Journal of Computers Vol. 28, No. 6, 2017, pp. 100-113
doi:10.3966/199115992017122806009

Improved Algorithm for Digital Chaos Sequences in Full Mapping

Shuai Chen[1*], Shui-feng Zhang[2]

[1] School of Electronic Engineering, Huainan Normal University,
Anhui 232038, China
Chen232001@126.com

[2] School of Electronic Engineering, Huainan Normal University,
Anhui 232038, China
253537334@qq.com

Received 16 July 2016; Revised 07 February 2017; Accepted 07 February 2017

Abstract. For convenient calculation and improvement of processing speed, the calculation method for a one-dimensional discrete chaotic sequence is improved, and an integer calculation of fully digital chaos sequences is proposed. Logistic chaotic variable substitution and adjustable parameters are introduced. Calculation results are achieved through a digital chaotic equation. Analysis shows that the method for calculating the resulting sequence is chaotic, whereas the method that employs an integer and avoids floating-point arithmetic is convenient and increases speed. The integer chaos is easy to identify in digital circuits.

Keywords: algorithm, analysis, chaos, digitization, improvement

1 Introduction

Chaotic sequences have numerous applications [1-3]. When a chaotic sequence is utilized in a secret communication [4-7], only a few parameters are necessarily transmitted. The characteristics of chaotic sequences include a similar random process, similar white noise, and extreme sensitivity to the initial value [8-9]. Generally, the use of floating-point digital chaos to calculate sequences according to the chaotic formula consumes an increased amount of computational resources. Numerous improved chaotic systems have been proposed [10-11]. The use of floating-point arithmetic improves accuracy. However, floating-point arithmetic is slow, requires several clock cycles, and entails inconvenient chip implementation. Under the condition of maintained precision, if digital chaos is used with integers, then the computation speed will increase.

In order to improve the computing speed, this paper improved the calculation method of one-dimension Logistic chaotic. Due to the characteristic of full mapping is good [12], the paper [13] analyzes the many bits digital sequence pseudo random characteristics, so the improved full map chaos in integer calculation are put forward in this paper. Through inspection the modified algorithm remains chaotic characteristics. The design and hardware implementation of the improved method has been completed, which can be realized through hardware quick bits of digital chaos.

The relative work was introduced in section 2. The improved algorithm is in section 3 and was analyzed in section 4. The hardware implementation of the modified algorithm was implemented in section 5. Conclusions were drawn in section 6.

* Corresponding Author

2 Relative Work

2.1 Logistic Mapping

The logistic mapping is shown as follows [14, 15, 16]:

$$x_{n+1} = \mu x_n(1-x_n), (0 \le x_n \le 1, 0 \le \mu \le 4), \tag{1}$$

where x_n is the iterating value by n times, x_{n+1} is the iterated value by $n+1$ times, and μ is a parameter. If

$$x_n = (\frac{\mu}{4} - 0.5) y_n, \tag{2}$$

$$\lambda = \frac{\mu^2}{4} - \frac{\mu}{2}, \tag{3}$$

where y_n is a variable, then Eq. (1) can be converted as

$$y_{n+1} = 1 - \lambda y_n^2, \tag{4}$$

or

$$x_{n+1} = 1 - \lambda x_n^2, \tag{5}$$

where parameter $0 \le \lambda \le 2$, and $x_n \in (-1, 1)$. Full mapping is when $\lambda = 2$ or $\mu = 4$.

2.2 Iterating Roots of Logistic Mapping [15, 17]

In the iteration process of calculation x_n, the iteration terminates when $x_{n+1} = x_n$. Thus, the x_n goes into an unchanged state with n, and the roots are called constant or fixed-point roots. In a single-turn iteration condition, the roots address the formula

$$x = f(x) = \mu(1-x). \tag{6}$$

The $f(x)$ is a function of the variable x. By solving the equation, the roots are

$$\begin{cases} x_{01} = 0 \\ x_{02} = 1 - \dfrac{1}{\mu} \end{cases}. \tag{7}$$

where the x_{01} and x_{02} are roots of the equation (6). The constant roots are the intersection points of the curves

$$\begin{cases} y = x \\ y = \mu x(1-x) \end{cases}. \tag{8}$$

evidently, (0, 0) point is one of the roots.

Differentiating the formula $y = \mu x(1-x)$, then

$$\frac{dy}{dx} = \mu - 2\mu x. \tag{9}$$

If $\mu < 1$, then only one intersection point (0, 0) exists for the two curves, as shown in Fig. 1. When $\frac{dy}{dx}|_{x=0} = (\mu - 2\mu x)|_{x=0} = \mu > 1$, then two intersection points exist for the two curves, as shown in Fig. 2.

The apex of the parabola $y = \mu x(1-x)$ is $(1/2, \mu/4)$. When $\mu = 2$, the line with 45° is across the apex of the parabola.

The iteration in two turns is

$$x = f(f(x)) = \mu[\mu x(1-x)][1-\mu x(1-x)], \tag{10}$$

which can be expressed as

$$x = f^2(x). \tag{11}$$

Its root is called the root for two cycles. The two-cycle root should address

$$x[x-(1-\frac{1}{\mu})][\mu^2 x^2 - (\mu^2+\mu)x + (\mu+1)] = 0, \tag{12}$$

$$\begin{cases} x_1 = 0 \\ x_2 = 1 - \dfrac{1}{\mu} \\ x_{3,4} = \dfrac{(\mu+1) \pm \sqrt{(\mu+1)(\mu-3)}}{2\mu} \end{cases}. \tag{13}$$

The root for two-cycle roots visibly contains the stable root for a single-turn iteration. Using Eq. (13), μ must be greater than 3 to guarantee that $x_{3,4}$ is real.

In the same way, three-cycle roots may be calculated. Let

$$\begin{aligned} x &= f[f[f(x)]]] = f^3(x) \\ &= \mu\{\mu[\mu x(1-x)][1-\mu x(1-x)]\}\{1-\mu[\mu x(1-x)][1-\mu x(1-x)]\}. \end{aligned} \tag{14}$$

The three-cycle root should address

$$\begin{aligned} &\mu^6 x^6 - (3\mu+1)\mu^5 x^5 + (3\mu+1)(\mu+1)\mu^4 x^4 \\ &-(\mu^3+5\mu^2+3\mu+1)\mu^3 x^3 + (2\mu+1)(\mu^2+\mu+1)\mu^2 x^2 \\ &-(\mu+1)(\mu^2+\mu+1)\mu x + (\mu^2+\mu+1) = 0 \end{aligned}. \tag{15}$$

The rules of the logistic mapping may be summarized as follows:

(1) When $\mu \le 1$, the curve $y = \mu x(1-x)$ may obtain the iteration value y_1 (point A) starting from a random initial $0 < x_0 \le 1$, as shown in Fig. 1. Given that the iteration value is x_1, then the horizontal line intersects the line $y = x$ in point B. Then, x_1 is used in the iteration formula to obtain x_2, that is, the vertical line intersects the parabola in point C. Given the iteration process, the point gradually becomes close to zero (0, 0).

(2) When $1 < \mu \le 3$, the iteration process eventually ends in fixed point P2 $(1-1/\mu, 1-1/\mu)$, as shown in Fig. 2.

(3) When $3 < \mu < 3.449$, the iteration process alternately appears in two cycles, as shown in Fig. 3.

(4) When $3.449 < \mu < 3.57$, the iteration process appears in a four-, eight-, …, and so on cycle point.

(5) When $3.57 \le \mu$, the iteration process appears chaotic, endless, and with infinite cycles points, as shown in Figs. 4 and Fig. 5.

Fig. 6 shows the sequence in full mapping (μ = 4 or λ = 2.0).

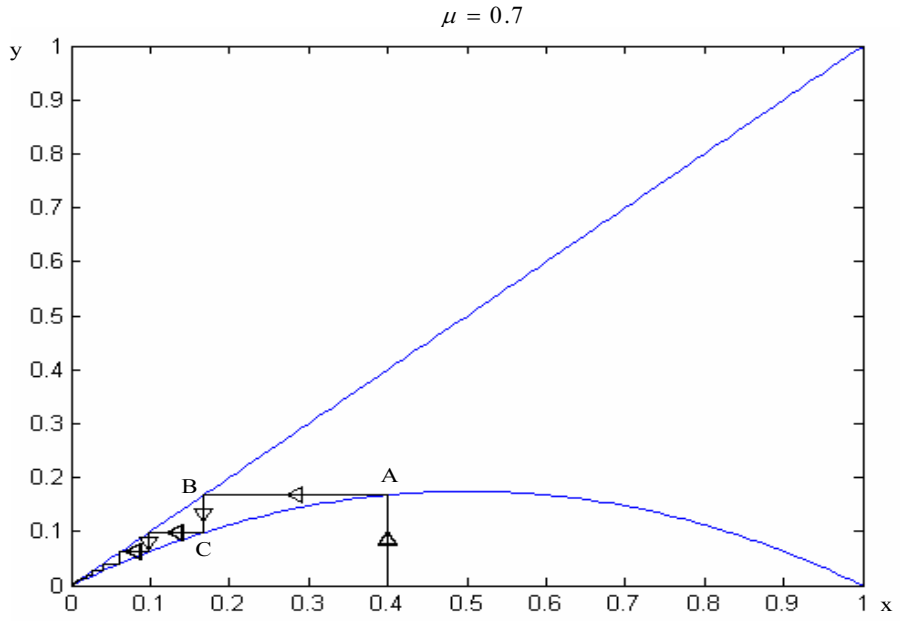

Fig. 1. Iteration process for logistic mapping (μ = 0.7)

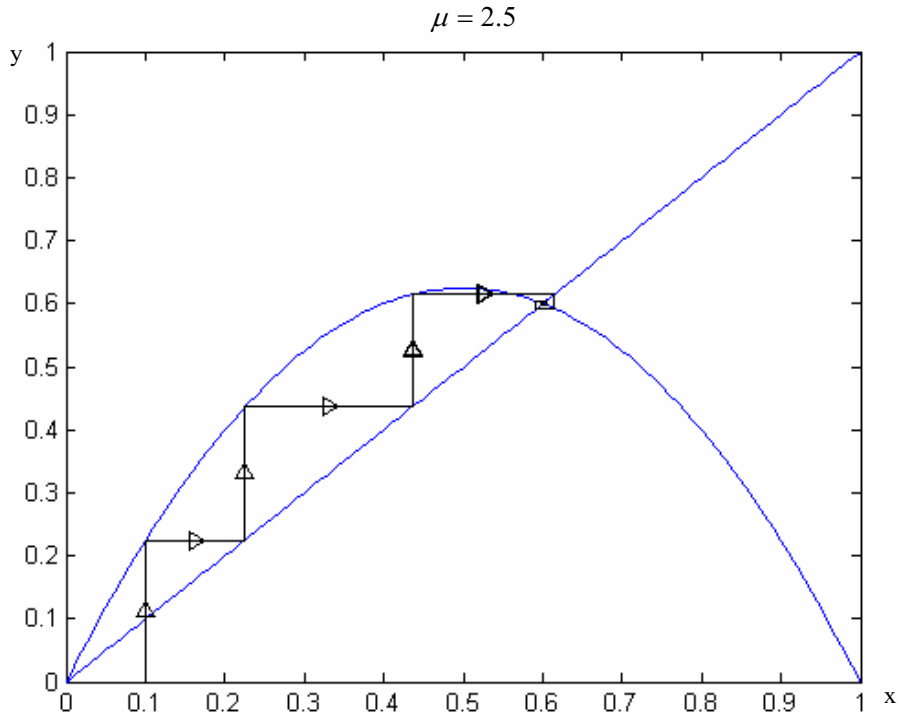

Fig. 2. Iteration process for logistic mapping (μ = 2.5)

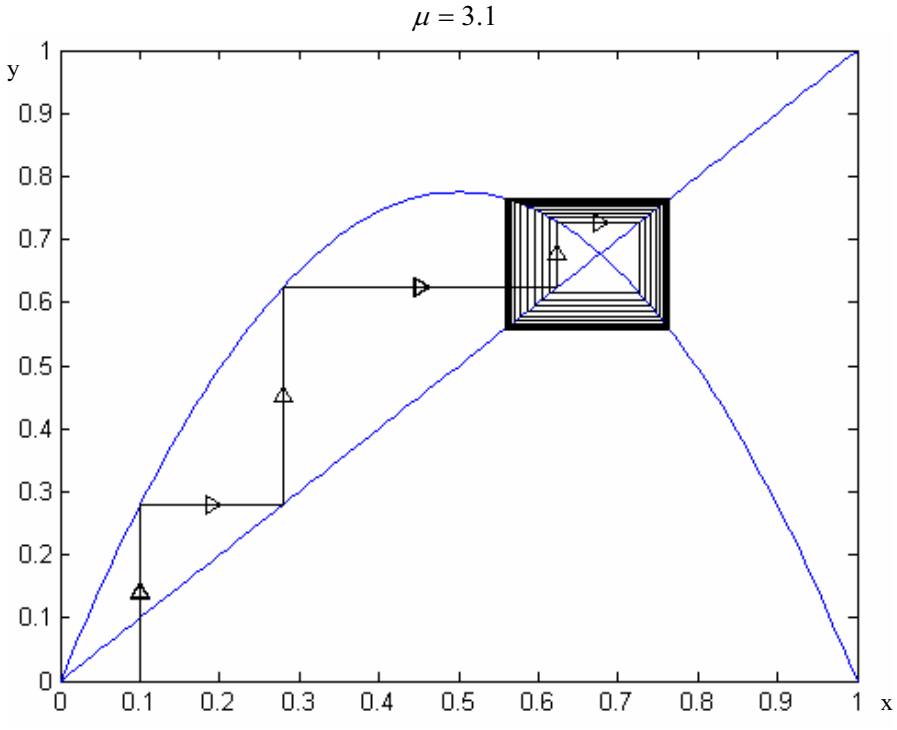

Fig. 3. Iteration process for logistic mapping (μ = 3.1)

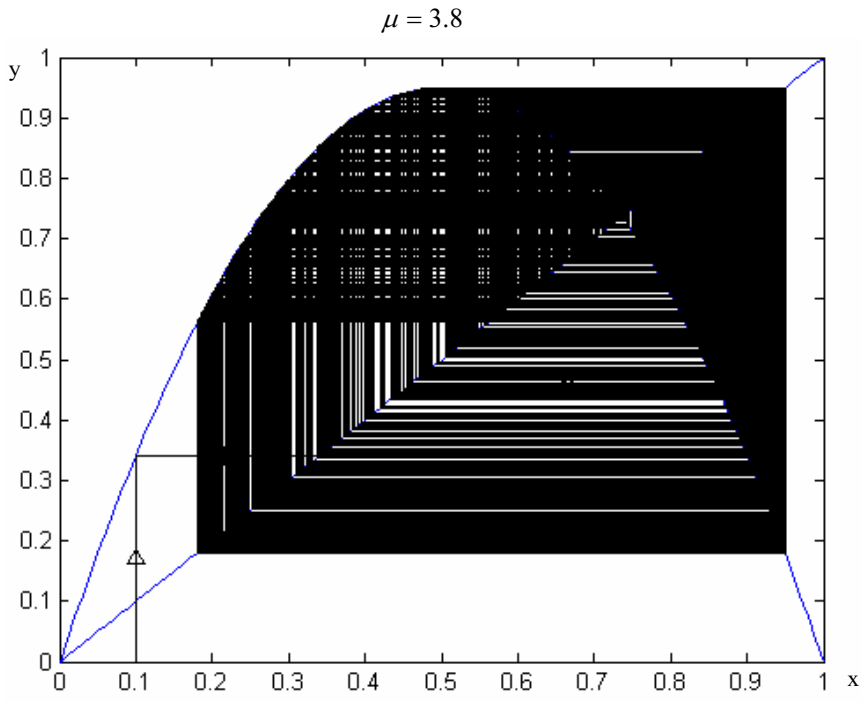

Fig. 4. Iteration process for logistic mapping (μ = 3.8)

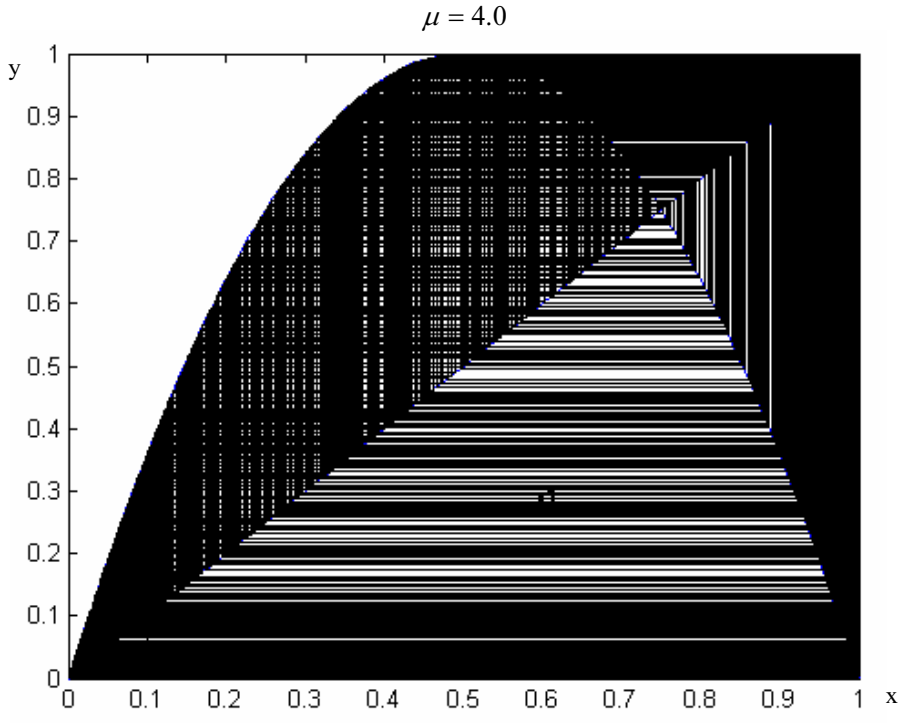

Fig. 5. Iteration process for logistic mapping (μ = 4.0)

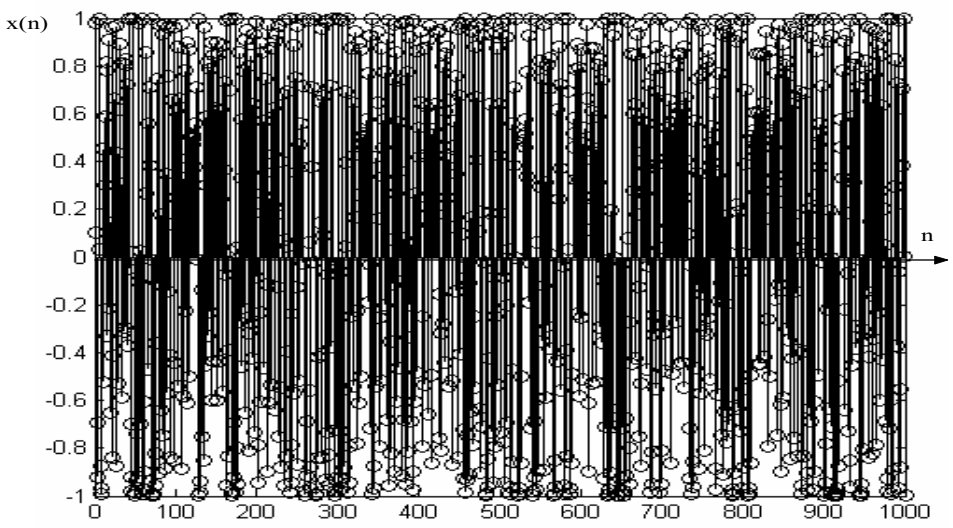

Fig. 6. Logistic full mapping (λ = 2.0) (x(0) = 0.1)

2.3 The Period-doubling Bifurcation of Logistic Mapping

With regard to the analysis of the iteration formula of the preceding logistic mapping, the diagram between the stable roots with the parameter μ, such that μ–x (∞) is shown in Fig. 7.

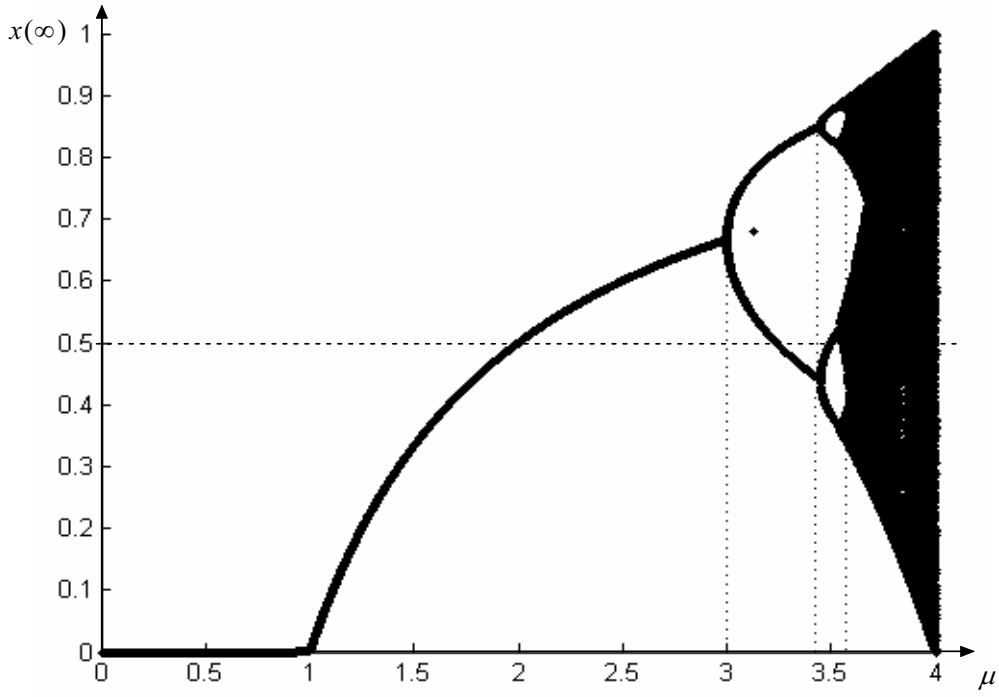

Fig. 7. Period-fork process of the stable trend of the logistic mapping

2.4 Probability Density of the Logistic Mapping [18]

For the logistic mapping, such as in Eq. (1), when $\mu = 4$, the probability density function of the sequence by the formula may be expressed as

$$\rho(x) = \begin{cases} \dfrac{1}{\pi\sqrt{x(1-x)}}, -1 < x < 1 \\ 0, else \end{cases}. \tag{16}$$

For the logistic mapping, such as in Eq. (5), when $\lambda = 2$, the probability density function of the sequence by the formula may be expressed as [19]

$$\rho(x) = \begin{cases} \dfrac{1}{\pi\sqrt{1-x^2}}, -1 < x < 1 \\ 0, else \end{cases}. \tag{17}$$

Then, the mean of the sequence is

$$\bar{x} = \lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} x_i = \int_0^1 x\rho(x)dx = 0 . \tag{18}$$

The two initial values x_{10} and x_{20} are utilized independently to generate two sequences. Then, the cross-correlation function is [20]

$$\begin{aligned} c(m) &= \lim_{N\to\infty} [\frac{1}{N} \sum_{i=0}^{N-1} (x_{1i} - \bar{x})(x_{2(i+m)} - \bar{x})] \\ &= \int_0^1 \int_0^1 x_1 f^m(x_2)\rho(x_1, x_2)dx_1 dx_2 - \bar{x}^2 . \\ &= 0 \end{aligned} \tag{19}$$

The joint probability distribution function of the two sequences is $\rho(x_1, x_2) = \rho(x_1)\rho(x_2)$, and the auto-correlation function of the sequences is equal to $\delta(m)$. The x_{01} and x_{02} are variables of the two sequence, where the x_{1i} and x_{2i} are discrete variable of the two sequences. The $x_{2(i+m)}$ is the m steps delayed value of the x_{2i}. These features indicate that the sequence is similar to the white noise sequence.

3 Improved Algorithm for Digital Chaos Sequence in Full Mapping

To multiply on both sides of Eq. (1), a^2 is

$$a^2 x_k = \mu a x_{k-1}(a - a x_{k-1}) \tag{20}$$

if

$$\begin{cases} y_k = a x_k \\ \mu = 4 \end{cases}. \tag{21}$$

Then, $y_{k-1} = ax_{k-1}$. Thus, Eq. (20) is changed into

$$a y_k = 4 y_{k-1}(a - y_{k-1}), \quad (22)$$

and

$$y_k = 4 y_{k-1}(a - y_{k-1})/a = 4 y_{k-1} - \frac{4}{a} y_{k-1}^2. \tag{23}$$

Given that $x_k \in [0,1]$, then $y_k \in [0, a]$. Let $a = 2^m$ (where the parameter m is an integer). Thus,

$$\begin{aligned} y_k &= 2^m x_k \\ &= x_k << m \end{aligned}. \tag{24}$$

Then, $y_k \in [0, 2^m]$, and Eq. (5) becomes

$$\begin{aligned} y_k &= 4 y_{k-1} - \frac{4}{a} y_{k-1}^2 \\ &= 4 y_{k-1} - y_{k-1}^2 / 2^{m-2} \\ &= 4 y_{k-1} - y_{k-1}^2 >> (m-2) \end{aligned}. \tag{25}$$

To avoid zeros in the iteration, let $y_k \in [1, 2^m - 1]$ and change Eq. (25) into

$$y_k = 4 y_{k-1} - y_{k-1}^2 >> (m-2) - 1, \tag{26}$$

where the parameter m is the binary digits of the sequence value. For example, if m = 32, the 32-bit sequence is

$$y_k = 4 y_{k-1} - y_{k-1}^2 >> 30 - 1, \tag{27}$$

where

$$y_k \in [1, 2^{32} - 1] = [1, 4294967295]. \tag{28}$$

The maximum value of y_k = 4294967295 is obtained when y_{k-1} = 2147483648.

4 Analysis

Let m = 32, and randomly set the initial value y_0. After producing a sequence through Eq. (26), the independence and distribution uniformity of this sequence are analyzed.

4.1 Independence Analysis

Let $y_0 = 1234$. The produced sequence is shown in Fig. 8.

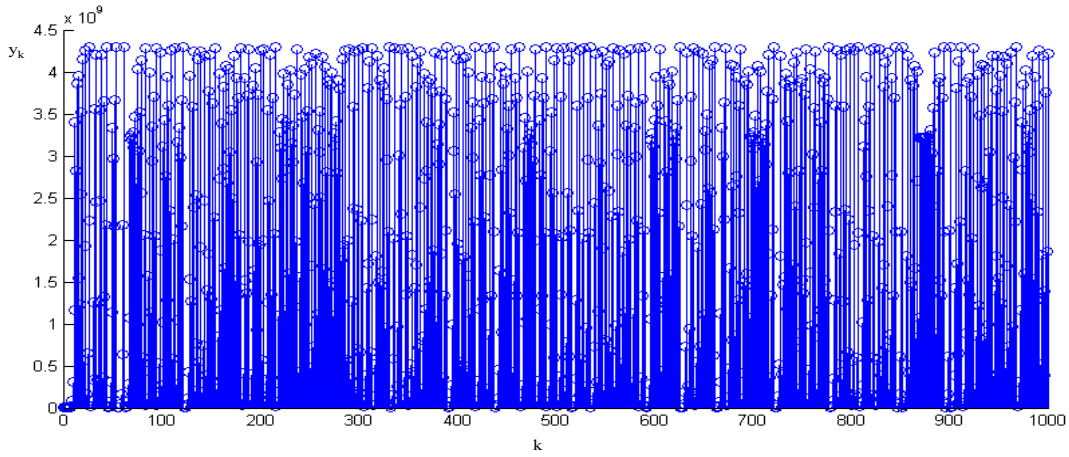

Fig. 8. Digital chaos sequence

The binary quantitative method is utilized for the quantitative sequence to calculate the correlation. The binary quantitative value of produced sequence y_k is

$$z_k = \begin{cases} 1, y_k \geq y_p \\ -1, y_k < y_p \end{cases}. \tag{29}$$

Where the y_p is the thresh value for quantitation. The auto- and cross-correlation of the binary quantitative sequence are then calculated. Assume that z_k and w_k are two sequences in domain $\{1, -1\}$ with a length of p, then their auto- and cross-correlation are $R_z(j)$ and $R_{zw}(j)$, respectively, as shown as follows:

$$R_z(j) = \frac{1}{p}\sum_{i=1}^{p} z_i z_{i+j}, \tag{30}$$

$$R_{zw}(j) = \frac{1}{p}\sum_{i=1}^{p} z_i w_{i+j}. \tag{31}$$

The auto- and cross-correlation are shown in Figs. 9 and Fig. 10, respectively. The auto-correlation of the sequence is binary, whereas its cross-correlation is a small value.

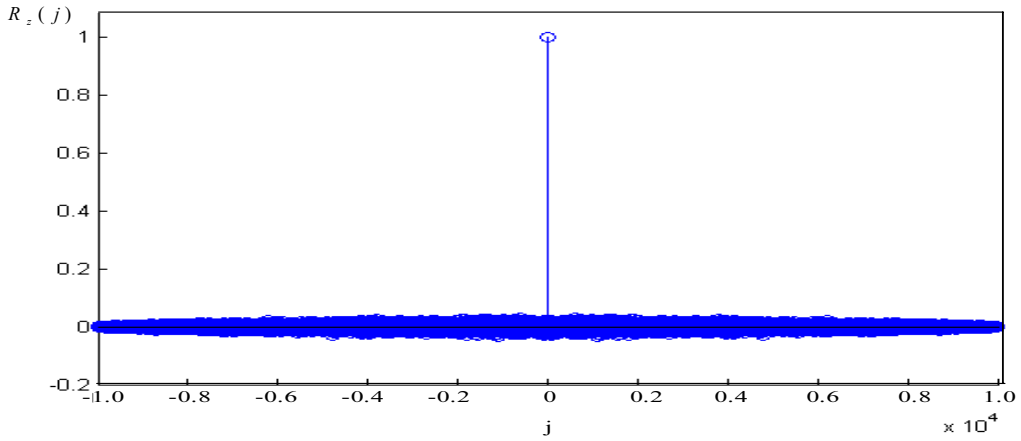

Fig. 9. Full map of the digital serial auto-correlation

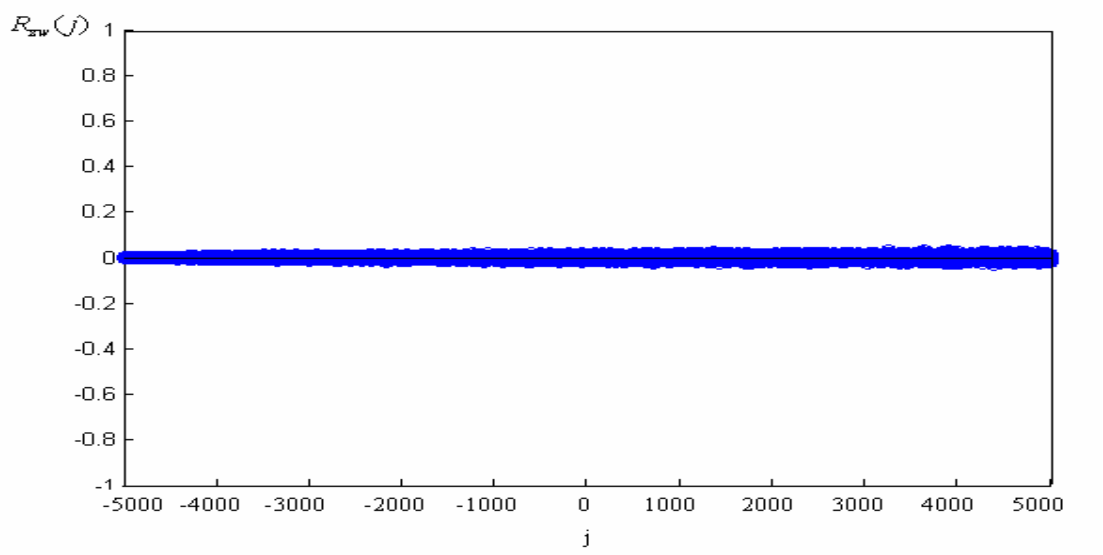

Fig. 10. Full map of the digital serial cross-correlation

Fig. 11 shows the full map of the phase space for the improved algorithm to generate a sequence. The original phase space of the logistic map is retained.

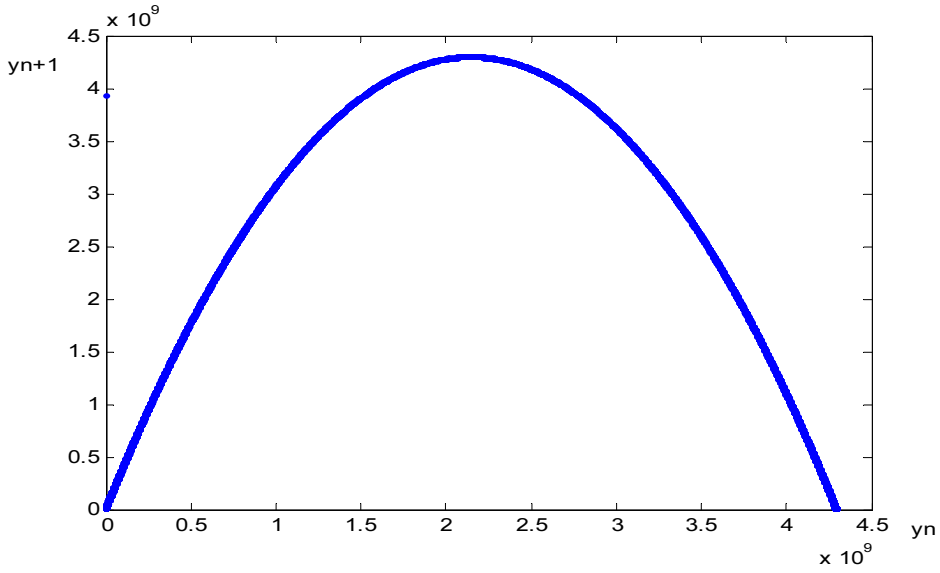

Fig. 11. Full map of the digital serial phase space

The Lyapunov coefficient reflects chaotic sensitivity to the initial value. We set the substitution of the chaotic operation point for x_i, and the chaos iteration after the value is x_{i+1} if

$$y = x_{i+1} = f(x_i). \tag{32}$$

Then, Lyapunov coefficient is expressed as

$$LE = \frac{1}{n}\sum_{i=0}^{n-1} \ln | f^{'}(x_i) |. \tag{33}$$

The discrete sequence is changed into a difference equation as

$$LE = \frac{1}{n}\sum_{i=0}^{n-1} \ln | f(x_{i+1}) - f(x_i) |. \tag{34}$$

According to Eq. (26), the Lyapunov coefficient for the preceding sequence is *LE* = 20.686. Inequality *LE* > 0 indicates the chaotic sequences generated by the new method.

4.2 Uniformity Analysis

We randomly set the initial value. For example, let y0 = 1234. The digital sequence is generated, and the number of times "1" appears in each bit of the 32-bit sequence is counted. The statistics is shown in Table 1. The calculated mean of the sample is 50002, which is close to the ideal value of 5000 (average). The maximum is 50282, and the minimum is 49727. Pearson introduced the χ^2 principle [21] to verify sequence uniformity. The statistical variable χ^2 is expressed as

$$K_n^2 = \frac{1}{n}\sum_{j=1}^{k}\frac{(n_j - np_j)^2}{p_j}, \tag{35}$$

where n_j is the count in the interval S_j of sequence $X_i, i = 1,2,\cdots,n$ (n is the count of the sample $n = \sum_{j=1}^{k} n_j$), k is the total number of the interval, and p_j is the probability of every interval (j is the interval code).

Table 1. Cumulative of bit "1" appearance in each bit of the 32-bit sequence.

bits	count	bits	count	bits	count
0	50110	11	50097	22	50126
1	49953	12	49908	23	50098
2	49902	13	49977	24	49974
3	49972	14	50045	25	50104
4	50068	15	50142	26	49940
5	49727	16	49809	27	49971
6	49882	17	49841	28	49784
7	50113	18	49808	29	50282
8	50180	19	50030	30	49991
9	50125	20	50077	31	49813
10	49994	21	50236		

For uniform distribution, $p_j = \frac{1}{k}$, and

$$\begin{aligned} K_n^2 &= \frac{1}{n}\sum_{j=1}^{k}\frac{(n_j - np_j)^2}{p_j} \\ &= \frac{1}{n}\sum_{j=1}^{k}\frac{[n_j - (n/k)]^2}{1/k} \\ &= \frac{k}{n}\sum_{j=1}^{k} n_j^2 - n \\ &= \frac{k}{\sum_{j=1}^{k} n_j}\times\sum_{j=1}^{k} n_j^2 - \sum_{j=1}^{k} n_j \end{aligned} . \tag{36}$$

For the given confidence level α, if $K_n^2 > \chi_{k-1,\alpha}^2$, then the distribution is not uniform in $x_1,\cdots,x_k$; otherwise, the distribution is uniform in $x_1,\cdots,x_k$.

According to Table 1, the statistic is $K_n^2 = 11.6553$.

Let level $\alpha = 0.05$; thus, $\chi_{k-1,\alpha}^2 = \chi_{31,0.05}^2 > \chi_{30,0.05}^2 = 43.773 > K_n^2 = 11.6553$ is known. Then, bit 1 and

bit 0 are evenly distributed in the sequence value of 32 bits.

5 Hardware Implementation

The implementation of Eq. (27) in a field-programmable gate array chip is shown in Fig. 12, where *in*[31..0] is the signal y_{k-1} for the input, and *out*[31..0] is the signal y_k for the output. The signal *clock* is the clock, and the chip works on the positive edge. The signal *set* is the controlling signal. The initial value is saved in the chip on the positive edge of the clock when *set* = 1. The chip iteration is calculated on the positive edge of the clock when *set* = 0. The function of the chip is shown in Fig. 13. The operation may be expressed as

$$\begin{cases} (clock \uparrow) \\ set = 0 \\ tep = in \end{cases}, \tag{37}$$

$$\begin{cases} (clock \uparrow) \\ set = 1 \\ out = tep << 2 - ((tep * tep) >> 30) \end{cases}. \tag{38}$$

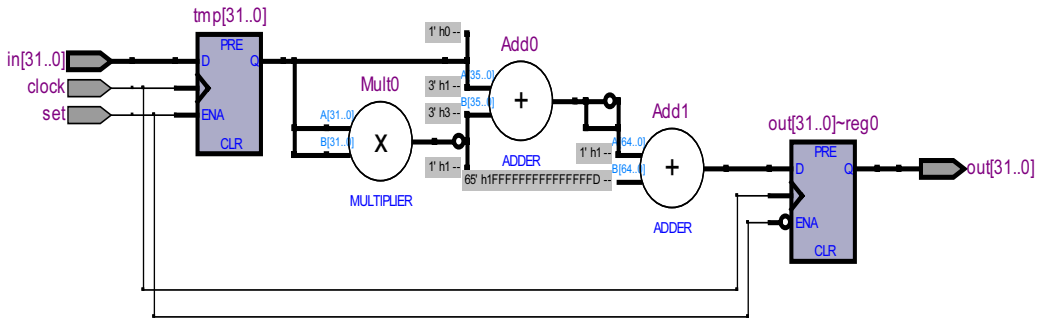

Fig. 12. Implementation of a 32-bit integer chaos in a chip

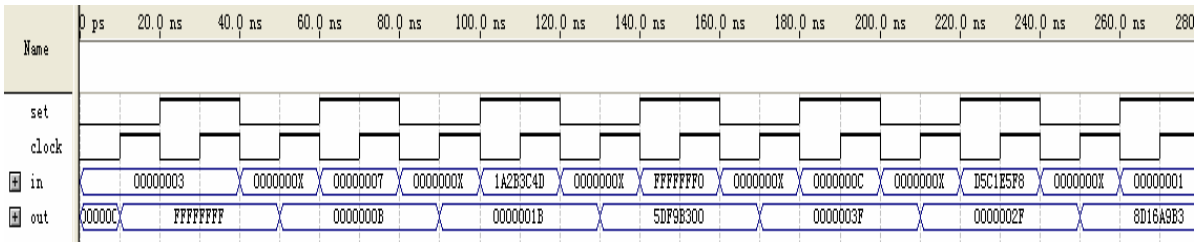

Fig. 13. Simulation function of the digital chaos chip

As shown in Fig. 13, when the input is 0x00000003, the output of the chip is 0x0000000B. When the input is 0x00000007, the output of the chip is 0x0000001B. When the input is 0x1A2B3C4D, the output of the chip is 0x5DF9B300. When the input is 0xfffffff0, the output of the chip is 0x0000003F. When the input is 0x0000000C, the output of the chip is 0x0000002F. When the input is 0xD5C1E5F8, the output of the chip is 0x8D16A9B3, …, which is similar to the manual calculation. The chip correctly completes the arithmetic of digital chaos.

Fig. 13 also shows that only one clock cycle is necessary for integer chaos, and the floating-point chaotic operation should be conducted in more than one clock cycle. Therefore, the improvement in

integer chaos calculation increases the processing speed and is convenient for hardware implementation.

6 Conclusions

A method of sequence generation was developed by improving the one-dimensional logistic chaotic sequence. The produced sequence was analyzed. The results showed that the sequence generated with the improved method has independent features and exhibits distribution uniformity. The new method that employed an integer and avoided floating-point arithmetic was convenient and increased speed. The integer chaos was easy to identify in digital circuits.

Acknowledgments

This work was partly supported by the Key Projects of the Provincial Natural Science Research for Colleges and Universities in Anhui Province, China (KJ2014A239), and was also supported by the fund of nature science for colleges and universities in Anhui Province, China (KJ2011Z342).

References

[1] L. Li, W.N. Wang, J.J. Li, Self-adaptive image encryption algorithm based on logistic map and hyper-chaos, Microelectronics & Computer 29(1)(2012) 42-46.

[2] H.G. Zhu, X.J. Lu, X.D. Zhang, A novel image encryption scheme with 2D-logistic map and quadratic residue, Journal of Northeastern University 35(1)(2014) 20-23.

[3] B. Xu, Y. Li, Research on image encryption algorithm logistic chaotic based on an improved digital mapping, Computer Measurement & Control 22(7)(2014) 2157-2159.

[4] Y.Y. Liang, Y. Wang, M.L. Yu. Simulation study on secure communication based on improved logistic-map, Experiment Science and Technology 81(13)(2013) 25-28.

[5] F.F. Zhu, H.Z. Liu, The search of internet of things security supervision based on logistic mapping of chaos theory, Journal of Guangxi University for Nationalities 35(2)(2013) 47-51.

[6] F.X. Yang, Image grouping encryption algorithm based on logistic mapping and z-mapping, Laser & Infrared 44(1)(2014) 103-107.

[7] X.Y. Sun, H.G. Zhang, M. Zhang, Triangular cryptosystem with logistic chaos disturbance, Computer Applications and Software 31(9)(2014) 268-271.

[8] C.H. Li, Y.B. Li, L. Zhao, Research on statistical characteristics of chaotic pseudorandom sequence for one-dimensional logistic map, Application Research of Computers 31(5)(2014) 1403-1406.

[9] X.Y. Pan, H.M. Zhao, Research on the entropy of logistic chaos, Acta Phy. Sin. 61(20)(2012) 1-7.

[10] Y.Z. Liu, C.S. Lin, The logistic-unified hybrid chaotic system, Acta Phy. Sin. 60(3)(2011) 1-6.

[11] J.L. Fan, X.F. Zhang, Piecewise logistic chaotic map and its performance analysis, Acta Electronica Sinica 31(4)(2009) 720-725.

[12] S.G. Yan, Y.B. Chen, Performance analysis of full mapping chaotic sequence about logistic, Modern electronic technology 314(3)(2010) 194-197.

[13] J.F. Yu, W.G. Yang, W.T. Lu, Generation and performance analysis of digital chaotic sequence from surjective logistic-map, Telecommunication Engineering 53(2)(2013) 140-145.

[14] H.O. Peitgen, H.D. Jurgens, Chaos and Fractals New Frontiers of Science, Springer, Germany, 2012.

[15] S.D. Liu, F.M. Liang, Chaos and Fractals in Nature Science, Peking University Press, Beijing, 2003.

[16] B.L. Hao, From a Parabolic: An Introduction to Chaos Dynamics, Shanghai Science and Technology Education Press, Shanghai, 1995.

[17] X. Sun, Z.Q. Wu, Fractals Principle and Application, University of Science and Technology of China Press, Hefei, 2003.

[18] X. Sun, K.X. Yi, Y.X. Sun, New image encryption algorithm based on chaos system, Journal of Computer-Aided Design & Computer Graphics 14(2)(2002) 1-5.

[19] H.G. Schuster, Deterministic Chaos: An Introduction, Second Revised Edition, Federal Republic of Germany: VCH, Weinheim, 1988.

[20] Y.H. Yu, Study on chaotic spread spectrum communication technology and its application, [dissertation] Jilin: Jilin University, 2005.

[21] X. Lu, The Foundation of Mathematic Statistics, Tsinghua University Press, Beijing, 1998.