Numerical Modelling of Chaotic System and Its FPGA Implementation

Subodh Kumar Pandey[1], Dr.Sanjeev Kumar Gupta[2]
[1,2]Dept. of ECE, Rabindranath Tagore University, Raisen (M.P.) India.

ABSTRACT
Now a days Chaos based systems play an important role specifically in secure communication and cryptography. Chaotic systems have wide applications in random number generators, image encryption, Optical secure circuits, and quantum applications. The FPGA implementation has certain advantages over analog one as FPGA implementation of any system is having a more flexible architecture and have low cost testing cycles, nowadays more emphasis is given to realize different chaotic systems in FPGA. This paper demonstrates the steps for FPGA implementation of a chaotic system using Euler's algorithm. Top level, second level and third level designs are also presented. The design is implemented using Verilog and tested with Xilinx vivado v.2017.3 design suite in Artix-7 Nexys 4 DDR. Simulation results presented demonstrates the timing diagram and resource utilization.

Key Words - Chaotic system, cryptography,Euler algorithm, FPGA

I INTRODUCTION

Due to random behavior of chaotic signal, the chaotic systems play an important role in cryptography and secure communication. In the 1990's it is identified that chaotic system can be synchronized, this fact leads to wide application of chaotic systems

The analog based design of chaotic systems is rigid in architecture and acquires a larger chip area. FPGA implementation has certain advantages over analog one, so more emphasis is given to realize different chaotic systems in FPGA. The chaotic generator is an integral part of any chaotic system, hence it is interesting to analyze the behavior and resource utilization of different chaotic generators when implemented in FPGA. Literature published in beginning demonstrated that the chaotic system is represented by the set of differential equations containing quadratic terms and constant parameter which decides the behavior of the system and systems are very sensitive to initial conditions. A very small change in initial conditions produces a very different kind of waveforms [1-7]. For FPGA implementation of any chaotic systems, numerical modelling is required. There are several numerical algorithms (Euler, Heun and RK4) are available. In this paper Numerical solution of the equations describing the Pandey-Baghel-Singh system [8] are obtained and demonstrated the procedure of FPGA implementation by developing the top level, second level and third level designs. This paper also presented the simulation result obtained using Xilinx Vivado 17.3 design suite on Nexys 4 DDR Artix-7 FPGA logic family.

II DIGITAL CHAOTIC GENERATORS AND ITS IMPLEMENTATION

In the recent years, chaotic systems were designed on digital platform. The basic building block of any chaotic system is chaotic generator. The chaos generators are realized as digital chaotic generators by finding out the numerical solution of the differential equations by which the chaotic generator is described. In the digital chaotic generator the system variables are realized using registers and by finding out the numerical solutions. In general the numerical solutions are implemented on digital platform as combinational blocks. The basic diagram of digital chaos generator in which numerical solution is represented by a combinational block is given in fig. 1

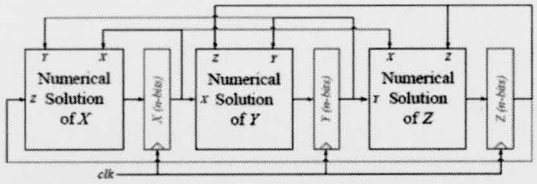

Fig.1 Basic diagram of digital chaos generator

For designing a digital chaos generator the selection of numerical technique is very important because the chaotic response of the system realized is highly depended on the numerical algorithm used.

III NUMERICAL ALGORITHMS

There are a number of methods available for the solution of differential equations. Some of them are only applicable for a limited class of differential equations. In general, for physical problem differential equations does not belongs to familiar type and we required to have numerical methods to solve differential equations. These numerical methods are becoming more useful when we realized differential equation using computers as they have reduced numerical work considerably.

There are a number of numerical methods available for solving the first order differential equationsof the type [9],

$$\dot{y} = f(x, y), given\ y(x_0) = y_0$$

These methods give the value of y in terms of power series of x through which we can calculate the value of y by direct substitution. The Picard and Taylor series methods belong to this class. In some methods solution of a differential equation is given in terms of a set of

values of x *and* y. The Euler, Heun and RK4 belong to this category in which we calculate the value of y in short steps for equal intervals of x and hence these methods are known as step by step methods. Nowadays, where computing time is largely reduced with the use of computers the numerical solutions of differential equations of these methods are more useful in engineering. The Euler, Heun and RK4 methods are used to find out the value of y over a limited range of x values.

In the Autonomous chaotic system the initial condition is defined and system is very sensitive to initial conditions. In the above equation the initial condition is defined at the point x_0. The problems in which the initial condition are defined are known as initial value problems, but the problems involving the second and higher order differential equation in which the values for more than one point are defined are known as boundary value problems. In the paper, we deal with the initial value problems for which several numerical algorithms are available like Euler, Heun and 4th degree Runge-Kutta (RK4) to get the numerical solution of the equations defining the chaotic system.

(a) Euler's Algorithm

This is a purely numerical method for solving the first order differential equation with initial conditions. Consider the equation $\dot{y} = f(x, y)$,with the initial condition $y(x_0) = y_0$ (1)

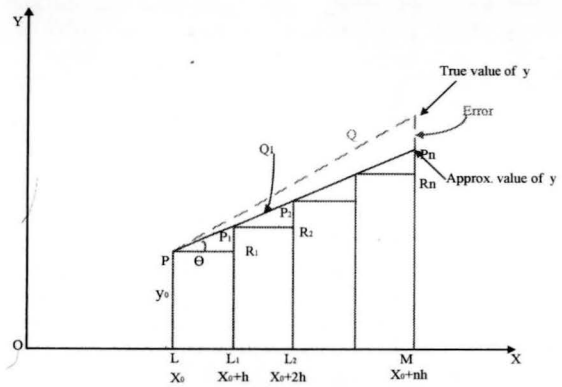

Fig. 2 Approximation using Euler method

As shown in fig 2 starting from the initial condition through $P\,(x_0, y_0)$ to find out value of y at another point Q the interval LM is divided into n subintervals with the step size of h. The step size should be small as possible otherwise the approximation error will be quite significant. For approximation in the interval LL_1 we approximate the tangent at point P is drawn which meets the ordinate through L_1 at point $P_1(x_0 + h, y_1)$ then

$y_1 = L_1P_1 = LP + R_1P_1$

$y_1 = y_0 + PR_1 \tan\theta$

$y_1 = y_0 + h\left(\frac{dy}{dx}\right)_P$

$y_1 = y_0 + hf(x_0, y_0)$

Which is the solution of the differential equation at point P_1 and let the tangent at P_1 meets the ordinate throgh L_2 at point $P_2(x_0 + 2h, y_2)$ then $y_2 = y_1 + hf(x_0 + h, y_1)$

If we repeat the process n times we reach to approximation MP_n of MQ given by

$$y_n = y_{n-1} + hf(x_0 + \overline{n-1}h, y_{n-1})$$

This is the Euler method for finding out the approximate solution of any differential equation as given in equation 1.

(b) Heun'sAlgorithm

In Euler's methods the curve of solution in the interval LL_1 is approximated by in $tangent\ at\ P$ so that we have

$$y_1 = y_0 + hf(x_0, y_0) \qquad (2)$$

Then $tangent\ at\ P_1$ is drawn find out the approximation

$$y_2 = y_1 + hf(x_0 + h, y_1) \quad (3)$$

In this way we got the solution of differential equation successively.

In a Heun method for better approximation the slope is taken as the mean of the slopes of the $tangent\ at\ P\ and\ P_1$ to get first approximate value as

$$y_1^{(1)} = y_0 + \frac{h}{2}[f(x_0, y_0) + f(x_0 + h, y_1)] \quad (4)$$

As the slope of the $tangent\ at\ P_1$ is not known we take y_1 as

$$y_1 = y_0 + hf(x_0, y_0)$$

Inserting the above value in eq. (4) we get the first modified value $y_1^{(1)}$. The eq. (2) is called the *predictor* and eq. (3) is *corrector* of y_1

To find out the better modified value $y_1^{(2)}$ corresponding to L_1 again the *corrector* may be applied and we get

$$y_1^{(2)} = y_0 + \frac{h}{2}[f(x_0, y_0) + f(x_0 + h, y_1^{(1)})]$$

We repeat this process till two consecutive value of y is approximately same to the acceptable limit after this point is taken as the starting point from next interval L_1L_2

Once y_1 is obtained to desired degree of accuracy y corresponding L_2 is found from the *predictor*.

$y_2 = y_1 + hf(x_0 + h, y_1)$

And a better approximation $y_1^{(2)}$ is obtained from the *corrector*

$$y_2^{(1)} = y_1 + \frac{h}{2}[f(x_0, h, y_1) + f(x_0 + 2h, y_2)]$$

We repeat this step until y_2 becomes stationary. Then we proceed to calculate y_3 as above and so on.The Heun's method produces the better approximations compare to Euler's method for same degree.

(c) Runge-Kutta (RK4) Algorithm

The Taylor's series method of solving for a differential equation is restricted as it requires finding the higher order derivatives. The Runge-Kutta method eliminate the problem of finding out the higher order derivatives. This method agrees with the Taylor's series solutions up to the term h^r where r called the order of the method. Euler method, Heun method and Runge method are the Runge-Kutta method of the first, second and third order respectively.

The fourth order Runge-Kutta (RK4) method is most commonly used and is often referred as Runge-Kutta method only.

In RK4 method the increment k of y corresponding to an increment h in x for the equation
$\dot{y} = f(x,y), given\ y(x_0) = y_0$
is as follows;
Calculate successively

$$k_1 = hf(x_0, y_0)$$
$$k_2 = hf(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}k_1)$$
$$k_3 = hf(x_0 + \frac{1}{2}h, y + \frac{1}{2}k_2)$$
$$k_4 = hf(x_0 + h, y_0 + k_3)$$

Finally compute $k = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$
Which gives the required approximate value $y_1 = y_0 + k$ where k is the weighted meanof $k_1, k_2, k_3\ and\ k_4$. One of the advantages of this method is that the operation is similar for linear as well as nonlinear equations.

IV FPGA IMPLEMENTATION OF CHAOTIC SYSTEM

For FPGA implementation first the chaotic system should be numerically modelled using any of Euler, Heun and RK 4 algorithms. Top level, second level and third level design can be developed which shows the functionality of the system based on FPGA. The numerical model is coded in Verilog and simulated using any suitable simulation tool. In this paper process is demonstrated using Pandey-Baghel-Singh Chaotic System and Euler algorithm. Numerical algorithm is implemented using Verilog and tested with Xilinx vivado design suite v.2017.3 in Nexys 4 DDR Artix-7 FPGA family.

The Pandey-Baghel-Singh chaotic system has four static variables, two equilibrium points and generates typical chaotic attractors. The system is simpler than other systems because it contains a single multiplier term.

The chaotic system is described by following three ordinary differential equations where x, y and z are the dynamic state variables.

$$\dot{x} = y$$
$$\dot{y} = z$$
$$\dot{z} = -a\,x - b\,y - c\,z - dx^2 \quad (5)$$

Where a, b, c and d are static parameters of the system. The system has six terms, including one quadratic nonlinearity term and four parameters.

For the numerical model using Euler algorithm initial value of x (n), y (n) and z (n) are taken as x (t_0) = x (n) = 0, y (t_0) = y (n) = 0 and z(t_0) = z(n) = 0.1, and the mathematical model of PBS chaotic system is described by the following Equation (6).

$$x(n+1) = x(n) + h.y(n)$$
$$y(n+1) = y(n) + h.z(n)$$
$$z(n+1) = z(n) + h.\{-a.x(n) - b.y(n) - c.z(n) - x(n)^2\} \quad (6)$$

The Top-level diagram using the Euler algorithms is shown in Fig.3. For the synchronization purpose one bit starts, reset and clock signal is used. A 32-bit input has been used and initial conditions are set in the beginning phase. The 32-bit signal are used as input parameter. There is three 32-bit output signals (Xn_out), (Yn_out) and (Zn_out) and ready signal is taken as one bit control signals for the proposed chaotic generator.

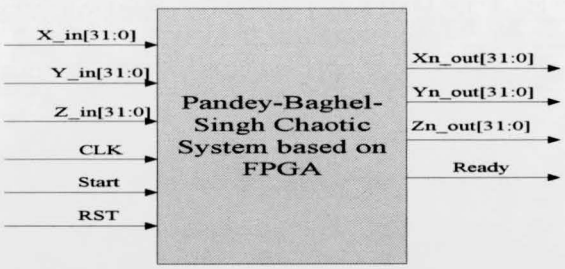

Fig.3 Top level diagram of Chaotic System based on FPGA

The second level block diagram of the chaotic generator is presented in Fig. 4 It has one multiplexer and a chaotic generator unit whichis FPGA based. The multiplexer is used to provide initial condition signals. For successive operation, it is provided by the output signals. When enable is at logic high, the output generates chaotic signal.

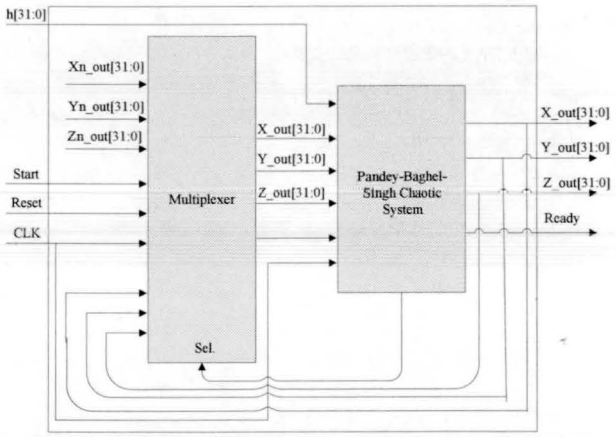

Fig.4 second level design of PBSCS based on FPGA

The third level block diagram of the Euler based chaotic generator is given in Fig.5 The system consists of multiplexer, function f, multiplier, adder/subtracter and filter. The system equations are calculated in the f unit and the output is multiplied by h in the multiplier. In the adder unit previously generated signals by the generators and the signal obtained from the multiplier are added. The filter unit eliminates the undesired signal. The system works sequentially and it generates the first value after the end of 42 clock cycles.

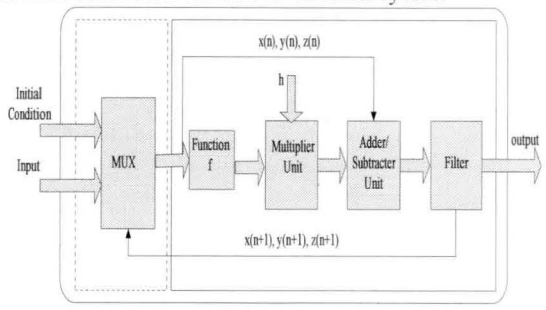

Fig. 5Third Level design of Euler based Chaotic System

IV SIMULATION RESULTS

The Euler algorithm based numerically modelled Chaotic generator have been synthesized on Nexys-4 DDR XC7A100TCSG-1 (Artix7) from the Xilinx vivado design suite v.2017.3.The simulation results include timing diagram, resource utilization, power utilization and other chip related Parameters and clock speed of the system. Fig 6. shows the timing diagram and fig.7 presents schematic diagram and other simulation results.

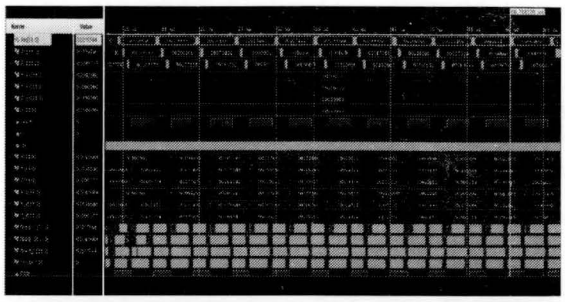

Fig. 6 Timing diagram of Euler based chaotic system

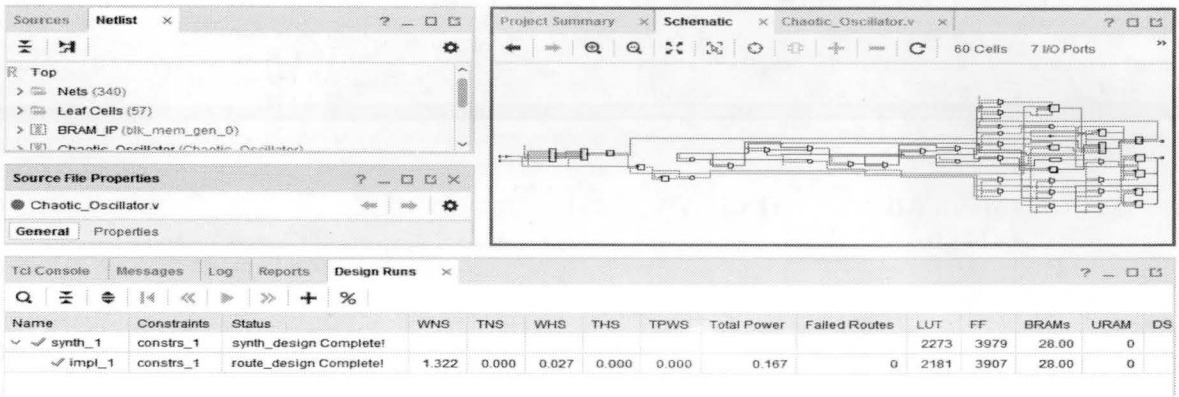

Fig.7 Schematic diagram and other simulation result of Euler based chaotic system

Table 1
Final report of the resources consumption

Parameter	Values
Maximum frequency (MHz)	359.71
No. of DSP	2
Number of 4 input LUTs	2181
Number of bonded IOBs	7
Number of Slice Flip Flops	3907
Total On-chip Power(W)	0.167
LUTRAM	71
BUFG	2
BRAM	28

V CONCLUSION

Any chaotic system which is described by ordinary differential equation can be numerically modelled using any one of numerical algorithms Euler, Heun and RK4.The performance of the system depends on the type of system and numerical algorithm selected. Numerical solution of the system is coded in Verilog and implemented in Nexys 4 DDR Artix 7 FPGA family in the environment of Xilinx Vivado design suite v.2017.3. The results show that any chaotic system can be numerically modelled and implemented in FPGA for secure communication and cryptography.

REFERENCES

[1] Ismail K., A. Tuaran O, IhsanPehlivan, "Implementation of FPGA-based real time novel chaotic oscillator", Nonlinear dynamics (2014) ; pp. 49-59.

[2] Murat Tunaa, Can BülentFidan, "Electronic circuit design, implementation and FPGA-basedrealization of a new 3D chaotic system with singleequilibrium point", Optic ElsevierOptik(2016) pp. 11786–11799.

[3] S. Banerjee, J. Kurths, "Chaos and cryptography: a new dimension in secure communications", Eur. Phys. J. Spec. Top. , (2014) pp. 1441–1445.

[4] I. Koyuncu, A.T. Ozcerit, I. Pehlivan, "An analog circuit design and FPGA-based implementation of the Burke-Shaw chaotic system", Optoelectron. Adv. Mater. Rapid Commun. (2013) pp. 635–638.

[5] L. Merah, A. Ali-pacha, N.H. Said, "A pseudo random number generator based on the chaotic system of Chua's circuit and its real time", FPGA Implementation (2013) pp. 2719–2734.

[6] S. C¸ ic¸ ek, A. Ferikog˘lu, I˙. Pehlivan, "A new 3D chaotic system: dynamical analysis, electronic circuit design, active control synchronization and chaotic masking communication application", Optik – Int. J. Light Electron Opt. (2016) pp. 4024–4030.

[7] M. Tuna, I. Koyuncu, C.B. Fidan, I. Pehlivan, "Real time implementation of a novel chaotic generator on FPGA", in: 2015 23rd Signal Processing andCommunications Applications Conference (SIU), IEEE, 2015, pp. 698–701.

[8] Alpana Pandey, R. K. Baghel, R.P. Singh, "Analysis and Circuit Realization of a NewAutonomous Chaotic System", International Journal of Electronics and Communication Engineering.ISSN 0974-2166 Volume 5, Number 4 (2012), pp. 487-495.

[9] B. S. Grewal, "Higher Engineering Mathematics", Khanna publishers, 39th edition, 2005.