International Journal of Hybrid Information Technology
Vol.8, No.4 (2015), pp.97-104
http://dx.doi.org/10.14257/ijhit.2015.8.4.12

The Hardware Circuit Design, Generation, Download and Tests of Chaotic Sequence Model

Lina Ding[1], Qi Chen[2] and Qun Ding[3]

1. Suihua University
2. Harbin COSLIGHT Electrical Automation Co.Ltd.
3. Heilongjiang University
dinglina@aliyun.com

Abstract

In the field of information security, chaos algorithm and its application in hardware encryption have important significance. The paper here takes a normally used Lorenz chaotic system as an example and introduces its FPGA hardware circuit design; generation and download method, then, statistical tests and instrument test are done. Through the test, it is demonstrated that the chaotic sequence generated by the circuit has good statistical properties and chaotic characteristics. By this way, researchers can easily and quickly get and test chaotic sequences in experimental environment.

Keywords: *Chaotic Sequence; Lorenz System; FPGA; Statistical Tests; Instrument Test*

1. Introduction

The security of many encryption systems is relied on the generation of non-predictable variables, which must be long and random enough. As everyone knows that chaos system has inherent nonlinear, initial value sensitiveness, periodicity and pseudorandom characteristics, which means that it is a kind of good random numbers generator. Hardware encryption is widely used in information security field. Compared with software encryption, it is good for encryption scheme. Large scale integrated circuit, FPGA is applied widely in modern digital equipment with its large capacity, complicated function and high reliability. More and more researchers begin to use it in hardware encryption system. In this study, chaotic sequence and FPGA are connected together to ensure the encryption is strong enough in both hardware encryption and random sequence. Here it is learned the normally cited and discussed Lorenz dynamic system first, and then the Lorenz equation is quantified, after equation model is established in DSP Builder environment, it is downloaded into FPGA. The statistical and instrument tests are done at last to ensure the sequences are secure enough and random enough.

2. Chaotic Sequence Model and Quantization

2.1. Chaotic Sequence Model

Chaos signals are a kind of pseudorandom, irreversible and dynamical signals generated by deterministic nonlinear equations, which process good characteristics of pseudorandom sequences and they are highly sensitive to initial parameters and if you set different initial values, the system will run in different orbits which are difficult to be analyzed and calculated. Chaos systems are fit for information security and secure communication fields [1-2].

ISSN: 1738-9968 IJHIT

Here we start from Lorenz chaos equations and establish its chaos system module and finally the chaos sequences are generated. Lorenz system is a nonlinear differential equation set; each variable is the function of time t . Lorenz system not only has bifurcation and chaotic phenomena but also diversified stability phenomena such as multiple periods and stagnant dots. The structure of this kind of system is complicated and has many system variables which are much more irregular and uncertain and has many system parameters, so it can be used in secure communication field. The mathematics model of Lorenz equation set is shown as (1):

$$\begin{cases} \frac{dX}{dt} = 9(Y - X) \\ \frac{dY}{dt} = 35X - Y - 20XZ \\ \frac{dZ}{dt} = 5XY - 1.5Z \end{cases} \tag{1}$$

2.2. Quantization of Chaotic Sequence

In order to generate chaotic sequence $x(n)$ by digital integration method, it is required to change the output sequence of equation (1) into binary sequence $s(n)$ [3].

$$T[x(n)] = \begin{cases} 0 & x(n) \in \bigcup_{k=0}^{2^m-1} I_{2k}^m \\ 1 & x(n) \in \bigcup_{k=0}^{2^m-1} I_{2k=1}^m \end{cases} \tag{2}$$

Here $m > 0$ and is an arbitrary integer, $I_0^m, I_1^m, I_2^m, \cdots\cdots$ are 2^m continuous equal intervals in $[0,1]$. The exchanged values are spread in corresponding interval of the quantification function, and then the values 0 or 1 are gotten. As chaotic signal $x(n)$ has a good random statistical characteristic, the stream $s(n)$ after quantification should have the same statistical characteristic.

3. Chaotic Sequence Circuit Design

3.1. Chaotic Sequence Model

In the design of Lorenz chaotic sequence circuit, adder, delay, multiplier, amplifier, data selector and digital integrator in DSP Builder library are used, the initial value setting is done by data selector. The quantification circuit is composed by barrel shift register and selector. The principle is shown as:

In order to simplify the hardware circuit and make the function value is in $[0,1]$. The chaotic output signals $x(n), y(n), z(n)$ are converted into signals $|x(n)|, |y(n)|, |z(n)|$ and then are compressed into $[0,1]$.

The sampling value is $X = \{ x(n) \mid n = 0,1,2 \cdots\cdots, x(n) \in [0,1] \}$, and according to equation (2), the stream value after quantification is $S = \{ s(n) \mid n = 0,1,2, \cdots\cdots, s(n) \in \{0,1\} \}$, quantification unit is $\Delta = 1/2^m$, where m is an arbitrary positive integer and the quantification interval is $[0\Delta\ \ 1\Delta) \cup [1\Delta\ \ 2\Delta) \cup [2\Delta\ \ 3\Delta) \cdots\cdots \cup [(2^m - 1)\Delta\ \ 2^m\Delta]$, here $k = 0,1,2, \cdots\cdots, 2^m - 1$. So the quantification function (2) can be realized by the following function.

$$s(n) = \begin{cases} 0, & x(n) \in [\,2k\Delta\quad (2k+1)\Delta\,) \\ 1, & x(n) \in [(2k+1)\Delta\quad (2k+2)\Delta] \end{cases} \tag{3}$$

In order to get a much easier circuit, a linear transformation to equation (3) is done.

$$s(n) = \begin{cases} 0, & 2^m x(n) \in [\,2k\quad (2k+1)\,) \\ 1, & 2^m x(n) \in [(2k+1)\quad (2k+2)] \end{cases} \tag{4}$$

Namely, the quantification unit is $\Delta = 1$, the whole quantification interval is $[0\ \ 1) \cup [1\ \ 2) \cup [2\ \ 3) \cdots\cdots \cup [(2^m - 1)\ \ 2^m]$, so the quantification interval can be judged from the integer part of $2^m x(n)$. According to the unit of the integer part, 0 or 1 of the output stream can be decided. The circuit diagram is shown as:

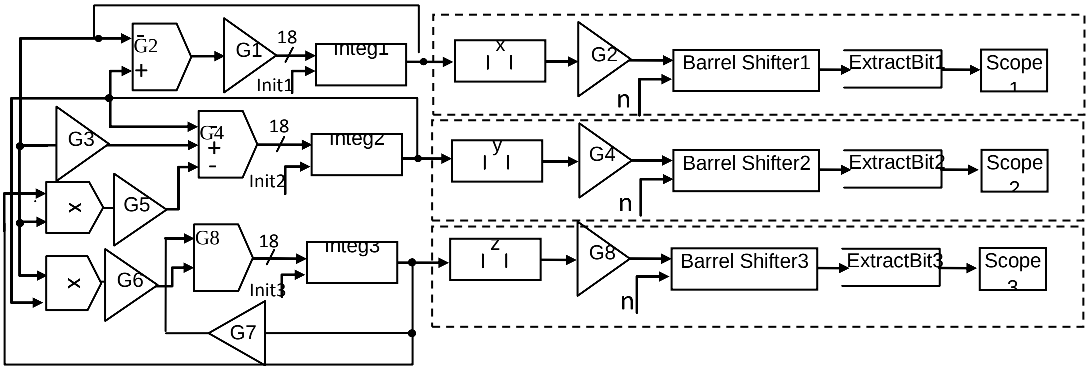


Figure 1. Lorenz Sequence Circuit

3.2. Matlab Simulation Circuit of Chaotic Sequence

The circuit is designed by DSP Builder tool based on FPGA technology from Altera Company. DSP Builder integrates the algorithm development, simulation, and verification capabilities of MathWorks MATLAB and Simulink system-level design tools with the Altera Quartus II software and third-party synthesis and simulation tools. You can combine Simulink blocks with DSP Builder blocks to verify system level specifications and perform simulation [4]. DSP Builder System-Level Design Flow is shown as:

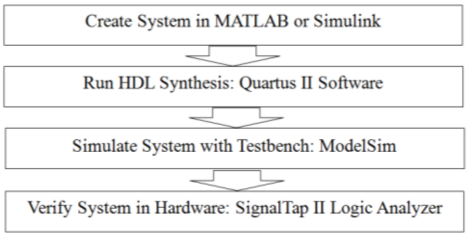

Create System in MATLAB or Simulink
Run HDL Synthesis: Quartus II Software
Simulate System with Testbench: ModelSim
Verify System in Hardware: SignalTap II Logic Analyzer

Figure 2. DSP Builder System-Level Design Flow

4. Chaotic Sequence Hardware Download and Generation

The Quartus II software can complete the synthesis and place-and-route process. Here the blocks in DSP Builder is automatically loaded into Quartus II software by clicking on the Run Quartus II block in the top-level model. Then the project files are downloaded into SOC/SOPC experiment and development system. The simulation diagram is shown as:

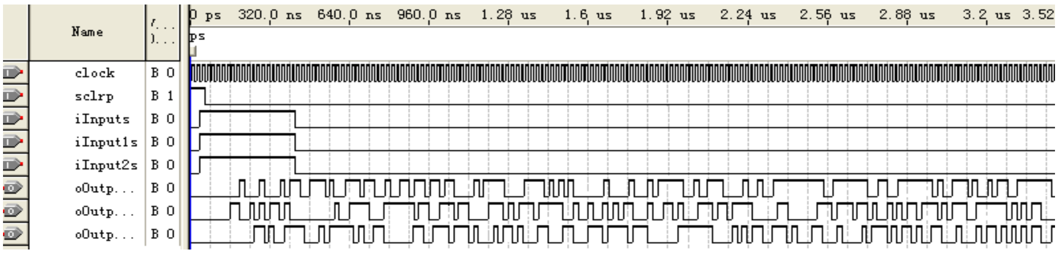


Figure 3. Simulation Diagram of Chaotic Sequences in Quartus II

5. Chaotic Sequence Tests

5.1. Statistical Tests of Chaotic Sequence

Here five statistical tests [5-6] are used to weigh the quality of the above chaos system. The tests here can be used to detect the demerits of the pseudorandom sequences generated by the system. Each test can ensure whether the sequence owns the characteristics of random sequences and if the sequence of this chaos system fails any test, it is considered to be insecure.

5.1.1 Frequency Test: Here let $s = s_0, s_1, ..., s_{n-1}$ be a binary sequence of length n (and it is the same in the following four tests).The frequency test [5] is used to make sure whether the 0's and 1's in s is approximately equal to each other. Let n_0 and n_1 stand for the number of 0's and 1's in s and the statistic used is

$$X = \frac{(n_0 - n_1)^2}{n} \tag{5}$$

The statistic approximately follows a χ^2 [6] distribution with 1 degree of freedom if $n \geq 10$. For significance level $\alpha = 0.05$, the threshold of statistic X is 3.8415.

In this test, the three Lorenz output sequences x, y, z are tested and in order to enlarge the period for limited precision effect and get a better contrast, we xor the three output sequences and get a new sequence $x \oplus y \oplus z$. So there are four equipment data (all the chaos output sequences in the five tests have filtrated the ahead 2000 datum), the result is shown in table 1:

Table 1. The Frequency Test Datum of Lorenz System

Output	0'S	1'S	Iteration	χ^2
x	500501	499499	100，0000	1.0040
y	500197	499803	100，0000	0.1552
z	499886	500114	100，0000	0.0520
$x \oplus y \oplus z$	499903	500097	100，0000	0.0376

From the table we can see the thresholds are all smaller than the standard ones, so the four sequences pass the frequency test.

5.1.2 Serial Test: The serial test [5] is used to make sure whether the subsequences 00,01,10,11 occur approximately the same number of times. Let $n_{00}, n_{01}, n_{10}, n_{11}$ stand for the occur times of sequences 00,01,10,11. Note that $n_{00} + n_{01} + n_{10} + n_{11} = (n-1)$, the statistic used here is

$$X = \frac{4}{n-1}\left(n_{00}{}^2 + n_{01}{}^2 + n_{10}{}^2 + n_{11}{}^2\right) - \frac{2}{n}\left(n_0{}^2 + n_1{}^2\right) + 1 \qquad (6)$$

The statistic approximately follows a χ^2 [6] distribution with 2 degree of freedom if $n \geq 21$. For significance level $\alpha = 0.05$, the threshold of statistic X is 5.9915. The result is shown in Table 2:

Table 2. The Serial Test Datum of Lorenz System

Output	00	01	10	11	0	1	Iteration	χ^2
x	249663	249984	249984	250369	500501	499499	1,000,000	0.9970
y	249688	250222	250222	249868	500197	499803	1,000,000	1.6981
z	249972	250160	250160	249708	499886	500114	1,000,000	1.4970
$x \oplus y \oplus z$	250276	249769	249769	250186	499903	500097	1,000,000	1.8323

From the table we can see the thresholds are all smaller than the standard ones, so the four sequences pass the serial test.

5.1.3 Poker Test: Let m be a positive integer that $\left\lfloor \frac{n}{m} \right\rfloor \geq 5 \cdot \left(2^m\right)$ and $k = \left\lfloor \frac{n}{m} \right\rfloor$ First divide sequence s into k subsequences each with length m and let n_i be the occur times of the i th sequence with length m and $1 \leq i \leq 2^m$. Poker test [5] is used to make sure whether the occur times of each sequence with length m is approximately the same. Here we choose m =3 and the statistic used here is

$$X = \frac{2^m}{k}\left(\sum_{i=1}^{2^m} n_i{}^2\right) - k \qquad (7)$$

The statistic approximately follows a χ^2 [6] distribution with $2^m - 1 = 7$ degrees of freedom. For significance level $\alpha = 0.05$, the threshold of statistic X is 14.0671. The result is shown in Table 3:

Table 3. The Poker Test Datum of Lorenz System

Output	The numbers of the sequences with length 3								Iteration	χ^2
	000	001	010	011	100	101	110	111		
x	4178	4142	4167	4218	4136	4168	4094	4232	99999	3.3269
y	4145	4240	4169	4130	4190	4113	4167	4181	99999	2.5983
z	4205	4144	4210	4098	4143	4118	4201	4216	99999	3.6279
$x \oplus y \oplus z$	4188	4151	4249	4152	4207	4091	4144	4153	99999	3.7791

From the table we can see the thresholds are all smaller than the standard one, so the four sequences pass the poker test.

5.1.4 Runs Test: Let s be a sequence. A run of s is a subsequence of s consisting of consecutive 0's or consecutive 1's which is neither preceded nor succeeded by the same symbol. A run of 0's is called a gap, while a run of 1's is called a block [5].

The run test can be used to determine whether the different run length (gaps or blocks) are as those in a random sequence. In a random sequence with length n, the expected number of gaps (or blocks) of length i is $e_i = (n - i + 3)/2^{i+2}$. Let k be equal to the largest integer i for which $e_i \geq 5$ and B_i , G_i be the number of blocks and gaps, respectively, of length i in s for each i that fulfills $1 \le i \le k$ and the statistic used here is

$$X = \sum_{i=1}^{k} \frac{(B_i - e_i)^2}{e_i} + \sum_{i=1}^{k} \frac{(G_i - e_i)^2}{e_i} \tag{8}$$

The statistic approximately follows a χ^2 [6] distribution with $2k-2$ degrees of freedom. Here k =10 by test and for significance level $\alpha = 0.05$, the threshold of statistic X is 28.8693. The result is shown in Table 4:

Table 4. The Run Test Datum of Lorenz System

Output	Sequence Length	Threshold	χ^2
x	10,000	28.8693	19.1648
y	10,000	28.8693	13.5529
z	10,000	28.8693	11.8142
$x \oplus y \oplus z$	10,000	28.8693	15.0321

From the table we can see the thresholds are all fulfill the standard, so the four sequences pass the poker test.

5.1.5 Autocorrelation Test: Autocorrelation test [5] can be used to check the correlation between s and (non-cyclic) shifted version of s .Let d be a shift that fulfills $1 \le d \le \lfloor n/2 \rfloor$ .The different number of bits in s and their d-shifts is $A(d) = \sum_{i=0}^{n-d-1} s_i \oplus s_{i+d}$ and the statistic used here is

$$X = 2\left(A(d) - \frac{n-d}{2}\right) \Big/ \sqrt{n-d} \tag{9}$$

The sequence approximately follows an $N(0,1)$ distribution if $n - d \ge 10$ .Since small values of $A(d)$ are as unexpected as large values of $A(d)$, a two-sided test should be used. For significance level $\alpha = 0.05$, the threshold of statistic X is 1.96. The result is shown in table 5:

Table 5. The Autocorrelation Test Datum of Lorenz System

Output	Iteration	χ^2
x	200，0000	-0.3215
y	200，0000	0.4886
z	200，0000	0.5598
$x \oplus y \oplus z$	200，0000	0.2351

From the test, it can be known that all the thresholds are between +1.96 and -1.96, which implies that the sequences above pass the correlation test.

5.2 Instrument Test of Chaotic Sequence

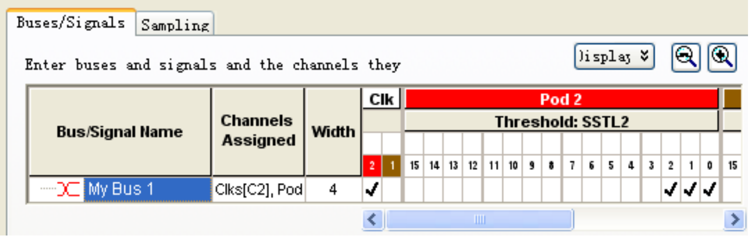
5.2.1 Logic Analyzer: A logic analyzer is an electronic instrument that captures and displays multiple signals from a digital system or digital circuit. A logic analyzer may convert the captured data into timing diagrams, protocol decodes, state machine traces, assembly language, or may correlate assembly with source-level software. Logic

Analyzers have advanced triggering capabilities, and are useful when a user needs to see the timing relationships between many signals in a digital system[7].

5.2.2 The Test of Chaotic Sequence Using Logic Analyzer:

(1) Bus/Signal configuration

As the threshold voltage of the experiment box is 1.25v, the threshold voltage of the analyzer is 1.25v. The diagram of Bus/Signal configuration is shown as:



Enter buses and signals and the channels they

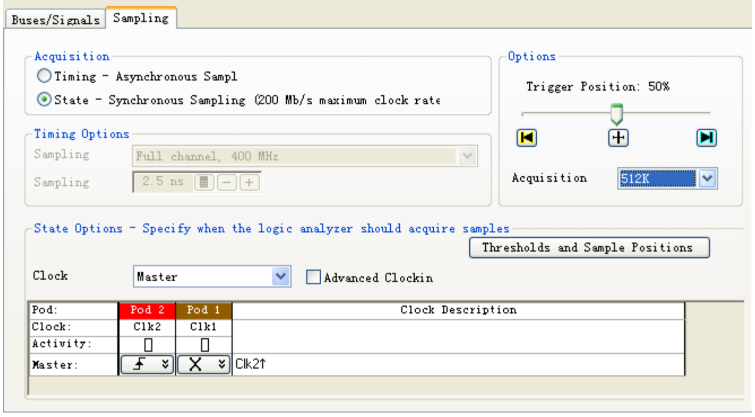
Bus/Signal Name	Channels Assigned	Width	Clk		Pod 2																
					Threshold: SSTL2																
			2	1	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	15
My Bus 1	Clks[C2], Pod	4	✓															✓	✓	✓	

Figure 4. Bus/Signal Configuration

(2) State Analyzer and Sampling Configurations

In the test, synchronous sampling mode is chosen, the steps are as follows:

Choose Setup>Timing/State (Sampling) option from the menu bar; choose State - Synchronous Sampling option from sampling configuration dialog box; set up the storage depth 512K and trigger position 50%.



Acquisition
○ Timing - Asynchronous Sampl
● State - Synchronous Sampling (200 Mb/s maximum clock rate

Timing Options
Sampling: Full channel, 400 MHz
Sampling: 2.5 ns

Options
Trigger Position: 50%
Acquisition: 512K

State Options - Specify when the logic analyzer should acquire samples

Clock: Master
□ Advanced Clockin

Pod:	Pod 2	Pod 1	Clock Description
Clock:	Clk2	Clk1	
Activity:	□	□	
Master:	↑	X	Clk2↑

Figure 5. Synchronous Sampling Mode

(3) The chaotic sequences from Logic Analyzer is shown below:

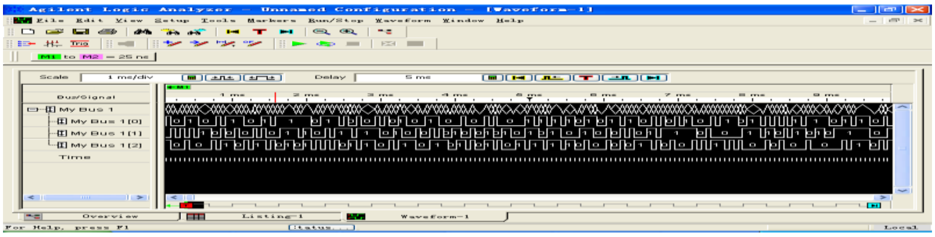


Figure 6. Chaotic Sequences from Logic Analyzer

6. Conclusion

In the paper, the way to realize continuous chaotic output sequences is studied and the chaotic sequences are gotten by FPGA. From the test, it is shown that the characteristics of chaos are achieved by hardware circuit. The Lorenz chaotic sequences can be used in chaotic coding secure communication which can improve the safety of secure communication. The method here is proved that it can be used to easily and quickly generate good chaotic sequences in hardware environment. Finally, any chaotic circuit can be achieved by this method in laboratory environment.

7. Acknowledgements

This research was financially supported by the Outstanding Youth Foundation Project of Suihua University, No.SJ13005.

8. References

8.1. Journal Article

[3] Y. Qiu, Chen He and H. Zhe, “一种无限折叠混沌映射及其量化序列[J]”, 上海交通大学学报, vol. 36, no. 12, (**2002**), pp. 1788-1790.

8.2. Book

[1] X. Wang, “Chaos in Complex Nonlinear System [M]”, Peking: Publishing House of Electronics Industry, (**2003**), pp. 46-47.

[2] R. Huang, “Chaos and Its Application [M]”, Wuhan: Wuhan University Press, (**2003**), pp. 46-47.

[4] HB_DSPB_INTRO-4.0, DSP Builder Handbook [M], Altera, (**2013).**

[5] A. J. Menezes, P. C. Van Oorschot and S. A. Vanstone, “Handbook of Applied Cryptography [M]”, CRC Press, (**1996**0, pp. 180-182.

[7] Agilent Technologies, Inc. Retrieved (**2012**) November 28.

8.3. Conference Proceedings

[6] J. Kimura and H. Shibasaki, “Recent Advances in Clinical Neurophysiology [C]”, Proceedings of the 10th International Congress of EMG and Clinical Neurophysiology, (**1995**) October 15-19, Kyoto, Japan.

Author

Lina Ding, Postgraduate, graduated from Heilongjiang University with a degree in Signal and Information Processing, engaged in Chaotic Information Security and Secure Communication, now is working in School of Electrical Engineering, Suihua University, mainly engaging in scientific research and teaching work.