

Paper

Generation of chaos-based random bit sequences with prescribed auto-correlations by post-processing using linear feedback shift registers

Tin Ni Ni Kyaw^{1a)} and Akio Tsuneda^{1b)}

¹ Department of Computer Science and Electrical Engineering, Kumamoto University, 2-39-1 Kurokami, Chuo-ku, Kumamoto 860-8555, Japan

^{a)} tinninikyaw@st.cs.kumamoto-u.ac.jp

^{b)} tsuneda@cs.kumamoto-u.ac.jp

Received November 10, 2016; Revised March 1, 2017; Published July 1, 2017

Abstract: Post-processing scheme using linear feedback shift registers (LFSRs) for generating chaos-based random bit sequences with infinite period for use in Monte-Carlo methods is discussed. We theoretically analyze auto-correlation functions of chaotic binary sequences generated by the Bernoulli chaotic map and a class of binary functions. The theoretical analyses can be applied to the LFSR-based post-processing for generating random bit sequences with prescribed auto-correlations because LFSRs can be regarded as a finite bit approximation of the Bernoulli map. Some results of numerical experiments are also given.

Key Words: random bit sequence, post-processing, linear feedback shift register, chaos, Bernoulli map, perron-frobenius operator

1. Introduction

Monte-Carlo methods are used to solve many numerical problems in science and engineering [1]. In Monte-Carlo method, random numbers with very long period, preferably with infinite period such as physical random numbers, are required to enhance the accuracy. Usually, a truly random binary (or M-ary) number should be a sequence of independent and identically distributed (*i.i.d.*) random variables with a uniform probability density. Its typical model, for example, is a sequence obtained by trials of fair coin tossing or dice throwing. Such randomness is very important for random numbers, especially in their cryptographic applications. However, not only uncorrelated random sequences but also correlated ones are useful in Monte-Carlo simulations [1, 2], for simulating various kinds of stochastic phenomena.

Chaos can be a good candidate of a random number generator for Monte Carlo methods because some of chaotic systems can be theoretically analyzed [3, 4] and we can theoretically design truly random (uncorrelated) sequences [5, 6] and correlated ones [2, 7] using one-dimensional (1-D) discrete-time chaotic systems. For example, the convergence rate of Monte-Carlo simulations can be increased

by using chaotic sequences with proper auto-correlations, which is called *super-efficient chaotic Monte-Carlo simulations* [2].

In order to generate chaotic sequences with infinite period for applications to Monte-Carlo methods, we should realize the chaotic systems by analog circuits because digital circuits generate only (eventually) periodic sequences. However, it is difficult for analog circuits to generate chaotic sequences with designed (prescribed) statistical properties due to non-idealities of analog circuit elements and inevitable noise [3, 6]. Therefore, several post-processing methods to improve their statistical properties have been proposed [8–12].

In [9], a random number generator using an aperiodic random bit generator and a linear feedback shift register (LFSR) was proposed for generating balanced and uncorrelated random bit sequences, where LFSRs are considered to approximate the Bernoulli chaotic map with finite bits. It was shown that the balance property of the output bit sequences is good in spite of the imbalance property of the original (input) bit sequences. Regarding to the auto-correlation property, the output bit sequence is almost uncorrelated when the length of LFSRs is large although the original aperiodic bit sequence is positively/negatively correlated.

In [10], another combinational logic circuit with 3-input/1-output is added to the generator given in [9] for generating random bit sequences with prescribed positive/negative auto-correlations, where the combinational logic circuits are also designed based on the chaos theory for the Bernoulli map. The 3-input/1-output combinational logic circuits were derived from the binary functions defined on the real-valued interval $[0, 1]$ divided into 8 subintervals $[\frac{j}{8}, \frac{j+1}{8}]$ ($j = 0, 1, \dots, 7$).

In this paper, we generalize the post-processing method given in [10] for generating aperiodic random bit sequences with more various kinds of positive/negative auto-correlations. First, we investigate the theoretical auto-correlation functions of chaotic binary sequences generated by the Bernoulli map, where the binary functions are defined on the real-valued interval $[0, 1]$ divided into 2^m subintervals $[\frac{j}{2^m}, \frac{j+1}{2^m}]$ ($j = 0, 1, \dots, 2^m - 1$). We give some examples of binary functions for $m = 4$ which generate binary sequences with different auto-correlation functions.

Next, we design 4-input/1-output combinational logic circuits based on the above binary functions for $m = 4$ and apply them to the post-processing with LFSRs for generating random bit sequences with the theoretical (target) auto-correlation properties. Several numerical experiments are performed for confirming the validity of our new post-processing circuits.

The rest of this paper is organized as follows. In Section 2, we give statistical analysis of 1-D chaos including generating the chaotic binary sequences by using the Bernoulli map and also describe the evaluation of correlation functions with Perron-Frobenius (PF) operator. In Section 3, we perform some numerical experiments for the proposed post-processing method. Finally, Section 4 gives the conclusions.

2. Chaotic binary sequences generated by bernoulli map

Using a 1-D nonlinear difference equation defined by

$$x_{n+1} = \tau(x_n), \quad x_n \in I = [0, 1], \quad n = 0, 1, 2, \dots \quad (1)$$

where $x_n = \tau^n(x)$ (τ^n is the n -th iterate of the map and $x_0 = x$ is an initial value), we can generate a chaotic real-valued sequence $\{\tau^n(x)\}_{n=0}^{\infty}$. We transform such a real-valued sequence into a binary sequence $\{B(x_n)\}_{n=0}^{\infty}$ using a binary function $B(x) \in \{0, 1\}$. Then, the theoretical auto-correlation function of such a binary sequence $\{B(x_n)\}_{n=0}^{\infty}$ is defined by

$$\langle C(l; B) \rangle = \int_I (B(x) - \langle B \rangle)(B(\tau^l(x)) - \langle B \rangle) f^*(x) dx, \quad (2)$$

under the assumption that $\tau(x)$ has an invariant density function $f^*(x)$, where $\langle B \rangle$ denotes the expectation of the binary sequence $\{B(x_n)\}_{n=0}^{\infty}$ defined by

$$\langle B \rangle = \int_I B(x) f^*(x) dx. \quad (3)$$

Now, we define the Perron-Frobenius (PF) operator P_τ of the map τ with an interval $I = [0, 1]$ by [4]

$$P_\tau f(x) = \frac{d}{dx} \int_{\tau^{-1}([0,x])} f(u) du, \quad (4)$$

which can also be written as

$$P_\tau f(x) = \sum_{i=1}^{N_\tau} |g'_i(x)| f(g_i(x)), \quad (5)$$

where $g_i(x)$ is counter image of x . The theoretical auto-correlation function with PF operator for a binary sequence $\{B(x_n)\}_{n=0}^\infty$ can be expressed by

$$\langle C(l; B) \rangle = \int_I P_\tau^l \{ (B(x) - \langle B \rangle) f^*(x) \} (B(x) - \langle B \rangle) dx. \quad (6)$$

Through this paper, we use the Bernoulli map $\tau_B(x)$ defined by

$$\tau_B(x) = \begin{cases} 2x & (0 \leq x < \frac{1}{2}) \\ 2x - 1 & (\frac{1}{2} \leq x \leq 1), \end{cases} \quad (7)$$

which has the real-valued interval $I = [0, 1]$ and the uniform invariant density $f^*(x) = 1$.

Next, a threshold function with a threshold t is defined by

$$\Theta_t(x) = \begin{cases} 0 & (x < t) \\ 1 & (x \geq t). \end{cases} \quad (8)$$

Using the threshold functions, we define binary functions by

$$B_i^{(m)}(x) = \sum_{j=0}^{2^m-1} c_j^{(i)} \cdot \Theta_{\frac{j}{2^m}}(x) \quad (i = 1, 2, \dots, 2^{2^m}), \quad (9)$$

where $c_j^{(i)} \in \{-1, 0, 1\}$.

We theoretically evaluate the auto-correlation functions of $\{B_i^{(m)}(x_n)\}_{n=0}^\infty$ generated by the Bernoulli map with the help of PF operator. For general piecewise linear onto maps with $f^*(x) = 1$, we can get [7]

$$P_\tau \{ (\Theta_t(x) - \langle \Theta_t \rangle) \} = \frac{1}{a_r} (\Theta_{\tau(t)}(x) - \langle \Theta_{\tau(t)} \rangle), \quad (10)$$

which also gives

$$P_\tau^l \{ (\Theta_t(x) - \langle \Theta_t \rangle) \} = \frac{1}{a_r^l} (\Theta_{\tau^l(t)}(x) - \langle \Theta_{\tau^l(t)} \rangle), \quad (11)$$

where a_r is a slope of the map in the subinterval in which the threshold t exists. For the Bernoulli map, we have

$$P_{\tau_B} \{ (\Theta_t(x) - \langle \Theta_t \rangle) \} = \frac{1}{2} (\Theta_{\tau_B(t)}(x) - \langle \Theta_{\tau_B(t)} \rangle), \quad (12)$$

and

$$P_{\tau_B}^l \{ (\Theta_t(x) - \langle \Theta_t \rangle) \} = \frac{1}{2^l} (\Theta_{\tau_B^l(t)}(x) - \langle \Theta_{\tau_B^l(t)} \rangle). \quad (13)$$

For the binary function $B_i^{(m)}(x)$ of Eq. (9), we have

$$P_{\tau_B}^l \{ (B_i^{(m)}(x) - \langle B_i^{(m)} \rangle) \} = \frac{1}{2^l} \sum_{j=0}^{2^m-1} c_j^{(i)} (\Theta_{\tau_B^l(\frac{j}{2^m})}(x) - \langle \Theta_{\tau_B^l(\frac{j}{2^m})} \rangle), \quad (14)$$

which leads us to get

$$\langle C(l; B_i^{(m)}) \rangle = \frac{1}{2^l} \sum_{j=0}^{2^m-1} c_j^{(i)} \int_I (\Theta_{\tau_B^l(\frac{j}{2^m})}(x) - \langle \Theta_{\tau_B^l(\frac{j}{2^m})} \rangle) (B_i^{(m)}(x) - \langle B_i^{(m)} \rangle) dx. \quad (15)$$

Table I. Values of $c_j^{(i)}$ in Eq. (9).

j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$c_j^{(1)}$	0	0	1	0	-1	1	-1	0	1	-1	1	0	0	-1	0	1
$c_j^{(2)}$	1	0	0	-1	0	1	-1	0	1	0	0	0	-1	0	0	0
$c_j^{(3)}$	1	-1	1	0	-1	0	0	0	1	0	0	0	-1	0	1	-1
$c_j^{(4)}$	1	0	0	0	0	-1	0	0	1	0	0	-1	0	0	0	0
$c_j^{(5)}$	0	1	-1	1	0	0	-1	1	-1	0	0	0	1	0	-1	1
$c_j^{(6)}$	0	0	0	0	0	0	1	0	-1	1	-1	1	0	0	0	0

Table II. Normalized auto-correlation values ε_l ($l = 1, 2, 3$) for $B_i^{(4)}(x)$.

	ε_1	ε_2	ε_3
B_1	-1/4	1/16	0
B_2	1/4	1/16	0
B_3	-1/2	1/4	-1/16
B_4	1/2	1/4	1/16
B_5	-1/2	3/8	-1/8
B_6	1/2	1/4	1/8

Furthermore, noting that $\tau_B^l(\frac{j}{2^m}) = 0$ for ($l \geq m$), we obtain

$$P_{\tau_B}^l(B_i^{(m)}(x) - \langle B_i^{(m)} \rangle) = \frac{1}{2^l} \sum_{j=0}^{2^m-1} c_j^{(i)} (\Theta_0(x) - \langle \Theta_0 \rangle) = 0, \quad (l \geq m), \quad (16)$$

which gives

$$\langle C(l; B_i^{(m)}) \rangle = 0, \quad (l \geq m). \quad (17)$$

Thus, the normalized theoretical auto-correlation functions for all the binary functions $B_i^{(m)}(x)$ ($i = 1, 2, \dots, 2^{2^m}$) are given by

$$\hat{C}(l; B_i^{(m)}) = \frac{\langle C(l; B_i^{(m)}) \rangle}{\langle C(0; B_i^{(m)}) \rangle} = \begin{cases} 1 & (l = 0) \\ \varepsilon_l & (l = 1, 2, \dots, m-1) \\ 0 & (l \geq m). \end{cases} \quad (18)$$

For $m \leq 3$, there exists only a small number of binary functions, that is, only a small number of auto-correlation functions can be obtained. To find more different binary functions and more kinds of auto-correlation functions, we consider the case $m = 4$ in this paper. For $m = 4$, the binary functions $B_i^{(4)}(x)$ ($i = 1, 2, \dots, 2^{16}$) will be possible. Here, we consider six binary functions $B_1^{(4)}(x)$, $B_2^{(4)}(x)$, $B_3^{(4)}(x)$, $B_4^{(4)}(x)$, $B_5^{(4)}(x)$ and $B_6^{(4)}(x)$ (as shown in Fig. 1) whose values of $c_j^{(i)}$ in Eq. (9) are given in Table I. These six binary functions are chosen as examples giving different auto-correlation functions and it is easy to get $\langle B_i^{(4)} \rangle = \frac{1}{2}$ using $f^*(x) = 1$, that is $\langle C(0; B_i^{(4)}) \rangle = \frac{1}{4}$ ($i = 1, 2, \dots, 6$).

Then, we calculate the auto-correlation values ($\varepsilon_1, \varepsilon_2, \varepsilon_3$) for $B_i^{(4)}(x)$ ($i = 1, 2, \dots, 6$) by using Eq. (15). The values ($\varepsilon_1, \varepsilon_2, \varepsilon_3$) are shown in Table II. For example, for the binary function $B_4^{(4)}(x) = \Theta_0(x) - \Theta_{\frac{5}{16}}(x) + \Theta_{\frac{1}{2}}(x) - \Theta_{\frac{11}{16}}(x)$, $\langle C(l; B_4^{(4)}) \rangle$ ($l = 1, 2, 3$) are obtained as

$$\langle C(1; B_4^{(4)}) \rangle = \frac{1}{2} \int_0^1 \{(-\Theta_{\frac{5}{8}}(x) + \langle \Theta_{\frac{5}{8}} \rangle) - (\Theta_{\frac{3}{8}}(x) - \langle \Theta_{\frac{3}{8}} \rangle)\} (B_4^{(4)}(x) - \langle B_4^{(4)} \rangle) dx = \frac{1}{8}, \quad (19)$$

$$\langle C(2; B_4^{(4)}) \rangle = \frac{1}{2^2} \int_0^1 \{(-\Theta_{\frac{1}{4}}(x) + \langle \Theta_{\frac{1}{4}} \rangle) - (\Theta_{\frac{3}{4}}(x) - \langle \Theta_{\frac{3}{4}} \rangle)\} (B_4^{(4)}(x) - \langle B_4^{(4)} \rangle) dx = \frac{1}{16}, \quad (20)$$

$$\langle C(3; B_4^{(4)}) \rangle = \frac{1}{2^3} \int_0^1 \{(-\Theta_{\frac{1}{2}}(x) + \langle \Theta_{\frac{1}{2}} \rangle) - (\Theta_{\frac{1}{2}}(x) - \langle \Theta_{\frac{1}{2}} \rangle)\} (B_4^{(4)}(x) - \langle B_4^{(4)} \rangle) dx = \frac{1}{64}. \quad (21)$$

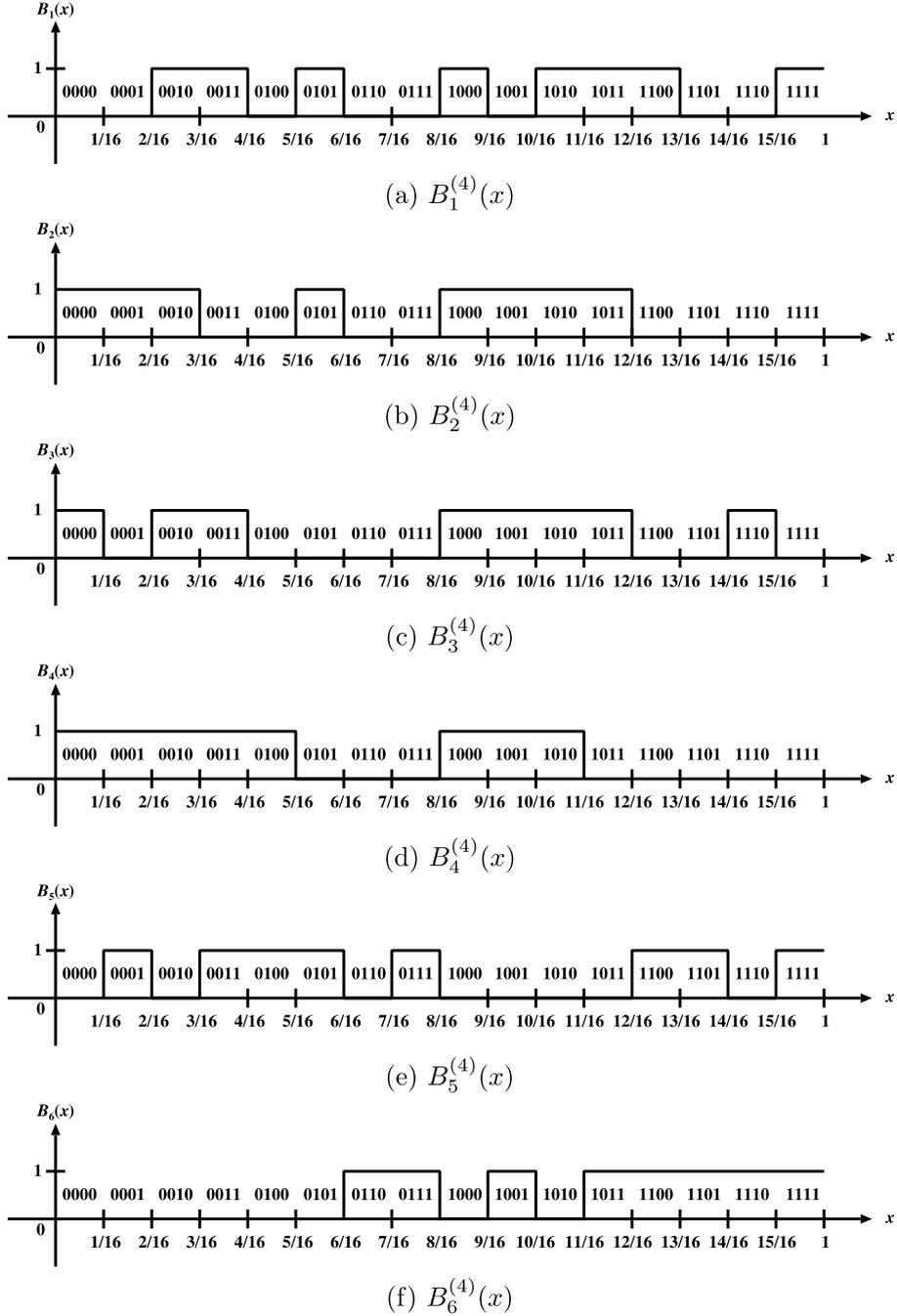


Fig. 1. Binary functions $B_i^{(4)}(x)$ ($i = 1, 2, \dots, 6$).

We can get the auto-correlation functions for the other binary functions in the similar calculation by using Eq. (15). Then, we get the normalized auto-correlation values ($\varepsilon_1, \varepsilon_2, \varepsilon_3$) for all the six binary functions as shown in Table II.

3. Proposed random bit generator and numerical experiments

Assume that we have a random bit generator (*e.g.* analog chaos circuit) generating an aperiodic random bit sequence $\{Y_n\}_{n=0}^{\infty}$. To generate aperiodic random bit sequences with designed auto-correlations, Y_n is post-processed by an LFSR. Figure 2 shows the proposed post-processing scheme based on a k -stage LFSR and a combinational logic circuit with 4 inputs (*i.e.* $m = 4$), where \oplus denotes modulo-2 addition, $h_i \in \{0, 1\}$. Since Y_n is aperiodic, the output $b_n^{(i)}$ is also aperiodic.

Note that the dynamics of the state of LFSRs approximates the Bernoulli map with finite bits as shown below [9, 10]. A state of LFSRs, denoted by $\{a_{k-1}(n), \dots, a_1(n), a_0(n)\}$, can be converted into

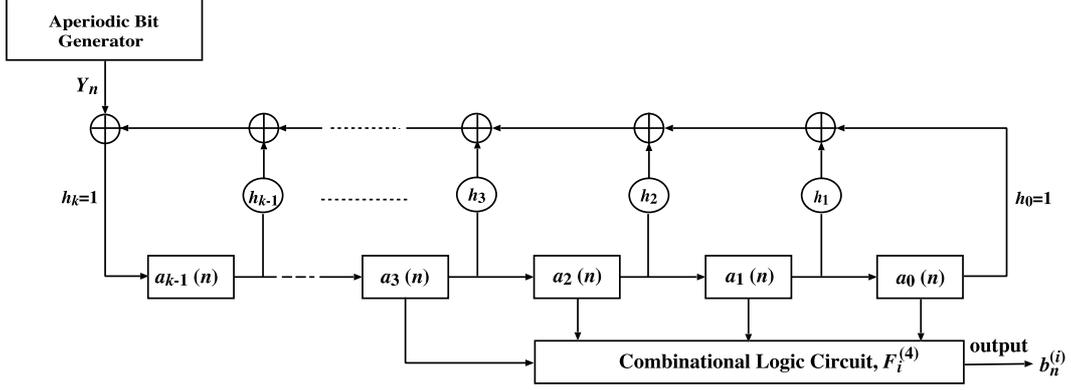


Fig. 2. Proposed random bit generator.

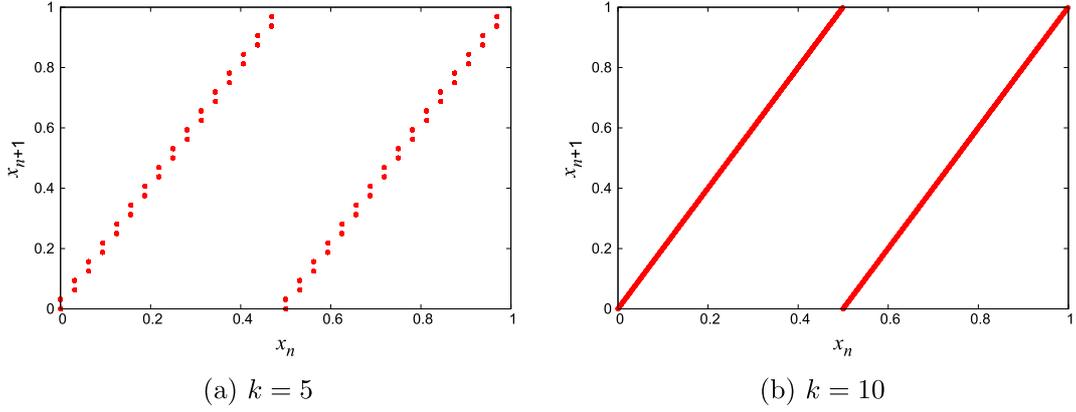


Fig. 3. Examples of return maps of k -stage LFSRs.

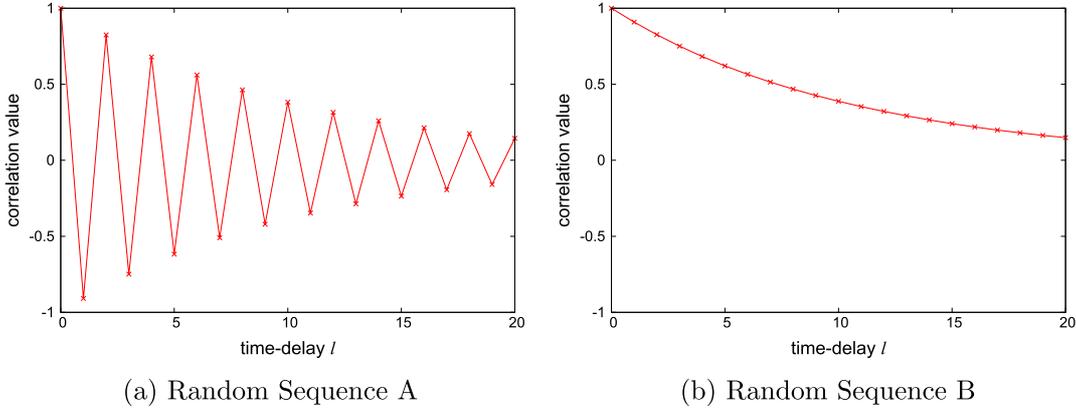


Fig. 4. Auto-correlation functions of Y_n before post-processing.

a real number $x_n \in [0, 1]$ by

$$x_n = a_0(n) \cdot 2^{-1} + a_1(n) \cdot 2^{-2} + \dots + a_{k-1}(n) \cdot 2^{-k}. \quad (22)$$

Plotting (x_n, x_{n+1}) , we can get a 1-D map (so-called a *return map*) as shown in Fig. 3, where (a) $k = 5$ and (b) $k = 10$. From Fig. 3, we can confirm that the shapes of such return maps are similar to the Bernoulli map defined by Eq. (7) and the precision increases as k increases [9, 10].

Then, the binary functions $B_i^{(4)}(x) (i = 1, 2, \dots, 6)$ are applied for designing the combinational logic functions with 4 inputs. Namely, we define six combinational logic functions by

$$F_i^{(4)}(a_0 a_1 a_2 a_3) = \begin{cases} 1 & a_0 a_1 a_2 a_3 \in M_i \\ 0 & \text{otherwise,} \end{cases} \quad (23)$$

where $M_i (i = 1, 2, \dots, 6)$ are given by

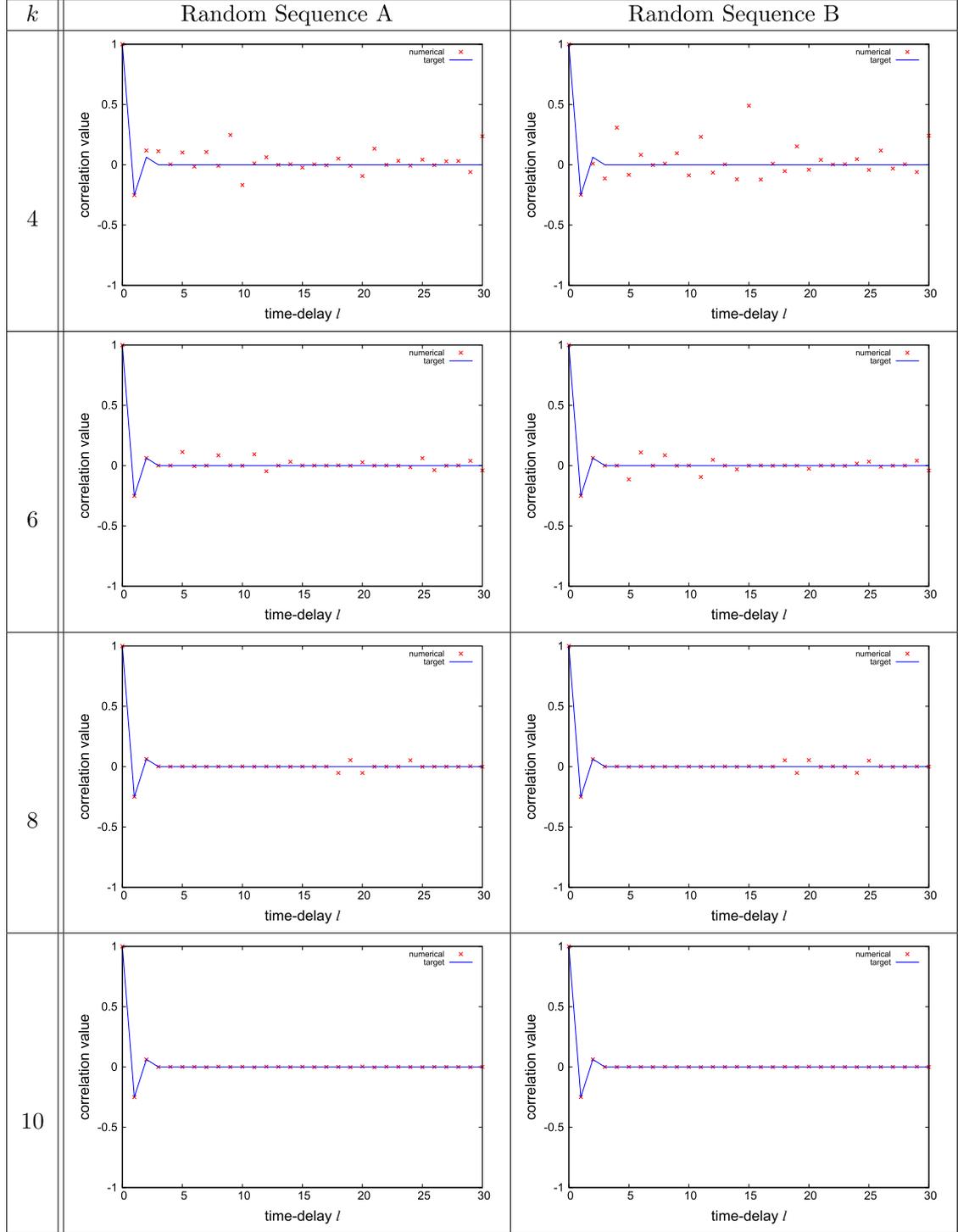


Fig. 5. Auto-correlation functions of $\{b_n^{(1)}\}_{n=0}^{N-1}$ after post-processing by $F_1^{(4)}$.

$$\begin{aligned}
M_1 &= \{0010, 0011, 0101, 1000, 1010, 1011, 1100, 1111\}, \\
M_2 &= \{0000, 0001, 0010, 0101, 1000, 1001, 1010, 1011\}, \\
M_3 &= \{0000, 0010, 0011, 1000, 1001, 1010, 1011, 1110\}, \\
M_4 &= \{0000, 0001, 0010, 0011, 0100, 1000, 1001, 1010\}, \\
M_5 &= \{0001, 0011, 0100, 0101, 0111, 1100, 1101, 1111\}, \\
M_6 &= \{0110, 0111, 1001, 1011, 1100, 1101, 1110, 1111\},
\end{aligned}$$

which are based on $B_i^{(4)}(x) (i = 1, 2, \dots, 6)$ in Fig. 1.

Now, we investigate the validity of the proposed post-processing by numerical experiments. The

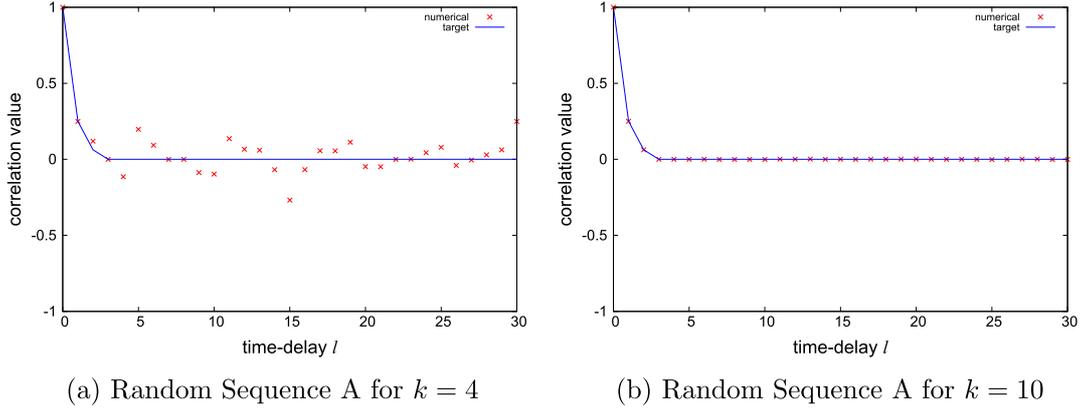


Fig. 6. Auto-correlation functions of $\{b_n^{(2)}\}_{n=0}^{N-1}$ after post-processing by $F_2^{(4)}$.

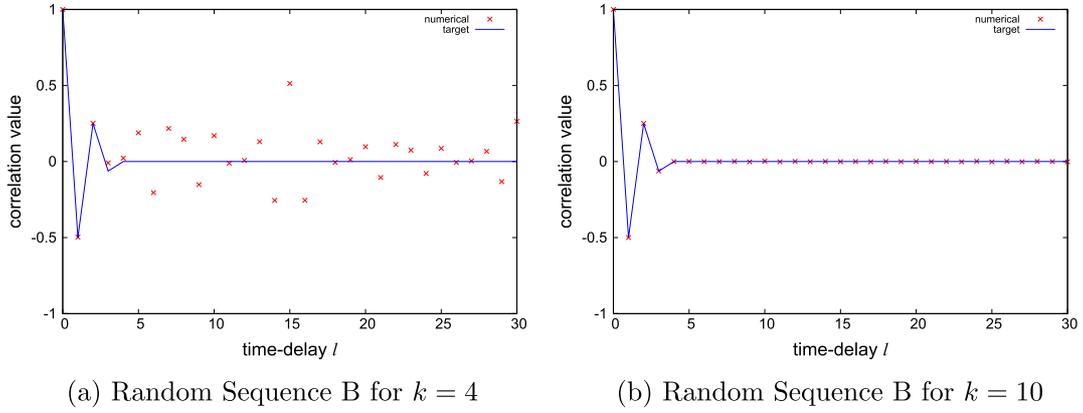


Fig. 7. Auto-correlation functions of $\{b_n^{(3)}\}_{n=0}^{N-1}$ after post-processing by $F_3^{(4)}$.

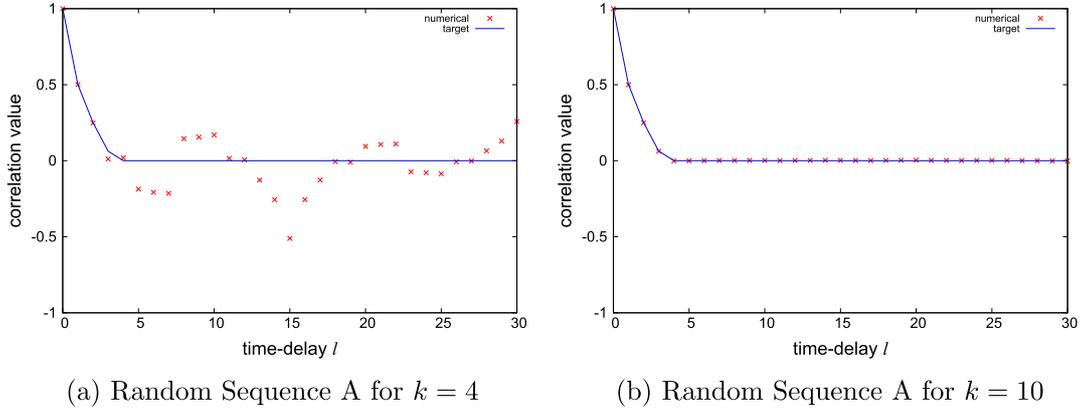


Fig. 8. Auto-correlation functions of $\{b_n^{(4)}\}_{n=0}^{N-1}$ after post-processing by $F_4^{(4)}$.

numerical auto-correlation function of the output random bit sequences $\{b_n^{(i)}\}_{n=0}^{N-1}$ ($i = 1, 2, \dots, 6$) is defined by

$$A_N(l; b_n^{(i)}) = \frac{1}{N} \sum_{n=0}^{N-1} (2b_n^{(i)} - 1)(2b_{n+l}^{(i)} - 1), \quad (24)$$

where we set $N = 1,000,000$ and the length (number of stages) of LFSRs is set to $k = 4, 5, \dots, 10$. Each LFSR itself generates an M-sequence with period $2^k - 1$. Here, we assume Y_n (before post-processing) is generated by an analog chaos circuit or a physical random number generator, which implies Y_n has infinite period. But, in general, such aperiodic sequences have *bad* statistical properties, that is, they are correlated. Thus, we also assume Y_n has strong auto-correlation given by a normalized auto-correlation function a^{-l} ($|a| > 1$) as shown in Fig. 4, where (a) Random Sequence A is a

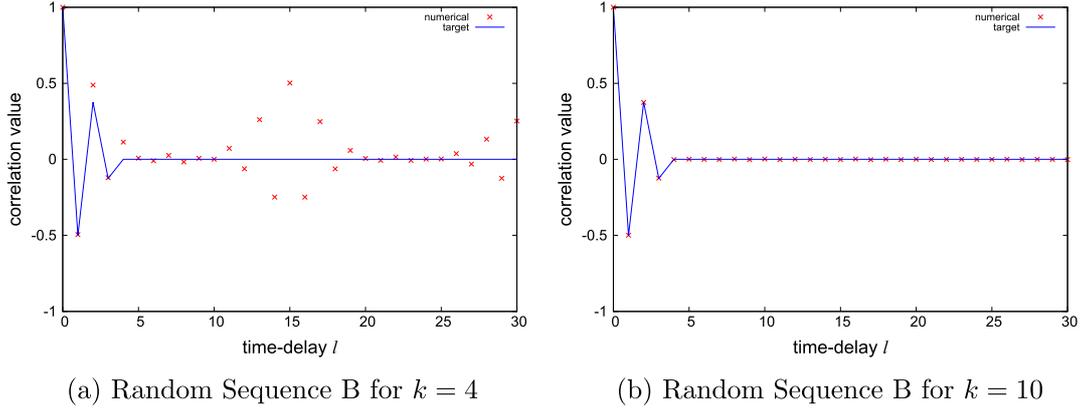


Fig. 9. Auto-correlation functions of $\{b_n^{(5)}\}_{n=0}^{N-1}$ after post-processing by $F_5^{(4)}$.

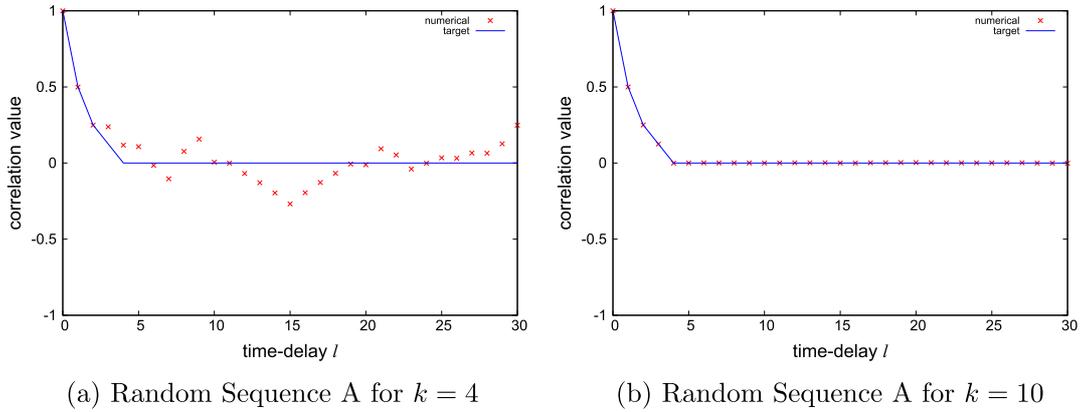


Fig. 10. Auto-correlation functions of $\{b_n^{(6)}\}_{n=0}^{N-1}$ after post-processing by $F_6^{(4)}$.

negatively auto-correlated sequence ($a = -1.1$) and (b) Random Sequence B is a positively auto-correlated sequence ($a = 1.1$). Such random sequences A and B are generated by using piecewise linear chaotic maps given in [7].

Figure 5 shows the auto-correlation functions of random bit sequences after post-processing by the LFSR with the combinational logic function $F_1^{(4)}$. In the figures, the lines represent the target (theoretical) auto-correlation functions and the crosses represent the numerical auto-correlation functions. According to the figures, if the length of LFSRs, k , is small, (*e.g.* $k = 4$), the numerical auto-correlation values are quite different from the target auto-correlation values. If k is increased, the difference between the numerical and target auto-correlation functions is getting small. If k is sufficiently large, (*e.g.* $k = 8, 9, 10$), the difference converges to zero. As shown in Fig. 5, for the case of $k = 10$, the numerical auto-correlation functions are almost the same as the target ones.

Similarly, Figs. 6–10 show the auto-correlation functions of random bit sequences after post-processing by the logic functions $F_i^{(4)}$ ($i = 2, 3, \dots, 6$) for the case of $k = 4$ and 10 respectively. By analyzing the resulting figures, it has been shown that if k increases, each numerical auto-correlation function is quite similar to the target auto-correlation function, which is independent of the statistical properties of the original random bit sequence Y_n .

Next, we investigate the convergence properties of auto-correlation functions of binary sequences generated by the proposed post-processing with LFSRs. We evaluate the mean square error (MSE) between $A_N(l; b_n^{(i)})$ and $\hat{C}(l; B_i^{(4)})$ defined by

$$MSE(i; k) = \frac{1}{L} \sum_{l=1}^L (A_N(l; b_n^{(i)}) - \hat{C}(l; B_i^{(4)}))^2, \quad (25)$$

where we set $L = 100$. Here we use several auto-correlation parameter a ($a = \pm 1.1, \pm 1.3, \pm 1.5, \pm 1.7, \pm 2.1, \pm 3, \pm 5, \pm 10$) for generating Y_n (Random Sequences A and B are generated by using

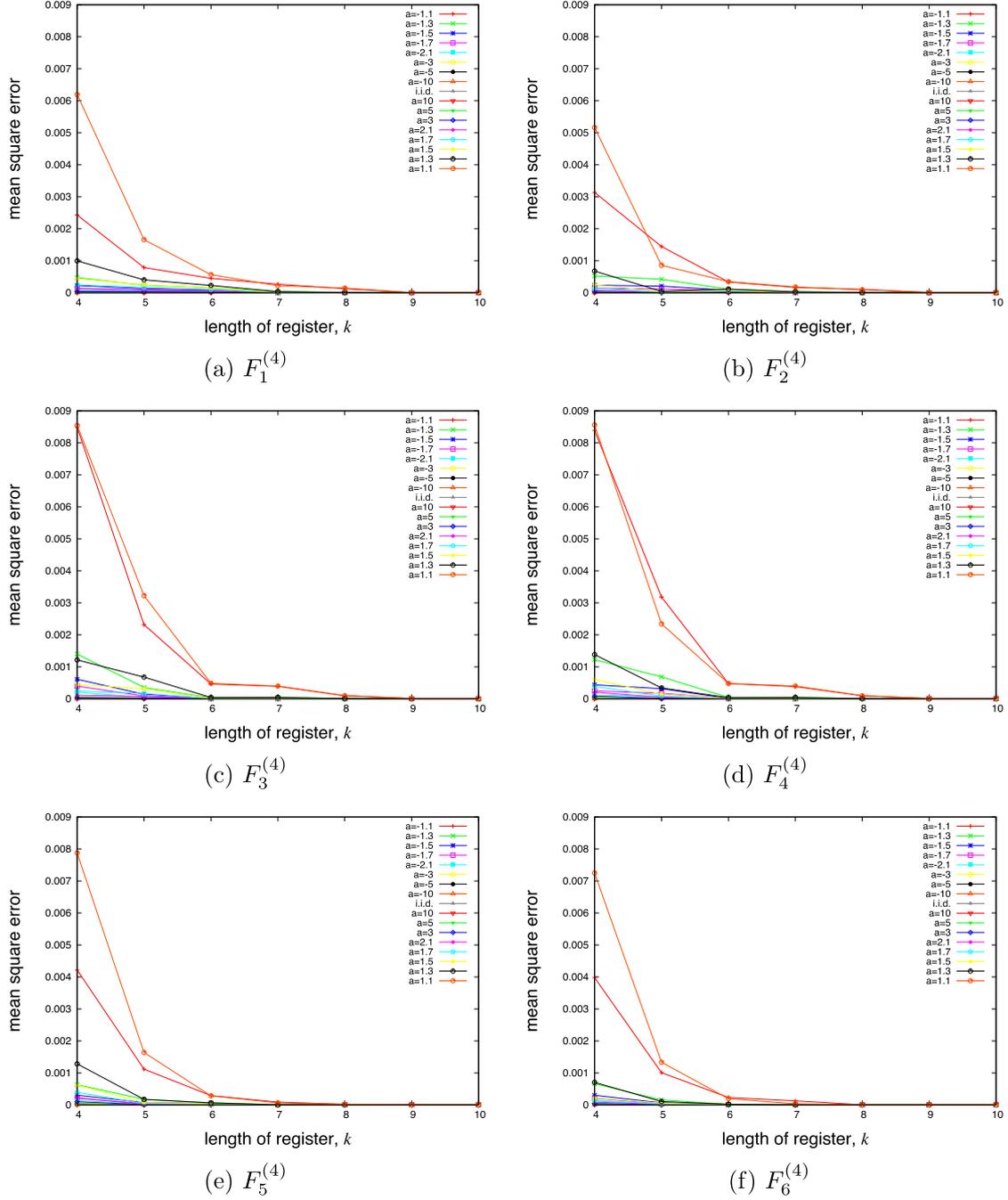


Fig. 11. $MSE(i; k)$ versus k for $F_i^{(4)}$ ($i = 1, 2, \dots, 6$).

$a = -1.1$ and 1.1). We also use *i.i.d.* binary sequences generated by the logistic map. Figure 11 shows the relation between the MSE and k for several values of a (and *i.i.d.*). From Fig. 11, it has been shown that the MSE converges to 0 as k increases and the convergence rate is slower for stronger auto-correlation (e.g., $a = \pm 1.1$).

Finally, we investigate the necessary value of k for achieving sufficiently small error between the numerical and target auto-correlation functions for each value of a . Table III shows the smallest value of k achieving $MSE(i; k) < 1 \times 10^{-5}$ for each a . According to the table, larger k is necessary for stronger auto-correlation and $k = 10$ is enough in this case ($|a| > 1.1$).

Table III. Smallest value of k achieving $MSE(i; k) < 1 \times 10^{-5}$.

a	-1.1	-1.3	-1.5	-1.7	-2.1	-3	-5	-10	<i>i.i.d.</i>	10	5	3	2.1	1.7	1.5	1.3	1.1
$F_1^{(4)}$	9	8	7	7	7	7	4	4	4	4	6	7	7	7	7	8	9
$F_2^{(4)}$	9	8	7	7	7	7	5	4	4	4	5	7	7	7	7	8	9
$F_3^{(4)}$	10	8	6	6	6	6	5	4	4	4	5	6	6	6	6	8	9
$F_4^{(4)}$	9	8	8	6	6	6	5	4	4	4	5	6	6	6	6	8	9
$F_5^{(4)}$	9	7	7	7	7	7	5	4	4	4	5	5	7	7	7	7	9
$F_6^{(4)}$	8	7	6	6	6	5	5	4	4	4	5	5	6	6	6	7	8

4. Conclusions

In this paper, we have given a generalization of the post-processing method based on k -stage LFSRs and chaos theory for the Bernoulli map to generate aperiodic random bit sequences with prescribed positive/negative auto-correlations. We have theoretically analyzed the auto-correlation functions of chaotic binary sequences $\{B_i^{(m)}(x_n)\}_{n=0}^{\infty}$ generated by the Bernoulli map. Using $B_i^{(4)}(x)$ ($i = 1, 2, \dots, 6$), we have designed the combinational logic circuits with 4 inputs for the post-processing. By numerical experiments, we have confirmed that if k is sufficiently large, random bit sequences with prescribed positive/negative auto-correlations can be obtained by the post-processing. In general, m -input/1-output combinational logic circuits can be designed and applied to the post-processing for $m \leq k$.

References

- [1] J.E. Gentle, *Random Number Generation and Monte-Carlo Method*, 2nd ed., Springer, 2003.
- [2] C.A. Yang, K. Yao, K. Umeno, and E. Biglieri, "Using deterministic chaos for superefficient Monte-Carlo simulations," *IEEE Circuits and Systems Magazine*, vol. 13, no. 4, pp. 26–35, 2013.
- [3] M.P. Kennedy, R. Rovatti, and G. Setti, Eds., "Chaotic electronics in telecommunications," Boca Raton, FL: CRC, 2000.
- [4] A. Lasota and M.C. Mackey, *Chaos, Fractals, and Noise*, 2nd ed., New York: Springer-Verlag, 1994.
- [5] T. Kohda and A. Tsuneda, "Statistics of chaotic binary sequences," *IEEE Trans. Information Theory*, vol. 43, no. 1, pp. 104–112, 1997.
- [6] A. Tsuneda, K. Eguchi, and T. Inoue, "Design of chaotic binary sequences with good statistical properties based on piecewise linear into maps," *Proc. of the 7th International Conference on Microelectronics for Neural, Fuzzy, and Bio-Inspired Systems*, pp. 261–266, 1999.
- [7] A. Tsuneda, "Design of binary sequences with tunable exponential autocorrelations and run statistics based on one-dimensional chaotic maps," *IEEE Trans. Circuits Syst. I*, vol. 52, no. 2, pp. 454–462, 2005.
- [8] F. Pareschi, R. Rovatti, and G. Setti, "Simple and effective post-processing stage for random stream generated by a chaos-based RNG," *Proc. of 2006 Int. Symp. Nonlinear Theory and its Applications*, pp. 383–386, 2006.
- [9] A. Tsuneda, S. Mitsuishi, and T. Inoue, "A study on generation of random bit sequences with post-processing by linear feedback shift registers," *International Journal of Innovative Computing, Information & Control*, vol. 4, no. 10, pp. 2631–2638, 2008.
- [10] A. Tsuneda and K. Morikawa, "A study on random bit sequences with prescribed auto-correlations by post-processing using linear feedback shift registers," *Proc. of 2013 European Conference on Circuit Theory and Design*, 2013.
- [11] E. Avaroğlu, T. Tuncer, A.B. Özer, B. Ergen, and M. Türk, "A novel chaos-based post-processing for TRNG," *Nonlinear Dynamics*, vol. 81, no. 1-2, pp. 189–199, 2015.
- [12] S. Loza, L. Matuszewski, and M. Jessa, "A random number generator using ring oscillators and SHA-256 as post-processing," *International Journal of Electronics and Telecommunications*, vol. 61, no. 2, pp. 199–204, 2015.