Performance Analysis and Implementation of Lorenz keygenerator

Thinesh R
PG Student, Department of CS,
Hindusthan College of Engineering & Technology, Coimbatore,India

Dr.P.K.Poonguzhali
Professor, Department of ECE,
Hindusthan College of Engineering & Technology, Coimbatore, India

***Abstract*—This work presents an approach for Implementation of Lorenz keygenerator. Chaos is one of the most significant topics in nonlinear science, and has been intensively studying since the Lorenz system was introduced. Investigation of nonlinear dynamics and chaos in electronic systems has advanced tremendously in recent years. A Lorenz system is a celebrated nonlinear dynamical dissipative system which was originally derived by Lorenz to study chaos in weather patterns. At first, the Lorenz chaotic oscillator model is constructed using MATLAB-Simulink model and later it is converted to the Xilinx System Generator model. The synchronization of the Lorenz oscillator is also done in this project. The Pecora-Carroll identical cascading synchronization method is adopted for achieving synchronization. Here the behavior of the response system depends on the behavior of the drive system. The Xilinx system generator technology is used for the conception of the Lorenz chaotic system and for generating the code.**

Keywords— Lorenz, MATLAB, Simulink, Xilinx SystemGenerator, Chaos.

I. INTRODUCTION

During the past decades, This technology allows the appearance of hardware that is as flexible as the programming paradigm in the realization of real-time applications. In the case of the implementation of a digital chaotic system, most approaches based on FPGA are designed using a non-optimal description embedded architecture by using automatic code generation [1-2].

Implement the chaotic behavior generators and the chaotic attractors associated with certain practical applications, many methods based on analogue circuits are used, such as switched capacitors or analogue Complementary Metal Oxide Semiconductor (CMOS) technology.

There are two types of approaches when using chaotic dynamics in cryptography. The first one used as key streams to mask the plaintext in a manifold of ways. The nextone is used as initial state and the cipher text follows from theorbit being generated

The Programmable devices are a class of general-purpose chips that can be configured for a wide variety of applications. They have capability of implementing the logic not only hundreds but also thousands of discrete devices.

The xilinx System Generator bridges the gap between conceptual architectural design and the actual implementation in a Xilinx.

The System Generator for Simulink, developed in partnership with The Math Works, Inc enables to develop high- performance DSP systems for Xilinx FPGAs using the popular MATLAB Simulink products from The MathWorks, Inc. As a plug-in to the Simulink modelling software, the Xilinx System Generator provides a bit-accurate model of FPGA circuits, and automatically generates a synthesizable Hardware Description Language (VHDL) code and a testbench. This VHDL design can then be synthesized for implementation in Xilinx.

II. LORENZ EQUATIONS SYSTEM

The Lorenz system, invented for Edward N. Lorenz, The Lorenz system is an example of a non-linear dynamic system these systems are corresponding to the long-term behavior of the Lorenz system. The Lorenz equation is based on the fundamental Navier-Stokes equation for fluid. Different types of chaotic synchronization exist.

They are complete or identical or conventional synchronization, generalized synchronization, phase synchronization, amplitude envelope synchronization etc. A consequence of extremely high sensitive dependence on initial conditions is the two identical but independently evolving chaotic systems can never synchronize with one another, that is the corresponding state variables of the two systems cannot evolve identically, as any infinitesimal deviations in the starting conditions or in the system specification can lead to exponentially diverging trajectories making synchroniation impossible.

Contrast this with the case of linear systems, where the evolution of two identical systems can be very naturally synchronized.

Chaotic system started with nearly the same initial conditions, having two chaotic system evolving in synchrony might appear quite surprising.

Chaos is one of the most significant topics in nonlinear science, and has been intensively studying since the Lorenz system was introduced. Investigation of nonlinear dynamics and chaos in electronic systems has advanced tremendously in recent years. A Lorenz system is a celebrated nonlinear dynamical dissipative system which was originally derived by Lorenz to study chaos in weather patterns.

The three equations that govern the Lorenz system following:

$$\frac{dx}{dt} = \sigma * (y - x) \qquad (1)$$

$$\frac{dy}{dt} = ((r * x) - (y) - (x * z)) \quad (2)$$

$$\frac{dz}{dt} = ((x * y) - (\beta * z)) \quad (3)$$

where, σ, β and r called the control parameters. All control parameters should be greater than zero (σ, r, β > 0), but usually σ = 10, β = 8/3 and r is varied. The system exhibits chaotic behaviour for r = 28. To resolving this Lorenz system.

III. Flow Diagram

At very first, the Lorenz chaotic oscillator model is designed using MATLAB code. Once code is successfully verified after that the model is constructed using MATLAB-Simulink model. A flowchart is a diagram which represents an algorithm or process, including a computer program. The goal is to show each step as a box of various kinds, and describe their order by connecting each box with arrows. While most experienced programmers don't bother with flow charts, they will be helpful in creating organized your code when you are a beginner. They are also pretty mandatory if you are trying to write code that will be used by other people, since they provide an easy way of allowing people to make sense of how you conceptualize your program.

The Xilinx System Generator bridges the gap between conceptual architectural design and the actual implementation in a Xilinx FPGA. Xilinx ISE design suite software is used to allows us to generate the FPGA's programming file. Fig 1 shows implementation flow diagram of this work.

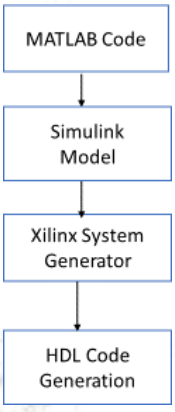

Fig 1: Flow diagram of this work

IV. Implementation of Lorenz System Using MATLAB Code

The generators are first represented by a set of nonlinear equations and a system-based model is developed to represent the equations directly. The Lorenz oscillator is a 3-dimensional dynamical system that exhibits chaotic flow, noted for its lemniscate shape. The map shows how the state of a dynamical system (the three variables of a three-dimensional system) evolves over time in a complex, non-repeating pattern.

unbreakable.

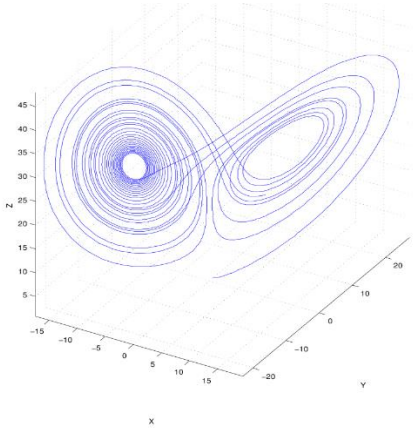

Fig 2: Lorenz equation model MATLAB code output

V. Implementation of Lorenz System Using MATLAB Simulink

Once MATLAB code is successfully verified after that the model is constructed using MATLAB-Simulink model. The Lorenz attractor, named for Edward N. Lorenz, is an example of a non-linear dynamic system corresponding to the long-term behavior of the Lorenz oscillator.The Lorenz oscillator is a 3-dimensional dynamical system that exhibits chaotic flow, noted for its lemniscate shape. The map shows how the state of a dynamical system (the three variables of a three-dimensional system) evolves over time in a complex, non-repeating pattern.

The products *xz* and *xy* that are performed by two multipliers.The equations are simulated using the Simulink block diagram presented by Fig 3. and Fig 4. The first parts relate to the phase plane (x, y) and (y, z) and final parts relate to thephase plane (x, z).

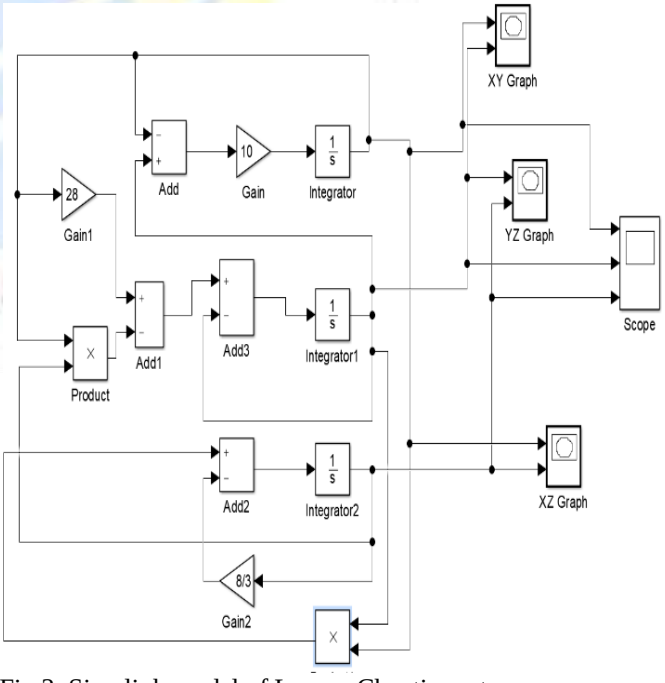

Fig 3: Simulink model of Lorenz Choatic system

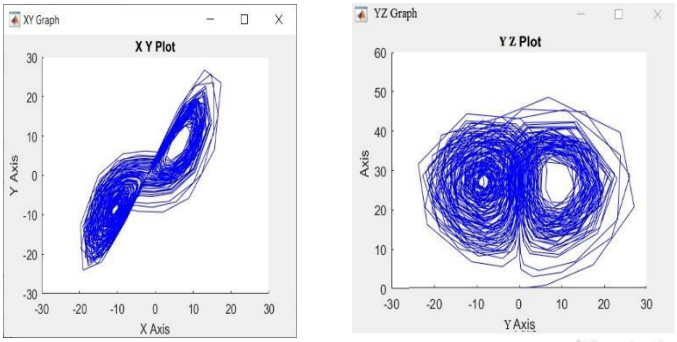

Fig 4: Space diagram of (XY) and (YZ) attractor.

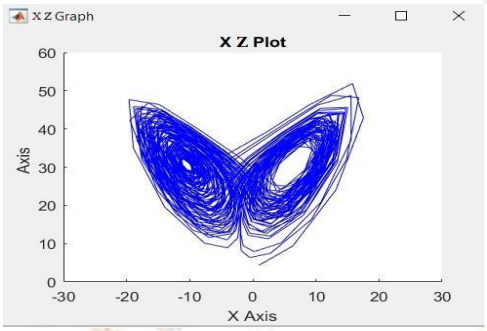

Fig 5: Space diagram of (ZX) attractor.

VI. IMPLEMENTATION OF LORENZ SYSTEM USING XILINXSYSTEM GENERATOR

Now the Simulink model was converted to Xilinx System Generator Model using Xilinx block set under the MATLAB. The main problem was there is no integrator within the Xilinx System Generator toolbox, so the integrator block was converted to model as shown in Fig 6. Parameters σ, r, and b denote the Prandtl number, Rayleigh number, and a geometric factor, respectively (Weisstein, 2002). Well-known parameter values for Lorenz system (48) showing chaotic behaviors are used for numerical simulations: σ = 3, r = 28, and b = 8/3.This circuit uses32-bits words with the binary point position after the bit number 16. The parameter dt presented in the block dt represents the integration step. The initial state of the integrator is stored inside the block Register, in the field Initial Value.

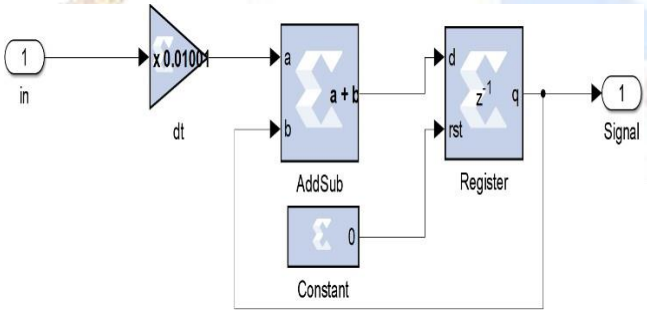

Fig 6: Integrator model using Xilinx system

generator The Lorenz system presented previously is now

implemented using Xilinx block set library's elements as can be seen in Fig 7 Three integrators blocks that can be seen in this structure are the same as already presented in Fig 6. From the reference paper [6] the author chooses three different value for different integrators.

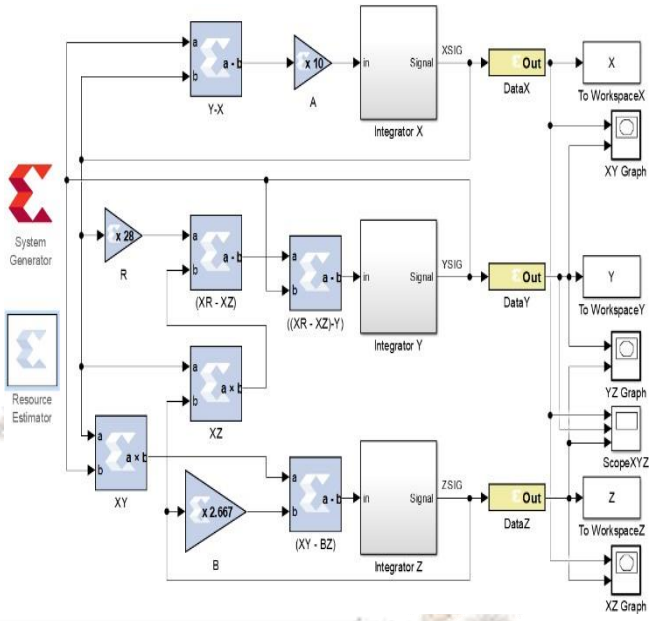

Fig 7: Lorenz chaotic generator using Xilinx System Generator blocks.

The all three Integrators has an initial condition equal to 10. The integration step dt used in this design is equal to 0.01. Many authors choose different initial values (x0, y0, z0) for getting perfect Lorenz output. The perfect output based on control parameters constant values.

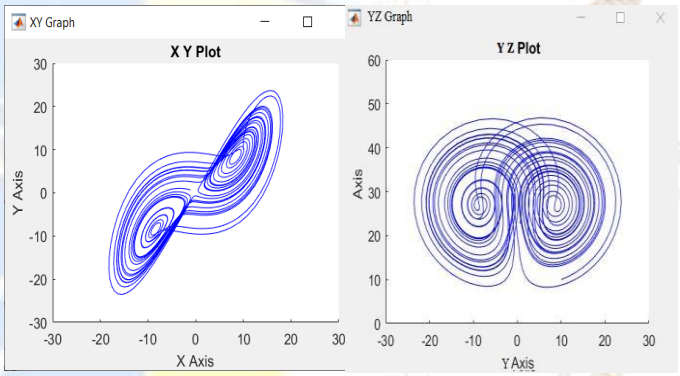

Fig 8: Space diagram of (XY) and (YZ) attractor.

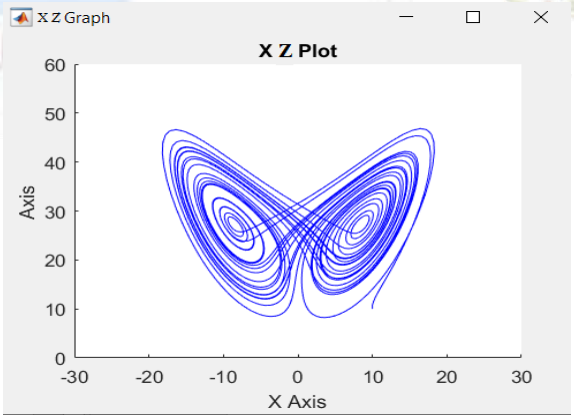

Fig 9: Space diagram of (ZX) attractor.

The Fig 8. The first parts relate to the phase plane (x, y) and second part is (y, z) and final part Fig 9 is related to the phase plane (x, z) of System generator output. Fig 10 shows Signals x, y and z generated by the Lorenz system created using Xilinx system generator. All the control parameter values in every cycle period in Fig 10. The figures below are obtained by fixing the following parameters, $\sigma = 10$, $\beta = 8/3$ and $r = 28$. The initial conditions $x0 = 10$, $y0 = 10$ and $z0 = 10$. It seems clearly that the different signals x, y and z have a random behaviour.

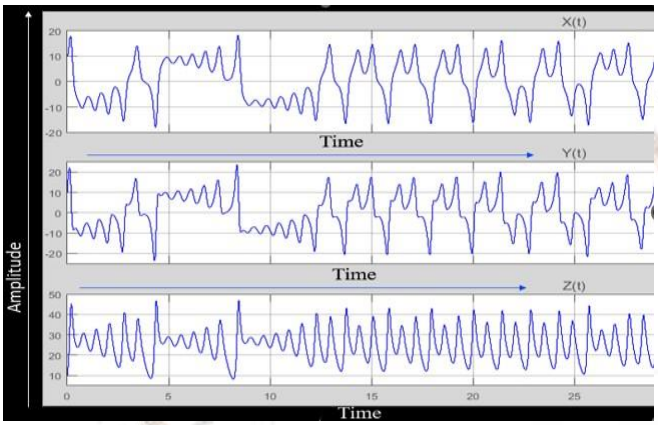

Fig 10 Signals x, y and z generated by the Lorenz system using system generator

VII. CONCLUSION

This work focuses on the Chaotic operation, Lorenz is one of the most significant topics in nonlinear science, and has been intensively studying since the Lorenz system was introduced. Investigation of nonlinear dynamics and chaos in electronic systems has advanced tremendously in recent years. A Lorenz system is a celebrated nonlinear dynamical dissipative system which was originally derived by Lorenz to study chaos in weather patterns. This implementation of a random key, based on Lorenz's chaotic generator for information security. The developed approach consists on using the implemented forth order Rung-Kutta method to resolve the differential equations system of the Lorenz chaotic generator. Lorenz chaotic oscillator model was designed and simulated using MATLAB code, MATLAB-Simulink and Xilinx System Generator. This design approach can be extended for FPGA implementation of other chaotic generator designs. The implementation of the proposed architecture allows a very useful and attractive trade-off between high speed, low area cost and data security transmission for an information security system.

REFERENCES

[1] Blaine C "A Concise Introduction for FPGA Design",pp.821-824,Feb.2011.
[2] Chen Henson tine,"EP32 RISC Processor IP: Description and Implementation Into FPGA". Appl. Math. Sci. 7(5), 237–246 (2020).
[3] Blaine C "A Concise Introduction for FPGA Design", Vol.51, No.8, Aug 2014.
[4] L. Zhang, "System generator model-based fpga design optimization and hardware co- simulation for lorenz chaotic generator", 2017 2nd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS 2017), pp. 170-174, June 2017.
[5] Robert zeidman, "Chaos Communications-Principles, Schemes and Systems analysis'', Proc. of the IEEE Inst. for Fundamentals of Electr.Eng. & Electron.,Dresden Univ. of Technol, 90. 2002. pp. 691-710.
[6] B N, Aryalekshmi, "FPGA Implementation of Lorenz's ChaoticGenerator", Universal review, vol viii, March 2019.
[7] Sadoudi S, Azzaz M.S., Djeddou, M., Benssalah, M. "An FPGAreal-time implementation of the Chen's chaotic system for securing chaotic communicate ons" International Journal, Nonlinear Science. 7, 467–474 (2009).
[8] IEEE Standard for Floating-Point Arithmetic, ANSI/IEEE Standard 754-2008, New York: IEEE, Inc., Aug. 29, 2008.
[9] IEEE standard for binary-floating point arithmetic, ANSI/IEEEStd 754-1985, The Institute of Electrical and Electronic Engineers Inc., New York, August 1985.
[10] M. Aseeri, M. I. Sobhy, and P. Lee "Lorenz chaotic model using field programmable gate array (fpga)," in The Midwest.
[11] M.Lakshmanan, S.Rajasekar, "Nonlinear Dynamics Integrability, Chaos and Patterns " 3rd edition Springer International Edition.