CHAPTER-1

INTRODUCTION

1.1 Need for Security

The present era of data explosion involves the necessity of high efficiency in terms of data capacity and data security [1]. As the data collection and processing capacity of the world as a whole is increasing at an exponential pace, there is a huge demand for attaining very high standards in terms of data throughput, processing capabilities and security, all at the same time [2, 3]. There is always a trade-off between security and capacity. However, if data capacity increases, high security is required to protect the data which imposes a Herculean demand on the signal processing, storage and communication equipment [4]. The most prominent beneficiaries in this era of Big Data are social networking sites and internet enabled services such as banking where privacy and security in data transmission is most crucial. Even though there are good protection measures currently, the data storage and transmission capacity offered by such systems is dismal in Big Data terms [5].

The typical examples that show there is a great need for security of big data are shown below:

- Gmail account hacking on September 11, 2014. Google, a very big multinational company with high security protection measures for data privacy, has succumbed to the hacking menace.
- The Snappening —a leak of approximately 90,000 photos and 9,000 videos stolen off the mobile app Snapchat. A UK Local Newspaper on October 13, 2014 confirmed that despite rumors of a hoax, the leak is genuine, and most of the affected users hail from Europe, which makes up 32 percent of its overall audience, according to Snapchat.

- Cars had been breached by a different group of scientists in 2011 through Bluetooth, GPRS and even car's media player CD is burned with a mischievous audio file. Self-driving cars will ascertain a tantalizing target for hackers if introduced in market, according to a top security executive.

1.2 Literature Survey of Secure Communication

Secure communication is a way of sharing information between two entities without any third party listening in. The key types of security are as follows [3]:

1. **The nature of a communication is hidden**

   Here we do not know the information content.

   Ex: Code, Encryption, Steganography etc.

2. **The parties to a communication are hidden – precluding identification, promoting anonymity**

   Here we don't know exactly the information content and the entities involved. For Example

   (i) "Crowds" and related anonymous groups – it is difficult to identify the source of information from a crowd.

   (ii) Unknown communication devices – fake cellphones, Internet centers.

   (iii) Unknown proxies.

3. **The fact that a communication takes place is hidden**.

   Here we do not know whether the communication has taken place or not. For Example

   (i) "Security by vagueness" – alike to needle in a haystack.

   (ii) Random traffic – forming indiscriminate data flow to make the occurrence of modest communication harder to identify and traffic investigation less reliable.

1.3 Role of Chaos in Secure Communication

The main reason to choose chaos in secure communications is because of its extreme sensitivity property [6]. The importance of chaos in secure communications has been explained by Chua and other people in the past and they have been successful in implementing it [6, 7].

The two aspects of chaos that are used as follows

1. Synchronization
2. Pseudorandom number

1.4 Objective

In this work we are trying to propose a novel kind of digital chaos which can be used in secure communications based on the above mentioned two aspects explained as follows [6, 7, 8].

1. **Synchronization**

   Most of the synchronized protocols are digital. In this work we used a simple circuitry to generate "Digital Chaos" from two square waves to be used as a futuristic clock/carrier. Following this we generate and characterize the digital chaos over various platforms like Application Specific Integrated Circuits (ASIC), Field Programmable Gate Arrays (FPGA), Microwind and MATLAB.

2. **Pseudorandom Number**

   We use a bit stream coming out of chaos which can be further used to generate a sequence of pseudorandom numbers. This sequence of pseudorandom numbers is validated using various statistical tests like Maurer's Universal Statistical Test, Discrete Fourier Transform Test, Non-Overlapping Template Matching Test, Runs Test, Rank Test and Monobit Test.

1.5 Project Flow

1. The basic principles of chaos and relavant analysis tools and techniques namely iterative map, cobweb plot, Largest Lyapunov Exponents (LLE), Fractal Dimension (D2), Kolmogorov entropy (K2), Phase Portrait and Recurrence Plot are used as standard parameters for characterization.
2. An iterative map representing digital chaos through the usage of square wave signals and XOR gate is formulated.
3. This iterative map is characterized using cobweb plots for different "r" values.
4. An ASIC based implementation of digital chaos generation is performed and standard characterization is done.
5. A layout level implementation of digital chaos is generated using 90nm CMOS technology is performed and standard characterization is done. The effect of wiring relating parasitics and associated delays on the nature of chaos generated is investigated.
6. In order to evaluate the tunability of the proposed digital chaos, an FPGA based implementation is carried out using Altera – DE1- Cyclone II FPGA.
7. Taking into account the recent development and prominence of software defined radio techniques, a purely software based implementation of digital chaos generation is done using MATLAB. The generated chaos and the nature of repetitions (or their absence) in the sequence is carried out using recurrence plots.
8. The MATLAB implementation of the chaos generator is used as the basis for Pseudo Random bit stream generation. The generated bit stream is tested for randomness using standard statistical tests like Histogram, Maurer's Universal Statistical Test, Non Overlapping Template Matching Test, Rank Test, Discrete Fourier Transform Test, Monobit Test and Runs Test.

CHAPTER-3

GENERATION AND CHARACTERIZATION OF DIGITAL CHAOS - ASIC

3.1 Introduction

In this chapter Digital Chaos is generated based on iterative map at Application Specific Integrated Circuit (ASIC) level in Hardware level using 7486 XOR IC and 556 Dual Timer IC and in layout level using Microwind Software based on the given iterative equation (3).

The Basic design principles as obtained from the equation are as follows:

1. Square waves are given as input signals.
2. The additive modulus function is implemented using basic logic operations such as XOR function (digital differential function). The below example will clear tells that the XOR operation is analogous to modulo operation.

XOR Truth Table

A	B	Y=A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Mod operation

$Mod_2(0,0)=0$
$Mod_2(0,1)=1$
$Mod_2(1,0)=1$
$Mod_2(1,1)=0$

The schematic diagram for digital chaos generation is

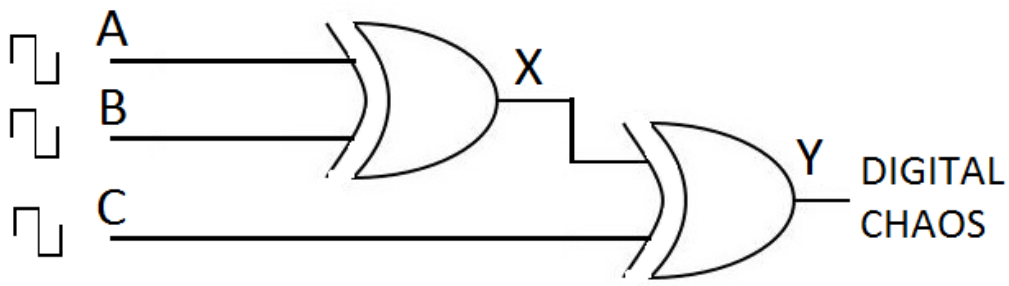


Fig. 3.1 Schematic Diagram for Digital Chaos generation

3.2 Implementation in Hardware level

Here the digital chaos is generated using XOR IC [22] with the square waves as input signals which are generated using 556 IC [21].

Square Waveforms generation using 556 Timer

The 556 IC used here is the Dual Timer from which we can get two square waves. The connection diagram of 556 is shown below.

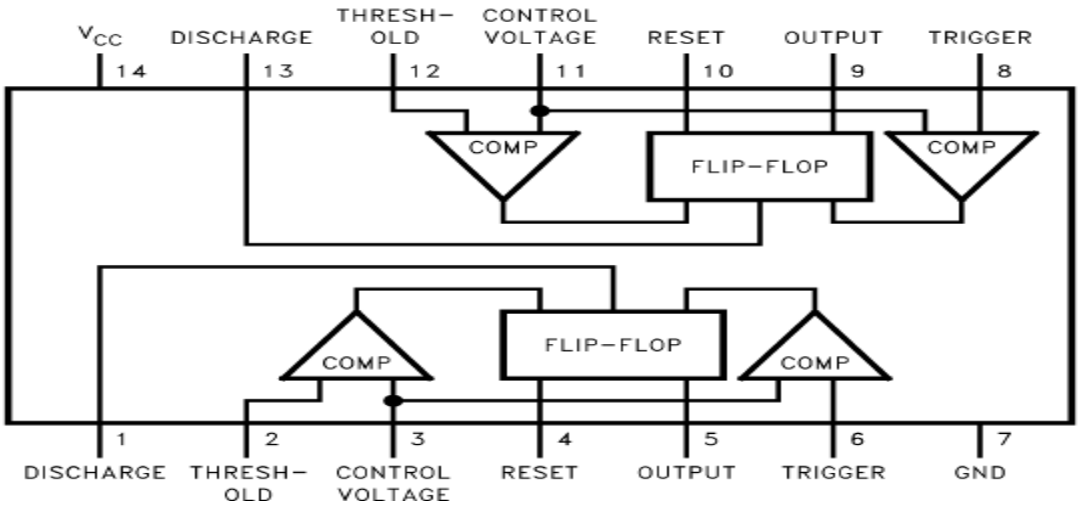


Fig. 3.2 Connection Diagram of 556 Dual Timer

The timer is operated in astable mode to generate square waveforms [21]. The formulae for square waveform frequency, low time and high time calculation are

$$f = \frac{1}{\ln(2).C.(R_1 + 2R_2)} \tag{9}$$

$$high = \ln(2).(R_1 + R_2).C \tag{10}$$

$$low = \ln(2).R_2.C \tag{11}$$

The circuit is as shown below

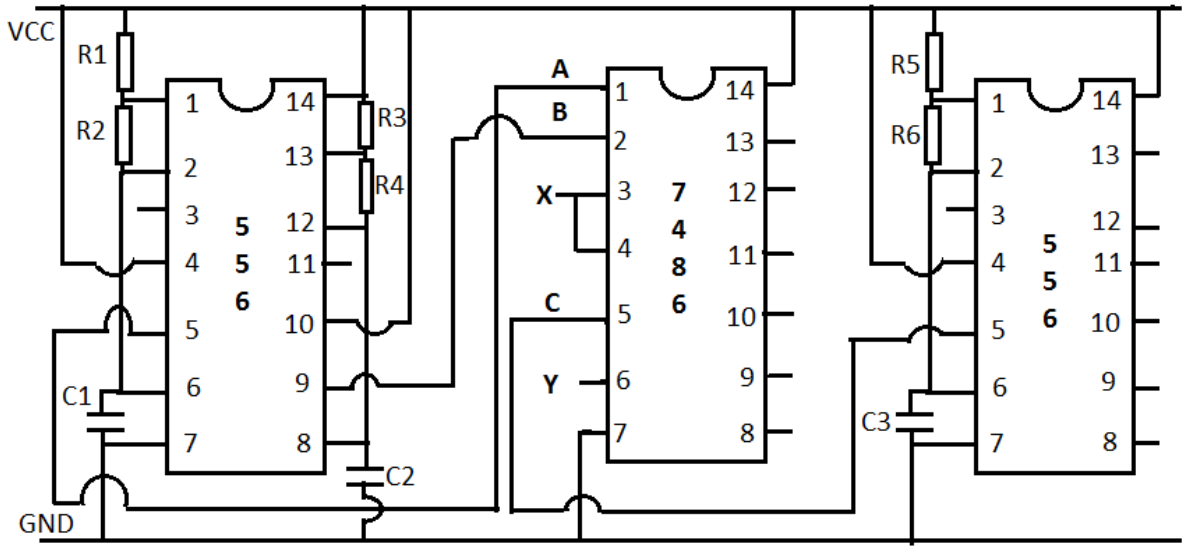


Fig. 3.3 Circuit Diagram for Digital Chaos Generation in Hardware

As shown in the above figure, the two square signals from Timer 1 are provided as inputs to first XOR gate in 7486 IC [22]. The output X is provided as first input to the second XOR gate and the square signal from Timer 2 is given as second input to the second XOR gate. The final output of the circuit is taken at Y.

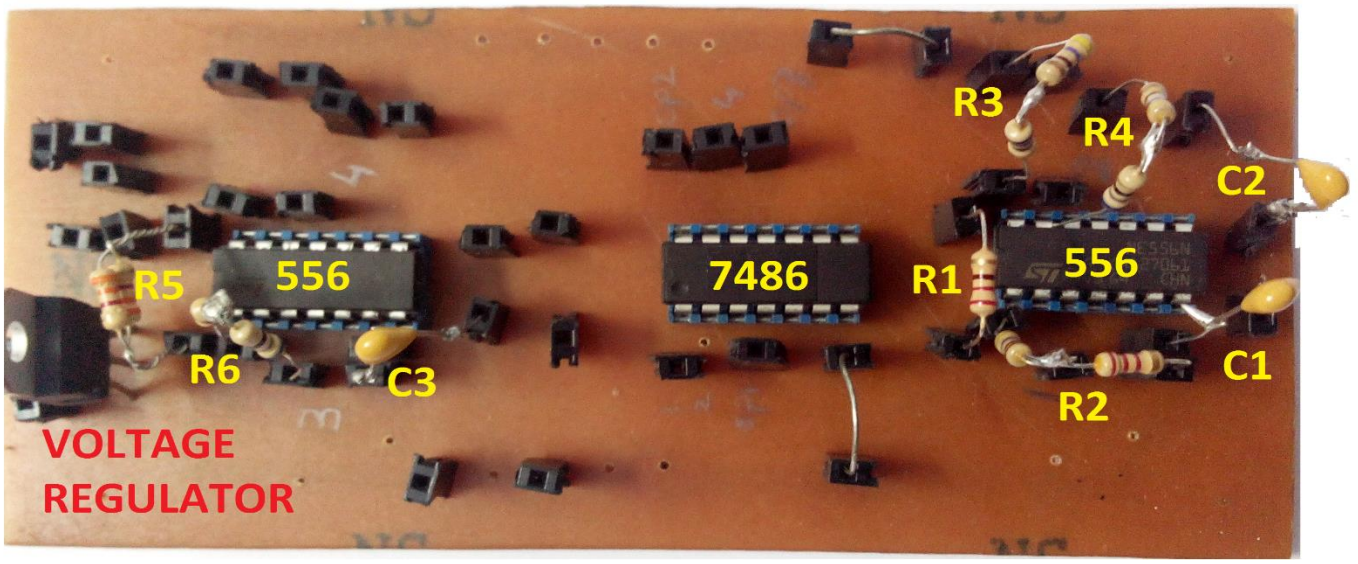


Fig.3.4 Printed Circuit Board design for Digital Chaos generation

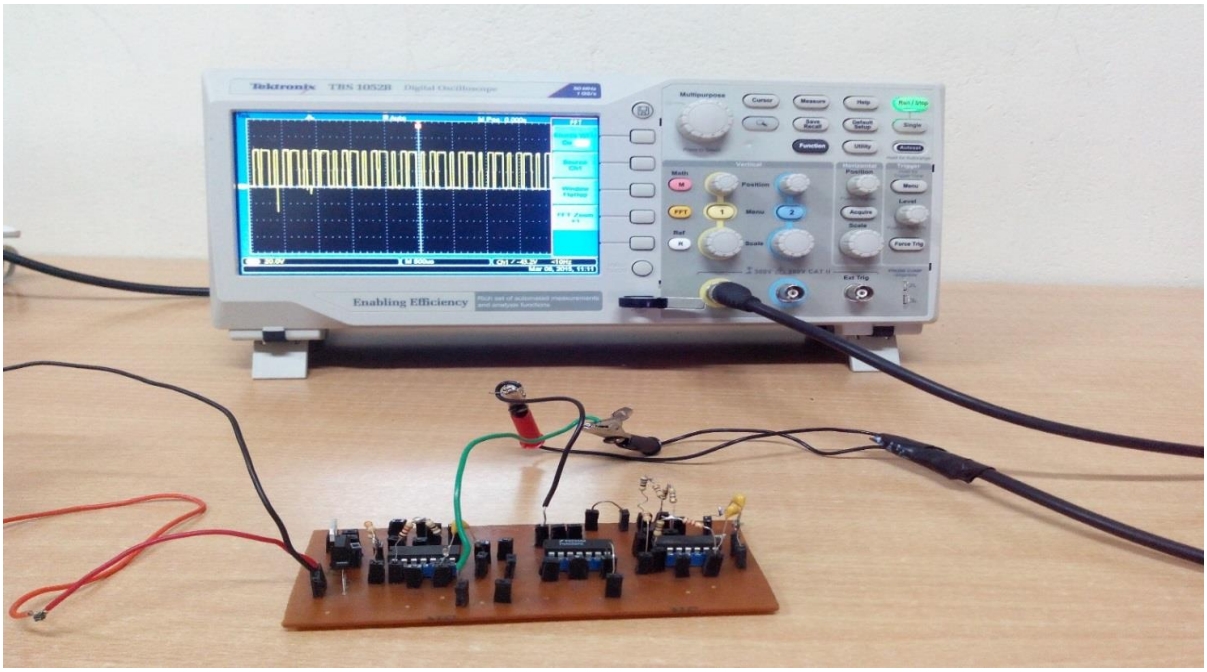


Fig.3.5 Experimental Setup for Digital Chaos in Hardware

After trial and error method, we found that for the Resistor values of R1= 220 Ω, R2=2.66 KΩ, R3= 533 Ω, R4=533 Ω, R5=1.606 KΩ, R6=1.07 KΩ and capacitors values of C1=C2=C3= 0.1μF possibility of chaos was observed. The Control Parameter for this case is obtained as r= 3.3. The Cobweb plot for this ratio is given in the Fig. 2.4.

The waveforms are as follows

Fig. 3.6 First input Square waveform

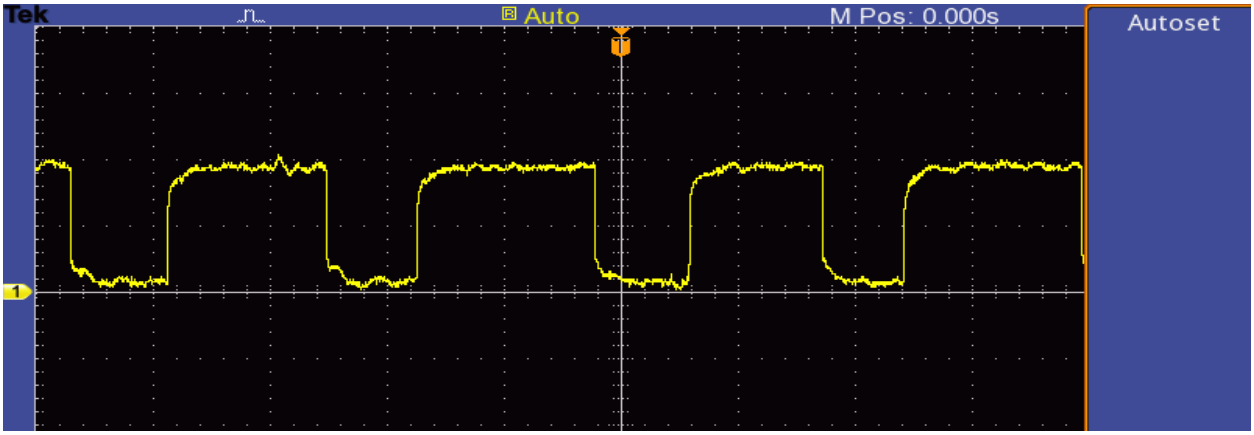


Fig. 3.7 Second input Square waveform

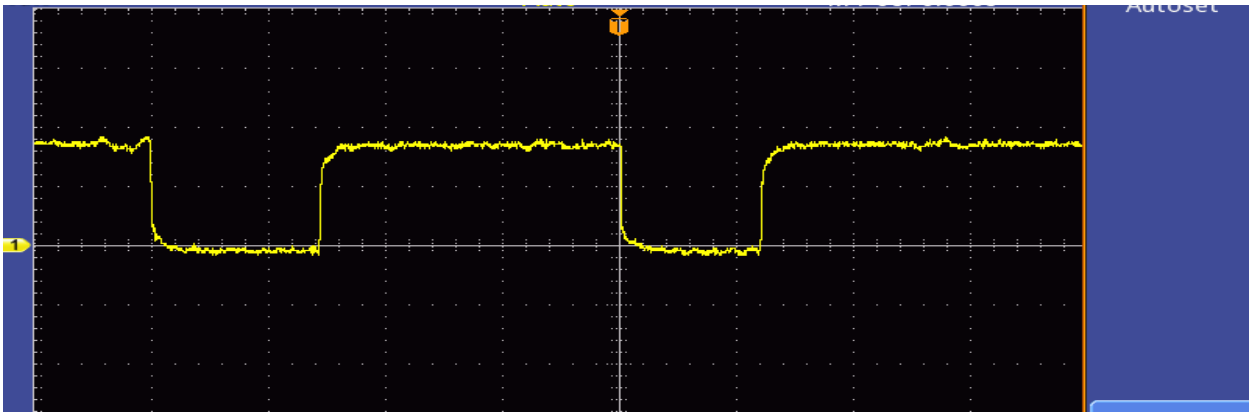


Fig. 3.8 Third input Square waveform

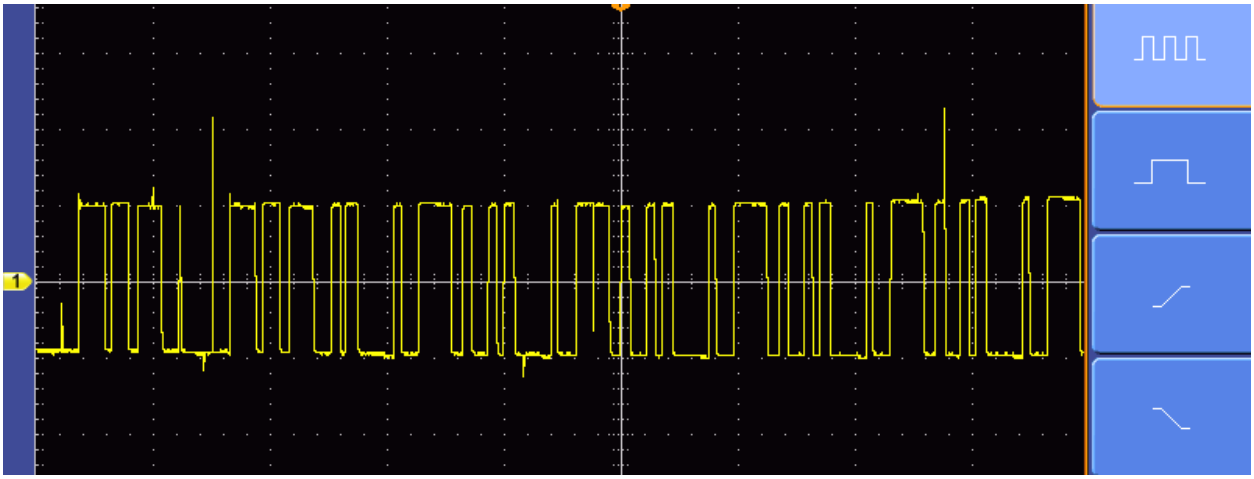


Fig. 3.9 Hardware level Digital Chaos output

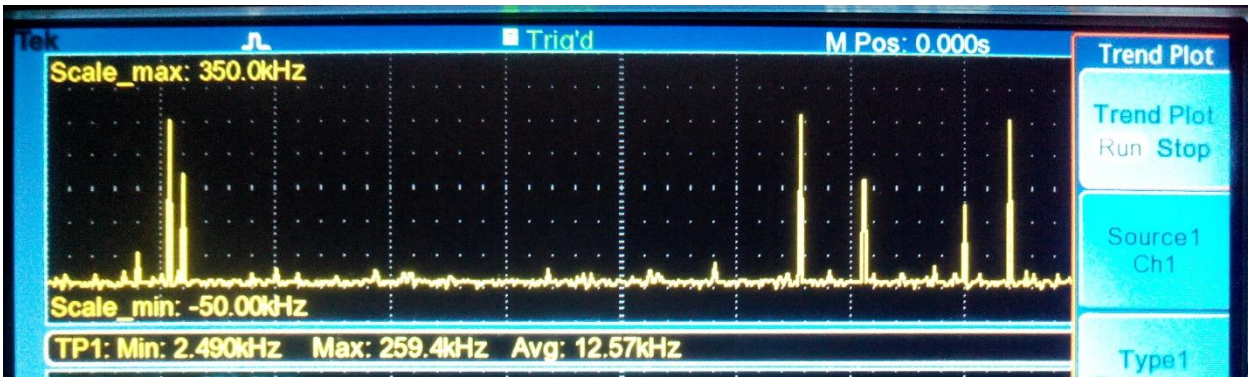


Fig. 3.10 Trend plot

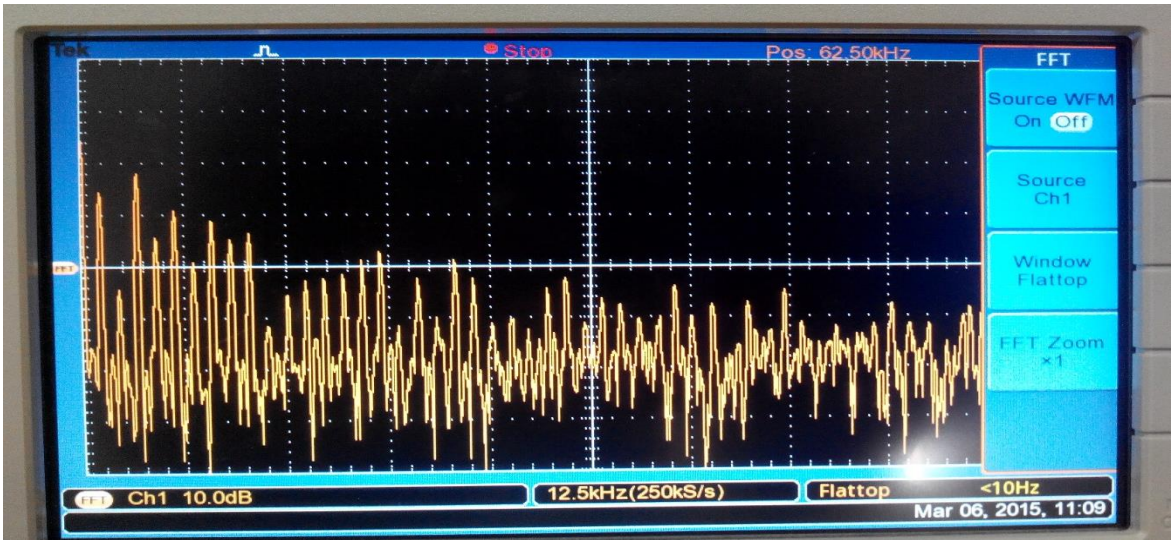


Fig. 3.11 FFT Spectrum of Digital Chaos - Hardware

The output chaotic data obtained from CRO has been characterized by calculating standard parameters in MATLAB.

Lyapunov Exponent (LLE) = 23.5865

Correlation Dimension (D2) = 0.661

Kolmogorov Entropy (K2) =7.2083 bits/symbol.

Finally the phase portrait and recurrence plot are as follows.

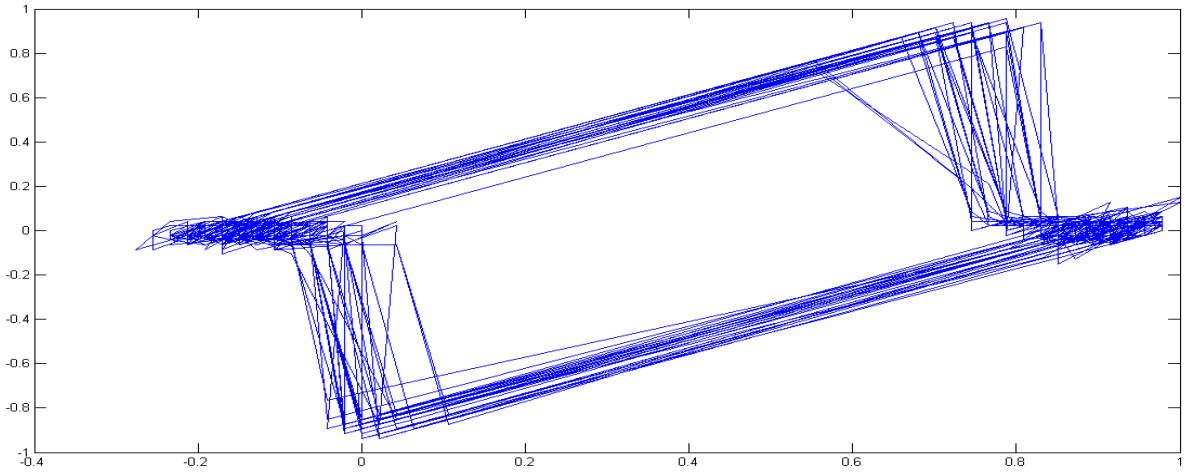


Fig. 3.12 Phase portrait of Digital Chaos - Hardware

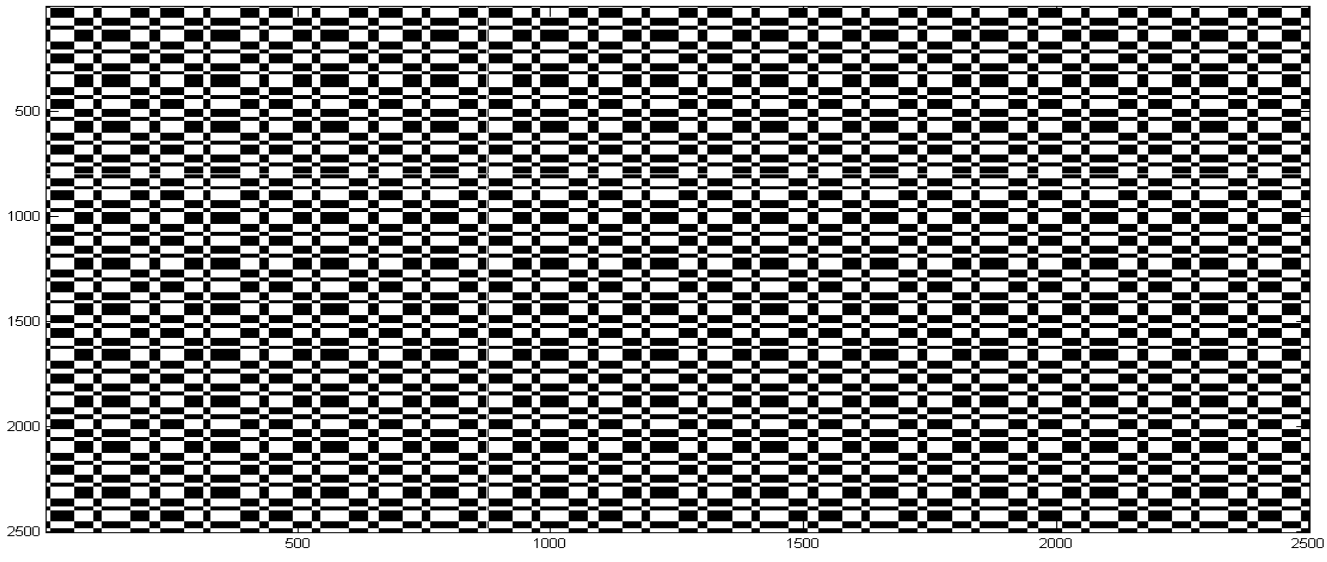


Fig. 3.13 Recurrence Plot for Digital Chaos - Hardware

3.3 Implementation in Layout level

Similar to the hardware case, here also two XOR gates have been used to generate Digital Chaos. The Square waveforms with same duty cycles as in Hardware level are given as inputs to the XOR gates.

The XOR is implemented in Microwind (90nm CMOS) using NAND gates [23]. The circuit for XOR using NAND gates is

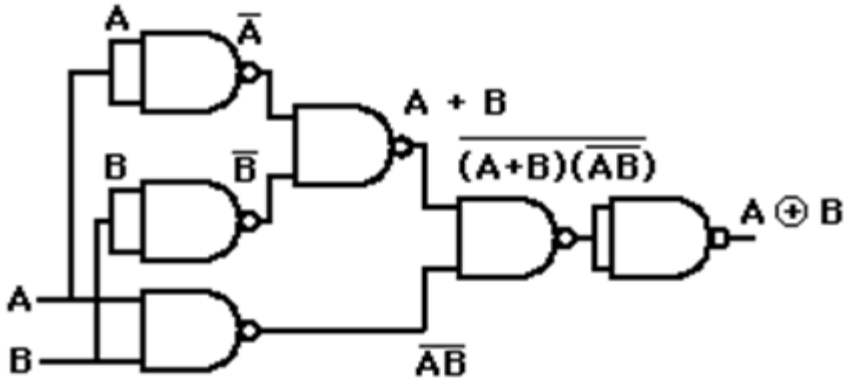


Fig. 3.14 Xor gate design using Nand gates

The Microwind Schematic for Digital Chaos Generation is

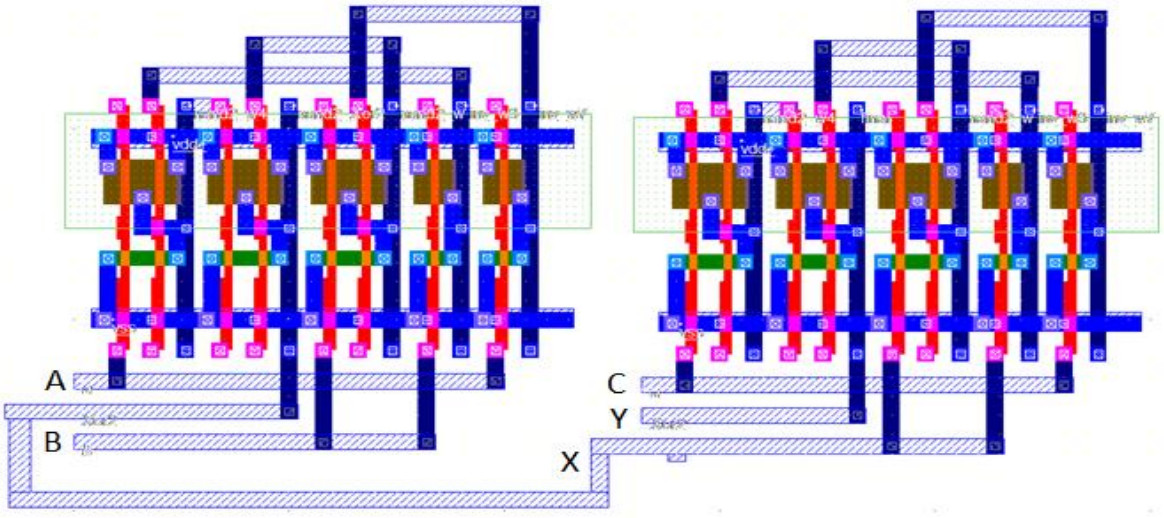


Fig. 3.15 Schematic diagram for Digital chaos - Microwind

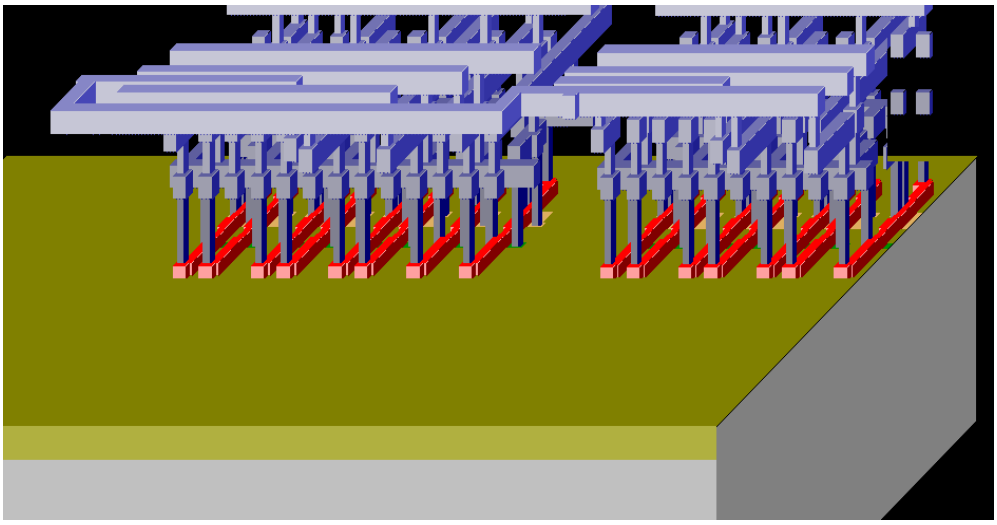


Fig. 3.16 3D view of the layout

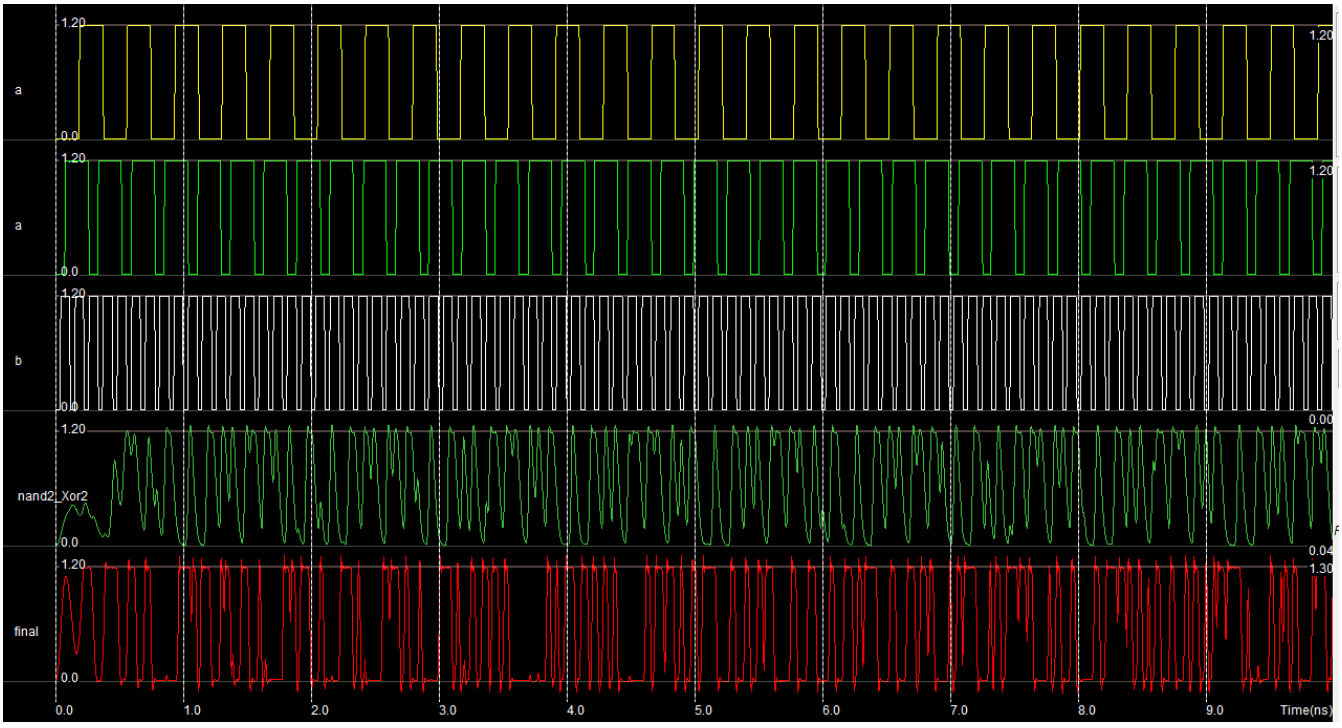


Fig. 3.17 Three Input Square Waveforms along With Stage 1 and Stage 2 Output Waveforms

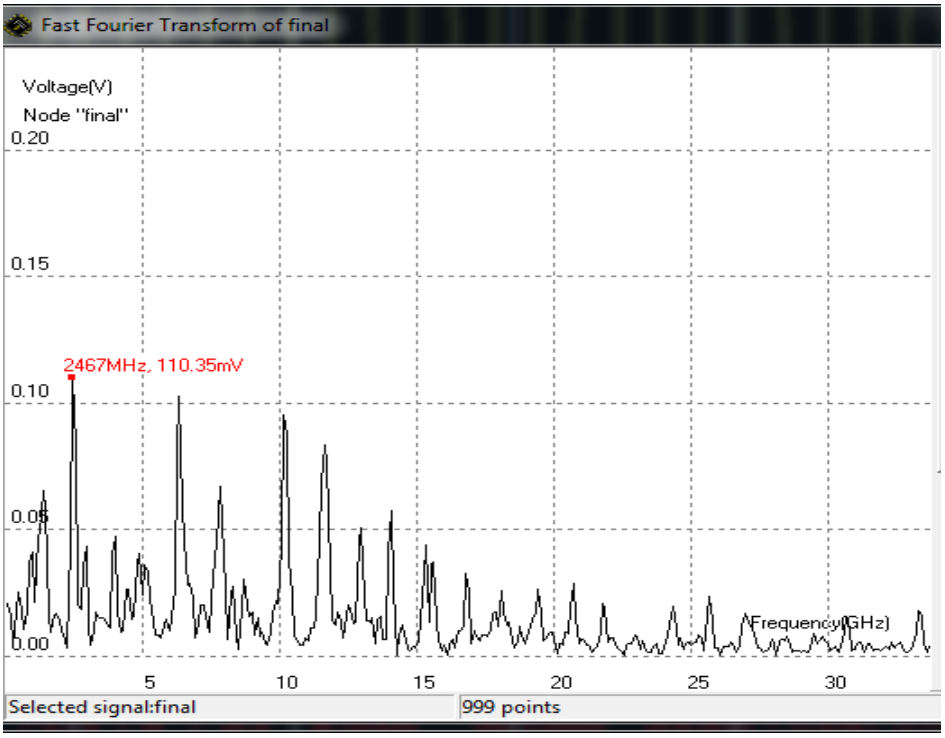


Fig. 3.18 Frequency Spectrum of Digital Chaos - Microwind

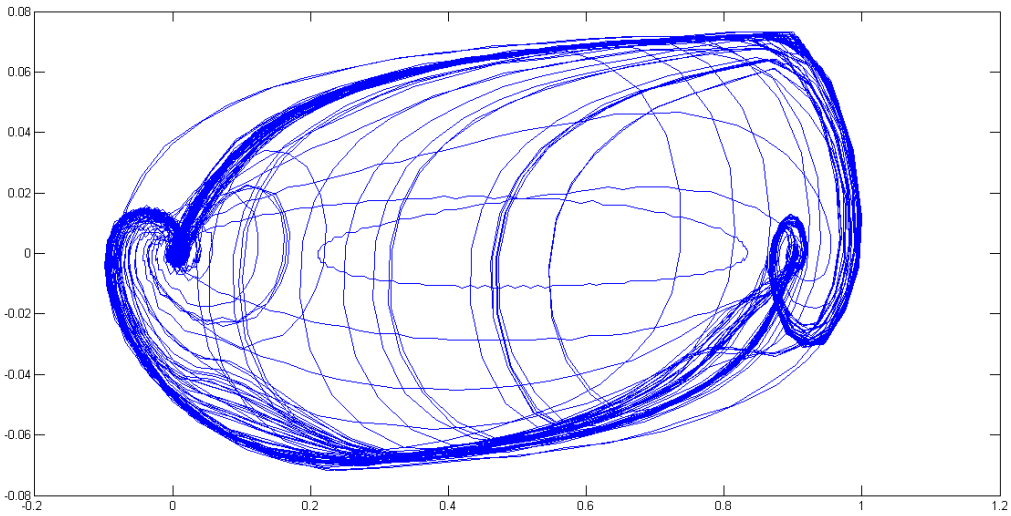


Fig. 3.19 Phase plot of Digital Chaos - Microwind

The generated digital chaos in Microwind has been characterized by calculating standard parameters in MATLAB.

Lyapunov Exponent (LLE) = 34.3618

Correlation Dimension (D2) = 0.661

Kolmogorov Entropy (K2) = 8.6606 bits/symbol.

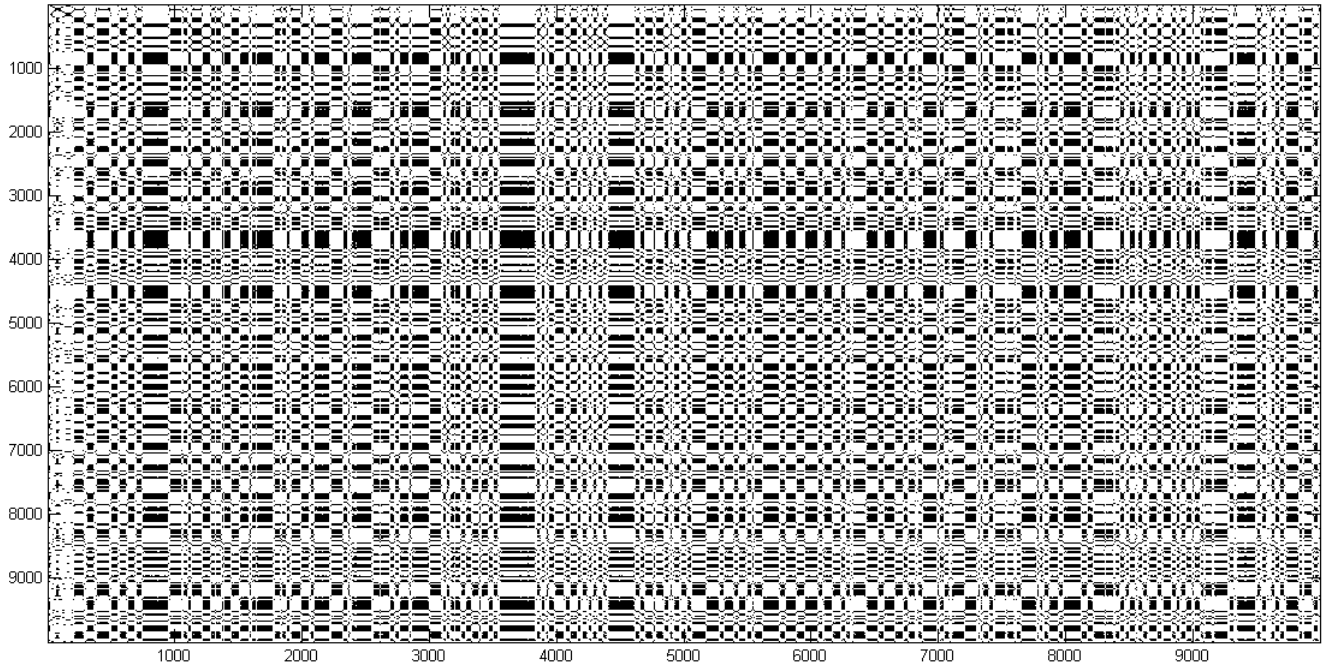


Fig. 3.20 Recurrence Plot of Digital Chaos - Microwind

3.4 Conclusion

Thus in this chapter we have generated digital chaos at the ASIC level. The effect of the parasitics in a typical 90nm CMOS Technology is assessed.

CHAPTER-4

GENERATION AND CHARACTERIZATION OF DIGITAL CHAOS - PROGRAMMABLE

4.1 Introduction

In the previous chapter the digital chaos had been generated at ASIC level. In this chapter the digital chaos generation is done at Programmable level. The basic idea is same in this case, except that here we can tune it easily for different ‘r’ value according to the requirement. Here we are going to generate the digital chaos in Field programmable Gate Array (FPGA) which is a programmable logic device (PLD). Similarly a case study is done in MATLAB characterizing the chaos in details with different parameters.

4.2 Implementation in FPGA

A Field-Programmable Gate Array (FPGA) is an integrated circuit intended to be configured by a customer after manufacturing [24]. The Hardware Description Language (HDL) is typically used to specify FPGA configuration. FPGA contains a ladder of reconfigurable interconnects and an array of programmable logic blocks, that permit the blocks to be "wired together" – similar to many logic gates that can be inter-wired in various configurations. Logic blocks can be constructed to perform complex combinational functions, or merely simple logic gates like AND and XOR gates. Logic blocks also comprise memory elements in some FPGA’s, which may be extra complete blocks of memory or simple flip-flops. [24].

The Altera’s Development and Education (DE1) Board which has Cyclone II FPGA is used to generate chaos.

Fig. 4.1 Altera DE1 Board

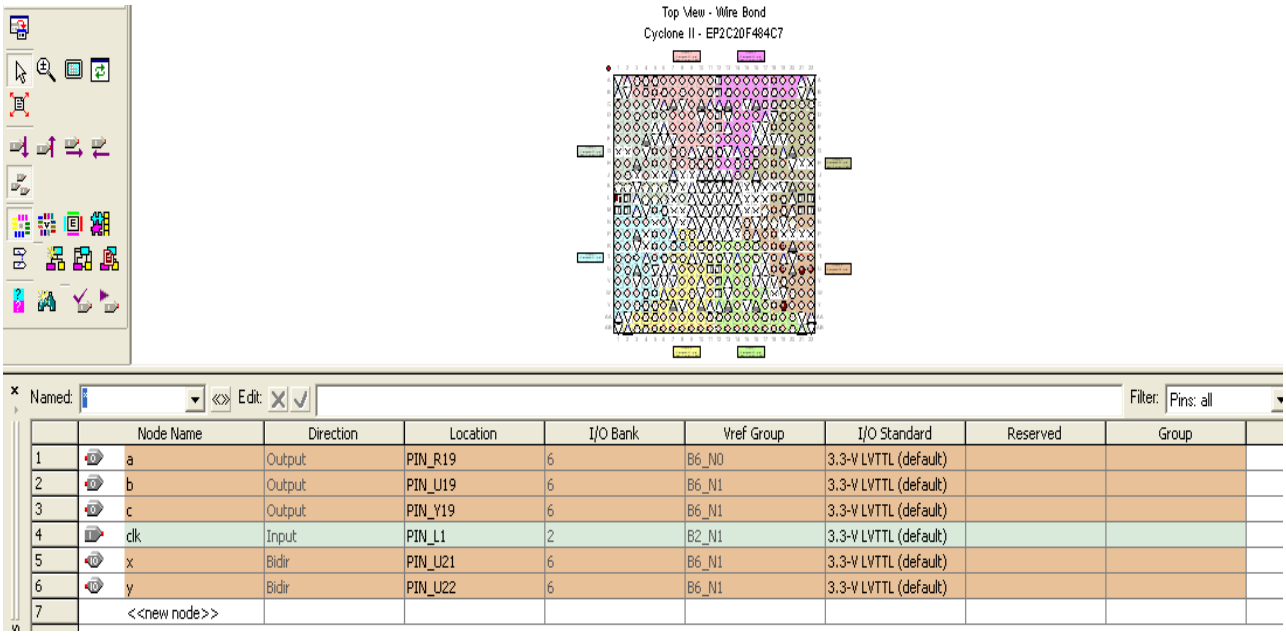
The main features of the DE1 board are

- Altera Cyclone II 2C20 FPGA device
- USB Blaster (on board) for programming and user API control; both JTAG and Active Serial (AS) programming modes are supported
- Altera Serial Configuration device – EPCS4
- Clock sources as 50-MHz oscillator, 27-MHz oscillator and 24-MHz oscillator.
- 8-Mbyte SDRAM
- 512-Kbyte SRAM
- 4-Mbyte Flash memory

Based on the Fig. 4.1, two XOR gates with three input Square waveforms are used to generate chaos in FPGA [24]. The three square signals are generated with the internal base clock of 27 Mhz frequency. These signals are provided to XOR gates to generate digital chaos. Similar to the ASIC case, here also we have considered r=3.3 for chaos generation. The functional simulation is used in Quartus II Software. The details of the compilation and simulation are as follows.

Flow Status	Successful - Thu Mar 05 10:57:26 2015
Quartus II Version	7.2 Build 151 09/26/2007 SJ Web Edition
Revision Name	sm
Top-level Entity Name	test
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Met timing requirements	Yes
Total logic elements	141 / 18,752 (< 1 %)
Total combinational functions	141 / 18,752 (< 1 %)
Dedicated logic registers	99 / 18,752 (< 1 %)
Total registers	99
Total pins	6 / 315 (2 %)
Total virtual pins	0
Total memory bits	0 / 239,616 (0 %)
Embedded Multiplier 9-bit elements	0 / 52 (0 %)
Total PLLs	0 / 4 (0 %)

Fig. 4.2 Quartus II Compilation Report



	Node Name	Direction	Location	I/O Bank	Vref Group	I/O Standard	Reserved	Group
1	a	Output	PIN_R19	6	B6_N0	3.3-V LVTTL (default)		
2	b	Output	PIN_U19	6	B6_N1	3.3-V LVTTL (default)		
3	c	Output	PIN_Y19	6	B6_N1	3.3-V LVTTL (default)		
4	clk	Input	PIN_L1	2	B2_N1	3.3-V LVTTL (default)		
5	x	Bidir	PIN_U21	6	B6_N1	3.3-V LVTTL (default)		
6	y	Bidir	PIN_U22	6	B6_N1	3.3-V LVTTL (default)		
7	<<new node>>							

Fig. 4.3 Pin Planner

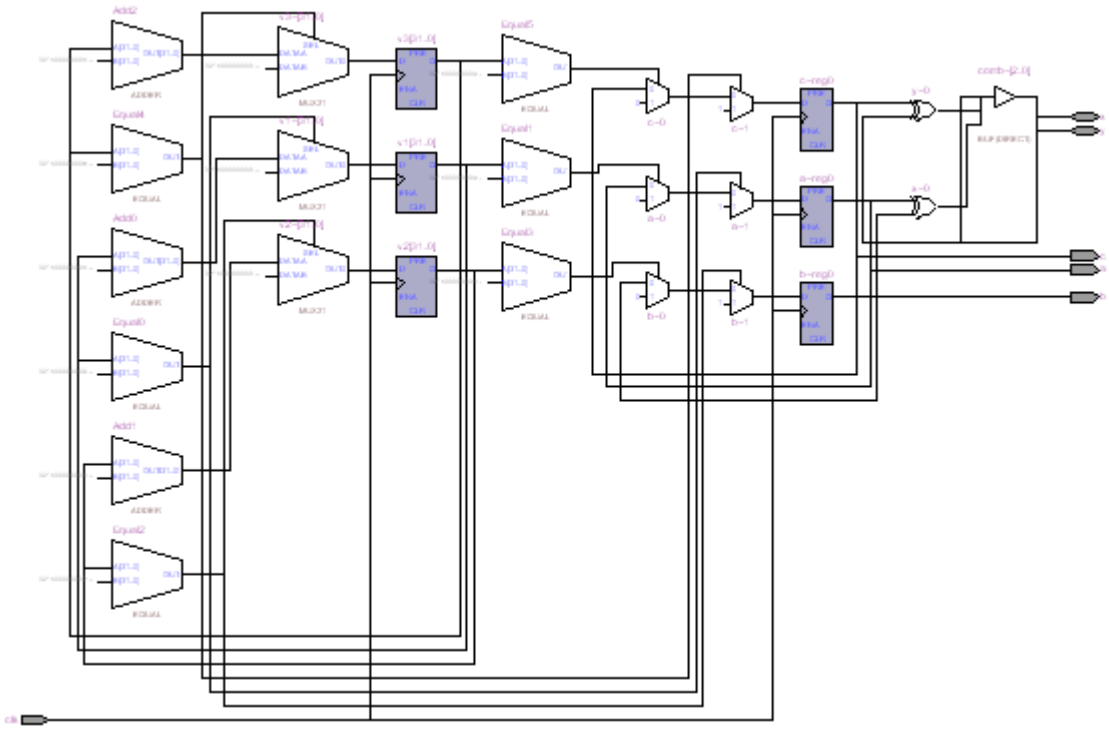


Fig. 4.4 RTL Schematic

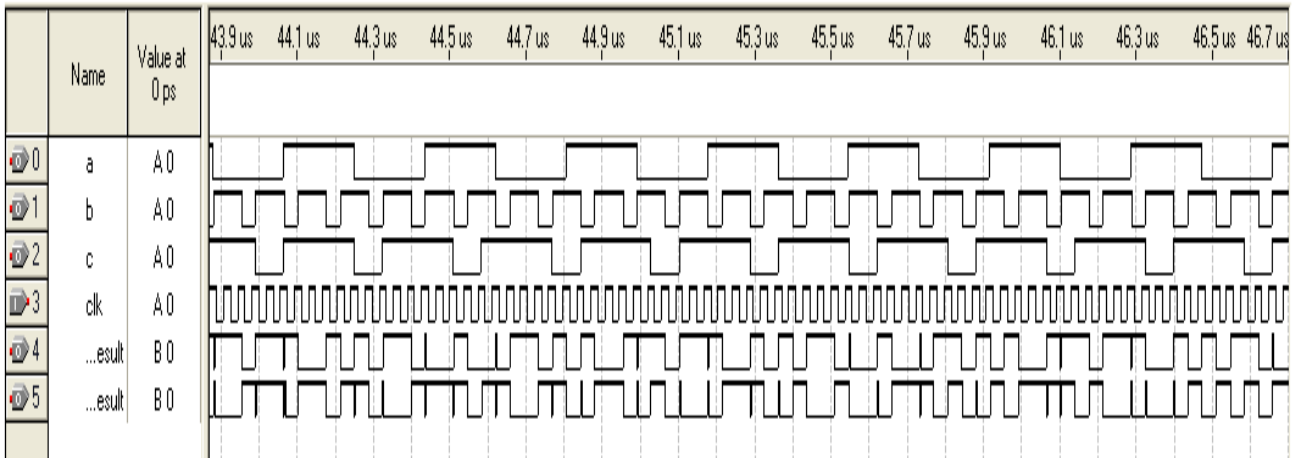


Fig. 4.5 Output Waveforms – FPGA

4.3 Implementation in MATLAB

MATLAB (matrix laboratory) is a multi-paradigm numerical computing environment and fourth-generation programming language [25]. Developed by Mathworks, MATLAB permits matrix manipulations, plotting of functions and data, and implementation of algorithms, interfacing with programs written in other languages, including C, C++, Java, Fortran and Python.

In this section the digital chaos is generated in MATLAB for different r values and a detailed characterization is done. Since the chaos generation is significantly dependent on the system, digital chaos which we have observed in FPGA Fig.3.5 is different from what we are getting in MATLAB. We have considered five different r values, one of which is the common case for all the generation cases (ASIC, FPGA) i.e r=3.3. The characterization in each case is done by calculating parameters like LLE, K2 (in bits/symbol) and D2, along with the waveform and recurrence plot (RP).

Case I: r=3.3

LLE=17.788 K2= 8.409 D2= 0.661

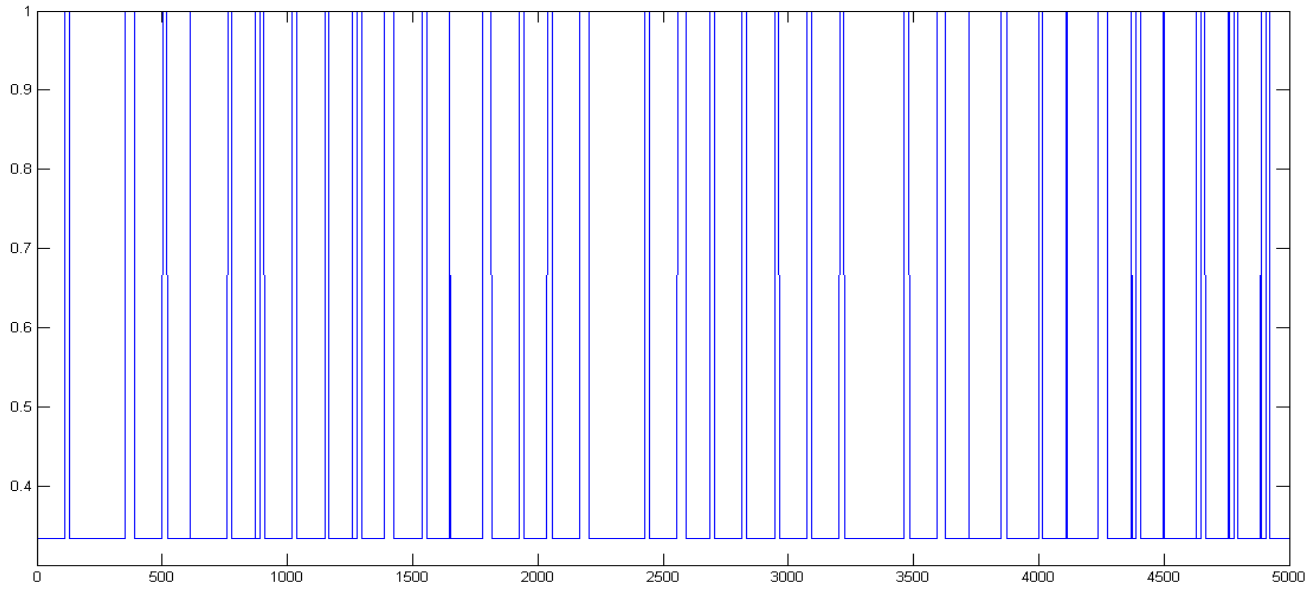


Fig. 4.6 Digital Chaos Waveform for r=3.3 - MATLAB

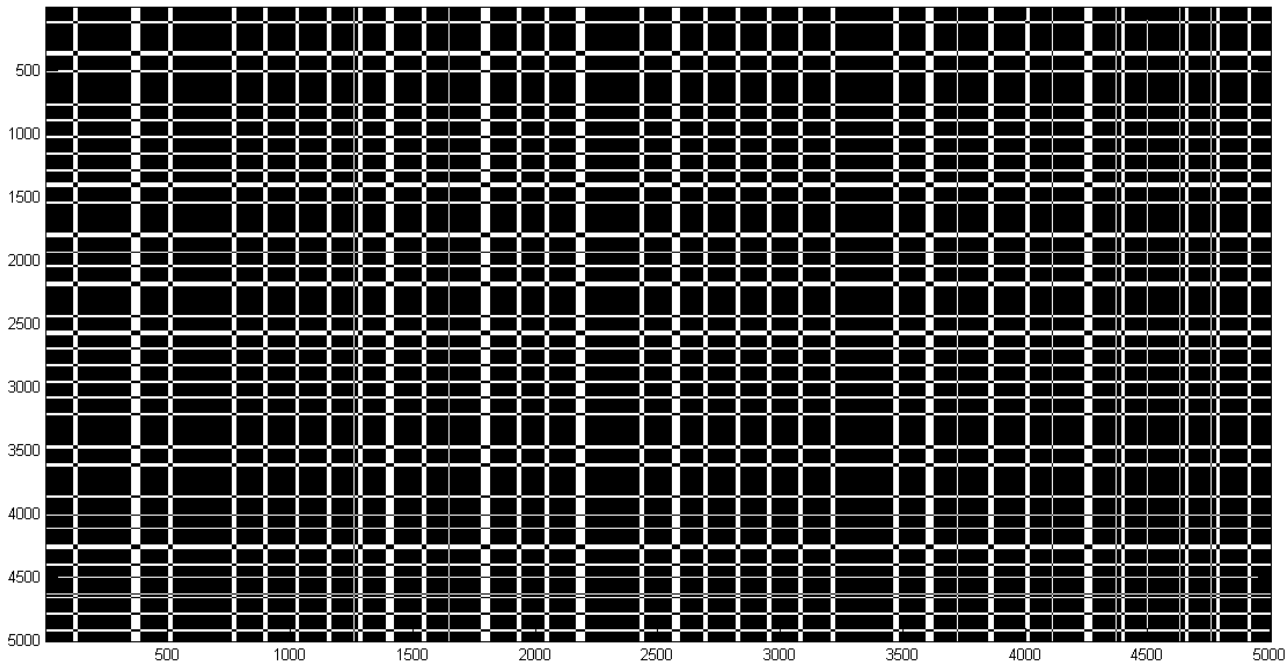


Fig. 4.7 Recurrence Plot for r= 3.3 - MATLAB

Case II: r=1.3

LLE= 22.3776 K2= 8.409 D2= 0.661

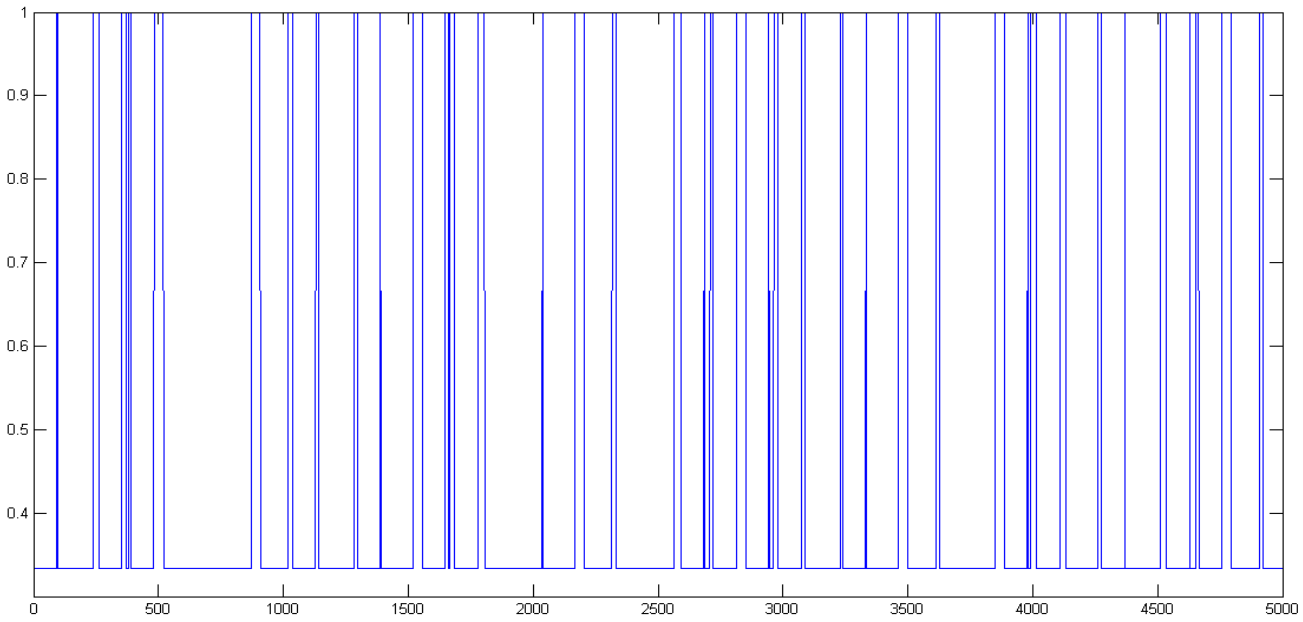


Fig. 4.8 Digital Chaos Waveform for r=1.3 - MATLAB

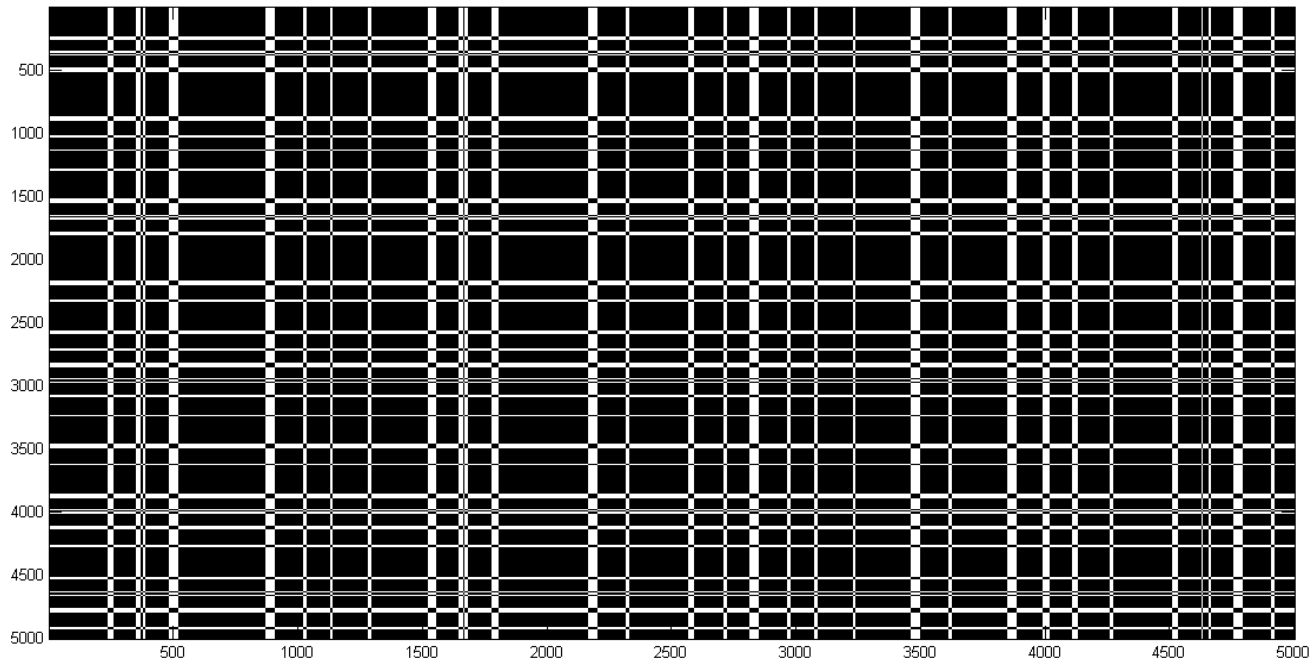


Fig. 4.9 Recurrence Plot for r= 1.3 - MATLAB

Case III: r=1.7

LLE= 18.6881 K2= 8.408 D2= 0.661

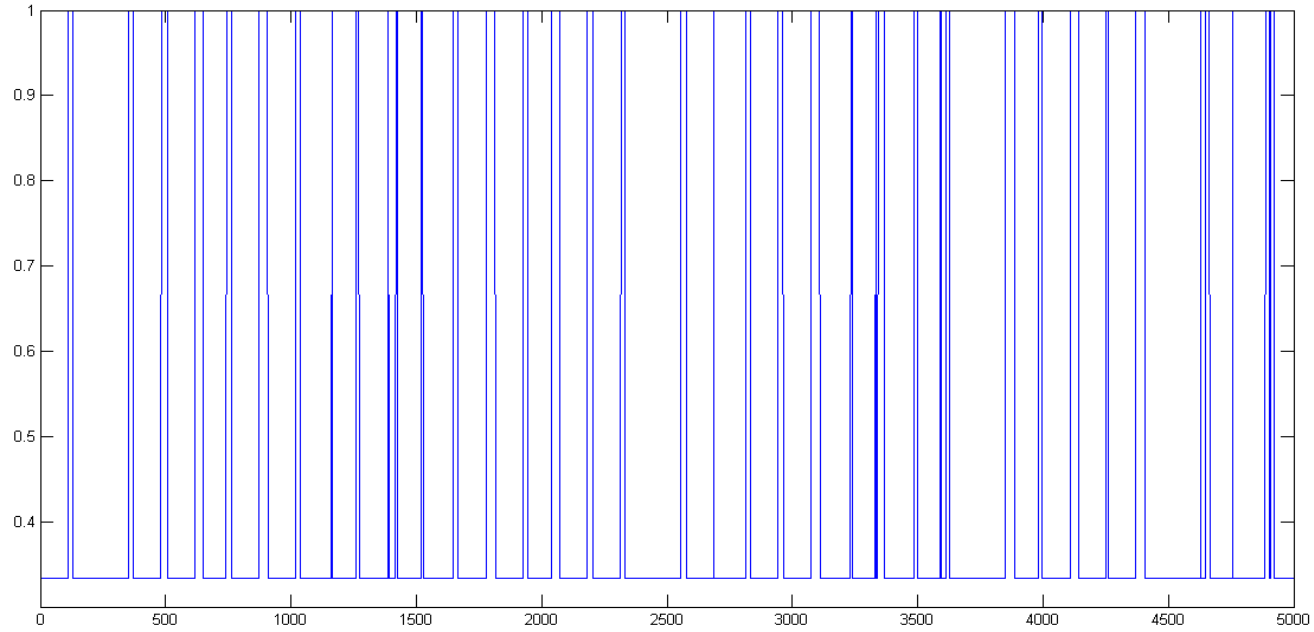


Fig. 4.10 Digital Chaos Waveform for r=1.7 - MATLAB

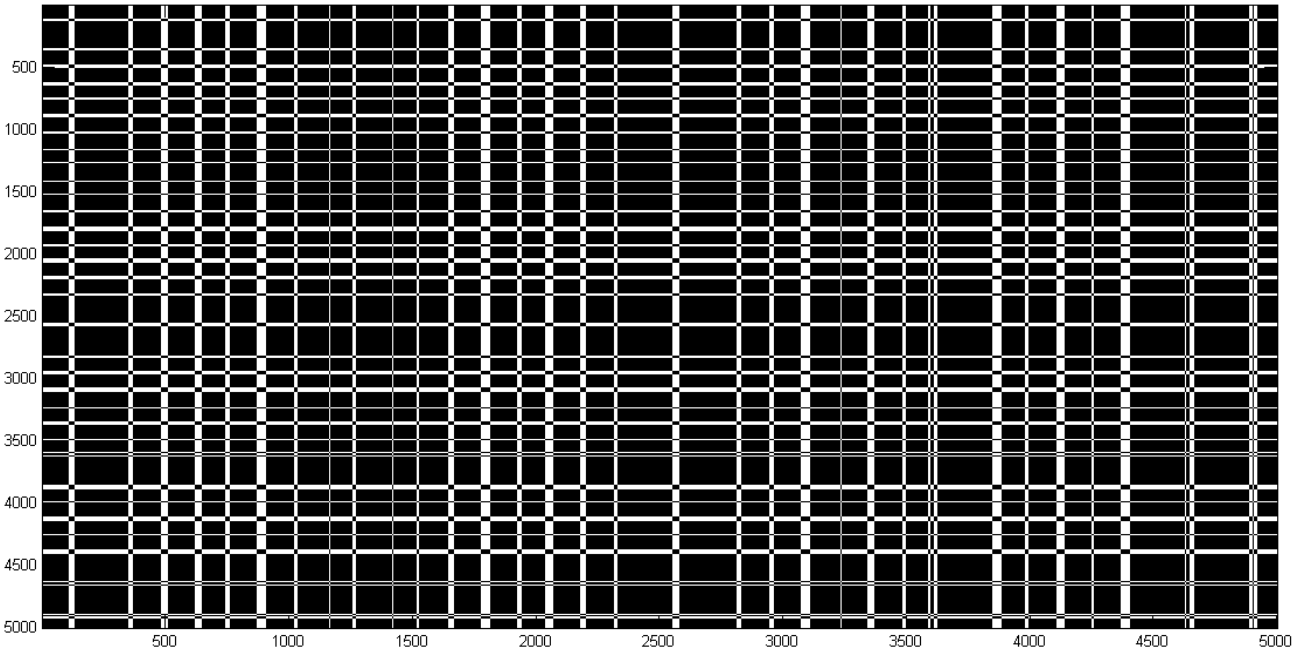


Fig. 4.11 Recurrence Plot for r= 1.7 - MATLAB

Case IV: r=5.8

LLE= 14.9199 K2= 8.408 D2= 0.661

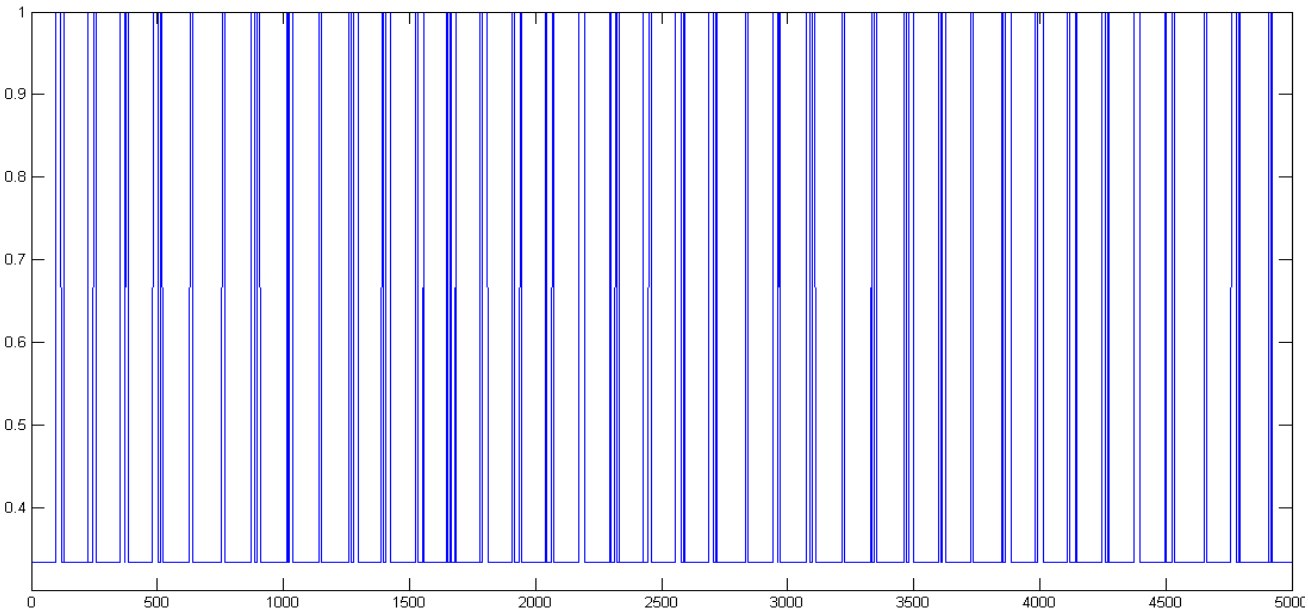


Fig. 4.12 Digital Chaos Waveform for r=5.8 - MATLAB

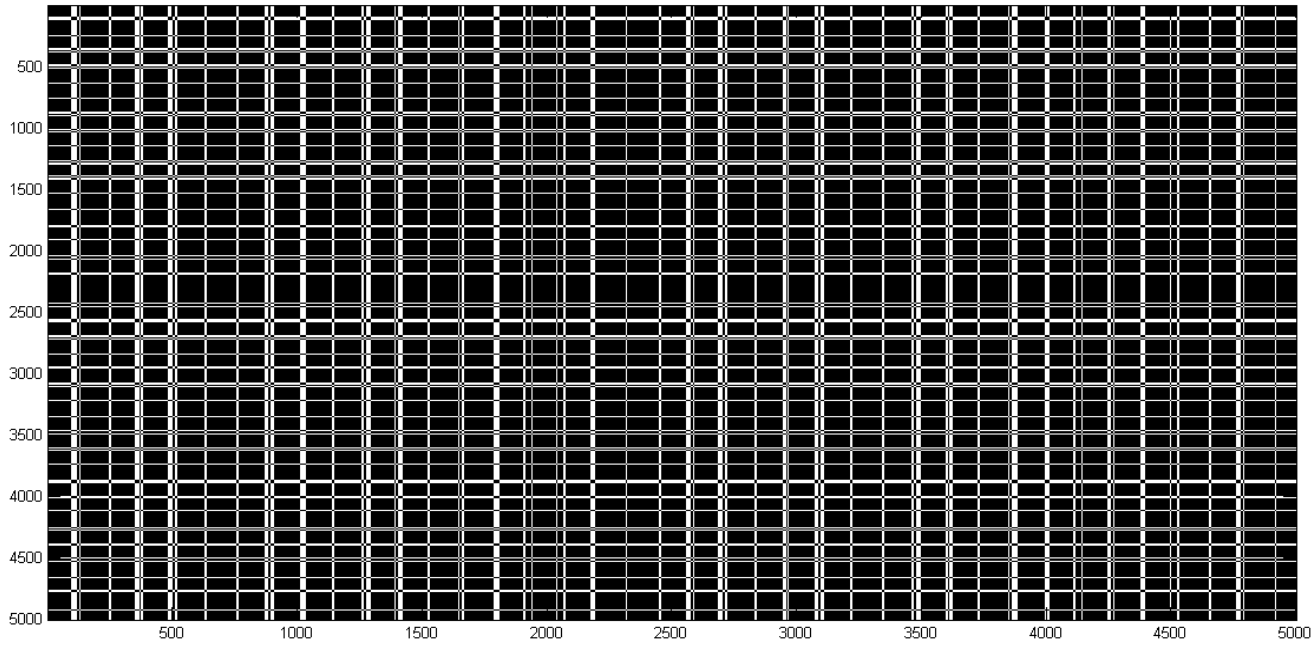


Fig. 4.13 Recurrence Plot for r= 5.8 - MATLAB

Case V: r=9.4

LLE= 14.7372 K2= 8.409 D2= 0.661

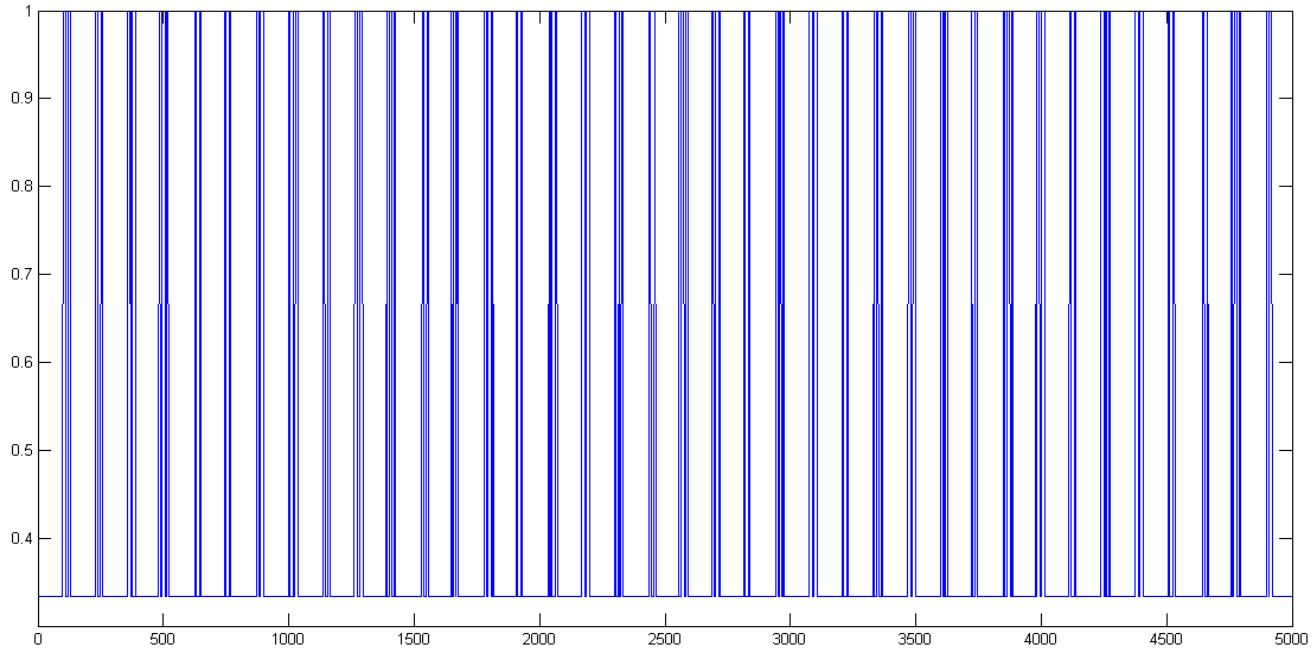


Fig. 4.14 Digital Chaos Waveform for r=9.4 - MATLAB

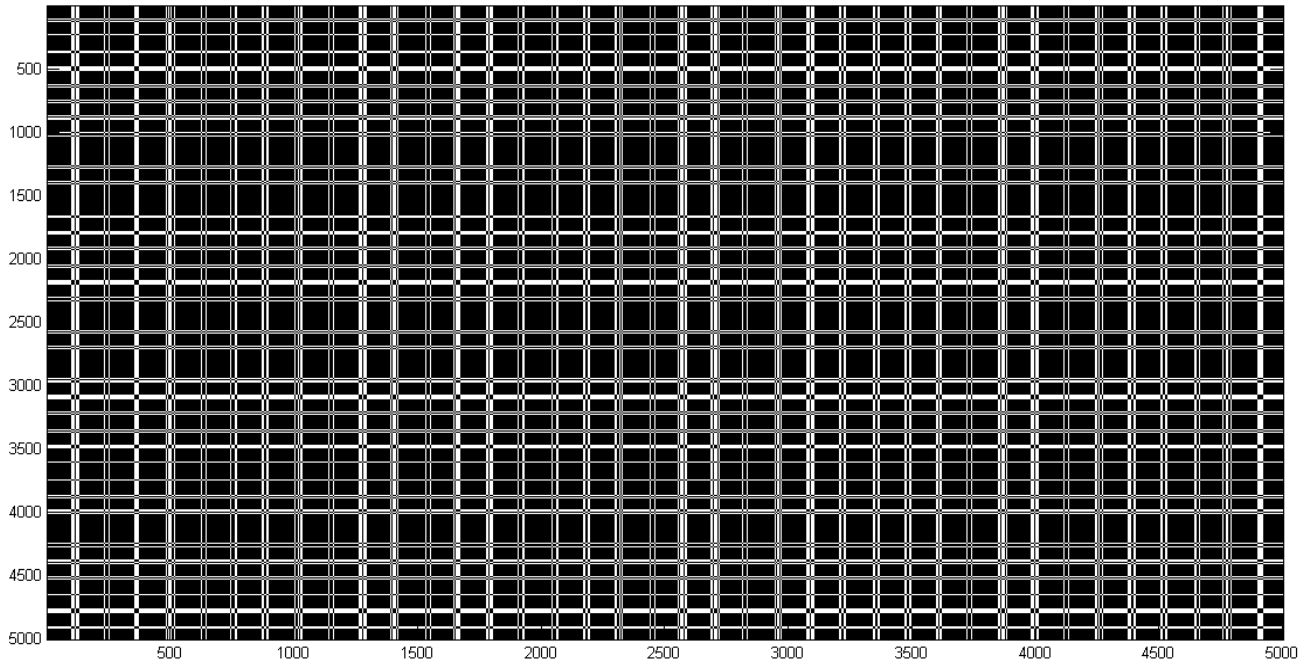


Fig. 4.15 Recurrence Plot for r= 9.4 - MATLAB

A graph between R value and LLE is plotted as shown in below figure.

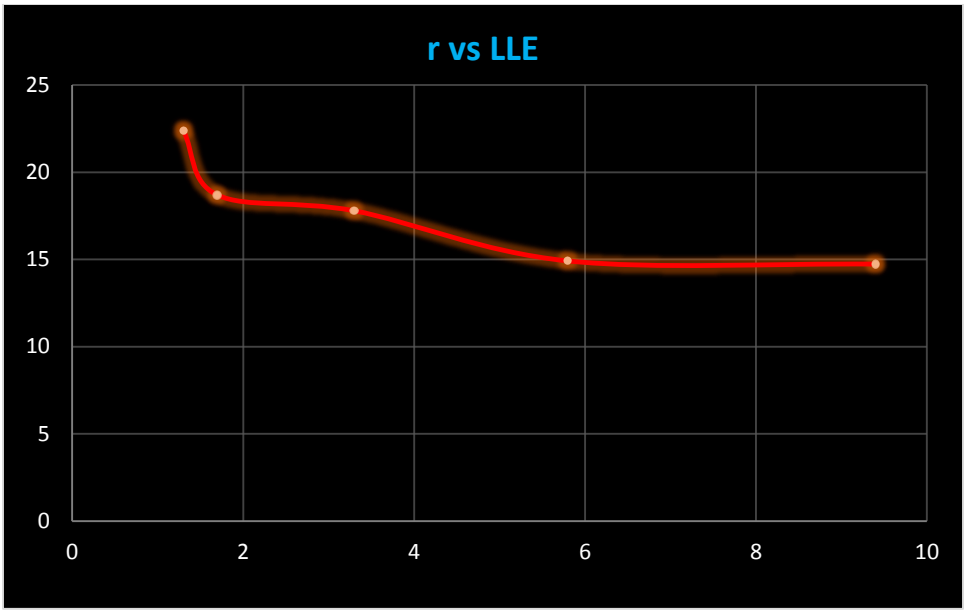


Fig. 4.16 Plot r vs LLE

From the plot we infer that as the value of r increases, LLE value is decreased until it reaches a constant value.

Different r values are used to generate Digital chaos their entropies are calculated. The variation of entropy (K2) with respect to r value is as shown in the Fig. 4.17. The average of entropies calculated for 10 different r values is 7.8915 bits/symbol which is the standard K2 for digital Chaos.

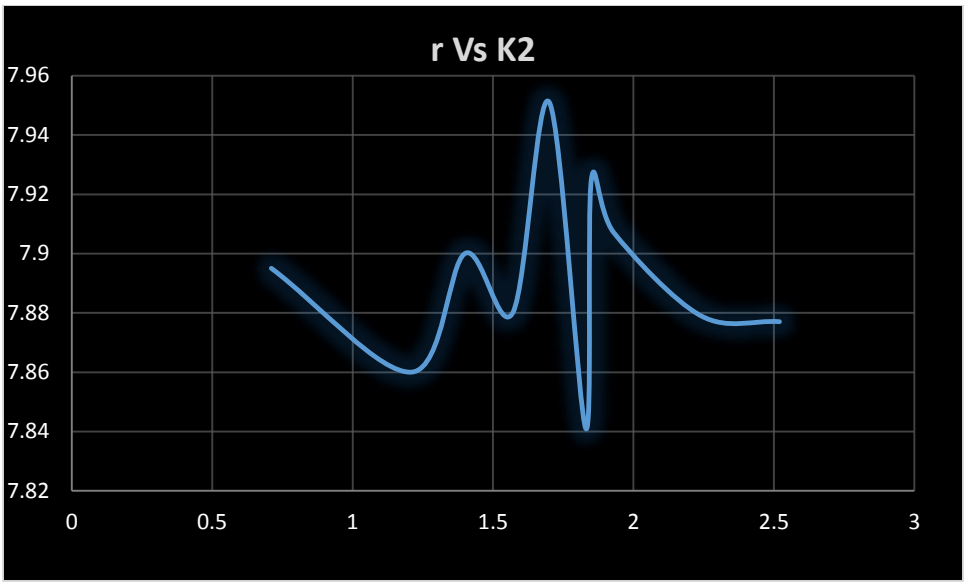


Fig. 4.17 Plot r vs K2

4.4 Conclusion

Thus in this chapter we have developed a programmable platform using FPGA and MATLAB to generate “Digital Chaos” and standard characterization is performed.

REFERENCES

[1] R. Broadhurst, P. Grabosky, M. Alazab, S. Chon, Organizations and Cybercrime: *An Analysis of the Nature of Groups engaged in Cyber Crime*, Int. J. Cybercriminology, **8** (2014).

[2] M. Hilbert, *How much of the global information and communication explosion is driven by more, and how much by better technology?*, Wiley Journal of the Association for Information Science and Technology, **65**, 856-861 (2014).

[3] K. E. Himma, *Internet Security: Hacking, Counterhacking, and Society*, (Jones and Bartlett, UK, 2007).

[4] T. H. Lan, M. F. Mansour, A. H. Tewfik, *Robust high capacity data embedding*, Image Processing 2000, **1**, 581-584 (2000).

[5] X.Wu, X.Zhu, G.Q.Wu and W.Ding, *Data mining with big data*, IEEE Trans. on Knowledge and Data Engineering, **26**, 97-107 (2014).

[6] M. Lakshmanan and K. Murali, *Synchronized Chaotic Systems and Secure Communication*, Chaos in Nonlinear Oscillators: Controlling and Synchronization, **13**, 235-283 (1996).

[7] Young-Sik Kim, Jong-Hwan Kim, Sang-Hyo Kim, *A Secure Information Transmission Scheme With a Secret Key Based on Polar Coding*, IEEE Communications Letters, **18**, 937-940 (2014).

[8] K. E. Barner and G. R. Arce, *Nonlinear Signal and Image Processing: Theory, Methods and Applications*, (CRC Press, U.S, 2003).

[9] E.Bilotta and P.Pantano, *A gallery of Chua attractors*, (World Scientific, Singapore, 2008).

[10] S. H. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*, (Westview Press, Cambridge, 2008).

[11] M. Ausloos, M. Dirickx, *The Logistic Map and the Route to Chaos: From the Beginnings to Modern Applications*, (Springer, US, [2006]).

[12] M. H. Jensen, P. Bak, T. Bohr, *Complete Devil's Staircase, Fractal Dimension, and Universality of Mode-Locking Structure in the Circle Map*, Phys. Rev. Lett., **50**, 1637 (1983).

[13] Xiaowu Wang, Ronnie Mainieri, J. H. Lowenstein, *Circle-map scaling in a two-dimensional setting*, Phys. Rev. A., **40**, 5382 (1989).

[14] R. G. James, K. Burke, J. P. Crutchfield, *Chaos forgets and remembers: Measuring information creation, destruction and storage*, Int. J Bifurcation Chaos, **378**, 2124 (2014).

[15] M. T. Rosenstein, J. J. Collins, C. J. De Luca, *A practical method for calculating largest Lyapunov exponents from small data sets*, Physica D, **65**, 117 (1993).

[16] P. Maragos, F. K. Sun, *Measuring the fractal dimension of signals: morphological covers and iterative optimization*, IEEE Trans. Signal Processing, **41**, 108-121 (1993).

[17] M. F. Barnsley, *Fractals Everywhere*, (Courier, Dover, 2008).

[18] G. B. Giannakis, F. Bach, R. Cendrillon, M. Mahoney, J. Neville, *Signal Processing for Big Data*, IEEE Signal Processing Magazine, **31**, 15-16 (2014).

[19] P. Grassberger and I. Procaccia, *Estimation of the Kolmogorov entropy from a chaotic signal*, Phys. Rev.A, **28**, 2591-2593 (1983).

[20] N. Marwan, M. C. Romano, M. Thiel and J. Kurths, *Recurrence Plots for the Analysis of Complex Systems*, Physics Reports, **438**, 237–329 (2007).

[21] D. Roy Choudhury and Shail B. Jain, *Linear Integrated Circuits*, (New Age International, INDIA, 2003).

[22] B. Razavi, *RF Microelectronics*, (Prentice Hall, US, 2011).

[23] J. P. Uyemura, *Chip Design for Submicron VLSI: CMOS Layout and Simulation*, (Thomson/Nelson, USA, 2006).

[24] V. A. Pedroni, *Circuit Design with VHDL*, (MIT, USA, 2004).

[25] A. Gilat, *MATLAB: An Introduction with Applications*, (Wiley, USA, 2014).

[26] L. Shujun, M. Xuanqin and C. Yuanlong, *Pseudo-random Bit Generator Based on Couple Chaotic Systems and Its Applications in Stream-Cipher Cryptography*, Springer, **2247**, 316-329 (2001).

[27] A.Uchida1, K. Amano, M. Inoue, K. Hirano, S. Naito, H. Someya, I. Oowada, T. Kurashige, M. Shiki, S. Yoshimori, K. Yoshimura, and P. Davis, *Fast physical random bit generation with chaotic semiconductor lasers*, Nature Photonics, **2**, 728-732 (2008).

[28] V. Patidar , K. K. Sud and N. K. Pareek, *A Pseudo Random Bit Generator Based on Chaotic Logistic Map and its Statistical Testing*, Informatica, **33**, 441-452 (2009).

[29] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray and S. Vo, *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, (NIST Publications, USA , 2010).

[30] K. D. Wagner, C. K. Chin, and E. J. McCluskey, *Pseudorandom Testing,* IEEE Transactions on Computers, **C-36** (1987).