PRAMANA
— journal of
physics

Vol. 85, No. 4
October 2015
pp. 617–627

An efficient parallel pseudorandom bit generator based on an asymmetric coupled chaotic map lattice

RENFU LIANG, XUE TAN, HU ZHOU and SHIHONG WANG*

School of Sciences, Beijing University of Posts and Telecommunications, Beijing 100876, China
*Corresponding author. E-mail: shwang@bupt.edu.cn

MS received 29 August 2013; revised 04 June 2014; accepted 21 August 2014
DOI: 10.1007/s12043-014-0905-4; ***e*Publication:** 24 January 2015

Abstract. In this paper, an asymmetric coupled map lattice (CML) combining sawtooth map as a local map is presented and its chaotic behaviours are analysed. Based on this asymmetric CML, a pseudorandom bit generator (PRBG) is proposed. The specific parameters of the system that make complicated floating-point computation and multiplication computation transform into simple shift bit operations are adopted, that not only ensures the nonlinear operations, but also increases the performance efficiency. The PRBG is implemented in software and hardware. The parallel output bit sequences pass all of the NIST SP800-22 statistical tests.

Keywords. Coupled map lattice; spatiotemporal chaos; pseudorandom bit generator; parallel performance.

PACS Nos 05.45.Ra; 05.45.Gg

1. Introduction

Random number generators (RNGs) are widely used in cryptography, communication and Monte Carlo simulation. While there exist some physics methods generating true random numbers [1,2], most of the random numbers are generated by computers with various deterministic algorithms and these generators are called as pseudorandom number generators (PRNGs). Due to the random-like behaviour of chaos and the sensitivity of chaotic trajectories to the initial conditions and the parameters, chaos dynamics have been applied to cryptography, including encryption ciphers [3], hash functions [4], image encryptions [5,6], PRNGs etc. [7–12]. Recently, some PRNGs based on the coupled map lattice (CML) was proposed [13–16]. As a type of spatiotemporal chaos systems, CMLs have three advantages: high complexity (i.e., the systems have large numbers of positive Lyapunov exponents), long period [17] and parallel implementation.

In [16], a symmetric CML where logistic map was chosen as the local map was digitized to develop a highly parallel pseudorandom bit generator (PRBG) that can be used in hardware. The study shows that the chaotic behaviours of this system

depend on the coupling strength, local map parameter and the lattice size. With the given parameters [16], this CML system can induce temporal period and partial space synchronization.

In this paper, an asymmetric CML is proposed by combining a sawtooth map as the local map, the chaotic behaviours of which depend only on the local map parameter, but not on the lattice size and coupling strength. Thus, this asymmetric CML is more flexibly utilized in different system sizes. Based on the asymmetric CML, a PRBG is proposed. Besides flexibility, high complexity (spatiotemporal chaos) and long period, this PRBG has the added advantage of being simple. By choosing specified coupling strengths and map parameters, the complicated multiplication operations of the CML are transformed into simple bit operations, which ensures simplicity and high performance efficiency in hardware implementation. The rest of this paper is organized as follows. Section 2 introduces the CML and the asymmetric CML used in the proposed PRBG. In §3, the proposed PRBG is described in detail. Statistical analysis is given in §4. Finally, this paper is concluded in §5.

2. The coupled map lattices

The one-dimensional symmetric CML is defined as

$$x_{n+1}(j) = (1-\varepsilon) f(x_n(j)) + \frac{\varepsilon}{2}(f(x_n(j-1)) + f(x_n(j+1))), \quad j = 1, 2, ..., L, \tag{1}$$

where n is the time index, j and L are site index and lattice size, respectively and ε is the coupling strength. The periodic boundary conditions are utilized. The logistic map is usually used as the local map $f(x)$ in some cryptosystems based on the CML [13,14,16]. $f(x) = ax(1-x)$, where $x \in [0, 1]$ and $a \in [0, 4]$. If $a > 3.57$, the logistic map is chaotic.

The dynamic behaviours of eq. (1) depend on the systemic parameters, including the coupling strength ε, the local map parameter a and the lattice size L. Figures 1 and 2 show the dependence of the largest Lyapunov exponent of eq. (1) on the coupling strength ε at $L = 8$ and 16, respectively. In figures 1 and 2, three curves of dots, squares and circles correspond to three different conditions, arbitrarily chosen initial condition, the perturbed forward solutions as initial conditions with ε ranging from 0 to 1.0 and from 1.0 to 0, respectively. Figure 3 shows the dependence of the largest Lyapunov exponents of eq. (1) on the parameter a at different ε for $L = 16$. From figures 1–3 one can observe that there exist complicated behaviours of eq. (1) for the different systemic parameters, such as chaos, period and multiattractors.

For parallel performance, eq. (1) needs good mutual correlation between any two map outputs. Tables 1 and 2 present the spatiotemporal behaviours of eq. (1) vs. L at $\varepsilon = 0.5$ and 0.25, respectively. In the two tables, the parameter a is fixed as 4.0. From tables 1 and 2 it can be observed that under certain conditions, the different map outputs of eq. (1) may be synchronous or partially synchronous even though the dynamics behaviours are chaos. Given the parameters $L = 16$, $a = 4.0$ and $\varepsilon = 0.5$ in [16], the behaviour of eq. (1) is temporal period and partially synchronous (spatial

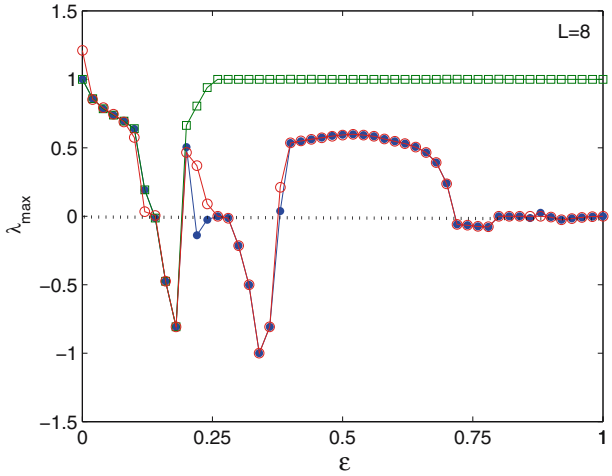

Figure 1. Dependence of the largest Lyapunov exponent of eq. (1) on ε at $L = 8$ and $a = 4$. Three curves correspond to three different initial conditions; the dots – arbitrarily chosen initial condition, the squares – the perturbed forward solutions as initial conditions with ε ranging from 0 to 1.0 and the circles – the perturbed forward solutions as initial conditions with ε ranging from 1.0 to 0.

ordering), thus eq. (1) using above parameters is not suitable to be used in the parallel environment.

By choosing the chaotic sawtooth map as a local map and asymmetric couplings, eq. (1) becomes

$$x_{n+1}(j) = (1 - \varepsilon_1 - \varepsilon_2) f(x_n(j)) + \varepsilon_1 f(x_n(j+1)) + \varepsilon_2 f(x_n(j-1)), \quad (2)$$

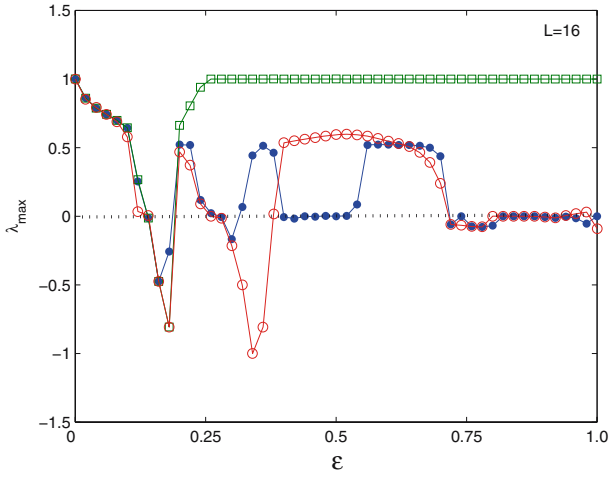

Figure 2. Dependence of the largest Lyapunov exponent of eq. (1) on ε at $L = 16$ and $a = 4$. The initial conditions of the three curves are the same as that of figure 1.

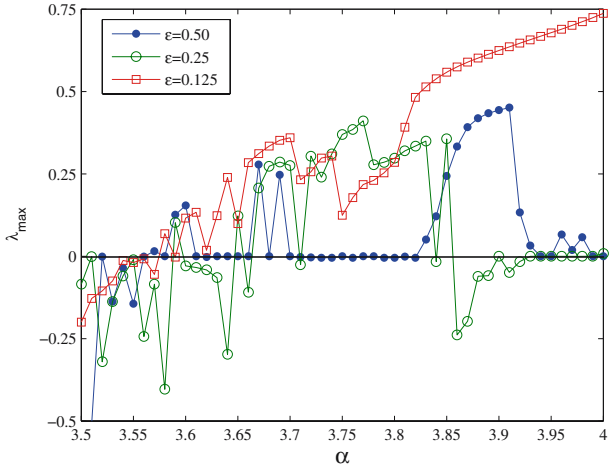

Figure 3. Dependence of the largest Lyapunov exponent of eq. (1) on a at three different couplings of ε and $L = 16$. The initial conditions are arbitrarily chosen.

Table 1. Dependence of the spatiotemporal behaviours of eq. (1) on the lattice size L. $\varepsilon = 0.5$ and $a = 4$.

L ($2 < L < 17$)	Space pattern	Time sequences
3	Completely synchronous	Chaos
5	Partially synchronous	Period
6	Partially synchronous	Quasiperiod
10	Partially synchronous	Period
11	Non-regular	Period
12	Near to partially synchronous, partially synchronous or non-regular	Period or quasiperiod
15	Partially synchronous or non-regular	Period or quasiperiod
16	Partially synchronous	Period
Others	Non-regular	Chaos

Table 2. Dependence of the spatiotemporal behaviours of eq. (1) on the lattice size L. $\varepsilon = 0.25$ and $a = 4$.

L ($2 < L < 21$)	Space pattern	Time sequences
4	Partially synchronous	Chaos
8	Partially synchronous	Chaos
12	Partially synchronous	Chaos
16	Partially synchronous or near to partially synchronous	Chaos
20	Partially synchronous or near to partially synchronous	Chaos
Others	Non-regular	Chaos

where $f(x) = ax \bmod 1.0$, $a \in (1, 2]$. $\varepsilon_1 > 0$, $\varepsilon_2 > 0$ and $\varepsilon_1 + \varepsilon_2 < 1.0$. The Jacobian matrix C of eq. (2) takes the form

$$\begin{bmatrix} (1-\varepsilon_1-\varepsilon_2)a & \varepsilon_1 a & 0 & \ldots & 0 & \varepsilon_2 a \\ \varepsilon_2 a & (1-\varepsilon_1-\varepsilon_2)a & \varepsilon_1 a & 0 & \ldots & 0 \\ 0 & \varepsilon_2 a & (1-\varepsilon_1-\varepsilon_2)a & \varepsilon_1 a & 0 & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \varepsilon_1 a & 0 & \ldots & 0 & \varepsilon_2 a & (1-\varepsilon_1-\varepsilon_2)a \end{bmatrix},$$

where the matrix C is a circulant matrix. The corresponding eigenvalues of the matrix C are given by $\lambda_j = a(1 - \varepsilon_1 - \varepsilon_2 + \varepsilon_1 \exp(2\pi i(j/L)) + \varepsilon_2 \exp(-2\pi i(j/L)))$, $j = 1, 2, ..., L$. We have the largest Lyapunov exponent $\lambda_{\max} = \ln(a)$. That shows the chaotic behaviour of eq. (2) only depends on the parameter a, not on the coupling parameters and system sizes, unlike eq. (1). Thus, the application of eq. (2) is more flexible. For $a \in (1, 2]$, $\lambda_{\max} = \ln(a) > 0$, eq. (2) is spatiotemporal chaos. Taking $a = 1.75$, $\varepsilon_1 = 3/32$ and $\varepsilon_2 = 1/32$, we further studied spatial ordering of eq. (2) for $3 \leq L < 17$ and could not find complete synchronization or partial synchronization. We also calculated autocorrelation coefficient of the sequences $x_n(j)$, $c_{j,j}(\tau)$, and mutual correlation coefficient between the two successive sequences $x_n(i)$ and $x_n(j)$, $c_{i,j}(\tau)$, according to the following formulas:

$$c_{j,j}(\tau) = \frac{\hat{c}_{j,j}(\tau)}{\hat{c}_{j,j}(0)}, \quad j = 1, 2, ..., L,$$

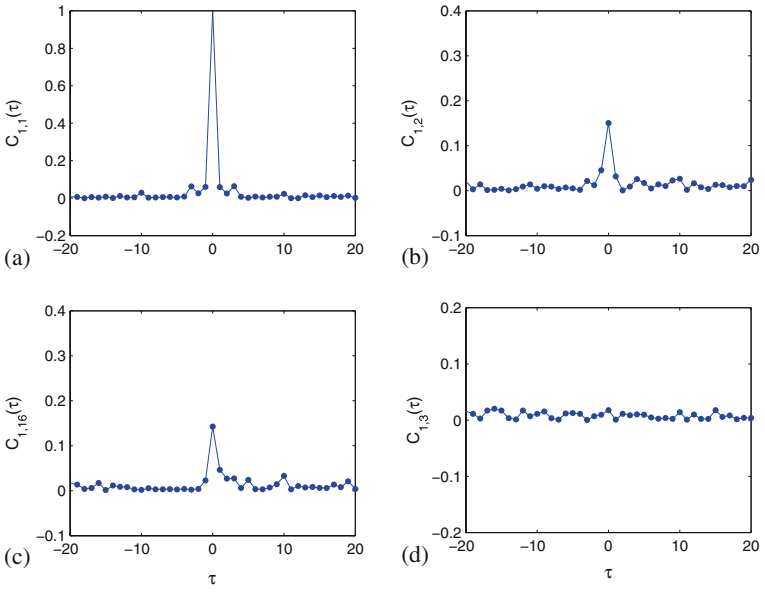

Figure 4. The autocorrelation curve $c_{1,1}(\tau)$ in (**a**) and the mutual correlation curves $c_{1,2}(\tau)$, $c_{1,16}(\tau)$ and $c_{1,3}(\tau)$ in (**b**), (**c**) and (**d**) at $L = 16$ and $T = 10^6$.

$$\hat{c}_{j,j}(\tau) = \frac{1}{T}\sum_{n=1}^{T}[x_n(j) - \bar{x}_n(j)][x_{n+\tau}(j) - \bar{x}_n(j)],$$

$$c_{i,j}(\tau) = \frac{\hat{c}_{i,j}(\tau)}{\sqrt{\hat{c}_{i,i}(0)\hat{c}_{j,j}(0)}}, \quad i, j = 1, 2, ..., L,$$

$$\hat{c}_{i,j}(\tau) = \frac{1}{T}\sum_{n=1}^{T}[x_n(i) - \bar{x}_n(i)][x_{n+\tau}(j) - \bar{x}_n(j)].$$

The curves of figures 4a–4d show the autocorrelation coefficient $c_{1,1}(\tau)$ and the mutual correlation coefficients $c_{1,2}(\tau)$, $c_{1,16}(\tau)$ and $c_{1,3}(\tau)$, respectively, at $L = 16$ and $T = 10^6$. In figure 4d, we observe the mutual correlation coefficient for non-adjacent site maps. We also observe the autocorrelation coefficient $c_{1,1}(\tau)$ for $|\tau| > 3$ in figure 4a and the mutual correlation coefficients $c_{1,2}(\tau)$ and $c_{1,16}(\tau)$ for adjacent site maps in figures 4b and 4c for $|\tau| > 1$.

3. A parallel PRBG based on the asymmetric CML

Based on the asymmetric CML combining some simple digital algebraic operations, a new PRBG was constructed, which can be used in hardware and software environment.

3.1 *Description of algorithm*

In this section, we provide a detailed description of the algorithm of the parallel PRBG. This algorithm is based on N coupled maps, $N \geq 5$.

This algorithm includes three parts: (i) Initialization process that expands an initial value (IV) from 64 bits to $32N$ bits and generates N initial variables of the asymmetric CML; (ii) dynamics of the asymmetric CML; (iii) output module giving random bit sequences.

3.1.1 *Initialization process.* This initialization process expands a IV from 64 bits to $32N$ bits and generates N initial variables of the asymmetric CML $x_0(i)$, $x_0(i) \in [0, 2^{32})$, $i = 1, 2, ..., N$. The expansion is made by nonlinear transformations such as hash functions. Here we define a nonlinear transformation

$$w(j + 8) = S(w(j) + w(j + 4)), \quad j = 1, 2, ..., 4N - 8, \tag{3}$$

where $w(j)$ are 8-bit integers and $w(1)$, $w(2)$,...,$w(8)$ compose 64-bit IV, i.e., IV $= w(1)\|w(2)\|...\|w(8)$. The operation $+$ of eq. (3) is a modular 2^8 addition. S is an S-box transformation, i.e., 8-bit to 8-bit nonlinear transformation. Here we consider S-box of AES.

After the expansion, the initial variables of the asymmetric CML are generated by a series of 8-bit $w(j)$ in the order, $x_0(1) = w(1)||w(2)||w(3)||w(4)$, $x_0(2) = w(5)||w(6)||w(7)||w(8)$,...,$x_0(N) = w(4N-3)||w(4N-2)||w(4N-1)||w(4N)$. If N initial variables are identical, they are perturbed by the transformation $x_0(i) = x_0(1) + 10000 \times i$, $i = 2, 3, ..., N$.

3.1.2 *Dynamics of the asymmetric CML.* An asymmetric CML is designed to perform bit confusion and diffusion between the state variables. The function reads as

$$x_{n+1}(j) = (1 - \varepsilon_1 - \varepsilon_2) f(x_n(j)) + \varepsilon_1 f(x_n(j+1)) + \varepsilon_2 f(x_n(j-1)), \quad (4)$$

where $\varepsilon_1, \varepsilon_2 \in (0, 1)$, $\varepsilon_1 \neq \varepsilon_2$. $f(x) = ax \bmod 2^{32}$, $a \in (1, 2]$ and x is defined as a finite integer domain $[0, 2^{32})$, which performs stretching and bending operations of chaotic system, as also, helps in hardware implementation. The periodic boundary conditions are used. If the iteration time $n > 100$, eq. (4) outputs the variables $x_n(j)$, $j = 1, 2, ..., N$.

In eq. (4), the multiplication computations require more time in hardware implementation. Thus, we may choose specified coupling strengths ε_1, ε_2 and the parameter a and let complicated multiplication operations transform into simple bit-shift operations. If we take $a = 1\frac{3}{4}$, $\varepsilon_1 = \frac{3}{32}$ and $\varepsilon_2 = \frac{1}{32}$, eq. (4) becomes

$$\begin{aligned} x_{n+1}(j) &= (1 - \varepsilon_1 - \varepsilon_2) X_n(j) + \varepsilon_1 X_n(j+1) + \varepsilon_2 X_n(j-1) \\ &= (X_n(j) \gg 1) + (X_n(j) \gg 2) + (X_n(j) \gg 3) \\ &\quad + (X_n(j+1) \gg 4) + (X_n(j+1) \gg 5) + (X_n(j-1) \gg 5), \quad (5) \end{aligned}$$

$$\begin{aligned} X_n(j) &= 1\frac{3}{4} x_n(j) \bmod 2^{32} \\ &= (x_n(j) + (x_n(j) \gg 1) + (x_n(j) \gg 2)) \bmod 2^{32}, \quad (6) \end{aligned}$$

where the operation $x \gg y$ stands for a right shift of the variable x by y bits.

Figures 5 and 6 present the performance of eq. (6) (i.e., the local map performance) and the parallel performance of eq. (5).

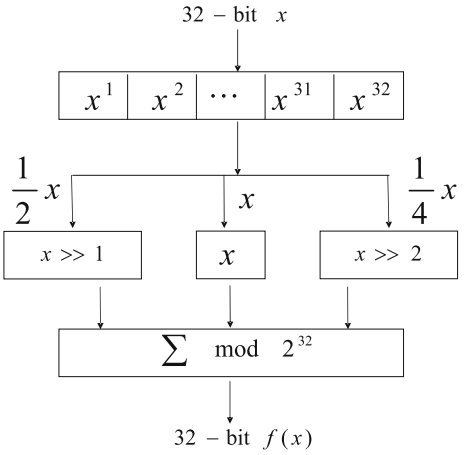

Figure 5. The local map performance of eq. (6).

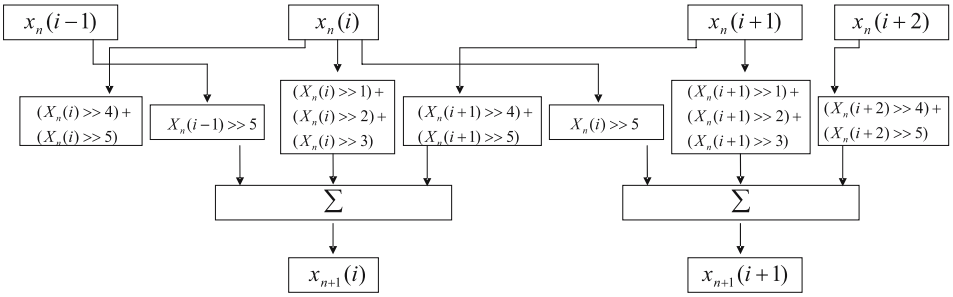

Figure 6. The parallel performance of eq. (5).

3.1.3 *Output module.* The output module transforms the variables of the CML, $x_n(j)$, into the binary format and outputs the partial bits.

$$x_n(j) = \sum_{i=1}^{32} b_n^i(j) \times 2^{32-i}, \quad b_n^i(j) \in \{0, 1\}.$$

For each map, we output 16 low significant bits $b_n^{17}(j)b_n^{18}(j), \ldots, b_n^{32}(j)$.

3.2 *Characteristics of the algorithm*

The structure of the proposed algorithm has the following characteristics:

(i) The parallel PRBG is mainly based on the asymmetric coupled chaotic map lattice. This structure yields strong confusion and diffusion rates among the state variables of all site maps $x_n(i)$, and is suitable for parallel implementation.

(ii) The complicated arithmetic operations are transformed into simple bit operations. In eq. (4), floating-point and multiplication computations need more operations in hardware implementation. In our algorithm, we adopt the specific parameters ε_1, ε_2 and a, which transform multiplication and floating-point computations in eq. (4) into simple bit shifts and additions in eqs (5) and (6). This ensures nonlinear operations, as also increases the efficiency of the system in hardware implementation. Table 3 shows the performance operations.

(iii) After processing 100 iterations, the algorithm outputs bit sequences. This rule assures strong confusion and diffusion among 64-bit IV.

(iv) The size of CML in eq. (4) and the maximal sequence length. The previous works [18–21] have recommended that the state size of a key-stream (here pseudorandom

Table 3. The operations of eqs (4)–(6) for each map.

	Multiplication (32-bit)	Addition (32-bit)	Shift (32-bit)	Modulo (2^{32})
Eq. (4)	4	4	0	1
Eqs (5) and (6)	0	7	8	1

sequence) generator should be at least twice or 2.5 times the key size (here IV size), to protect against what is now usually called the Babbage–Golic TMD attack and Biyukov–Shomir TMD attack, respectively. We take the size of CML in eq. (4) to be $N \geq 5$ because of the 64-bit size of IV. The entropy decreases consistently at each iteration output. For a random system state, if a 2^m-bit-length sequence is the output, the entropy decreases about m-bit. Due to the 64-bit IV size, we suggest that 128-bit entropy remains after the sequences output. Thus, we gave a constructive sequence length of 2^{32} bits for $5 \leq N < 8$ and 2^{64} bits for $N \geq 8$.

4. Statistical analysis

4.1 *Statistical analysis of output sequences*

In this section, randomness tests are explained by using NIST 800-22 test suite (Revision 1) [22]. The NIST 800-22 test suite issued by NIST (National Institute of Standard Technologies) of the United States is a statistical package used for testing the randomness of bit sequences generated from pseudorandom number generators. Table 4 shows the 15 items of NIST 800-22 Test Suite and the parameters used in this paper. Each test item produces the corresponding P_{values}. To interpret these results of P_{values}, two approaches have been adopted [22]: the examination of the proportion of sequences passing a statistical test and the uniform distribution examination of P_{values}. In the sequences' proportion examination, given a significance level α and a sample size of sequences m, a confidence interval can be obtained. If the proportion falls within this interval, then the data achieve satisfactory randomness. In our analysis, $\alpha = 0.01$ and $m = 1000$, and the confidence interval is [0.9805608, 0.9994392]. In the uniform distribution examination,

Table 4. Fifteen items of NIST 800-22 Test Suite (Revision 1) and the parameters used in this paper.

No.	Test type	Parameter
1	The frequency (monobit) test	
2	Frequency test within a block	Block size $m = 128$
3	The runs test	
4	Tests for the longest-run-of-ones in a block	$M = 10000$
5	The binary matrix rank test	$M = 32$, $Q = 32$
6	The discrete Fourier transform (spectral) test	$n = 50000$
7	The non-overlapping template matching test	$m = 9$
8	The overlapping template matching test	$m = 10$, $M = 2057$
9	Maurer's universal statistical test	
10	The linear complexity test	$M = 1000$
11	The serial test	$m = 10$
12	The approximate entropy test	$m = 10$
13	The cumulative sums (Cusums) test	
14	The random excursions test	
15	The random excursions variant test	

a parameter P_{valueT} was calculated (§4.2.2 in [22]). If $P_{valueT} \geq 0.0001$, then the sequences can be considered to be uniformly distributed. To provide statistically meaningful results, 1000 sequences were processed in this analysis.

We analysed the system of eqs (5) and (6) with $N = 8$. By arbitrarily giving a 64-bit IV, we parallel output 1000 bit sequences for each site, each of which is 1000,000-bit length. The dataset of the first site map passes all the NIST statistical tests, and we obtained similar results for the other site maps. We also constructed the XORed sequence by XOR processing two sequences from adjacent sites. The XORed sequences were also observed to pass all the requisite statistical tests, which further confirms the independence of the sequences of different site maps.

4.2 *Statistical analysis of IV*

For a PRBG, different IVs should output independent binary sequences. In our design we take two approaches to enhance the sensitivity of IVs.

First, the initialization process expands a 64-bit IV to $32N$ bits and generates N variables $x_0(i)$. For simplicity we assume that the lowest significant bit of $w(4)$ in a 64-bit IV is changed, according to eq. (3) we have changed $w(12)$, $w(16)$,...,$w(4j)$, $j = 3, 4, ..., N$. Thus, one-bit change of a 64-bit IV can result in $8(N - 2) + 1$ bits change for $32N$ bits. These changed expanded bits induce changed $x_0(1)$, $x_0(3)$, $x_0(4)$,...,$x_N(i)$ except $x_0(2)$.

Second, in the dynamics of asymmetric CML of eq. (4), if the iteration time $n > 100$, eq. (4) outputs the variables $x_n(i)$ that ensures effective bit diffusion and confusion among the initial variables $x_0(1)$, $x_0(2)$, $x_0(3)$,...,$x_N(i)$.

Our simulation results verify the above analysis and show satisfied sensitivity of IVs.

5. Conclusion

We proposed a new parallel pseudorandom bit generator based on a coupled chaotic map lattice. We adopted the specific systemic parameters that transformed complicated floating-point computation into simple bit operations. This not only ensured the complexity of nonlinear operations, but also enhanced the efficiency of the system. Statistical tests showed that the output sequences have satisfied random characteristics.

Acknowledgements

This work was supported by National Natural Science Foundation of China under No. 60973109.

References

[1] H Guo, W Tang, Y Liu and W Wei, *Phys. Rev. E* 81, 051137 (2010)
[2] X Li, A B Cohen, T E Murphy and R Roy, *Opt. Lett.* **36**(**6**), 1020 (2011)
[3] L M Cuomo and A V Oppenheim, *Phys. Rev. Lett.* **71**, 65 (1993)

[4] S H Wang and G Hu, *Information Sci.* **195**, 266 (2012)
[5] G Chen, Y Mao and C K Chui, *Chaos, Solitons and Fractals* **21**, 749 (2003)
[6] N Bigdeli, Y Farid and K Afshar, *Comput. Electr. Eng.* **38**, 356 (2012)
[7] J A Gonzalez and R Pino, *Comput. Phys. Commun.* **120**, 109 (1999)
[8] T Stojanovski and L Kocarev, *IEEE Trans. Circuits Syst.* **48**, 281 (2001)
[9] V Patidar, K K Sud and N K Pareek, *Informatica* **33**, 441 (2009)
[10] A Kanso and N Smaoui, *Chaos, Solitons and Fractals* **40**, 2557 (2009)
[11] S Behnia, A Akhavan and A A Samsudin, *J. Comput. Appl. Math.* **235**, 3455 (2011)
[12] N Liu, *Commun. Nonlinear Sci. Numer. Simulat.* **16**, 761 (2011)
[13] H P Lu, S H Wang and G Hu, *Int. J. Mod. Phys. B* **18**, 2409 (2004)
[14] P Li, Z H Li, W A Halang and G Chen, *Phys. Lett. A* **349**, 467 (2006)
[15] S H Wang and D Li, *Chin. Phys. B* **19**, 080505 (2010)
[16] Y B Mao, L Cao and W B Liu, *Circuits and Systems Proceedings, International Conf.* **3**, 2114 (2006)
[17] S H Wang, H P Lu and G Hu, *Int. J. Mod. Phys. B* **18**, 2617 (2004)
[18] S Babbage, *IEE Conf. Publication* **408**, 161 (1995)
[19] S Babbage, European Convention On Security And Detection, *IEE Conf. Publication* **408** (1995)
[20] J Golic, *Proc. EURCRYPT'97 LNCS 1233*, **239** (Springer-Verlag, 1997)
[21] A Biryukov and A Shamir, *Asiacrypt 2000 LNCS 1976 1* (Springer-Verlag. 2000)
[22] A Rukhin *et al*, *A statistical test suite for random and pseudorandom number generators for cryptographic applications*, Special Publication 800-22 Revision 1 (August 2008)