feature Size F)	180-nm	40-nm, 180-nm	28-nm	65-nm	14-nm	**130-nm**
Entropy Source	Chaotic System	Jitter	BL Leakage Noise Jitter	Metastability	Metastability	**Metastability**
Structure	SAR ADC	Ring Oscillator	SRAM	Sense-amp	Latch	**Latch**
TRNG Area (μm^2) [Normalized by F ($\times 10^6$ F^2)]	4500 [0.139]	836 @ 40-nm [0.523 @ 40-nm]	12.54[a] [0.016]	10000 [2.367]	2114[b] [10.786]	**661/5561[c] [0.039/0.329]**
Calibration/ Feedback Control	No	Yes	No	Yes	Yes	**No**
Measured Voltage (V)	0.6–0.9	0.6–0.9 @ 40-nm	0.8–1.0	0.5–1.05	0.55–0.75	**0.3–1.0**
Measured Temperature (°C)	N/A	–40–120 @ 40-nm	–10–75	–20, 100	25–110	**–20–100**
Throughput (Mb/s)	0.27	0.45–2 @ 40-nm 0.18–1.08 @ 180-nm	3.6 @ 1.0 V	3.2	1480 @ 0.65 V	**0.00787 @ 0.3 V 2.39 @ 1.0 V**
Power (nW)	82	5000 @ 40-nm 3700 @ 180-nm	-	8330	3700000 @ 0.65 V	**1.47 @ 0.3 V**
Energy (pJ/bit)	0.3	11 @ 40-nm 21 @ 180-nm	9.6 @ 0.8 V	2.58	2.5 @ 0.65 V	**0.186 @ 0.3 V**
Post-Processing	4-bit XOR	No	1b VN (off-chip)	4-bit Markov IVN_7, 16-bit LFSR	Hierarchical VN (off-chip)	**4-bit XOR VN8W**
Power Attack Tolerant	N/A	Yes	N/A	N/A	Yes	**Yes**
Long-Term Reliability	N/A	N/A	N/A	N/A	N/A	**Yes 11-year**

[a] TRNG area overhead per random output stream.
[b] PUF circuit is included.
[c] TRNG core (4 ESs + 4-bit XOR) occupied 661 μm^2, VN8W occupied 4900 μm^2, total area is 5561 μm^2.

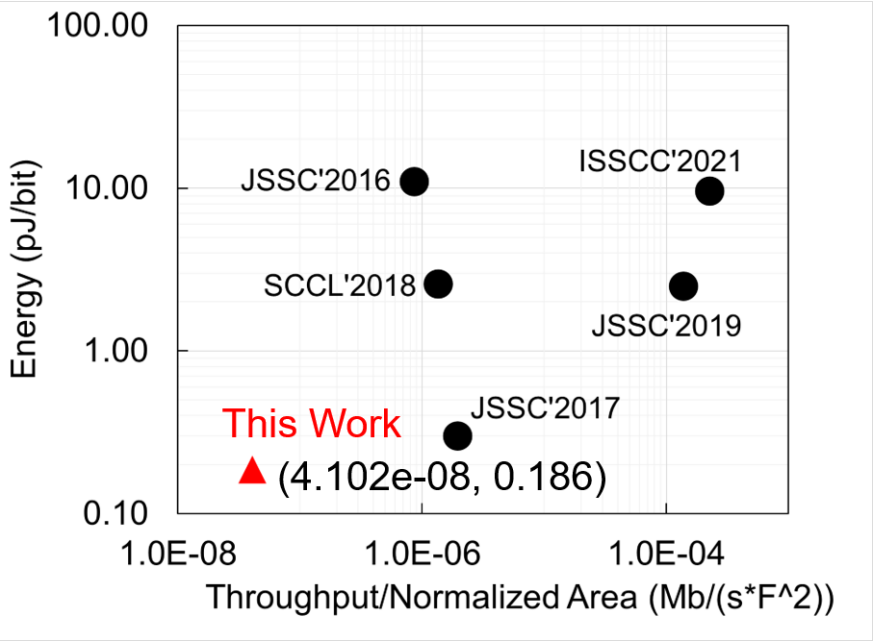


FIGURE 4.28: Energy versus throughput per area.

4.8 Discussion about Tradeoff among Energy, Throughput, and Area

On the premise of the high randomness of output data, there are several requirements for the performance of a whole TRNG. The main performance includes energy consumption, throughput, and area. According to the specific application purpose, a good tradeoff should be made. Figure 4.28 compares this work with prior arts. This work achieves the lowest energy consumption at 0.3 V. However, the throughput per normalized area is also lowest. As shown by the black solid line with a circle mark in the Figure 4.29, by simply increasing the supply voltage from 0.3 V to 1.0 V, the throughput per normalized area is improved. However, the energy consumption is also increased. To improve throughput per normalized factor with low energy consumption, there are two possible ways:

First, improving the throughput. In the TRNG core design, about 50% of cycle time is used for LR phase time to ensure the well equivalent of drain and gate voltages to avoid autocorrelation. Therefore, high throughput can be realized by reducing the LR phase

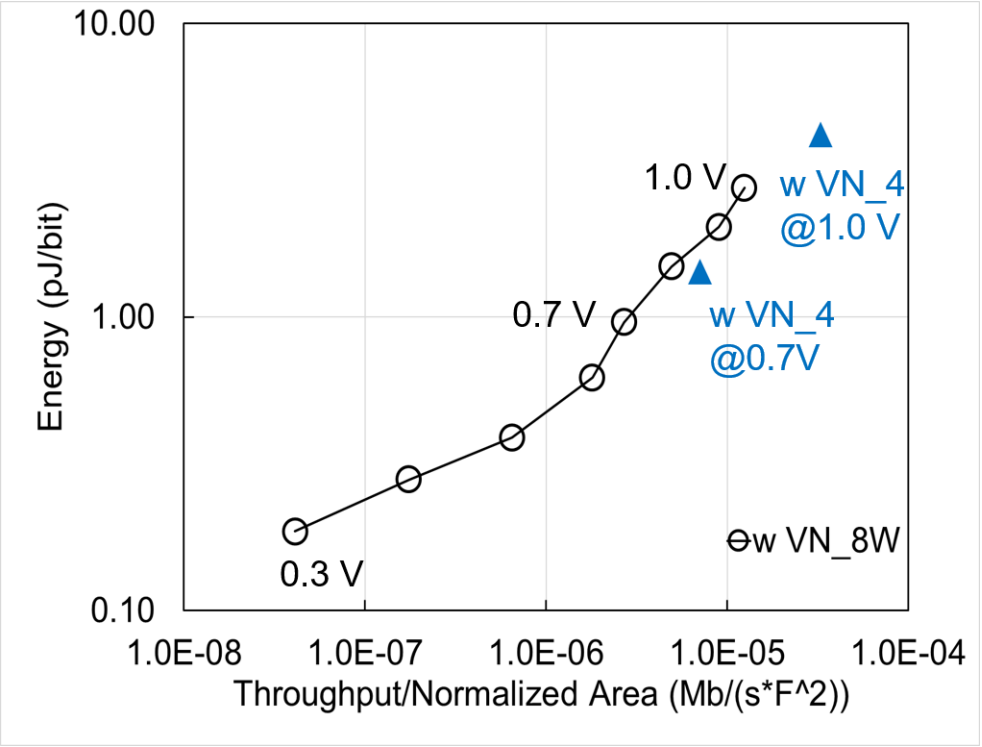


FIGURE 4.29: Energy versus throughput per area in this work under voltage variations(0.3–1.0 V).

time. The LR phase time is proportional to the product of the value of on-resistance of switch and the values of gate capacitance and parasitic drain capacitance.

From a circuit design aspect, (a) increasing the width of nMOS and pMOS pairs in switches, the on-resistance is reduced so that the LR phase time becomes shorter. (b) Reducing the value of gate capacitance, high throughput also can be achieved. However, small gate capacitance yields low randomness output due to the lower mismatch compensation efficiency. But, it can be partially compensated by improving the noise enhancement. (c) Using a post-processing circuit with a strong decorrelation ability. The LR phase time can be shortened if the correlation is tolerated in the post-processing circuit.

From a non-circuit design aspect, (a) using an advanced technology node, both on-resistance of the switch and the parasitic capacitance values are scaled down. For the identical mismatch compensation efficiency, the smaller the parasitic capacitance, the smaller the gate capacitance required. In this way, the throughput is improved. (b)

Increasing the supply voltage, the drain and gate voltages are quickly equalized so that the LR phase time is reduced. However, it is a tradeoff between the throughput and energy consumption, as shown in Figure 4.29.

Second, reducing the area. The total TRNG area consists of the TRNG core area of 661 μm^2 and the post-processing circuit (VN_8W) area of 4900 μm^2 (including the power and ground line area, while the core circuit area is 2583 μm^2). The post-processing area dominates the total area. Note that the power and ground line area in post-processing circuit is not optimal, which could be improved in future work. Therefore, the core circuit area is used for the analysis of the total area of post-processing. Therefore, by using a post-processing circuit with a small area, the total area can be reduced. For example, VN_4 has a core circuit area of 575 μm^2, which is 4.49 times smaller than the core circuit area of VN_8W. As shown in Figure 4.29, by using VN_8W (denoted by the circled points), the total TRNG achieves $1.763e{-}06$ Mb/(s*F^2) with 0.959pJ/bit at 0.7 V and $8.141e{-}06$ Mb/(s*F^2) with 2.756 pJ/bit at 1.0 V. By using VN_4 (denoted by the blue points), the total TRNG achieves $7.088e{-}06$ Mb/(s*F^2) with 1.429 pJ/bit at 0.7 V and $32.716e{-}06$ Mb/(s*F^2) with 4.183 pJ/bit at 1.0 V. Although the energy consumption is 1.49 times increased by the lower *ExE* (40.63%) using VN_4, the throughput per normalized area is improved 4.02 times.

4.9 Conclusion

In this Chapter, a latch-based TRNG core and total TRNG with VN_8W post-processing are presented. The calibration and feedback circuits in the previous work are removed from the proposed TRNG core by improving the mismatch-to-noise ratio in three ways.

First, mismatch self-compensation. It is achieved by setting the initial state point close to the metastable point by newly added gate capacitor. Compared with the fixed initial state point in the previous work, the proposed initial state point follows the metastable point, which changes position in response to mismatch variations. Considering the

tradeoff among the mismatch compensation efficiency, noise filtering effect, and energy consumption, the capacitor is set to 10 fF, which achieves 63.3% compensation efficiency.

Second, noise enhancement. Damped oscillation using larger resistor is applied for the first time. Large resistor yields large enhanced noise Vpp. At the same time, the enhanced noise is shaped by the damped oscillation process and the noise bandwidth is reduced. Considering the tradeoff between the noise Vpp and bandwidth, the resistor gate length is set to 5 μm , which achieves 3 times of noise enhancement.

Third, effective mismatch reduction by XORing entropy source latch circuits. Stochastic calculations combined with measurement results demonstrate the TRNG core has 6σ robustness against random mismatch variations with only 4 entropy source latch circuits. This is 1/64 times smaller than the conventional work with 256 latches.

The total TRNG consisting of the latch-based TRNG core and VN_8W is fabricated in 130-nm CMOS. The total TRNG occupies a TRNG core area of 661 μm^2 and a total area of 5561 μm^2 including of VN_8W. It operates across a wide voltage (0.3–1.0 V) and temperature (–20–100°C) range. Cryptographic-grade high randomness is verified by NIST SP 800-22 and NIST SP 800-90B IID tests. Power noise injection attacks result reveals the robustness of TRNG. An accelerated aging test demonstrates that the TRNG has the long-term reliability of an equivalent 11 year life. The proposed TRNG achieves the state-of-the-art low energy of 0.186 pJ/bit at 0.3 V with high randomness and robustness, suitable for energy-constrained IoT devices.

As for other performance of the TRNG, such as throughput per normalized area, it can be improved using small area post-processing such as VN_4.

Chapter 5

Conclusions

5.1 Conclusions

In this dissertation, a low energy TRNG for hardware security is presented. It consists of an energy-efficient von Neumann based post-processing technique and a latch-based TRNG core featuring mismatch self-compensation and random noise enhancement. Based on 130-nm CMOS implementation, it operates across a wide voltage (0.3–1.0 V) and temperature (–20–100°C) range without any calibration circuit. The randomness is verified by NIST SP 800-22 and NIST SP 800-90B IID tests. It achieves the state-of-art energy of 0.186 pJ/bit at 0.3 V, suitable for energy-constrained IoT devices.

In Chapter 1, as an introduction, the hardware security in the IoT era is introduced. The application of random number generator in hardware security is shown.

In Chapter 2, as preliminaries, the design requirements of TRNG are presented with the previous works in post-processing techniques and TRNG cores.

In Chapter 3, an energy-efficient post-processing technique having high extraction efficiency is presented. An improved N-bit von Neumann method is proposed. It solves the mapping table complexity (2^N) problem in three ways. First, at the algorithm level, a waiting strategy is proposed to relieve the *ExE* drop between the theoretical and realistic values in conventional N-bit von Neumann. High *ExE* with a small N value is achieved by using waiting strategy. For example, VN_4W achieves 46.88% *ExE*, which is 1.125x larger than conventional 6-bit VN (VN_6). VN_8W achieves 62.21% *ExE*, which approaches conventional 12-bit VN (VN_12) with 64.63% *ExE*. Second, at the architectural level, a Hamming weight mapping-based structure is proposed to reconstruct the large mapping table using smaller tables. The mapping table complexity is roughly reduced 4.5 times. In addition, the hierarchical structure also can relieve the lag-1 correlation problem. Third, at the logic level, an input-symbol-based code assignment is proposed for logic reduction.

VN_8W with 62.21% *ExE* is designed and fabricated in 130-nm CMOS. It achieves low energy of 0.18 pJ/bit at 0.45 V, 1 MHz. Compared with IVN_7 with 59.23% *ExE*, it achieves more than 20% energy reduction at same supply voltage. The conventional VN based methods only have de-bias function. VN_8W also enables the de-correlation

function, thanks to the hierarchical structure. The randomness of its post-processed bitstreams are verified by NIST SP 800-22 and NIST SP 800-90B tests.

In Chapter 4, a low energy latch-based TRNG is presented. The calibration and feedback control circuit in the previous work is removed in the proposed TRNG core. It is realized by reducing the mismatch-to-noise ratio in three ways. First, the mismatch self-compensation is achieved by setting the initial state point close to the metastable point using gate capacitance. In this way, 63.3% mismatch self-compensation is realized. Second, the noise is enhanced by a *RC* delay induced damped oscillation process. A small noise seed is enhanced 3 times using a large resister. Third, 4-bit XOR is used to combine four entropy source latch circuits to reduce the effective mismatch. More than 6σ robustness against the random mismatch variations is achieved.

The total TRNG consists of the latch-based TRNG core and VN_8W. It is fabricated in 130-nm CMOS with a core area of 661 μm^2 and a total area of 5561 μm^2 including VN_8W. It operates across a wide voltage (0.3–1.0 V) and temperature (–20°C–100°C) range. Furthermore, it has robustness against power noise injection attacks. An accelerated aging test shows that the TRNG has an equivalent 11-year life. NIST SP 800-22 and 800-90B IID tests verified cryptographic-grade randomness of the TRNG. It achieves low energy of 0.186 pJ/bit with high randomness and robustness, suitable for energy-constrained IoT devices.

For wider application fields, another important performance metric: throughput per area is discussed including supply voltage dependence. One solution is using a small area with medium *ExE* post-processing circuit such as VN_4.

Bibliography

[1] A. Mosenia and N. K. Jha, "A comprehensive study of security of internet-of-things," *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 4, pp.586–602, 2016.

[2] M. Alioto, "Trends in hardware security: From basics to asics," *IEEE Solid-State Circuits Magazine*, vol. 11, no. 3, pp. 56–74, 2019.

[3] I. Verbauwhede, "Security adds an extra dimension to ic design: Future ic design must focus on security in addition to low power and energy," *IEEE Solid-State Circuits Magazine*, vol. 9, no. 4, pp. 41–45, 2017.

[4] E. Aerabi et al., "Design space exploration for ultra-low-energy and secure IoT MCUs," *ACM Transactions on Embedded Computing Systems (TECS)* vol. 19, no. 3, May 2020.

[5] M. Kim et al., "An analysis of energy consumption under various memory mappings for FRAM-based IoT devices," in *IEEE World Forum on Internet of Things (WF-IoT)*, Feb. 2018, pp.574-579.

[6] N. Shafiee et al., "Infrastructure circuits for lifetime improvement of ultra-low power IoT devices," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 9, pp. 2598–2610, Sept. 2017.

[7] J. Henkel et al., "Ultra-low power and dependability for IoT devices, " in *IEEE Design, Automation & Test in Europe (DATE)*, Mar. 2017, pp. 954-959.

[8] V. M. van Santen et al., "Aging-aware voltage scaling," in *IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Mar. 2016, pp. 576-581.

[9] Johnston, David. Random Number Generators—Principles and Practices: A Guide for Engineers and Programmers, Berlin, Boston: De—G Press, 2018.

[10] National Institute of Standards and Technologies (NIST). (2015). *NIST SP 800-90A Rev.1: recommendation for random number generation using deterministic random bit generators*. [Online]. Available: https://csrc.nist.gov/publications/detail/sp/800-90a/rev-1/final

[11] V. Fischer, "A closer look at security in random number generators design," in *International workshop on constructive side-channel analysis and secure design*. Springer, 2012, pp. 167–182.

[12] S. Satpathy et al., "An all-digital unified physically unclonable function and true random number generator featuring self-calibrating hierarchical von Neumann extraction in 14-nm Tri-gate CMOS," *IEEE J. of Solid-State Circuits*, vol. 54, no. 4, pp. 1074-1085, Apr. 2019.

[13] National Institute of Standards and Technologies (NIST). (2010). *NIST SP 800-22: Download Documents and Software*. [Online]. Available: https://csrc.nist.gov/Projects/Random-Bit-Generation/Documentation-and-Software

[14] National Institute of Standards and Technologies (NIST). (2018). *NIST SP 800-90B: Recommendation for the Entropy Sources Used for Random Bit Generation*. [Online]. Available: https://csrc.nist.gov/publications/detail/ sp/800-90b/final

[15] R. Maes, V. Rozic, I. Verbauwhede, P. Koeberl, E. Van der Sluis, and V. van der Leest, "Experimental evaluation of physically unclonable functions in 65 nm cmos," in *Proc. IEEE ESSCIRC*, Sep. 2012, pp. 486–489.

[16] K. Liu, X. Chen, H. Pu, and H. Shinohara, "A 0.5-V Hybrid SRAM Physically Unclonable Function Using Hot Carrier Injection Burn-In for Stability Reinforcement," *IEEE J. of Solid-State Circuits*, vol. 56, no. 7, pp. 2193-2204, July 2021.

[17] A. T. Markettos and S. W. Moore, "The frequency injection attack on ring-oscillator-based true random number generators," in *Cryptographic Hardware and Embedded Systems. (CHES)*, 2009, pp. 317–331.

[18] V. Rozic and I. Verbauwhede, "Hardware-efficient post-processing architectures ´ for true random number generators," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 7, pp. 1242–1246, 2018.

[19] S. K. Mathew et al., "μRNG: A 300-950 mV, 323 Gbps/W all-digital full-entropy true random number generator in 14 nm FinFET CMOS," *IEEE J. of Solid-State Circuits*, vol. 51, no. 7, pp. 1695-1704, July 2016.

[20] V. B. Suresh and W. P. Burleson, "Entropy and energy bounds for metastability based trng with lightweight post-processing," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 7, pp. 1785–1793, 2015.

[21] P. A. Samuelson, "Constructing an unbiased random sequence," *Journal of the American Statistical Association*, vol. 63, no. 324, pp. 1526–1527, 1968.

[22] A. T. Do and X. Liu, "25 fj/bit, 5mb/s, 0.3 v true random number generator with capacitively-coupled chaos system and dual-edge sampling scheme," in *IEEE Asian Solid-State Circuits Conference (A-SSCC)*, Nov. 2017, pp. 61–64.

[23] V. R. Pamula, X. Sun, S. M. Kim, F. u. Rahman, B. Zhang and V. S. Sathe, "A 65-nm CMOS 3.2-to-86 Mb/s 2.58 pJ/bit highly digital true-random-number generator with integrated de-correlation and bias correction," *IEEE Solid-State Circuits Letters*, vol. 1, no. 12, pp. 237-240, Dec. 2018.

[24] M. Kim, U. Ha, K. J. Lee, Y. Lee, and H. Yoo, "A 82-nW chaotic map true random number generator based on a sub-ranging SAR ADC," *IEEE J. of Solid State Circuits*, vol. 52, no. 7, pp. 1953-1965, July 2017.

[25] P. Lacharme, "Post-processing functions for a biased physical random number generator," in *International Workshop on Fast Software Encryption*. Springer, 2008, pp. 334–342.

[26] M. Dichtl, "Bad and good ways of post-processing biased physical random numbers," in *International Workshop on Fast Software Encryption*. Springer, 2007, pp. 137–152.

[27] S.-H. Kwok, Y.-L. Ee, G. Chew, K. Zheng, K. Khoo, and C.-H. Tan, "A comparison of post-processing techniques for biased random number generators," in *International Workshop on Information Security Theory and Practices (IFIP)*. Springer, 2011, pp. 175–190.

[28] J. Von Neumann, "Various techniques used in connection with random digits," *Collected Works*, vol. 5, pp. 768–770, 1963.

[29] Y. Peres, "Iterating von neumann's procedure for extracting random bits," *The Annals of Statistics*, pp. 590–597, 1992.

[30] P. Elias, "The efficient construction of an unbiased random sequence," *The Annals of Mathematical Statistics*, vol. 43, no. 3, pp. 865–870, 1972.

[31] V. Rozic, B. Yang, W. Dehaene, and I. Verbauwhede, "Iterating von neumann's ´ post-processing under hardware constraints," in *IEEE international symposium on hardware oriented security and trust (HOST)*, May 2016, pp.37–42.

[32] R. Brederlow, R. Prakash, C. Paulus, and R. Thewes, "A low-power true random number generator using random telegraph noise of single oxide-traps," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2006, pp. 1666-1675.

[33] S. Fujita, K. Uchida, S. Yasuda, R. Ohba, H. Nozaki, and T. Tanamoto, "Si nanodevices for random number generating circuits for cryptographic security," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2004, pp. 294-295.

[34] M. Matsumoto, S. Yasuda, R. Ohba, K. Ikegami, T. Tanamoto, and S. Fujita, "1200μm^2 Physical Random-Number Generators Based on SiN MOSFET for Secure Smart-Card Application," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2008, pp. 414-624.

[35] N. Liu, N. Pinckney, S. Hanson, D. Sylvester, and D. Blaauw, "A true random number generator using time-dependent dielectric breakdown," in *IEEE Symp. VLSI Circuits Dig. of Tech. Papers*, June 2011, pp. 216-217.

[36] W. T. Holman, J. A. Connelly, and A. B. Dowlatabadi, "An integrated analog/digital random noise source," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 44, no. 6, pp. 521–528, 1997.

[37] W. Penzhorn, "The design of a truly random monolithic noise generator," *Microelectronics Journal*, vol. 15, no. 4, pp. 29–40, 1984.

[38] C. S. Petrie and J. A. Connelly, "A noise-based ic random number generator for applications in cryptography," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 5, pp. 615–621, May 2000.

[39] X. Wang, H. Liu, R. Zhang, K. Liu, and H. Shinohara, "An inverter-based true random number generator with 4-Bit von-Neumann post-processing circuit," in *Proc. IEEE Int. Midwest Symp. on Circuits & Systems. (MWSCAS)*, pp. 285-288, Aug. 2020.

[40] 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 smart card ic," *IEEE transactions on computers*, vol. 52, no. 4, pp. 403–409, 2003.

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

[42] S. Taneja, V. K. Rajanna, and M. Alioto,"36.1 Unified In-Memory Dynamic TRNG and Multi-Bit Static PUF Entropy Generation for Ubiquitous Hardware Security," in *IEEE Int. Solid-State Circuits Conf. (ISSCC)*, Feb. 2021, pp. 498-500.

[43] S. Callegari, R. Rovatti, and G. Setti, "Embeddable adc-based true random number generator for cryptographic applications exploiting nonlinear signal processing

and chaos," *IEEE transactions on signal processing*, vol. 53, no. 2, pp. 793–805, Feb. 2005.

[44] F. Pareschi, G. Setti, and R. Rovatti, "Implementation and testing of high-speed cmos true random number generators based on chaotic systems," *IEEE transactions on circuits and systems I: regular papers*, vol. 57, no. 12, pp. 3124–3137, Dec. 2010.

[45] D. J. Kinniment and E. G. Chester, "Design of an on-chip random number generator using metastability," in *Proc. ESSCIRC*, Sept. 2002, pp. 595–598.

[46] C. Tokunaga, D. Blaauw, and T. Mudge, "True random number generator with a metastability-based quality control," *IEEE J. of Solid State Circuits*, vol. 43, no. 1, pp. 78-85, Jan. 2008.

[47] S. K. Mathew et al., "2.4 Gbps, 7 mW All-Digital PVT-Variation Tolerant True Random Number Generator for 45 nm CMOS High-Performance Microprocessors," *IEEE J. of Solid-State Circuits*, vol. 47, no. 11, pp. 2807-2821, Nov. 2012.

[48] N. Torii et al., "ASIC implementation of random number generators using SR latches and its evaluation," *EURASIP J. on Info. Security*, no. 10, May 2016.

[49] R. Zhang, X. Wang, L. Wang, X. Chen, F. Yang, K. Liu, and H. Shinohara, "A 0.186 pJ per bit Latch-based true random number generator with mismatch compensation and random noise enhancement," in *IEEE Symp. VLSI Circuits*, June 2021, pp. 1-2.

[50] H. Shinohara, B. Zheng, Y. piao, B. Liu, and S. Liu, "Analysis and reduction of SRAM PUF Bit Error Rate," in *Proc. IEEE Int. Symp. on VLSI Design, Automat. and Test. (VLSI-DAT)*, Apr. 2017, pp. 1-4.

[51] R. Zhang, S. Chen, C. Wang, and H. Shinohara, "High-throughput von Neumann post-processing for random number generator," in *Proc. IEEE Int. Symp. on VLSI Design, Automat. and Test. (VLSI-DAT)*, Apr. 2018, pp. 1-4.

[52] R. Zhang, X. Wang, and H. Shinohara, "Energy-Efficient Post-Processing Technique Having High Extraction Efficiency for True Random Number Generators," *IEICE Trans. Electro.*, vol. E104.C, no. 7, pp. 300-308, July 2021.

[53] R. Zhang, X. Wang, K. Liu, and H. Shinohara, "A 0.186-pJ per Bit Latch-Based True Random Number Generator Featuring Mismatch Compensation and Random Noise Enhancement," *IEEE Journal of Solid-State Circuits*, accepted on December 16th, 2021.

[54] MathWorks, "*arima*," [Online]. Available: https://www.mathworks.com/help/econ/arima.html

[55] J. R. Moore and D. A. Maguire, "Natural sway frequencies and damping ratios of trees: concepts, review and synthesis of previous studies," *Springer Trees - Structure and Function*, vol. 18, no. 2, pp. 195-203, 2004.

[56] R. J. Parker, "Entropy justification for metastability based nondeterministic random bit generator," in *IEEE International Verification and Security Workshop (IVSW)*, 2017, pp. 25–30.

Publications

Journal Papers

[1] Ruilin Zhang, Xingyu Wang, Kunyang Liu, and Hirofumi Shinohara, "A 0.186-pJ per Bit Latch-Based True Random Number Generator Featuring Mismatch Compensation and Random Noise Enhancement," *IEEE Journal of Solid-State Circuits*, doi: 10.1109/JSSC.2021.3137312.

[2] Ruilin Zhang, Xingyu Wang, and Hirofumi Shinohara, "Energy-Efficient Post-Processing Technique Having High Extraction Efficiency for True Random Number Generators," *IEICE Transactions on Electronics*, vol. E104.C, no. 7, pp. 300-308, July 2021.

International Conference Papers

[1] Ruilin Zhang, Xingyu Wang, Luying Wang, Xinpeng Chen, Fan Yang, Kunyang Liu, and Hirofumi Shinohara, "0.186 pJ per bit Latch-based true random number generator with mismatch compensation and random noise enhancement," in *Proc. IEEE Symposium on VLSI Circuits (VLSI)*, June 2021, pp. 1-2.

[2] Xingyu Wang, Hongjie Liu, Ruilin Zhang, Kunyang Liu, and Hirofumi Shinohara, "An inverter-based true random number generator with 4-Bit von-Neumann post-processing circuit," in *Proc. IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, Aug. 2020, pp. 285-288.

[3] Ruilin Zhang and Hirofumi Shinohara, "High-throughput & power efficiency 8 bits von Neumann post-processing with waiting strategy for true random number generators," in *Proc. Taiwan and Japan Conference on Circuits and Systems (TJCAS)*, Aug. 2019.

[4] Ruilin Zhang, Sijia Chen, Chao Wan, and Hirofumi Shinohara, "High-throughput von Neumann post-processing for random number generator," in *Proc. IEEE International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, Apr. 2018, pp. 1-4.

Reports

[1] Ruilin Zhang and Hirofumi Shinohara, “Robustness Improvement by XORing Multiple Entropy Sources for True Random Number Generator”, in *Proc. IEICE HWS/ICD*, Japan, Oct. 2021.

[2] Ruilin Zhang and Hirofumi Shinohara, “Energy-efficient 8 Bits Von Neumann with Waiting Strategy Post-Processing technique for True Random Number Generators”, in *Proc. International collaboration Symposium on Information, Production and Systems (ISIPS)*, Japan, Nov. 2020.

[3] Ruilin Zhang and Hirofumi Shinohara, “High Output Rate 8 Bits Von Neumann with Waiting Post-Processing for True Random Number Generators”, in *Proc. International collaboration Symposium on Information, Production and Systems (ISIPS)*, Japan, Nov. 2019.

[4] Ruilin Zhang and Takeshi Yoshimura, “Thermal-aware Floorplan Design based on 3D NoC for MPSoC in Dark Silicon Era”, in *Proc. International collaboration Symposium on Information, Production and Systems (ISIPS)*, Japan, Nov. 2017.

Acknowledgements

During my five years of doctor course in Graduate School of IPS, Waseda University, I highly appreciate all the people I have met there.

First, I would like to express my deepest appreciation to my supervisor Professor Hirofumi Shinohara. He takes me into the world of silicon chips. He teaches me to find clues by observing the raw data. His engineering spirit always inspires me to find better solutions. Without his help, I cannot achieve what I have achieved. I feel very fortunate and proud to be his student.

Second, I would like to thank Professor Toshihiko Yoshimasu and Professor Takashi Ohsawa. I highly appreciate their enlightening advice and guidance, which help me a lot to improve my research and dissertation.

Third, I would like to thank Emeritus Professor Takahiro Watanabe, Professor Shinji Kimura, Professor Takeshi Ikenaga, Professor Shoji Makino, and all the other Professors at IPS for their comments and reviews, which help me a lot to improve my dissertation and presentation.

I would like to thank Emeritus Professor Takeshi Yoshimura. He guided my master course in Graduate School of IPS, Waseda University. I highly appreciate his teaching and recommendation to become a student of Professor Hirofumi Shinohara.

I would also like to thank Mr. Koichi Takanashi from ROHM Company Ltd. I am grateful for his technical discussions and his help to fabricate our circuits.

I would like to thank all the members in our laboratory, who helped me a lot during my research. Especially, I would like to thank the senior alumni Dr. Jing Wang, Dr. Kunyang Liu, Dr. course student Xingyu Wang, all the co-authors of the related publications Sijia Chen, Chao Wan, Hongjie Liu, Luying Wang, Xinpeng Chen, Fan Yang, and all the senior students Ning Wan, Hanjun Ying, Donglong Jiang, and Pan Shu.

Last but not least, I must thank all my family members, for their support and love throughout my whole life.