De-Correlation and De-Bias Post-Processing Circuits for True Random Number Generator

Ruilin Zhang, *Member, IEEE*, Haochen Zhang, Xingyu Wang, *Member, IEEE*, Ye Ziyang, *Graduate Student Member, IEEE*, Kunyang Liu, *Member, IEEE*, Shinichi Nishizawa, *Member, IEEE*, Kiichi Niitsu, *Member, IEEE*, and Hirofumi Shinohara, *Member, IEEE*

***Abstract*— True random number generators (TRNGs) are commonly used in hardware security for secure authentication, data encryption, etc. The raw random numbers often exhibit defects. The most commonly observed defects are bias and correlations. Post processing techniques have been developed to address them. The von Neumann method addresses bias, but it requires input that is uncorrelated and has an identical distribution. On the other hand, the Markov chain can address correlation but introduce bias. In this work, we research the lightweight combination of two techniques. We verified that MKV2(QL4)/VN2 performs well for both Markov and non-Markov model bitstreams. MKV1(QL8)/VN8W is effective for the Markov model. The randomness is verified by NIST SP 800-22 and 800-90B, and ENT, respectively. Both of these circuits require only 16 bits of memory, which is 12 times smaller than in previous work. MKV1(QL8)/VN8W is implemented using 65-nm CMOS. A prototype chip demonstrates a minimum energy consumption of 0.149 pJ/bit at 0.45V. When applied to a latch-based TRNG, it can double the operation frequency thanks to the enhanced decorrelation. The total energy consumption is reduced by 21%.**

***Index Terms*— De-autocorrelation, de-bias, Markov chain, von Neumann post-processing, true random number generator, hardware security.**

I. INTRODUCTION

A TRUE random number generator (TRNG) that harvests physical noise to generate true random and unpredictable numbers is a fundamental component of hardware security.

Manuscript received 29 April 2024; revised 14 June 2024; accepted 23 June 2024. This work was supported in part by the Asahi Kohsan Group; in part by the Kitakyushu Foundation for the Advancement of Industry, Science and Technology (FAIS); in part by Grant 23SGB06X; in part by Grant JPMJMS2214; in part by JST [Moonshot Research and Development] under Grant PMJMS2214-5; in part by Grant NICT JPJ012368C06201; in part by Grant JST JPMJPR2034; in part by Grant JPNP14004; in part by the New Energy and Industrial Technology Development Organization (NEDO); in part by Grant JSPS 22H03557; and in part by the activities of VDEC, The University of Tokyo, in Collaboration with Cadence Design Systems and NIHON SYNOPSYS G.K. This article was recommended by Associate Editor Y. Lao. *(Corresponding author: Ruilin Zhang.)*

Ruilin Zhang, Kunyang Liu, Kiichi Niitsu, and Hirofumi Shinohara are with the Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan (e-mail: zhang.ruilin.5w@kyoto-u.ac.jp).

Haochen Zhang is with Lenovo, Chengdu 610045, China.

Xingyu Wang and Shinichi Nishizawa are with the Graduate School of Information, Production and Systems, Waseda University, Kitakyushu 808-0135, Japan.

Ye Ziyang is with the Department of Electrical Engineering and Information Systems, Graduate School of Engineering, The University of Tokyo, Tokyo 113-8656, Japan.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSI.2024.3421663.

Digital Object Identifier 10.1109/TCSI.2024.3421663

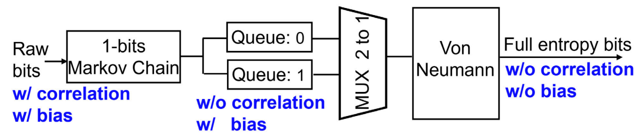


Fig. 1. Diagram of de-correlation and de-bias by Markov chain and von Neumann post-processing.

These random numbers are utilized as keys or nonces in secure operations, such as communication protocols and authentication in Internet of Things (IoT) devices, smart cards, etc. The raw data from TRNGs often exhibit statistical defects due to processes, voltage variations, temperature fluctuations, clock period variations, or intentional attacks [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]. The most commonly observed defects are bias and auto-correlation (AC). Bias indicates the deviation from 0.5, while AC indicates the correlations between bits. Both of these issues can be addressed by post-processing techniques, which can be divided into two groups: cryptographic algorithm-based methods and arithmetic-based methods.

Cryptographic algorithm-based post-processing techniques, such as hash functions, AES, CBC-MAC, CMAC, and HMAC, etc., are typically used in a cryptographic secure random number generator [5]. These techniques require a minimum entropy guarantee of the entropy source and can extract full randomness per bit. However, these techniques come with energy and hardware costs, making them unsuitable for resource-limited IoT devices. To provide a low-cost alternative, mathematical constructs such as finite field addition, matrices, substitution, and permutation networks have been employed. Examples include BIW [3], strong blenders [15], and the PRESENT cipher [16], etc. However, these methods still require a guarantee of minimum entropy in the raw data, necessitating additional circuits for entropy monitoring.

On the other hand, arithmetic methods provide a more lightweight solution. According to the extraction efficiency (ExE, which is defined as the output bitstream length over the input bitstream length), these methods can be further divided into three groups: 1. ExE less than one, such as the XOR and von Neumann methods; 2. ExE equal to one, such as Linear Feedback Shift Registers (LFSR) and Markov chains; 3. ExE greater than one, such as the middle square method. Among the above-mentioned arithmetic methods, XOR with 50% ExE is the simplest technique. However, bias can never be fully eliminated. LFSRs are often used for whitening the bitstream.

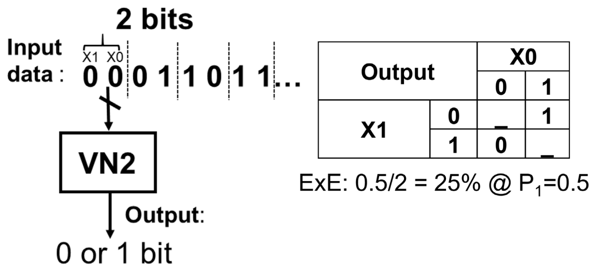


Fig. 2. Diagram of von Neumann (VN2) post-processing.

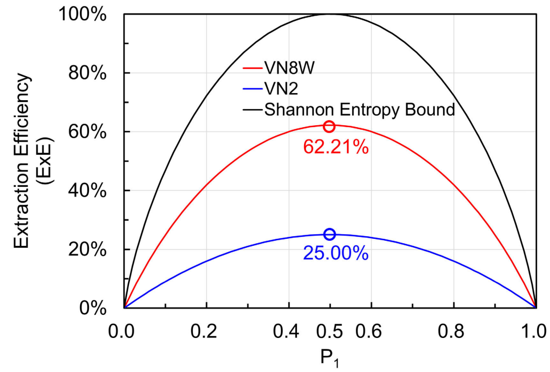


Fig. 3. Extraction efficiency for VN2 and VN8W.

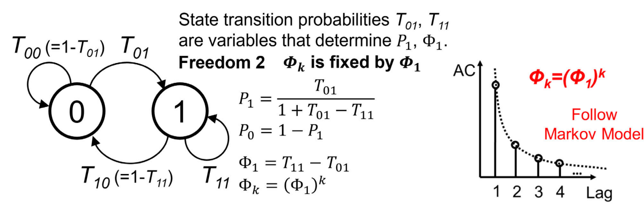


Fig. 4. Diagram of MKV1 random bitstream generation and autocorrelation of generated random bitstream. (Source: [20] modified).

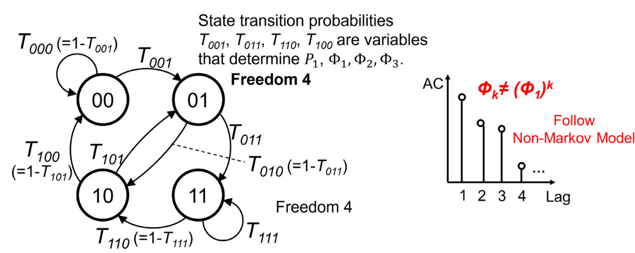


Fig. 5. Diagram of MKV2 and autocorrelation based on MKV2 model. (Source: [20] modified).

Even though the apparent entropy is increased, the true entropy per bit cannot be improved since no bits are discarded during processing. The middle square method, once a pseudo-random number generator, is used as a post-processing circuit by combining a 16-bit shift register to feed the raw TRNG's bitstream [14]. However, in that work, 8 bits are generated with only one raw bit. The raw bits with bias and correlations are XORed with the bits in the last square stage to obtain the final output. This presents a risk for attacks.

The von Neumann (VN) [17] can achieve zero bias without a minimum entropy guarantee. However, its ExE decreases with input bias. While this property can serve as an entropy detector to indicate the current bias condition in the raw bits. It leads to increased energy consumption due to the reduced ExE. To improve ExE, iterative von Neumann (IVN) [18] methods, and N-bit von Neumann [19] methods have been proposed. However, all these methods cannot solve the correlation problem. 8-bit von Neumann with waiting (VN8W) [19] can tolerate less than 0.03 lag 1 correlation (AC lag1) at input. However, the application is limited for low correlations. Based on our measurement results on latch-based TRNG [11], if the equalization time, which is the setting time to initialize the latch internal node to metastable voltage, is insufficient, the correlation can exceed 0.03, reaching values as high as 0.3.

The N-bits Markov chain works for decorrelation by separating the input bits into 2^N memory queues based on the current N-bits. The bits within the same queue are assumed to be independent of each other. However, bias is added to each queue's bits. Combining the Markov chain for decorrelation and von Neumann methods for debiasing presents a promising solution. Research in [4] proposes a combination of a 4-bit Markov chain and IVN structure. However, it requires significant memory of 192 bits (12 bits in each queue by 16 states) to store the decorrelated bitstreams before sending them to IVN. Additionally, a 16-bit LFSR is still necessary for removing residual correlation after IVN processing.

In this study, we explore a lightweight approach to combining a Markov chain and von Neumann. We extend the theoretical analysis of the Markov model for generating correlated bitstreams presented in [20]. We verify that even with 1-bit MKV (MKV1) and 2-bit (MKV2), they work efficiently: MKV2(QL4)/VN2 performs well for both Markov and non-Markov model bitstreams. MKV1(QL8)/VN8W is effective for the Markov model. The randomness is verified by NIST SP 800-22 [21], 800-90B [22], and ENT [23], respectively. Both of these approaches require only 16 bits of memory, which is 12 times smaller than in previous work. MKV1(QL8)/VN8W, as illustrated in Fig. 1, is implemented using 65-nm CMOS. When applied to a latch-based TRNG, it can double the operation frequency thanks to the enhanced decorrelation. The total energy consumption is reduced by 21%.

The following sections are organized as follows: Section II reviews the basic concepts of autocorrelation, the Markov model, and von Neumann post-processing. Section III presents the theoretical analysis of MKV1 and MKV2 for generating both Markov and non-Markov model-based bitstreams. Section IV introduces the usage of the MKV model for decorrelation. Section V explores the combination of the MKV model and the von Neumann method. Chip measurement results are presented in Section VI, followed by the conclusion in Section VII. The details of relationship between transition probabilities and generated bitstream in MKV2 are provided in the Appendix.

II. Review of Autocorrelation, Markov Model and Von Neumann

This section reviews the fundamental knowledge of autocorrelation and Markov model. The information presented in

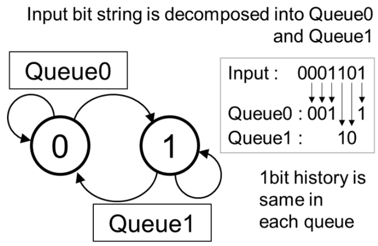


Fig. 6. Decorrelation by MKV1. (Source: [20] modified).

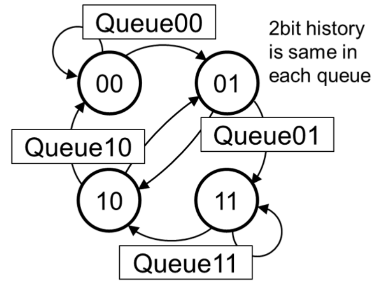


Fig. 7. Decorrelation by MKV2.

this section will support the analysis conducted in subsequent sections.

A. Autocorrelation

Given a random n-bit bitstream X, the autocorrelation at lag k, denoted by ϕ_k, characterizes the degree of correlation between bits at positions t and $t+k$, where t ranges from 1 to $n-k$. ϕ_k is calculated as:

$$\phi_k = \frac{\mathrm{Cov}(X_t, X_{t+k})}{\sigma_X^2} = \frac{\sum_{t=1}^{t=n-k} (X_t - \mu)(X_{t+k} - \mu)}{\sum_{t=1}^{t=n} (X_t - \mu)^2}, \quad (1)$$

where μ represents the mean value of X. In particular, for $k = 1$, the autocorrelation ϕ_1 can be computed utilizing Pearson's autocorrelation formula:

$$\phi_1 = \frac{n_{11}n_{00} - n_{10}n_{01}}{n_1 n_0}, \quad (2)$$

where n_{11}, n_{00}, n_{10}, and n_{01} denote the frequencies of 1-bit overlap patterns of 11, 00, 10, and 01, respectively. n_1 and n_0 represent the total counts of 1s and 0s within the bitstream, respectively. For an n-bit bitstream, the acceptable correlation level falls within the 95% confidence interval (CI), which is approximately $\pm 1.96/\sqrt{n}$.

B. Markov Model

A Markov model is a memoryless model where the future state of a system is determined solely by its current state. The most common type of Markov model is the Discrete-Time Markov chain, where the state space and time are discrete. A Markov chain consists of the following components: state space, transition probability matrix, and initial state distribution. The state space is a finite set of possible states that the system can be in. Each state is represented by binary bits, resulting in 2^N possible states, where N denotes the number of binary bits. For example, in a 1-bit Markov chain, there are 2 states. The transition probability matrix specifies the probabilities of transitioning from one state to another. The initial state distribution is a probability distribution that specifies the initial state of the system. Markov chains provide a flexible framework for generating bit sequences with specific correlation. Further details will be presented in the next Section.

C. Von Neumann

The diagram of von Neumann (VN) [17] is shown in Fig. 2. It processes each pair of input bits and retains only the first bit when the two input bits are different. It can achieve zero bias when input follows independent, identically, distributed. The ExE is defined as the output bit length over the input bit length. As shown in Fig. 3, when the probability of ones P_1 is 0.5, ExE = 25%, and 75% of bits are discarded. To improve ExE, iterative von Neumann (IVN) [18] methods, which reuse discarded bits, and N-bit von Neumann [19] methods, which process more than two bits simultaneously, have been proposed. For example, in the work [19], the author proposed 8-bit von Neumann with waiting strategy (VN8W) achieves 62.21% ExE when $P_1 = 0.5$, as shown in Fig. 3. By separating the 8-bit into odd and even numbers and processing separately, VN8W can tolerate less than 0.03 lag1 correlation.

III. Bitstream Generation by Markov Chain

A. Bitstream Generation by MKV1

The diagram of 1-bits Markov chain (MKV1) is illustrated in Fig. 4. Its transition probabilities from the current state i to the next state j are denoted as T_{ij}, where both i and j are values of either 0 or 1. These transitions adhere to the constraints $T_{00} + T_{01} = 1$ and $T_{10} + T_{11} = 1$. The probabilities associated with states 0 and 1 are represented by P_0 and P_1, respectively. Upon achieving stability after time t, the following relationship holds:

$$[P_0, P_1] \begin{bmatrix} T_{00} & T_{01} \\ T_{10} & T_{11} \end{bmatrix} = [P_0, P_1], \quad (3)$$

which leads to the expressions:

$$P_1 = \frac{T_{01}}{1 + T_{01} - T_{11}}, \quad P_0 = 1 - P_1. \quad (4)$$

In the scenario where $T_{01} + T_{11} = 1$, equilibrium is reached with $P_0 = P_1 = 0.5$.

Now, let's reconsider ϕ_1 as illustrated in Equation (2). For the n bits, then we have: $n11 = P_1 * T_{11} * n$, $n00 = P_0 * T_{00} * n$, $n10 = P_1 * T_{10} * n$, $n01 = P_0 * T_{01} * n$, $n1 = P_1 * n$ and $n0 = P_0 * n$. Thus, ϕ_1 can be expressed as:

$$\phi_1 = T_{11} - T_{01}, \quad (5)$$

If T_{11} equals T_{01}, ϕ_1 equals zero, indicating the absence of autocorrelation. For a general ϕ_k, it can be obtained by substituting T_{11} and T_{01} with T_{1-1} and T_{0-1} in Equation (5), where $-$ denotes all intermediate states. For instance, when $k = 2$, ϕ_2 can be calculated as,

$$\phi_2 = T_{1-1} - T_{0-1}$$

$$= (T_{111} + T_{101}) - (T_{011} + T_{001})$$
$$= (T_{11} \cdot T_{11} + T_{10} \cdot T_{01}) - (T_{01} \cdot T_{11} + T_{00} \cdot T_{01}), \quad (6)$$

substituting $T_{10} = 1 - T_{11}$, $T_{00} = 1 - T_{01}$, we get

$$\phi_2 = (T_{11} - T_{01})^2 = \phi_1^2. \quad (7)$$

Hence,

$$\phi_k = (T_{11} - T_{01})^k = \phi_1^k. \quad (8)$$

Thus, MKV1 offers two degrees of freedom through T_{01} and T_{11}. As MKV1 represents a 1-bit state machine, it can be employed to generate Markov model bitstreams with target P_1 and ϕ_1 by calculation:

$$T_{01} = P_1(1 - \phi_1), \quad (9)$$
$$T_{11} = \phi_1 + T_{01}, \quad (10)$$

therefore, new bits are generated based on the transition matrix.

B. Bitstream Generation by MKV2

The diagram of 2-bits Markov chain (MKV2) is illustrated in Fig. 5. MKV2 has four states. And the transition probabilities from the current state ij to the subsequent state jk, with a one-bit of j overlap, are represented by T_{ijk}, where i, j and k are either 0 or 1. For instance, T_{101} represents the probability of transition from state 10 to state 01.

The characteristics of P_1 and autocorrelations at lag 1, 2, 3 (ϕ_1, ϕ_2, ϕ_3) in random bitstream generated by MKV2 can be extracted from its four independent transition probabilities T_{001}, T_{011}, T_{110}, T_{100}. For simplicity, they are denoted as a, b, c, d, respectively. The equations are show in following:

$$\begin{cases} P_1 = \dfrac{1+s}{2+q+s} \\ \phi_1 = \dfrac{qs-1}{(1+q)(1+s)} \\ \phi_2 = 1 - \dfrac{2+q+s}{(1+q)(1+s)}(b+d) \\ \phi_3 = \dfrac{qs-1+(2+q+s)(b^2/s+d^2/q-2bd)}{(1+q)(1+s)}, \end{cases} \quad (11)$$

where $q = d/a$, $s = b/c$.

Random bit streams with arbitral characteristics P_1, ϕ_1, ϕ_2, ϕ_3 can be generated by applying calculated T_{001}, T_{011}, T_{110}, T_{100} to MKV2 state transition model. Inverse functions that calculate T_{001}, T_{011}, T_{110}, T_{100} from P_1, ϕ_1, ϕ_2, ϕ_3 are also clarified. The details are exhibited in Appendix.

IV. Markov De-Autocorrelation

In this section MKV1 and MKV2 are used inversely for decorrelation. Here, random bitstream with correlation is input of them, while it is output in Section III.

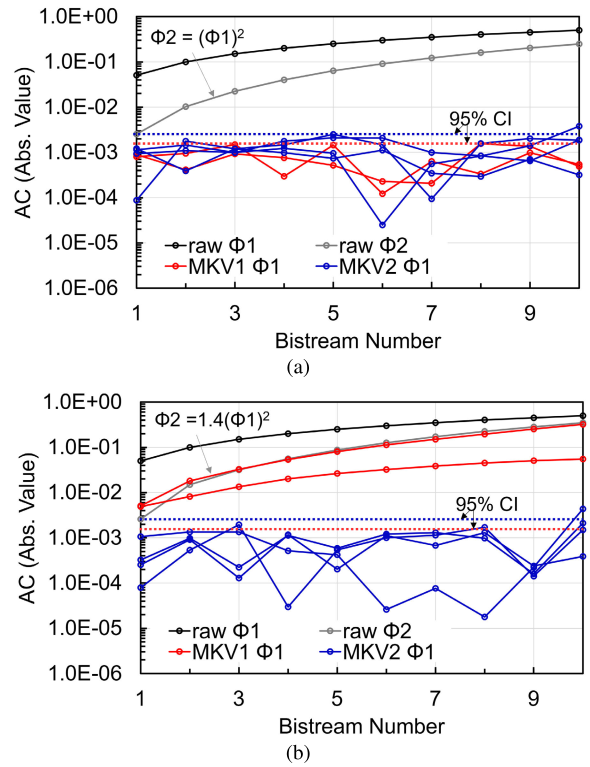


Fig. 8. Decorrealtion analysis with (a) Markov model bitstreams. (b) Non–Markov model bistream.

TABLE I
Parameters: ϕ_1 and r in 41 Pseudorandom Bitstreams

ϕ_1	r	Count
±0.05, ±0.1, ±0.15, ±0.2	1, 1.4, 2, 3	32
0.25, 0.3, 0.35	1, 1.4, 2	9

A. De-Correlation by MKV1 and MKV2

The decorrelation diagram using MKV1 is depicted in Fig. 6. The input bitstream is routed into two queues, Queue0 and Queue1, depending on the preceding bit: if the previous bit is 0, the current bit is routed into Queue0; if it's 1, it is routed into Queue1. As a result, correlation due to one bit history in the raw bitstream is eliminated.

Similarly, the diagram for decorrelation using MKV2 is shown in Fig. 7. The input bitstream is routed into four queues, Queue00 to Queue11, based on the previous 2-bit states. MKV2 is expected to handle both Markov model and non-Markov model bitstreams effectively.

B. Evaluation of Decorrelation by MKV1 and MKV2

To evaluate the decorrelation achieved by MKV1 and MKV2, pseudorandom bitstreams, each with a length of 3 million bits, are generated based on MKV2 equations shown in Appendix. $P_1 = 0.5$, the autocorrelation (AC) lag 1 ranges from 0.05 to 0.5 with increments of 0.05. The first ten bitstreams are generated with $\phi_2 = (\phi_1)^2$, representing Markov model bitstreams. The other ten bitstreams are generated with $\phi_2 = 1.4(\phi_1)^2$, where the ratio of 1.4 is multiplied, indicating non-Markov model bitstreams. The absolute values of ACs for raw data and Markov post-processed data are shown in Fig. 8.

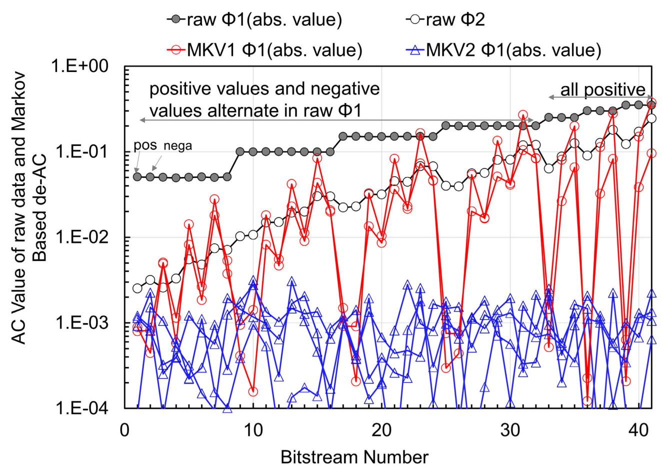


Fig. 9. Autocorrelation values for raw data and post-processed data by MKV for each queue. (Source: [20] modified).

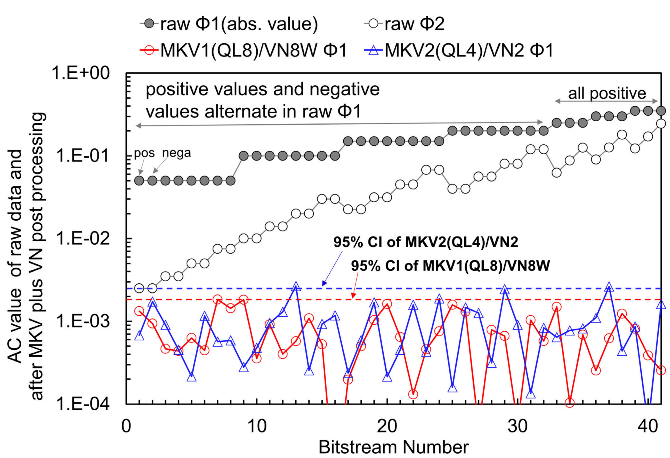


Fig. 10. Autocorrelation values for raw data and post-processed data by MKV+VN. (Source: [20] modified).

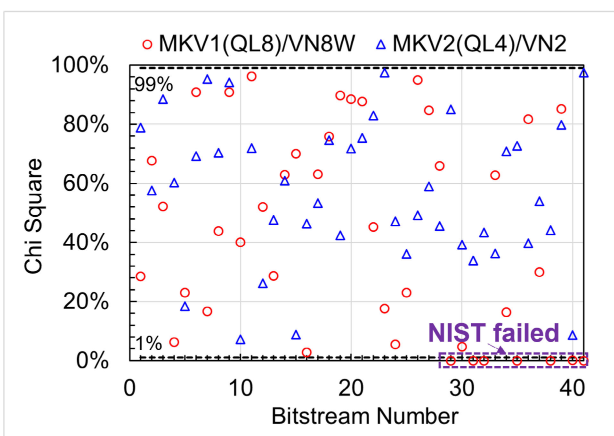


Fig. 11. Chi-square analysis using ENT.

There are two lines for MKV1, representing the bitstreams in Queue0 and Queue1, respectively. Similarly, there are four lines for MKV2, representing the bitstreams in Queue00, Queue01, Queue10, and Queue11, respectively. As observed, by MKV2, AC are removed to below 95% CI boundary (within stochastic error) in all bitstreams including non-MKV

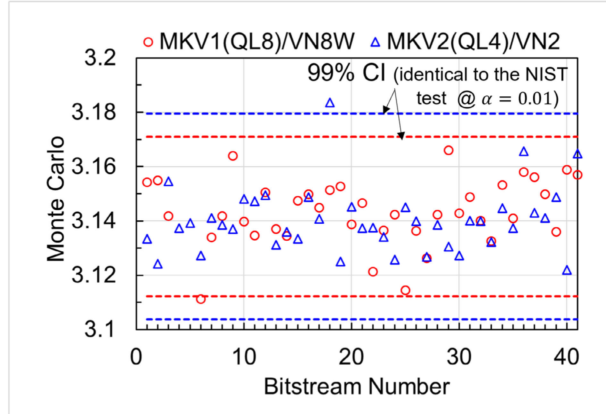


Fig. 12. Monte Carlo analysis using ENT.

TABLE II

NIST SP 800-22 Results For Random Data From Markov Chain

	MKV1(QL8) /VN8W Input: 1M*41		MKV2(QL4) /VN2 Input: 600k*41	
	Pass Rate	Ave. P-Val	Pass Rate	Ave. P-Val
Frequency	34/41[a]	0.44	41/41	0.57
Block Frequency	41/41	0.42	41/41	0.44
Cumulative Sums	34/41[a]	0.43	41/41	0.57
Runs	36/41[a]	0.39	40/41	0.50
Longest Run	40/41	0.47	40/41	0.49
Rank	41/41	0.48	41/41	0.54
FFT	41/41	0.48	41/41	0.49
Non-overlapping Template	34/41[a]	0.48	39/41	0.50
Overlapping Template	40/41	0.37	41/41	0.48
Universal	39/41	0.39	40/41	0.44
Approximate Entropy	41/41	0.35	41/41	0.49
Random Excursions	22/22	0.50	-	-
Random Excursions Variant	22/22	0.49	-	-
Serial	39/41	0.43	41/41	0.50
Linear Complexity	39/41	0.49	41/41	0.59

[a] Acceptable pass rate is greater than 38/41 [21]. Failure happened in non-Markov model with high ϕ_1 bitstreams ($r = 1.4$ $\phi_1 = 0.35$, $r \geq 2$ $|\phi_1| \geq 0.2$).

model. MKV1, can remove AC by MKV model. For non-MKV model, while it can not remove perfectly, it still reduces AC several times smaller. For example in case of bistream #6, AC is reduced from 0.30 to 0.11 or 0.03, bistream #4 AC is reduced from 0.20 to 0.05 or 0.02.

V. Combination of Markov and Von Neumann

This section presents Markov chain and von Neumann combined circuits.

A. Discussion on Decorrelation Ability Combined With Markov Chain and Von Neumann

There can be many combinations of N1-bit Markov chains and N2-bit VN or IVN. We focus on 2-bit MKV and VN2, denoted as MKV2/VN2; and 1-bit MKV and VN8W, denoted as MKV1/VN8W. MKV2/VN2 has stronger decorrelation ability, with a maximum ExE of 25.0%. MKV1/VN8W is applied for removing Markov model-based correlation with an ExE of 62.2%. As mentioned, based on measurement

TABLE III
ENT TEST RESULTS FOR PSEUDORANDOM DATA AFTER POST-PROCESSING

	MKV1(QL8)/VN8W, Input: 1M*41			MKV2(QL4)/VN2, Input: 600k*41		
	Average	Pass Rate	Failed Bitstream #	Average	Pass Rate	Failed Bitstream #
Mean	0.4994	34/41	29,31,32,35,38,40,41	0.4999	41/41	-
Entropy	1.0000	34/41	29,31,32,35,38,40,41	1.0000	41/41	-
Chi square	48.53%	34/41	29,31,32,35,38,40,41	57.12%	41/41	-
Monte Carlo	3.1439	40/41	6	3.1401	40/41	18
Serial correlation	-0.0001	38/41	9,25,41	0.0001	40/41	21

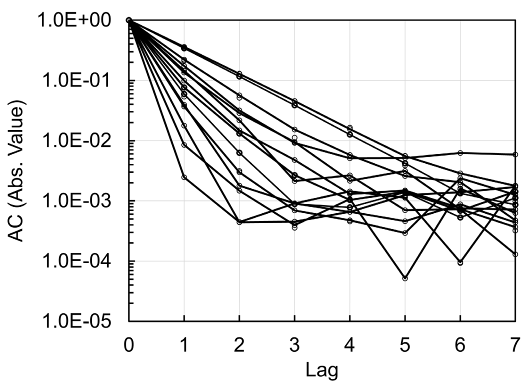


Fig. 13. Autocorrelation analysis of raw bistreams of latch-based TRNG.

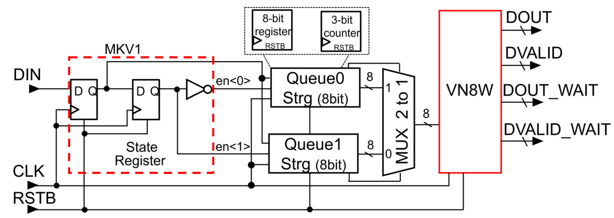


Fig. 14. Block diagram of MKV1(QL8)/VN8W. (Source: [20] modified).

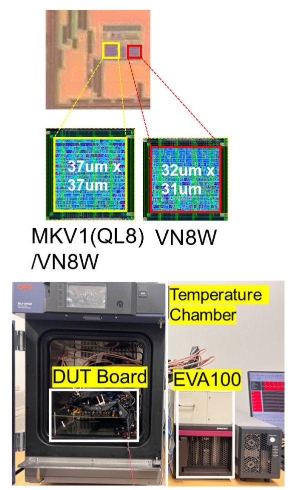


Fig. 15. Die-photo, layout, and measurement setup.

results, we found that under insufficient equalization time, the AC lag 1 in latch-based TRNG can approach 0.3, and the correlations are mostly positive. Therefore, to test most correlation cases, 41 bitstreams, each with a length of 3 million bits, are generated based on the MKV2 model. P_1 is set to 0.5, ϕ_1 is set to $\{\pm 0.05, \pm 0.1, \pm 0.15, \pm 0.2, 0.25, 0.3, 0.35\}$. $\phi_2 = \phi_1^2 \times r$, $\phi_3 = \phi_1^3 \times r^2$, where r quantifies the deviation from the MKV1 model. When r equals 1, the bitstream is a Markov model bitstream; otherwise, the bitstream is a non-Markov bitstream. The 41 combinations of ϕ_1 and r are summarized in Table I. When $|\phi_1|$ is less than or equal to 0.2, r is set to 1, 1.4, 2, or 3. When ϕ_1 exceeds 0.2, since it is already large, r is set to 1, 1.4, and 2 for each ϕ_1 condition.

First, the AC value and post-processed data using MKV1 and MKV2 are shown in Fig. 9. As observed, MKV1 cannot handle highly correlated raw data. Conversely, MKV2 effectively processes nearly all cases and achieves an AC lag 1 less than 0.004. When combined VN, the queue length should also be considered. For a lightweight implementation, MKV2 with a 4-bit queue length with VN2 [MKV2(QL4)/VN2] and MKV1 with an 8-bit queue length with VN8W [MKV1(QL8)/VN8W] are considered. The results are summarized in Fig. 10. As observed, under both circuits, almost all correlations at lag 1 are reduced within the 95% CI. For a full randomness check, NIST SP 800-22 was tested, and the results are summarized in Table II. Note that the NIST results in Table I in our previous work [20] were incorrect due to a bitstream misconnection. The corrected results are presented in this work. It is observed that MKV2(QL4)/VN2 performs well for both Markov and non-Markov model bitstreams. MKV1(QL8)/VN8W works for Markov model ($r = 1$) and non- Markov model when $r = 1.4$ with $|\phi_1| \leq 0.3$; and $r = 2$ with $|\phi_1| \leq 0.15$.

To visually display the random characteristics, we applied the ENT [23] test battery, which includes tests for mean, entropy, serial correlation, chi-square, and Monte Carlo. The results are summarized in Table III. The mean test calculates the P_1 in the bitstream. Entropy is the Shannon entropy. Serial correlation is calculated at the byte level. The chi-square test detects deviations from a uniform distribution by comparing observed frequencies to expected frequencies within 8 bits. The results are depicted in Fig.11. As can be seen, 7 bitstreams fall below 1%, indicating they are almost certainly not random. These 7 bitstreams also failed the NIST test. The Monte Carlo test estimates the value of π using 48 bits by calculating the ratio of points inside a circle to those in a square. Its results are summarized in Fig.12. As observed, in almost all cases, the values are close to the ideal value of π within the 99% confidence interval (identical to the NIST test at $\alpha = 0.01$).

B. Circuit Design

Many combinations of MKV and VN are possible depending on the correlation conditions and throughput requirements

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

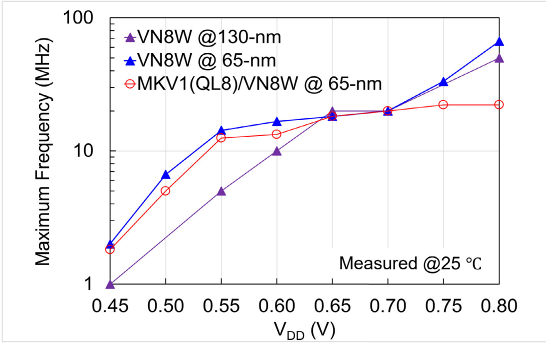


Fig. 16. Maximum frequency versus supply voltage.

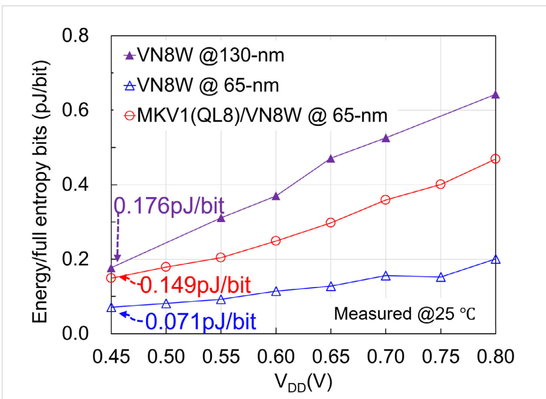


Fig. 17. Energy per full entropy bit versus supply voltage.

of the entropy source. In this work, a latch-based TRNG [11] is used. It has two operation phases: the equalization phase and the evaluation phase. During the evaluation phase, the internal node of the latch is forced to the metastable voltage. The random data is obtained during the evaluation phase. We observed that, under insufficient equalization time, the correlation of raw data exhibits properties similar to MKV1, as shown in Fig. 13. Although increasing the equalization time can reduce the correlation, this approach consumes a lot of energy. Therefore, MKV1(QL8)/VN8W is chosen in this work for hardware implementation to reduce the equalization time and achieve lower energy consumption.

Fig. 14 displays the block diagram of MKV1(QL8)/VN8W. It consists of the MKV1 part, queue storage registers, MUX part for sharing VN8W, and VN8W part. The MKV1 part only includes two registers and one inverter. The input ports are DIN, CLK, and RSTB. The initial bit is first registered in the state register. Then, the subsequent bits are routed to either Queue0 Storage or Queue1 Storage based on the previous bit's state. Each Queue Storage comprises an 8-bit shift register for storing the bits and a 3-bit counter. Once one of the 3-bit counters reaches '111', the stored bits in the Queue Storage are sent to the VN8W block for debiasing. The VN8W processes 8-bits simultaneously and generates output in two parts: the direct output denoted by DOUT and DVALID, and the waiting output denoted by DOUT_WAIT and DVALID_WAIT. The details of VN8W are provided in [19].

TABLE IV
NIST SP 800-22 TEST RESULTS FOR RANDOM DATA FROM LTRNG

	MKV1(QL8)/VN8W Input: 1M*5	
	Pass Rate	Ave. P-Val
Frequency	5/5	0.25
Block Frequency	5/5	0.36
Cumulative Sums	5/5	0.26
Runs	5/5	0.62
Longest Run	5/5	0.44
Rank	5/5	0.58
FFT	5/5	0.54
Non-overlapping Template	5/5	0.49
Overlapping Template	4/5[a]	0.50
Universal	5/5	0.66
Approximate Entropy	5/5	0.32
Random Excursions	2/2	0.46
Random Excursions Variant	2/2	0.42
Serial	5/5	0.38
Linear Comlexity	5/5	0.56

[a] One failure is reasonable due to the limited bitstream numbers [21].

TABLE V
ENT TEST RESULTS FOR RANDOM DATA FROM LTRNG

	MKV1(QL8)/VN8W Input: 1M*5	
	Average	Pass Rate
Mean	0.4995	5/5
Entropy	1.0000	5/5
Chi square	25.14%	5/5
Monte Carlo	3.1480	5/5
Serial correlation	-0.0002	5/5

TABLE VI
NIST SP 800-90B IID TEST RESULTS FOR RANDOM DATA FROM LTRNG

	MKV1(QL8)/VN8W Input: 1M*5
	Passed
Chi square independence	5/5
Chi square goodness of fit	5/5
Length of longest repeated substring test	5/5
IID permutation tests	5/5
Min-Entropy(max/min/ave.)	0.995/0.994/0.993

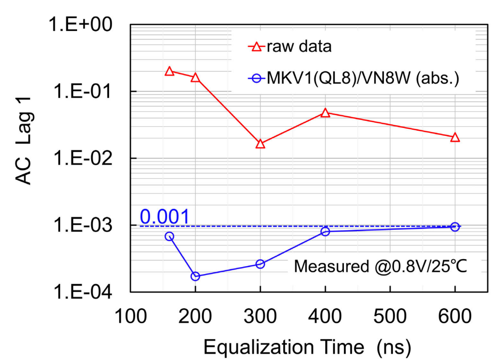


Fig. 18. Autocorrelation under equalization time variations. (Source: [20] modified).

VI. CHIP MEASUREMENT RESULTS

MKV1(QL8)/VN8W was designed using an automatic Place and Route tool (IC Compiler) and fabricated using

TABLE VII
COMPARISON WITH PRIOR POST-PROCESSING TECHNIQUES

	This work		IEICE'2022 [19]	SSCL'2018 [4]	JSSC'2016 [3]	TCAS-II'2019 [15]	TCAS-I'2015 [16]
	MKV1(QL8) /VN8W	**VN8W**	VN8W	MKV4(QL12)/ IVN16/LFSR	Decorrelators/ BIW	Strong Blenders	PRESENT
Process Technology	**65-nm CMOS**	**65-nm CMOS**	130-nm CMOS	65-nm CMOS	14-nm CMOS	45-nm NanGate	32-nm CMOS PTM
Gate Equivalent(GE)	**950**	**689**	381	NA	586	166.3 ~13K	1171
Queue Memory (bits)	**16**	**-**	-	192	-	-	-
Max. Extraction Efficiency	**62.21%**	**62.21%**	62.21%	≈ 78%	12.5%	4%~20%	50%/80%
De-Biasing	**Yes**	**Yes**	Yes	Yes	Yes	Yes	Yes
De-Correlation	**AC lag 1 ≤0.35[b]**	**AC lag 1 ≤0.03**	AC lag 1 ≤0.03	Yes[a]	Mutual Correlation <0.003	Yes[a]	Yes[a]
Energy per full-entropy bit (pJ/bit)	**0.149 @0.45 V**	**0.071 @0.45 V**	0.176 @0.45 V	2.58[c] @0.53 V	9 @0.75 V	NA	1.0-2.5[c] @0.9V

[a] No specific data has been provided in the paper. MKV4(QL12) in [4] is designed for removing AC up to lag 4. The works in [15], [16] are described for removing any correlation.
[b] MKV1(QL8)/VN8W works for Markov model ($r = 1$) with $|\phi_1| \leq 0.35$ and non- Markov model when $r = 1.4$ with $|\phi_1| \leq 0.3$; and $r = 2$ with $|\phi_1| \leq 0.15$.
[c] Including TRNG cores.

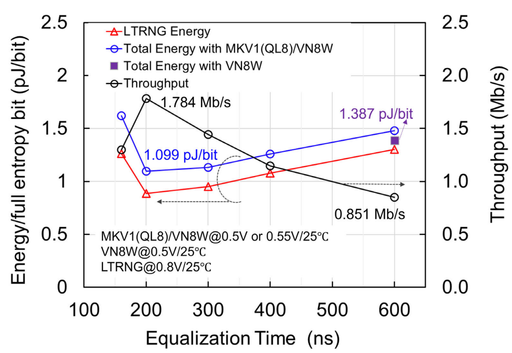


Fig. 19. Energy and throughput under equalization time variations.

TSMC 65-nm CMOS technology. To ensure the comparison, VN8W, as previously implemented in 130-nm CMOS according to [19], was also redesigned using the 65-nm CMOS process. MKV1(QL8)/VN8W occupies an area of 1369 μm^2, equivalent to 950 gate equivalents (GEs). VN8W occupies an area of 992 μm^2, equivalent to 689 GEs. The area of MKV1(QL8)/VN8W is 1.38 times larger than that of VN8W, primarily due to the memory register storing the queue data, leading to area overhead. The die photo, layout image and measurement setup are depicted in Fig. 15. One test chip is utilized to verify the performance of the proposed circuit. Further details are provided in the following.

A. Frequency and Energy Results of Post-Processing Circuits

MKV1(QL8)/VN8W and VN8W were tested with a supply voltage ranging from 0.45 V to 0.8 V, incremented by 0.05 V per step. Fig. 16 displays the maximum frequency results. The tested input data consists of raw output data from latch-based TRNG [11] with 1.2% bias and 0.0006 AC at lag 1. For comparison purposes, the measurement results of VN8W in 130-nm technology are also depicted in the graph. As observed, the maximum frequency increases with the rise in supply voltage, a trend consistent across all three circuits. At 0.45 V, MKV1(QL8)/VN8W and VN8W @ 65-nm operate at 1.82 MHz and 2.00 MHz, respectively, while VN8W @ 130-nm operates at 1 MHz. The maximum frequency of MKV1(QL8)/VN8W increases slowly after 0.65 V and achieves 22.22 MHz at 0.8 V. The reduced highest frequency in MKV1(QL8)/VN8W is attributed to the delay in the memory shift registers for storing queue data. The maximum frequency of VN8W @ 65-nm is, on average, 1.63 times larger than VN8W @ 130-nm, demonstrating the merit of technology scaling.

Fig. 17 shows the energy results, wherein the energy per full entropy bit [19] is calculated as follows:

$$E_{post-processing} = \frac{Power}{Frequency \times ExE}, \tag{12}$$

where the ExE of input test data is 62%, a value identical across all three circuits. As can be seen, the minimum energy at 0.45 V of MKV1(QL8)/VN8W, VN8W @ 65-nm, and VN8W @ 130-nm are 0.149 pJ/bit, 0.071 pJ/bit, and 0.176 pJ/bit, respectively. Compared with VN8W @ 65-nm, MKV1(QL8)/VN8W costs 2.3 times, on average, larger energy. This is due to the fact that MKV1(QL8)/VN8W has 2 times larger register count than VN8W @65-nm, which consumes a significant amount of energy. Additionally, VN8W @ 130-nm consumes 3.2 times larger energy than VN8W @ 65-nm.

B. Application in Latch-Based TRNG

MKV1(QL8)/VN8W is utilized in the latch-based TRNG (LTRNG) [11], which requires a prolonged equalization time to reduce correlation. The autocorrelation lag 1 value versus equalization time, ranging from 160 ns to 600 ns, is shown in Fig. 18. As observed, the raw data consistently exhibits values larger than 0.02 across all equalization times. After post-processed by MKV1(QL8)/VN8W, all AC lag 1 values remain consistently below 0.001. The randomness of post-processed data is verified by NIST SP 800-22, ENT, and NIST SP 800-90B, as summarized in Table IV, Table V, and Table VI, respectively.

The energy and throughput results are depicted in Fig. 19. As observed, by utilizing MKV1(QL8)/VN8W, the total energy, calculated as the energy of LTRNG (divided by ExE) + the energy of post-processing [19]–can be reduced to 1.099 pJ/bit (= 0.522/0.588+0.212). This represents a 21% improvement compared to the total energy with VN8W, which is 1.387 pJ/bit. Additionally, the throughput is also improved by 2.09 times (= 1.784/0.851), owing to the enhanced decorrelation ability by MKV1(QL8)/VN8W.

C. Comparisons

Table VII shows the comparisons with previous works. For higher correlation post-processing circuit, MKV1(QL8)/VN8W with 62.21% ExE costs considerably fewer GEs when compared with [15] and [16]. Compared with work in [4], the queue memory is reduced 12 times. Although MKV1(QL8)/VN8W is 1.38 times larger than the VN8W implemented in 65-nm, it can handle Markov model data with AC lag 1 up to 0.35. VN8W implemented in 65-nm achieves the minimum energy among others, which is more than two times smaller than the energy of the previous work [19], demonstrating the merit of using advanced technology.

VII. Conclusion

In this study, we introduce a lightweight combination of Markov chains and von Neumann. We verify the decorrelation capability of the combined circuits. MKV1(QL8)/VN8W was implemented in a 65-nm CMOS. Compared to previous works, the queue memory was reduced by a factor of 12. It achieved a low energy consumption of 0.149 pJ/bit at 0.45 V. When applied in a latch-based TRNG, the throughput improved by 2.09 times, and the total energy decreased by 21% due to the decorrelation enhancement provided by MKV1(QL8)/VN8W.

As for the practical use of MKV and VN combined post-processing circuits, the designer should first characterize their TRNGs' correlation properties. If the correlation follows a Markov model, i.e., $\phi_k = \phi_1^k$, then MKV1 is appropriate. If the correlation follows a non-Markov model, first judge if it has less than AC lag 3 correlations. If yes, MKV2 is appropriate. MKV2 has four degrees of freedom: P_1, ϕ_1, ϕ_2, and ϕ_3. In theory, it can solve up to AC lag 3 correlations. If the raw bitstream has long-term correlations such as AC lag 4 or more complex statistical defects, a higher-order MKV model is needed, such as a 4-bit 16-state Markov model. Then, based on the trade-off between logic complexity and throughput requirements, choose a VN circuit with an appropriate queue memory bit length.

The correlation test cases in this work are still not enough due to the limited variety and number of tested scenarios. In future work, we would like to survey more practical TRNG types, including those used in different applications and environments, to better understand their correlation properties. Additionally, we plan to extend our MKV model and VN circuits to provide appropriate solutions.

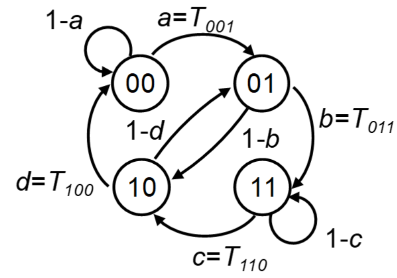


Fig. 20. The diagram of MKV2.

Appendix

In this appendix, characteristics of frequency P_1 and autocorrelations at lag 1, 2, 3 (ϕ_1, ϕ_2, ϕ_3) in random bitstream generated by two-bit (four state) Markov chain MKV2 are extracted from its four independent transition probabilities T_{001}, T_{011}, T_{110}, T_{100}. Inverse functions that calculate T_{001}, T_{011}, T_{110}, T_{100} from P_1, ϕ_1, ϕ_2, ϕ_3 are also clarified. Random bit streams with arbitral characteristics P_1, ϕ_1, ϕ_2, ϕ_3 can be generated by applying calculated T_{001}, T_{011}, T_{110}, T_{100} to MKV2 state transition model.

A. Definitions

T_{ijk} : Transition probability from state $[i, j]$ to $[j, k]$ $(i, j, k \in \{0, 1\})$

P_{ij} : Probability to be in a state $[i, j]$. This is same as the probability that two adjacent (lag 1) bit pair in generated bit stream is (i, j).

P_{ijk} : Probability that three consecutive bits in bit stream is (i, j, k)

P_{ijkl} : Probability that four consecutive bits in bit stream is (i, j, k, l)

P_{i-j} : Probability that pair of bits with a distance 2 (lag 2) in bit stream is (i, j)

P_{i--j} : Probability that pair of bits with a distance 3 (lag 3) in bit stream is (i, j)

For simplicity, variables a, b, c, d are used as $a = T_{001}$, $b = T_{011}$, $c = T_{110}$, $d = T_{100}$, as shown in Fig. 20.

B. Relation Between P_{ij} and Four Independent Transition Probabilities

In the equilibrium, probabilities of the four states P_{00}, P_{01}, P_{10}, P_{11} don't change after transition. Thus,

$$P_{00} = (1-a)P_{00} + dP_{10} \quad (13)$$

$$P_{01} = aP_{00} + (1-d)P_{10} \quad (14)$$

$$P_{10} = (1-b)P_{01} + cP_{11} \tag{15}$$
$$P_{01} = bP_{00} + (1-c)P_{10} \tag{16}$$

By adding equations (13) and (14),

$$\begin{aligned} P_{00} + P_{01} &= (1-a)P_{00} + dP_{10} + aP_{00} + (1-d)P_{10} \\ &= P_{00} + P_{11} \end{aligned}$$

Thus,

$$P_{01} = P_{10}. \tag{17}$$

Same result can be obtained by adding equations (15) and (16). From (13), $aP_{00} = dP_{10}$, $P_{00} = qP_{10} = qP_{01}$, where $q = d/a$. From (15), $bP_{01} = cP_{11}$, $P_{11} = sP_{01} = sP_{10}$, where $s = b/c$.

C. Extract P_1, P_0, ϕ_1 From q, s

Since $P_{00} + P_{01} + P_{10} + P_{11} = 1$, $(q + s + 2)P_{01} = 1$.Thus,

$$\begin{cases} P_{01} = P_{10} = \dfrac{1}{2+q+s} \\ P_{00} = \dfrac{q}{2+q+s} \\ P_{11} = \dfrac{s}{2+q+s} \end{cases} \tag{18}$$

$$\begin{cases} P_1 = \dfrac{P_{01} + P_{10} + 2P_{11}}{2} = \dfrac{1+s}{2+q+s} \\ P_0 = 1 - P_1 = \dfrac{1+q}{2+q+s} \end{cases} \tag{19}$$

Applying (18) (19) in Pearson's autocorrelation formula,

$$\phi_1 = \frac{n_{11}n_{00} - n_{10}n_{01}}{n_1 n_0} = \frac{P_{11}P_{00} - P_{10}P_{01}}{P_1 P0} = \frac{qs - 1}{(1+q)(1+s)} \tag{20}$$

D. Extract ϕ_2 From q, s, b, d

P_{i-j} are decomposed as

$$\begin{cases} P_{0-0} = P_{000} + P_{010} \\ P_{0-1} = P_{001} + P_{011} \\ P_{1-0} = P_{100} + P_{110} \\ P_{1-1} = P_{101} + P_{111} \end{cases} \tag{21}$$

Bit string [000] happens when 0 appears next to state [00]. Thus,

$$P_{000} = (1-a)P_{00} \tag{22}$$

In the same way,

$$\begin{cases} P_{001} = aP_{00}P_{010} = (1-b)P_{01}P_{011} = bP_{01} \\ P_{100} = dP_{10}P_{101} = (1-d)P_{10} \\ P_{110} = cP_{11}P_{111} = (1-c)P_{11} \end{cases} \tag{23}$$

Substituting (22), (23) and (20) into (21),

$$\begin{cases} P_{0-0} = (1-a)P_{00} + (1-b)P_{10} \\ = \dfrac{1+q-(aq+b)}{2+q+s} = \dfrac{1+q-(b+d)}{2+q+s} \\ P_{0-1} = aP_{00} + bP_{01} = \dfrac{aq+b}{2+q+s} = \dfrac{b+d}{2+q+s} \\ P_{1-0} = dP_{10} + cP_{11} = \dfrac{d+cs}{2+q+s} = \dfrac{b+d}{2+q+s} \\ P_{1-1} = (1-d)P_{10} + (1-c)P_{11} \\ = \dfrac{1+s-(cs+d)}{2+q+s} = \dfrac{1+s-(b+d)}{2+q+s} \end{cases} \tag{24}$$

Applying (24) (19) in Pearson's autocorrelation formula,

$$\begin{aligned} \phi_2 &= \frac{P_{1-1}P_{0-0} - P_{1-0}P_{0-1}}{P_1 P_0} \\ &= \frac{(1+q)(1+s) - (1+q)(b+d) - (1+s)(b+d)}{(1+q)(1+s)} \\ &= 1 - \frac{2+q+s}{(1+q)(1+s)}(b+d) \end{aligned} \tag{25}$$

E. Extract ϕ_3 From q, s, b, d

P_{i--j} are decomposed as

$$\begin{cases} P_{0--0} = P_{0000} + P_{0010} + P_{0100} + P_{0110} \\ P_{0--1} = P_{0001} + P_{0011} + P_{0101} + P_{0111} \\ P_{1--0} = P_{1000} + P_{1010} + P_{1100} + P_{1110} \\ P_{1--1} = P_{1001} + P_{1011} + P_{1101} + P_{1111} \end{cases} \tag{26}$$

Bit string [0000] happens when 0 and 0 appear next to state [00]. Thus,

$$P_{0000} = (1-a)^2 P_{00} \tag{27}$$

In the same way,

$$\begin{cases} P_{0001} = (1-a)aP_{00} \\ P_{0010} = a(1-b)P_{00} \\ P_{0011} = abP_{00} \\ P_{0100} = (1-b)dP_{01} \\ P_{0101} = (1-b)(1-d)P_{01} \\ P_{0110} = bcP_{01} \\ P_{0111} = b(1-c)P_{01} \\ P_{1000} = d(1-a)P_{10} \\ P_{1001} = daP_{10} \\ P_{1010} = (1-d)(1-b)P_{10} \\ P_{1011} = (1-d)bP_{10} \\ P_{1100} = cdP_{11} \\ P_{1101} = c(1-d)P_{11} \\ P_{1110} = (1-c)cP_{11} \\ P_{1111} = (1-c)^2 P_{11} \end{cases} \tag{28}$$

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Substituting (27), (28) and (20) into (26),

$$\begin{cases} P_{0--0} = (1-a)^2 P_{00} + a(1-b)P_{00} + \\ (1-b)dP_{01} + bcP_{01} \\ = (1-A)P_{00} + DP_{01} \\ P_{0--1} = (1-a)aP_{00} + abP_{00} + \\ (1-b)(1-d)P_{01} + b(1-c)P_{01} \\ = AP_{00} + (1-D)P_{01} \\ P_{1--0} = d(1-a)P_{10} + (1-d)(1-b)P_{10} + \\ cdP_{11} + (1-c)cP_{11} \\ = (1-B)P_{10} + CP_{11} \\ P_{1--1} = daP_{10} + (1-d)bP_{10} + \\ c(1-d)P_{11} + (1-c)^2 P_{11} \\ = BP_{10} + (1-C)P_{11} \end{cases} \tag{29}$$

where $A = a + ab - a^2$, $B = b + ad - bd$, $C = c + cd - c^2$, $D = d + bc - bd$. Applying (29) (19) in Pearson's autocorrelation formula,

$$\phi_3 = \frac{P_{1--1}P_{0--0} - P_{1--0}P_{0--1}}{P_1 P_0} = \frac{qs - 1 + (D - Aq)(s+1) + (B - Cs)(q+1)}{(1+q)(1+s)} \tag{30}$$

Here,

$$D - Aq = d + bc - bd - d(1 + b - a) = ad + bc - 2bd \tag{31}$$

$$B - Cs = b + ad - bd - b(1 + d - c) = ad + bc - 2bd \tag{32}$$

Substituting (31) (32) into (30)

$$\phi_3 = \frac{qs - 1 + (2 + q + s)(ad + bc - 2bd)}{(1+q)(1+s)} = \frac{qs - 1 + (2 + q + s)(b^2/s + d^2/q - 2bd)}{(1+q)(1+s)} \tag{33}$$

F. Extract q, s From P_1, ϕ_1

Set $t = 1 + q$, $u = 1 + s$, From (19)

$$P_1 = \frac{u}{t+u} \tag{34}$$

$$P_0 = \frac{t}{t+u} \tag{35}$$

From (20)

$$\begin{cases} \phi_1 = \dfrac{tu - t - u}{tu} = 1 - \dfrac{t+u}{tu} \\ \dfrac{1}{1-\phi_1} = \dfrac{tu}{t+u} \end{cases} \tag{36}$$

Divide (36) with (34)

$$t = \frac{1}{P_1(1-\phi_1)}$$

Divide (36) with (35)

$$u = \frac{1}{P_0(1-\phi_1)}$$

Thus,

$$q = \frac{1}{P_1(1-\phi_1)} - 1$$

$$s = \frac{1}{P_0(1-\phi_1)} - 1 = \frac{1}{(1-P_1)(1-\phi_1)} - 1$$

Extracted q and s are used in the followings.

G. Extract b From q, s, ϕ_1, ϕ_2, ϕ_3

Substituting (36) into (25)

$$\begin{cases} \phi_2 = 1 - \dfrac{2+q+s}{(1+q)(1+s)}(b+d) \\ = 1 - \dfrac{t+u}{tu}(b+d) = 1 - (1-\phi_1)(b+d) \\ b + d = \dfrac{1-\phi_2}{1-\phi_1} = E \\ d = E - b \end{cases} \tag{37}$$

where $E = \frac{1-\phi_2}{1-\phi_1}$. Substituting (20) and (36) into (33)

$$\begin{cases} \phi_3 = \dfrac{qs - 1 + (2+q+s)(\frac{b^2}{s} + \frac{d^2}{q} - 2bd)}{(1+q)(1+s)} \\ = \phi_1 + (1-\phi_1)(\dfrac{b^2}{s} + \dfrac{d^2}{q} - 2bd) \\ \dfrac{b^2}{s} + \dfrac{d^2}{q} - 2bd = \dfrac{\phi_3 - \phi_1}{1-\phi_1} = F \end{cases} \tag{38}$$

where $F = \frac{\phi_3 - \phi_1}{1-\phi_1}$. Substituting (37) into (38),

$$(1/s + 1/q + 2)b^2 - 2E(1/q + 1)b + \frac{E^2}{q} - F = 0.$$

Solving this quadratic equation,

$$b = \frac{E(1/q+1) \pm \sqrt{E^2(1/q+1)^2 - (1/s+1/q+2)(E^2/q - F)}}{1/s + 1/q + 2}.$$

Generally, b has two solutions. But in the case of the Markov model where bit stream is generated by MKV1, the square root term becomes zero, and b has only one solution.

H. Find T_{001}, T_{011}, T_{110}, T_{100} From b and Other Extracted Parameters

$$T_{011} = b$$

$$T_{100} = d = E - b$$

$$T_{001} = a = \frac{d}{q}$$

$$T_{110} = c = \frac{b}{s}$$

End.

References

[1] 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. Solid-State Circuits*, vol. 47, no. 11, pp. 2807–2821, Nov. 2012.

[2] K. Yang, D. Blaauw, and D. Sylvester, "An all-digital edge racing true random number generator robust against PVT variations," *IEEE J. Solid-State Circuits*, vol. 51, no. 4, pp. 1022–1031, Apr. 2016.

[3] 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. Solid-State Circuits*, vol. 51, no. 7, pp. 1695–1704, Jul. 2016.

[4] 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 Lett.*, vol. 1, no. 12, pp. 237–240, Dec. 2018.

[5] D. Johnston, *Random Number Generators—Principles and Practices, a Guide for Engineers and Programmers*. Berlin, Germany: De Gruyter Press, Sep. 2018.

[6] S. K. 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. Solid-State Circuits*, vol. 54, no. 4, pp. 1074–1085, Apr. 2019.

[7] M. Alioto, "Trends in hardware security: From basics to ASICs," *IEEE Solid State Circuits Mag.*, vol. 11, no. 3, pp. 56–74, Aug. 2019.

[8] 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 63rd Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2020, pp. 285–288.

[9] 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) Dig. Tech. Papers*, Feb. 2021, pp. 498–500.

[10] M. Grujic and I. Verbauwhede, "TROT: A three-edge ring oscillator based true random number generator with time-to-digital conversion," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 6, pp. 2435–2448, Jun. 2022.

[11] 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 J. Solid-State Circuits*, vol. 57, no. 8, pp. 2498–2508, Aug. 2022.

[12] P. Keshavarzian et al., "A 3.3-Gb/s SPAD-based quantum random number generator," *IEEE J. Solid-State Circuits*, vol. 58, no. 9, pp. 2632–2647, Sep. 2023.

[13] X. Wang, R. Zhang, K. Liu, and H. Shinohara, "A 0.116 pJ/bit latch-based true random number generator featuring static inverter selection and noise enhancement," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 32, no. 3, pp. 564–572, Mar. 2024.

[14] J. Kim and H. Chae, "A 10-Gb/s true random number generator using ML-resistant middle square method," *IEEE J. Solid-State Circuits*, vol. 59, no. 7, pp. 2321–2329, Jul. 2024.

[15] V. Rožic and I. Verbauwhede, "Hardware-efficient post-processing architectures for true random number generators," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 66, no. 7, pp. 1242–1246, Jul. 2019.

[16] V. B. Suresh and W. P. Burleson, "Entropy and energy bounds for metastability based TRNG with lightweight post-processing," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 7, pp. 1785–1793, Jul. 2015.

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

[18] V. Rožic, B. Yang, W. Dehaene, and I. Verbauwhede, "Iterating von Neumann's post-processing under hardware constraints," in *Proc. IEEE Int. Symp. Hardw. Oriented Secur. Trust (HOST)*, May 2016, pp. 37–42.

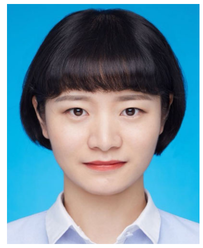
[19] R. Zhang, X. Wang, and H. Shinohara, "Energy-efficient post-processing technique having high extraction efficiency for true random number generators," *IEICE Trans. Electron.*, vol. 104, no. 7, pp. 300–308, 2021.

[20] R. Zhang, H. Zhang, X. Wang, Y. Ziyang, K. Liu, and H. Shinohara, "Practical Markov chain and von Neumann based post-processing circuits for true random number generators," in *Proc. IEEE 66th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2023, pp. 841–845.

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

[22] M. S. Turan, E. Barker, J. Kelsey, K. A. McKay, M. L. Baish, and M. Boyle, *Recommendation for the Entropy Sources Used for Random Bit Generation*, document NIST SP 800-90B, Jan. 2018. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90B.pdf

[23] J. Walker. (Jan. 2008). *ENT: A Pseudorandom Number Sequence Test Program*. [Online]. Available: https://fourmilab.ch/random/



Ruilin Zhang (Member, IEEE) received the B.S. degree from Beijing University of Chemical Technology, Beijing, China, in 2015, and the M.S. and Ph.D. degrees from Waseda University, Fukuoka, Japan, in 2017 and 2022, respectively.

From June 2022 to April 2024, she was an Adjunct Researcher with the Information, Production, and Systems Research Center, Waseda University, Kitakyushu, Japan. Since December 2023, she has been a Program-Specified Assistant Professor with the Graduate School of Informatics, Kyoto University. Her current research interests include the design of true random number generators (TRNGs), TRNG post-processing circuits, stochastic number generators, and hardware security.

Haochen Zhang received the B.S. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2022, and the M.S. degree from Waseda University, Fukuoka, Japan, in 2023. He is currently a Chip Engineer with Lenovo.

Xingyu Wang (Member, IEEE) received the B.S. degree in advanced energy material and devices from Southeast University (SEU), Nanjing, China, in 2018, and the M.E. and Ph.D. degrees from the Graduate School of Information, Production and Systems, Waseda University, Fukuoka, Japan, in 2019 and 2024, respectively.

His research interests include true random number generator (TRNG) circuit design, TRNG post-processing techniques, physically unclonable function (PUF) circuit design, and hardware security. He was a recipient of the IEEE VLSI-DAT Best Paper Award in 2022 and the IEEE Fukuoka Section Excellent Student Award in 2023.

Ye Ziyang (Graduate Student Member, IEEE) received the B.E. degree from Southeast University and the M.E. degree from Waseda University. He is currently pursuing the Ph.D. degree in computing and communication systems with the Department of Electrical Engineering and Information Systems, The University of Tokyo. His research interests include integrated circuit encryption systems, emphasizing security primitives, and logic circuit design.

Kunyang Liu (Member, IEEE) received the B.E. degree in electronic and information engineering from South China University of Technology (SCUT), Guangzhou, China, in 2015, and the M.E. and Ph.D. degrees from the Graduate School of Information, Production and Systems, Waseda University, Kitakyushu, Japan, in 2017 and 2021, respectively.

From 2021 to 2024, he was an Assistant Professor with the Information, Production and Systems Research Center, Waseda University. Since 2024, he has been with the Graduate School of Informatics, Kyoto University, Kyoto, Japan, where he is currently an Assistant Professor. His research interests include physically unclonable function (PUF) and true random number generator (TRNG) integrated circuits, memory-centric circuits, hot carrier injection (HCI) effect, and hardware security.

Dr. Liu was a Technical Committee Member of the Technical Committee on Integrated Circuits and Devices (ICD) of the Institute of Electronics, Information, and Communication Engineers (IEICE), a Steering Committee Member of the Special Interest Group on System and LSI Design Methodology (SLDM) of Information Processing Society in Japan (IPSJ), and an Editorial Committee Member of the *Journal of Information Processing*. He is a member of IEICE and IPSJ. He was a recipient or co-recipient of the Azusa Ono Memorial Scholarship in 2016, the IEEE A-SSCC Student Travel Grant Award in 2018, the Excellent Student Award of the IEEE Fukuoka Section in 2020, and the Best Paper Award of the International Symposium on VLSI Design, Automation and Test (VLSI-DAT) in 2022.

Shinichi Nishizawa (Member, IEEE) received the B.E. degree from Ritsumeikan University, Japan, in 2009, and the Ph.D. degree from Kyoto University, Japan, in 2015.

He was an Assistant Professor with Saitama University from 2015 to 2019 and Fukuoka University from 2019 to 2022. He has been an Assistant Professor with Waseda University since 2022. His research interests include the physical layout of VLSI design and its CAD technologies.

Kiichi Niitsu (Member, IEEE) was born in Japan, in 1983. He received the B.S. (summa cum laude), M.S., and Ph.D. degrees in electrical engineering from Keio University, Yokohama, Japan, in 2006, 2008, and 2010, respectively.

In 2010, he was an Assistant Professor with Gunma University, Kiryu, Japan. In 2012, he was a Lecturer with Nagoya University, Nagoya, Japan. Since 2015, he has been a Precursory Research for Embryonic Science and Technology (PRESTO) Researcher with Japan Science and Technology Agency (JST). Since 2018, he has been an Associate Professor with Nagoya University. Since 2022, he has been a Professor with Kyoto University, Kyoto, Japan. From 2008 to 2010, he was a Research Fellow of Japan Society for the Promotion of Science (JSPS), a Research Assistant with the Global Center of Excellence (GCOE) Program, Keio University, and a Collaboration Researcher with the Keio Advanced Research Center (KARC). He has published 82 referred original journal articles, 187 international conference papers, and three book chapters, including (five) IEEE JOURNAL OF SOLID-STATE CIRCUITS, (four) IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS, (one) IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, (two) IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS, (one) IEEE OPEN JOURNAL OF CIRCUITS AND SYSTEMS, (one) IEEE SOLID-STATE CIRCUITS LETTERS, (five) IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, (two) ISSCC, (four) Symposium on VLSI Circuits, (one) CICC, (four) A-SSCC, (one) ESSCIRC, (14) BioCAS, (two) ISCAS, (two) ISICAS, (eight) ICECS, and (nine) APCCAS. His current research interests include the low-power and high-speed technologies of analog and digital VLSI circuits for biomedical applications.

Dr. Niitsu received the 2006 KEIO KOUGAKUKAI Award, the 2007 INOSE Science Promotion Award, the 2008 IEEE SSCS Japan Chapter Young Researcher Award and the 2009 IEEE SSCS Japan Chapter Academic Research Award both from the IEEE Solid-State Circuits Society Japan Chapter, the 2008 FUJIWARA Award from the FUJIWARA Foundation, the 2011 YASUJIRO NIWA Outstanding Paper Award, the 2011 FUNAI Research Promotion Award, the 2011 Ando Incentive Prize for the Study of Electronics, the 2011 Ericsson Young Scientist Award, the 2012 ASP-DAC University LSI Design Contest Design Award, the NF Foundation Research and Development Encouragement Award, the AKASAKI Award from Nagoya University, the IEEE Nagoya Section Young Researcher Award, the 2016 IEEE Biomedical Circuits and Systems Conference (BioCAS 2016) Best Paper Award, the 2017 Commendation for Science and Technology by the Minister of Education, Culture, Sports, Science and Technology, the Young Scientists' Prize, the 2018 IEICE SUEMATSU-Yasuharu Award, the IEEE Biomedical Circuits and Systems Conference 2018 (BioCAS 2018) Best Live Demonstration Award, the 2019 ASP-DAC UDC Special Feature Award, the 2019 Bio Industry Research Award, the IEEE ICECS 2020 Young Professional Best Paper Award, the 2022 ASP-DAC Best Design Award, and the 2023 ISCAS IEEE BioCAS-TC Best Paper Recognition. He was an Associate Editor of IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS; a Technical Committee of IEEE Biomedical Circuits and Systems (BioCAS TC); the Live Demo Session Co-Chair of the 2019 BioCAS; a Review Committee Member of ISCAS 2017/2018/2019; a Technical Program Committee Member of CICC since 2019, ICECS since 2018, and LASCAS since 2021; a Review Committee Member of APCCAS 2014; an Editor of IEICE ELEX; an Editorial Committee Member of *IEICE Transactions on Electronics*, Special Section on Analog Circuits and Related SoC Integration Technologies; and an Editorial Committee Member of the IEICE ESS Fundamental Review. He is a member of the Institute of Electronics, Information and Communication Engineers of Japan (IEICE) and Japan Society of Applied Physics (JSAP).

Hirofumi Shinohara (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering and the Ph.D. degree in informatics from Kyoto University, Kyoto, Japan, in 1976, 1978, and 2008, respectively.

In 1978, he joined the LSI Laboratory, Mitsubishi Electric Corporation, Itami, Japan, where he was involved in the research and development of metal oxide semiconductor (MOS) static random access memories (SRAMs), memory compilers, logic building blocks, and neuro-chips. From 2003 to 2009, he was engaged in the development of basic logic circuits, memory macros, and design methodology for advanced CMOS technologies with Renesas Technology Corporation, Itami. He moved to the Semiconductor Technology Research Academic Center (STARC), Yokohama, Japan, in 2009, and directed a joint research project on extremely low-power circuits and systems that operate near/sub-threshold regions with universities in Japan. He moved to academia in 2015. From 2015 to 2024, he was a Professor with the Graduate School of Information, Production and Systems, Waseda University, Kitakyushu, Japan. After retiring from Waseda University in 2024, he moved to the Graduate School of Informatics, Kyoto University, to continue his research. His current research interests include energy-efficient random circuits for security, such as physical unclonable functions, true random number generators, and energy-efficient analog and digital circuits.

Dr. Shinohara is a member of IEICE. He has been serving on the International Technical Program Committee of the IEEE International Solid-State Circuits Conference (ISSCC) from 2017 to 2021 and the IEEE International Symposium on VLSI Design, Automation and Test (VLSI-DAT, currently VLSI-TSA), since 2016.