ATLANTIS PRESS

Advances in Engineering Research, volume 118

2nd International Conference on Automation, Mechanical Control and Computational Engineering (AMCCE 2017)

FPGA Realization of Duffing Chaotic Oscillator Based on Runge-Kutta Algorithm

Beiming Li [a], Yixin Xu [b], Hongyang Shi [c] and Wei Xue [d]

School of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China

[a]libeiming@hrbeu.edu.cn

Keywords: Runge-Kutta algorithm, Duffing chaotic oscillator, FPGA.

Abstract: Duffing chaotic oscillator shows good detection and communication effect since it is sensitive to initial condition and has good resistance to noise. This paper presents an effective method for the digital hardware realization of Duffing chaotic oscillator on FPGA by using the Verilog HDL hardware description language directly and the fourth-order Runge-Kutta algorithm. Firstly, deduces the iterative process of Duffing chaotic oscillator according to the fourth-order Runge-Kutta algorithm. Secondly, uses the Verilog HDL language directly to the hardware realization of chaotic oscillator. Finally, outputs the generated signal on FPGA by the designed high-speed DAC. This method solves the problem that Duffing chaotic oscillator cannot realize the strict matching in the analog circuit because of the parameter errors due to component manufacturing and aging. This method also consumes fewer FPGA resources and the generated Duffing chaotic oscillator can be invoked easily by other modules for signal detection or communication. As a result it is practicable for use.

1. Introduction

In 1963, the fellow of the American Academy of Sciences E.N.Lorenz found Lorenz system [1]. The system becomes the starting point and cornerstone of chaos theory research [2, 3]. As a kind of chaotic oscillator, Duffing chaotic oscillator is suitable for weak signal detection and communication because of its good anti-noise ability. In the analog circuit, the same type of components of the parameters have a certain error, and has the components of the aging problem. Chaotic oscillator is high sensitivity, so it will lead to a problem that Duffing chaotic oscillator can detect actually or communication signal frequency or the required driving force amplitude and the theoretical value having a certain error when using the analog circuit to build Duffing chaotic oscillator, affecting the use. Therefore, we use the FPGA digital realization method [4], the precise generation of Duffing chaotic oscillator, effectively avoid such problems. The current FPGA realization of Duffing chaotic oscillator is based on the tools such as DSP Builder. This approach is simple, but the subsequent processing is very inconvenient, and cannot be called by other modules. In this paper, we will use the Verilog language to directly describe the Duffing chaotic oscillator [5] and use the fourth-order Runge-Kutta algorithm [6] to ensure the accuracy. In addition, the use of fixed-point decimal conversion to floating-point decimal way to sufficiently reduce the amount of logic resources.

2. Realization of Duffing Chaotic Oscillator Based on Fourth-Order Runge-Kutta Algorithm

2.1 Duffing Chaotic Oscillator Model

The Duffing chaotic oscillator is one of the nonlinear vibration equations, which is the vibration model of the soft spring vibrator. The expression of the Duffing chaotic oscillator model is as follows:

$$\begin{cases} dx/dt = y \\ dy/dt = x - x^3 + F\cos(wt) - ky \end{cases} \tag{1}$$

k is the damping coefficient of the nonlinear elastic system, $x - x^3$ is the nonlinear restoring force, when F is equal to 0, it is doing the free vibration, $F\cos(wt)$ is the external periodic disturbance

force. Duffing equation is a representative of nonlinear equations of motion, and its motion exhibits abundant nonlinear dynamics characteristics. After a lot of simulation experiments, Duffing oscillator can produce chaos.

When F is equal to 0, the Duffing equation does not be interfered by the periodic force. By calculating the system has a saddle point (0,0), two focal points (+1,0) and (-1,0), the oscillator will end up in one of two focal points.

When F is not equal to 0, the system has periodic perturbations, and the motion state of the oscillator becomes very complicated. With the different values of F, the Duffing oscillator has a complicated dynamic form. After a lot of simulation and numerical calculation, Duffing oscillator can be divided into periodic oscillation, homoclinic orbit, period-doubling bifurcation, chaotic state, critical state, large-scale periodic state. For example, when $F=0.822$ $k=0.5$ $w=1$, the chaotic oscillator is in a chaotic state. Matlab simulation as shown:

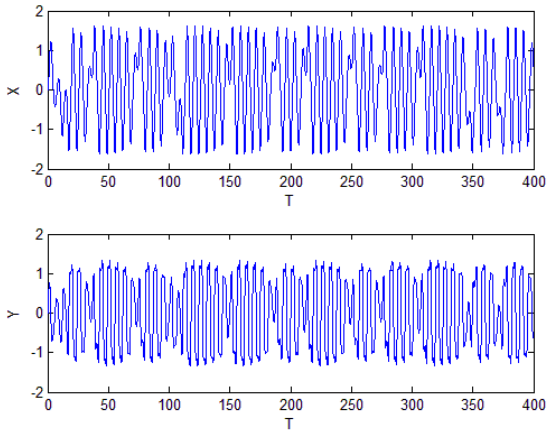

Fig. 1 Time domain signal of two channels

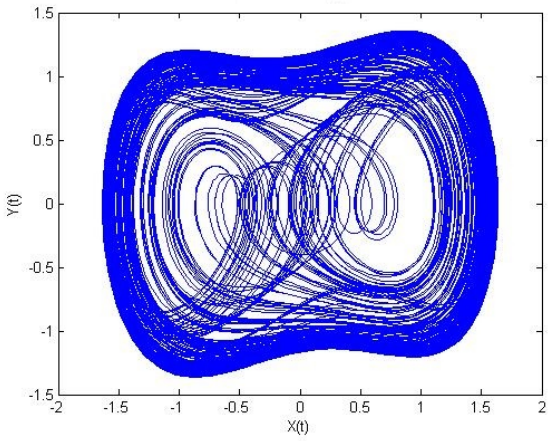

Fig. 2 Phase diagrams of Duffing oscillator

2.2 FPGA Realization

The Duffing chaotic oscillator is first transformed in proportion, let $w=10000$, used to improve the operation speed. The formula is as follows:

$$\begin{cases} dx/dt = wy \\ dy/dt = w(x - x^3 + F\cos(wt) - ky) \end{cases} \tag{2}$$

The value of F and k do not change, $k=0.5$, $F=0.822$.

Then use fourth-order Runge-Kutta algorithm to decompose the transformed Duffing chaotic oscillator.

$$x_{n+1} = x_n + \frac{h}{6}(f_1 + 2f_2 + 2f_3 + f_4) \tag{3}$$

$$y_{n+1} = y_n + \frac{h}{6}(g_1 + 2g_2 + 2g_3 + g_4) \tag{4}$$

Thereinto:

$$f_1 = wy_n \qquad g_1 = w(x_n - x_n^3 + F\cos(wt_n) - ky_n)$$

$$f_2 = w(y_n + \frac{h}{2}g_1) \qquad g_2 = w((x_n + \frac{h}{2}f_1) - (x_n + \frac{h}{2}f_1)^3 + F\cos(w(t_n + \frac{h}{2})) - k(y_n + \frac{h}{2}g_1))$$

$$f_3 = w(y_n + \frac{h}{2}g_2) \qquad g_3 = w((x_n + \frac{h}{2}f_2) - (x_n + \frac{h}{2}f_2)^3 + F\cos(w(t_n + \frac{h}{2})) - k(y_n + \frac{h}{2}g_2))$$

$$f_4 = w(y_n + hg_3) \qquad g_4 = w((x_n + hf_3) - (x_n + hf_3)^3 + F\cos(w(t_n + h)) - k(y_n + hg_3))$$

Where h is the step size, this algorithm can calculate the next solution of Xn after step h and solve the solution of Duffing chaotic oscillator in turn. Using this method, compiled into Verilog HDL language, we can achieve the Duffing chaotic oscillator by FPGA. The hardware realization state diagram is shown in Fig. 3:

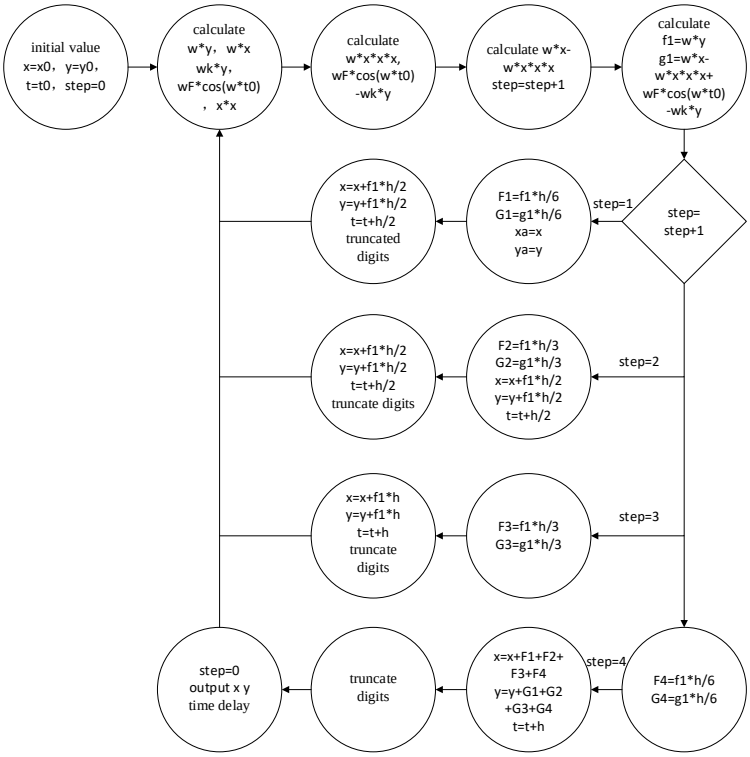

Fig. 3 Hardware realization of Duffing chaotic oscillator

In the implementation of the above operation, we use 64-bit fixed-point conversion of floating-point way to reduce the amount of logic resources. In which the last 32 bits for the decimal places, the middle 31 bits for the integer, the first bit is the sign bit, to ensure that data will not overflow. FPGA board uses a 20M crystal, the driving force frequency $w=10000$, using 256 points represent a cycle, each using three points and multiplexing a point to complete a fourth-order Runge-Kutta algorithm operation. Each state in the above state diagram is completed in one clock. From the state diagram shows that the completion of an output need to iteration 4 times, the first 3 iterations need 6 clocks to complete, the last 1 need to calculate the final output, so need 8 clocks to complete, a total of 26 clock cycles. To complete a computing time theory requires h=4.9*10-6s, so the need to complete the final calculation to do 72 clock delays. So that the actual time and theoretical time are consistent to ensure that the Duffing chaotic oscillator output is correct.

2.3 Experimental Test Results

Using Cyclone IV EP4CGX150DF27I7N hardware realize Duffing chaotic oscillator using 8 bit DA for the observation of the output signal, the design of the DA circuit as shown in Fig. 4:

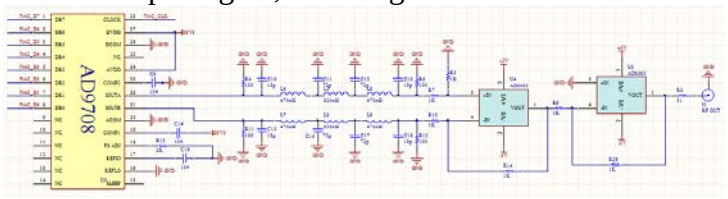

Fig. 4 The design of DA circuit

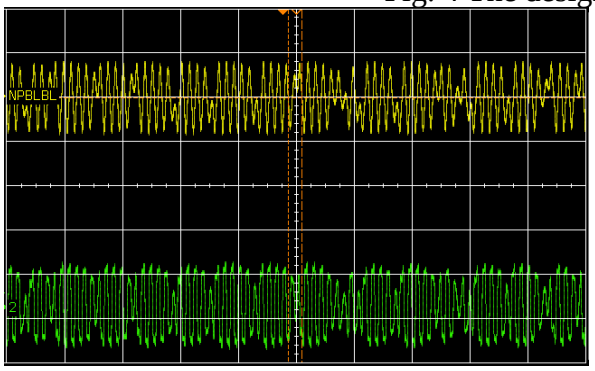

Fig. 5 FPGA realization of time domain signal

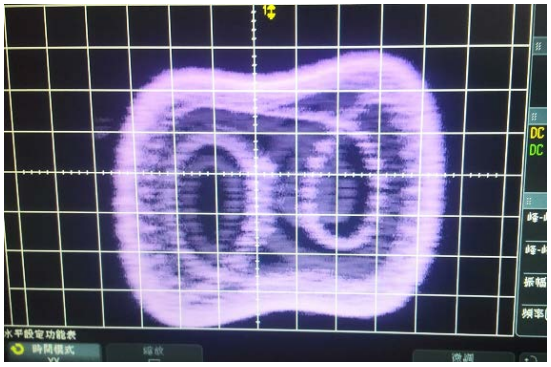

Fig. 6 FPGA realization of phase diagram

The final observed Duffing chaotic oscillator output is shown in Figs 5 and 6. In Fig. 5, the yellow signal is X-channel time-domain signal of Duffing chaotic oscillator and the green signal is Y-channel time-domain signal of Duffing chaotic oscillator. As show in Fig. 6, the phase diagram consists of a collection of fine discrete points. It can be seen from Figs. 5 and 6 that the successful digital hardware implements the Duffing chaotic oscillator.

Resource consumption as shown:

Device	EP4CGX150DF27I7
Timing Models	Final
Total logic elements	8,708 / 149,760 (6 %)

Fig. 7 FPGA resource consumption

It can be seen from the figure that the overall occupancy of FPGA resources less, with a very strong practicality.

3. Conclusions

In this paper, we propose the Duffing chaotic oscillator based on the fourth-order Runge-Kutta algorithm and achieve high-precision output of the chaotic oscillator. Use Verilog HDL hardware description language and describe Duffing chaotic oscillator directly, instead of using DSP Builder tools of this kind to convert matlab code directly into Verilog language, which makes the Duffing chaotic oscillator with easy to call, modify, and maintenance features. In addition, the use of fixed-point decimal conversion to floating-point decimal way substantially reduces the amount of logic resources. This kind of FPGA digital hardware realization solves the problem that the Duffing chaotic oscillator cannot realize the strict matching in the analog circuit because of the error of the component parameters and the aging problem, and consumes less FPGA resources. It provides an effective way for Duffing chaotic oscillator to realize the digital hardware of signal detection and communication, and has strong practicability.

4. Acknowledgments

This research has been supported by International Science and Technology Cooperation Program of China (2014 DFR10240), Hei Long Jiang Postdoctoral Foundation (LBH-Z14066), Science Foundation of Heilongjiang Province QC2015075, and Fundamental Research Funds for the Central Universities GK2080260146.

5. References

[1] Chert G, Lii J. Dynamics of the Lorenz System Family: Analysis, Control and Synchronization. Beijing: Science Press, 2003.
[2] Lü, Hu J, Yu S M. Theoretical Design and Circuit Realization of Complex Multi-Scroll Chaotic Systems[J]. Acta Mathematicae Applicatae Sinica, 2009, 32(6):997-1007.
[3] Zhou W J. Design and implementation of chaotic generators based on IEEE-754 standard and field programmable gate array technology[J]. Acta Physica Sinica -Chinese Edition-, 2008, 57(8):4738-4747.
[4] Koyuncu, I., Ozcerit, A.T., Pehlivan, I. Implementation of FPGA-based real time novel chaotic oscillator. Nonlinear Dyn. 77(1–2), 49–59 (2014).
[5] Tlelo-Cuautle,E.,Rangel-Magdaleno, J.J., Pano-Azucena,A.D.,Obeso-Rodelo, P.J., Nuñez-Perez, J.C. FPGA realization of multi-scroll chaotic oscillators. Commun. Nonlinear Sci. Numer. Simul. 27(3), 66–80 (2015).
[6] D. J. Evans, N. Yaacob. A fourth order runge-kutta method based on the heronian mean formula[J]. International Journal of Computer Mathematics, 1995, 58(58):103-115.