Chaos, Solitons and Fractals 77 (2015) 247–252

Contents lists available at ScienceDirect

Chaos, Solitons and Fractals

Nonlinear Science, and Nonequilibrium and Complex Phenomena

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

Chaotic logic gate: A new approach in set and design by genetic algorithm

Mahmood Beyki*, Mahdi Yaghoobi

Department of Electrical Engineering, Islamic Azad University, Mashhad Branch, Mashhad, Iran

ARTICLE INFO

Article history:
Received 13 December 2014
Accepted 29 May 2015

Keywords:
Chaotic logic gate
Logistic map
Setting the parameter values
Genetic algorithm

ABSTRACT

How to reconfigure a logic gate is an attractive subject for different applications. Chaotic systems can yield a wide variety of patterns and here we use this feature to produce a logic gate. This feature forms the basis for designing a dynamical computing device that can be rapidly reconfigured to become any wanted logical operator. This logic gate that can reconfigure to any logical operator when placed in its chaotic state is called chaotic logic gate. The reconfiguration realize by setting the parameter values of chaotic logic gate. In this paper we present mechanisms about how to produce a logic gate based on the logistic map in its chaotic state and genetic algorithm is used to set the parameter values. We use three well-known selection methods used in genetic algorithm: tournament selection, Roulette wheel selection and random selection. The results show the tournament selection method is the best method for set the parameter values. Further, genetic algorithm is a powerful tool to set the parameter values of chaotic logic gate.

1. Introduction

Chaotic systems are known by the richness of their dynamics. Richness of dynamics in these systems can be described as sensitivity to perturbation or instability. Recently, there has been a new research direction in chaos application that is exploiting the rich patterns inherent in the chaotic systems to do chaos computing. Chaos computing is attractive because it provides a new reconfigurable computing scheme with real circuit implementation models.

In 1998 Sinha and Ditto proposed a new computing scheme based on chaotic dynamical systems and showed the ability of chaotic maps to perform simple computations [1]. In 2002 Munakata, Sinha and Ditto described basic principles of implementing the most fundamental computing functions by chaotic elements [2]. The use of one-dimensional schemes like tent map and logistic map are possible to create different logic functions [2,3].In 2010 Ditto, Miliotis, Murali, Sinha and Spano used this feature that chaotic systems can yield a wide variety of patterns to produce main logic operators [4]. Low and one-dimensional chaotic systems can express a stunning variety of different behaviors as a function of time, of their initial conditions or of their parameters [5].

The searching for alternative solutions in the hardware used in computational systems is an important issue; One of the issues is developing reconfigurable hardware, where the chaotic circuits are candidates to perform these types of tasks [6–8]. As we know, chaotic elements might produce different logical functions with reconfiguration abilities [9,10].

Existing reconfigurable chips, called field programmable gate arrays (FPGAs), contain programmable interconnects that can be rewired to perform different functions. Some benefits of FPGA technology are in performance, time to market, cost, reliability and long term-maintenance, but for evolutionary computing applications, FPGAs are slow to reconfigure, typically taking milliseconds for each rewiring. Reconfiguring a chip within a single clock cycle would be a great benefit.

* Corresponding author. Tel.: +989151602242.
E-mail addresses: beyki.mshdiau@yahoo.com, mahmood.beyki@yahoo.com (M. Beyki), yaghoobi@mshdiau.ac.ir (M. Yaghoobi).

http://dx.doi.org/10.1016/j.chaos.2015.05.032

Chaotic logic gate developed by ChaoLogix Company, work by exploiting inherent chaotic behavior within the integrated circuits and makes it possible to morph an operator from one type into another in an instant. Having the ability to effectively redesign gates an unlimited number of times after they've been manufactured could make chaotic logic gate faster and more robust than those FPGAs.

The common notion that chaotic systems are unstable and unpredictable is not accurate. Such systems can be extremely sensitive to changes, and it is possible to produce desired states reliably and reproducibly provided you ensure that only minor changes are made to the inputs. Just making small changes to the input, you can adapt to do totally different things. This creates a greater degree of flexibility, because it makes more states available in a given system.

In short, chaotic logic gates have the following features and applications:

- The ability to switch easily between different logical operators.
- Reconfigurable ability to an unlimited number of times.
- These gates reduce production costs of integrated circuits.
- With these gates we can make faster chips.
- These gates can be used in computers.

In this paper, we use a logistic map to build a chaotic logic gate. This gate produces different logical basic functions. Setting the parameter values of this logic gate performs by genetic algorithm [11]. With this algorithm, building a single and two-input logic functions is possible. The main contribution of this paper is on a chaotic logic gate. In Section 2, we explain logistic map. In Section 3, we present a chaotic logic gate with reconfiguration abilities. In Section 4, we describe the genetic algorithm and use it to set the parameter values of chaotic logic gate. In Section 5, we give results of setting the parameter values by genetic algorithm. In the last section we give conclusions.

2. Logistic map

The logistic map is used as a basic demonstration of a mathematical equation that, despite its simplicity, can give rise to chaotic behavior. Chaos, in this context, simply means that the values generated by the logistic map, at a given $r$value, will become unpredictable.

Before we implement chaotic logic gate we should formulate the behavior of a chaotic system. We can use the one-dimensional, quadratic, logistic map to demonstrate complex, dynamic phenomena that also occur in chaos theory. The basic form of the logistic map is:

$$x_{n+1} = f(x_n) = rx_n(1 - x_n) \quad (1)$$

It is clear that the function is controlled by the three variables x_n, r, n; each in principle unbounded. There is no need though to go to such lengths, as we can confine ourselves to the domains $0 \leq x_n \leq 1$, $0 \prec r \leq 4$, $n \geq 0$.

In many applications, the map is a model for the dynamics of a population, and x_n is the population of the nth generation. By varying the parameter r, the following behavior is observed:

$0 \prec r \prec 1:$

In this case, $x_{n+1} = f(x_n) = rx_n(1 - x_n) = x_n$ gives us a single fixed point $x_n = 0$.

$1 \prec r \prec 3:$

In this case, solving $x_{n+1} = f(x_n) = rx_n(1 - x_n) = x_n$ gives us two fixed points $x_1 = 0, x_2 = 1 - 1lr$

$r \succ 3:$

In this case, both of the two fixed points are repelling. But the periodic points appear.

$3.57 \prec r \leq 4:$

For many values of $3.57 \prec r \leq 4$, the logistic map displays chaotic behavior.

In the following, we consider a special case of the logistic map by selecting $r = 4$. Although other values of r close to 4 can also be used, $r = 4$ guarantees that the logistic map displays chaotic behavior.

3. Chaotic logic gate with reconfiguration abilities

Chaos is typically understood as a mathematical property of a dynamical system and provides flexibility in the performance of a system and provides a wide range of dynamic behaviors that can be utilized to improve performance. Chaotic systems have three defining characteristics as follows:

- Sensitivity to initial conditions.
- Periodic long-term behavior. In other words, there are no fixed points, periodic orbits, or quasi-periodic orbits in the behavior of the system.
- Deterministic: This means that the chaos in the system is solely a function of the system's inherent nonlinearity as opposed to some other driving force.

We describe below a theoretical method for obtaining all basic logical operators with a chaotic element. The broad aim here is to use the rich temporal patterns embedded in a nonlinear time series in a controlled manner to obtain a computing device that is flexible and reconfigurable. Consider a chaotic element whose state is x_{gate}. See Fig. 1.

We implement chaotic logic gates as follows in three steps. In chaotic elements, we have corresponding actual input values denoted as I_1, I_2 or I, and actual output value denoted as Z. The actual values here are normalized rather than true physical values, and they range over [0, 1] including fractions. The relationships between interpreted and actual values (interpreted $\leftrightarrow$ actual), are: $(0 \leftrightarrow 0)$ and $(1 \leftrightarrow x_u)$. Here, x_u is a variable with positive value. In our implementation, we put x_u instead of 1 when output and inputs are 1 and we put 0 when output and inputs are 0.

(Step 1) we obtain initialization state of chaotic logic gate by adding external inputs to excitation state of this gate.

Two-input gates:

$$x_0 = x_{gate} + I_1 + I_2 \quad (2)$$

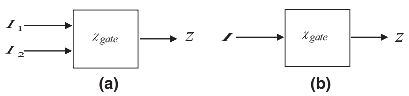

Fig. 1. (a) Two-inputs chaotic element; (b) Single-input chaotic element.