Journal of Information Hiding and Multimedia Signal Processing ©2016 ISSN 2073-4212
Ubiquitous International Volume **7**, Number **1**, January **2016**

Chaotic Signal Generator Design Based on Discrete System

Baishan Zhao and Chengyu Qi

College of Information Science and Engineering
Shenyang University of Technology
111 Shen Liao West Road,Shenyang,110870, Liaoning,China
zhaobaishan@sut.edu.cn

Received October, 2014; revised September, 2015

Abstract. *A novel design method of chaotic signal generator based on digital signal processing for three-dimension system is presented in this paper. We illustrate the structure and function of each module in this system. By comparing the experimental results with the simulation results, the feasibility of this method is verified. The proposed chaotic signal generator can offer a new alternative in random testing system or in secured data communication applications.*

Keywords: Chaotic signal, Module design, Discrete system, Discrete difference equations

1. Introduction.

Chaotic signals have been used in many fields for the characteristics of pseudo random and noise [1, 2, 3, 4]. Chaotic signals are generated by the differential dynamic systems [5, 6, 7]. Because of the effect of component parameters which are used to establish the dynamic analog systems, it is difficult to achieve the desired chaotic signals. With the development of the modern digital signal processor, the speed of numerical calculation is increasing. It is possible to simulate the movement of differential dynamic system by using discrete system [8], and acquire the chaotic signals. By means of this method, the design process of the chaotic system is simplified, and the overall volume and total effective cost of the system can be reduced. Meanwhile, the reliability of the system is improved. If the simulation algorithm is designed properly, this method can generate all types of chaotic signals needed in different chaotic system. At the same time, it is easy to ascertain the parameters of chaotic model in digital system which is very important for chaotic synchronization. This paper describes a method of generating chaotic signal by digital system.

2. Digital Chaotic System Realization.

The structure of overall chaotic signal processing is shown in Fig.1.

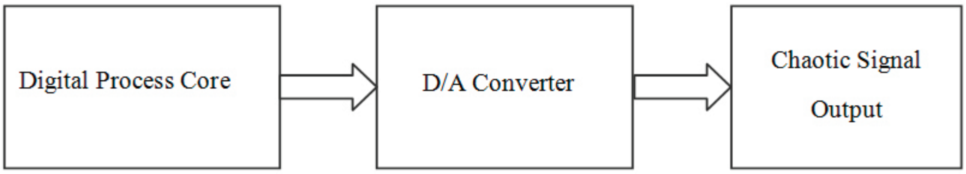


Figure 1. Chaotic signal processing flow

In this method, the differential dynamic equations are replaced by discrete difference equations in order to obtain the digital computing core. Take the Rossler three-dimension model as an example [4], the equations are listed as follows:

$$\begin{cases} \dot{u} = -v - w \\ \dot{v} = u + av + 0.1uln(w) \\ \dot{w} = c + w(u - b) \end{cases} \tag{1}$$

When The differential dynamic equations are transformed into discrete difference equations, the rule of equation discretization applies the forward difference quotient to replace the derivative, By solving the differential equation to calculate the numerical solutions of discrete nodes, the numerical solutions are not equal to the solution of differential equations or the formulas, while giving approximate values at each discrete node of the simulation process, this algorithm is more suitable for digital signal processing. The numerical solutions are calculated by using two order Runge-Kutta method shown in equations(2) which gives more exact solutions of the difference equations.

$$\begin{cases} y_{i+1} = y_i + h(K_1 + K_2)/2 \\ K_1 = f(x_i, y_i) \\ K_2 = f(x_i + h, y_i + hK_1) \end{cases} \tag{2}$$

So the new discrete difference equations are shown as follows:

$$\begin{cases} u_{k+1} = u_k + (-v_k - w_k)(h + 0.5h^2) \\ v_{k+1} = v_k + Kv + h/2(u_k + Kv)ln(w_k + Kv) \\ w_{k+1} = w_k + Kw + h/2[c + ((w_k + Kw)((u_k + Kw - b)] \\ Kv = h/2[u_k + av_k + 1/10u_k lnw_k] \\ Kw = h/2[c + w_k([u_k - b)] \end{cases} \tag{3}$$

where u , v and w are three chaotic signals,a , b and c are constants, k is the number of iterations, h is step length. We can adjust the step length h to change the variation rate of discrete chaotic signal. The smaller of the step length, the more of discrete sampling points. The precision of simulation is higher, meanwhile, the costs are more iterative times and large amount of calculation.

The difference equations(3) are divided into different modules when operation. The chaotic differential equations are analyzed to find the required operation types in this design. The functional design for the digital logic circuit of each operation type need to be carried on, and perform the simulation for each module, to ensure that it reaches the expected design goal, finally to deal with the timing design of the digital circuit. Due to the use of the iterative solution of the differential equation algorithm, the input state variables of each step are influenced by the output state variables of the previous step. However the path complexities of three signals are not the same, and the calculation time delays of each module are not the same, we need to design a sequential circuit to control the calculation time and hold time of each step for the whole computing time. According to the equations (3), we can give the structure diagrams of arithmetic operations as shown in fig.2. Each diagram shows the calculation process of a variable in the difference equations (3). We can find out that three kinds of modules are required, such as adder module, multiplier module and logarithmic module. For the logarithmic operation is difficult to realize in digital processor, it is replaced with a look-up table structure in the design. Before constructing all the modules, the data width is fixed as 20 bit, where bit 20, '1' or '0' indicates that negative or positive, and bit 19 to 7 represent the integer part, the others represent the decimal part of a real number , then we can describe each module by using digital logic circuit.

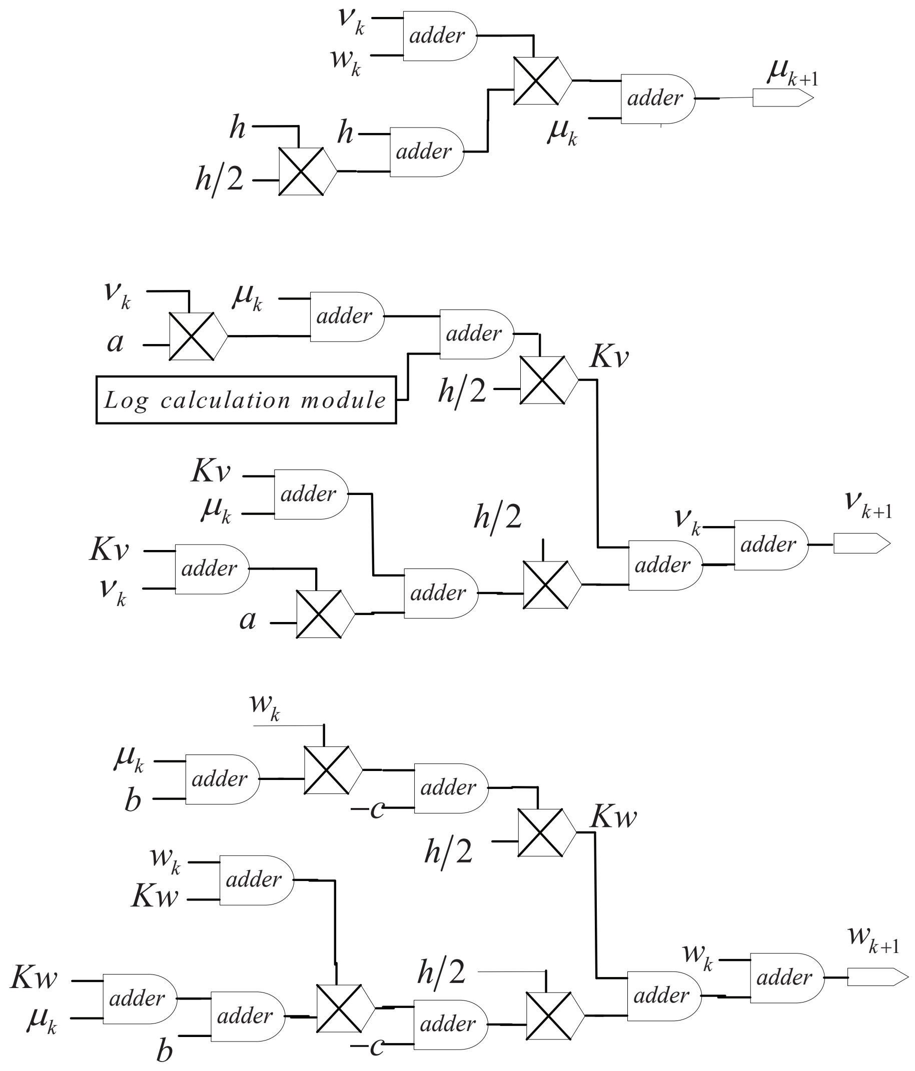


Figure 2. Electric circuit structure diagram

2.1. **Adder module design.** In this paper, the complementary binary code is used when designing the adder module. it is requirement to built a 20 bits adder. Here we use 1bit full adder to make the 20bits adder. Then the 20bits adder can be used in diagrams where signed as an adder. The structure of 20 bits adder is shown in Fig.3 which is in the form of digital logic format. The lowest effective bit begins to add two numbers correspondingly, and obtains the carry which is used as an input data for higher bit. Through this method, we can design the 20bits adder easily with hardware description language.

2.2. **Multiplier module design.** For the binary multiplication in this design the number is signed 20bits binary which has 12 bits represent decimal part of a real number, we can consider the number as an integer first, after operating the binary multiplication, the 40bits result is shifted 12 bit to the right, then the highest signed bit, and 19bits counted from lowest bit are combined into the data used as the input data of the 20bits adder. For multiplication, the logic circuit is quite complex, take the $5bits * 5bits$ as an example, let

$$Multiplicand\ M = m_4 m_3 m_2 m_1 m_0$$

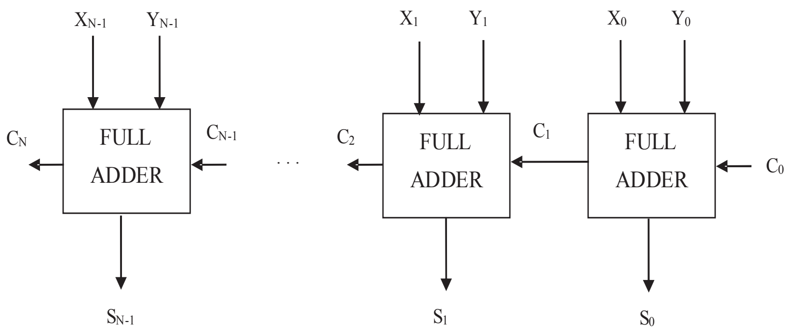


Figure 3. N bits full adder schematic diagram

$$Multiplier\ Q = q_4q_3q_2q_1q_0$$

Fig.4 shows the rule of the multiplication, in order to obtain right result of multiplication, the binary Multiplicand and Multiplier have to be added and signed bit and changed into complementary binary code. After operating the multiplication, the result is the correct data.

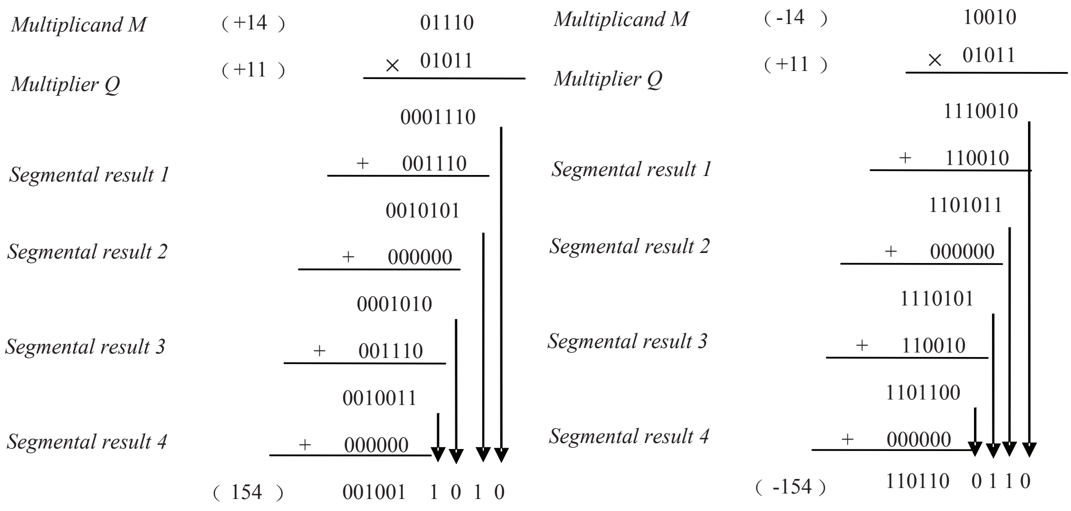


Figure 4. Signed multiplication

Based on the calculation rules of multiplication, we can construct a Mult_IP core shown in Fig.5. A(19:0) and B(19:0) are the input data of the ip core , Q (39:0) is the 40 bits output data. Meanwhile an enable input CE and a clock input CLK are added to the IP core.

2.3. **Look-up table structure design for the logarithmic module.** The logarithm operation is complex for digital signal processing. Usually we make a mapping for the variable and result. The mapping is saved in the read only memory (ROM) of the processor, so we can simplify the process of operation. In this design, we use a lookup method CASE statement in VHDL language to make the mapping. According to the variable of the Logarithm operation, the CASE expression can be acquired, and the result will be

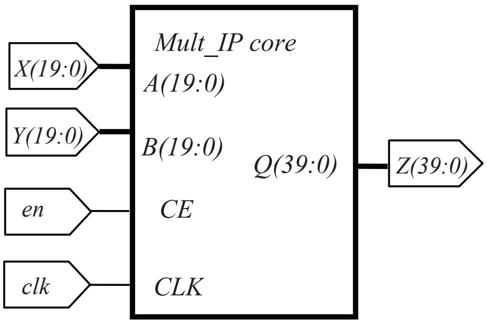


Figure 5. Schematic of Mult_IP core

found as soon as possible. The CASE statement is written as:
The CASE expression IS
WHEN conditional expression value = 1: sequential processing statement;
WHEN conditional expression value = 2: sequential processing statement;
. . . ;
WHEN conditional expression value = n: sequential processing a n statement;
WHEN OTHERS = ¿ sequential processing a n+1 statement;
Using look up table, the operation time can be greatly reduced.

2.4. **Sequential circuit design.** When using a lookup table, a decoding circuit is required, which belongs to the combinational logic circuit in digital circuits. For a combinational logic circuit, the output value is only related to the input values at the moment, regardless of the original state of the circuit. However, sequential logic circuit is different from the combinational logic circuit, which is another major digital circuit types. The major characteristic of sequential logic circuit is that the output status is not only related to the input signal, but also related to the circuit state before the moment. Because the system designed in this paper is the simulation of the continuous system in time domain, it is needed to synchronize the statements of$u$, v and w when the calculation steps into the next iterations. So the shift register and frequency divider need to be constructed.

A flip-flop can store 1 bit of data. If taking N flipflop as a whole, it can be used to store N bits of information. So we can construct the shift register by using a serial of flip-flop with a common clock. The shift register can be used for data maintenance and data conversion when the system need to synchronize the statements of$u$, v and w. A 4 bits shift register is shown in Fig.6 which is used for delaying the din data four clock cycles.

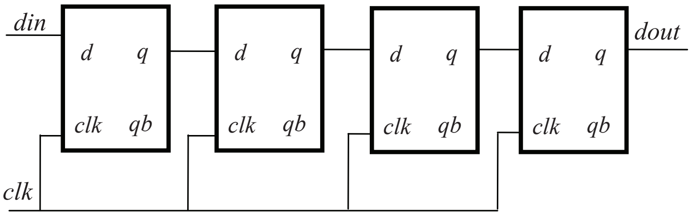


Figure 6. Shift register schematic diagram

In this design the system requires multiple clocks to the incentive for different modules, these clocks are obtained by master clock divided for the adder and multiplier modules. Two clock waveforms are shown in Fig.7 used for the adder and multiplier modules. Calculation delay of the adder is 1 main clock cycle, and calculation delay of the multiplier is 4 main clock cycles. While the operation period between two calculation intervals is 20 clock cycles, so we design the CE signals for the adder and multiplier respectively. The control signal of the adder is shown in Fig.7, as the 2-18 divider. The control signal of the multiplier is shown in Fig.8, as the 5-15 divider. Part of Sequential control signals of the system is shown in Fig.9. It can be seen from the figure that when the previous adding operation finished, the control signal en00 becomes low level, the control signal en1 of the next operation changes into a high level. The multiplier modules begin calculating and output the result when control signal turns to low level, taking the calculation process of forth to sixth channel signals as an example to analyze the process of the dada. When all of the stages have been performed, the calculation process has acquired a new set of values of the variables u, v and w, and begins another operation cycle at the next iteration. Then the sequences of the variable u could be produced when the operation goes round and round.

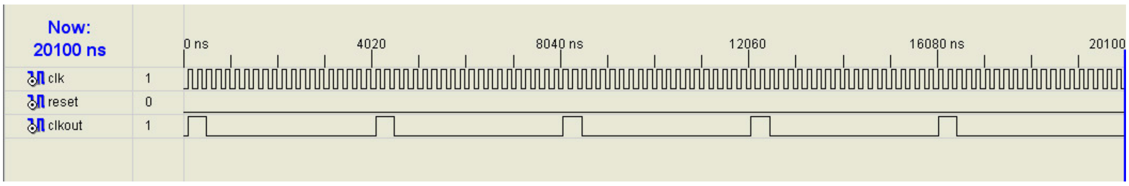


Figure 7. The control signal of the adder

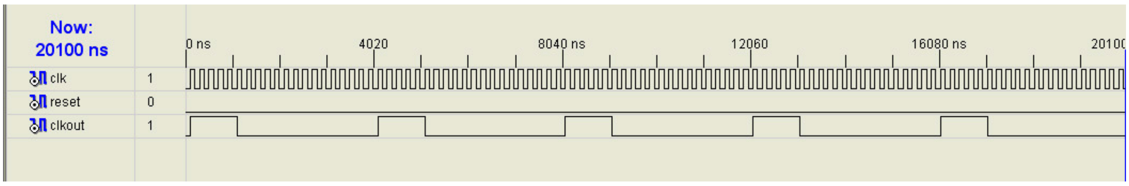


Figure 8. The control signal of the multiplier

3. **Synthesis and emulation.** After finishing the design of each module, the whole calculation process for variables u, v and $w$as shown in Figure 2 is designed to verify whether the design meets the design requirements by simulating the calculation process of the whole system. In the simulation analysis, we use VHDL hardware description language to realize the digital circuit, and then analyze the results in the timing simulation software. Firstly, the wave files in the ISE file is established, setting the main periodic clock cycle 40us.The output results are shown in Fig.10, followed by variables u, v and w. Comparing the time domain waveform of signal u , as shown in Fig.11, it can be concluded from the results that the chaotic signal waveform generated by the new system has the same characteristic with the signal produced by the software MATLAB. Because the lower data width is adopted in the hardware design language, the time domain signal has a certain error in comparison with MATLAB simulation.

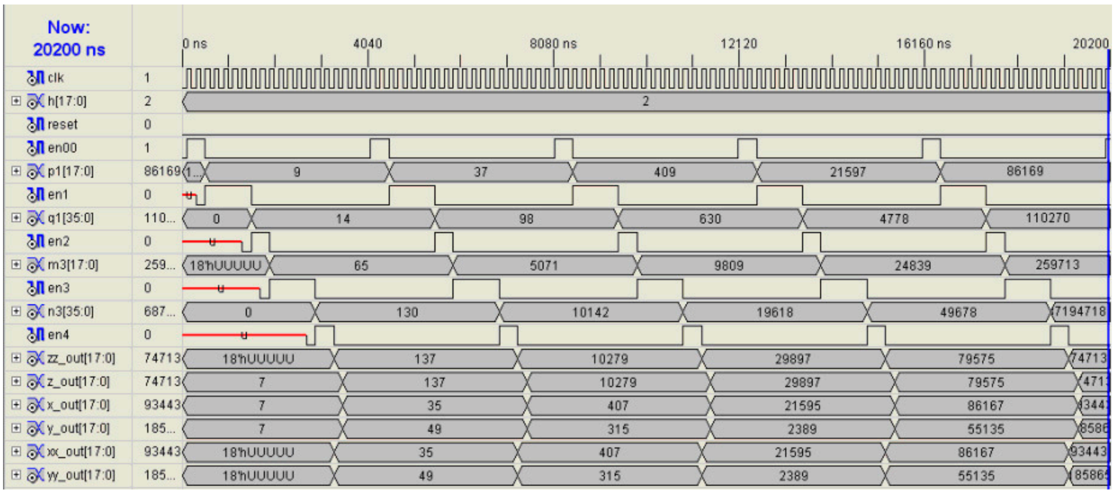


FIGURE 9. Sequential circuit control

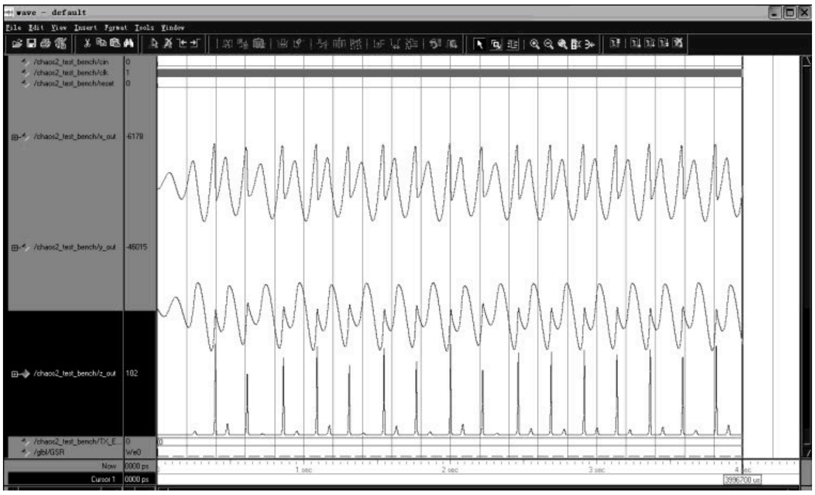
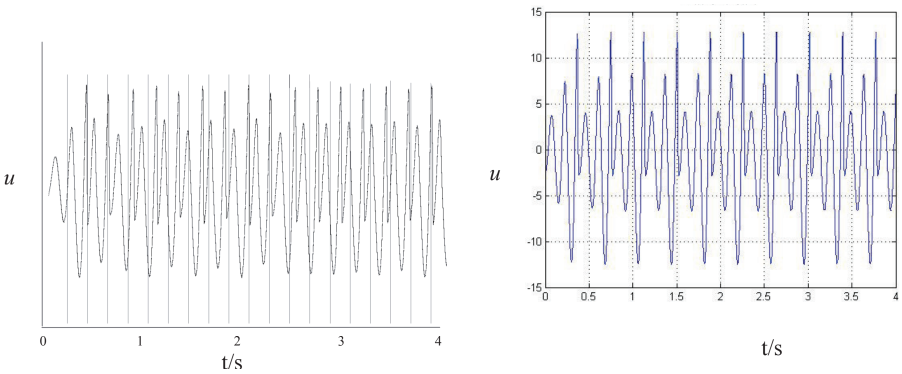


FIGURE 10. Hardware circuit simulation waveform diagram



a) Hardware simulation waveform

b)Software simulation waveform

FIGURE 11. Waveform comparison of hardware simulation and software simulation

4. **Test results.** The devices used in hardware system to construct the digital chaotic signal generator includes: Xilinx Spartan 3 XC3S400-F-Core FPGA development board shown in Fig.12 for the FPGA has more hardware multipliers. The schematic diagram of D/A converter expansion board shown in Fig.13.

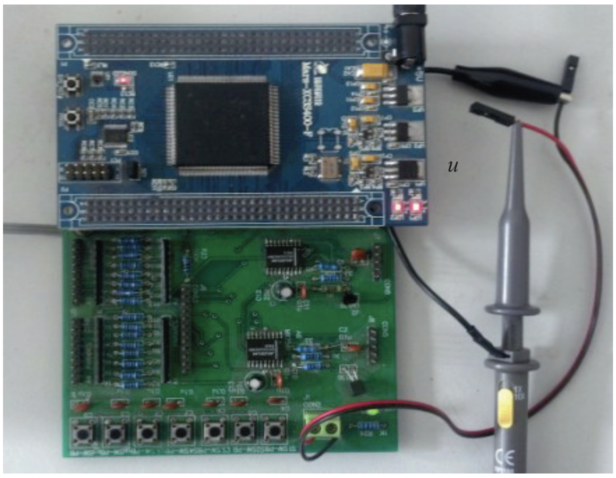


FIGURE 12. FPGA design platform

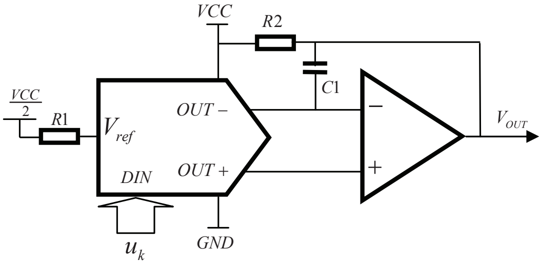


FIGURE 13. D/A converter circuit principle diagram

When we download the digital logic algorithm to the processor, the output signal of u can be test by the oscilloscope as shown in Fig.14. The plot of the signal has the same characteristic with software simulation.

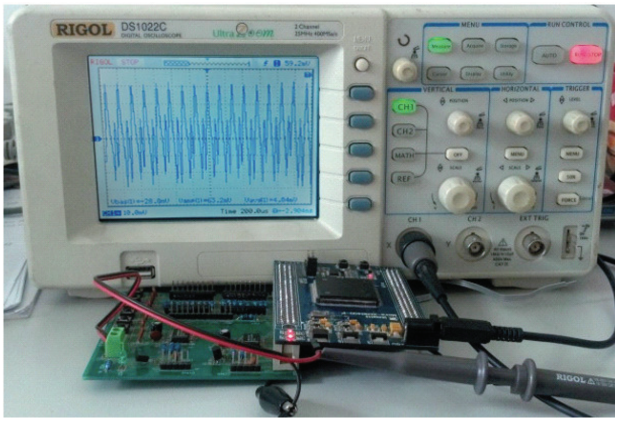


FIGURE 14. Chaotic signal generator and the waveform tested

5. **Conclusion.** We propose a DSP based approach to generate chaotic signal on the FPGA platform in this paper. The experimental results show that by the way of realization of hardware module, it can not only realize the generation of chaotic signal accurately and quickly, but also can be combined with the adjustment of parameters of chaotic model which may modify the spectrum characteristic of chaotic signal to satisfy the requirement of chaotic secure communication system flexibly.

Acknowledgment. This work is partially supported by Ministry of Education Key Project of Science and Technology of China (No.211037) and the general scientific research project of the Education Department of Liaoning Province, China (No.L2014053)

REFERENCES

[1] K. S. Halle, C. W. Wu and M. Itoh, Spread spectrum communication through Modulation of chaos, *Int J Bifurcation and Chaos*, vol.3, no.2, pp.469–477, 1993.

[2] M. Hasler, H .Dedien and M. P. Kennedy, Secure communication via Chua's circuit. *Proc. NOLTA 1993 workshops*, Hawaii, pp.87–92, 1993.

[3] V. K. Er, Controlling attractor structure in bistable chaotic oscillator, *Technical Physics Letters*, vol.37, no.6, pp.568–570, 2011.

[4] N. Farzbod. Chaos synchronization of multi time-delayed chaotic Rossler system with uncertainties and external disturbances via sliding adaptive fuzzy controller,*Proc.of Control, Instrumentation and Automation* , Shiraz, Iran, pp.355–360, 2011.

[5] K. G. Mishagin,V. V. Matrosov and V. D. Shalfeev, Generation of chaotic oscillations in the experimental scheme of two cascade-coupled phase systems,*Journal of Communications Technology and Electronics*, vol. 52, no. 10, pp.1146–1152, 2007.

[6] M. J. Xu, Y. Zhao and X. C. Han, Generalized asymptotic synchronization between Chen hyperchaotic system and Liu hyperchaotic system: A fuzzy modeling method, *Control and Decision Conference*, Guilin,China, pp. 361–366, 2009.

[7] C. K. Liu and F. Y.Sun, A new scheme to synchronization a fractional-orderhyperchaotic system and computer simulation. *Proc. of Computer Science and Automation Engineering (CSAE)*, Guangzhou,China, pp. 155–159, 2012.

[8] W. Chatchai and S. U. Wimol, Digital chaotic signal generator using robust chaos in compound sinusoidal maps,*IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E96-A(3), pp. 781–783, 2014.