Application of Digital Chaos Module for an Equivalent Logistic Chaotic Equation

Ziheng Yang
School of Electronic Engineering, Heilongjiang University, Harbin150080, Heilongjiang, China

Lei Wang*
Information and Network Center, Heilongjiang University, Harbin150080, Heilongjiang, China
*Corresponding author(E-mail: wanglei@hlju.edu.cn)

Abstract
The chaotic deterministic nonlinear equation produces a random, irreversible, and dynamic signal with good pseudorandom sequence properties and has potential applications in chaotic sequence generators, chaotic cryptosystems, and chaotic secure communications. In this paper, the design of an equivalent chaotic system is proposed. The chaotic module has different characteristics from the Logistic chaotic module designed by SIMLINK description method in MATLAB. Its advantage is equivalent to the equivalent precision. The chaotic digital module can be expressed by the mathematical equation, and the chaotic characteristic can be realized without chaotic property analysis. Because the original chaotic sequence can be directly used in the chaotic encryption system.

Key words: Digital Chaos, Hardware Encryption, Logistic, Verilog.

1. INTRODUCTION

Today's information security has received sufficient attention, how to ensure the confidentiality, integrity and availability of information, this is a fundamental problem that needs to be solved urgently for the application of information security technology, the core of information security is always the research on cryptography theory and encryption technology (Kim and Lee, 2016; Shi and Zhang, 2016). While the chaotic deterministic nonlinear equation produces a random, irreversible, and dynamic signal, which has good pseudorandom sequence properties (Turgut and Coban, 2014), chaotic signals are extremely sensitive to initial parameters, setting different initial parameters will cause the system to run on different tracks, it is difficult to carry on the analysis and the extrapolation, is a deterministic system with extremely complex internal stochastic properties, the system is suitable for information security and confidential communications applications. These characteristics of chaos have similar characteristics of the encryption system, which makes chaotic encryption get full of concern (Liu and Kadir, 2016). The sensitivity of chaos make it more dependent on the digital system implementation, because the simulation system is influenced by environmental factors and device, it is difficult to construct a stable secure communication system. In the digital system implementation process, the implementation based on FPGA (Field Programmable Gate Array) achieves a great advantage, because FPGA inherits ASIC (Application Specific Integrated Circuit) of large-scale, high integration, high reliability advantages, and gradually become the ideal choice of complex digital hardware circuit design, so the realization of chaotic sequence generator or chaotic cipher based on FPGA is still the main design method of chaotic digital application (Chen and Ma, 2015; Fang and Wang, 2014).

The chaos realization method includes the analog circuit method and the digital circuit method, at present, the advantage of digital circuit implementation has been widely recognized by researchers, which has been researched more is chaotic sequence generator and digital chaotic secure communication (Hou and Chen, 2012), analog system implementation method is mainly based on chaos characteristic analysis and chaos weak signal detection. In the digital circuit implementation method, chaos operation itself is based on floating-point operations, but in Verilog language, only integers can be synthesized, of course, here can use signed keyword modifier to implement signed numbers, but there is no specific expression of floating point data keywords, so in the realization of chaos algorithm based on Verilog has some programming difficulty (Tlelo-Cuautle and Rangel-Magdaleno, 2016; Mangeot and Launay, 2012). At present, the use of the SIMULINK description method in MATLAB to realize Logistic chaotic equation occupies the main position (Luo and Duan, 2014), but the output chaotic sequence can not be consistent with the MATLAB simulation results, so whether the characteristics of digital chaotic sequence analysis is suitable for cryptographic applications become a heavy task, which increase the cycle of research and development. In addition, the use of more inherent computing core does not have advantages in resources and speed.
In this paper, we study a chaotic Logistic chaotic equation based on FPGA, using Verilog language to design, take advantage of floating point operation to realize a chaotic system which can follow the original mathematical expression under the agreed condition, do the comparison of the original study and use the Logistic chaotic equation described and achieved by SIMULINK in MATLAB, its characteristic is equivalent to the original kinetic state completely, omitted the analysis of the chaotic dynamic characteristics after the original approximation state, and under the agreed conditions, which can be expressed by the original chaotic formula, finished the accurate mathematical expression description of chaotic equations based on hardware logic, through this research, it can be extended to other chaotic sequence generators and chaotic sequence cipher. This paper proves that the chaotic system based on hardware logic can have the characteristics of the original dynamical system, because it is almost equivalent to the output of the dynamical system, in the application-oriented, avoiding the large data volume characteristics analysis process, has great significance to chaotic sequence generator and the application of chaotic sequence cipher.

2. IMPLEMENTATION METHOD OF LOGISTIC CHAOTIC SEQUENCE BY VERILOG

Different parameter choices have a significant effect on the chaotic output sequence, and the construction of chaotic block diagram is given according to the chaos equation, so the chaotic equation and its parameters should be first determined.

Logistic chaotic mapping is defined as follows:

\[ x_{n+1} = f(x_n) = \mu \cdot x_n (1 - x_n) \]

Among them, \( \mu \) is the key parameter, \( x_n \) indicates the previous state, \( x_{n+1} \) represents the latter state, different \( \mu \) will affect the system to show different states, figure 1 is the bifurcation diagram of MATLAB simulation of the Logistic chaotic map on the interval \([3.5,4]\), figure 2 is the corresponding phase diagram.

Figure 1. The bifurcation diagram of logistic map in \([3.5,4]\)

Figure 2. The phase diagram of system
From figure 1 we can see that the logistic chaotic map shows chaotic state with different values, so the value is chosen as 3.815, the initial value is 0.4 in the interval [0,1].

This paper uses ALTERA's hardware development platform to achieve the logic validation, so the synthesizer choose the company's QUARTUS II. In order to verify the easy, the high version of the ModelSim simulator has not been choose, so finally use QUARTUS II 9.0's simulation tools. From the equation (1) we can see that, logistic chaotic equations themselves are not complicated, its main operation is floating-point multiplication and subtraction. But it has been mentioned above that in the Verilog language, only intregetyypedata can be synthesized, and in the simulation tool, the data type is also limited to a signed number, unsigned number, fixed-point decimal, which can achieve the input and output. Thus, in the synthesis process, the multiplication operator can not use the "*" operator (because it can only synthesis fixed-point multiplier), the subtraction operator can not use the "-" operator as well (because it can only synthesis fixed-point subtractor). In this paper, the author wrote a special calculation for the calculation of Logistic equation and floating-point multiplier design and floating-point subtractor shown in figure 3 and figure 4:

![Floating point multiplier diagram](image1)

**Figure 3.** Floating point multiplier diagram of symbol

![Floating point adder diagram](image2)

**Figure 4.** Floating point adder diagram of symbol

In figure 3, dataa is the multiplicand, datab is the multiplier, the multiplicand, multiplier, and the result output are all expressed as 64-bit floating point numbers in order to be consistent with the matlab simulation data later, Clock is the primary clock, clk_en is the work enable signal, and aclr is the reset pin. Because the floating-point multiplier internal use pipeline structure to achieve floating-point operations, so the data can be continuously load into the multiplier in the rising edge of the clock and clk_en is high, when clk_en is low, the multiplier stops working. At any time during the course of work, if aclr is high the multiplier will reset and the result output is set to '0'. In figure 4, the situation is similar to figure 3, which is also denoted with 64-bit floating-point number, dataa is the multiplicand, datab is the multiplier.

With the floating-point multiplier and subtractor, the recursive equation of equation (1) has been clearly expressed. It can be seen from (1) realizing logistic equation recursive expression only need two floating-point multiplier and a floating-point subtractor, where u and '1' are constants. Coupled with the initial value of the chaotic system and recursive logic processing module, the author design the method based on Verilog HDL language in the environment of QUARTUS II schematic editor.

Chaotic equation (1) schematic diagram symbol file, the block diagram is shown in figure 5. In figure 5, logistic equation hardware expression operation’s main body is floating-point subtractor (represented by sub_float) and two floating-point multiplier (represented by mul_float), the newx module is a register used to implement the recursive function of the equation, main_stm module is used to control the state machine to control the entire circuit operation, loadkey is used to initialize the unit module to achieve the initial value of the Logistic equation set.
In the input pin, the "u" pin is the equation coefficient, the "cons" pin is the constant "1"., the "clk" pin clock frequency, the "rst" pin is the reset pin, the "load" pin is the initial load control pin, when it is high, the 64-bit initial value of the "key" pin is latched to the "r_key" output.

On the output pin, the "x" pin is the current chaotic value and the "y" pin is the next chaotic value. The "newdt" pin is used to control the update of the chaotic value. After each calculation of the logistic recursive value, "newdt" generates a control pulse to update the "y" value to the "x" side. The "work" pin is used to control the clock enable signal of the arithmetic unit, "de" is used to indicate the output of valid data.

A logistic recursive operation is performed in the manner described below. The "u" value is constant in system, in this paper, the value is 3.815, "key" is the initial value of the Logistic system, this value is 0.4, the "key" value is latched to the "r_key" terminal when the "load" pin is high, and "rdy" is set low, then the state machine, floating-point subtractor and floating-point multiplier are in a reset state. After the "load" pin returns low, the state machine will control the circuit, and assign the value of "r_key" terminal to the "x" terminal to prepare for a logistic operation. After this because the "cons" pin is constant '1' and the "dataa" terminal is the minuend, which is sent to the floating-point subtractor, "x" side data is sent to the "datab" end of the floating-point subtractor as a subtrahend, after a few clocks the difference is out from the "result" side of the floating-point subtractor. The result of the output of the floating-point subtractor and the "x" side data are fed to the first floating-point multiplier as the multiplicand and the multiplier, assign the value of "u" side are fed into the second floating-point multiplier. The result of the output of the first floating-point multiplier and the value of the "u" side are fed into the second floating-point multiplier as the multiplicand and...
the multiplier, respectively, after a few clocks, the product result is out from the "result" side of the floating-point multiplier as "y".

At this point a logistic recursive operation is successfully completed, the state machine will generate a "newdt" pulse to update the "y" value to the "x" side at this time to prepare for the next Logistic recursive operation. The following figure shows the timing simulation for completing three times iterations of logistic recursion, which is completely simulated with QUARTUS II 9.0’s simulation tool.

3. DIGITAL SYSTEM EXPERIMENTS OF LOGISTIC CHAOTIC SEQUENCES

In the previous section, we have described in detail the working principle of the hardware expression circuit system of Logistic equation based on verilog HDL, and gives the timing simulation diagram by the simulation software carried by QUARTUS II 9.0 itself. From figure 6, we only see the timing of the circuit system and work process, but because all the data generated are displayed in hexadecimal, so it is not enough to explain the correctness of the logic of the chaotic equation circuit. Therefore, the next content is comparing the floating point generated by Matlab simulation with the results of hardware timing simulation to make sure the correctness of the circuit logic finally.

We know that double-precision floating-point numbers are represented by 64-bit binary numbers, the most significant bit is the sign bit, the middle 11 bits are the index bits, and the lower 52 bits are the mantissas. To this end, this article get the mantissa through the exponential bit value and make the 64-bit double-precision floating-point number into a fixed-point integer and fractional two-part to display.

In order to facilitate comparison, the author write the Matlab simulation code based on the formula (1) and generate 100 Logistic iterative output values between 0 and 1 to store in the double variable chaos, among them the value of u is 3.815, and the initial value is 0.4. The intermediate data of the simulation result is used in verilog hardware debugging, the final matlab simulation result is shown as in figure 7. As the contrasted data, use the first 10 chaotic output values which has the initial value is 0.4 to make the screen shot of the chaos variable is shown in figure 8.

![Figure 7. Logistic iterative operation Matlab simulation diagram](image7)

![Figure 8. Top 10 logistic data simulation diagram](image8)
In the QUARTUS II 9.0 own simulation tool, the types that can be selected for the input data are ASCII, Binary, Fractional, Hexadecimal, Octal, Signed Decimal, and Unsigned Decimal, among them only the fractional type can represent a fraction. However, fractional can only be expressed as the normalized fraction between 0 and 1, more than 1 decimal input is illegal. So the author divide a decimal fraction into two parts as the integer and decimal expression to achieve hardware logic timing simulation based on verilog HDL in the simulation process.

To this end, the author specially prepared a floating-point to hexadecimal number module and hexadecimal to floating-point module used to facilitate the simulation data display, so it is easy to make a compare between matlab simulation data and timing simulation. The timing simulation results are shown in figure 9.

<table>
<thead>
<tr>
<th>Name</th>
<th>rst</th>
<th>clk</th>
<th>reset</th>
<th>init</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
</tr>
</tbody>
</table>

**Figure 9.** Top 10 logistic equation floating-point interface timing simulation diagram

It can be seen from figure 9 that the integer part of the value of u is denoted by ui, and the fractional part is denoted by uf. The integer part of the initial value is denoted by ki, and the fractional part is denoted by kf. The integer part of the output is represented by mi and the fractional part is represented by mf. Each de-rising edge will come out a chaotic value, due to limited space, the system can only have 8 values output, after contrast that are exactly the same with matlab simulation data, unless the fractional part of the data representation is more accurate.

The digital chaotic module can be directly applied to the data transmission encryption system which is shown in figure 10, the system uses full duplex mode when two devices communicate, uses the serial port level conversion chip to complete the level conversion. Every system chip is divided into data encryption channel and data decryption channel, and the chaotic encryption kernel provides the encryption core algorithm. The data encryption channel and data decryption channel is divided into seven parts [20]: asynchronous serial receiving module, adding and deciphering module, chaotic key sequence generating module, public key arithmetic module, main control module, multiplexing module, asynchronous serial sending module, we will replace the chaotic key sequence generation module with the equivalent dynamic chaotic system by using the digital chaotic module in this paper.

**Figure 10.** Data encryption transmission function block diagram

4. CONCLUSIONS

This paper describes a method to determine the Logistic chaotic mapping equation by verilog HDL, and do editing, synthesis and timing simulation for verilog HDL language by ALTERA's QUARTUS II software development system, generate the Logistic chaotic mapping equation module circuit and get the timing
simulation result graph. At the same time, it is proved that the logistic chaotic map equation can be used to determine the realizability of the digital chaotic module by comparing with the matlab simulation result, the advantage of the digital chaotic module is under the same precision operation condition, can be completely equivalent to the original kinetic logistic chaotic equation, so that the chaotic digital module can be expressed by the mathematical equation and generated without chaotic characteristics analysis, as a result of preserving the original chaotic sequence random characteristic that can be directly applied to the chaotic encryption system, which play an important role to promote the application of hardware-based chaotic encryption in future.

ACKNOWLEDGEMENTS

This work was supported by National Natural Science Foundation of China(Grant Nos.61072072).

REFERENCES