Chaos, Solitons and Fractals 120 (2019) 62–66

Contents lists available at ScienceDirect

Chaos, Solitons and Fractals

Nonlinear Science, and Nonequilibrium and Complex Phenomena

journal homepage: www.elsevier.com/locate/chaos

Minimal digital chaotic system

Erivelton G. Nepomuceno[a], Arthur M. Lima[a], Janier Arias-García[b], Matjaž Perc[c,d,e,*], Robert Repnik[c]

[a] Control and Modelling Group (GCOM), Department of Electrical Engineering, Federal University of São João del-Rei, São João del-Rei, MG 36307-352, Brazil
[b] Department of Electronic Engineering, Federal University of Minas Gerais, Brazil
[c] Faculty of Natural Sciences and Mathematics, University of Maribor, Koroška cesta 160, Maribor SI-2000, Slovenia
[d] CAMTP – Center for Applied Mathematics and Theoretical Physics, University of Maribor, Mladinska 3, Maribor SI-2000, Slovenia
[e] Complexity Science Hub Vienna, Josefstädterstraße 39, Vienna A-1080, Austria

ARTICLE INFO

Article history:
Received 11 January 2019
Accepted 20 January 2019

Keywords:
Chaos
Nonlinear dynamics
FPGA synthesis
Computer arithmetic
Digital system

ABSTRACT

Over the past few decades, many works have been devoted to designing simple chaotic systems based on analog electronic circuits. However, the same attention is not observed in digital chaotic systems. This paper presents a design of a digital chaotic system using a digit complement. This special case of fixed-point number representation allows us to reduce the silicon area and the number of logic elements to perform the arithmetic operations. The design presents a configurable number of bits, and it is based on the logistic map. The proposed circuit has been implemented on a reconfigurable hardware, FPGA Cyclone V, showing that the number of logic elements has been significantly reduced compared to other works in the literature.

1. Introduction

Much research in recent years has focused on exploiting chaotic systems in cryptography [1–5]. Chaos theory has emerged as a promising solution to develop schemes for cryptographic protocols, used to multimedia encryption [3,6,7] or secure transactions and crypto-currencies [8,9]. An important aspect of this research is the development of simple and reliable chaotic systems. Although, a great attention has been observed to design simple analog electronic circuits [10–12], the same attention is not observed to digital chaotic systems.

In general, the field-programmable gate array (FPGA) has been widely applied as an flexible and efficient digital platform to implement digital chaotic systems. Chaotic system using FPGA has generated considerable recent research interest [13]. Generally, the main focus is the reproduction of equations by means of a detailed description of arithmetic operations [14]. As an example, Hua et al. [15] have proposed a sine-transform-based chaotic system of generating one-dimensional (1-D) chaotic maps and implemented on FPGA. The authors suggested that the designed chaotic system is robust, and it presents a simple hardware implementation. The reader can refer to [16–20] and references therein for numerous contributions on chaotic systems designed on FPGA.

Notwithstanding the extensive research effort that has gone into chaotic systems on FPGA, few studies have focused on reducing the number of logic elements using number representation. For instance, Falk and Houk [21] have used residue number system arithmetic operations to respectively determine solutions for the polynomial equations. The solutions are iteratively computed and expressed as residue values. Giard et al. [22] has implemented a discrete-time chaotic generators focusing on power consumption, resource usage, and maximum execution frequency. The authors have used fixed point number representation. A different approach has been adopted by Masmoudi et al. [23], who have proposed a scheme for image encryption based on the use of a chaotic map with large key space and continued fractions. In this work, we have proposed a novel design of digital chaotic system based on digit complement. The digit complement is a case of fixed-point number representation. With this representation, it has been possible to design a digital chaotic system with a smaller number of logic elements to perform the arithmetic operations. The proposed digital chaotic system also presents a configurable number of bits and it has been implemented on reconfigurable hardware, FPGA Cyclone V. The chaotic property has been validated by computing the largest positive Lyapunov [24].

* Corresponding author at: Faculty of Natural Sciences and Mathematics, University of Maribor, Koroška cesta 160, SI-2000 Maribor, Slovenia.
E-mail addresses: nepomuceno@ufsj.edu.br (E.G. Nepomuceno), arthurlima67@yahoo.com.br (A.M. Lima), janier-arias@ufmg.br (J. Arias-García), matjaz.perc@um.si (M. Perc), robert.repnik@um.si (R. Repnik).

https://doi.org/10.1016/j.chaos.2019.01.019
0960-0779/

The paper is organized as follows. In the next section, we describe the application of digit complement to design of digital chaotic systems. Research results are summarised in Section 3 and some concluding remarks are given in Section 4.

2. Design of digital chaotic system

This section presents the design of a digital chaotic system. We have used a modified version of the logistic map [25]. First, the numerical representation used is introduced, followed by the description of the arithmetic operations and the algorithm of the multiplier.

2.1. Numerical representation

Representation by complement is one of the most used binary number representations. There are basically two types of complement: the radix complements and the digit complement. What differs from each other is how a constant of complement M is conceived depending on the number of bits k. This form of representation is highly detailed by Parhami [26]. Consider the digit complement:

$$M = 2^k - ulp, \tag{1}$$

where *ulp* is *unit in the least position*. The logistic map is given as follows [25]:

$$x_{n+1} = rx_n(1 - x_n), \tag{2}$$

where $r \in [0, 4]$ is the bifurcation parameter and the sequence obtained by iterating Eq. (2) is represented by $x_n \in [0, 1]$.

Let $x_{2,n}$ be the n*th* term of the sequence of logistic map in base 2. A binary number is represented by $x_{2,n} = b_1 b_2 b_3 b_4, \ldots, b_k$ and $x_{2,n}^{dc}$ is the digit complement such as

$$x_{2,n}^{dc} + x_{2,n} = M = 2^k - ulp,$$

where

$$x_{2,n}^{dc} = \bar{b}_1 \bar{b}_2 \bar{b}_3 \bar{b}_4, \ldots, \bar{b}_k, \tag{3}$$

The summation of a number and its digit complement is $x_{2,n} + x_{2,n}^{dc} = 1111 \ldots 11_2$. In the radix complements we have $M = 2^k$. It is clear that two types of complements are differed by one *ulp*. Thus, the radix complements $x_{2,n}^{rc}$ is given by:

$$x_{2,n}^{rc} = x_{2,n}^{dc} + ulp. \tag{4}$$

2.2. Proposed numerical representation by digit complement

According to Eq. (2), we have $x_n \in [0, 1]_{10}$ for $0 < r \leq 4$. Thus, using fixed-point, $x_{2,n}$ is represented by fractional binary number, such as:

$$x_{2,n} = 0.b_1 b_2 b_3 b_4, \ldots, b_k,$$

in which the conversion to decimal is:

$$x_{10} = \sum_{i=1}^{k} b_i \times 2^{-i}.$$

Therefore, a fractional binary number and the digit complement of its fractional part is

$$x_{2,n} + x_{2,n}^{dc} = 0.111 \ldots 1 = 1_2 - ulp.$$

An interesting aspect is that $x_{2,n}^{dc} \approx 1 - x_{2,n}$ as long as $k \to \infty$ and $x_n \in [0, 1]$. A factor can easily be applied to correct this difference. The number 1_2 can be represented as a fractional binary and at the same time set the constant $M = 1$. In such way, digit complement of the fractional part of $x_{2,n}$ is $1 - x_{2,n}$. Thus,

$$M - x_{2,n} = 1_2 - x_{2,n} = x_{2,n}^{dc}.$$

Hence, the logistic map is represented by:

$$x_{2,n+1} = r_2 x_{2,n} x_{2,n}^{dc}. \tag{5}$$

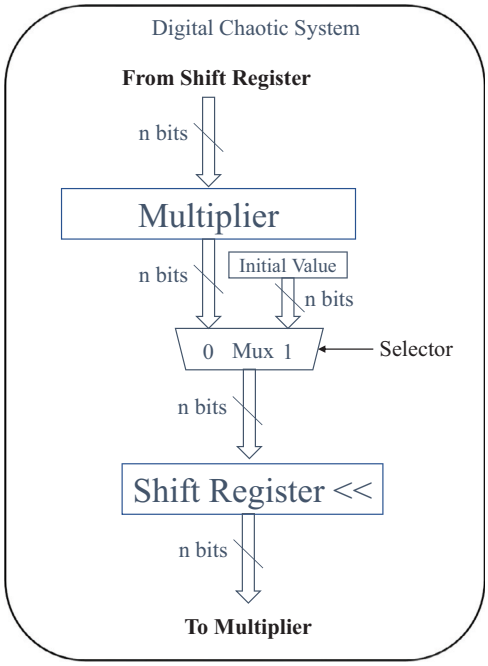

Fig. 1. General scheme of the proposed digital chaotic system using digit complement and based on the logistic map. With the digit complement, the logistic map is reduced to only one arithmetic operation, a multiplication as described in Algorithm 1 and a shift register, which can be employed since bifurcation parameter is $r = 4$.

2.3. Multiplier

The multiplication operation $x_{2,n} x_{2,n}^{dc}$ is responsible for the only arithmetic operation involved in this system, as shown in (5). First, let us present this multiplication with 2 bits and $r_2 = 1$. The result of multiplication is $b_0\bar{b}_1 + b_1\bar{b}_0$, which can be implemented by a *XOR* gate. Notice that there is no need to establish subsystems to represent the first element nor the last element. This observation has been used to develop the multiplication algorithm as follows. The multiplication operation is composed by subsystems which performs the summation of each row element that belongs to the same column [26]. The sum of the elements presented in a single column can be represented by a XOR gate. In this paper, we have considered summation from each column of the shift-add algorithm, which has been implemented by means of hardware description language in Intel Quartus Prime Software. The multiplier is presented in Algorithm 1 for an hardware description language representation.

The adopted multiplier procedure has a huge impact on the number of elements used to synthesise the logistic map. If it is possible to reduce the amount of input, by consequence, the amount of carry is also reduced. At the output of the multiplier (see Fig. 1), there is a shift register that receives the value of the variable s with $2 \times n$ bits width, which excludes the two most significant bits corresponding to a multiplication by 4 sending the 8 most significant bits to the output. This can be done because the orbit of the logistic map is within [0,1] and $x_n\overline{x_n} \leq 0.25$ as $r = 4$.

The bifurcation parameter r is set as 4, which allows to be implemented as a simple *shift register*, since the multiplication of a binary number by power of 2 is a left shift operation. Using the multiplier as presented in the Algorithm 1 and by means of Eq. (5), the overall digital system is presented in Fig. 1. This design is able to reproduce the logistic map using only one arithmetic operation, the multiplication as described in Algorithm 1 and a shift register.

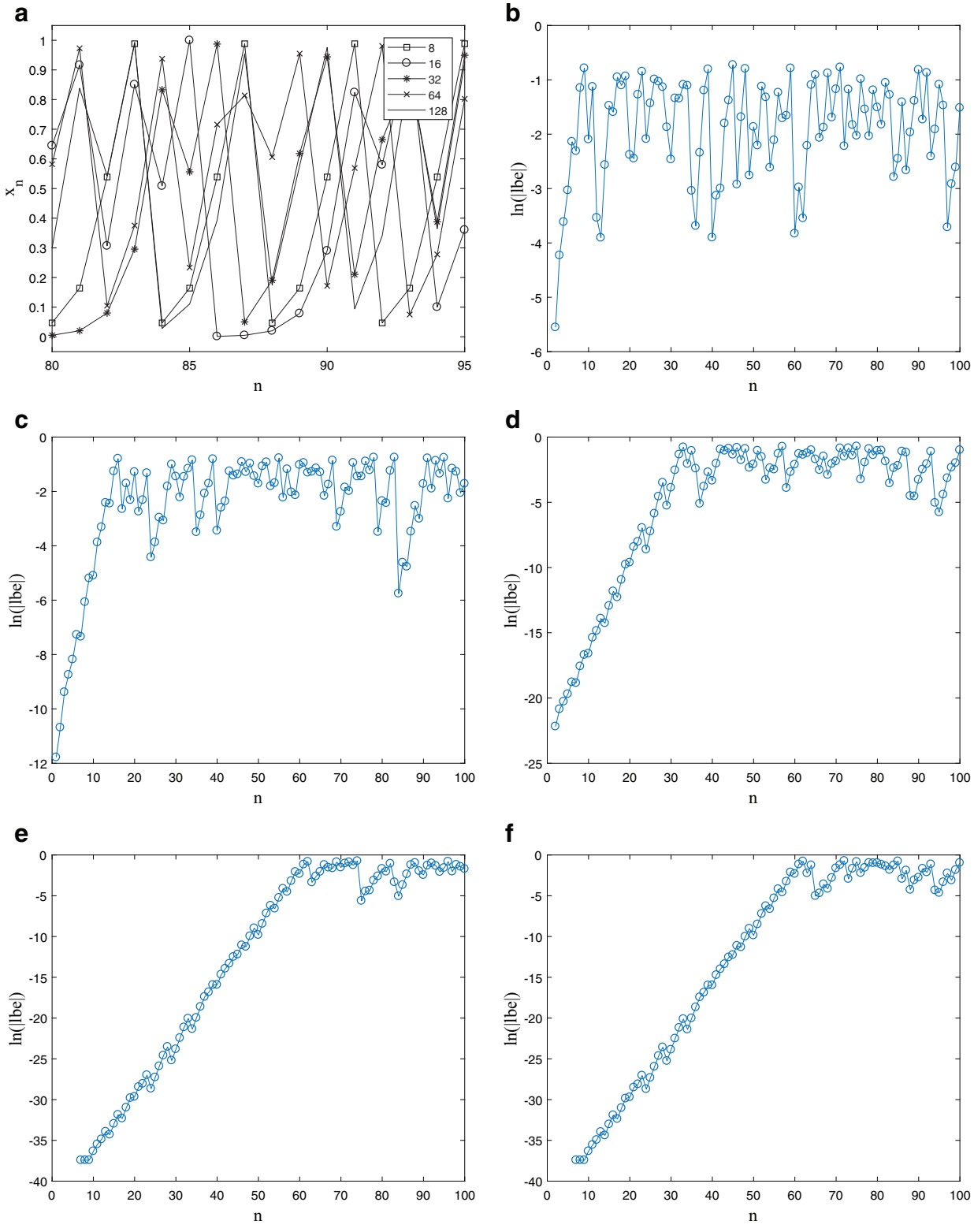

Fig. 2. (a) Iteration of logistic map represented in the digital system for 5 bit-widths, namely 8, 16, 32, 64 and 128 bits. Although the sequences have the same initial condition, they diverge exponentially as different number representation yields different computer realisations. According to Peixoto et al. [27], this feature is equivalent to sensitivity to initial conditions. n is the iterate number. In this figure, we have shown the range of n between 80 and 95. (b)–(f) represent the natural logarithm of the lower bound error, $\ln(|lbe|)$, calculated according to [24]. These figures are in line with Fig. 1(a) shown in [24].

Table 1
Circuit Resource Utilisation on Selected FPGA for Different Bit-Width. The sixth column presents the Lyapunov exponent (λ), calculated using the technique proposed by Mendes and Nepomuceno [24]. DSP stands for digital signal processing.

Bits	Logic Utilization	Registers	DSP	Freq. (MHz)	(λ)	Literature [24]
8	5	16	1	125	0.6809	0.693
16	9	32	1	125	0.7325	
32	39	68	3	108	0.6902	
64	169	86	9	68	0.6906	
128	1,061	197	25	43	0.6910	

3. Results and discussion

The proposed digital chaotic system has been implemented using the VHDL language in the Intel Quartus Prime Software. This system has been targeted on the low-cost device Cyclone *V* 5*CSEMA*4*U*23*C*6 and its functional simulation has been performed on *ModelSim-Intel FPGA Starter Edition*. The Intel Quartus Prime Software was settled to infer its use of multipliers, which usually increases performance and reduces logic resource utilisation.

The digital chaotic system with (8, 16, 32, 64, 128) has been iterated and converted to decimal base, as shown in Fig. 2. As

Algorithm 1 Multiplier $x_{2,n}x_{2,n}^{dc}$. DW stands for data width.

```
1: io ← Multiplier Value
2: for j in 0 to DW do
3:     for i in 0 to DW do
4:         if i > j then
5:             pc(i + j) = (io(i) xor io(j))
6:         else
7:             pc(i + j) = '0'
8:         end if
9:         pv = pv + pc
10:        pc = '0'
11:    end for
12: end for
13: s = pv
```

pointed out by Peixoto et al. [27], it is possible to observe divergence of the pseudo-orbits due to different number representation, which is expected for a chaotic system. The digital chaotic system has been validated by means of the largest Lyapunov exponent. Table 1 presents the calculated values of the Lyapunov exponent, which are in good agreement with the correct value. Moreover, according to [24], the largest Lyapunov exponent can be identified through slope of line of the logarithm of the lower bound error. Fig. 2(b)–(f) presents this slope for 8, 16, 32, 64 and 128 bit-widths, respectively. These figures are in line with Fig. 1(a) shown in [24], which strongly confirms the chaotic behaviour of designed digital chaotic system.

The resource consumption of the proposed design of digital chaotic system is shown in Table 1. The performance results have revealed a reliable frequency from 125MHz to 43MHz for 8 to 128 bit-widths. The adaptive logic module used in Cyclone *V* with 8-input, four registers and digital signal processing blocks have applied an average logic utilisation of 1% to 7% and a digital signal processing usage up to 30% (see Table 1). This is an interesting result regarding the authors purposes of simplifying arithmetic operation systems. If it is compared to systems of higher level of abstraction as discussed by Silva et al. [20] it is clear the efficiency (lower amount of logic elements) that our novel technique has achieved. It has been used 1471 logic elements, 16 9-bits multipliers and 1358 registers for a digital chaotic system with 32 bits targeted on Cyclone IV as described in [20].

Other similar solutions in literature have presented more complex outcomes. Pande and Zambreno [28] have presented a modified logistic map dealing with double floating point precision to increase the Lyapunov exponent and uniformity of bifurcation map, which has been implemented in one of the top family FPGA devices of Xilinx, the Virtex-6. Tolba et al. [29] have investigated Speech Encryption using a Virtex-5 using modified logistic map with 3 × registers and 2 × multipliers as well as 2 × adders to deal with 32-bits in fixed-point representation. Dabal and Pelka [30] has adopted logistic map using the expression $x_{n+1} = (4x_n)(1 - x_n)$ which implies in an extra subtraction operation to perform the $(1 - x_n)$. Comparing the results in [30], the digital signal processing units is 16 for 64-bits, while our proposed design has only employed 9 units.

4. Conclusion

This paper has presented a design of digital chaotic system using digit complement. The design has been implemented in a hardware language description (VHDL), which allows flexibility regarding the number of bits. The computation of the largest Lyapunov exponent for 8, 16, 32, 64 and 128 bits is in good agreement with the literature which is an evidence of the chaotic properties of the system. The advantage of the proposed approach has been shown by a substantial reduction in the number of digital devices. As many of applications for chaotic system are based on digital systems, such as cryptography, we believe that this work opens a new path for future research endeavours. As future research the authors are investigating the use of other chaotic maps, such as the tent map and quadratic map.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Erivelton. G. Nepomuceno was supported by Brazilian Research Agencies: CNPq/INERGE (Grant No. 465704/2014-0), CNPq (Grant No. 425509/2018-4) and FAPEMIG (Grant No. APQ-00870-17). Matjaž Perc was supported by the Slovenian Research Agency (Grants J1-7009, J4-9302, J1-9112 and P1-0403). We would also like to express our gratitude to Josefredo Gadelha da Silva for his valuable advice and discussions.

References

[1] Pisarchik AN, Flores-Carmona NJ, Carpio-Valadez M. Encryption and decryption of images with chaotic map lattices. Chaos 2006;16(3):033118.
[2] Machicao J, Bruno OM. Improving the pseudo-randomness properties of chaotic maps using deep-zoom. Chaos 2017;27(5):053116.
[3] Li C, Xie T, Liu Q, Cheng G. Cryptanalyzing image encryption using chaotic logistic map. Nonlinear Dyn 2014;78(2):1545–51.
[4] Koshkin S, Styers T. From golden to unimodular cryptography. Chaos Solitons Fractals 2017;105:208–14.
[5] Saha R, Geetha G. Symmetric random function generator (SRFG): a novel cryptographic primitive for designing fast and robust algorithms. Chaos Solitons Fractals 2017;104:371–7.
[6] Wu X, Wang D, Kurths J, Kan H. A novel lossless color image encryption scheme using 2d DWT and 6d hyperchaotic system. Inf Sci (NY) 2016;349–350:137–53.
[7] Kuiate GF, Rajagopal K, Kingni ST, Tamba VK, Jafari S. Autonomous Van der polduffing snap oscillator: analysis, synchronization and applications to real–time image encryption. J Dyn Control Syst 2018;6(3):1008–22.
[8] Asgari Chenaghlu M, Jamali S, Nikzad Khasmakhi N. A novel keyed parallel hashing scheme based on a new chaotic system. Chaos Solitons Fractals 2016;87:216–25.
[9] Kocarev L, Amigó JM, Szczepanski J. Chaos-based cryptography: an overview. In: Proceedings of the international symposium on nonlinear theory and its applications (NOLTA); 2005. p. 453–6.

[10] Tchitnga R, Fotsin HB, Nana B, Louodop Fotso PH, Woafo P. Hartley's oscillator: the simplest chaotic two-component circuit. Chaos Solitons Fractals 2012;45(3):306–13.
[11] Piper JR, Sprott JC. Simple autonomous chaotic circuits. IEEE Trans Circuits Syst II Express Briefs 2010;57(9):730–4.
[12] Muthuswamy B, Chua LO. Simplest chaotic circuit. Int J Bifurcation Chaos 2010;20(05):1567–80.
[13] Muthuswamy B, Banerjee S. A route to chaos using FPGAs. Springer International Publishing; 2015.
[14] Wang Q, Yu S, Li C, Lu J, Fang X, Guyeux C, Bahi JM. Theoretical design and FPGA-based implementation of higher-dimensional digital chaotic systems. IEEE Trans Circuits Syst I Regul Pap 2016;63(3):401–12.
[15] Hua Z, Zhou B, Zhou Y. Sine-transform-based chaotic system with FPGA implementation. IEEE Trans Ind Electron 2018;65(3):2557–66.
[16] Tlelo-Cuautle E, Pano-Azucena AD, Rangel-Magdaleno JJ, Carbajal-Gomez VH, Rodriguez-Gomez G. Generating a 50-scroll chaotic attractor at 66 MHz by using FPGAs. Nonlinear Dyn 2016;85(4):2143–57.
[17] Wang H, Song B, Liu Q, Pan J, Ding Q. FPGA design and applicable analysis of discrete chaotic maps. Int J Bifurc Chaos 2014;24(04):1450054.
[18] Lai Q, Zhao XW, Rajagopal K, Xu G, Akgul A, Guleryuz E. Dynamic analyses, FPGA implementation and engineering applications of multi-butterfly chaotic attractors generated from generalised sprott c system. Pramana 2018;90(1):6.
[19] Torres-Perez E, de la Fraga LG, Tlelo-Cuautle E, Leon-Salas WD. On the FPGA implementation of random number generators from chaotic maps. In: Proceedings of the IEEE XXIV international conference on electronics, electrical engineering and computing (INTERCON). IEEE; 2017. p. 1–4.
[20] Silva DA, Pereira EB, Nepomuceno EG. Implementation of the logistic map with FPGA using 32 bits fixed point standard. In: Proceedings of the XIII SBAI; 2017. p. 1–6.
[21] Falk R.A., Houk T.L. Residue number encryption and decryption system. 1991. US Patent 5,077,793.
[22] Giard P, Kaddoum G, Gagnon F, Thibeault C. FPGA implementation and evaluation of discrete-time chaotic generators circuits. In: Proceedings of the thirty-eighth annual conference on IEEE industrial electronics society (IECON). IEEE; 2012. p. 3221–4.
[23] Masmoudi A, Bouhlel MS, Puech W. A new image cryptosystem based on chaotic map and continued fractions. In: Proceedings of the European signal processing conference; 2010. p. 1504–8.
[24] Mendes EMAM, Nepomuceno EG. A very simple method to calculate the (positive) largest Lyapunov exponent using interval extensions. Int J Bifurc Chaos 2016;26(13):1650226.
[25] May RM. Simple mathematical models with very complicated dynamics. Nature 1976;261(5560):459–67.
[26] Parhami B. Computer arithmetic algorithms and hardware architectures. New York: Oxford University Press; 2012.
[27] Peixoto ML, Nepomuceno EG, Martins SA, Lacerda MJ. Computation of the largest positive Lyapunov exponent using rounding mode and recursive least square algorithm. Chaos Solitons Fractals 2018;112:36–43.
[28] Pande A, Zambreno J. A chaotic encryption scheme for real-time embedded systems: design and implementation. Telecommun Syst 2013;52(2):551–61.
[29] Tolba MF, Sayed WS, Radwan AG, Abd-El-Hfiz SK. FPGA realization of speech encryption based on modified chaotic logistic map. In: Proceedings of the 2018 IEEE international conference on industrial technology (ICIT); 2018. p. 1412–17.
[30] Dabal P, Pelka R. A study on fast pipelined pseudo-random number generator based on chaotic logistic map. In: Proceedings of the seventh international symposium on design and diagnostics of electronic circuits & systems. IEEE; 2014. p. 195–200.