Digital Communication System with High Security and High Immunity

A Thesis Submitted to The University of Kent

For The Degree Of DOCTOR OF PHILOSOPHY

In ELECTRONICS ENGINEERING

By
Ahmed S. Alshammari

August, 2017

Abstract

Today, security issues are increased due to huge data transmissions over communication media such as mobile phones, TV cables, online games, Wi-Fi and satellite transmission etc. for uses such as medical, military or entertainment. This creates a challenge for government and commercial companies to keep these data transmissions secure. Traditional secure ciphers, either block ciphers such as Advanced Encryption Standard (AES) or stream ciphers, are not fast or completely secure. However, the unique properties of a chaotic system, such as structure complexity, deterministic dynamics, random output response and extreme sensitivity to the initial condition, make it motivating for researchers in the field of communication system security. These properties establish an increased relationship between chaos and cryptography that create strong and fast cipher compared to conventional algorithms, which are weak and slow ciphers. Additionally, chaotic synchronisation has sparked many studies on the application of chaos in communication security, for example, the chaotic synchronisation between two different systems in which the transmitter (master system) is driving the receiver (slave system) by its output signal.

For this reason, it is essential to design a secure communication system for data transmission in noisy environments that robust to different types of attacks (such as a brute force attack). In this thesis, a digital communication system with high immunity and security, based on a Lorenz stream cipher chaotic signal, has been perfectly applied.

A new cryptosystem approach based on Lorenz chaotic systems was designed for secure data transmission. The system uses a stream cipher, in which the encryption key varies continuously in a chaotic manner. Furthermore, one or more of the parameters of the Lorenz generator is controlled by an auxiliary chaotic generator for increased security. In this thesis, the two Lorenz chaotic systems are called the Main Lorenz Generator and the Auxiliary Lorenz Generator. The system was designed using the SIMULINK tool. The system performance in the presence of noise was tested, and the simulation results are provided. Then, the clock-recovery technique is presented, with real-time results of the clock recovery. The receiver demonstrated its ability to recover and lock the clock successfully. Furthermore, the technique for

synchronisation between two separate FPGA boards (transmitter and receiver) is detailed, in which the master system transmits specific data to trigger a slave system in order to run synchronously. The real-time results are provided, which show the achieved synchronisation. The receiver was able to recover user data without error, and the real-time results are listed.

The randomness test (NIST) results of the Lorenz chaotic signals are also given. Finally, the security analysis determined the system to have a high degree of security compared to other communication systems.

Acknowledgments

Firstly, I would like to express my sincere to my supervisor Prof. Mohamed Sobhy for the continuous support of my PhD study and related research and, for his patience and motivation. It has been a great honor to work with him and gained invaluable expert throughout this work. Furthermore, I would like to thank my second supervisor Dr. Peter Lee for his support and encouragement throughout this work.

Beside my supervisors, I would like to thank Dr. Mark Esdale for his support throughout this work.

I would like to express my special thanks to my mother, Rouf Saad Alshammari, for her spiritually support. Words cannot express how grateful I am to my wife, Suaad Suliman Alshammari, my kids, Saud, Fahad, Abdulmailk and my little daughter Haneen, for their support and patience through my ups downs, without them, I never would have been able to achieve my goals.

I would like to convey my special thanks to my father in low Mr. Suliman Namer Alshammari and Mrs. Juzaiah Alshammari for their continual support.

Last but not the least, thank you to everyone who support and encourage me. In particular, my brothers, Mamdouh Saud Alshammari, Fahad Kulaif Alenzi, Alhumadi Salamh Alshammari and Tuwallai Turki Alshammari.

Contents

List of Figures

Glossary of Abbreviations

ADC ..... Analogue to Digital Converter
AES ..... Advanced Encryption Standard
AHM ..... Hyper Chaos Masking
AWGN ..... Added White Gaussian Noise
BER ..... Bit Error Rate
BEP ..... Bit Erorr Probablity
BIT ..... Bitstream
BPSK ..... Binary Phase Shift Keying
CCS_PRBG ..... Coupled Chaotic Systems_ Pesudorandom Binary Generator
CDMA ..... Code Division Multiple Access
CFB ..... Cipher FeedBack
CSK ..... Chaotic Shift Keying
CUSUM ..... Cumulative Sum
GPU ..... Default Clock Driver
DAC ..... Digital to Analog
DCSK ..... Differential Chaotic Shift-Keying
DCR ..... Data and Clock Recovery
DCM ..... Digital Clock Mangement
DES ..... Data Encryption Standard
DS-CDMA ..... Direct Sequence-Code Division Multiple Access
XOR ..... Exclusive OR Gate
FFT ..... Fast Fourier Transform
FIPS ..... Federal Information Processing Standard
FIR ..... Finite Impulse Response
FPGA ..... Field Programmable Gate Array
FSM ..... Finite State Machine
GPU ..... Graphics Processing Unit
HDL ..... Hardware Description Language
IC ..... Integrated Circuit
ICA ..... Independent Component Analysis

ISE..... Integrated Software Environment

LFSR .....Liner Feedback Shift Register

LP .....Length of the Plain Text

LPI..... Low Probablity of Interception

LUTs ..... Look-Up-Tables

MAI ..... Multiple Access Interference

m-sequence .....Maximal-Length Sequences

MSB ..... Most Significant Bit

NIST .....National Institue of Standard and Technonlgy

NRZ ..... Non-Return to Zero

OFB .....Output FeedBack

OTP .....One Time Pad

PD.....Phase Detector

Pe..... Probability of error

PLL.....Phase Lock Loop

PM..... Peripheral Module

PN..... Pesudo-Random Noise

RC4 .....Rivest Cipher

RHCS ..... Robust Hyper Chaotic System

ROM.....Read Only Memory

RSA.....Rivest-Shamir-Adleman

RTL .....Register Transfer Level

SEAL .....Software-Optimised Encryption Algorithm

SEL ..... Select

SPI .....Serial Peripheral Interface

SNR ..... Signal to Noise Ratio

SS ..... Spread Spectrum

UCF ..... User Constraint File

UNG .....Uniform Noise Generator

VCO ..... Voltage Controlled Oscillator

VCMOS ..... Voltage Complementary Metal Oxide Semiconductor

VHDL..... Very high speed integrated circuit Hardware Description Language

VLSI ..... Very Large Scale Integration

XCORR ..... Cross-Correlation

List of Table

Chapter 1

INTRODUCTION

1.1 The Bases of Secure Communication

Much of the interaction today tends to be electronic, such as online shopping or social engagements occurring within social media. Sensitive electronic information such as bank transactions, telecommunications and private image transmissions increases the responsibility of the government (as well as commercial companies) concerning ways to increase the security level to prevent hacking. Modern high speed processing computers, powerful Field Programmable Gate Array (FPGA) boards and Graphics Processing Unit (GPU) boards help hackers perform brute force attacks that attempt every possible key [1, 2]. These electronic threats make the research area of digital communication security important and attractive for researchers to protect transmission information from unauthorised people.

Any secure communication system between two parties often contains three main parts: authentication, confidentiality and integrity [3]. Authentication in communication systems is done through identity. This means that, in order to establish communication between sender and receiver, the identity of both must be confirmed. Confidentiality takes place when only the sender and receiver are able to understand the content of the transmitted message. The process is started by the sender, who encrypts the content of the message using a cryptography algorithm and then transmits it through the channel. The receiver must know the private key to decrypt the message. Integrity means that the message content has not been altered when the sender and receiver are communicating.

Modern cryptography software, such as the Advanced Encryption Standard (AES), has become the heart of security since 2001 and has been used for many sensitive applications. These encryption-technique algorithms are based on a symmetric, or private, key. However, every algorithm may have pros and cos, due to attack by powerful modern computers and newly-developed boards used for high-speed application requirements [4, 5].

In spread spectrum communication systems, the disadvantage of using conventional data encryption based on traditional pseudorandom sequence generators such as maximal-length sequences (m-sequences), Walsh sequences and Gold sequences is the limited number of iterations performed by encryption transformation. Furthermore, lacks of correlation properties due to a limited available number of PN sequences [6-15].

1.2 Block Cipher and Stream Cipher

Symmetric (private) keys are divided into two main categories: block ciphers and stream ciphers. The private key must be known for both the transmitter and the receiver to be able to retrieve the encrypted message. A block cipher is based on sequences of fixed numbers of bits called 'blocks'. It encrypts an entire block of plaintext bits with the same key [16]. A stream cipher generates an infinite cryptographic keystream that encrypts bits individually, similar to the one-time pad (See chapter 3) [16]. Fig. 1.1 shows basic diagrams of the block and stream ciphers.

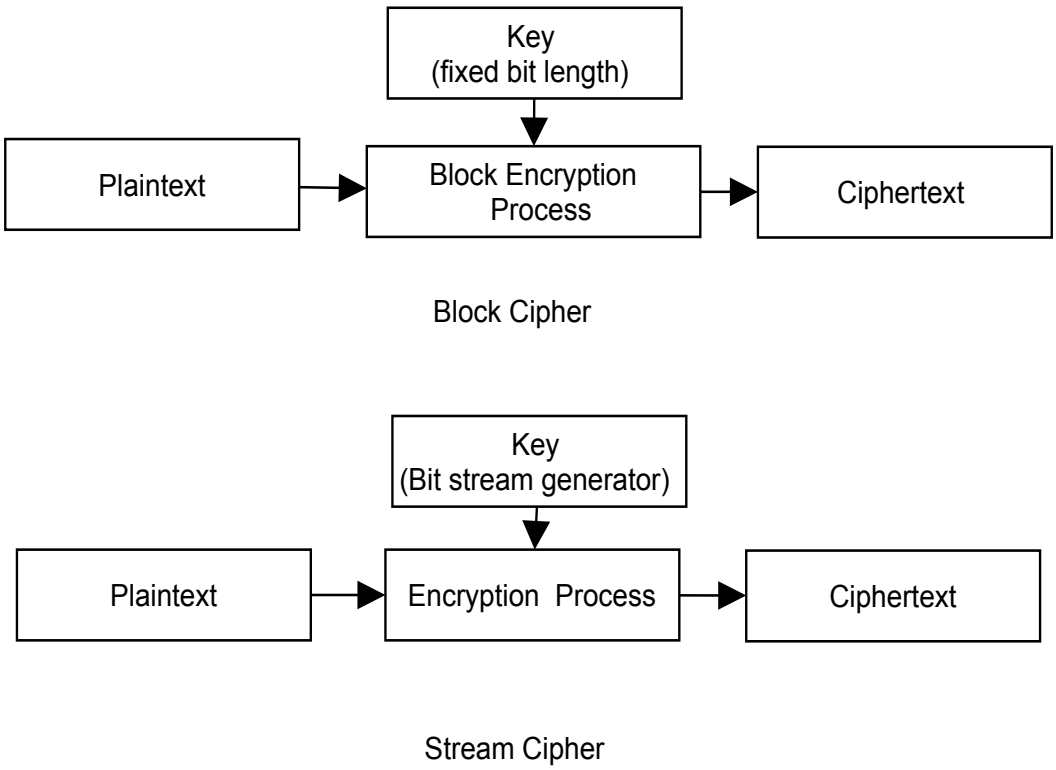

Fig. 1.1. Basic Diagrams of Block and Stream Ciphers.

1.3 Properties of Chaos in Communication System Applications

A digital communication system based on chaos equations for data transmission security has advantages over traditional secure communication systems. A main advantage of this system is its ability to generate pseudorandom numbers based on non-linear behaviour. Therefore, we can generate an infinite number of uncorrelated binary streams that can be used in the Code Division Multiple Access (CDMA) communication system as a code for large value of users.

Chaos is an aperiodic long-term behaviour in a deterministic system that exhibits sensitive dependence on initial condition [17]. One chaos system property is its deterministic quality, meaning that the system has no random or noisy inputs or parameters and system randomness behaviour comes from the nonlinearity, not the effect of noisy driving forces [17]. Another chaos property is its aperiodicity; the system trajectories do not settle down to fixed points and periodic orbits [17]. The chaos system also has a high sensitivity to the initial condition. Any slight change in the initial condition leads the chaotic model to a very quick change in output response. This means that, the new output response after the initial condition has been changed is unrelated to the previous output generated. The trajectories separate exponentially fast and the system has a positive Lyapunov exponent [17].

Cross-correlation is a measure of similarity between two signals (an input signal and a reference signal) and used in this system as a signal detector, whereas the autocorrelation measures the correlation between the signal and time-delayed version of itself. [18]. This means that, each piece of user data transmitted in one channel can be discriminated based on auto-correlation functions and other users' data are rejected based on cross-correlation functions.

The cross-correlation in analogue and digital forms equations are given as:

$$C_{xy}(\tau) \equiv \int_{-\infty}^{\infty} x(t) * y(t-\tau)\, dt \quad (1)$$

$$C_{xy}(m) \equiv \sum_{n=-\infty}^{\infty} y(x) * y(n-m) \quad (2)$$

Where the time shift τ or m is called the lag.

The auto-correlation in analogue and digital forms equations are given as:

$$C_{s}(\tau) \equiv \int_{-\infty}^{\infty} s(t) * s(t-\tau)\, dt \quad (3)$$

$$C_{s}(m) \equiv \sum_{n=-\infty}^{\infty} s(x) * s(n-m) \quad (4)$$

Where the time shift τ or m is called the lag.

Fig. 1.2 shows the two Lorenz systems' output responses (illustrated after the arrow). All initial conditions and system parameters are identical for the two Lorenz systems except one system parameter is different.

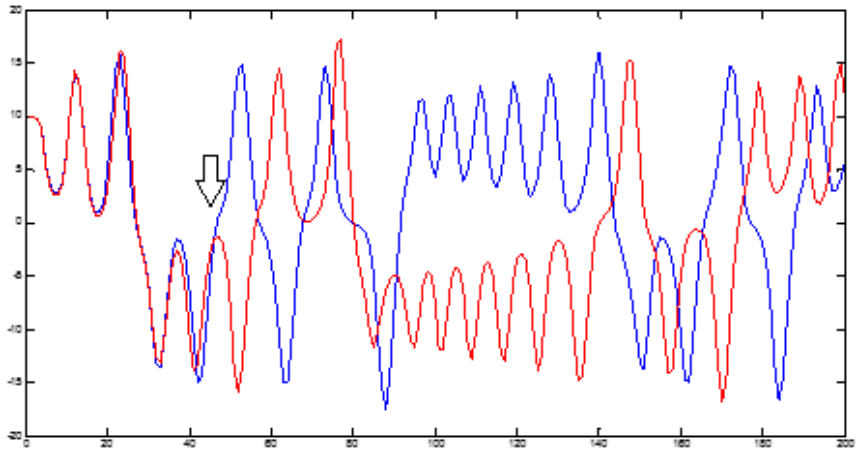

Fig. 1.2. Responses of Two Lorenz Systems with Different Initial Conditions.

Additionally, chaotic signals are broadband [19]. Using chaos equations for data transmission security has attracted researcher's attention on implementing secure communication system using chaotic signals [7, 9, 20-26].

1.4 Chaos for Spread-Spectrum Technology

Chaotic signals can be used in different communication system applications, some of which include spread-spectrum applications [3]. Spread-spectrum technology was originally invented by the military because of the anti-jam and low probability of intercept (LPI) properties of this technology. More recently, spread-spectrum technology has been used by commercial telecommunication companies for wireless systems. Spread-spectrum uses wideband, noise-like signals that are hard to detect and jam. Spread-spectrum signals are made to be a much wider band than the information they are carrying. One feature is that the transmitter uses the same transmit power level as a narrowband transmitter. This is because the spread-spectrum signal is wider, thus it transmits at a much lower power density. Additionally, the spread-spectrum can combine noise immunity and a high data rate, making this technology suitable for wireless data communication networks in noisy environments.

1.5 Reception Types Used with Chaos-Based Communication Systems

Two types of chaos-based communication systems have been widely studied. The first is a Coherent Detection. The chaotic signal is used to modulate the information signal, for example, chaos-based direct-sequence code division multiple access (DS-CDMA). In this scheme, synchronisation is necessary for the receiver to be able to recover the information signal. The synchronisation means that at the receiver, the chaotic generator is regenerating an exact replica of the chaotic bit stream [27-30]. The aim of using chaotic signals, specifically in chaos-based spread-spectrum systems, is to overcome the weakness in conventional Pseudo-Random Noise (PN) sequences [27]. The second type of chaos-based communication system has a Non-Coherent Detection scheme. The receiver design estimates the received signal to recover the data. A transmitted signal is not required to regenerate the chaotic signal at the receiver [22, 27, 31, 32].

1.6 Spreading and De-Spreading Methods for CMDA

At the transmitter, the transmitted user stream $b_k(t)$ of bit duration T_b is spread using multiplication by a specific spreading code $c_k(t)$. And $s_k(t)$ is the encoded user data at the output of k^{th} encoder and is expressed as shown below.

$$s_k(t) = b_k(t)c_k(t) \tag{5}$$

where $b_k(t)$ is the k^{th} user's binary data signal donated by

$$b_k = \sum_{l=-\infty}^{\infty} b_l^{(k)} P_{T_d}(t - lT) \tag{6}$$

where $b_l^{(k)}$ represents the k^{th} data cycle that takes on 0 or 1 for each l with the same probability and P_{T_d} is the rectangular pulse of interval T which starts at $t = 0$. In addition, $c_k(t)$, the code sequence of the k^{th} user is expressed by

$$c_k(t) = \sum_{j=-\infty}^{\infty} A_j^{(k)} P_{T_c}(t - jT_c) \tag{7}$$

where $A_j^{(k)} \in \{0,1\}$ represents the j^{th} value of the k^{th} user spreading code and P_{T_c} is the rectangular pulse of duration T_c

Then, all users data are summed and transmited through one channel. $r(t)$ denotes the summed signal and is shown below.

$$r\,(\mathrm{t}) = \sum_{k=1}^{n} s_{\,\mathrm{k}}\,(\mathrm{t} - \tau_{\,\mathrm{k}}) \tag{8}$$

where $s_k\ (t-\tau_k)$ corresponds to the k^{th} user's signal, n is number of user's, and τ_k represents the random time delay associated with the k^{th} signal.

The conventional method that the received signal $r(t)$ that goes into the correlation block has been multiplied by a replica of the spreading code (chaotic signal) as stated below

$$r_{CORR}(t) = r(t)c_1(t) = \left(\sum_{k=1}^{n} b_k(t)c_k(t)\right)c_1(t) \tag{9}$$

Where $b_k(t)$ the transmitted user data, $c_k(t)$ a specific spreading code. The output of the first user's integrator is given by

$$Y_1 = w * b_i^1 + \sum_{k=1}^{n} b_i^k \int_0^T c_k(t)c_1(t)dt = w * b_i^1 + I_1 \tag{10}$$

where b_i^1 is the i^{th} data of the first user that can take on two values "0" or "1" with equal probability and w is the number of ones in the spreading code sequence. The first term of (5) $w * b_i^{(1)}$ is the desired signal and the second term I_1, is the undesired signal (total interference signal) which is the effect of the n^{th} user's signal on one of the chosen user in the receiver. Then, the signal Y_1 goes through a limiter, with chosen threshold $S = w$. As a result, the output of the limiter will take values

$$\text{if } Y_1 \geq S \Rightarrow b_i^{(1)} = 1$$

and

$$\text{if } Y_i^{(1)} < S \Rightarrow b_i^{(1)} = 0$$

The Multiple Access Interference (MAI) is the only limitation of transmitting data from transmitter to the receiver.

1.7 Generation of Chaotic Signals for Encryption.

In this work, we need to generate chaotic signals to perform the encryption of data. Chaotic generators are divided into two categories, autonomous and non-autonomous. Autonomous systems generate their own signals and do not need to be driven by an external source. Non-autonomous systems convert an external signal, such as a harmonic signal, to a chaotic signal.

All chaotic system must have a nonlinear element and a number of state variables and the order of the system is the number of independent state variables. The minimum order for an autonomous system is three and for a non-autonomous system is two.

There are three well known autonomous minimum order systems, the Chua, Rössler and Lorenz systems. In the following section we evaluate the suitability of these systems for our application.

1.7.1 The Chua system

The Chua system is based on a nonlinear analogue circuit that is easy to build. It is popular in studying and demonstrating chaos. The circuit is shown in Fig. 1.3.

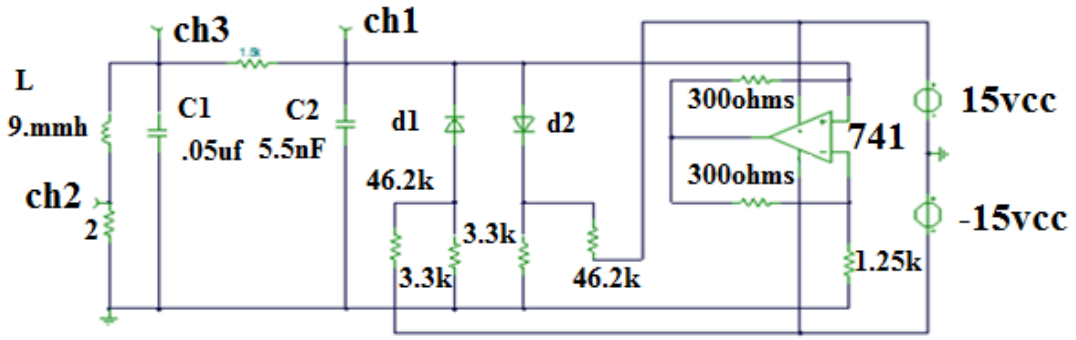

Fig. 1.3. The Chua Circuit.

Analysis of the circuit gives the state equations.

$$\begin{aligned} \dot{v}_{C_1} &= \frac{G}{C_1}(v_{C_2} - v_{C_1}) - \frac{1}{C_1} g(v_{C_1}) \\ \dot{v}_{C_2} &= \frac{G}{C_2}(v_{C_1} - v_{C_2}) + \frac{1}{C_2} i_L \\ i_L &= -\frac{1}{L} v_{C_2} \end{aligned} \tag{11}$$

Where $g(v_{c1})$ is the non-linear function represented by the diodes in Fig. 1.3.

In order to use this circuit in any digital communication system a SIMULINK model has been developed and is shown in Fig. 1.4. The state equations are written in integral form to correspond directly to the SIMULINK model.

$$v_{C1} = \int S_1(G(v_{C2} - v_{C1}) - f(v_{C1}))\, dt + I_1 \qquad \text{(a)}$$

$$v_{C2} = \int S_2(G(v_{C1} - v_{C2}) - i)dt + I_2 \qquad \text{(b)} \quad (12)$$

$$I_L = \frac{1}{L}\int S_3((v_{C2} - i_L R_o))dt + I_3 \qquad \text{(c)}$$

Where S_1, S_2 and S_3 are frequency multipliers and I_1, I_2 and I_3 are initial conditions.

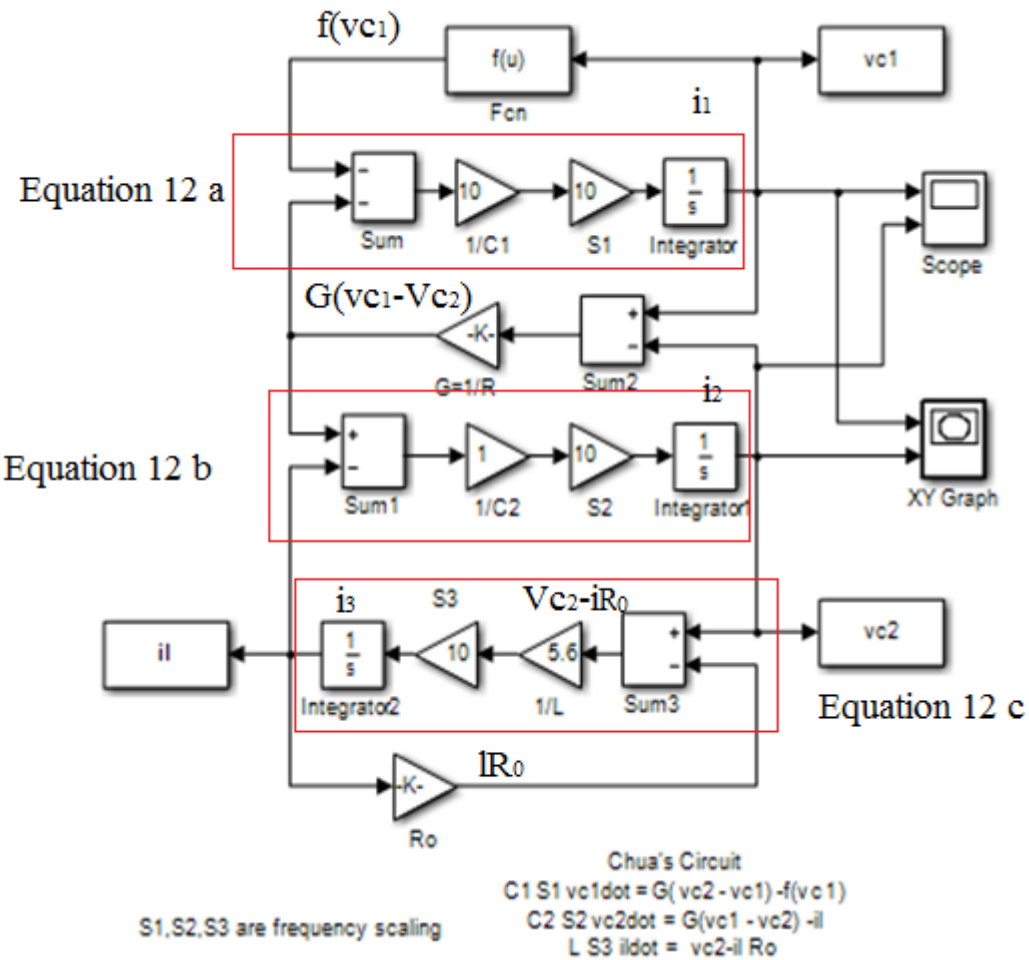

Fig. 1.4. SIMULINK model of the Chua circuit.

The v_{C1} and v_{C2} signals from the SIMULINK model are shown in Fig. 1.5. It shows v_{C1} and v_{C2} as calculated from equation 12. The v_{C1} and v_{C2} show analogue chaotic. The v_{C1}-v_{C2} attractor is shown in Fig. 1.6. It shows the phase plane of v_{C1} versus v_{C2} as calculated from equation 12.

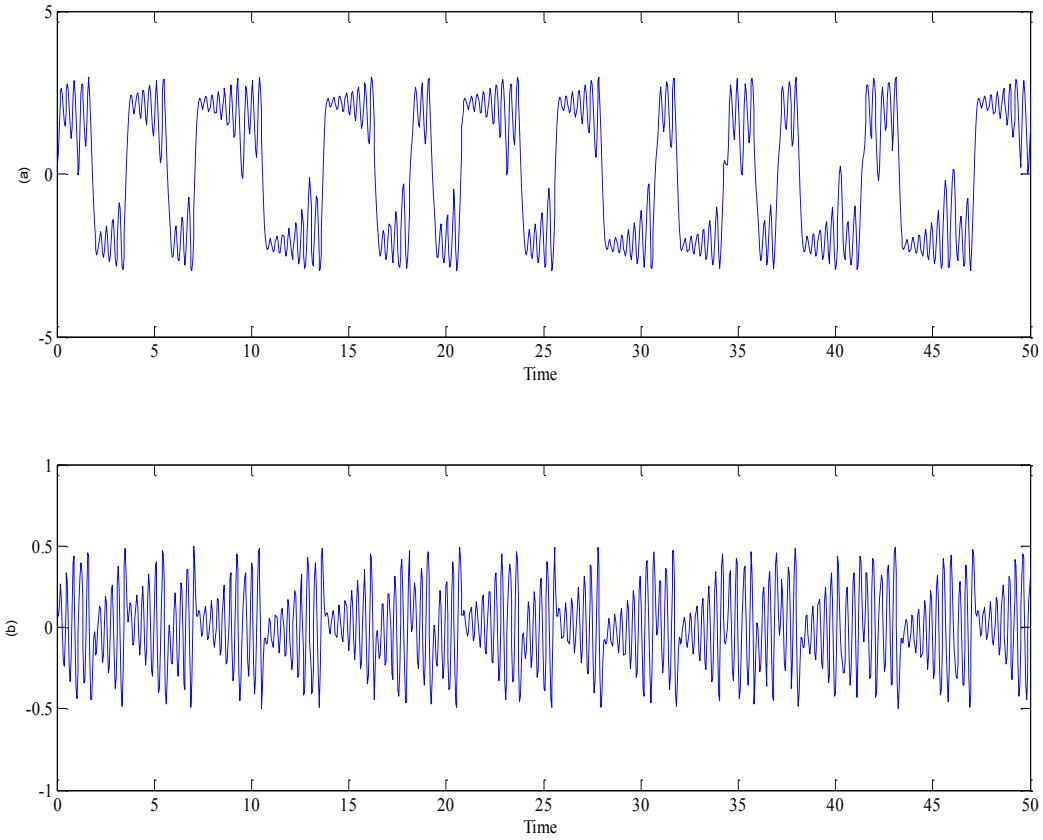

Fig. 1.5. Simulated signals of the Chua circuit. (a) v_{C1} Signal and (b) v_{C2} signal.

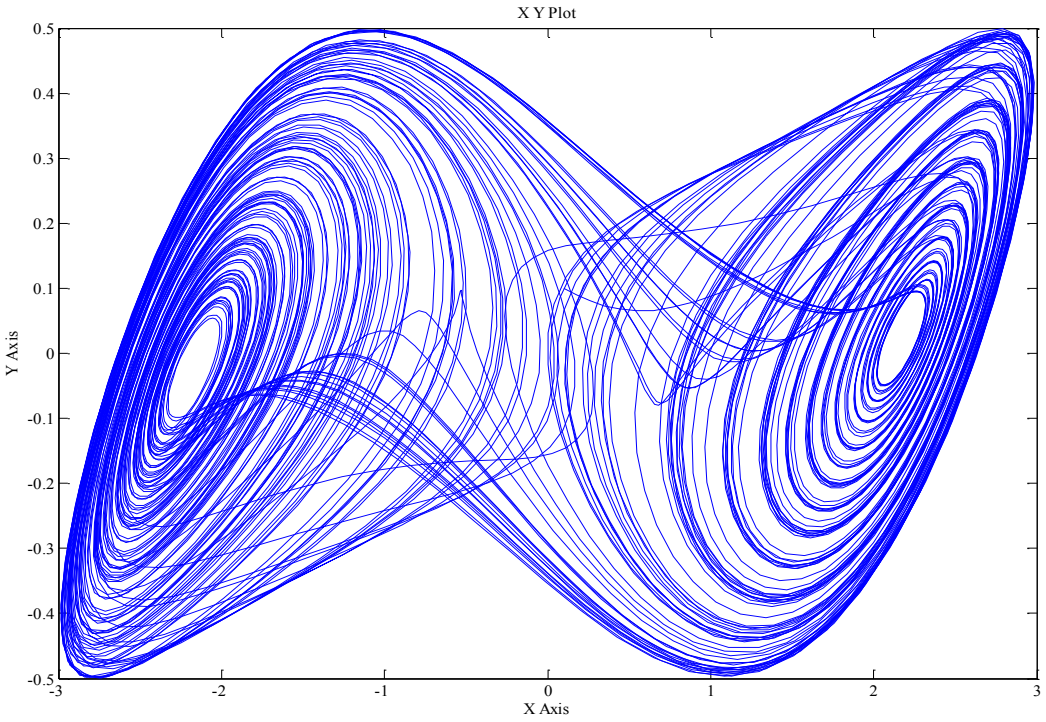

Fig. 1.6. The v_{C1}-v_{C2} attractor.

The Chua circuit is simple and has a high auto-correlation, low cross correlation and almost random signals. However the need to model the non-linear function adds an unnecessary complexity if the system has to be implemented in digital hardware.

1.7.2 The Rössler System

The Rössler system is described by the following state equations which are written in differential equation form.

$$\dot{x} = -y - z \quad \text{(a)}$$
$$\dot{y} = x + ay \quad \text{(b)} \qquad (13)$$
$$\dot{z} = bx - cz + xz \quad \text{(c)}$$

In this system the non-linearity is represented by a multiplier which is easy to implement in digital hardware. The SIMULINK model of the Rössler is shown in Fig. 1.7. The *x* and *y* signals are shown in Fig. 1.8. It shows *x y* as calculated from equation 13. The *x*-*y* attractor shows in Fig. 1.9. It shows the phase plane of *x* versus *y* as calculated from equation 13. The attractor *x y* shows one is single scroll.

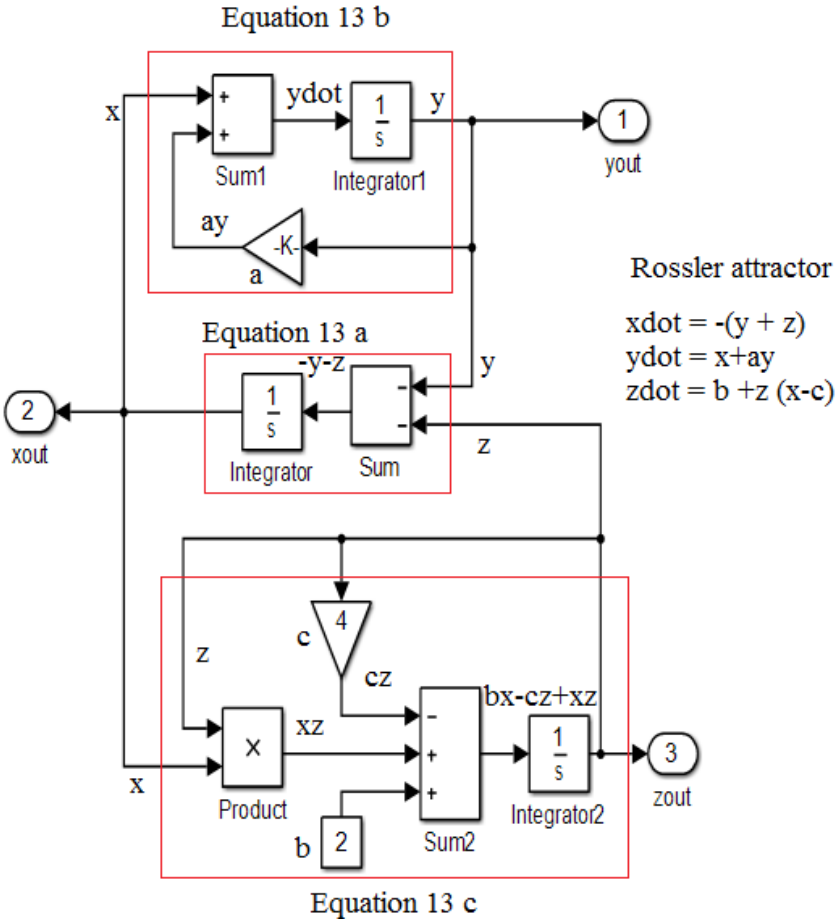

Fig. 1.7. SIMULINK model of The Rössler.

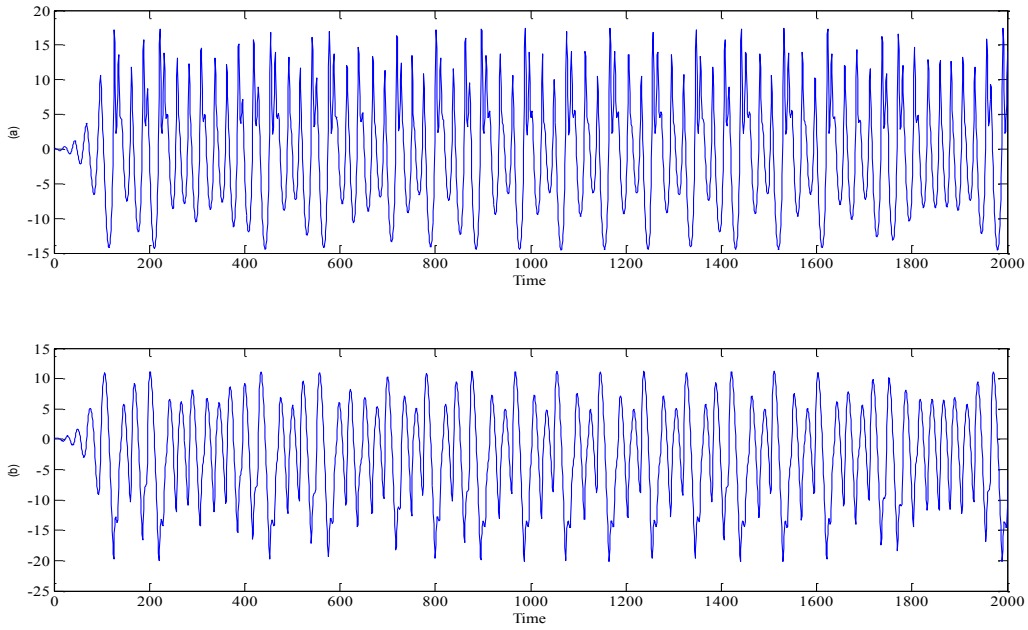

Fig. 1.8. The Rössler system, (a) The *x* signal and (b) The *y* signal.

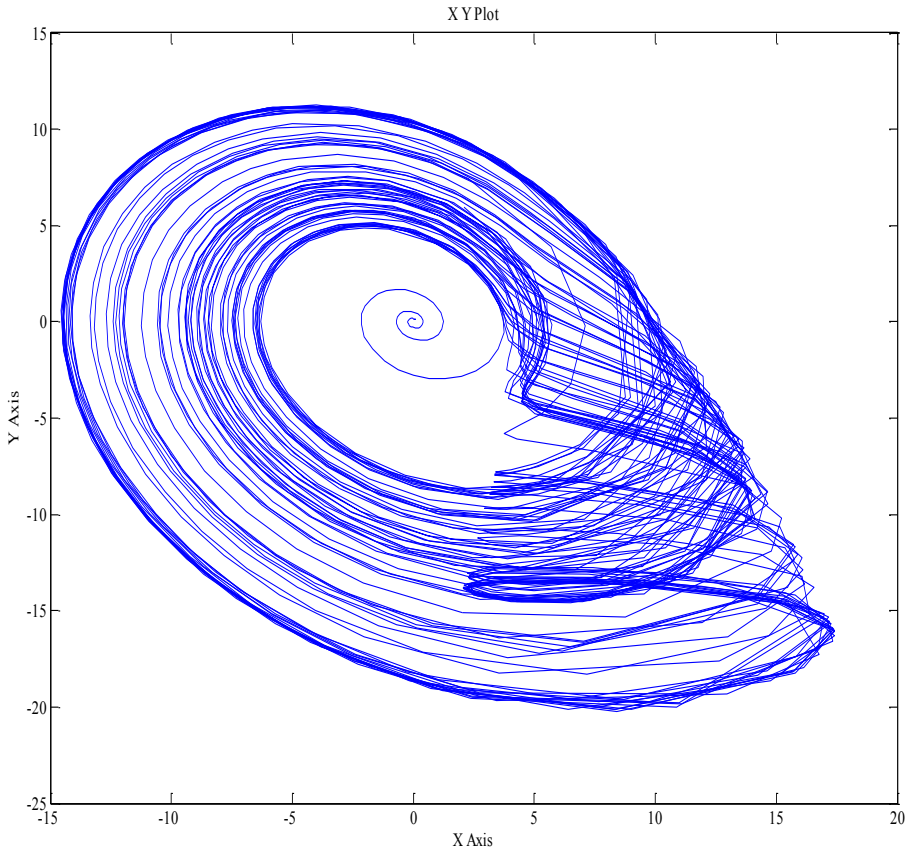

Fig. 1.9. The *x*-*y* attractor of the Rössler system.

The only draw-back of the Rössler system is that the attractor is a single loop which results in a higher cross correlation between the state variables. This affects the performance on an encryption system as will be explained in later chapters.

1.7.3 The Lorenz system

The Lorenz system is described by the following state equations which are written in differential equation form.

$$\dot{x} = \mathrm{A}\,(y - x)$$

$$\dot{y} = \mathrm{B}\,x - y - x\,z \tag{14}$$

$$\dot{z} = x\,y - \mathrm{C}\,z$$

Fig. 1.10 shows the SIMULINK Lorenz model where A, B and C are system parameters. x, y and z are state variables The scaling factors S_1, S_2 and S_3 are used to control the output signals frequency band and they are also part of the key in the encryption system. The x and y signals are shown in Fig. 1.11 and the x-y attractor in Fig. 1.12.

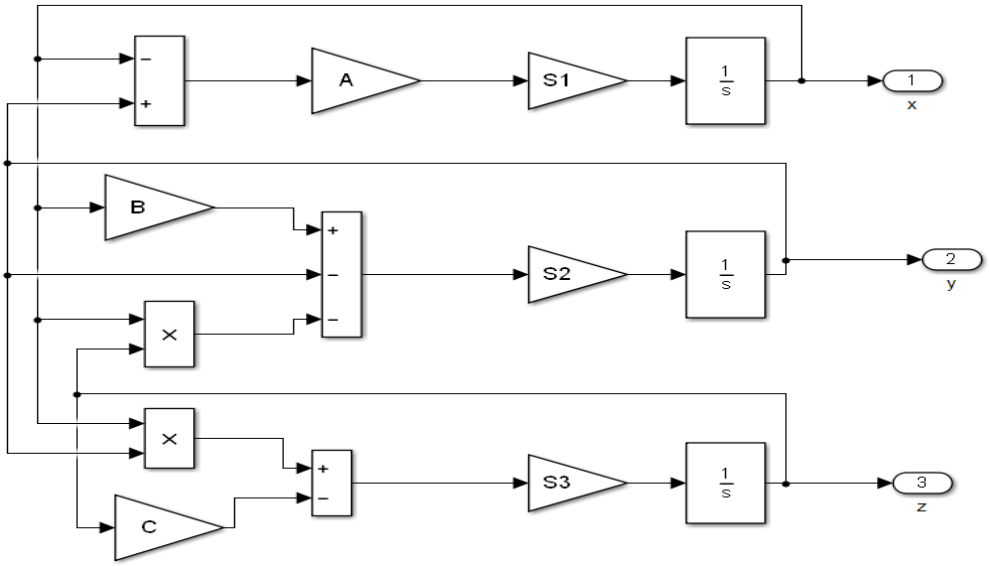

Fig. 1.10. SIMULINK model of The Lorenz system.

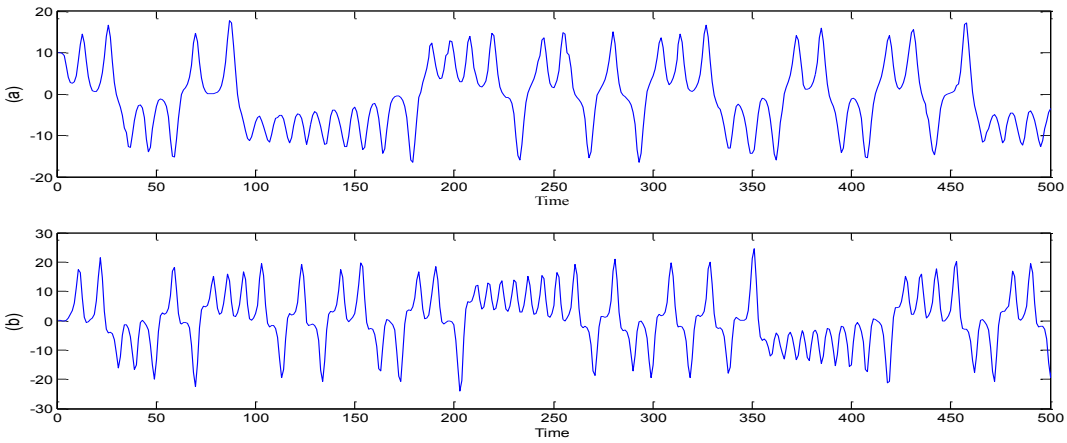

Fig. 1.11. The simulated signals of the Lorenz System. (a) x signal and (b) y signal.

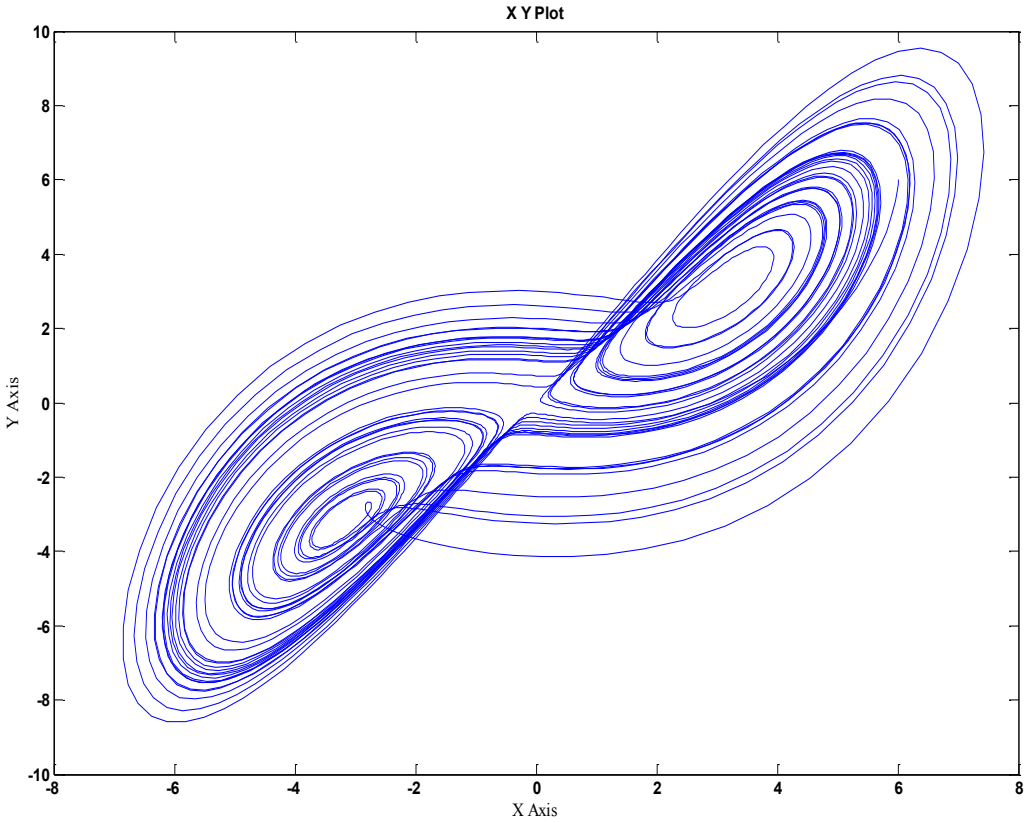

Fig. 1.12. The *x*-*y* attractor of the Lorenz system.

From the above analysis of the Chua, Rössler and Lorenz systems, we have chosen the Lorenz system as it is easier to implement using digital hardware and has a double loop attractor.

From Fig. 1.6, the Chua attractor has double scroll. It has high auto-correlation and high cross-correlation. However, we avoided to use the Chua's system because the need to model the non-linear function adds an unnecessary complexity if the system has to be implemented in digital hardware. From Fig. 1.9, the Rössler system a single scroll. Thus, we avoided to use it because of the cross-correlation between state variables are high. This affects the performance of the system. From Fig. 1.12, the Lorenz system has double scroll. The auto-correlation is high and the cross-correlation is low.

1.8 Communication Systems Overview

There are three main parts to any communication system: the transmitter, channel and receiver. In the transmitter, spread-spectrum technology has been used to encrypt the user data by using a chaotic signal. Before the encrypted user data is transmitted through the channel, it is encoded using a Manchester encoder for clock recovery. In the receiver, synchronisation is necessary for it to be able to retrieve the original message. Fig. 1.13 shows the basic block diagram of the communication system working at baseband frequencies. The synchronisation between two transmitters and a receiver is required to retrieve the information signals. Therefore, clock-recovery is used to recover the transmitter's clock and lock the desired frequency using a Phase Locked Loop (PLL).

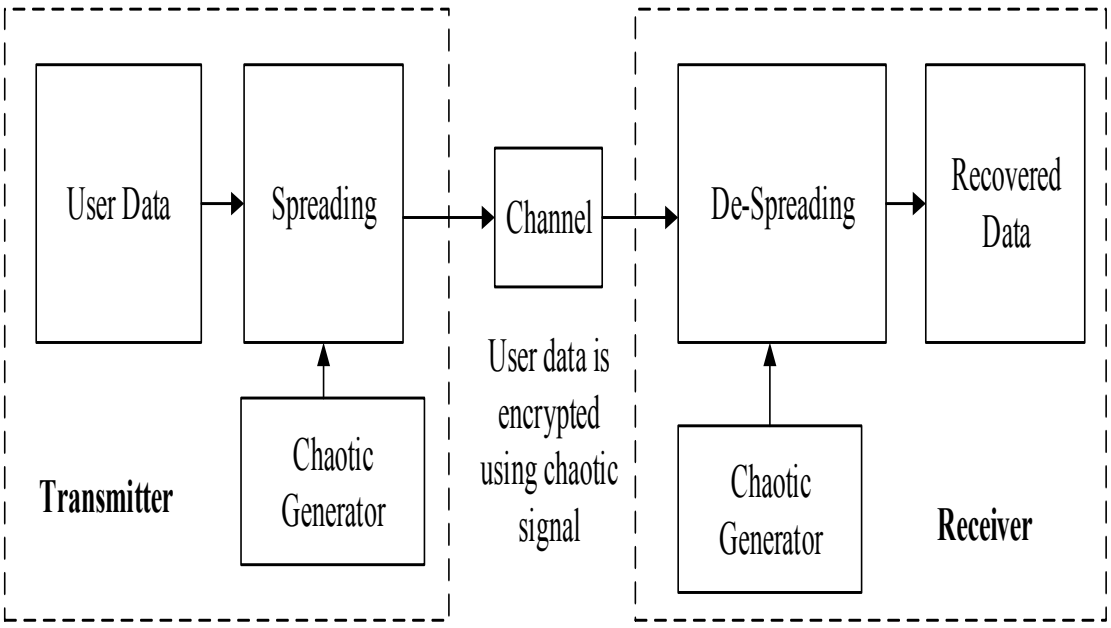

Fig. 1.13. Block Diagram of the Baseband System.

1.9 Proposed Communication System

In this research, we will demonstrate a digital communication system with high immunity and security based on a Lorenz chaotic signal. Therefore, the motivation of this research is to establish a new cryptosystem for secure data transmission. The system uses a stream cipher in which the encryption key is continuously varying. For increased security, one or more of the parameters of the Lorenz generator are

controlled by an auxiliary chaotic generator, which is robust to various known attack methods like brute force. A list of the original contributions from this research follows.

1.10 Original Contributions

The original contributions produced from this research are outlined below.

I. We propose a digital communication system with high immunity and security based on the digitization of a Lorenz chaotic stream cipher. This technique was built on digitizing two Lorenz chaotic models to increase the security level. Spread-spectrum technology was used for user data spreading. The scrambling model was developed for Lorenz generators. The binary stream of Lorenz generators are passed randomness test. The security analysis of the cryptosystem is described and also compared with the other systems such as one time pad, DES, AES, blowfish. The communication system was designed, and we obtained simulation results, as well as performance results and the bit error rate (BER) of the system on a noisy channel. These are described in chapter 3.

II. The clock and data recovery and synchronisation of chaotic signals in secure CDMA communication systems are described in chapter 4. The design methodology of the system is described. Clock recovery was designed and tested with the SIMULINK. The clock recovery design is then converted to System Generator® blocks. The clock recovery is validated in real time implementation that shows the desired frequency was recovered and locked. Hardware results are described in chapter 4. Data recovery was also tested in both simulation and real-time implementation, the results of which are described in chapter 4. Moreover, we discuss and detail a practical system for synchronising two chaotic generators used in the digital Code Division Multiple Access (CDMA). Synchronisation is achieved and maintained at the receiver.

III. An FPGA implementation of a chaotic signals for a block-spreading communication system is described in chapter 5. A detailed explanation of the implementation for transmitter and receiver systems are presented in chapter 5. The real time results are obtained.

IV. An FPGA implementation of stream cipher based on Lorenz generators is described in chapter 6. A detailed explanation of the implementation for transmitter and receiver systems are presented in chapter 6. The real time results are obtained. Moreover, a Lorenz chaotic signal was implemented and chaotic signal attractors were visualized in an oscilloscope, employing the Spartan 6 boards and a Pmod4 SPI protocol. The randomness test results of the Lorenz chaotic model output were obtained.

1.11 List of Publications

[1] Ahmed Alshammari, Mohamed. I. Sobhy and Peter Lee, "Secure Digital Communication Based on Lorenz Stream Cipher," SOCC 2017, "Munich, September 2017.

[2] Ahmed Alshammari, Mohamed. I. Sobhy and Peter Lee, "Synchronisation of Chaotic Signals in Secure CDMA Communication Systems," submitted to the PEMWN, Paris, November 2017.

[3] Ahmed Alshammari, Mohamed. I. Sobhy and Peter Lee, "Digital Communication System With High Security and High Noise Immunity: Security Analysis and Simulation," submitted to the BWCCA, Barcelona, November 2017.

[4] Digital Communication Systems with High Security and Jamming Immunity using Stream Cipher and Multiple Chaotic Systems.
Under preparation. To be submitted to The International Journal of Bifurcation and Chaos.

[5] Ahmed Alshammari, Mohamed. I. Sobhy and Peter Lee, "FPGA Implementation of Secure Communication System using Chaotic Signal," School Research Conference 2015, Poster, School of Enginerring and Digital Arts, University of Kent.

1.12 Organization of the Thesis

The thesis is divided into seven chapters. In chapter 1, we provide an introduction, in which we discuss block ciphers, stream ciphers, the properties of chaos in

communication system applications and chaos for spread-spectrum technology. Finally, a communication system overview is given.

In chapter 2, a literature review is provided.

In chapter 3, we present a secure digital communication system based on a Lorenz stream cipher. Following this, we evaluate the cryptosystem in terms of key length and key space, parameters sensitivity and randomness test. We outline the communication system performance in the presence of noise based on simulation results.

In chapter 4, clock and data recovery and synchronization of two chaotic signal in secure communication systems is presented. We outline the clock recovery method and SIMULINK results are obtained. Real time results are also obtained. a practical system for synchronising two chaotic generators used in a digital CDMA is described. We also explain the synchronisation technique, based on a sync/stream subsystem. The proposed technique was validated experimentally, and the practical results demonstrate the robustness of the system. Hardware results are detailed, as well as observations of any disadvantages of the synchronisation technique.

In chapter 5, a digital communication system for four users based on a Lorenz DS-CDMA, listing a description of the system and the real-time results.

Chapter 6 presents a digital communication system with high security based on a Lorenz stream cipher. The system was implemented using two separate Spartan 6 FPGA boards. We demonstrate a new approach for user data encryption using two Lorenz chaotic systems in which the encryption key varies continuously. A detailed description of this approach is provided. User data recovery results based on simulation and real-time results (implemented on FPGA boards) are also provided.

Finally, conclusion and future works are given in chapter 7.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

The modern science of cryptography-based chaos, which is applied in communications systems, was introduced by Shannon [33-35]. In 1980, Chua [36] presented the implementation of a practical chaotic circuit. The properties of chaotic systems, such as their aperiodic nature, make the long term prediction of their trajectories impossible. Additionally, these systems are very sensitive to even the slightest deviation from the initial conditions and parameter settings, which means that the chaos theory in cryptography meets Shannon's requirements of confusion and diffusion [37]. The chaos signal has proven to be well suited for applications in multi-user Spread-Spectrum (SS) communication systems [10, 21, 27, 31, 38-40], which can generate an infinite amount of uncorrelated chaotic signals. The chaotic system is also a deterministic system, which means the system is not random, and there are no stochastic input parameters. The strange behaviour of the chaotic response is due to the system's intrinsic non-linearity rather than noise [3] . Moreover, the chaotic signal is characterized by an inherent wideband, which makes it well-suited for SS communication [20, 41, 42]. The advantages of a communication system based on a chaotic signal include noise immunity, fading mitigation, multiple access capability and low probability of interception [12, 24, 43-45].

On the other hand, the cryptography-based traditional pseudorandom sequence generators, such as maximal length sequence (m-sequences), Walsh sequence and GOLD sequences are used in multi-user SS communication systems. However, there are several disadvantages such as a limited number of rounds (iterations) which is performed by encryption transformation. Also, a lack of correlation properties due to a limited number of available PN sequences. Moreover, when there is a delay, there is also poor cross-correlation of the Walsh codes properties, which causes poor performance in multipath environments [6-15]. From the perspective of security, these

issues degrade the system, which leads to the need for research to identify alternative, robust techniques.

In coherent chaos-based communication system schemes, such as chaos–based, direct sequence code division multiple access (DS-CDMA), the chaotic signal is used as a carrier for user data, and the chaotic synchronization at the receiver side is necessary in order to generate a symmetric chaotic sequence at the transmitter side and demodulate the transmitted sequence [46-48]. Previous research [30] presents a theoretical simulation of a communication system based on chaotic signal when the receiver is synchronized and when the receiver is not synchronized. The author in [27] concluded that provide significant advantages in terms of noise performance and data rate compared to non- coherent detection. However, these advantages are present only if the synchronization is well-maintained. However, the author stated that when the synchronization at the receiver is not well-maintained due to poor conditions, the non-coherent detection results in greater robustness and less complexity than in coherent detection. In addition, the coherent chaos-based communication system enhanced performance, security and synchronization robustness.

2.2 Coherent Chaos Modulation Schemes

Using chaotic systems to establish a secure communication system shows that there are four generations of schemes, the first three of which are known as continuous chaotic synchronization. The first generation includes additive chaos masking, an analogue modulation scheme, and chaotic shift keying (CSK), a digital modulation scheme. The additive chaos masking scheme consists of one chaotic system at the transmitter and an identical chaotic system at the receiver. The information message is encoded at the transmitter by adding it into the chaotic signal [27, 49, 50]. The CSK scheme consists of two chaotic generators with the same structure and different parameters. The information signal is in binary form, which is used to switch transmitted signals between chaotic systems. More specifically, the two different generators are used to encode bit 0 and bit 1 of the information signal [27, 29, 30, 49]. The second generation is known as chaotic modulation, which is coherent analogue modulation. It consists of two techniques to modulate message signals using a chaotic

signal. The first technique is a chaotic parameter modulation, and the second technique is a chaotic non-autonomous modulation [49]. The chaotic parameter modulation technique uses the information signal to modulate some parameters of the chaotic system at the transmitter, whereas the chaotic non-autonomous technique uses the information signal to disturb the chaotic attractor directly in its phase space [49, 51].

Another study [52] presented a secure communications system based on chaotic switching, which is a form of chaotic parameter modulation. The information signal, a binary sequence, is used a modulator when there are one or more parameters of the chaotic switching transmitter. The third generation was proposed to enhance the system security provided by the first two generations. The chaotic cryptosystem is a technique in which the plain text signal is encrypted using an encryption rule based on a key signal that is generated using a chaotic system. The scrambled signal is then used to drive the chaotic system in order to continually change the system dynamic [49].

2.3 Synchronization Methods

Although the advantages of the chaotic system make it a good candidate for providing data transmission protection from unwanted people, the synchronization between two chaotic generators is the primary issue and is not easily implemented in real application systems. Synchronization means that the two chaotic systems must generate the same output signals at the same time to retrieve the transmitted information. The chaotic signals are broadband in nature, and the frequency is unknown, the synchronization needs to be captured using different methods.

Over the last two decades, several methods have been used to perform synchronization, such as impulsive synchronization and adaptive synchronization [3]. The first three generations of communications systems-based chaotic systems, which are known as continuous synchronization, include additive chaos masking, chaotic modulation and chaotic cryptosystem. These types of synchronization methods are used to transmit the drive signal and are injected onto the dynamics of the chaotic system at the salve system (receiver). On the other hand, impulsive synchronization, which is known as fourth generation synchronization, uses a drive signal at the master (transmitter) system to control the chaotic system at the receiver. The drive signal is

not continuous, but is sent in the form of impulses. The disadvantages of using the third generation synchronization techniques are the high consumption of bandwidth due to the continual injection of the synchronization framework into the salve (receiver) system, which affects the efficiency of channel usage [53].

Generally, fourth generation synchronization is believed to satisfactorily address security concerns by making the transmitted signal more complex and reducing the redundancy in the transmitted signal. This technique combines a cryptographic scheme and a low dimensional chaotic system to satisfy signal complexity. It also uses an impulsive synchronization method to reduce redundancy at the transmitter. By using this technique, the synchronization between the transmitter and receiver will consume much less bandwidth than the continuous synchronization [49]. On the other hand, other research [53] indicates that impulsive and continuous synchronization are not reliable for chaotic communications systems, because they both inject the drive signal into the salve system, which does not provide a robust channel noise sensibility. An alternative technique is based on an asynchronous serial communication protocol to synchronize the slave system chaotic generator without the need for the drive signal to be injected into the dynamics of slave system. In this case, the drive signal is not affected by any perturbations caused by the channel noise sensitivity[53].

In another study [54], the synchronization scheme was divided into two encryption schemes, block cipher and stream cipher. The block cipher is a static transformation that operates on the plaintext segment, whereas the stream cipher is configured as a dynamic system with memory and a combining function, such as a cipher feedback mode (CFB) and output feedback mode (OFB). Other research examined a real-time FPGA implementation of the synchronization of two Lorenz chaotic generators [55]. This approach focuses on improving security where the master (transmitter chaotic system) and salve (receiver system) are implemented at two separate FPGA boards.

Another proposed system uses the ZigBee protocol to overcome the issue of chaotic synchronization sensibility in the presence of noise and has been tested and validated in hardware [56]. The principle of the proposed scheme is to trigger the salve chaotic generator each time received data is detected, and then synchronize the driving signal with the chaotic generator at the receiver for de-spreading the transmitted data. The main purpose of this synchronization scheme is to avoid disrupting the chaotic

generator dynamics at the receiver. Thus, the chaotic generator at the receiver side is not affected by the presence of noise in the channel and can generate a signal identical to that generated by the chaotic generator at the transmitter side. In this example, a wireless communication system based on a hyper-chaotic system is implemented on FPGA boards.

Other research proposed a secure communication system based on a Lorenz chaotic system in which data were encrypted using a parameter modulation method [57] . The novelty of the proposed system was its ability to overcome the received signal that was contaminated by the presence of strong noise (white Gaussian noise) in the channel. The received chaotic signal was extracted from the noise with a filter using a gradient algorithm and Independent Component Analysis (ICA). A computer simulation verified the proposed system.

2.4 Chaos-based Spread Spectrum

A spread spectrum signal is nearly impossible to jam unless the spreading pattern is known and there is a low probability of interception. Regarding commercial communication system advantages, the SS system spectrum provides resilience to fading and interference injection, and allows multiple users to use the same set of frequencies [26, 58-62]. The SS scheme qualifies as a broadband system, because the information signal is spread across a very large bandwidth rather than that of the data bandwidth [10]. There are many SS communication schemes in which the DS-SS scheme uses orthogonal or nearly orthogonal spreading sequences for the user's multiplexing [63]. Moreover, having good cross correlation and autocorrelation values are important in spreading sequence, as they help overcome the multiple access interference (MAI) and assist with multipath performance [9].

The first time investigation of the chaotic DSSS system data spreading using multiplication of the data bits with discrete chaotic signal are presented in [9, 64] . The performance of the chaos based DSSS system with multiple - access had analysis using bit error rate (BER) in presence of noise and fading channel are presented in [65, 66]. The system's performance was studied using chaotic complex spreading sequence of DS/CDMA is presented in [10, 23].

Previous research tested a chaos-based digital communication system scheme for multiple access [67]. The study was conducted under the assumption that chaotic reference signals are transmitted and then followed by an information bearing signal. The binary training sequence is sent periodically and used for modulation. Additionally, the binary sequence is transmitted and used to modulate the same chaotic signal. In this scheme, multiple access is provided by using different chaotic signals, as well as training sequences, which are assigned to different users.

Another study involved designing and analysing a new phase-coded spread spectrum communication system based on a chaotic generator. At the transmitter, the information encryption scheme is based on a spread spectrum technique by directly multiplying the variable duration bit by the phase code carrier. The transmitted data are recovered at the receiver by using a coherent receiver that relies on a direct correlator. The study also included a theoretical investigation of BEP in the presence of the AWGN for a single user, as well as the multiusers systems. System performance was also studied in term of the BERs, which show that it completely compared to BEPs theoretical analysis [68].

In another study [13], the researchers investigated the physical layer security of a chaos-shift keying (CSK) modulation scheme and a differential chaotic shift-keying (DCSK) in the presence of channel noise by using AWGN and a Rayleigh fading channel. The findings demonstrate the average secrecy capacity and outage probability by computing and analysing the variation of the bit energy coming from the chaotic signal usage of information encoding. The authors concluded that the CSK modulation performs better than DCSK and SS-BPSK system for physical layer security. Other research has concentrated on SS communication based on a chaotic signal and presented various methods and schemes of digital chaos communication in terms of chaotic modulation and demodulation, as well as channel encoding [69].

The design of the chaotic spreading sequence has been discussed and evaluated in many papers. Two schemes have been used to modulate the transmitted bits. The first scheme uses the real value of the chaotic signal, whereas the second scheme uses the quantized chaotic signal for user data modulation [27]. CSK and DS-CDMA are used in the same reception method, in which the de-spread of the received data is based on a replica of the chaotic signal at the receiver side [27].

Many papers have also studied and analysed the correlation properties and synchronization performance of using quantized chaotic signals for user data modulation [23, 70-72]. Some studies [73-75] show that there are different schemes for using new binary chaos based sequence. In other studies, transmitted bit mapping based on the real values of chaotic signals for different schemes were studied and analysed [38, 39, 72, 76], and the correlation properties were studied and presented in [76, 77].

The findings of one study [27] showed that the quantified chaotic sequences performed better than the real value chaotic signals in terms of bit error rate due to the sequence encoding used (+1 or -1), which makes the instantaneous energy of the signal constant. However, the properties of chaotic signals were affected, which makes the system less secure. Moreover, the quantization of the chaotic system in real time must take security issues into consideration [27]. Still, the application of a coherent chaos-based communication system is not yet sufficient for application in wireless systems, due to synchronisation issues. However, there are different works archived in this area, which is a part of a communication system [27].

Other research [78] demonstrated a chaotic generator in which the output sequence is truly a random number and of low complexity. The results indicated that the chaotic signals have a better Low Probability of Interception (LPI) performance than the PN signals [12]. For example, one study [79] examined the performance of a direct-sequence code division multiple access (DS-CDMA) system based on various chaotic sequences, such as a logistic map, modified Bernoulli chaotic, logistic Bernoulli chaotic and Gold sequences, in terms of bit error rate (BER), cross correlation, Multiple Access Interference (MAI) and LPI. The researchers concluded that the logistic chaotic sequence generator with an XOR operation performs better than other chaotic sequence generators.

Another study [80] provided an exact analytical expression for the BER in a multiple access chaos-based digital communication system. The researchers used a chaos model to provide multiple access and test the system performance in the presence of the Additive White Gaussian Noise (AWGN). They concluded that based on the analytical expression and to optimise the BER, the chaotic sequence must have a low correlation for each user, and the bit energy should be always constant.

A cryptographic method based on chaotic encryption algorithm to generate sequences for random numbers which he claimed that the possibility of using it to be a replacement of the one-time pad system [81]. Later, Wheeler [49] concluded that there are problems in the chaotic encryption algorithm presented by Matthews, due to the production of repeating cycles of values, which affects the system security and is not suitable for cryptographic applications [81].

The new chaotic secure communication was presented in another study [82], which is primarily based on the impulse synchronization method. However, the main goal of the method was to improve the system's sensitivity by enlarging and observing minor parameter mismatches, where the Chua's circuit output sequence is used instead of a random key sequence. The secure concept used for the communication system is based on a one-time pad, which is called a magnifying glass. The researchers of another study [83] presented a chaos bit sequence for cryptographic applications based on a PN sequence generator. The chaos sequence was extracted based on a logistic map and Cubic maps. Then, statistical tests, randomness test and encryption performance of the chaotic binary sequence based on PN sequence were compared with an Linear Feed Back Shift Register (LFSR) based generator. The researchers concluded that the characteristics of the chaotic sequence were very close to the white noise sequence, whereas the chaos-based generator is secure and can be used in stream encryption applications. Similarly, other research [58] presented a secure system of the pseudo-random bit generator based stream cipher by using Coupled Chaotic Systems called CCS_PRBG.

Another study [84] proposed a digital communication system based on a robust hyper chaotic system (RHCS) to overcome the weakness identified by other research [85] in which they attacked the chaotic stream cipher by plotting the map and the output sequence, as the chaotic pattern of each single chaotic is unique, and to determine the easiest types of chaotic systems. The existing literature review includes three main types of systems to address these issues. In the first system, the initial condition must be made unpredictable by using two chaotic maps, one of which must be responsible for generating the initial conditions for the others. The second system uses two chaotic maps to switch between them at any time based on a predefined order or using a user-defined mechanism. The last secure system combines the first two systems. The

authors added three criteria for enhancing system security. First, the digital precision length must be long enough to be robust against a state enumeration attack. Second, in practical use, the parameter space must be large enough to prevent from being attack. Finally, it must be impossible to reconstruct the system using current computational technology. The proposed system RHCS for encryption and decryption is constructed to solve the issues mentioned above and satisfy these new criteria. The secure system is based on coupling the robust logistic chaotic map with another hidden map to increase the system's complexity compared to the traditional secure communication system. In addition, the system satisfies the large parameter space and, thus, the system precision increases. Hence, the system reconstruction is very hard based on statistical analysis by current computational technology.

Additional research [86] focused on the multiuser chaos based on DS-CDMA performance in the presence of AWGN, the Rayleigh fading channel and inter-user interference. The spreading sequence was based on a chaotic logistic map, which was assumed to generate a Pseudorandom Binary Sequence (PRBS). The synchronization technique was achieved by using code acquisition, and a code tracking phase was used to maintain the synchronization. In terms of BER, the Direct Sequence Code Division Multiple Access (DS-CDMA) chaotic communication system performance was robust to a noise effect of AWGN for 1-5 users. However, the system performance in the presence of Rayleigh fading was affected due to failure to satisfy the maximum allowable limit of the BER, which is 10^{-3}.

A similar study [73] examined the performance of a DS-CDMA system based on a 1D and 3D chaotic sequence generator in the presence of a flat fading channel, AWGN and user interference. The researchers used a three-dimensional chaotic system to generate a binary code to be used in a spread spectrum communication system. The spreading code used is 127 bit, and the communication system is assumed to be synchronized. In terms of BER, the system performance simulation test was a match for one dimensional chaotic and gold codes in regards to optimal codes and well-known spreading codes in modern digital communication system. Moreover, the proposed algorithm chaotic generator provided high transmission security and an infinite number of sequences.

Another study [87] addressed data encryption based on a stream cipher algorithm generator called Verna's cipher, in which the system complicity in terms of keystream was based on the multi-chaotic function. The cryptographic key was in 16, and there were five chaotic functions; thus, the key space is 10^{80}. A FIPS-140-2 test, which consists of four tests is used to test the randomness of the chaotic sequence generated, as well as the correlation test. The authors concluded that the chaotic algorithm can be used to encrypt different applications, such as text, image and multimedia files.

The researchers of another study [88] presented a cryptosystem based on a chaotic encryption algorithm by using an alternate of a stream cipher and a block cipher. The proposed algorithm was used as a masking sequence and an encryption mode based on three chaotic maps. The system performance analysis of the cryptosystem was evaluated using diffusion and confusion tests, a randomness test (NIST), a correlations test and encryption speed. In addition, the cryptanalysis was complete by testing the key space to insure the system is robust against attacking, such as brute force attack. Similarly, other research [89] proposed a modified chaotic cryptographic method from a chaotic cryptographic scheme, which was published previously [90]. The secure scheme is based on a logistic map where the distribution of the cipher-text and the encryption time can be controlled by a single parameter.

2.5 Related Surveys based on Real Time Implementations

The digital communication system based on chaos requires components of high accuracy. The accuracy of the parameters is important, and any mismatch between the transmitter and receiver will cause a synchronization problem, which means the receiver is unable to recover the transmitted data. A communication system based on an analogue circuit chaotic generators has practical difficulties because components vary with age and temperature [25, 91]. Synchronization sensibility and parameter mismatch issues are the primary concern in real time communication systems [92, 93]. In recent developments, most of the numerical generation of chaos is based on a field programmable gate array (FPGA), because it provides high accuracy, making it suitable for communication based chaos. FPGA architecture design is suitable for high performance DSP applications, especially in digital communication security, because

of parallel structures and arithmetic, which includes Multiply and Accumulate (MAC) and a variety of other arithmetic functions, such as fast Fourier transforms (FFTs), convolution and FIR design [28]. The implementation of chaotic generators based FPGA boards have been studied previously [94-96]. A discrete-time chaotic generator in real time is studied and evaluated in terms of power consumption, maximum execution frequency and resource usage based on two FPGAs board [97].

One study [14] implemented a secure DS-CDMA spreading code by using a digital multi-dimensional multi-scroll chaos based on Vertex 4 FPGA. 512 uncorrelated output streams were constructed by concatenated, low significant bit from each dimension to pass a statistical test suite for random and pseudorandom numbers for cryptographic application (NIST). The performance of the DS-CDMA system was tested in the presence of the AWGN and multipath, which is equivalent to gold codes. The system implementation in FPGA shows that the throughput is up to 10.92Gbits/s.

Another study [15] involved an experiment based on low-cost DSP boards to show the MAI reduction for chaos-based DS-CDMA systems. They assumed that synchronization was achieved and that the multipath effects were negligible. The data recovery method at the receiver was achieved by multiplying the incoming signal with a synchronized replica spreading sequence of the user. To extract the information symbol, the correlation techniques were used by applying the integrate and dump stages. The theoretical and measured results of the bit error probability (P_{err}) for m-, Gold and chaos-based sequences were provided, which indicated that the chaos-based spreading sequence performed better compared to others. Additionally, the P_{err} demonstrated good agreement between the measured result and the theoretical predication.

In other research [98], binary sequences were constructed from Chua's circuit as a way to generate a pseudo random number sequence (PRNS), which satisfied the cryptography requirement. The proposed method was based on taking only a fraction of each signal from the three Chau's output states, then assembling the extracted parts to build one binary sequence. The generated binary sequences passed the randomness test (NIST). Moreover, the chaotic generator based on Chua's circuit was implemented in hardware using a FPGA board.

The authors of another study [99] presented a stream cipher algorithm based on a chaos system in order to increase the system degree complexity in terms of chaotic sequence properties. The proposed method of generation of the key stream depended on two logistic maps generators, one logistic map generator to generate a random number in a manner that satisfied some condition to replace the other logistic map generator parameters. The results based on the theoretical analysis and numerical test indicated that the method is suitable for a stream cipher with high efficiency. In addition, the proposed system was implemented in real time using VLSI. Similarly, other research [37] presented a stream cipher based on a modified logistic map, which satisfied the higher confusion compared to a logistic map and a flatter distribution for different parameters in the bifurcation diagram. The proposed stream cipher was implemented in Spartan 6 FPGA boards. The stream cipher scheme worked on a 93MHz clock frequency and provided 16-bit encrypted data per clock cycle, which gave the throughput of 1.5 Gbps.

In another study [100], the researchers proposed and implemented a hardware implementation of a new additive hyper chaos masking (AHM) algorithm for secure digital chaotic communications by using a FPGA board. They used a hyper chaotic Lorenz system as the keys generator and mixed the information signal with the hyper chaotic signal using an XOR operator. This operation was done before the additive masking operation to increase the security level. Several intercept receiver techniques, such as an energy detector, synchronous and asynchronous structures and coherent and non-coherent structures have been used to investigate the performance detection of chaotic spreading signal in terms of low probability intercept (LPI) and to compare it with the binary sequences.

Other research [101] focused on a discrete wheel-switching chaotic system in which the output sequence of the chaotic generator was changed based on pre-configured chaotic maps. They findings indicate that the chaotic sequence of the proposed generator passed all NIST 800-22 statistical randomness tests. The system was implemented on a FPGA board.

Chapter 3

DIGITAL COMMUNICATION SYSTEM WITH HIGH SECURITY AND HIGH NOISE IMMUNITY: SECURITY ANALYSIS AND SIMULATION

3.1 The Bases of Security Analysis

Cryptography and cryptanalysis are two primary techniques for facilitating secure communication. Cryptography is the art to build of a secure system to prevent the transmitted data (plaintext, the key, or both) to be intercepted from unauthorised people, which is known as a cryptosystem. Cryptanalysis is the process used to evaluate the cryptosystem. Furthermore, it is used to recover the data transmitted from unauthorised people by finding weaknesses in a cryptosystem. Successful cryptanalysis can retrieve plaintext or the key. The primary goal of the cryptosystem is to hide the data transmitted from unauthorised people by using a secret key. When the original data is mapped to another form using a cipher, the technique is known as data encryption, whereas the reverse operation of using the same cipher to recover the original transmitted data is known as the decryption. The attempted cryptanalysis is called the attack. When the cryptanalyst already has a knowledge of the cryptosystem, there are different types of attacks used to break the cryptosystem, such as the cipher text-only attack, known-plaintext attack, chosen attack and brute force attack.

The cryptosystem splits into four major components. Plaintext is the original message for transmitting. The cryptographic is responsible for encryption and decryption. The cipher-text is the output sequence of the encryption operation and the key that is shared between two systems to encrypt and decrypt the original message. The cryptanalysis is one of the important steps for evaluating components of the new cryptosystem, such as security and reliability [102].

There are two types of cryptosystems, symmetric (private) key and asymmetric (public) key. The symmetric key cryptosystem uses the same private key in both the transmitter and receiver to quickly encrypt and decrypt the plaintext, which is suitable

for applications that require a high data rate, such as video encryption. Furthermore, the symmetric key is divided into two types, block cipher and stream cipher. The block cipher always encrypts and decrypts the plaintext in the same way by using a fixed binary key. The block cipher is widely used in applications, such as triple Data Encryption Standard (DES), Advanced Encryption Standard (AES) and Rivest Cipher (RC5). On the other hand, the stream cipher is generated using a random binary stream, which is used as a secret key that is mixed with plaintext to encrypt it to output, which is known as a keystream. The stream cipher key length is based on cryptosystem features, which could range from 32 to 256 bits. The most popular used stream cipher cryptosystems are Encryption Algorithm (A5/2), RC4 and Software-Optimised Encryption Algorithm (SEAL).

The cryptosystem is asymmetric when it uses two different keys for encryption and decryption, where the first key is publicly distributed and the second key is private. This kind of cryptosystem is primarily used for a small amount of data, such as authentication, secret key agreement and digital signature, because it is slower due to the arithmetic operation with large integers. The public key length could range from 1024 to 4096 bits. A widely used public key algorithm is the Rivest-Shamir-Adleman (RSA) [9].

In terms of security and systematic performance, it is not easy to evaluate a new cryptosystem, such as a communication system based on a chaotic system. However, there are guidelines for designers and researchers to enhance the system's robustness and security, which has major cryptographic requirements for building up a new chaos-based cryptosystem and analysis, which is the most important component of the cryptosystem. The major constituting cryptographic requirements and analysis for chaos-based cryptography are discussed in this chapter. Furthermore, the system should pass the tests for confusion and diffusion, randomness of bit stream sequence, encryption speed and sensitivity of mismatched key [88, 103] [104].

There are conditions, if satisfied, at which we might say that the algorithm is probably safe [19]. These conditions are:

- When the time required to break the algorithm is longer than the time which the encrypted information must kept secret.

- When the cost required to break the algorithm is greater than the value of the encrypted information.
- When the amount of data encrypted using one key is less than the amount of data required to break the algorithm.

Unconditionally secure and computationally secure are two terms used to describe the algorithm. When we say that the algorithm is unconditionally secure, the cryptanalyst does not have complete information to retrieve the plaintext, no matter how much of the cipher text is present. When we say that the algorithm is a computationally secure, it is hard to break with the available resource or even future resources [19].

The categories of breaking an algorithm are as follows [105]:

- Total break: the cryptanalyst finds the key.
- Global deduction: the cryptanalyst determines a different algorithm that decrypts the cipher text without knowing the key.
- Local deduction: the cryptanalyst extracts a plaintext from the received cipher text.
- Information deduction: the cryptanalyst finds some information from the key or plaintext.

3.1.1 The key length

One of the fundamental factors of the cryptosystem security is the key. The key length is the number of the bit used as a key. Also, the key should be strong enough to counter any super computational scan, such as a brute force attack. In contrast, if the key space of the cryptosystem is small, it will be easily broken. It is important to specify the key used for encryption and decryption, such as when the key of a chaotic system is made from system parameters and initial conditions. However, it should be mentioned that the parameters and precision can vary.

3.1.2 The key space

The key space of a cryptosystem is the total number of possible keys. The key space size of the cryptosystem should be defined in order to evaluate the system's security. It is calculated by all the possible valid keys. Furthermore, the number of the encryption and decryption key pairs in the cipher system is stated as the size of the key space.

3.1.3 Confusion and diffusion

Apart from satisfying the need for a long key space, a good cryptosystem should have enough sensitivity so that one bit change in the key leads to a completely different cipher text. Additionally, one bit change in the plaintext leads to a completely different cipher text. These two properties represent the diffusion. Moreover, any pattern in the plaintext should not appear in the cipher text. This property represents confusion [103].

3.1.4 Brute force attack

The brute force attack is one of the attack types which is used to evaluate the new cryptosystem. The brute force attack is used to try every possible key in order to break the cipher. This type of attack method is completely based on the machine powerful such as computation processing speed. When the key space is small, the brute force attack to break the cipher is faster. The recommended key space is of size $> 2^{100}$, and any cryptosystem key space size below that is considered to be insecure [104].

3.1.5 Binary sequence randomness test of the encryption generator

The cryptosystem's main task is to encrypt the plaintext based on a random bit sequence generator. The plaintext should be completely hidden in the cipher-text. The bit sequence generator should be unpredictable in order to ensure the plaintext is mapped in a secure manner. A convincing way to evaluate the quality of the binary stream, is performed using NIST SP800-22 test that was published by the National Institute of Standards and Technology [106]. It is a statistical test for a random and pseudorandom number generators for cryptographic applications. It is the most well-known method used to test the randomness of a binary sequence. The test statistics is used to calculate a P-value which is the probability that the chosen statistics will assume values are equal to or worse than the observed test statistics values when considering the null hypothesis. If P-value ≥ 0.01 for each of the 13 tests, the test is considered to have passed, and if the sequence passes all 13 tests, then we can say that the sequence is cryptographically secured. The 13 tests are listed below:

1- Frequency (Monobit) test;
2- Frequency test within a block;
3- Runs test;
4- Test for the longest runs of ones in a block;
5- Binary matrix rank test;

6- Discrete Fourier transform (Spectral) test;
7- Nonoverlapping template matching test;
8- Overlapping template matching test;
9- Linear complexity test;
10- Serial test;
11- Approximate entropy test;
12- Cumulative sums (Cusum) test; and
13- Random excursions test.

3.2 Communication system with High Security and High Noise Immunity

Code Division Multiple Access (CDMA) technology allows many users to simultaneously use the same communication system and share the same frequency. In a CDMA scheme, each user is assigned a particular spreading sequence to map the information signal. Thus, the spreading information signal using a particular sequence increases the bandwidth of the information signal by a factor of N, which is known as the spreading factor or processing gain. The main feature of the spreading process is resistance to natural or artificial narrowband jamming. Furthermore, the information message can be easily hidden within the noise floor, preventing it from being detected by unwanted people. To recover the information signal, the receiver must know the spreading sequence to be eligible to retrieve the information message, which makes the information signal hard to intercept for unauthorised people. The CDMA does have limitations such as multiple access interference.

Pseudorandom scrambling based on traditional sequences is used for information privacy for CDMA systems, such as in the physical layer [107]. However, the traditional sequences are not secure [6-12]. In this chapter, we report the design a CDMA system based on a Lorenz stream cipher to enhance system security. The proposed cryptosystem for the CDMA system is compared to a one-time pad. One-time pad encryption is an unbreakable encryption method [108-110]. The encryption method of the one-time pad involves one truly random bit (Letter pad) corresponding to one bit of the plaintext by using the bitwise Exclusive OR gate (XOR) to produce one bit [108]. Fig. 3.1 shows the one-time pad scheme.

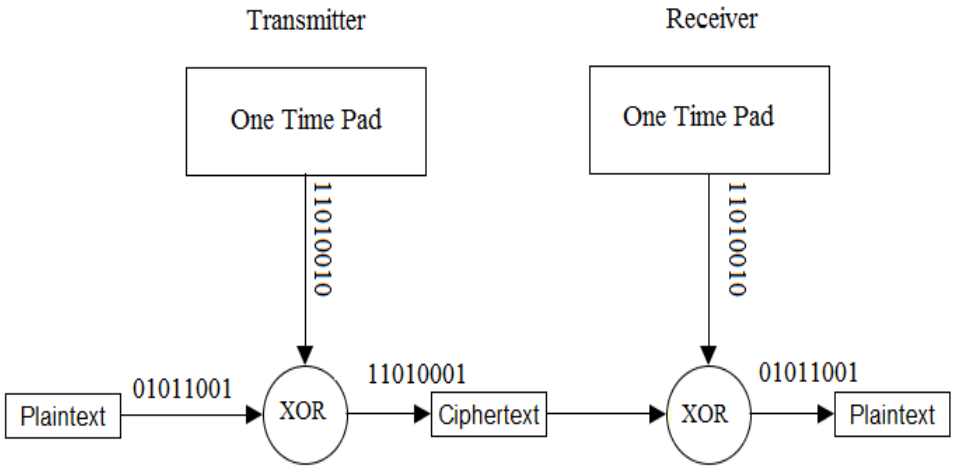

Fig. 3.1 Block diagram of the one-time pad scheme.

Any stream cipher can be unbreakable (one-time pad) if the following requirements are met [108]. The key and plaintext must be equal length, the key must satisfy a randomness test and the key must only be used once.

It is important to mention the difference between the encryption key of the one-time pad and chaotic system key. The encryption key of the one time-pad is always the same size as the plaintext being sent. Thus, in real applications, it is difficult to implement the system because of key distribution. This means that the key distribution must be kept secure. Moreover, the key has to be the same length as the message which is inconvenient or costly and can pose a security risk. In contrast, the system key of the chaotic system depends on system parameters and initial conditions and the key distribution easier and secure.

3.2.1 Comparison of the chaotic system cryptosystem with other cryptography systems

We want to compare our cryptosystem with existing symmetric key cryptography systems, such as DES, 3DES, AES, Blowfish and One-Time Pad. The size of the key space of cryptography system must be long enough to be protected from attacks. The larger the size of the key space, the longer the time needed for computation steps to perform a brute force attack, thus, the higher the security. The key is the information needed to recover the plaintext.

Data Encryption Standard (DES) was the first encryption standard to be published by NIST. DES uses a 56 bit key, a key size of 56 would provide 2^{56} key space. Triple

DES (3DES) was developed to extend the key size of the DES by applying the logarithm three times in succession with different keys. Thus, the key space is 2^{168}. Advance Encryption Standard (AES) is also a symmetric block cipher that uses 128, 192 or 256 bits. Blowfish is a symmetric block cipher uses variable key from 32 bits to 448 bits. The One-Time Pad (OTP) is unconditionally secure because of the truly random key stream that is used only once. Table 3.1 shows all five system properties.

Factors	Chaotic System	DES	3DES	Blowfish	AES	OTP
Key Length	576 bits	56 bits	168 bits	Varies between 32 bits to 448 bits	128,192 or 256 bits	Same as Length of the Plaintext (LP)
Cipher Type	Symmetric stream Cipher	Symmetric block Cipher	Symmetric block Cipher	Symmetric block Cipher	Symmetric block Cipher	Symmetric stream Cipher
Block Size	32 bits	64 bits	64 bits	64 bits	128,192 ,or 256	-
Key Space	2^{576}	2^{56}	2^{168}	2^{32}~2^{448}	2^{128}, 2^{192} or 2^{256}	2^{LP}
Security	Considered Secure	Not secure against brute force attack	Not Secure	Considered Secure	Considered Secure	Considered Secure

Table 3.1. Comparison between chaotic system, DES, 3DES, Blowfish, AES and one-time pad.

In this chapter, our approach is to provide a cryptosystem that can be compared to a One-Time Pad. The first Lorenz system is called the Main Lorenz Generator, and the second Lorenz system is called the Auxiliary Lorenz Generator. The aim of the Auxiliary Lorenz Generator is to make one of the Main Lorenz Generator parameters vary continually with time in a chaotic manner. Fig. 3.2 is a block diagram of the cryptosystem. One of the main Lorenz generator varies with time based on the Auxiliary Lorenz output signal. The plaintext is encrypted by the key stream generated

from the Main Lorenz generator using multiplication block. The cipher text is generated and transmitted to receiver side. The receiver system has an identical Auxiliary and Main Lorenz generator of the transmitter. The decryption process has started by multiplying the key stream with the cipher text to retrieve the plaintext.

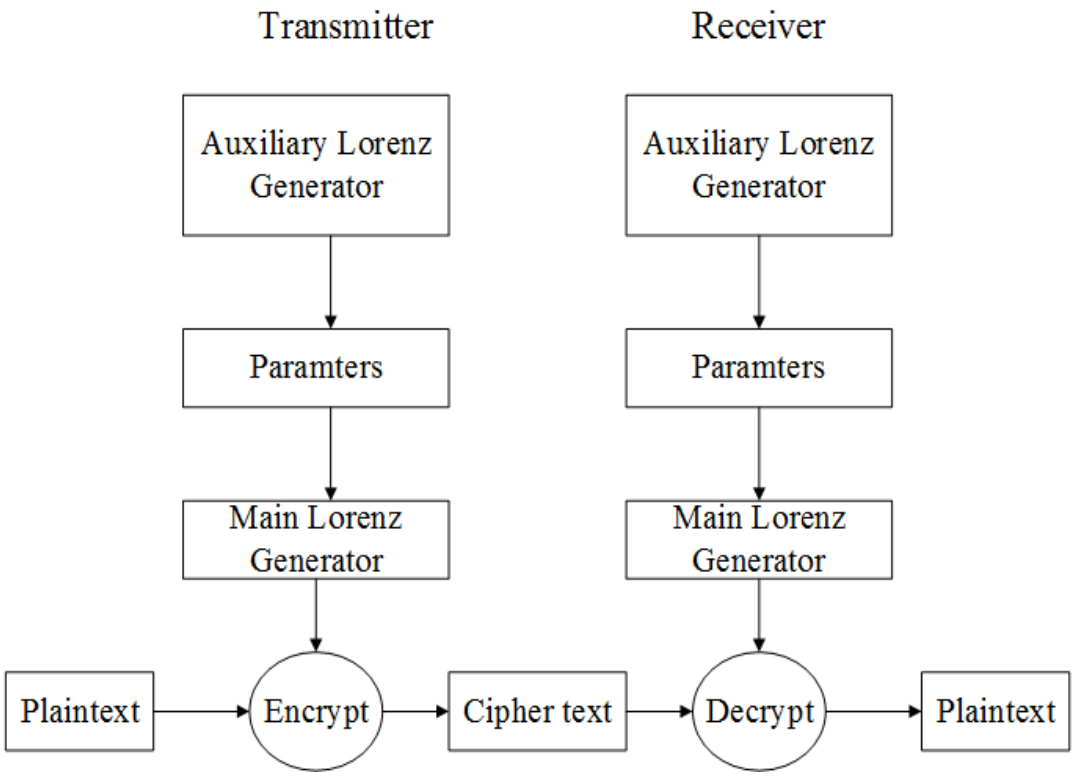

Fig. 3.2. Block diagram of cryptosystem.

3.2.2 An encryption system

To decrypt the cipher text produced using chaos-based communication, the receiver must have an identical chaotic generator to the transmitter. This means any intruder must have a complete knowledge of the chaos system parameters and initial conditions in order to be able to decipher the message.

The encryption technique utilises the output of the Main Lorenz Generator to encrypt the data stream. Both the Main Lorenz Generator and the Auxiliary Lorenz Generator are based on the equations (14) that is shown in chapter 1 section 1.7.3.

The A parameter of the main system is continuously variable by the auxiliary generator. Furthermore, the parameters and initial conditions of the cryptosystem are changing for every usage to satisfy the condition 3 of the one-time pad.

The Auxiliary Lorenz Generator is pre-configured with a different set of initial conditions and system parameters. This system serves to continuously vary one or more parameter(s) of the main Lorenz generator; in the case of the system described in this thesis, only the A parameter is varied. Care is taken to ensure that the main generator always remains in the chaotic region, the output of the Auxiliary Lorenz Generator (A[n]) must remain within the range ($7 \leq x[n] \leq 11$). Therefore, the signal response of the Main Lorenz Generator changes continually in a chaotic manner, based on the parameter supplied by the Auxiliary Lorenz Generator.

Fig. 3.3 shows the SIMULINK Lorenz model where A, B and C are system parameters. The scaling factors S_1, S_2 and S_3 are used to control the output signals frequency band and they are also part of the key in the encryption system.

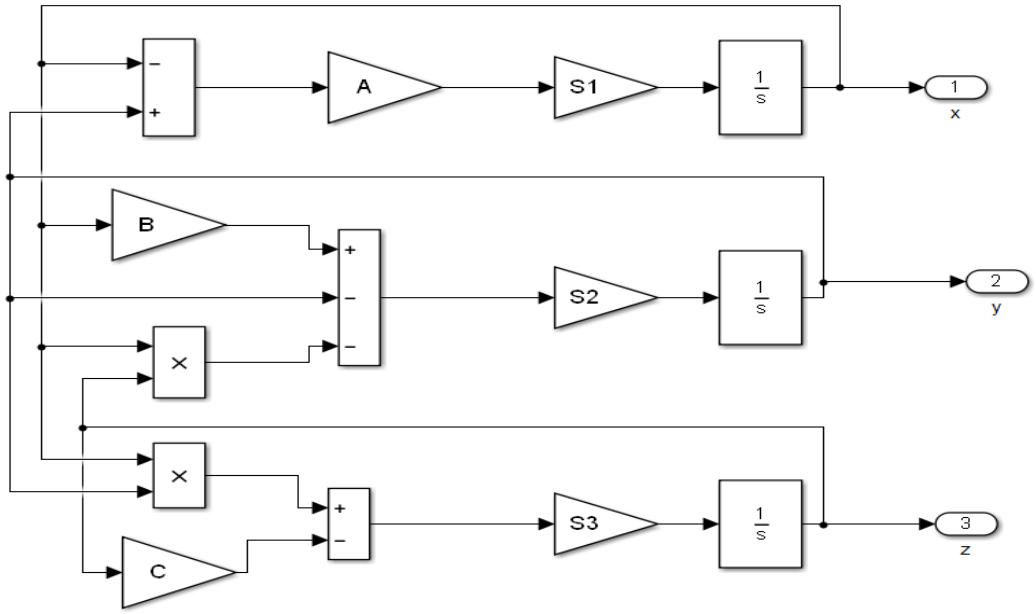

Fig. 3.3. Lorenz chaotic generator.

The CDMA system for four users has been tested using SIMULINK. Each user has two Lorenz chaotic generators, the Main Lorenz Generator and the Auxiliary Lorenz Generator. Fig. 3.4 shows the results from SIMULINK of the Lorenz State Variables, *x, y* and *z*.

The time scale in the results show is for a normalised simulation time. The user can determine the denormalisation factor to relate the results to practical system.

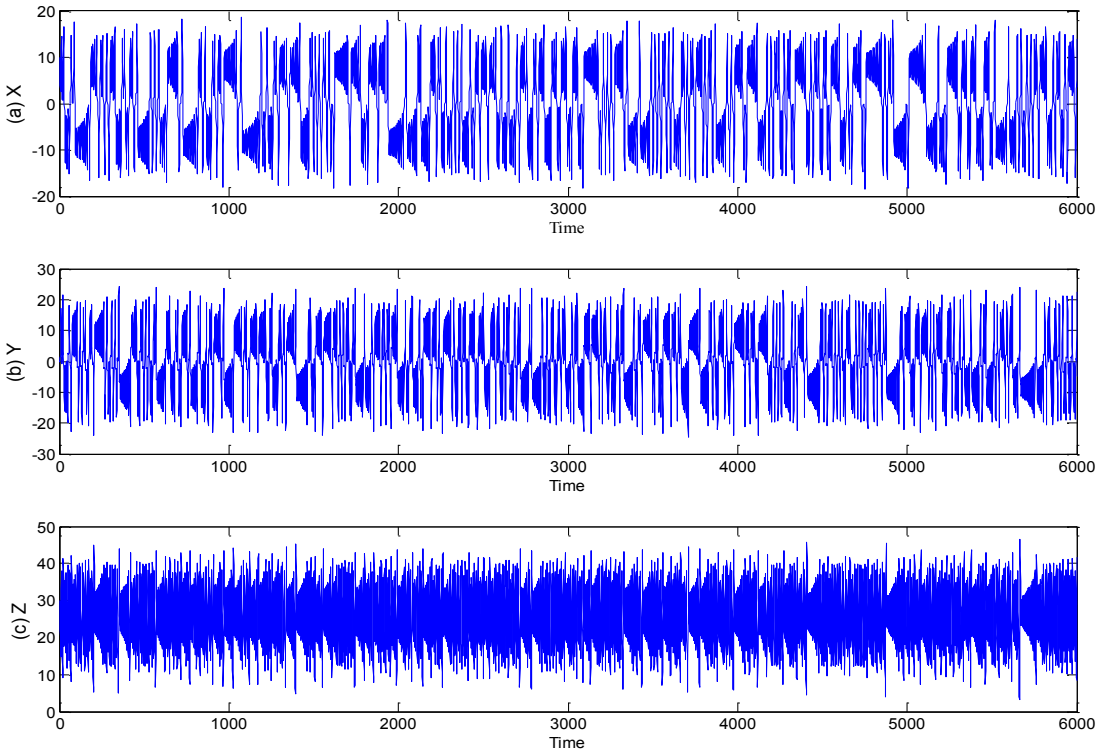

Fig. 3.4. Lorenz state variables. (a) *x*-state variable, (b) *y*-state variable and (c) *z*-state variable.

3.2.3 Analogue to Digital Conversion

Since the Lorenz system generates an analogue signal, an analogue-to-digital convertor (ADC) is necessary in the digital applications. Since the developed system is intended to be implemented using FPGAs, an ADC block is developed so that the whole system is self-sufficient. The ADC process starts by sampling the signal using a zero-order hold block. After that, the input is quantized and encoded into a 32-bit signed integer through a uniform encoder. The vector of the integer is mapped to a vector of unsigned bit values. The output order is the most significant bit (MSB), so that the signal can used in a digital communication system. Fig. 3.5 shows the ADC block diagram. Fig. 3.6 shows the simulation results of the ADC. The time scale in the results (refers to 3.2.2).

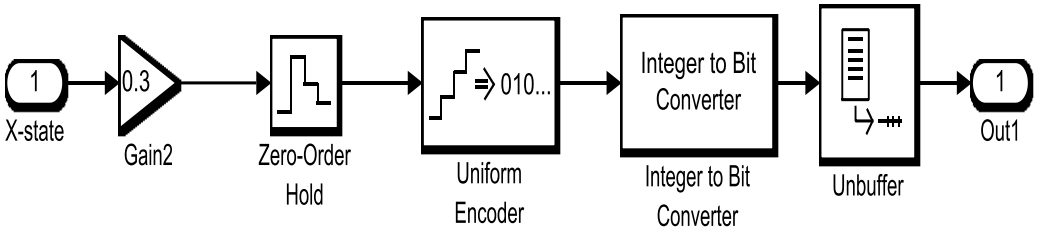

Fig. 3.5. Analogue-to-digital signal convertor.

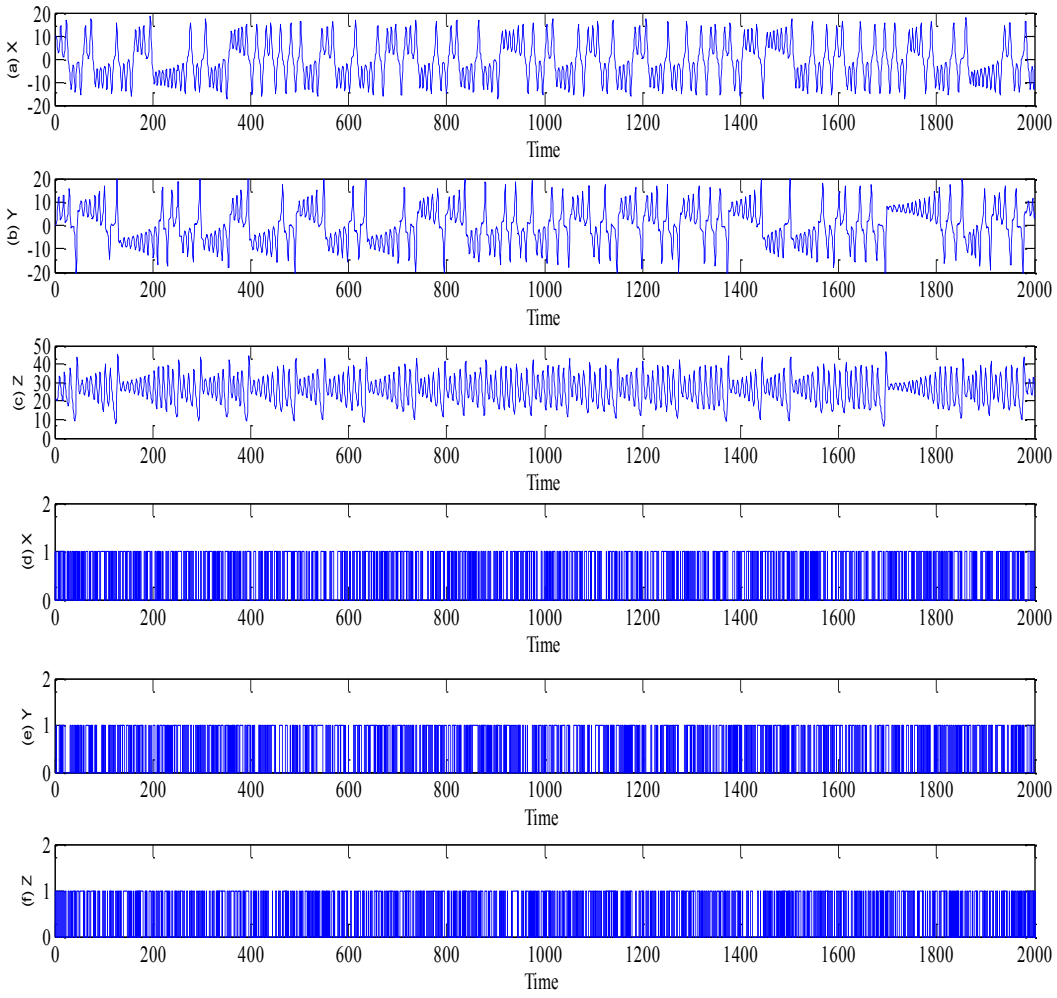

Fig. 3.6. Analogue chaotic signal converted to digital signal. (a) Analogue signal of *x*-state variable, (b) Analogue signal of *y*-state variable, (c) Analogue signal of *z*-state variable, (d) Digital signal *x*-state variable, (e) Digital signal of *y*-state variable and (f) Digital signal of *z*-state variable.

3.2.4 Randomness Test

The cipher stream must satisfy the randomness test to avoid any weaknesses in system security. In this experiment, 100 binary sequences each with a size of 1,000,000 bits are generated by the Lorenz Generator and tested by NIST.

Initially, the Lorenz Generator bit stream failed to pass the NIST randomness test. Therefore, an additional SIMULINK subsystem was developed to scramble the chaotic bit stream to generate a truly random bit stream. Table 3.2 shows the NIST randomness test of the three chaotic signals: *x*-state, *y*-state and *z*-state before scrambling.

Lorenz *x*-state bit stream of $1e^6$ bits, shown in Fig. 3.7, has a repetition of long stream of ones and zeros, which affected the randomness test.

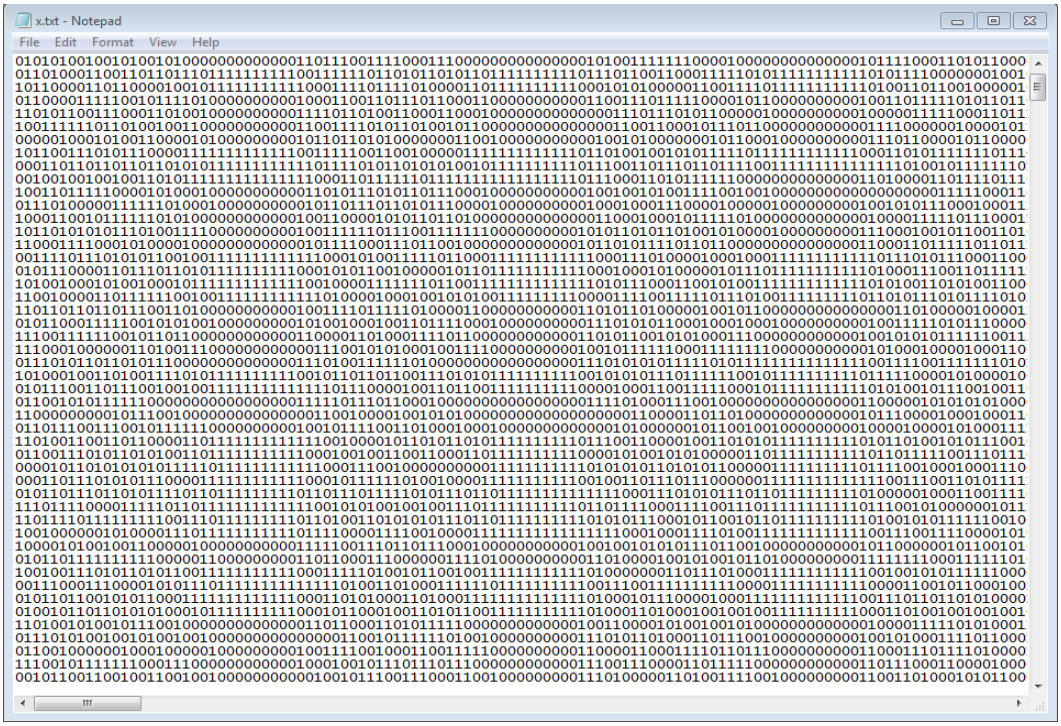

Fig. 3.7. *x*-state bit stream before scrambling which shows long repetition of ones and zeros.

Statistical Test	*x*-state		*y*-state		*z*-state	
	Status	P-value	Status	P-value	Status	P-value
Frequency	Fail	0.000000	Fail	0.000000	Fail	0.000000
Block Frequency	Fail	0.000000	Fail	0.000000	Fail	0.000000
CUSUM-Forward	Fail	0.000000	Fail	0.000000	Fail	0.000000
CUSUM-Reverse	Fail	0.000000	Fail	0.000000	Fail	0.000000
Runs	Fail	0.000000	Fail	0.000000	Fail	0.000000
Long Runs of Ones	Fail	0.000000	Fail	0.000000	Fail	0.000000
Rank	Fail	0.000000	Fail	0.000000	Fail	0.000000
FFT Test	Fail	0.000000	Fail	0.000000	Fail	0.000000
Non-Overlapping	Fail	0.000000	Fail	0.000000	Fail	0.000000
Overlapping	Fail	0.000000	Fail	0.000000	Fail	0.000000
Approximate Entropy	Fail	0.000000	Fail	0.000000	Fail	0.000000
Linear Complexity	Fail	0.000000	Fail	0.000000	Fail	0.000000
Serial	pass	0.350485	Fail	0.000000	Fail	0.000000

Table 3.2. First NIST randomness test.

3.2.5 Scrambling scheme of Lorenz chaotic signal

Two chaotic bit streams (*x*-state and *y*-state) have been used to generate a truly random key. The last 12 bits in row are extracted from *x*-state and last 20 bits are extracted from *y*-state. Then, the 32 bits are assembled with a concatenate block. The 32 bits are then serialized to generate a bit stream, which is used as a key stream for data encryption. Fig. 3.8 shows the SIMULINK model of the scrambling method. The bit

stream of the signed data type has been converted into unsigned. The constant block has been used to manipulate the 32 word length. Thus, the last 12 bits from *x*-state key stream have been extracted. The 12 bits word length has started from the least significant bit. The variable selector block has been used to extract a subject of rows from each matrix. The same operation has been used for *y*-state key stream. However, the 20 bits have been extracted from the y-state out of 32 bits word length that has started from least significant bit. After that, we concatenated the 12 bits and 20 bits using Matrix concatenation block to produce 32 word length. Then, the 32 bits has been sterilized using unbuffered block. The bit stream after scrambling is shown in Fig. 3.9. Table 3.3 indicates that the key stream now passes the NIST randomness test.

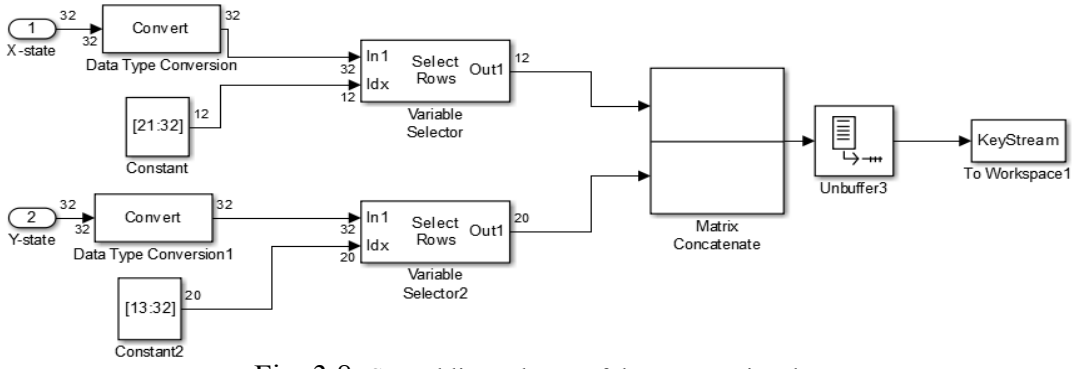

Fig. 3.8. Scrambling scheme of the Lorenz signals.

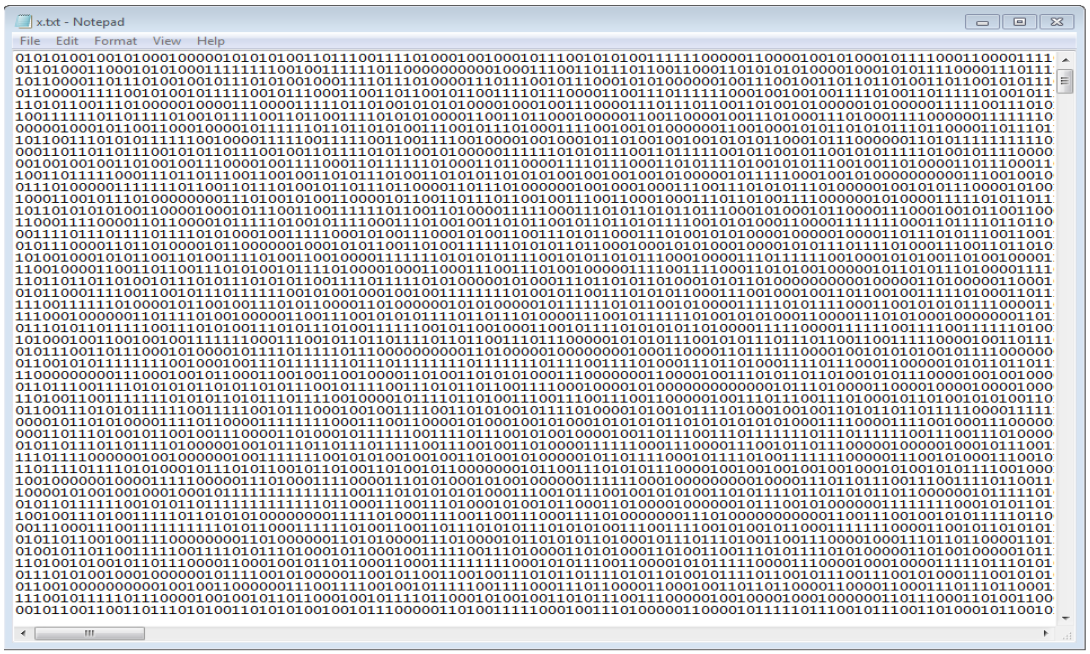

Fig. 3.9. The bit stream after scrambling.

Statistical Test	Status	P-value
Frequency	Pass	0.350485
Block Frequency	Pass	0.350485

CUSUM-Forward	Pass	0.739918
CUSUM-Reverse	Pass	0.534146
Runs	Pass	0.350485
Long Runs of Ones	Pass	0.911413
Rank	Pass	0.739918
FFT Test	Pass	0.534146
Non-overlapping	Pass	0.066882
Overlapping	Pass	0.122325
Approximate Entropy	Pass	0.911413
Linear Complexity	Pass	0.739918
Serial	Pass	0.739918

Table 3.3. Second NIST randomness test.

3.2.6 High system parameter sensitivity

The system parameters of the first Lorenz generator is A=9.7, B=26.2 and C=2.44. The Parameters A, B and C are chosen to be within the dynamic range of the chaotic generator to ensure that the generator remains in the chaotic range. The system parameters of the second Lorenz generator has the same parameters except for one which is (A parameter). This parameter has been changed from 9.7 to $9.7+10^{-15}$. The system output signals for *x* and *z* signals respond differently as shown in Fig. 3.10. Thus, we can generate infinite spreading codes for infinite number of users. The time scale in the results (refers to 3.2.2).

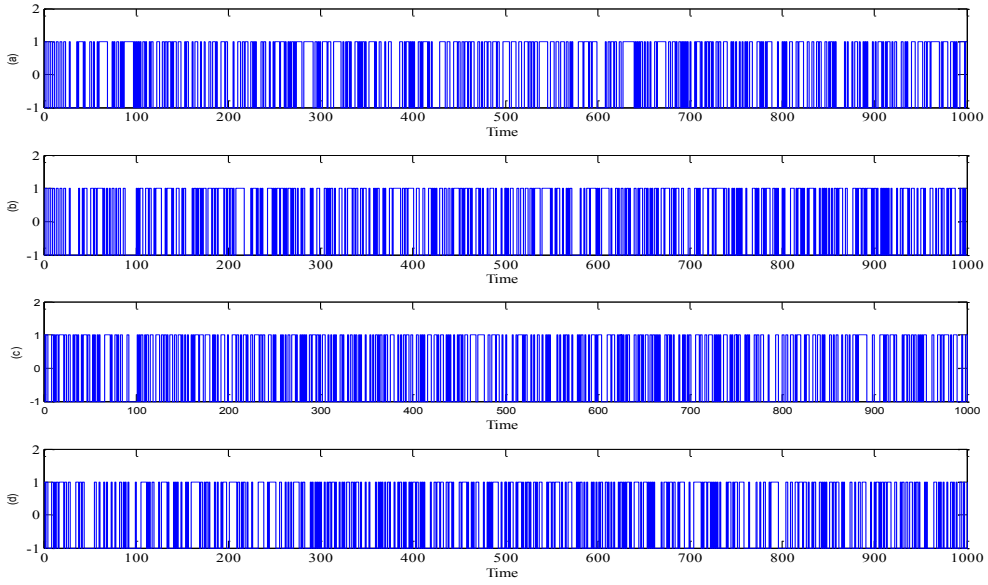

Fig. 3.10. Lorenz binary stream of two Lorenz generators. (a) *x*-state signal of first Lorenz generator, (b) *x*-state signal of second Lorenz generator, (c) *z*-state signal of first Lorenz generator and (d) *z*-state signal of second Lorenz generator.

The plots for 32-bits show low cross-correlation for *x* and *y* which is below 10. The similar result produces for *x* and *z*. The plot of cross-correlation of *y* and *z* shows low cross- correlation. The cross correlation for 32-bits is shown in Fig. 3.11. The time scale in the results (refers to 3.2.2).

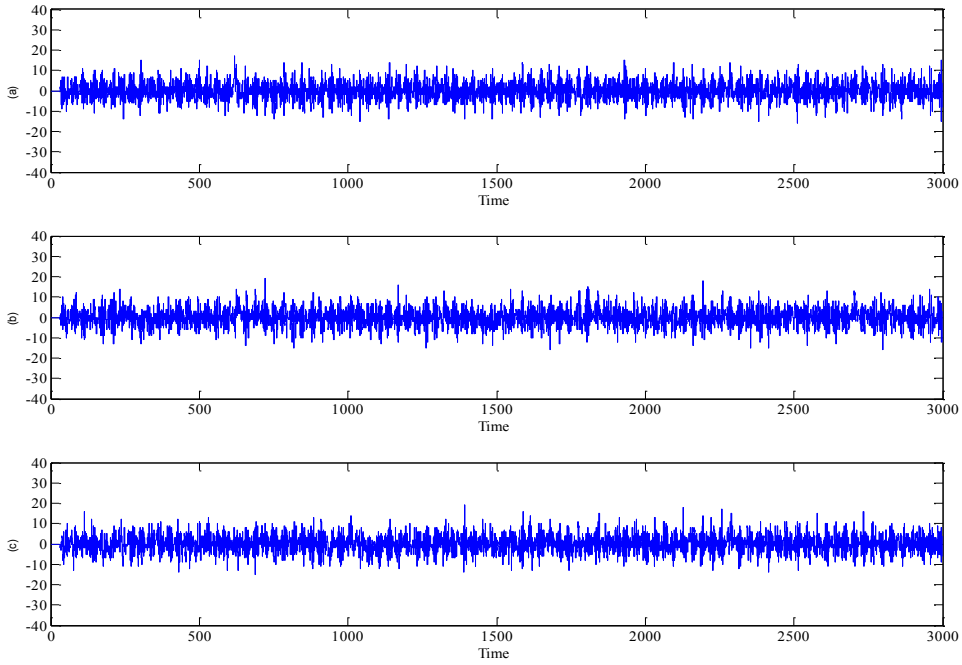

Fig. 3.11. The plot of the cross-correlation function for 32-bits (a) *x* xcorr *y* (b) *x* xcorr *z* and (c) *y* xcorr *z*.

3.2.7 The key space of the proposed cryptosystem

At the transmitter system, there are two Lorenz generators, and each generator has three constants, three initial conditions and three frequency multipliers. Thus, the total number of the parameters are 18. The word length represented by 32-bits.The key space of the system is $2^{(18*32)} = 2^{576}$. The key space of a secure cryptosystem as is suggested by previous research [2] should be greater than 2^{100}. Thus, the cryptosystem key space of 2^{576} is huge and enough to resist any brute force attack.

3.3 System Overview

The digital communication system is composed of three main parts: the transmitter system, the channel, and the receiver system. A transmitter system is constituted by the following subsystems, Lorenz chaotic generator, adder block and user data generator. The Goto block has been used to pass its input to its corresponding From

blocks. The input can be a real data type or a complex signal or vector. Thus, the Goto and From blocks are allowed designer to pass a signal from one block to another without connecting them. The channel is constituted by one subsystem which is a uniform Noise Generator. The receiver is constituted by the following subsystems, Lorenz chaotic generator, data extraction and Bit Error Rate (BER) calculation.

Fig. 3.12 shows the SIMULINK design of the four-user digital communication system based on a stream cipher.

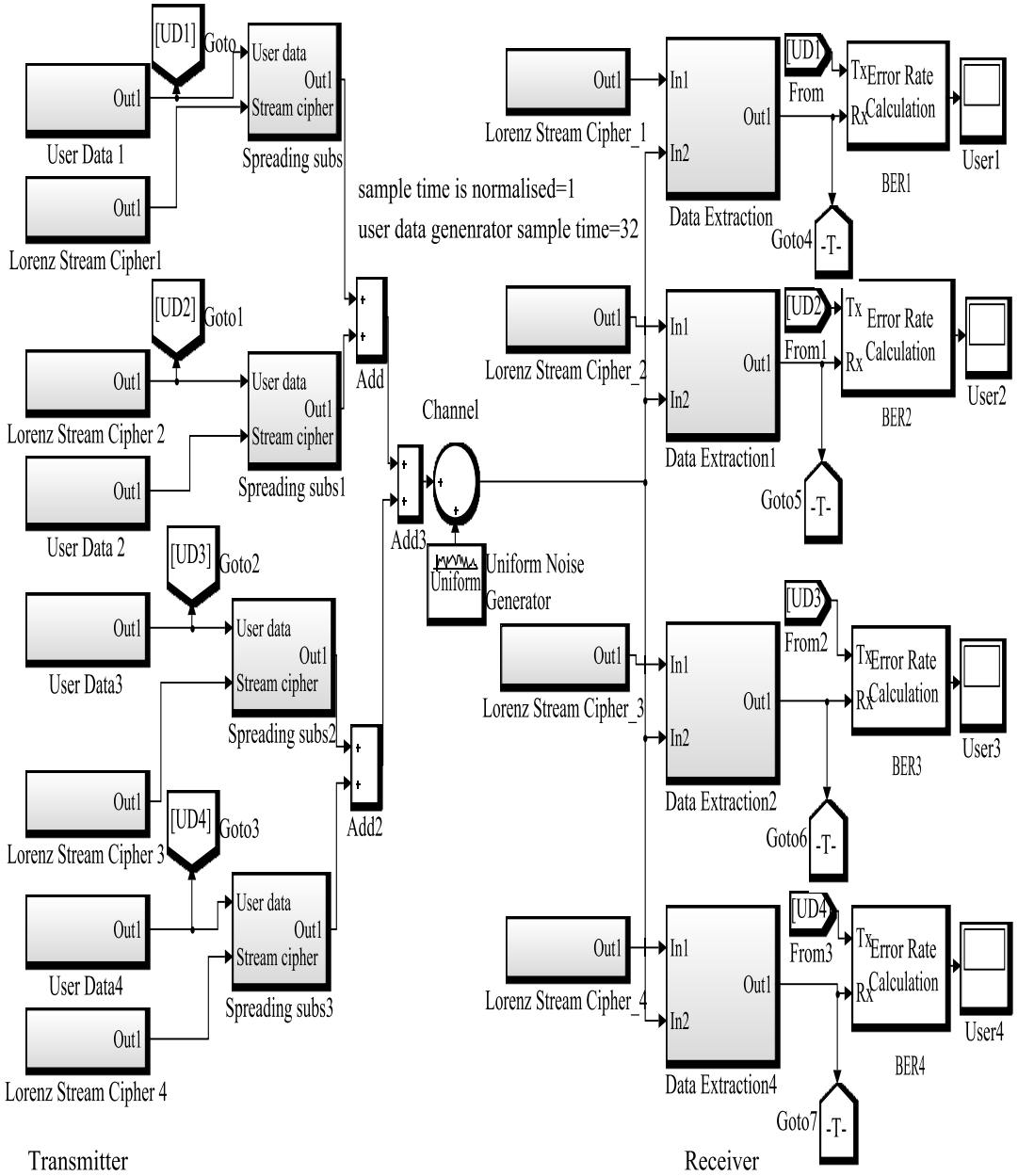

Fig. 3.12. A four-user digital communication system based on a Lorenz stream cipher.

3.3.1 Transmitter system

First, a Bernoulli Binary Generator is employed as a user data generator. To guarantee different random binary numbers for each user, each user has a different initial seed. The user data is spread by 32-bits that are generated from the Main Lorenz Generator using a product block. Fig. 3.13 presents a block diagram of the user data encrypted via multiplication. The normalized sample time of the user data generator is 32 and sample time of the Lorenz stream is 1.

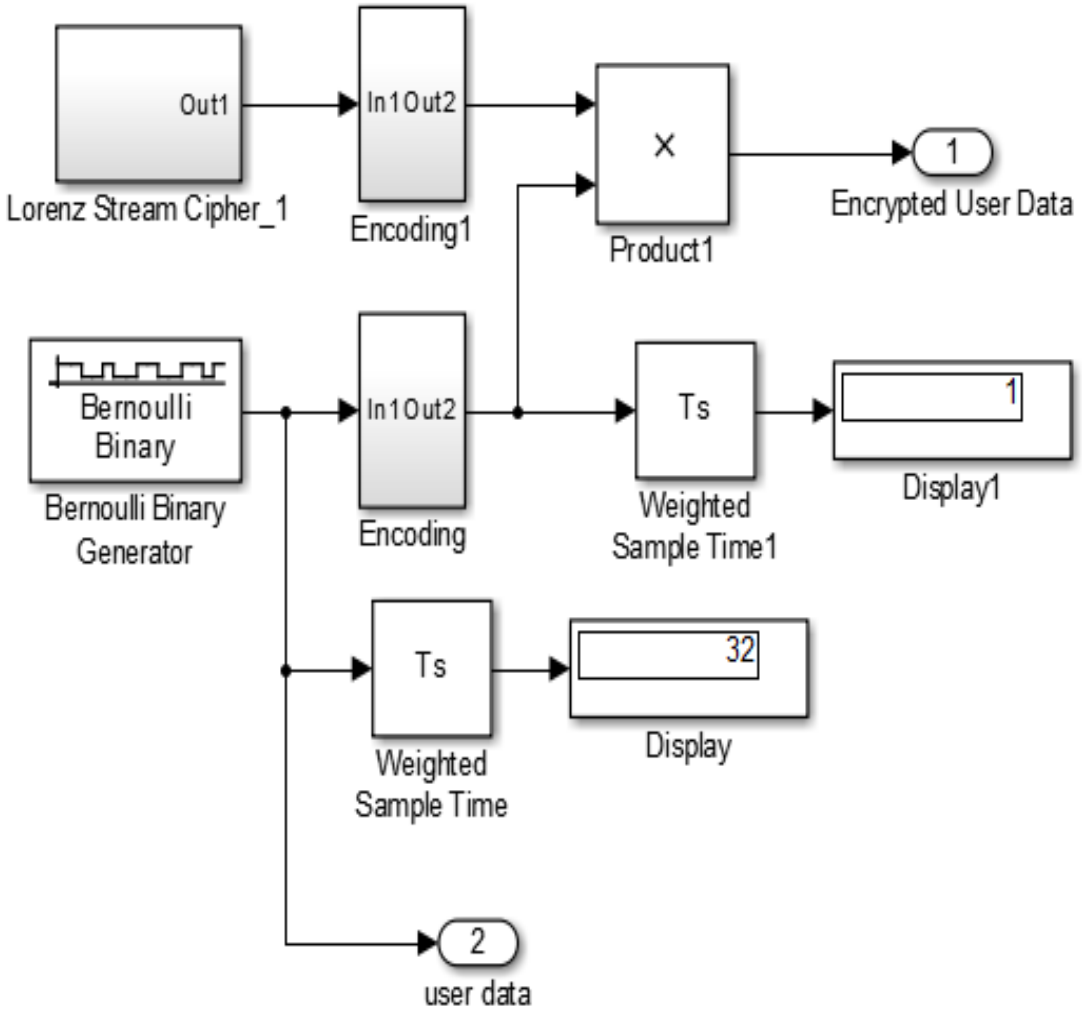

Fig. 3.13. SIMULINK model of the user data spreading based on Lorenz system.

In this application, the spread sequence is a chaotic digital signal generated using Lorenz chaotic systems [111]. The simulation results of the information signal spread using 32-bit length is shown in Fig. 3.14. The time scale in the results (refers to 3.2.2).

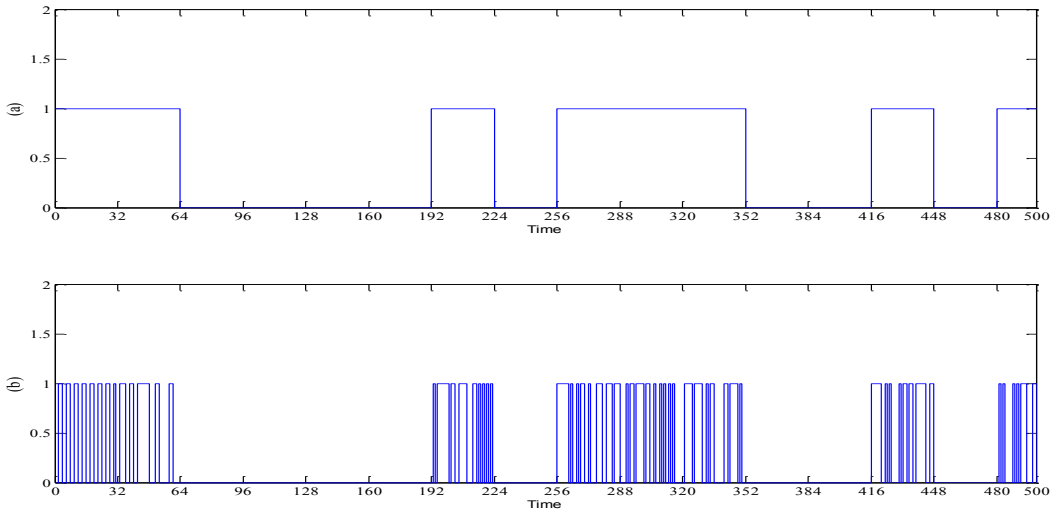

Fig. 3.14. Simulation results of user data spreading. (a) Information signal, and (b) Spreading the information signal using 32-bits length.

The encrypted unipolar stream consisting of ones and zeros is encoded to a bipolar stream of ± 1, so 0 is encoded to -1 and 1 stays, the same. The aim from encoding is to overcome the channel noise and to reduce the bit error probability at the receiver. The bipolar signaling has a 3-dB signal to noise improvement than on-off keying system[112]. Another reason to use bipolar encoding is that there is no security for a one user system. The user data transmitted through the channel can be easily recognised. However, if we have multiple-users, then it is not case. Fig. 3.15 shows the bipolar encoding scheme.

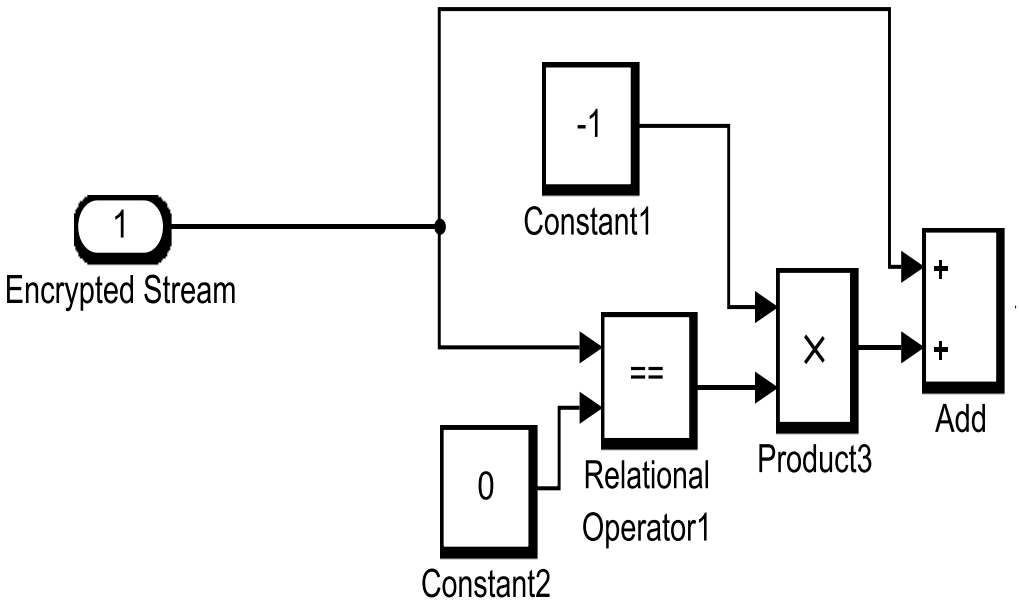

Fig. 3.15. The bipolar encoding scheme.

All four user data are combined using an adder block. Fig. 3.16 shows the user data encryption process. The information signal has been encoded from unipolar into bipolar. Also, the Lorenz key stream has been encoded into bipolar. The two encoded signals have been added together using addition block. The aim from bipolar encoding is to add security and immunity against white noise signals. Fig. 3.17 shows the all four user data are combined. The time scale in the results (refers to 3.2.2).

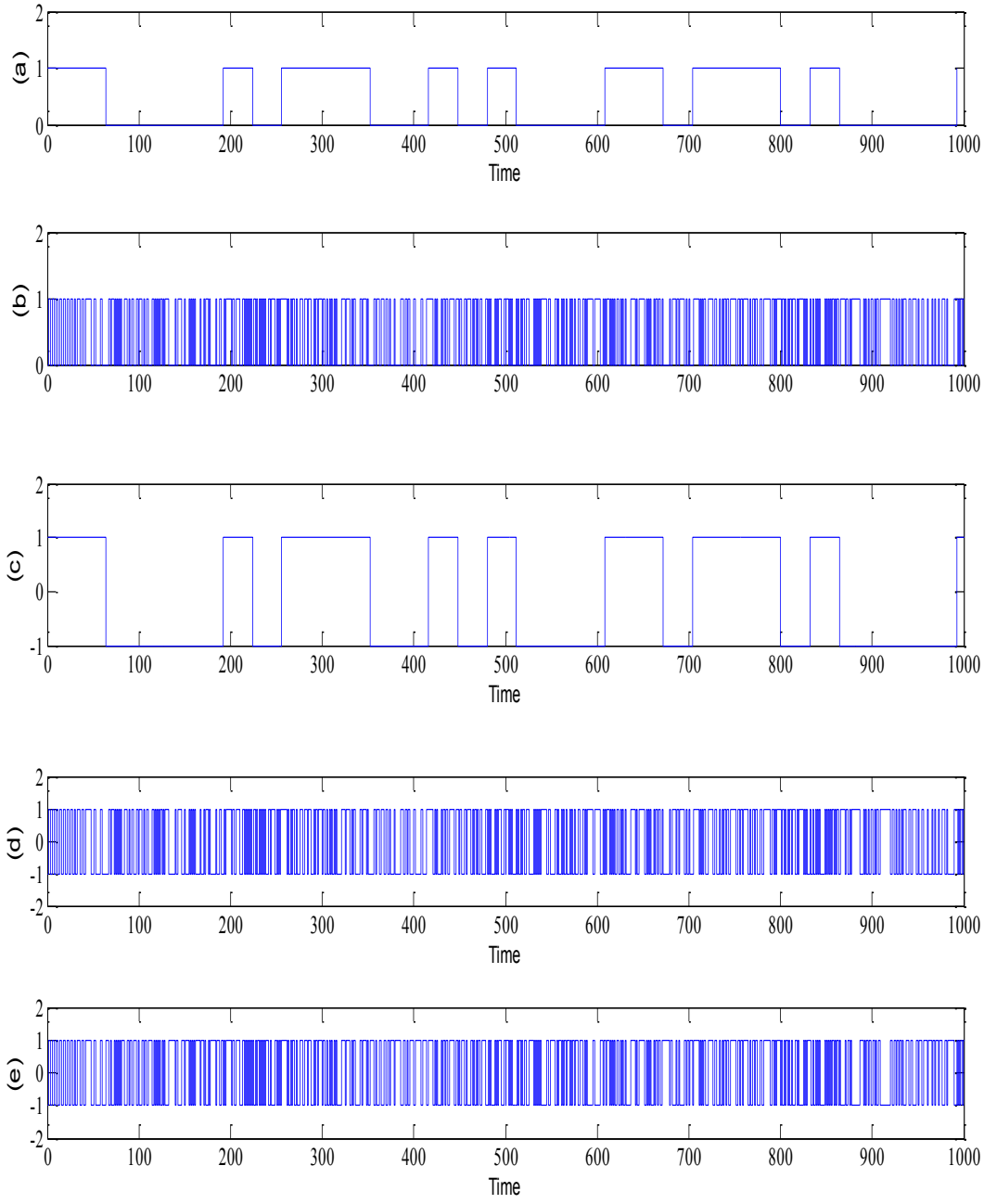

Fig. 3.16. SIMULINK results of the user data encryption process. (a) Information signal, (b) Lorenz binary stream, (c) Information signal is encoded to bipolar, (d) Lorenz binary stream is encoded to bipolar and (e) Encrypted information signal.

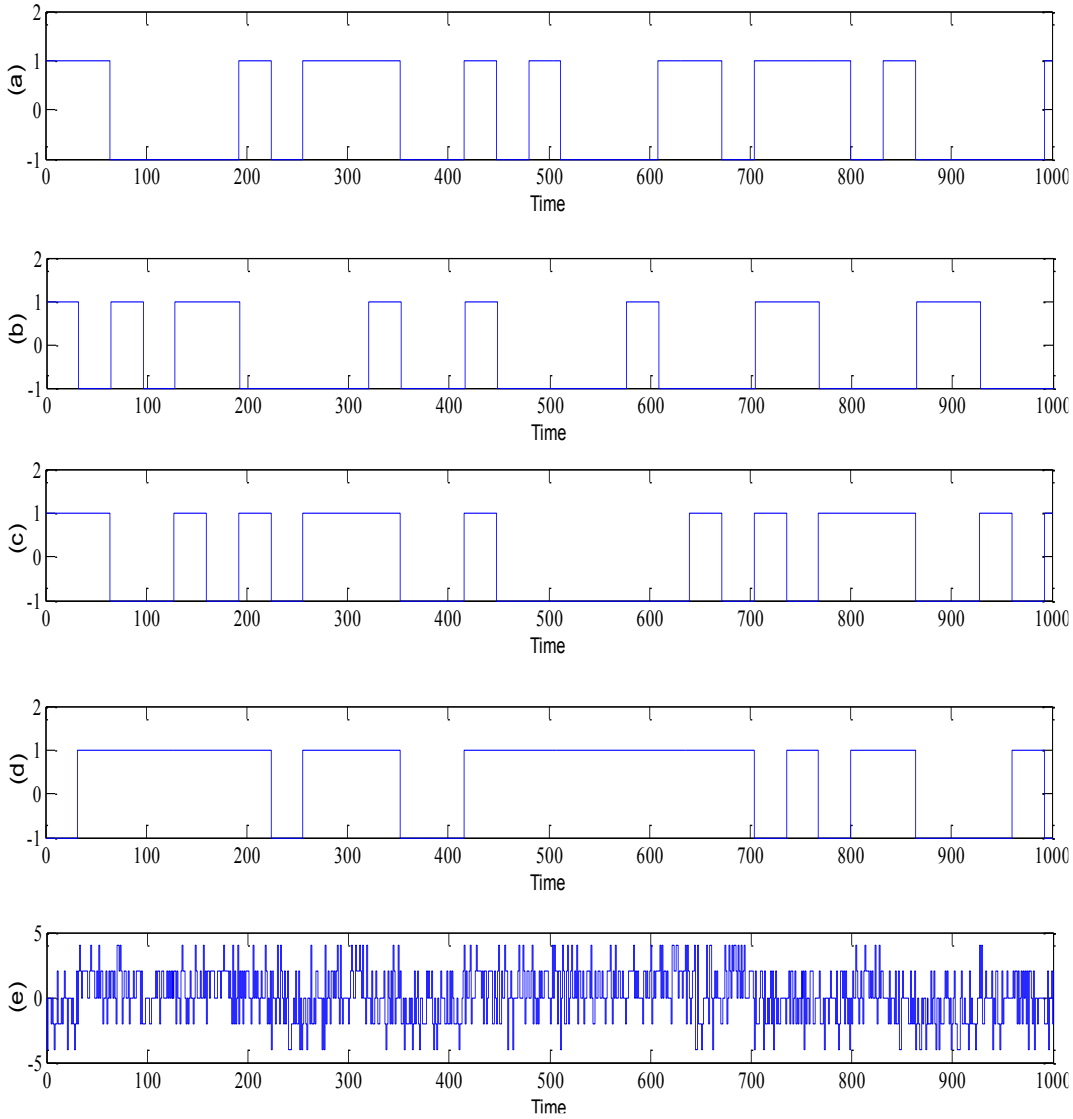

Fig. 3.17. SIMULINK results for combined four user data. (a) User data 1, (b) User data 2, (c) User data 3, (d) User data 4, and (e) All four user data are combined.

3.3.2 Receiver system

3.3.2.1 Auto-correlation based on Lorenz Generator

We used auto-correlation function to test the Lorenz binary stream for *x*-state. Four different code sizes have been used, 32, 64, 128 and 256-bits. The auto-correlation of the *y*-state, *z*-state and *x*-state are similar. Moreover, we have tested four different Lorenz generators with different parameters, the auto-correlation result of the shows the similar results. When the spreading code is longer, the auto-correlation value is

improved and vice versa. However, from a practical point of view, the long spreading code consumes more hardware resources.

In Fig. 3.18 (a), the plot of auto-correlation for 32-bits shows the maximum value is 32 and similarly to the 64, 128 and 256-bits. The time scale in the results (refers to 3.2.2).

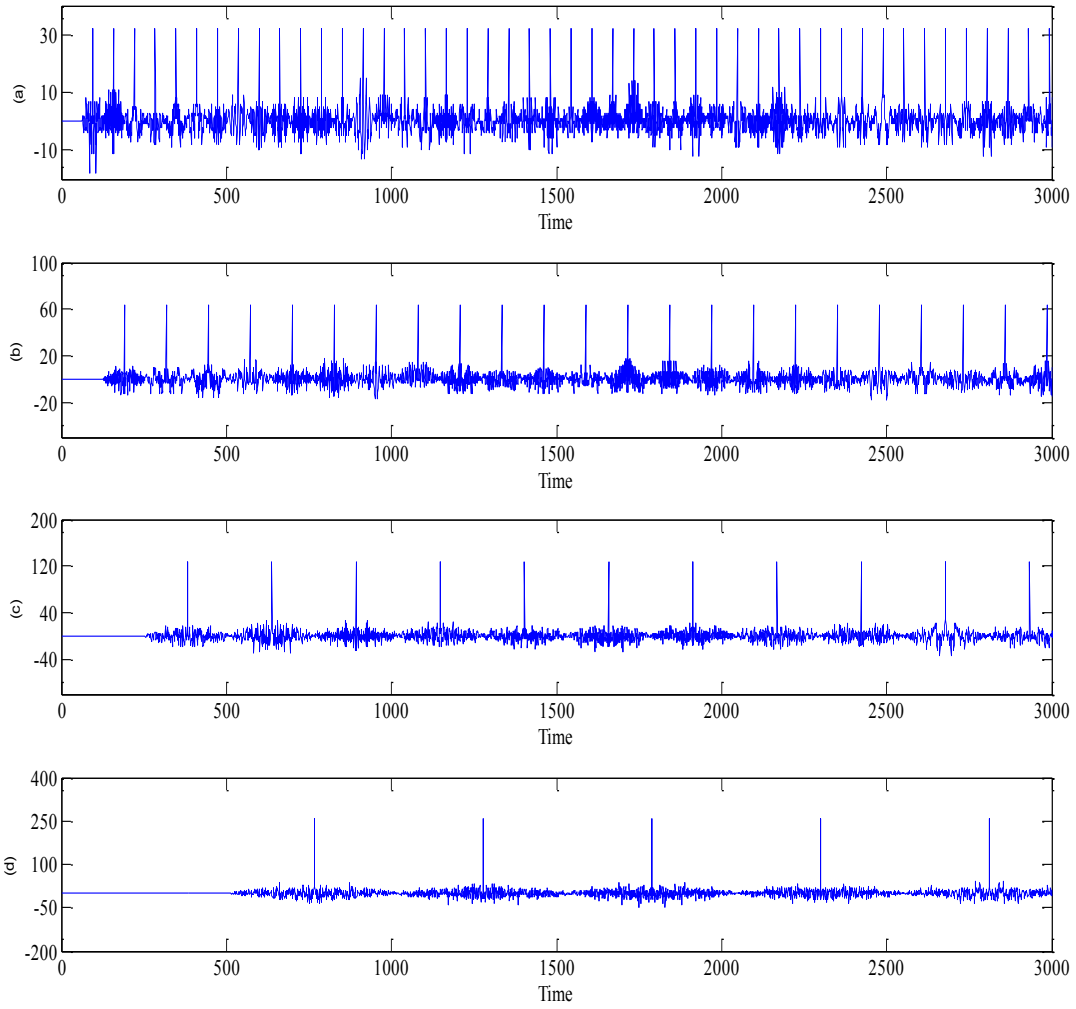

Fig. 3.18. The plot of the auto-correlation function. (a) Auto-correlation function of 32-bits long, (b) Auto-correlation function of 64-bits long, (c) Auto-correlation function of 128-bits long and (d) Auto-correlation function of 256 bit-long.

Comparison between the cross-correlation and auto-correlation of the chaotic signals is given in Table 3.4.

3.3.2.2 Cross-correlation based on one Lorenz Generator

The cross correlation of the Lorenz binary stream for different spreading codes length (32, 64, 128 and 256-bits) are shown in Fig. 3.19, Fig. 3.20, Fig. 3.21 and Fig. 3.22. In Fig. 3.19 shows the cross-correlation shows low for 32-bits of *x* and *y, x* and *z*. Thus, using *x* and *y, x* and *z* for user data spreading are recommend to mitigate the multi-

user interference and achieved better bit error rate. On the other hand, using *y* and *z* should be avoided due to three spikes in time series. These spikes are shown because of the high cross-correlation between two signals (y and z). The cross-correlation in Fig. 3.19, Fig. 3.20, Fig. 3.21 and Fig. 3.22 for 64, 128 and 256-bits show the same result as the 32-bits. The time scale in the results (refers to 3.2.2).

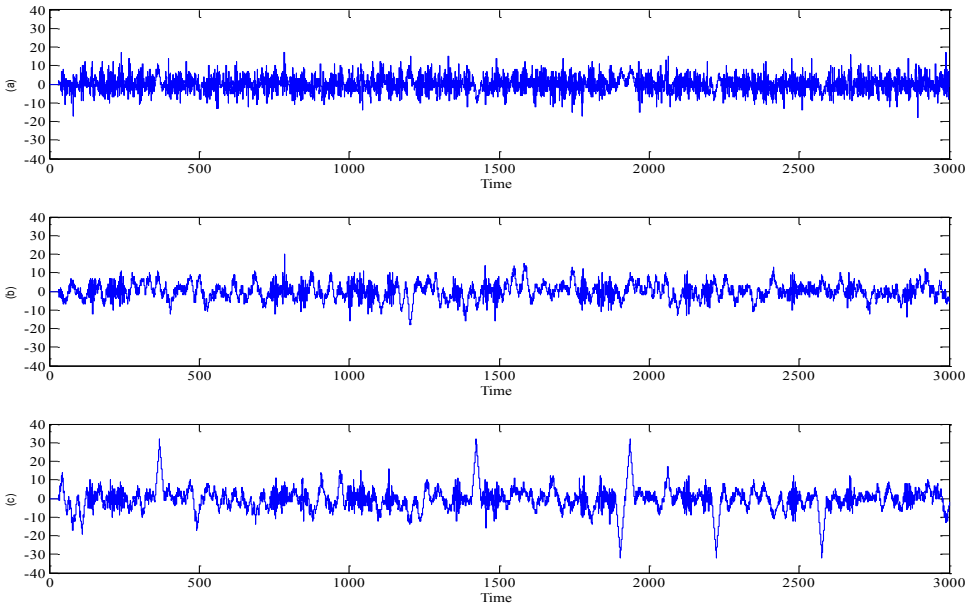

Fig. 3.19. The plot of the cross-correlation function for 32-bits (a) *x* xcorr *y* (b) *x* xcorr *z* and (c) *y* xcorr *z*.

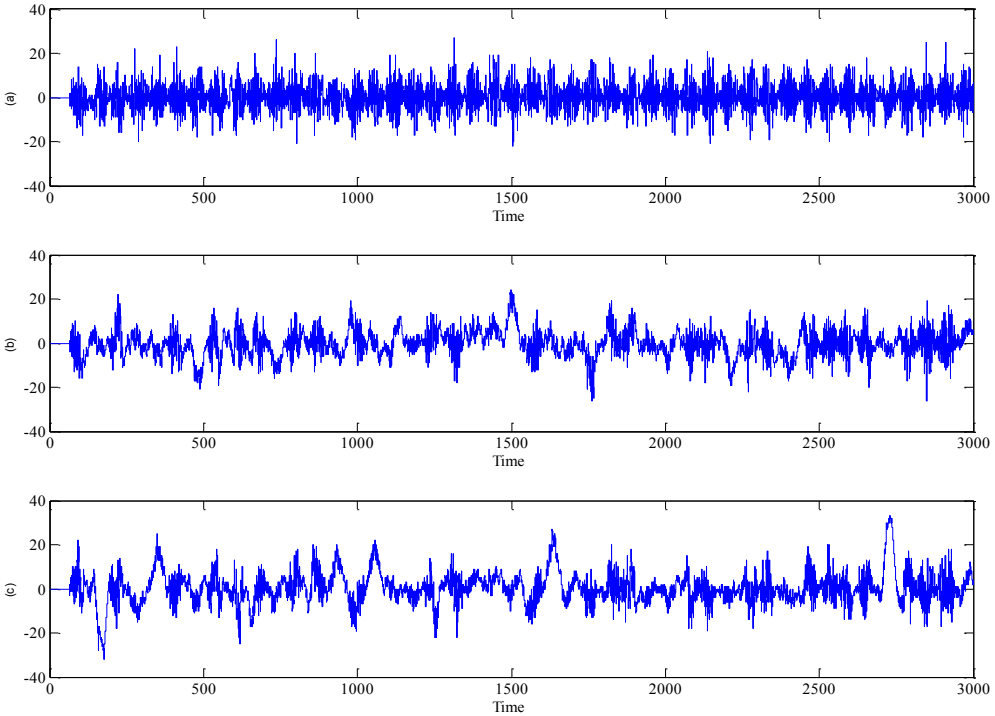

Fig. 3.20. The plot of the cross-correlation function for 64-bits (a) *x* xcorr *y* (b) *x* xcorr *z* and (c) *y* xcorr *z*.

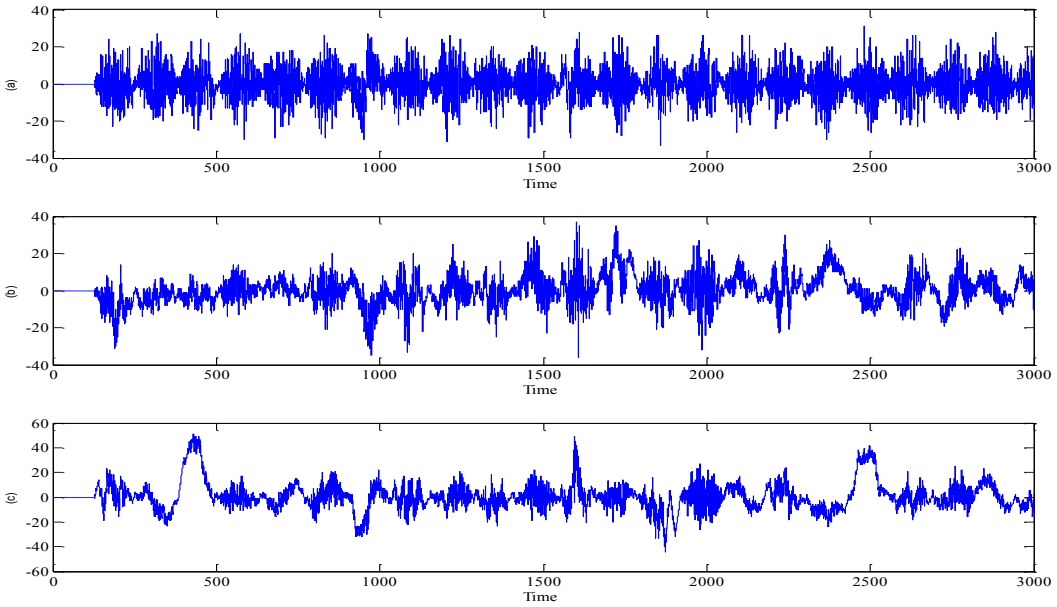

Fig. 3.21. The plot of the cross-correlation function for 128-bits (a) *x* xcorr *y* (b) *x* xcorr *z* and (c) *y* xcorr *z*.

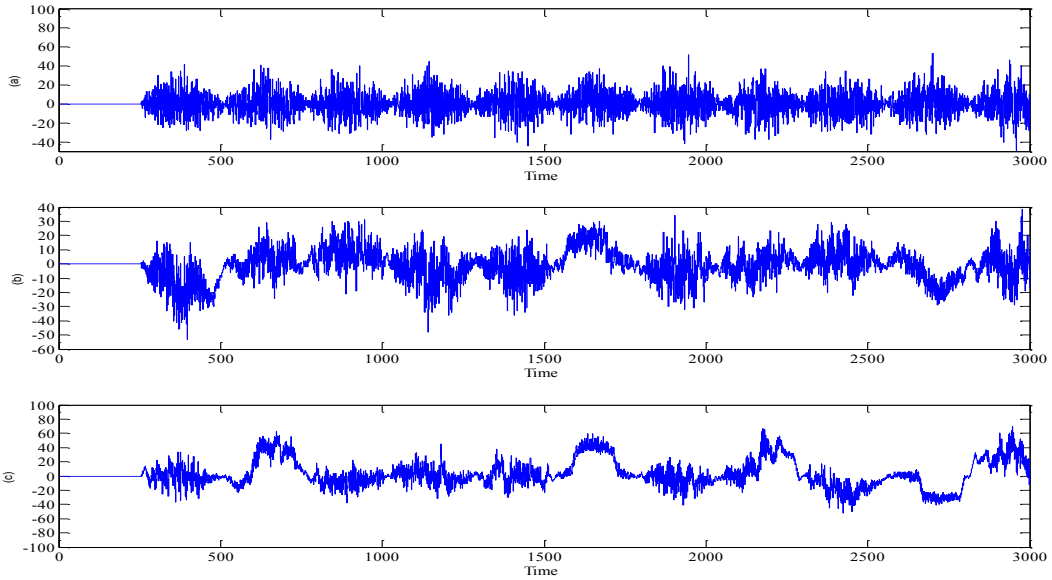

Fig. 3.22. The plot of the cross-correlation function for 256-bits (a) *x* xcorr *y* (b) *x* xcorr *z* and (c) *y* xcorr *z*.

The threshold is a value that is predefined at the receiver to discriminate a desire signal from all others. Thus, choosing a proper threshold value for detection is important. In this work, the threshold value depends on auto-correlation and cross-correlation functions. For example, in table 3.4, the auto-correlation of the signal is 32 and the cross-correlation is 8. Thus the threshold value should be between 8 and 32. Moreover, Table 3.4 shows the auto-correlation, cross correlation, threshold and difference between auto-correlation and cross-correlation for 32, 64, 128 and 256-bits.

Word-length	Auto-correlation	Cross-correlation	Threshold value	difference between auto-correlation & cross-correlation)
32-bits	32	8	$8 \leq \text{Threshold} \leq 32$	24
64-bits	64	20	$20 \leq \text{Threshold} \leq 46$	44
128-bits	128	25	$25 \leq \text{Threshold} \leq 128$	103
256-bits	256	40	$40 \leq \text{Threshold} \leq 256$	216

Table 3.4. Auto-correlation and cross-correlation for 32, 64,128 and 256-bits.

From these results, the auto-correlation is much larger than cross-correlation and these mean that we can use the chaotic systems for detecting the CDMA signals.

3.3.2.3 Cross-correlation based on two Lorenz Generators with different parameters

The system parameters of the first Lorenz generator is A=10, B=28 and C=2.6667. The system parameters of the second Lorenz generator is A= 8.8, B= 28.8 and C =1.9 (refers to section 3.2.6). In Fig. 3.23, the plot for 32-bits of *x* and *y*, *x* and *z*, *y* and *z* show low cross-correlation which is below 10. The time scale in the results (refers to 3.2.2).

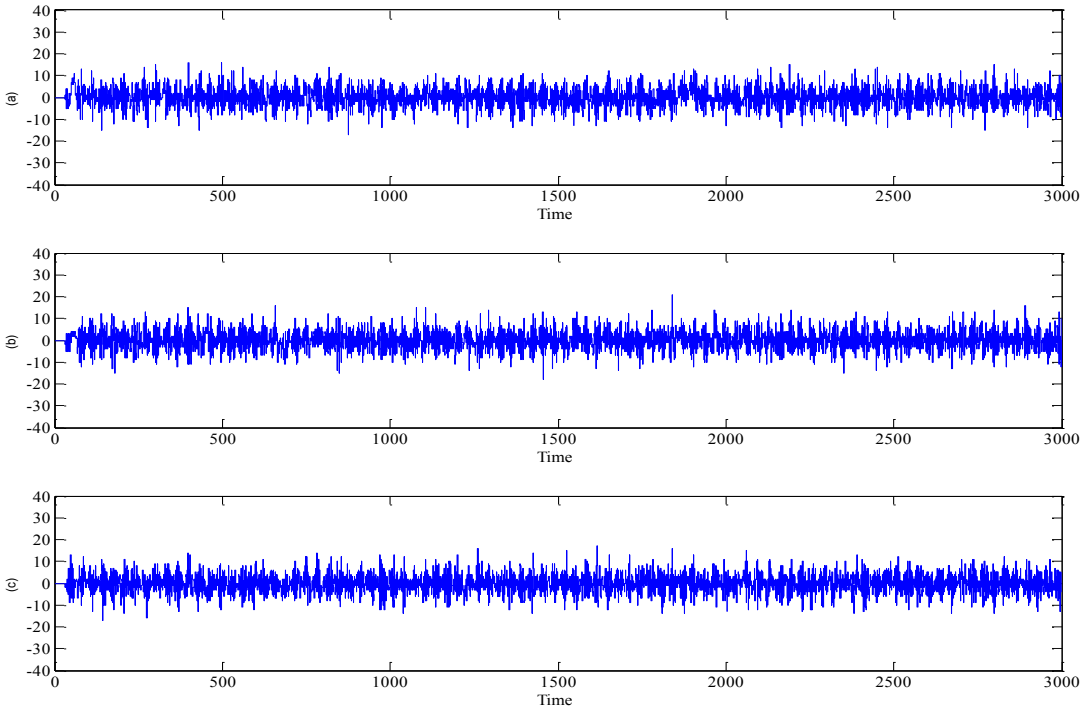

Fig. 3.23. The plot of the cross-correlation function for 32-bits (a) *x* xcorr *y* (b) *x* xcorr *z* and (c) *y* xcorr *z*.

In Fig. 3.24, the plot for 64-bits of *x* and *y*, *x* and *z*, *y* and *z* show low cross-correlation which are below 20.

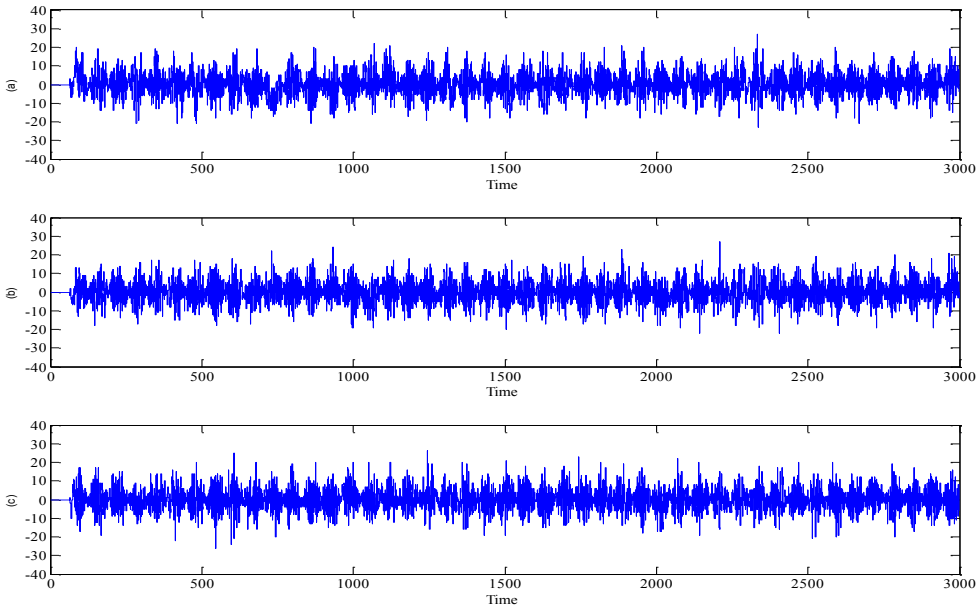

Fig. 3.24. The plot of the cross-correlation function for 64-bits (a) *x* xcorr *y* (b) *x* xcorr *z* and (c) *y* xcorr *z*.

In Fig. 3.25, the plot for 128-bits for *x* and *y, x* and *z, y* and *z* shows low cross-correlation which are below 25. Also, the plot for 128-bits of *x* and *z* shows low cross-correlation which is below 25.

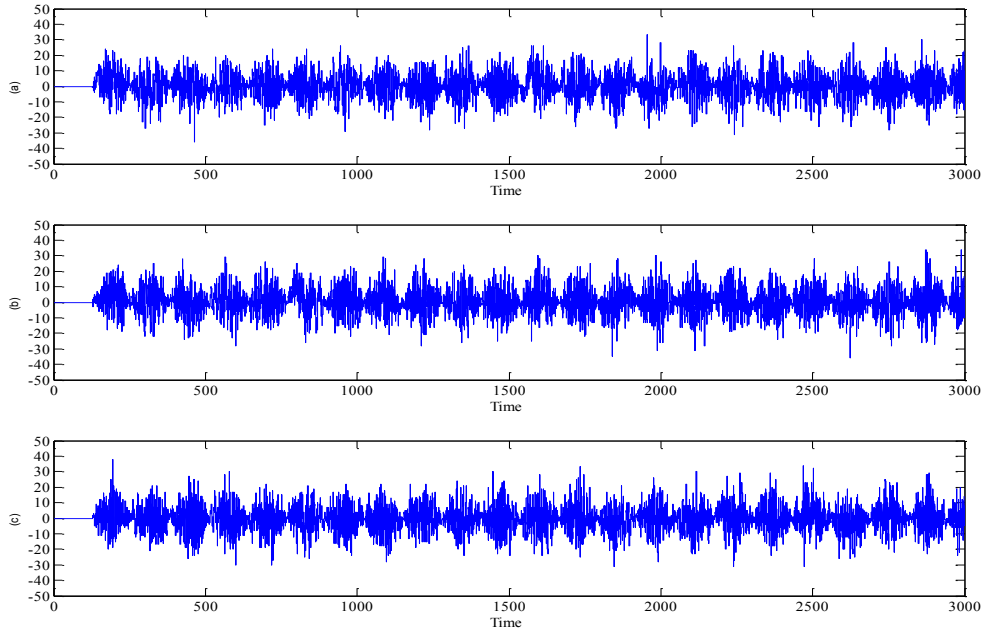

Fig. 3.25. The plot of the cross-correlation function for 128-bits (a) *x* xcorr *y* (b) *x* xcorr *z* and (c) *y* xcorr *z*.

In Fig. 3.26, the plot of cross-correlation for word length of 256-bits of *x* and *y*, *x* and *z*, *y* and *z* show good result which are below 50.

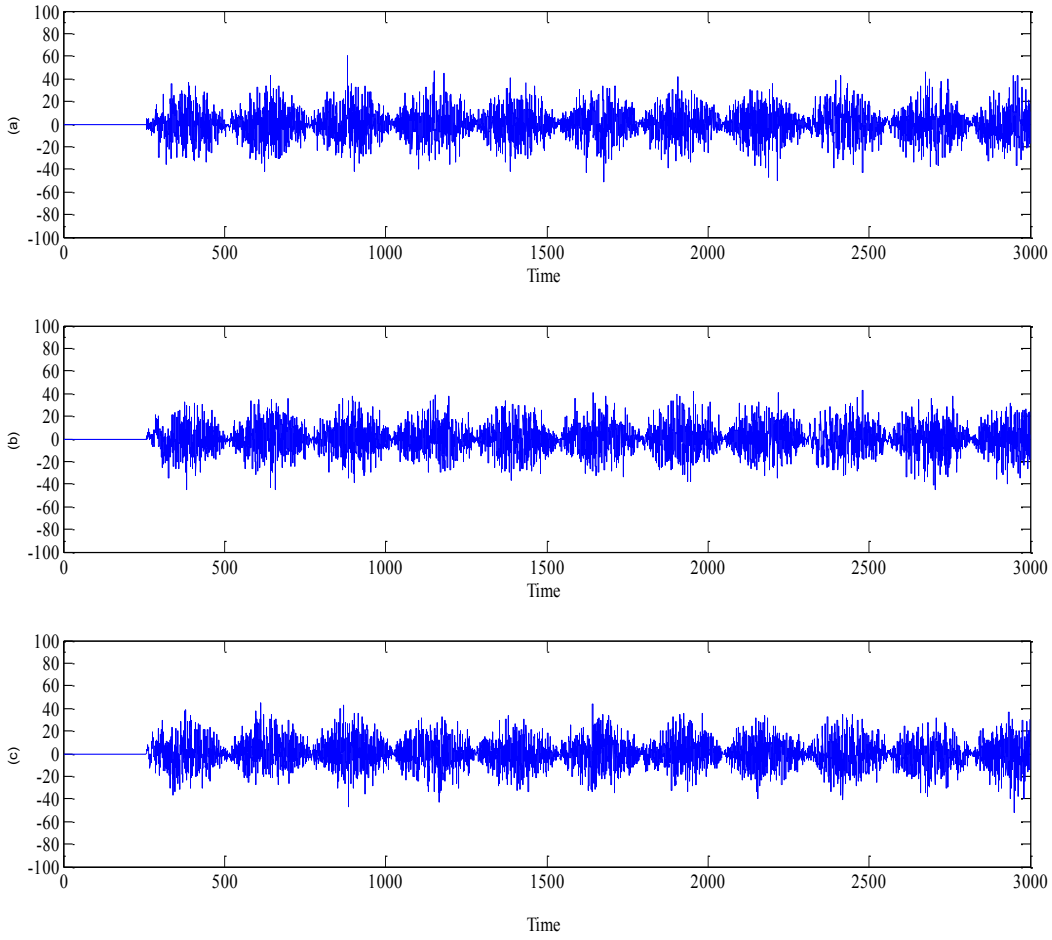

Fig. 3.26. The plot of the cross-correlation function for 128-bits (a) *x* xcorr *y* (b) *x* xcorr *z* and (c) *y* xcorr *z*.

Three different methods have been developed to extract the data at the receiver which are shown in subsections below.

3.3.2.4 De-spreading based on cross-correlation

One of the models has been designed using the SIMULINK XCORR block. This block performs the cross-correlation of two inputs. The replica of the chaotic sequence same as the one that used to encrypt the data in the transmitter and received signal. The scheme of the XCORR block is shown in Fig. 3.27. The Max block selects the maximum signal corresponding to the present user. The threshold value is calculated from the study of auto-correlation and cross correlation as explained in subsection 3.2.11.2. The divisor is chosen based on the maximum amplitude. The floor Function block has been used which rounds each bit to the nearest integer value towards zero.

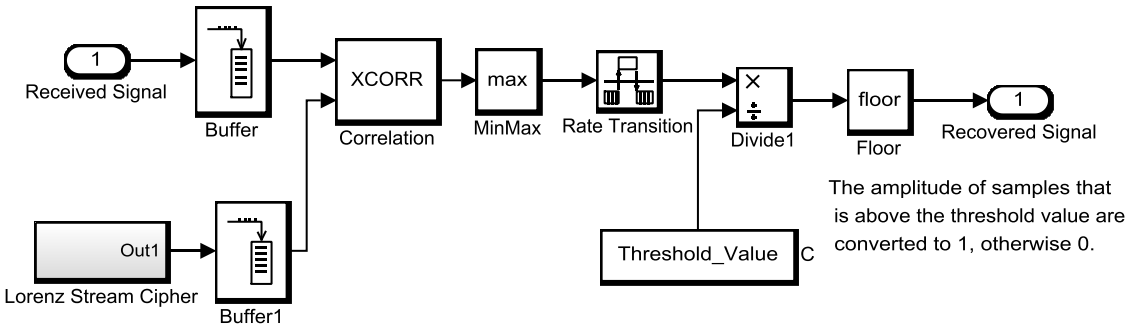

Fig. 3.27. De-spreading using cross-correlation block.

Plot of Fig. 3.28 shows the output of divide block. The divisor is chosen to be 110 because the maximum amplitude is 110. Using the floor Function block which rounds each bit to the nearest integer value towards zero. The time scale in the results (refers to 3.2.2).

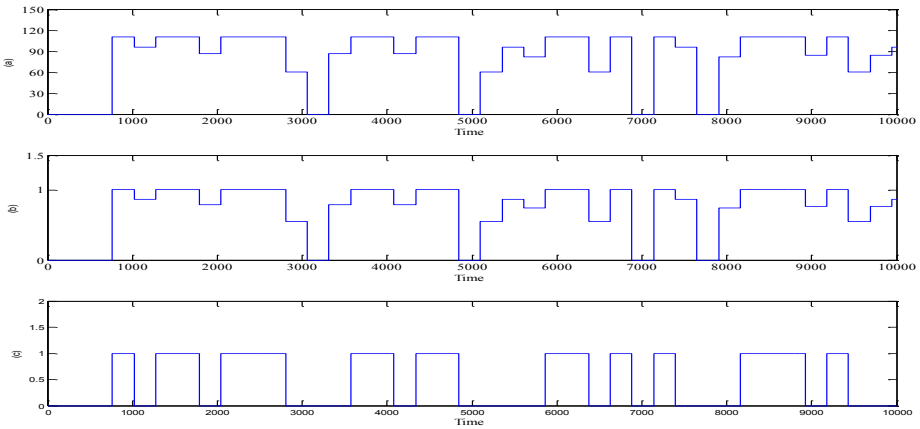

Fig. 3.28. De-spreading process. (a) Signal after dot-product. (b) Signal after division block and (c) Signal after floor.

Fig. 3.29 shows transmitted signal received and signal which is signal same as the one shown in Fig. 3.28.

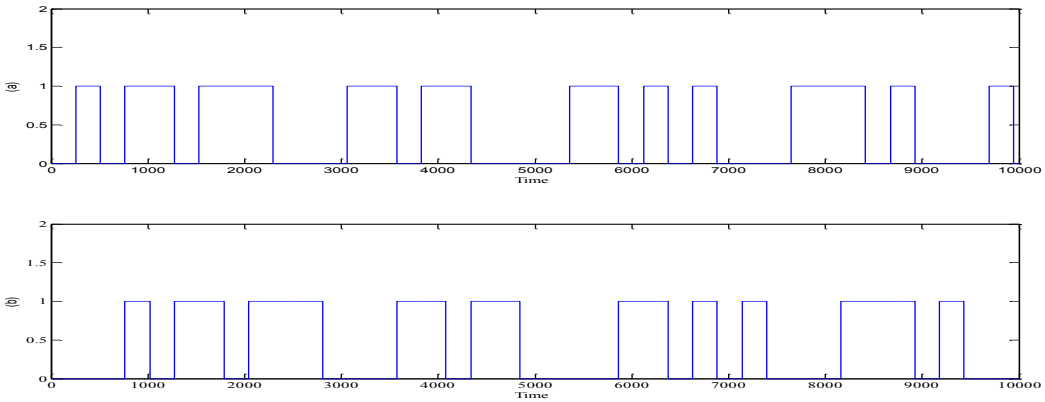

Fig. 3.29. (a) Transmitted signal and (b) Received signal.

3.3.2.5 De-spreading based on cross product and summation

It was necessary to use an alternative types of de-spreading method for data extraction. In sub-section 3.2.10.4, the de-spreading with the XCORR block has been described. By using the XCORR block, we are able to recover the signal without error. However, it is not possible to design same model in Xilinx System Generator because some of the SIMULINK blocks are not yet available in the Xilinx System Generator. Thus, the cross product block has been used instead of XCORR block which is shown in Fig. 3.30.

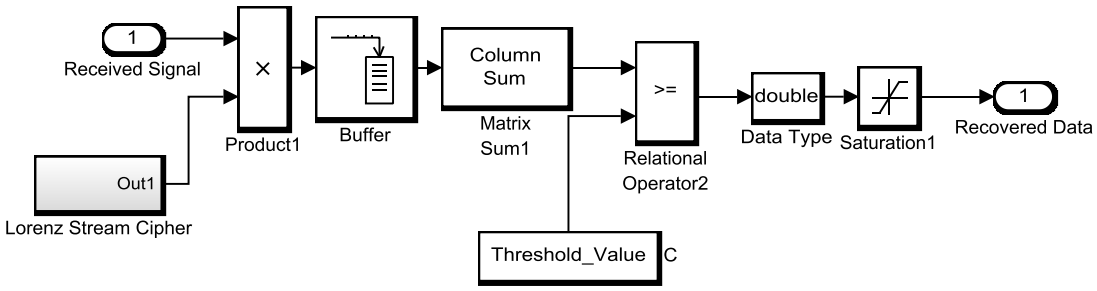

Fig. 3.30. De-spreading using cross product and summation.

The first process begins with two signals (the received signal and a replica of the transmitter binary stream at receiver) are multiplied using a cross product block. The output of product block goes into the Buffer which converts the signal into frame base. The framed signal goes into a Matrix Sum block, which sums the elements of an M-by-N input matrix along its rows, its columns, or over all its elements. Then, the user data can be discriminate from other users based on the maximum amplitude. The maximum amplitude value is then compared with the threshold value by using relational operator(≥). This method has been used to test system performance as it will be shown in sub-section 3.2.13.

3.3.2.6 De-spreading based dot product

One of the methods that can be used for data extraction is dot product. The dot product block generates the dot of the vectors at its inputs. The scalar output, y is equal to the MATLAB® operation stated below.

$$y = \text{sum}(\text{conj}(u_1) .* u_2) \quad (15)$$

Where u_1 and u_2 are vectors. The synchronized replica chaotic sequence is dot multiplied with the received signal. The scheme of the dot product block is shown in Fig. 3.31. The rest of SIMULINK blocks are similar to the XCORR scheme which have explained previously. The performance of this method is similar to the cross product.

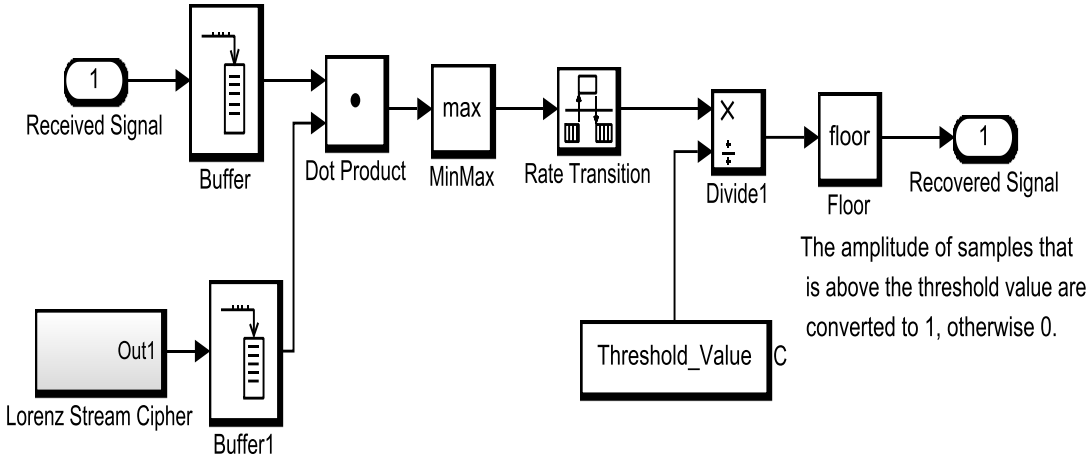

Fig. 3.31. De-spreading based on dot product.

In Fig. 3.32, the Plot shows the output of divide block. The divisor is chosen to be 110 because the maximum amplitude is 110. Using the floor Function block which rounds each bit to the nearest integer value towards zero.

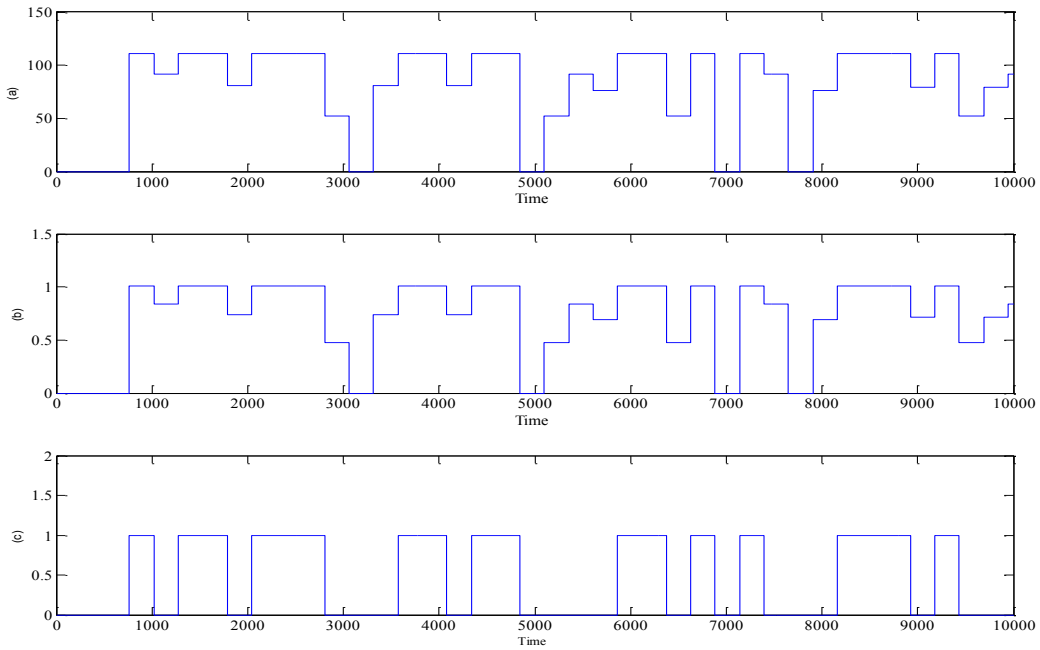

Fig. 3.32. De-spreading process. (a) Signal after dot-product. (b) Signal after division block and (c) signal after floor.

3.3.3 Data extraction

The product and summation for 32-bit is low which is shown in Fig. 3.33. The maximum value is below 10. Using *y* and *z* should be avoided due to high correlation which show spikes in time series.

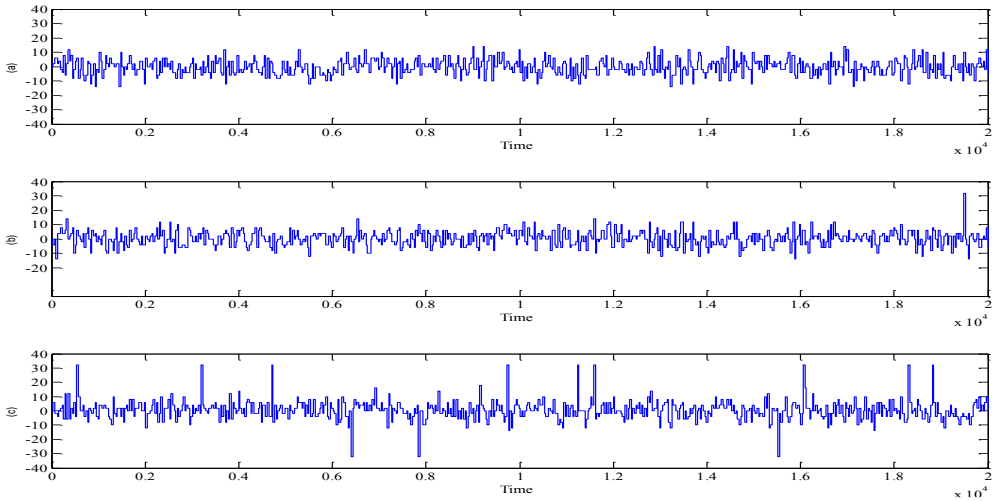

Fig. 3.33. Simulation results of product and summation for 32-bits. (a) *x* and *y* ,(b) *x* and *z* and (c) *y* and *z*.

Since the received signal is synchronised with a Lorenz binary stream at the receiver side, the process of the user data extraction is based on product and summation. Fig. 3.34 displays the data extraction process.

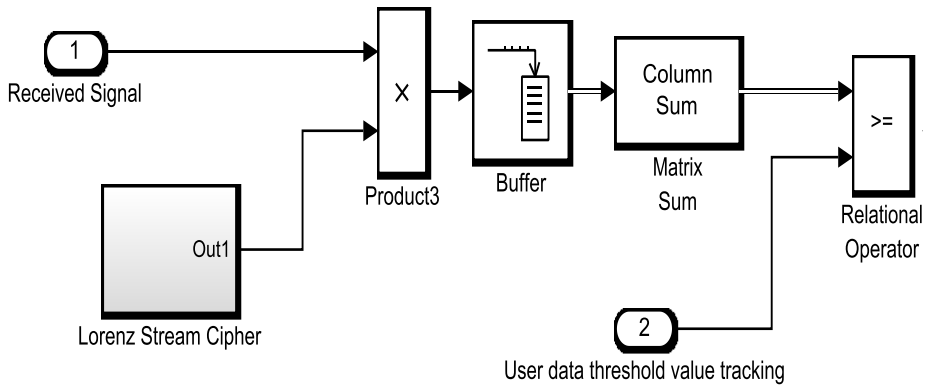

Fig. 3.34. SIMULINK block diagram of the data extraction process.

The process of the de-spreading of the received signal using cross product method has been described in sub-section 3.2.10.5.

The threshold is a value chosen to discriminate each user data from all others. Since the stream cipher binary stream is random. Not like a block spreading system where the block is fixed and we can predefined the threshold value, the stream cipher is a

random binary generator. Thus, we need to develop a tracking subsystem that can provides a proper threshold value which is varying each time based on number of ones every 32 samples.

Tracking subsystem contains Lorenz generator, Encoding, buffer, matrix sum block, subtract block and operational operator. Since the whole system is synchronized, the process of the threshold tracking is started by counting ones that are generated from stream cipher. The buffer-output size of 32 has been used. The matrix sum block has been used to accumulate 32 samples and generate a threshold value. In order to provide a save margin for threshold value, a subtract block is developed which is subtracts the maximum threshold value with the constant number. The output value from the previous step is now the threshold value. After that, this threshold values is compared with the result of correlation process using relational operator. Fig. 3.35 displays this user data threshold tracking subsystem.

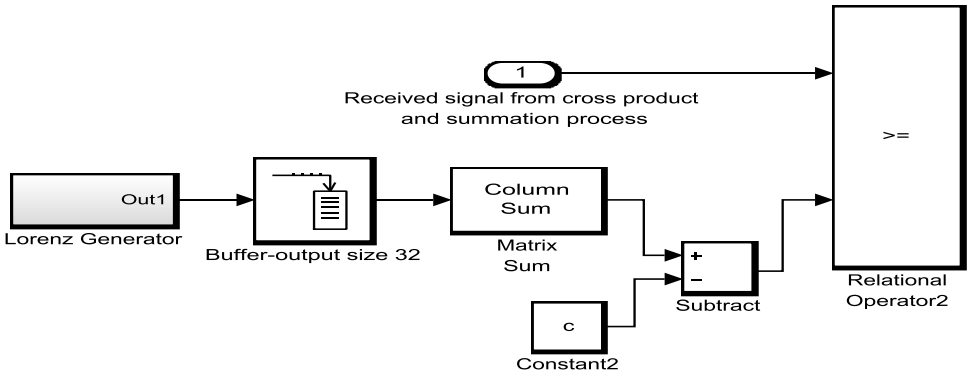

Fig. 3.35. SIMULINK block diagram of the user data threshold tracking value.

The relational operator block is used to extract the user data by comparing the threshold value with the replica of the user spreading binary stream. Fig. 3.36 shows a whole user data extraction process. The synchronized received signal and the encoded Lorenz bit stream are multiplied using cross product block. The results of multiplication are accumulated every 32 samples. After that, the threshold is used to extract the desired signal. The transmitted information have been recovered with no error. Fig. 3.37 shows the four users' data transmitted and recovered. The 10^4 bits have been transmitted for each user and recovered at the receiver side with no error. The time scale in the results (refers to 3.2.2).

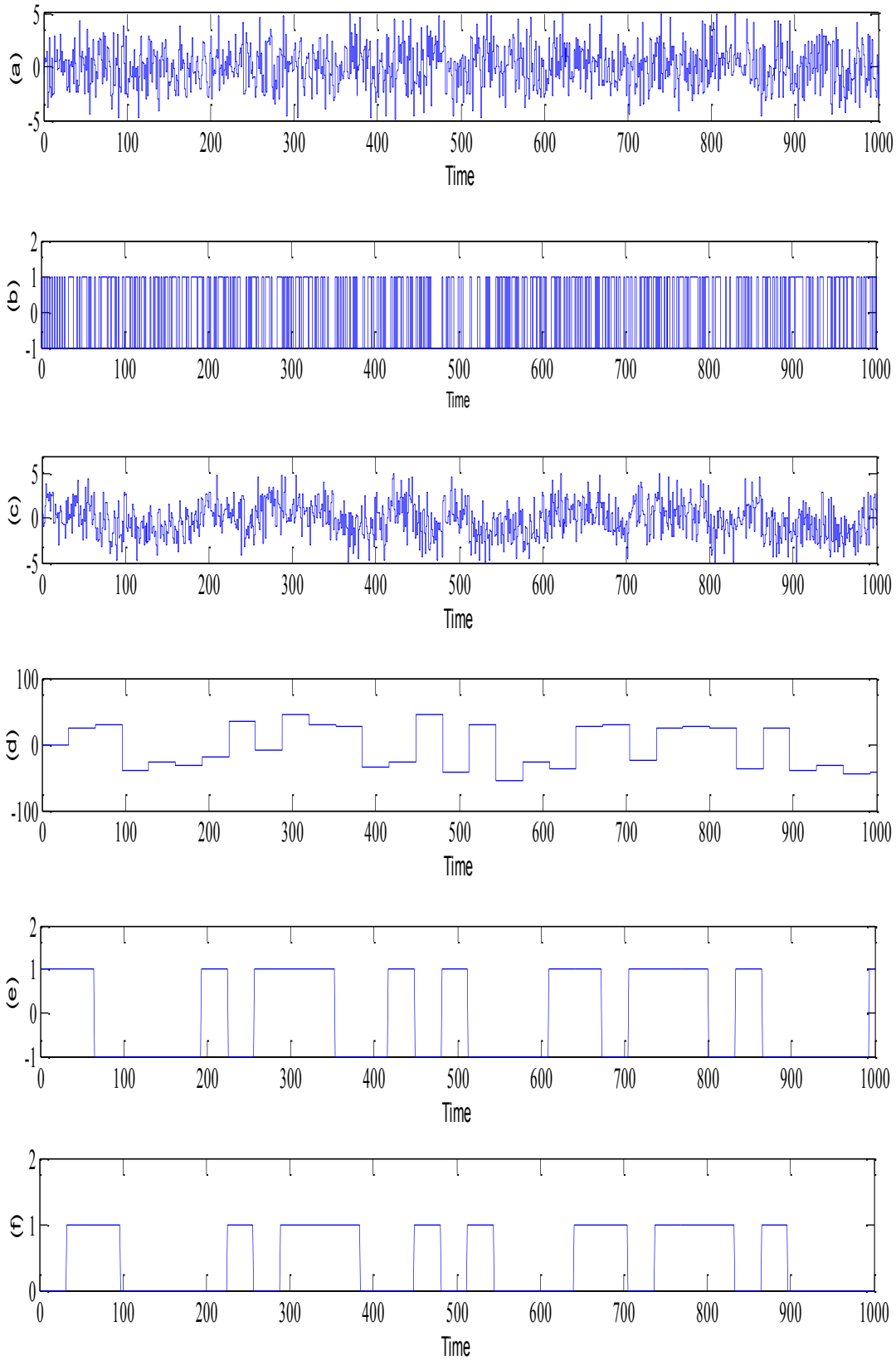

Fig. 3.36. User data extraction process.(a) Lorenz chaotic signal, (b) Multiplication process, (c) accumulator, (d) latched signal, (e) User data transmitted and (f) Recovered data.

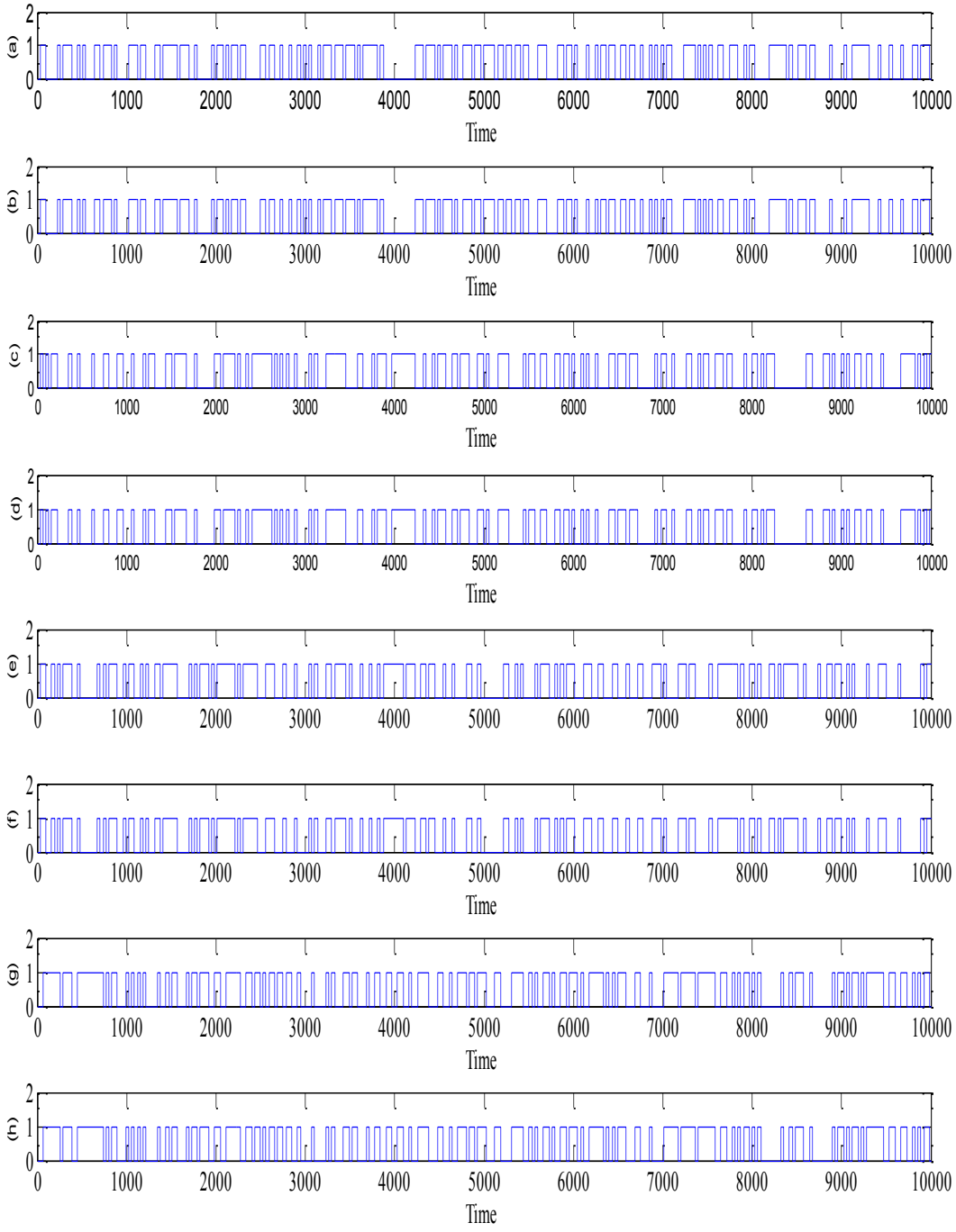

Fig. 3.37. Four user data transmitted and recovered. (a) User data 1, (b) user data 1 recovered, (c) user data 2, (d) user data 2 recovered, (e) user data 3, (f) user data 4 recovered, (g) user data 4, and (h) user data 4 recovered.

3.4 Communication System with Added Noise

One of the important tests to evaluate the system performance of the communication system is to add a noise in the channel. The Uniform Noise Generator (UNG)

SIMULINK block is used to generate uniformly distributed noise between upper and lower bounds in the system channel [113]. The system performance result-based simulation shows a graph of the Bit Error Rate (BER) versus the Signal-to-Noise Ratio (SNR) for multi-users. The UNG has four parameters: noise in the lower and upper bounds of the interval, initial seed value to generate a random number and sample time, which is a period of each sample based vector [113]. The received signal $r\ (t)$ is given by

$$r\ (t) = \ s(t) + \ n(t) + j(t) + c(t) \tag{16}$$

Where $s(t)$ is the desired signal and $n(t)$ is the noise from the channel and $j(t)$ is the jamming signal and $c(t)$ is the cross-talk noise from other users. The system with added noise is demonstrated in Fig. 3.38. The transmitter sub-system shows the all four user information data which are combined together using addition block. All four users' data are transmitted in one channel which is then contaminated by using the uniform noise generator. The power has been calculated before the noise has added and after the noise signal. Also, the signal to noise ratio has been calculated and displayed in dB. The receiver sub-system contains the data extraction process for each user. The bit error rate for each user has shown at the end using bit error rate block.

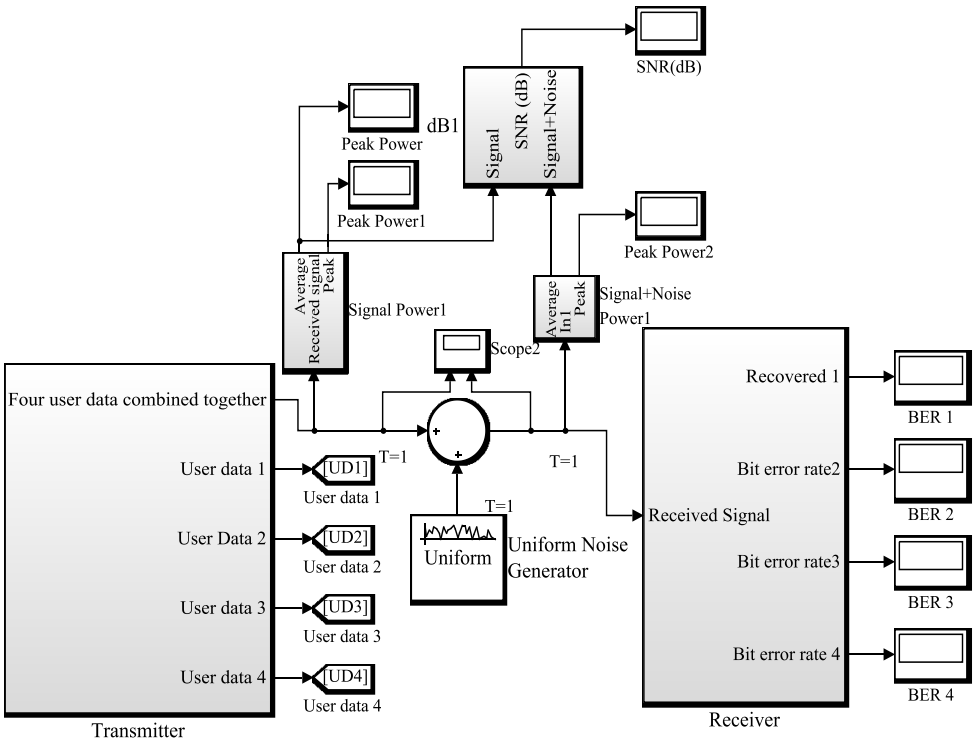

Fig. 3.38. The Lorenz stream cipher for four users with added noise, viewed in the SIMULINK.

The average and peak power of the signal in the input and output have been calculated as follows. The average peak power calculation process begins with using a square function block. After that, the mean is calculated with a mean block (the average power of the signal). The peak power calculation process begins with using the square function of the signal. Then a minimax block and integer delay blocks are implemented to give the peak power value of the signal. Fig. 3.39 shows the SIMULINK subsystem of the average and peak power. The same SIMULINK subsystem is used to calculate the peak and average power after the added noise.

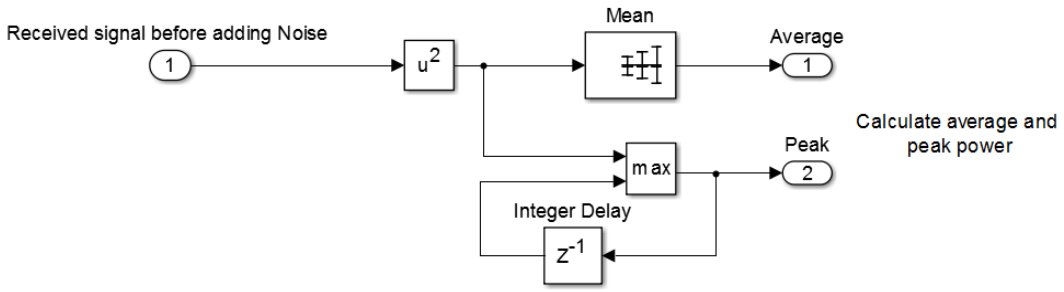

Fig. 3.39. SIMULINK block diagram of the average and peak power subsystem.

The SNR is calculated using the equation below, in which P_1 is the signal power and P_2 is the signal with added noise. Fig. 3.40 shows the SNR calculation subsystem.

$$SNR_{dB} = 10\log_{10}\left(\frac{P_1}{P_2 - P_1}\right) \tag{17}$$

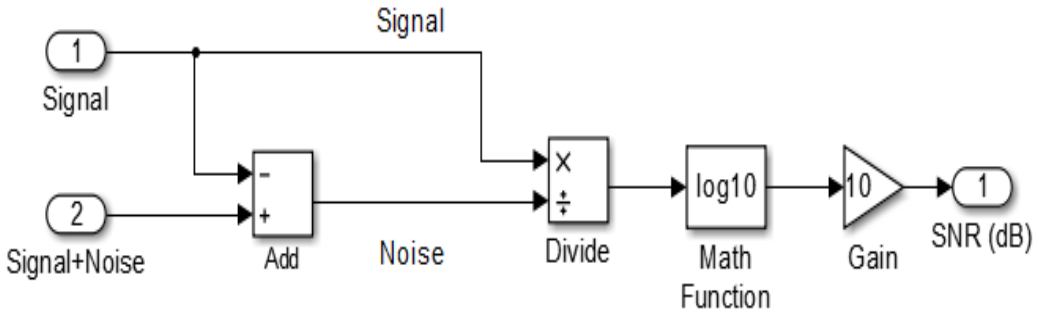

Fig. 3.40. SIMULINK block diagram of SNR calculation subsystem.

3.4.1 System performance

The Error Rate Calculation based on the SIMULINK block was used to compare the transmitted user data and recovered user data. The output result is the comparison based on the ratio between the two input signals, which in this case are the transmitted and recovered signals. Conversely, the bit error rate (BER) of the communication system is the number of bits recovered in error divided by the total number of bits transmitted. It is normal to have a time delay between the transmitted signal and the recovered signal; however, this delay must be determined to calculate the BER.

The performance of CDMA system has been analyzed in this research work. The performance results were evaluated in terms of Signal to Noise Ratio (SNR) of the CDMA system for four users. Results have been evaluated numerically and compared to standard accepted BER of 10^{-6}. In Fig. 3.41, the plot shows the upper noise bound versus the signal-to-noise ratio. The plot shows the SNR targeted at which noise bound.

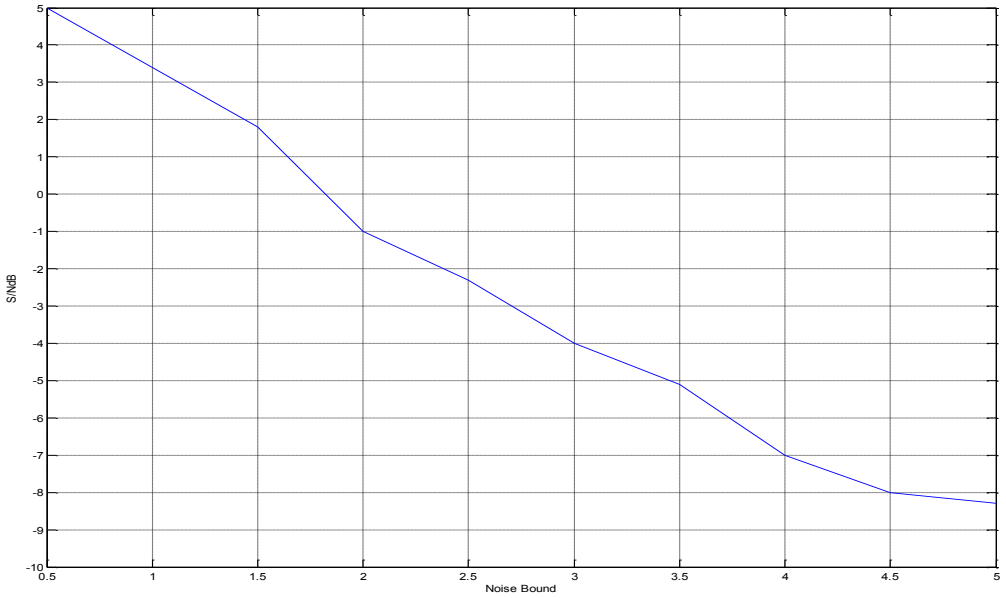

Fig. 3.41. Upper noise bound vs. S/N (dB).

In this simulation test, the number of bits transmitted was $1e^6$ for each user. The plot of BER versus Signal to Noise Ratio (SNR) is shown in Fig. 3.42. The system performance has achieved no bit error at the signal to noise ratio of -2.974dB. The system has achieved a good performance based on the results obtained and compared to other communication systems [114]. Table 3.5 shows the system performance.

Number of bit transmitted	S/N dB	User#1 bits error	User#2 bits error	User#3 bits error	User#4 bits error
$1e^6$	-0.20	0	0	0	0
$1e^6$	-2.97	0	0	0	0
$1e^6$	-3.27	0	18	9	2
$1e^6$	-3.56	2	32	12	2
$1e^6$	-3.97	5	63	22	4
$1e^6$	-4.36	8	103	38	12
$1e^6$	-5.09	14	210	116	44
$1e^6$	-6.57	112	800	600	249
$1e^6$	-7.48	294	1558	1150	497
$1e^6$	-8.30	600	2600	2000	1000
$1e^6$	-8.75	900	3402	2402	1294
$1e^6$	-9.04	1129	3981	2936	1530

Table 3.5. System performance.

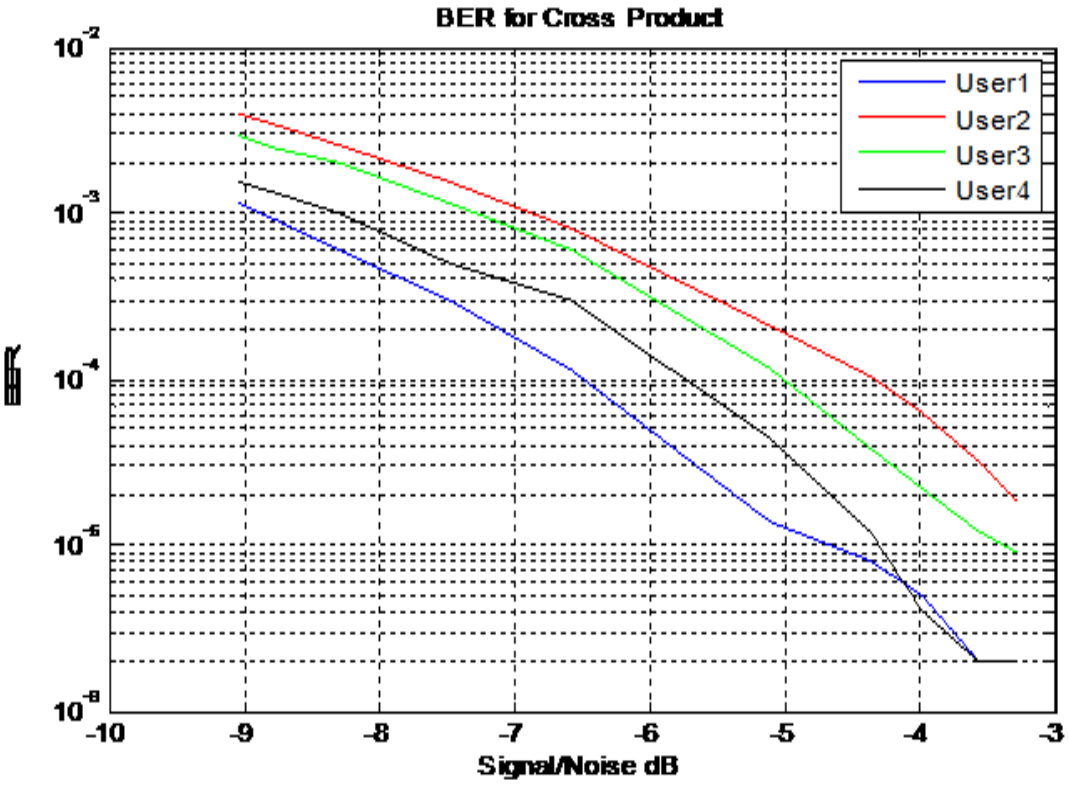

Fig. 3.42. BER vs. signal-to-noise ratio.

3.5 Conclusion

In this chapter a complete CDMA system based on Lorenz stream cipher has been described. The data encryption is based on two Lorenz generators (main and auxiliary). The auxiliary Lorenz generator serves to continuously vary one or more parameter(s) of the main Lorenz generator; in the case of the system described in this chapter, only the A parameter is varied. The data encryption uses a symmetric cipher which is a key length of 576-bits. This is a key space of the system is 2^{576}. The scrambling scheme was developed and Lorenz stream cipher binary stream passed the NIST randomness test successfully. In addition, the system output signal has a high sensitivity to small changes in any parameter. Moreover, the auto-correlation and cross correlation for 32-bits have good results. The maximum auto-correlation and cross-correlation functions for (32, 64, 128 and 256-bits) have shown good results. The auto-correlation of 32-bits was 32 and the cross-correlation value was 8. Moreover, when word length is longer, the auto-correlation and cross correlation are improved. In this chapter, the communication system was designed and tested for 32-bits. The aim of choosing the 32-bit was to reduce hardware resources consumed and to increase the data rate.

On the other hand, we have compared a cross-correlation of one Lorenz generator in terms of *x* and *y*, *x* and *z* and *y* and *z* and with two Lorenz generators with different parameters in terms *x* and *y*, *x* and *z, y* and *z*, we found that both of them are the same and result in low correlation.

Three different methods have been developed to extract the data at the receiver. De-spreading based on cross-correlation, on cross product and summation and de-spreading on dot product and summation. All three methods have shown the ability to extract the user data transmitted successfully. However, we used product and summation method which is easily converted to Xilinx ® System Generator blocks. In contrast, de-spreading based on cross correlation method is hard to be converted to Xilinx System Generator® because some SIMULINK blocks were not yet available.

The performance results were evaluated in terms of Signal to Noise Ratio (SNR) of the CDMA system for four users. Results have been evaluated numerically and compared to standard accepted BER of 10^{-6}. The system performance were shown good results. At -2.974 signal to noise ratio, the system achieved no bit error with $1e^{6}$

bits transmitted for four users. The system has achieved a good performance based on the results obtained and compared to other communication systems [114].

Chapter 4

DESIGN METHODOLOGY, CLOCK AND DATA RECOVERY AND SYNCHRONISATION OF CHAOTIC SIGNALS

4.1 FPGA Technology Features

The FPGA technology comes in the middle between the Application Specific Integrated Circuit (ASIC) and Digital signal processor (DSP). The ASIC technology can be used for this project. However, to configure these devices, it must be sent to the manufacturer, which increases the cost and consumes time. The drawbacks of the DSP technology are that they occupy maximum area, the power consumption is high.

Field-programmable gate array (FPGA) is a reprogrammable device that has attracted interest amongst researchers and engineers involved in a diverse range of fields because of its:

- Computational performance (the data processed per given unit of time) without compromising on (hardware) resource efficiency [115].
- Low cost.
- Short time to market.
- Reliability and robustness as it does not require short/mid-term maintenance.

The proposed cryptosystem falls within the communication security branch of FPGA applications. FPGAs have a number of important advantages for implementation of cryptosystems:

1- Fast execution of arithmetics on real numbers (for actualizing the Lorenz model) whilst maintaining a suitable degree of numeric precision [116].
2- Resource efficiency. Advantage over processor-based implementations (requiring O/S or kernel) because of the ability to maintain time-accuracy of the execution of processes [116].
3- FPGAs consist of logic blocks with specialized architectures designed specifically for high-speed digital signal processing (e.g. DSP48A1a multiplier).

4- Software-based implementation of the same cryptosystem would not guarantee a sufficient level of time-accuracy of any particular process that constitutes the system because of hardware (number of registers; data caching) and software constraints (kernel requires periodic access to CPU core, interrupts, other threads, and processes operating on the OS). (Each CPU is bound by the fetch-execute cycle whereby instructions within the program are fetched and executed on a particular (small) set of data; the CPU itself is limited to a certain number of registers on which data is held for execution for that particular instruction).

5- Every process that constitutes the system is actualized as a dedicated microarchitecture that cannot be interrupted by an external signal or process. Furthermore, FPGAs are designed for implementation of systems *synchronous* to a single system clock via registers, and thus, execution-time of any particular sub-process of the system can be guaranteed to a particular level of accuracy.

6- Modern communication systems define a maximum data rate; all constituting hardware must be conducive for operating at this data rate (e.g. 1 Gbps Ethernet). Hence, the guarantee of the execution-time of any particular process is important. The project required fast and accurate executions especially of the nonlinear system (Lorenz model) that is used to generate the cipher stream for cryptography applications. The FPGA technology plays a significant role in this implementation.

4.2 Design Methodology

The design proceeds in three steps. MATLAB-SIMULINK, System Generator® and Integrated System Environment (ISE). Fig. 4.1 shows the basic block diagram of the system design process. First, we designed the system by SIMULINK model. The system has been tested until the desired results have been obtained. After that, the SIMULINK blocks have been converted into Xilinx © system generator. The system has been tested again until the desired results have been obtained. The ISE© is then has been used for the configuration and implementation process. The final step is to generate the bit file and download it into the target FPGA board.

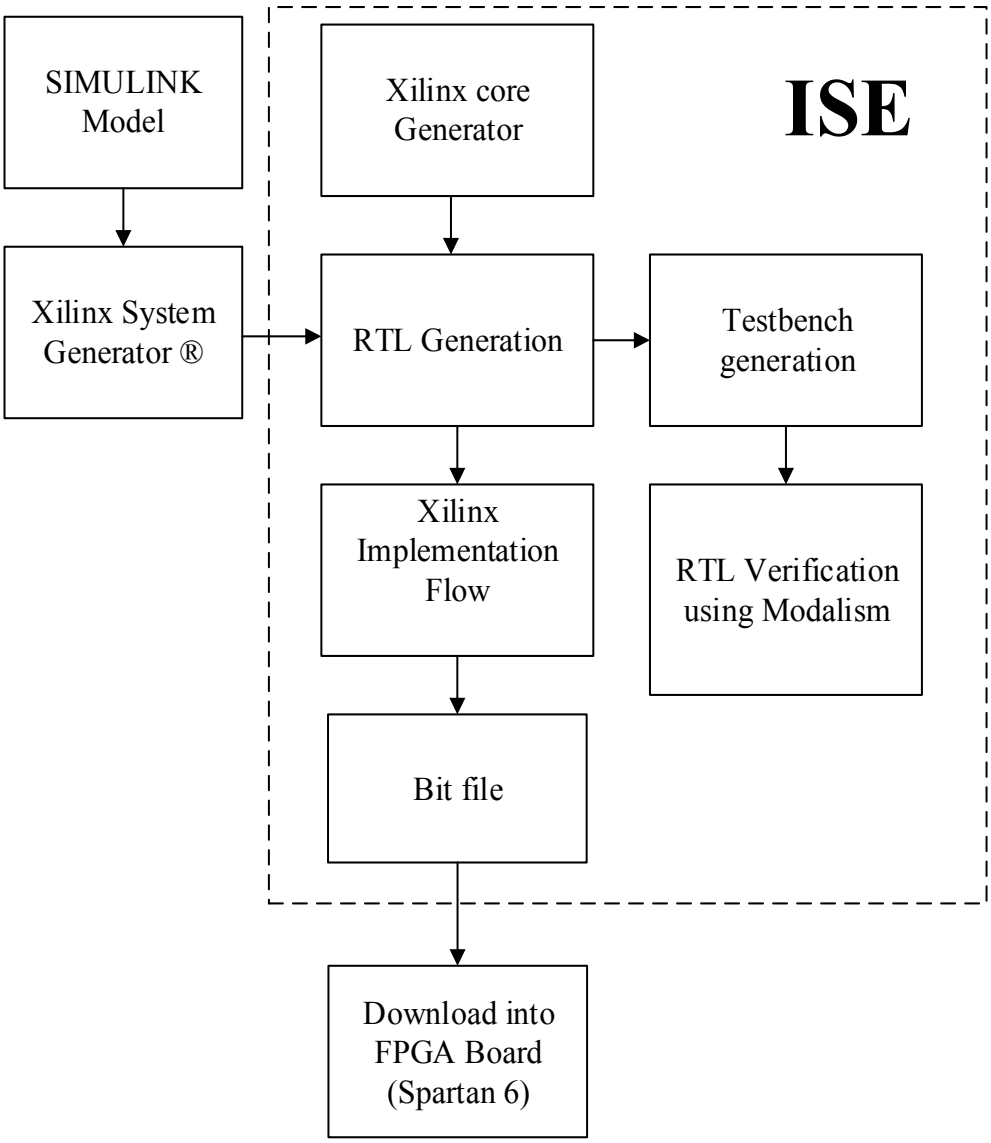

Fig. 4.1. Basic block diagram of the design flow.

This section describes the hardware implementation of the complete communication system. The procedures of the design and implementation are given below.

- SIMULINK based Matlab environment has been used to test and analyse the communication system as described in chapter 3. In this step, the system performance is optimized and the desired results are obtained.
- Some SIMULINK blocks have been developed to be compatible with Xilinx blocks.
- Some SIMULINK blocks are not yet available in the Xilinx System Generator® such as a buffer. Thus, Xilinx blocks have been developed.

- VHDL is then generated from the System Generator® for different designs such as Lorenz system, clock recovery, data recovery and synchronization of chaotic signals.
- ISE ® is then used to synthesize, configuration of the target device. The program file is generated for a specific device chosen by the designer. This file can be downloaded into the FPGA board through the computer.
- Finally the system is tested and the results are analysed and compared with the simulation results.

4.2.1 Software tools

The software tools used for system design and analysis are summarised in this section.

4.2.2 MATLAB

MATLAB is an interactive software for high performance languages for numerical computations. It is easy to use for integrates computation, visualization and programing. MATLAB can be used in different scientific areas such as signal processing, control system, neural network image processing and video processing. Moreover, MATLAB can be used for mathematical computations, algorithm development, modelling, simulation, data analysis, exploration, visualisation and scientific and engineering graphics.

4.2.3 SIMULINK

SIMULINK is a software package that is compatible with MATLAB's environment for modelling, simulating and analysing dynamics system. SIMULINK supports linear and nonlinear systems which are modelled in continuous time, sampled time or hybrid. Moreover, it is a high level language that uses block diagrams using click and drag. SIMULINK provides a Graphical User Interface for building models. When the model has been designed, the SIMULINK tool allows testing the model in a fast and flexible way and makes sure that the system performance is as expected. Fig. 4.2 shows an example of the SIMULINK model.

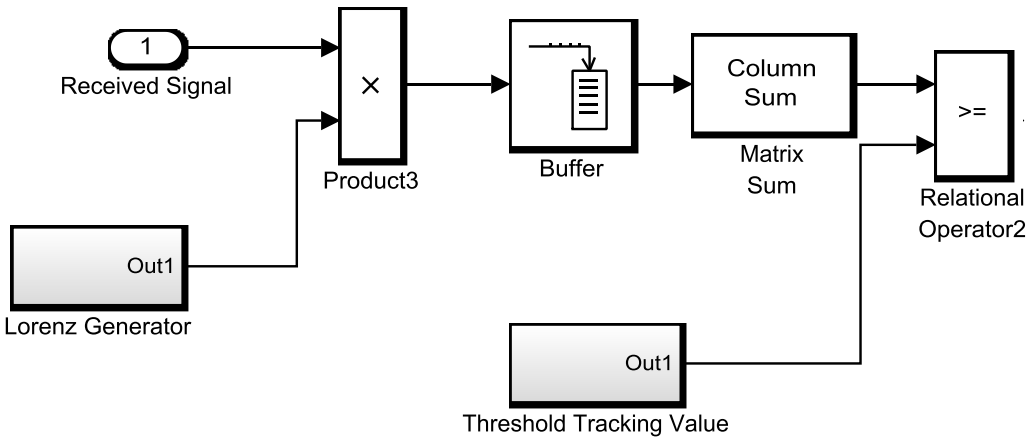

Fig. 4.2. SIMULINK blocks.

Two main tools have been used to implement the communication system successfully, and upload the synthesised hardware logic bit file onto the target FPGA.

1- Xilinx System Generator®.
2- Integrated Synthesis Environment (Xilinx ISE).

4.2.4 Xilinx System Generator®

The System Generator® is provided by the Xilinx Company that is one of the leading FPGA manufacturers. The System Generator is a design environment for FPGAs. It is a high-level development tool that provides a platform for the design of high performance DSP systems and the testing of the model. In addition, the System Generator offers numerous system blocks (abstraction of particular Xilinx logic blocks of particular architectures within the FPGA device for which VHDL code is generated [117].

In this work, Xilinx System Generator has been used to model, simulate and analyse the system design. The main advantage of the System Generator® tool is minimizing the time spent on the description and simulation of the circuit. Furthermore, the flexibility of the design allows the system parameters to be easily changed and optimises design performance. In addition, the System Generator allows for functional simulation before compilation of the model designed. When the functional simulation meets the desired target, the files of the structural description of the system in standard hardware description language are generated by compilation. These files are used in the Integrated System Environment (ISE) for Xilinx FPGAs. Gateway In and Gateway

Out blocks are used as FPGA boundaries in the System Generator model. The purpose of the Gateway In block is to convert SIMULINK floating points into a fixed point format. The designer can also define the type of quantization such as saturation and rounding modes. The Gateway Out block converts the FPGA fixed point format into double precisions floating point format.

In the System Generator, the signals are changed automatically because of operators that force it to be in a suitable format (a signed or unsigned fixed point) in the outputs. Furthermore, the designer is allowed to include any functional system to the design that is described in VHDL or Verilog by using a Black Box Block.

The simulation test of the model in the System Generator tool is bit and cycle accurate. This means that the results of the simulation are exactly the same as the result in real time implementation. Since the System Generator uses a discrete time system, the sample rate is automatically generated from the signals and blocks. Furthermore, System Generator supports multi-rate designs. Fig. 4.3 shows the Xilinx blocks that have been converted from SIMULINK blocks shown above in Fig. 4.2.

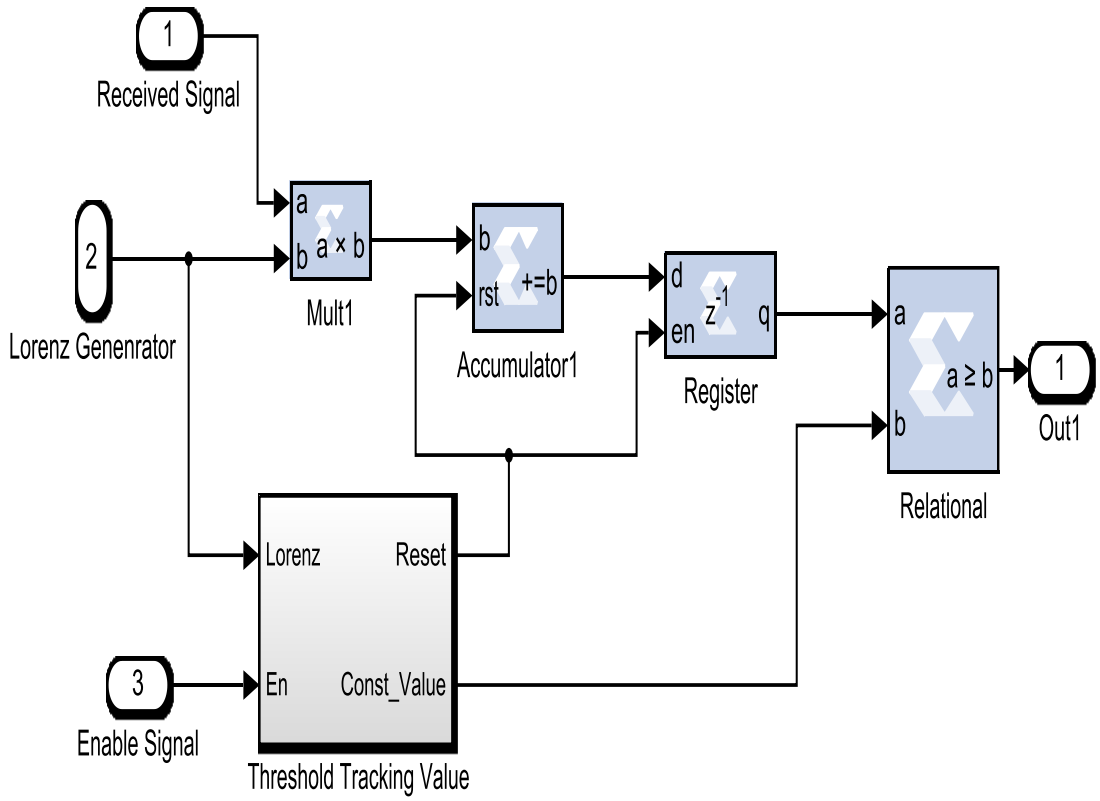

Fig. 4.3. Xilinx blocks.

4.2.5 Integrated System Environment (ISE®)

ISE is a design suite for the Windows environment that allows the configuration of the target device for a particular application. The primary user interface of ISE is the Project Navigator which provides full control of the design entry, synthesis, implementation of the design, functional simulation, testing, and verification [118]. The Project Navigator Interface consists of four panel sub-windows, as shown in Fig. 4.4. On the top left are the Start, Design, Files and Libraries panels that include display and access to the sources files in the project. The start panel provides access to opening projects, reference material and documentation.

Fig. 4.4 shows the design panel that provides access to the View, Hierarchy and Process panes. The View pane enables the user to view the source modules associated with the implementation or simulation design View in the Hierarchy pane. The Hierarchy pane shows the project name, the target device, user document and design source files associated with the selected design view. Each file in the hierarchy pane has an associated icon. The icon indicates the file type such as HDL file, schematic or core. From the process pane, the user can run functions necessary to define, run and analyse the design such as design summary/reports, design utilities, user constraints, synthesis, implement design, generate programming file and configure target device.

The design summary provides a summary of the main design information and also access to all messages and detailed reports from the synthesis and implementation tools. The high level information about the project can be provided from the summary such as overview information, a device utilisation summary, performance data report, constraints information and summary information from all reports.

Workspace contains design editor, viewer and analysis tools. These include ISE test Editor, schematic editor, constraint editor, design summary/report viewer, RTL, technology viewer and timing analyser. The user constraints are used to view floor plan areas, IO pins and Logic of the chip. The post-synthesis can be chosen from user constraints.

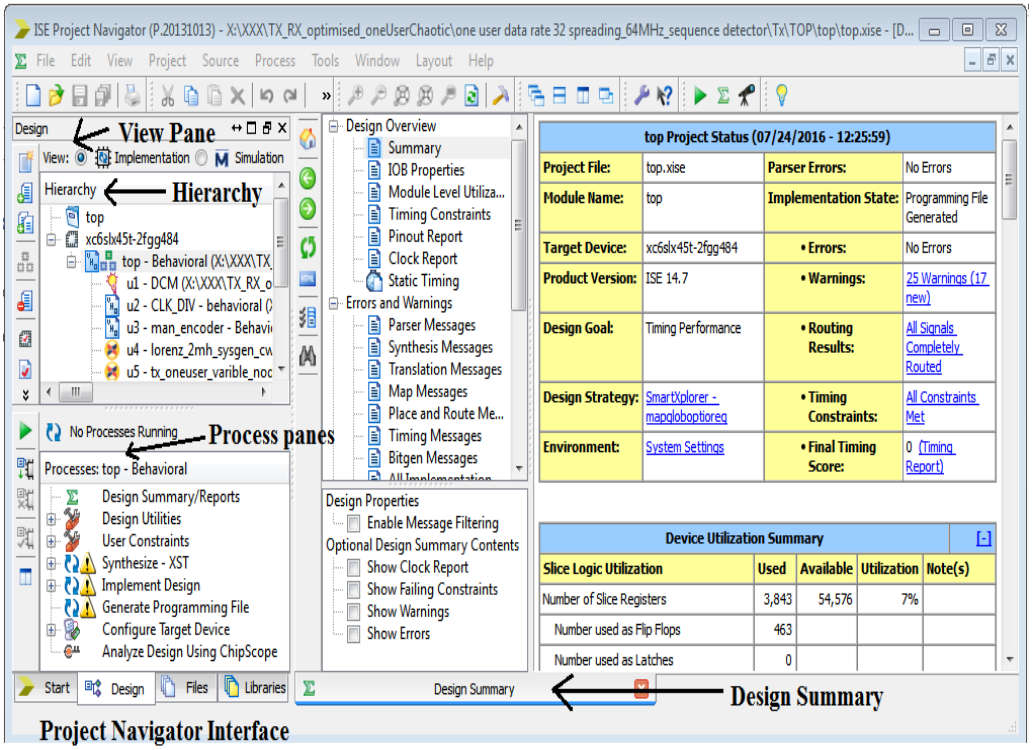

Fig. 4.4. ISE project.

ISE allows for several tasks such as: compilation of the hardware description language files, simulation of system behavioural or timing analysis. In addition, it allows to check the occupancy rate, power consumption. The final task that the ISE tool does is to generate the program file for a specific device chosen by the designer. This file can be downloaded into the FPGA board through the computer. Then, the hardware test results can be checked using electronic measuring equipment such as an oscilloscope.

The structural description files can be obtained from the System Generator® model when 'Generate' in dialog is clicked. Thus, a project is created and ready to use in Integrated System Environment. After that, the syntax of HDL files are available to check. The compilation process is started by synthesizing the system. The aim of this type of process is to create Xilinx specific netlist files called NGC (Native Generic Circuit) files. These files contain both logical design data and constraints. There are two options that can be chosen for optimization in terms of normal and high effort area or speed. The default option is set to speed optimisation. After the synthesis has taken place, a synthesis report is created for the designer to be analyzed. The RTL view is a Register Transfer Level graphical representation of the design. The synthesis tool is

responsible for this step. The aim of the RTL view is to reflect the original HDL code of the design before the technology mapping takes place. Thus, it helps to discover design issues early in the design process. Fig. 4.5 shows the RTL schematic as an example for Digital clock management (DCM) which shows the input signal and output signal of the DCM component.

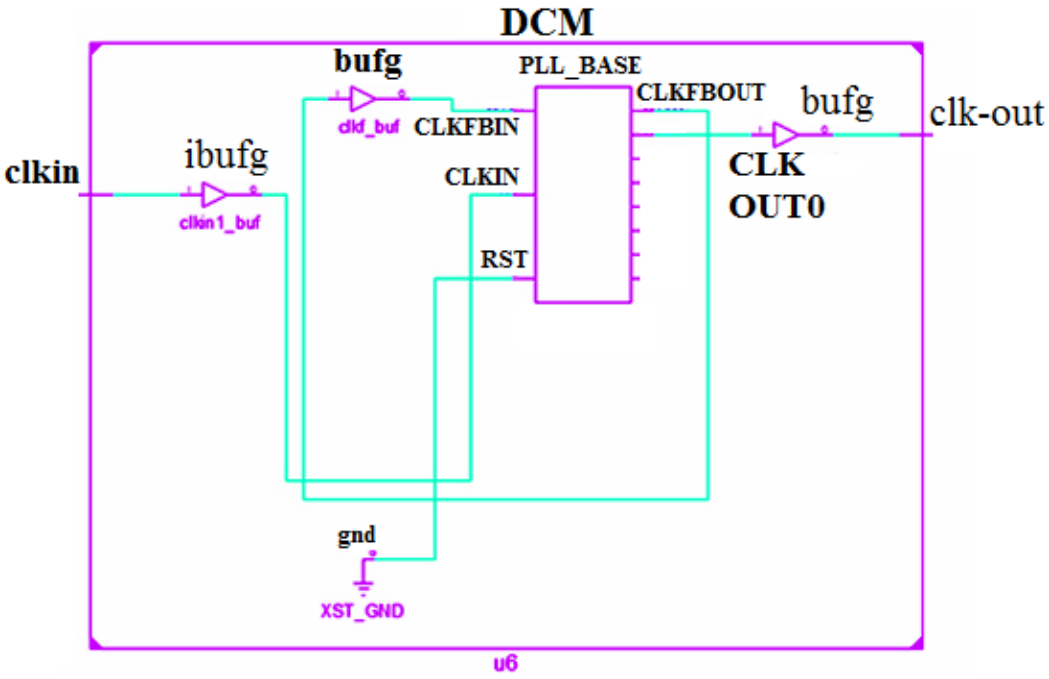

Fig. 4.5. An example of the RTL schematic for the PLL.

When the synthesis is done, four stages are required for implementation: translate, map, place and route. The purpose of the translation process is to create a Xilinx Native Generic Database (NGD) file from the all netlists and design constraints data. After that, the targeted FPGA device can be mapped using the NGD file. Some mapping process is done for NFD file that include design rule checker and then maps the logic design to the Xilinx FPGA device. A Native Circuit Design (NCD) file is created from the previous process that is used for place and route stages. NCD file is used by place and route to generate a new NCD file that is used by the programming file generator. Then, bit stream generation program (BitGen) runs by using generator programming file to produce a bit file. This file is used for Xilinx device configuration. Now, the bit file is used to configure the FPGA target device through the configuration target device process.

Xilinx Core generator system provides access to highly parametrised intellectual properties (IP) for Xilinx FPGAs which can accelerate the design time. Fig. 4.6 shows the Xilinx core generator.

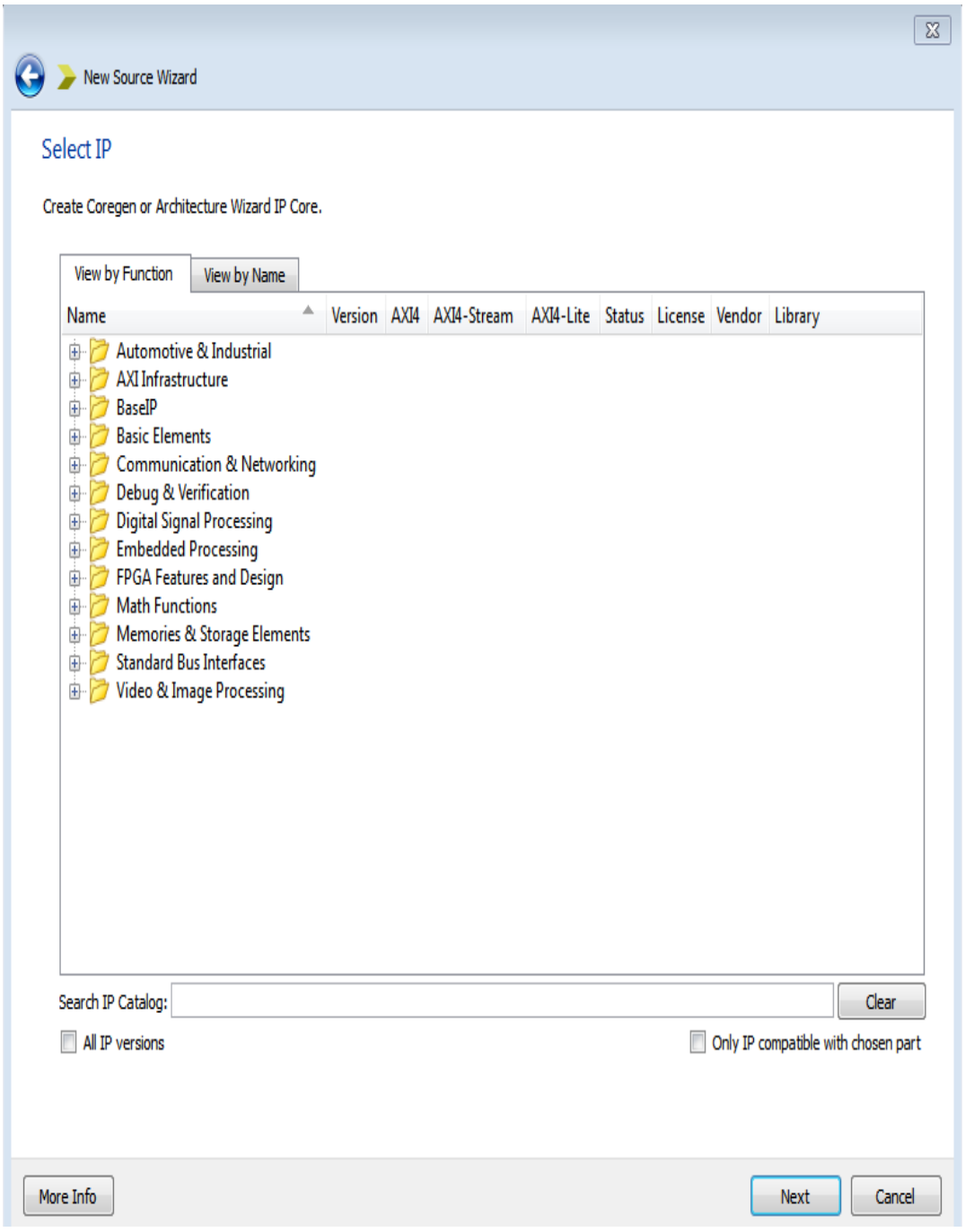

Fig. 4.6. Xilinx core generator.

The last step is to generate the bit file and then configure the desired board. After that, the bit file is downloaded into the FPGA board. Fig. 4.7 shows the final step.

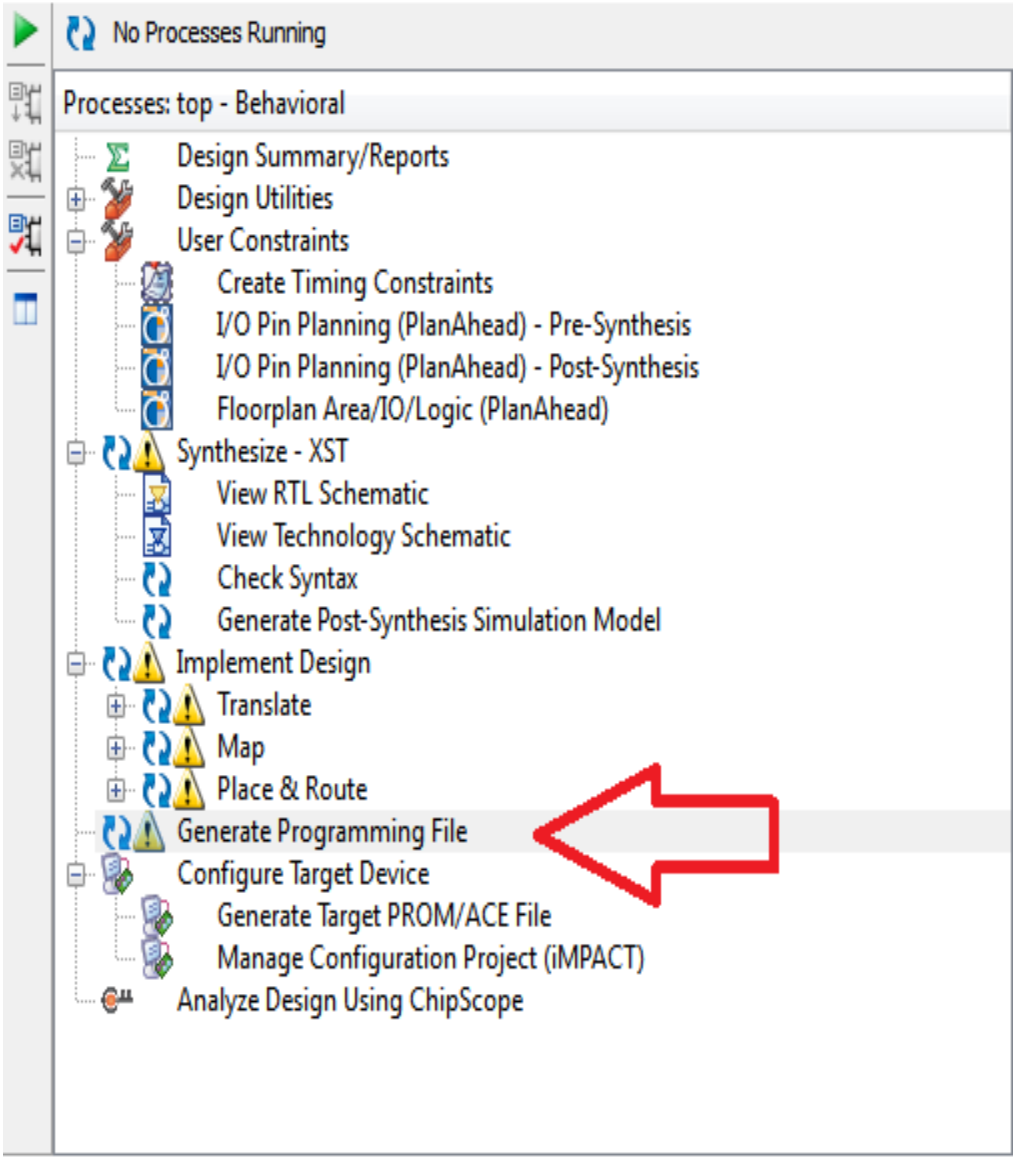

Fig. 4.7. Generates Programing file (bit file).

4.2.6 XILINX-FPGA boards

The board that has been used in this work is the SP605 board. The chip name of the Spartan ®-6 is XC6SLX45T-3FGG484 FPGA. The SP605 provides board features such as 128 MB DDR3 component memory, SPIx4 Flash, USB JTAG, Clock generation SMA, PCI etc. The SP605 board contains four physical headers, additional user desired features can be added through mezzanine cards that can be connected to the board. There are two types of oscillators, fixed 200 MHz (differential) and socket with a 2.5V 27 MHz (single-ended). Two SMA connections can be used, A3 and B3 pins. In this application two SP605 boards have been used, one for the transmitter and the second board for the receiver. Fig. 4.8 shows the two SP605 boards.

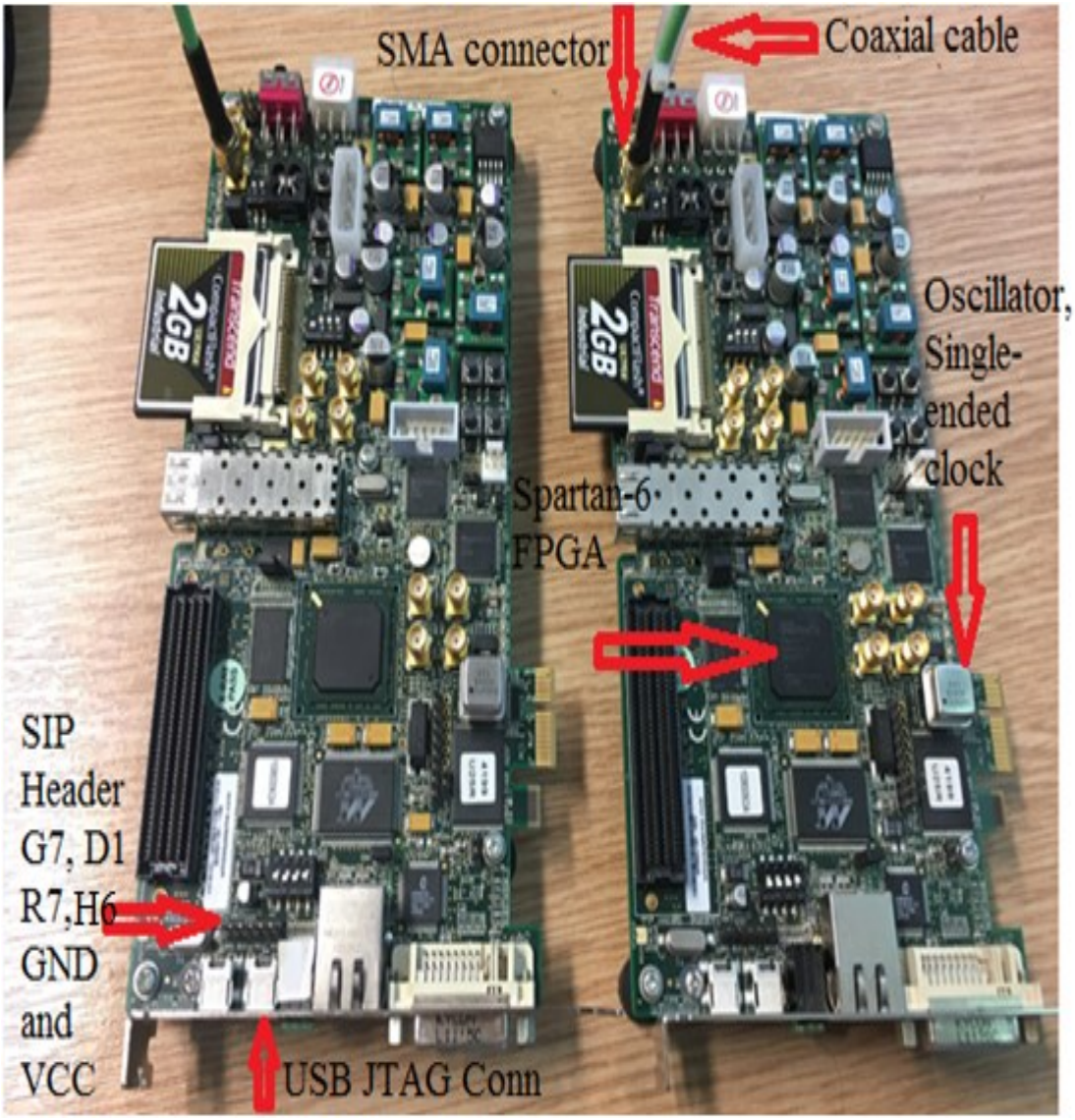

Fig. 4.8. SP605 boards, transmitter and receiver.

4.3 Clock and Data Recovery (CDR)

In serial digital data transmission, the local clock at the receiver must be adjusted in terms of frequency and phase to the clock rate of the incoming signals [119]. The purpose of the recovered clock is to re-time the incoming serial data, so the received signal can be processed synchronously and the data transmitted can be properly retrieved [120]. From a hardware point of view, when the sampling clock pulse occurs at the midpoint of each received bit interval, an accurate detection must be high and the bit error rate low [121]. Fig. 4.9 shows the fundamental block diagram of clock recovery and data retiming for data recovery.

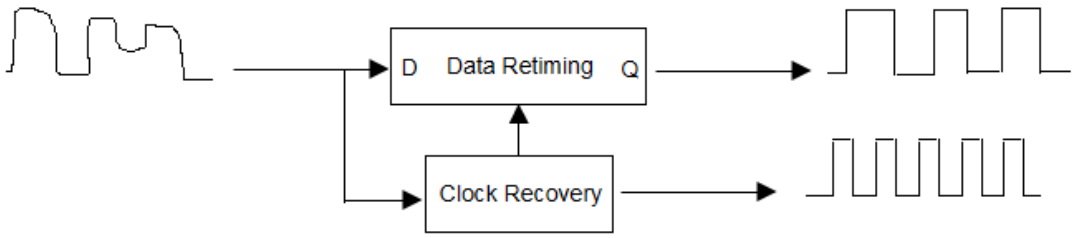

Fig. 4.9. Clock recovery and data retiming for a CDR.

Since the clock is embedded in the high speed serial data received signal, CDR is one of the most challenging implementations in real time applications. The receiver system collects information signals that are affected by natural weather or artificial noises as well as its own asynchronous signal [122] [123].

A spectral component of the clock frequency is missing in NRZ data. Thus, the clock signal frequency and phase must appear within the signal to recover the clock. A prior art approach that solves this issue is represented in a basic block diagram in Fig. 4.10. the received signal is delayed by one sample and then XORing with received signal. The aim of this process is to detect the rising and falling edge of the received signal. After that, the output signal of the XORing process enters the bandpass filter in order to capture a desired frequency. When the desired frequency has been produced, the next step is to lock it using phase locked loop. This approach involves producing a frequency component for the NRZ clock frequency and the received non-linearity by using a delay and an Exclusive OR gate with an introduced frequency for the bandpass filter. The bandpass filter is used to detect the introduced frequency component at the clock frequency and produces a clock that helps to lock the frequency using a Phase Locked Loop (PLL) [124]. The purpose of the PLL is to follow or track the bit transition pulse rates of the received signal. In other words, the PLL allows for phase detecting that enables the local oscillator to follow both the phase and frequency of the received data stream [125].

On the other hand, when the binary data stream alternates between ones and zeros (such as 101010), this represents the maximum frequency of the binary data stream, a clock frequency that is twice that of the maximum data rate. In some systems, such as spread spectrum technology, the clock signal is sent with the data. This kind of system is designed to process at twice the speed of the data rate in order to carry the clock

signal. When a data rate is 100 Mbps, the system clock signal rate must be 200 Mbps [126]. This of course reduces the channel efficiency. When the system operates on high clock rates, this will lead to attendant noise and power consumption [127].

In this work, the clock recovery consists of three main tasks:

- Phase detection of the received signal.
- Clock recovery.
- Clock locking.

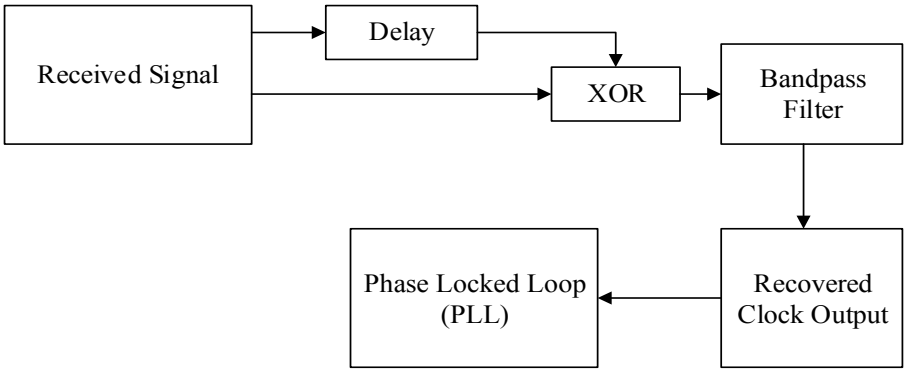

Fig. 4.10. The basic blocks diagram of the clock recovery.

First, a Bernoulli binary generator is used as a random data generator. Fig. 4.11 shows that the clock is missing in FFT plots of the random data. The time scale in the results (refers to 3.2.2).

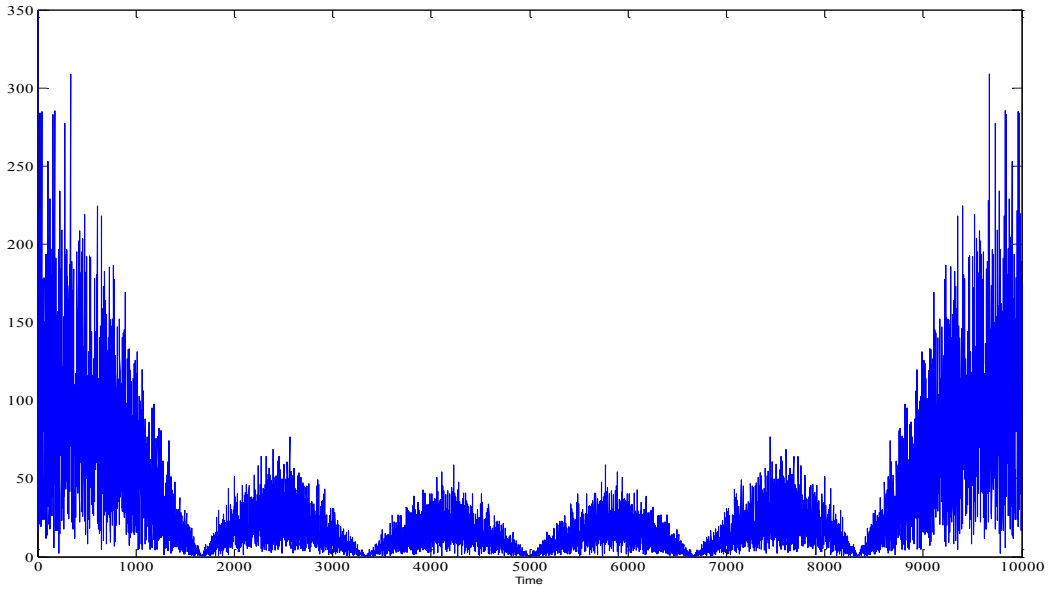

Fig. 4.11. FFT of the random signal (Bernoulli binary generator).

4.3.1 Phase detector

A phase detector is used to detect the phase difference between two data signals. A phase detector is simply an Exclusive OR gate that compares the two signals to locate the data transition. The output signal depends on variation between the two input signals [128]. Fig. 4.12 shows the rising and falling edges detected. The FFT function shows that the clock signal and its harmonics now appear in the signal as shown in Fig. 4.13.

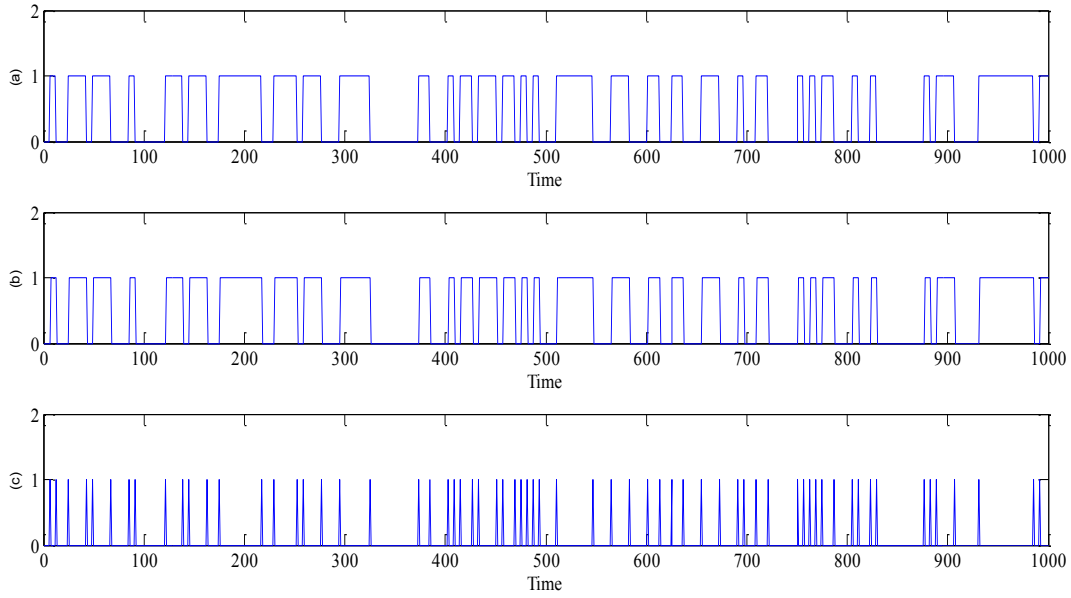

Fig. 4.12. Simulated test, (a) Binary random generator (Bernoulli), (b) Delayed data by one sample, (c) Rising and falling edge detections.

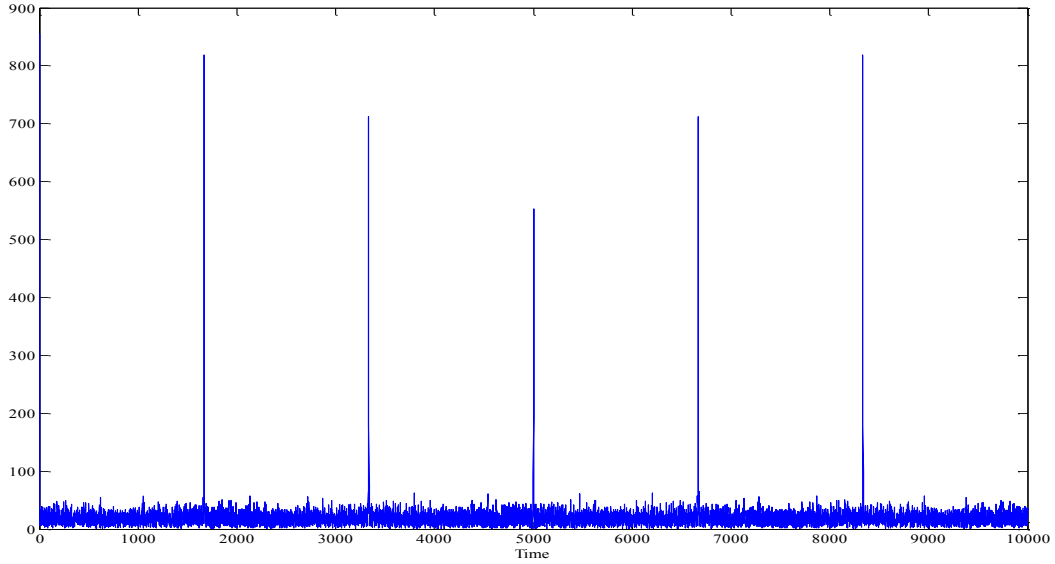

Fig. 4.13. FFT after delay multiply techniques are applied on the Bernoulli generator.

4.3.2 Clock recovery design based on SIMULINK

When the phase of the signal is detected, a bandpass filter is used to recover the desired frequency. The centre frequency is the clock frequency from the transmitting side. Fig. 4.14 shows the clock recovery based on the SIMULINK. Fig. 4.15 shows the clock recovery from the random data signal.

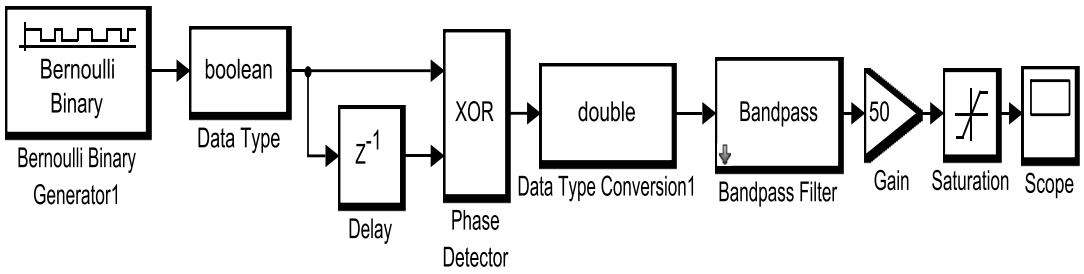

Fig. 4.14. Clock recovery as viewed in the SIMULINK.

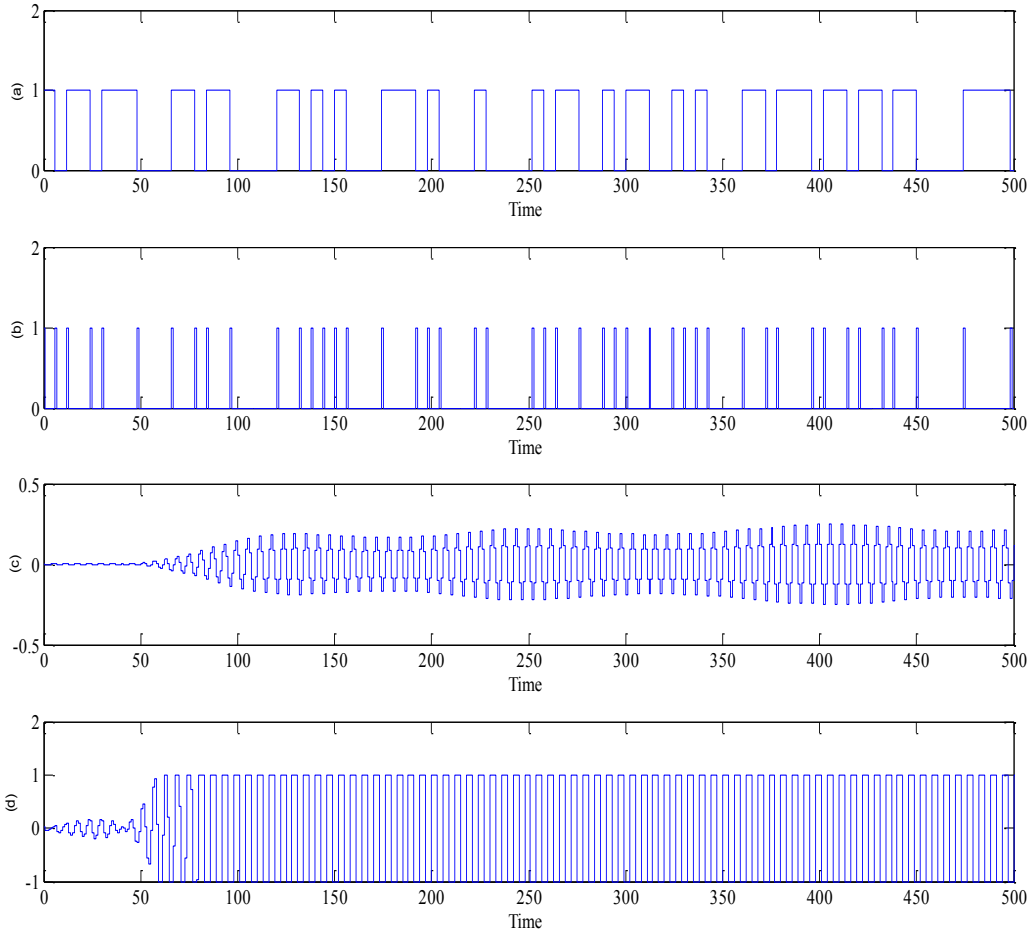

Fig. 4.15. Simulated test of clock recovery based on SIMULINK, (a) Binary random generator (Bernoulli), (b) Rising and falling edges detectors, (c) Bandpass filter response and (d) recovered clock.

4.4 Real time implementation of the clock recovery

The SIMULINK model of the clock recovery is converted to System Generator® model as shown in Fig. 4.16.

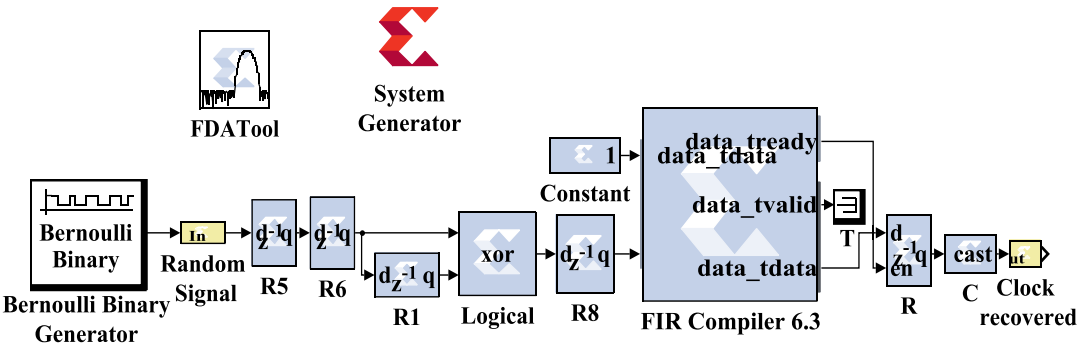

Fig. 4.16. Clock recovery technique as viewed in the Xilinx System Generator®.

A Bernoulli Generator is used as a random data generator to test the ability of the clock recovery model. The first two registers are used for sync purposes and in order to detect the phase of the received signal, the received signal XORing with the delayed signal by using a register block. Finite Impulse Response (FIR) Filter is then used to recover the desired frequency by using the Bandpass Response Type. The bandpass filter bandwidth is selected based on the clock of the incoming stream signal [119]. The FIR filter is digitally implemented by using a series of delays, multipliers, and adders to create the desired output. Fig. 4.17 shows the basic diagram of an FIR filter of length N. The h_k values are the coefficients used for the multiplication, as the result of the output at time n is the summation of all delayed samples multiplied by appropriate coefficients.

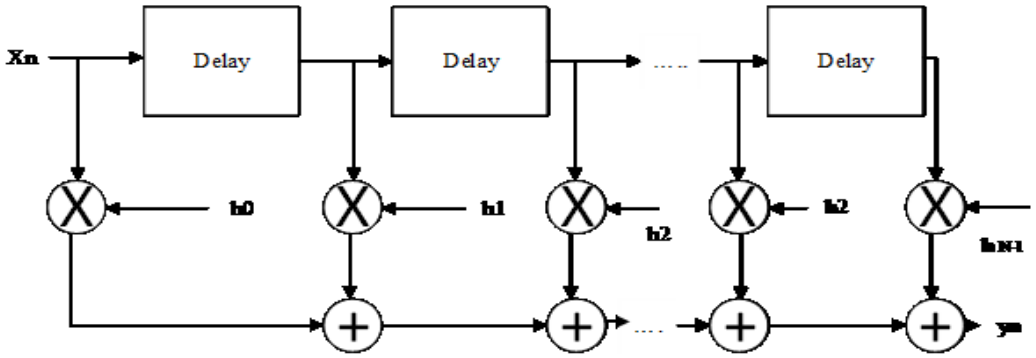

Fig. 4.17. The basic diagram of an FIR filter process.

FDA tool is a MATLAB basic tool that is used to design a filter of required specifications. Different response types such as high pass, low pass, and bandpass can be selected. The design method for the filter implementation can be IIR and FIR. The FDA tool provides windows which can be customised by providing such as order of the filter, cut-off, sampling, pass-band and stop-band frequencies and magnitude specifications. When the specifications of the filter are selected, the tool creates coefficients which is save as matrix in MATLAB workspace [129]. The FIR filter is able to recover the desired frequency from the random data generator. Fig. 4.18 shows that the clock is recovered from the random data generator.

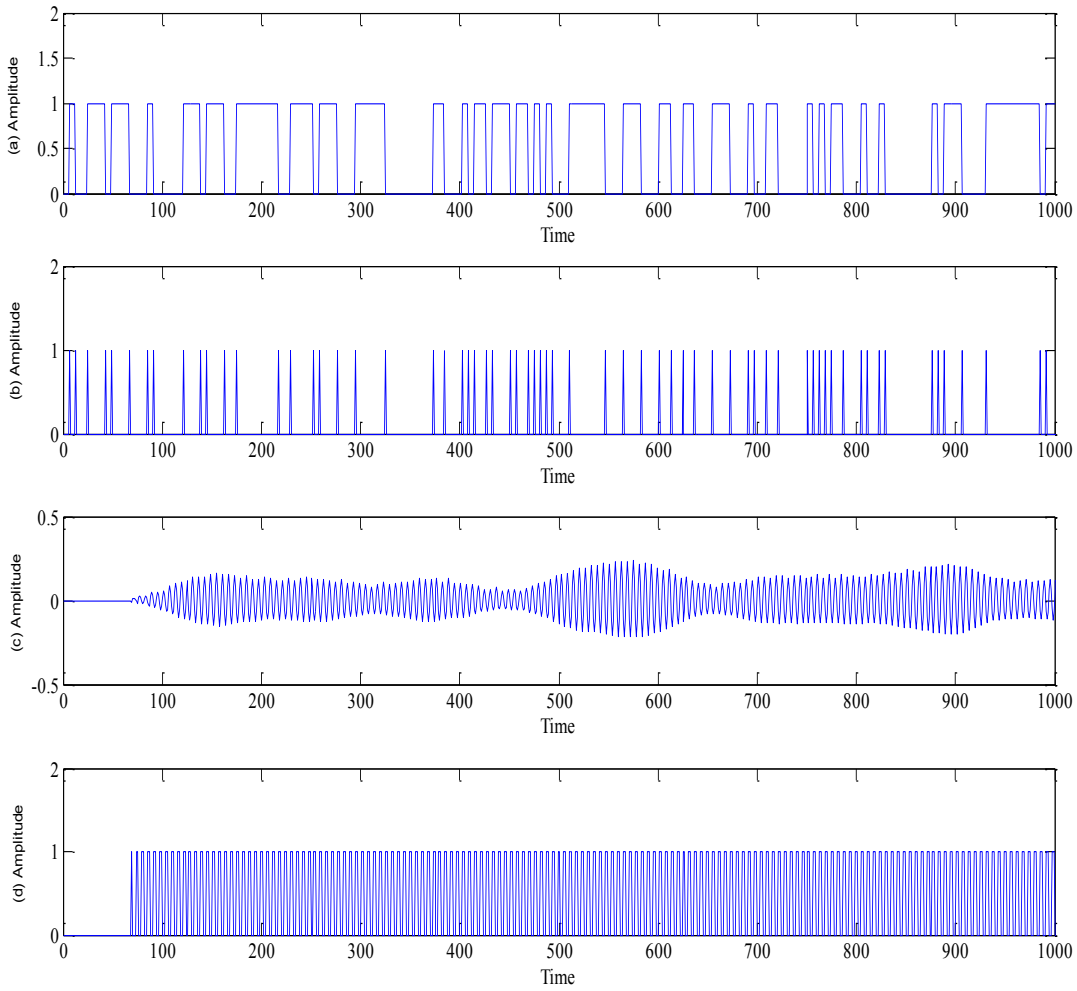

Fig. 4.18. Simulated test of the clock recovery based on System Generator®, (a) Bernoulli random generator, (b) Rising and falling edge detections, (c) FIR filter response and recovered clock.

The main clock that is used at the receiver FPGA board is an oscillator-socket single-ended, Low Voltage Complementary Metal Oxide Semiconductor (LVCMOS). The clock rate is 20MHz. A clocking wizard is used to set up a desired clock frequency.

Fig. 4.19 shows the basic blocks diagram of the clock recovery process.

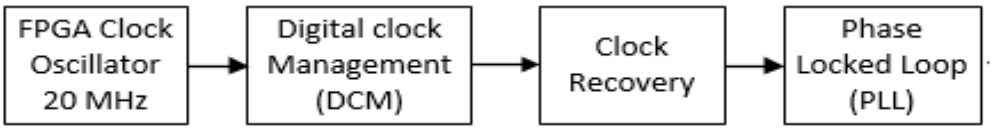

Fig. 4.19. Basic blocks of the clock recovery process.

The received signal frequency is 65 MHz, and since the clock recovery uses double the received signal frequency (130 MHz), the Digital Clock Management (DCM) is used to feed the FIR filter for a clock rate of 390 MHz (triple the clock rate of 130 MHz frequency) to be able to recover the desired frequency. The FIR filter specifications are listed below.

- Sampling Frequency (Fs) = 390 MHz
- Fstop1 = 100MHz
- Fpass1 = 129 MHz
- Fpass2 = 130 MHz
- Fstop2 = 160 MHz
- Attenuation on both sides of the passband = 120 dB
- Pass band ripple = 1.
- Filter's order is 48.
- Design method is FIR Equiripple.

The FIR filter is specified, simulated, and VHDL has generated using the System Generator®. Fig. 4.20 shows the clock recovery. The received signal has been synchronised using two registers. After that, the output signal has been delayed using register block. The received signal XORing with the delayed signal and the output enters to the FIR filter to capture the desired frequency. Fig. 4.21 shows the RTL. The Register Transfer Level view represents the graphical design. There are two main components, Digital clock management (DCM) which used to produce a clock rate of 390MHz that is used for sampling in FIR filter. The clock in for the DCM is 20MHz. The output frequency from the FIR filter is 130MHz clock rate which is used for the

rest of the system. Fig. 4.22 shows the components inside the clock recovery. The RTL of the clock recovery shows the components the represents the clock recovery. These components represent the XILINX © system generator design that are shown in Fig. 4.20.

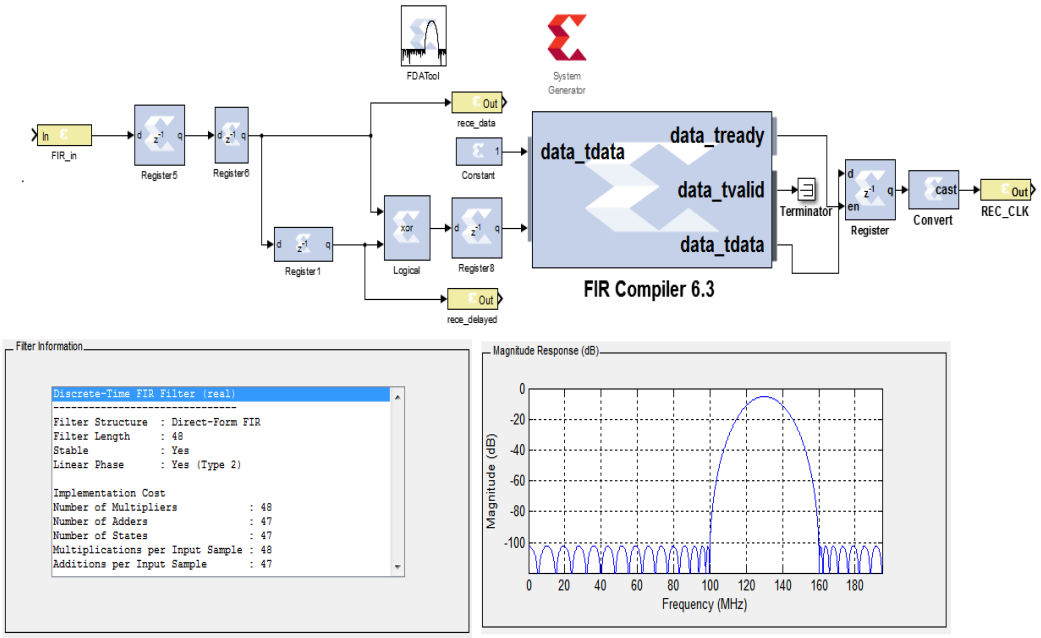

Fig. 4.20. Clock recovery model as viewed in the Xilinx System Generator®.

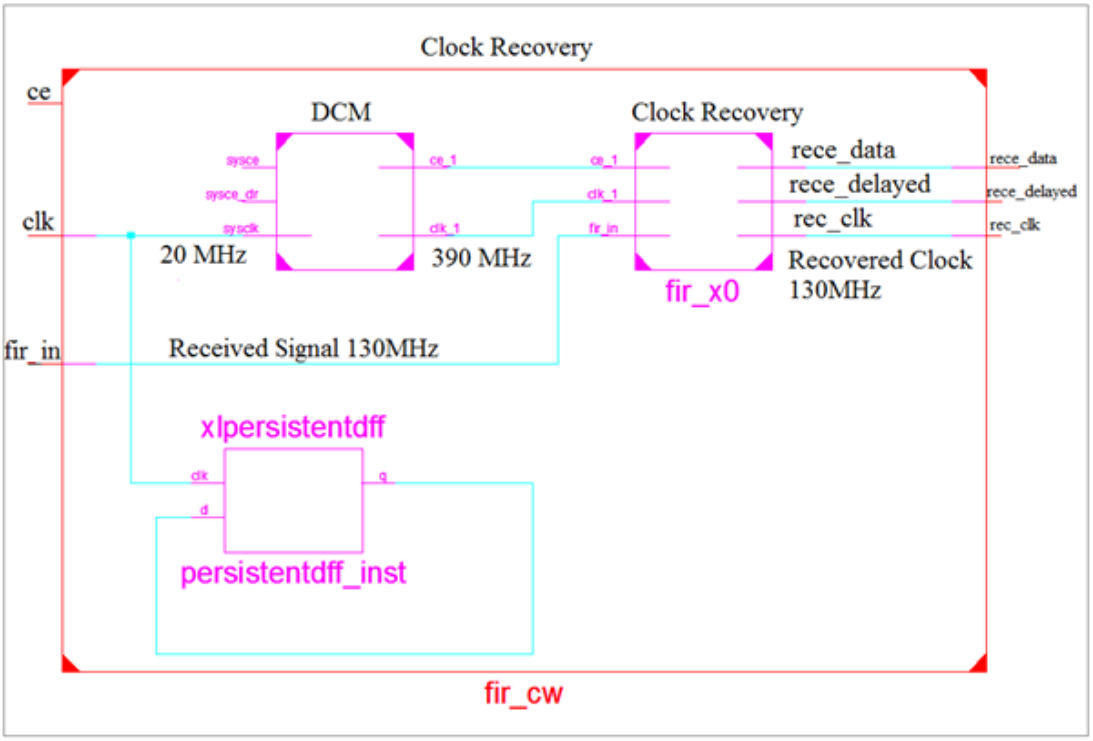

Fig. 4.21. RTL of the clock recovery.

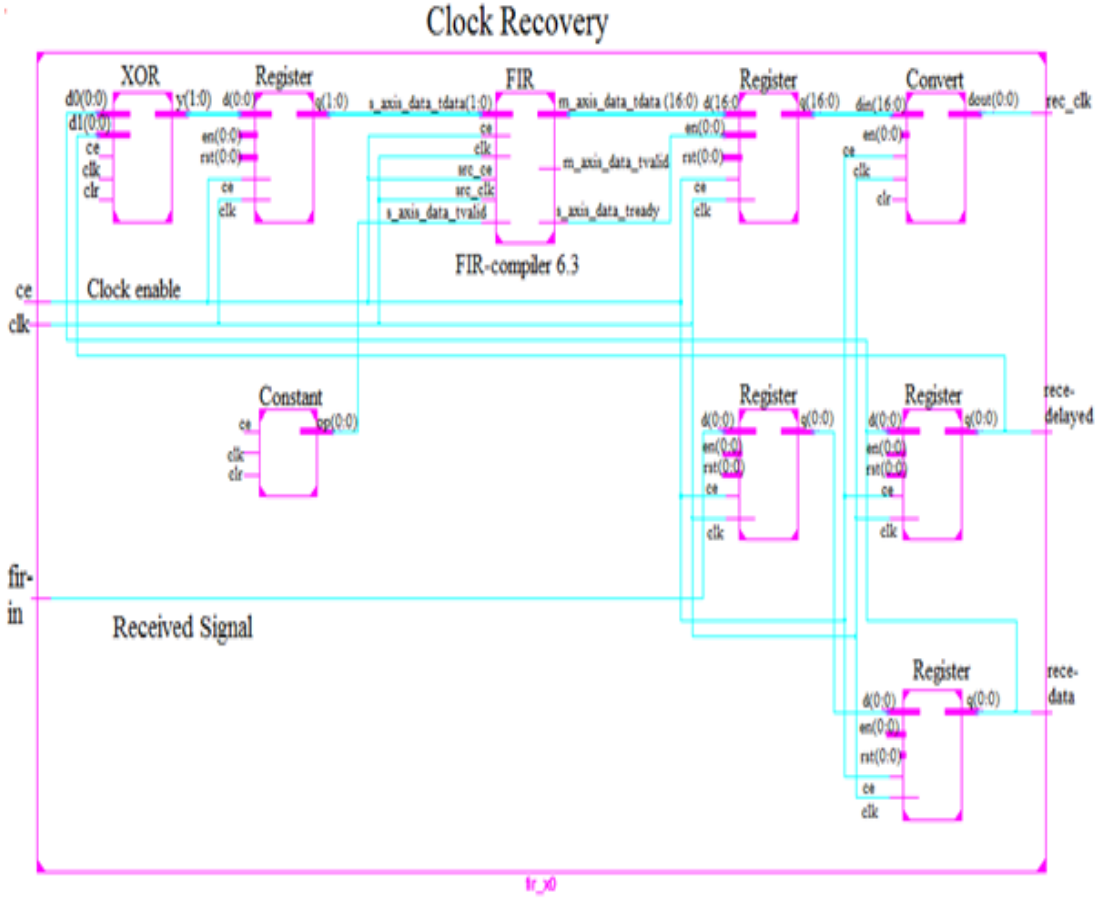

Fig. 4.22. RTL of the clock recovery inside components.

A PLL is a frequency control system (closed-loop) that recognises the phase difference between the input clock signal and the feedback clock signal of a controlled oscillator. The main blocks of the PLL are as follows:

- Phase frequency detector (PFD).
- Charge pump.
- Loop Filter.
- Voltage-controlled oscillator (VCO) and counter.

In this implementation, we used instantiation statement to design PLL that is provided by ISE ®. A component instantiation statement describes a subcomponent of the design entity. Moreover, it associates signals with ports of the subcomponent. Component instantiation defines as plugging a hardware component in the board through a socket. In this work, the PLL is instantiated in the top file. The PLL is used in the system to lock the desired frequency. The instance declaration is placed in the body of the design code. All inputs and outputs are connected. Fig. 4.23 shows the transmitter's clock and the recovered clock at the receiver's FPGA board.

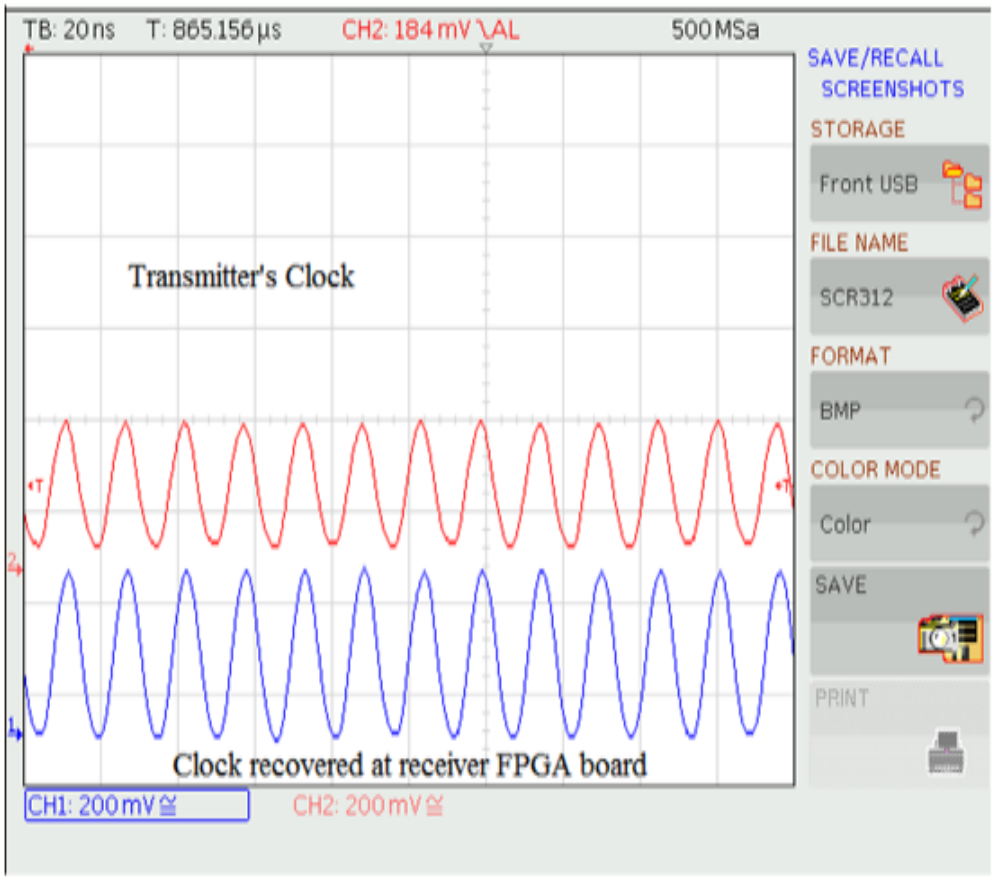

Fig. 4.23. The transmitter clock (red signal) and recovered clock at receiver side (blue signal) as visualized by the oscilloscope.

4.5 Data recovery

Fig. 4.24 shows the data recovery block diagram. The data recovery consists of stream sequence detector, user data tracking threshold, Bit basher, Lorenz generators, Manchester decoder and parallel to serial convertor. When the clock recovery has achieved properly at the receiver, the received signal is decoded by using Manchester decoder. After that, when the sync stream sequence is detected, the enable signal is generated to start the Lorenz generators and user data tracking threshold. The Lorenz binary stream has been scrambled by using Bit basher and then serialised by using parallel to serial convertor. The decoded signal and Lorenz generator must be synchronized to retrieve the transmitted data. The decoded signal is multiplied with the Lorenz binary stream. The output signal is further processed to retrieve the transmitted data using accumulator and comparison (≥) block.

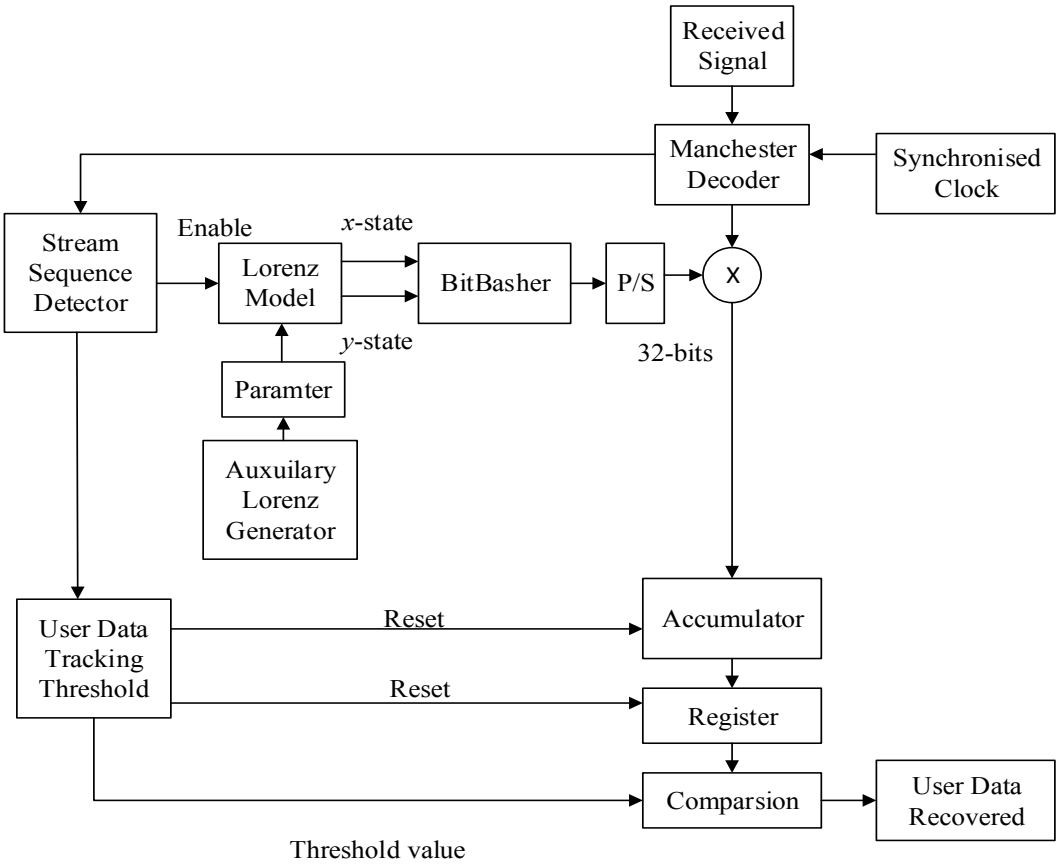

Fig. 4.24. The data recovery block diagram.

The user-data-threshold tracking model consists of a counter, a comparator, an accumulator, a register, and a subtractor. The process of the threshold-tracking subsystem begins when the counter receives an enable signal. The data-tracking subsystem performs like a buffer for the 32 bits. When the enable signal is received at the correct time, it begins to count 32 samples and compare them with a constant value of 32 by using a comparator. It then sends a reset signal to the accumulator once it reaches 32 samples and starts again. The accumulator is used to add up 32 samples before sending a threshold value to the relational block, see Fig. 4.25, Fig. 4.26 and Fig. 4.27 show the data recovery models, user data tracking model and simulated test signals of the data recovery process. The Xilinx © system generator design of the data recovery has started by multiplied the received signal with the Lorenz chaotic signal where the both signals are synchronised. The output of the multiplication have been accumulated every 32 samples. Then, the value of the accumulation is compared with the threshold value using the relational block. The Xilinx© data tracking sub-system has been designed to produce a proper threshold value. When the enable signal has

received, the counter is started to count from 0 until 31 based on constant and relational block. Then, every 32 samples are accumulated and produces a constant value that it is used as threshold value. The simulation results show the received signal that has included the preamble signal, sync signal and encrypted user information. The second signal shows the Lorenz chaotic signal which is synchronised with the received signal. Every 32 samples are accumulated .The reset signal is appeared every 32 samples. The accumulated signals have been converted into values using latched signals. The threshold values are shown in the simulation results which are compared with latched values. The transmitted information and recovered information are shown where there is a delay due to the process of the extraction process and there is some Xilinx© blocks causes delay. Fig. 4.28 shows the user data that is recovered at the receiver.

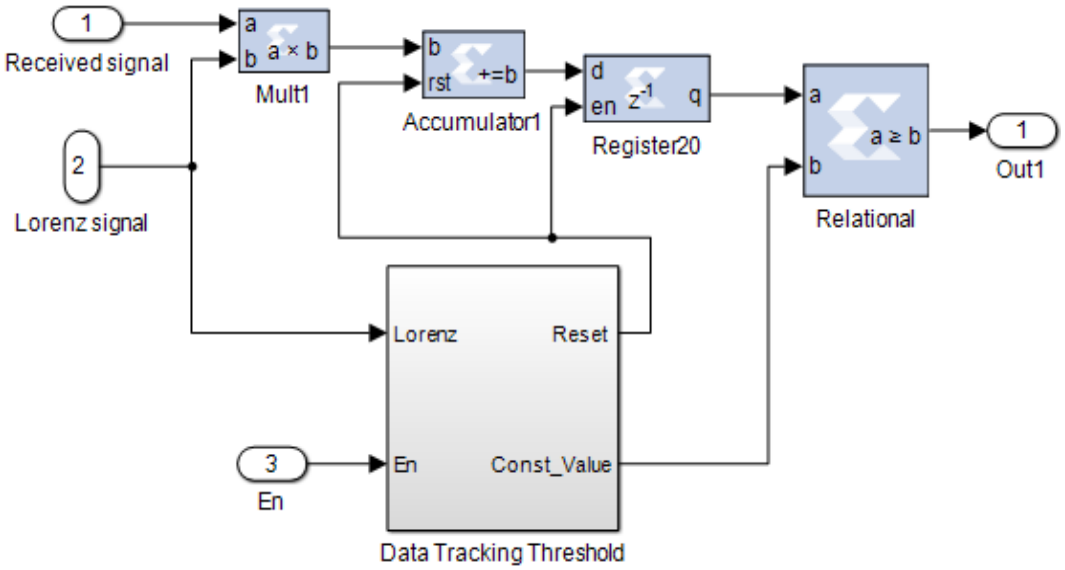

Fig. 4.25. Data recovery as viewed in the Xilinx System Generator®.

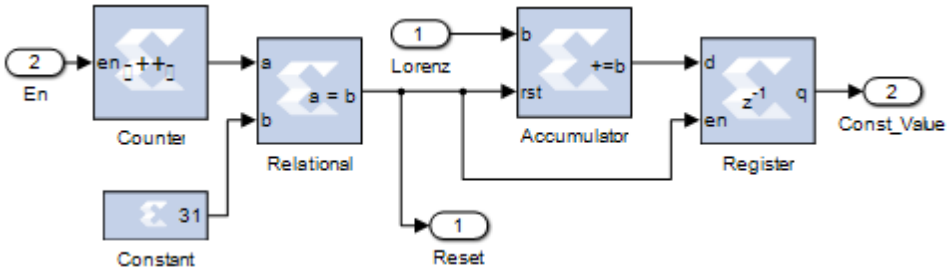

Fig. 4.26. Data tracking threshold subsystem as viewed in the Xilinx System Generator®.

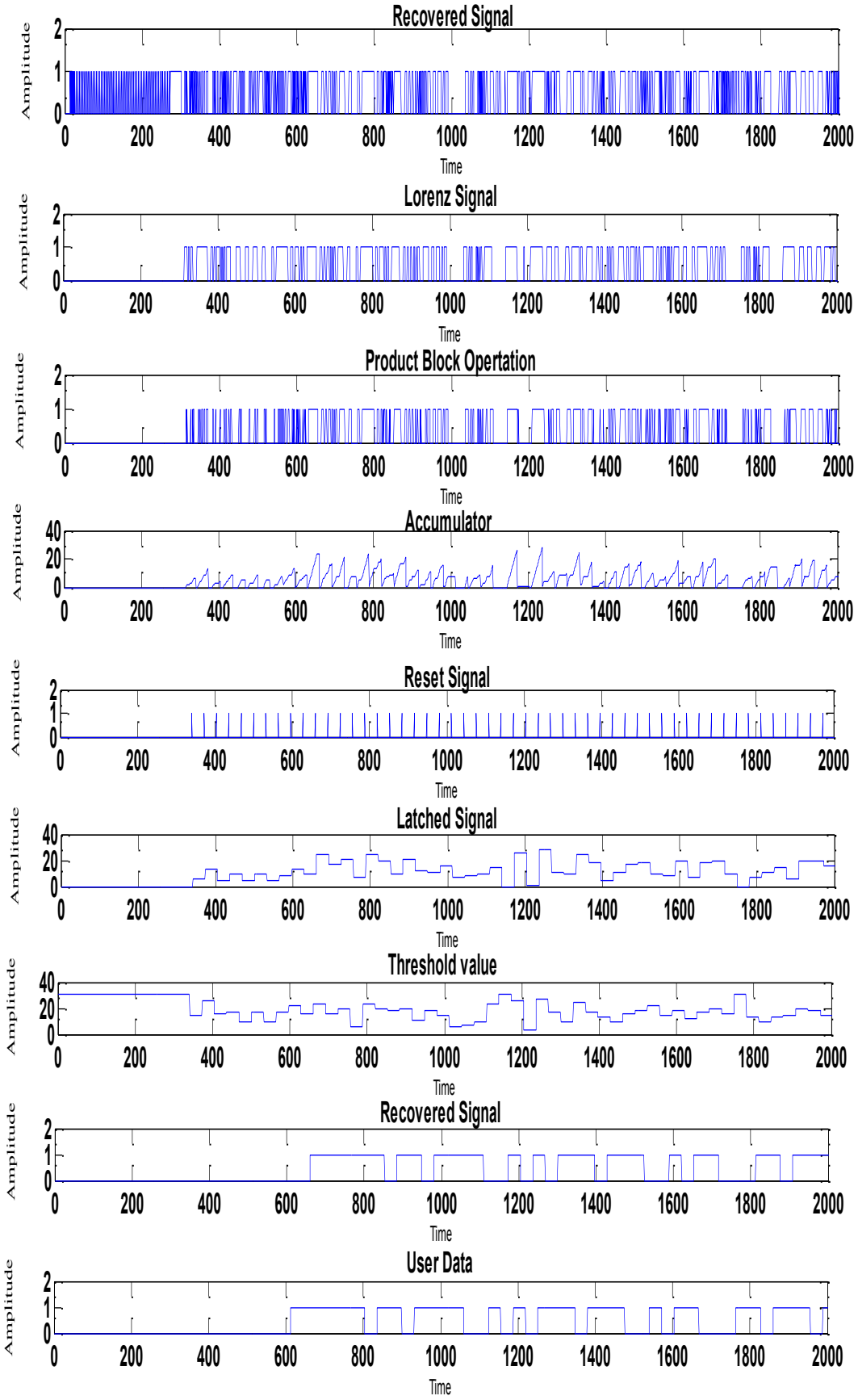

Fig. 4.27. Simulated result of data recovery signal processing.

Fig. 4.28. User data is recovered perfectly.

4.6 Synchronisation of Chaotic Signals

Synchronisation is one of the most important techniques in coherent receiver systems. Synchronisation-based user data recovery, in which noise is present, has a significant advantage in terms of its noise performance and data rate in comparison to non-coherent detection methods; however, these advantages are only realised if the synchronisation is maintained well. In contrast, when the synchronisation is not maintained well due to poor conditions, the non-coherent detection method is more robust and less complex than the coherent detection method [30].

Chaotic synchronisation between two different systems is one of the challenges of a coherent chaos-based communication system [130]. A coherent-based communication system design consists of two categories:

1. Chaos synchronisation technique [27, 130-136].
2. Chaos synchronisation based on conventional synchronisation techniques [46, 48, 137, 138].

The chaos synchronisation technique consists of master and slave systems. The master system (transmitter) is used to drive the slave system (receiver) by using a synchronisation signal. When the synchronisation is achieved properly, the slave system generates the same chaotic response signal as the master system [130].

In a conventional synchronisation system, the synchronisation depends on the similarities of the initial conditions of both the transmitter and the receiver.

Four methods for synchronising analogue chaotic generators have been reported. The first three methods, known as continuous chaotic synchronisation, include additive chaos masking, chaotic modulation and a chaotic cryptosystem [28, 93]. The disadvantage of using these techniques is that they require a large bandwidth due to the need to continually inject the synchronisation framework into the slave system [49]. The fourth method, impulsive synchronisation, was developed to overcome this issue; it consumes much less bandwidth than continuous synchronisation [49]. In [53], it is stated that neither impulsive synchronisation nor continuous synchronisation are reliable techniques for chaotic communication systems because both inject the drive signal into the slave system, which does not provide robust channel noise sensitivity. The technique presented in [53] is based on an asynchronous serial communication protocol that does not need to inject a signal into the dynamics of the slave system.

This chapter describes a practical system for synchronising two chaotic generators used in the digital Code Division Multiple Access (CDMA) method. Synchronisation is achieved and maintained through a sync sequence with known data that is transmitted to the receiver. The sync sequence triggers the chaotic generator at the receiver to start synchronously. This study presents a finite state machine (FSM) based on a Black Box Xilinx System Generator® for sync sequence detection. The aim of using this approach is to overcome the sensitivity of chaotic signal mismatch between two separate chaotic generators due to the presence of noise affecting the drive signal that is injected continuously or impulsively and transmitted through the channel, which causes poor synchronisation and affects data recovery. Chaotic generators are used to spread the data and provide security against attacks. Both the receiver and transmitter are implemented using two separate Spartan 6 Field Programmable Gate Array (FPGA) boards. Practical results demonstrating the robustness of the system are provided.

Digital CDMA systems using chaotic signals for data spreading offer a high degree of security against unauthorized reception. However, this technique adds an extra level of difficulty regarding synchronisation between the transmitter and the receiver. Not only must the clock signal be recovered and the two clocks be synchronised, but the

receiver on the digital chaotic generator must also be synchronised with the transmitter. If a passband system is designed, then a procedure for carrier recovery is also required. A block diagram of a baseband system is shown in Fig. 4.29.

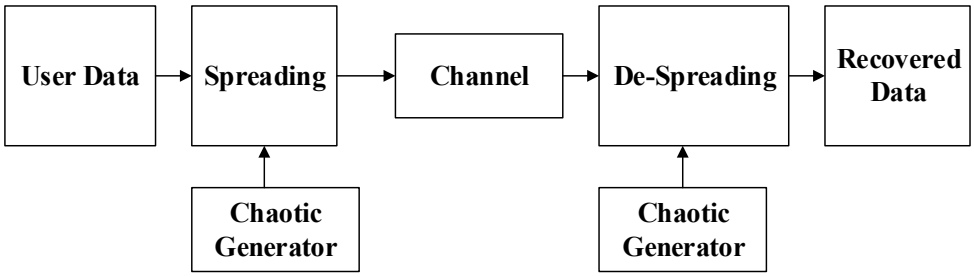

Fig. 4.29. A block diagram of the baseband system.

This chapter describes the technique that we used for synchronising the chaotic generators. Both the transmitter and receiver chaotic generators use the same parameters, but they also need to start at the same time step in relation to the transmitted data in order to ensure that the data is recovered correctly. In this work, the chaotic generators are implemented using the Lorenz model.

In this work, synchronisation is based on sending a 'sync stream' block, which is used to trigger the receiver chaotic generator. When the synch stream is detected, the transmitter chaotic generator output signal is identical to that of the receiver generator signal, since both chaotic generators are also clock synchronised. The synchronisation technique was tested and validated using two FPGA boards, as described later in this chapter. The system is implemented in hardware, and the design procedure is based on different design tools, such as: MATLAB, the Xilinx System Generator®, ISE and the Hardware Description Language VHDL.

4.6.1 Synchronisation principle

First the master clock of the digital system is recovered at the receiver. The second process is to synchronize the chaotic generators of the transmitter and receiver. The principle scheme of the synchronisation is presented in Fig. 4.30. When the sync stream block transmits the 32-bit stream, a known constant delay is added before the Lorenz generator starts generating the signal at the transmitter. At the receiver, the synchronisation unit uses the received signal $r(t)$ to generate the de-spreading signal,

which is identical to the Lorenz chaotic generator at the transmitter *c*(t), and both are clocked with the same clock rate. For data recovery, the received signal is multiplied with the chaotic signal that is synchronised with the received signal and then accumulated. The cross-product and summation process consists of the product block and accumulator, the accumulator and the rational block, which compares the threshold value with the accumulated value.

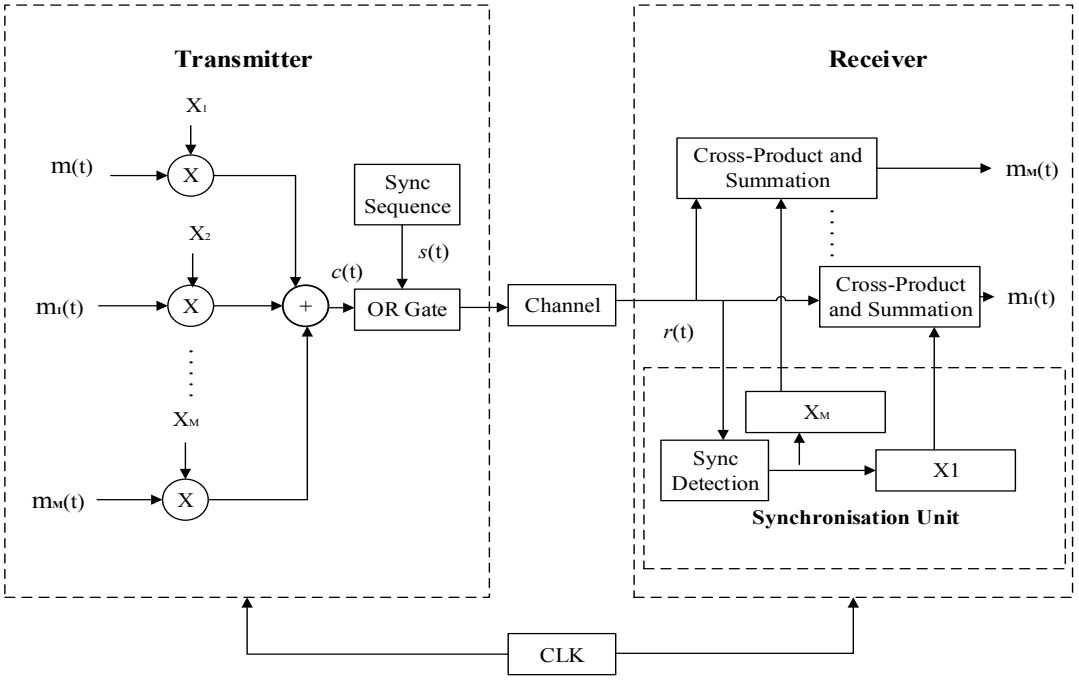

Fig. 4.30. The DS-CDMA digital communication system-based chaotic signal with the synchronisation unit.

4.6.2 Transmitter system

The transmitter system consists of the Lorenz generator, a spreading block and a sync stream. The Lorenz generators are represented by a 32-bit fixed point with 20 fractional bits. The chaotic signals are then serialised by using a parallel to serial convertor.

The synch stream generates the 32-bit length known data [1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1], which is used for synchronisation. The sync stream block contains a counter, Read Only Memory (ROM), comparison block, accumulator block, register and convert block. Fig. 4.31 shows the sync sequence block. When the enabled signal is received, the ROM of the known data is enabled. The counter begins to count to 32 when the ROM is enabled. Once this is completed, the enabled signal is set for the next subsystem. Fig. 4.32

shows the simulated sync stream. When the counter has received the enable signal, it started to count form 0 until 31. The rom block is used to produce the sync bit stream which is 32 bits. The rest of the sub-system is used to rest the counter block and rom block once the sync bits stream has been sent.

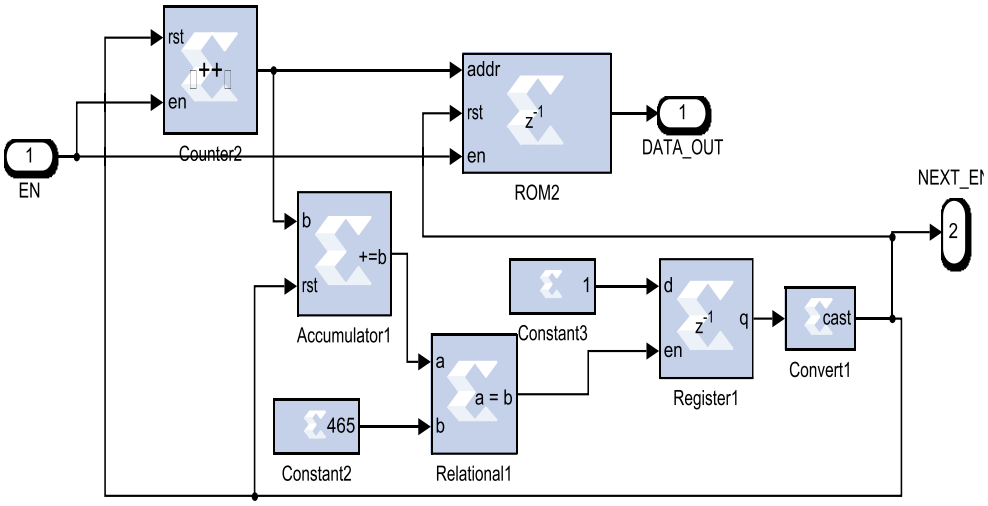

Fig. 4.31. Sync sequence block, as viewed using the Xilinx System Generator®.

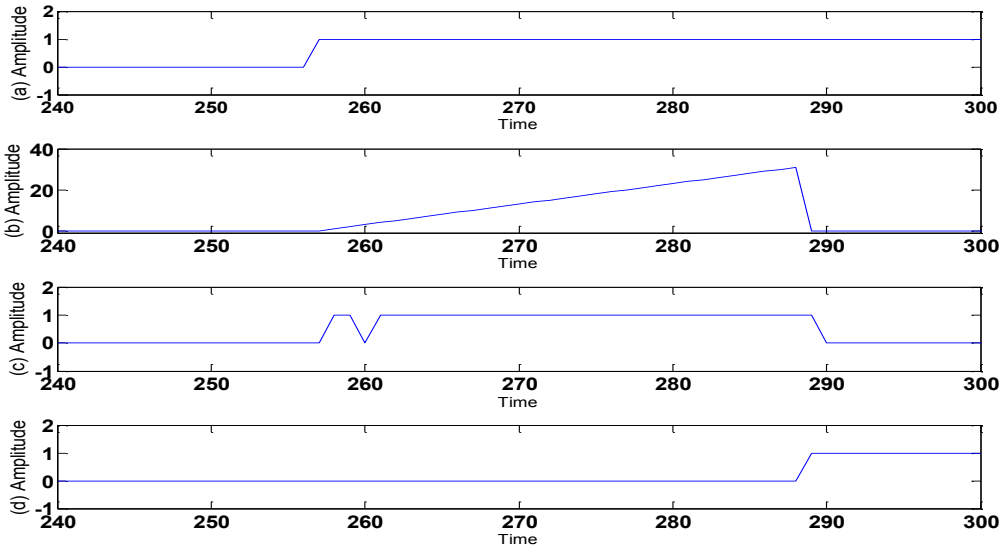

Fig. 4.32. Simulated test of the sync stream block.

4.6.3 Receiver system

The receiver system consists of Lorenz generators that are identical to those in the transmitter system, and a synch detector block. Both systems are clocked with the same clock rate. The proposed scheme is tested and validated experimentally using two Xilinx® Spartan 6 (SP605) FPGA (IC) boards (Model Number xc6slx45t-

3fgg484). To achieve this task, we developed a structural hardware architecture based on a VHDL description for the sync sequence detection using Black Box, where more details about the FPGA implementation of the system are also given.

The sync detector is performed using an FSM based on VHDL code. When the sequence with a 32-bit length is received, the FSM is able to detect it and enables a signal for the chaotic generators to start. The machine will keep checking for the proper sequence until it recognises it. The output becomes '1' when the sequence is detected in state S32; otherwise, it remains '0' for other states. The sync detector system model is shown in Fig. 4.33. Fig. 4.34 shows the simulated signal of the received signal and the sync sequence signal detection at the receiver system. It is clear that the desired sync signal is detected perfectly, and constantly remains at 1.

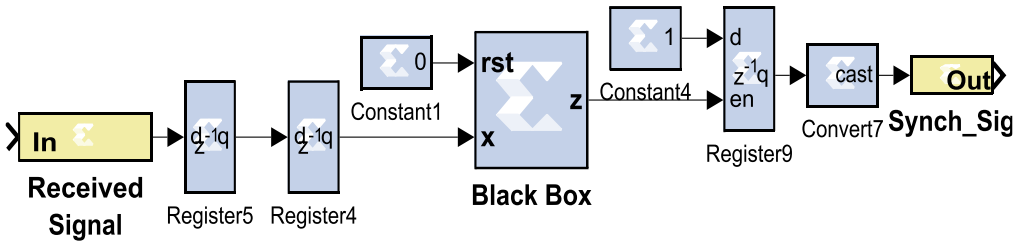

Fig. 4.33. The sync detector, as viewed using the Xilinx® System Generator®.

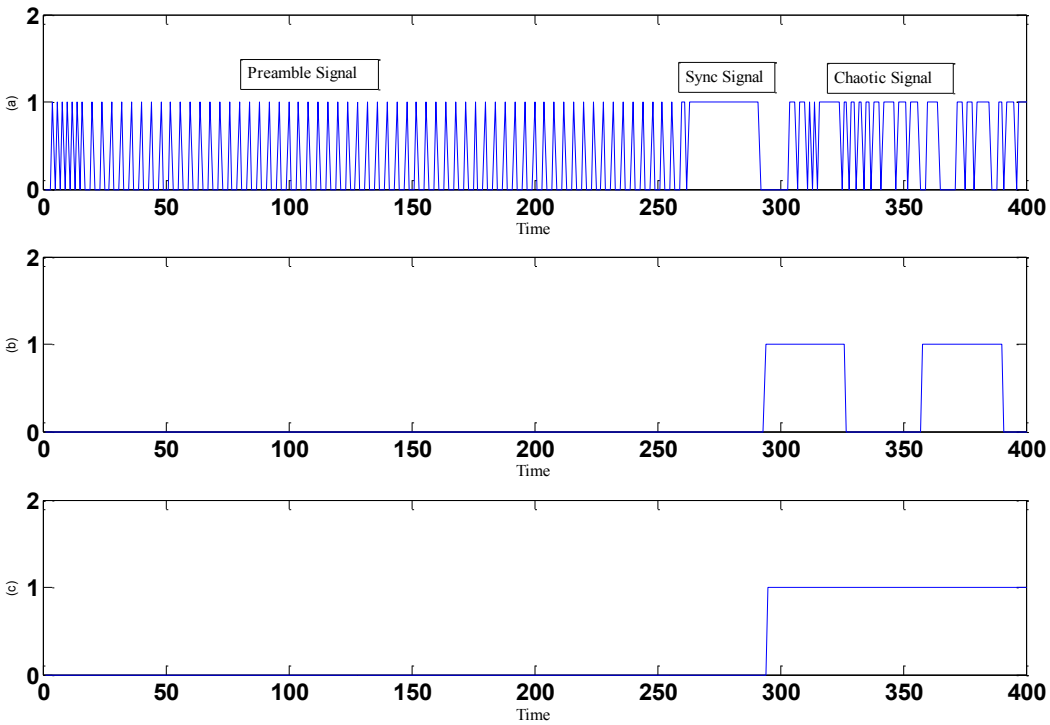

Fig. 4.34. Simulated test of the sync stream block.

4.7 FPGA Implementation Results

Fig. 4.35 shows the Register Transfer Level (RTL) schematic of the sync sequence subsystem where the enabled signal of the sync sequence subsystem is received from the preamble subsystem. Fig. 4.36 shows the RTL of the synch sequence subsystem.

The real-time results show that the two chaotic signals are synchronised perfectly at the receiver, as shown in Fig. 4.37 and Fig. 4.38. The red signal is the Lorenz Chaotic signal of the transmitter and the blue signal is the Lorenz chaotic generator at the receiver side. The two Lorenz generators are synchronised well. Many experimental tests are performed to evaluate the performance of the synchronisation scheme in terms of user data recovery. The synchronisation is well maintained and the transmitted user data is recovered with no error at the receiver with a transmission data rate of 2 Mbps.

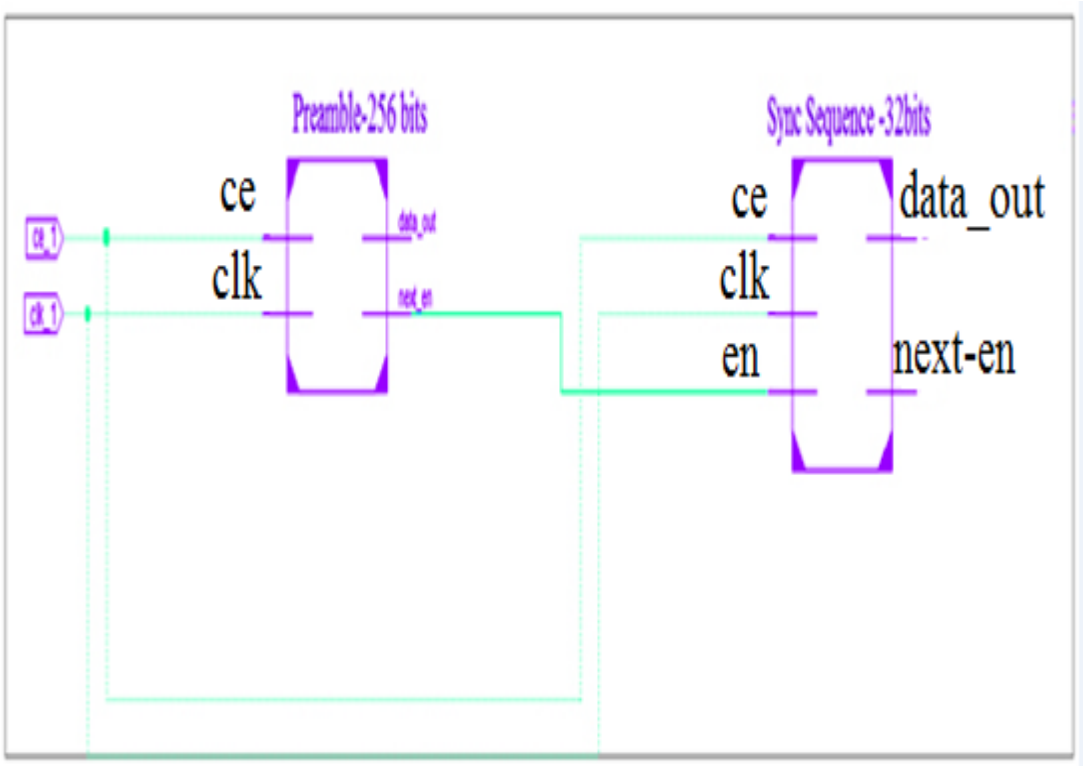

Fig. 4.35. RTL schematic of the sync sequence subsystem.

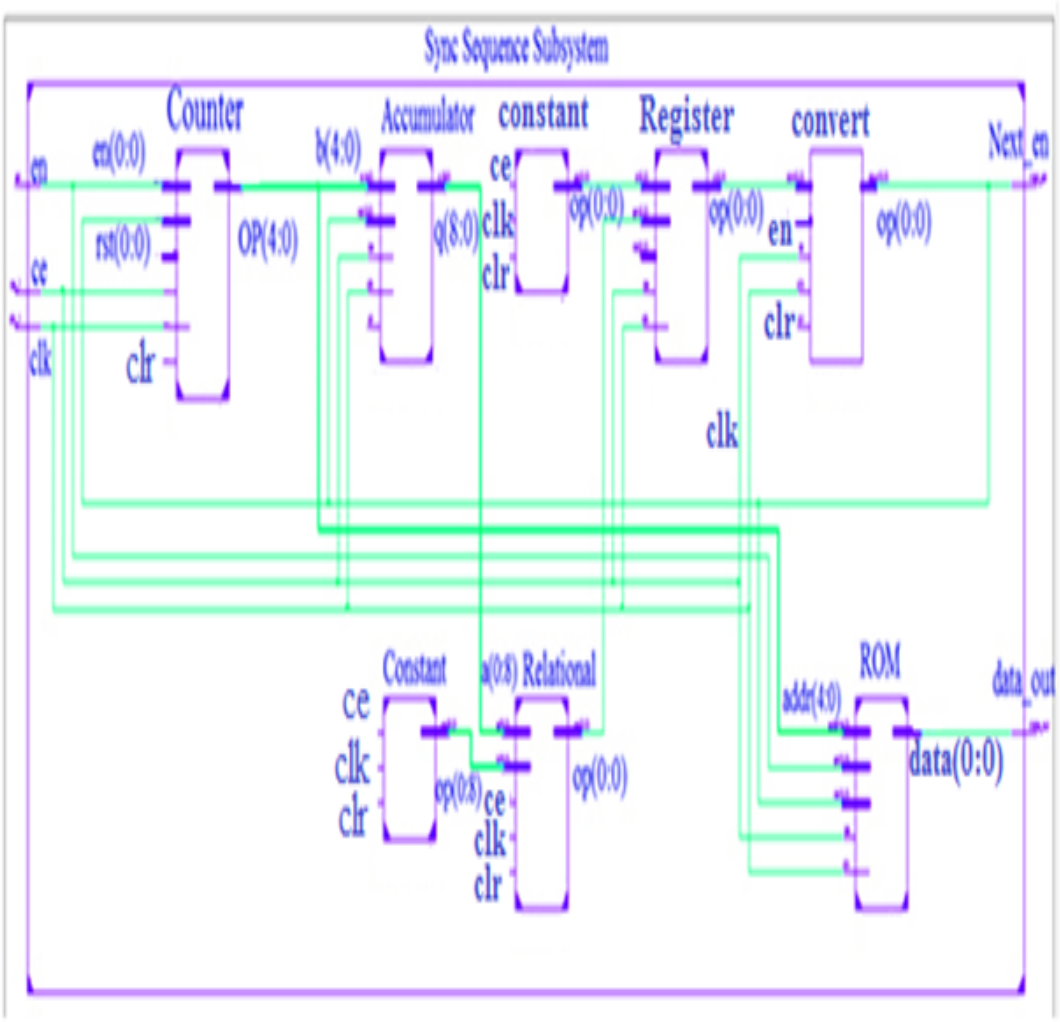

Fig. 4.36. RTL schematic of the sync sequence subsystem.

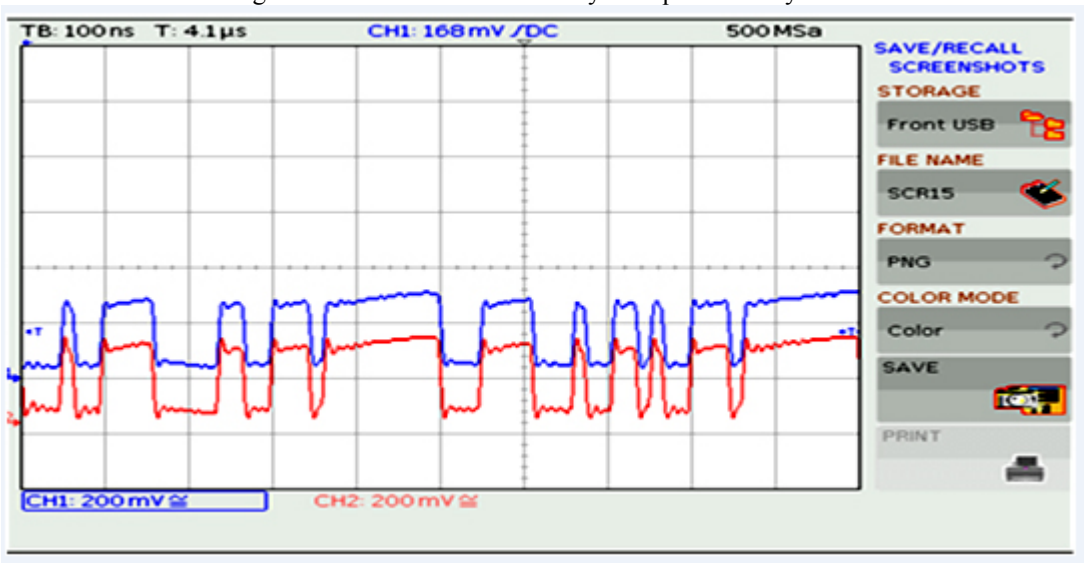

Fig. 4.37. Chaotic generators are synchronised perfectly at the receiver in real-time, as visualised by the oscilloscope.

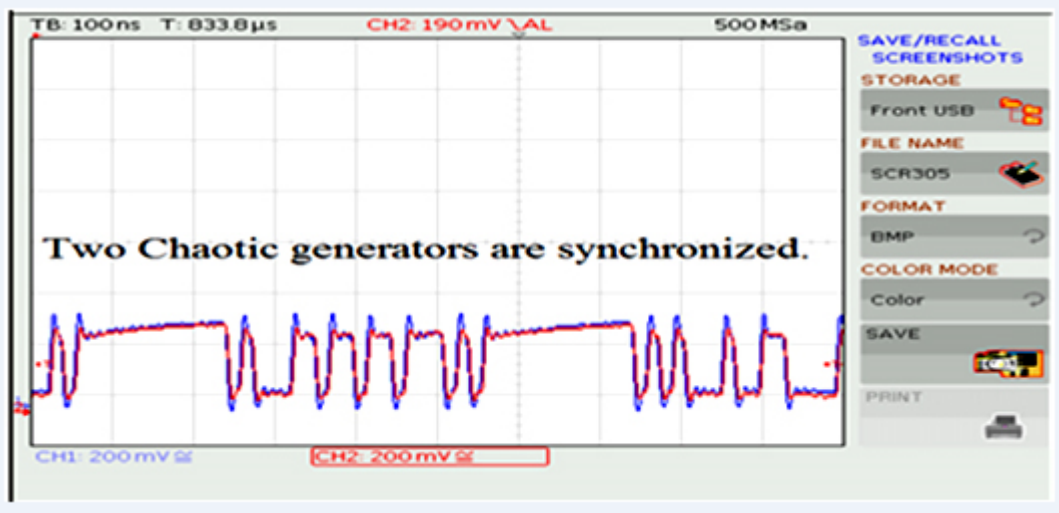

Fig. 4.38. Chaotic generators are synchronised perfectly at the receiver in real-time, as visualised by the oscilloscope.

4.8 Conclusion

In this chapter, clock recovery is detailed. The simulated results of the clock recovery technique are obtained. The clock recovery SIMULINK blocks are then converted to System Generator® blocks. The simulated test shows that the clock is recovered well from the random data generator (Bernoulli generator). The clock recovery is then implemented in real time by using two separate Spartan 6 FPGA boards. The clock rate of 65 MHz is recovered well at the receiver. The PLL is locked at the desired frequency at clock rate of 65 MHz. The data recovery based on a System Generator® is presented. The simulated results are obtained.

The disadvantage of the clock recovery technique emerges when the received signal frequency is doubled because of the rising and falling edge detector used to detect the phase. This means that in real time implementation, we require a high speed clock source for the bandpass filter sampling frequency to recover the clock.

In this chapter, also we have described a practical system for synchronising two chaotic generators used in a digital CDMA. The techniques are based on a sync stream that is transmitted to the receiver in order to trigger the Lorenz chaotic generator in the same time step. The chaotic synchronisation is well maintained at the receiver. Both the receiver and transmitter are implemented using two separate Spartan 6 FPGA boards. The proposed technique is validated experimentally, and the results are obtained.

The advantage of a synchronisation method is that there is no need to inject a signal into the dynamics of the slave system, which affects the channel efficiency. In addition, this method is not affected by a high noise environment.

Chapter 5

FPGA IMPLEMENTATION OF COMMUNICATION SYSTEMS USING CHAOTIC BLOCK CIPHER

5.1 Introduction

The ultimate aim of this research is to implement a system with high security and noise immunity using a chaotic stream cipher. However, it was decided to start with the implementation of a block cipher system in order to develop reliable method of synchronisation and data recovery. The CDMA spreading blocks were generated first outside the digital implementation and thus avoiding the need to implement digital chaotic generators and to avoid developing methods of synchronising the transmitter and receiver chaotic generators.

The block cipher system has the same noise immunity characteristics as the stream cipher but does not have the same security. In chapter 6 we will extend the block cipher systems developed in this chapter to full stream cipher systems.

This chapter describes the practical implementation of a digital communication system based on block spreading for four users. Codes that have been used in this work are extracted from the Lorenz generators. Each user code length is 32-bits. The receiver system is able to discriminate all transmitted user data by applying cross-product and summation. This digital communication system has been implemented successfully by using two Xilinx® Spartan 6 FPGA boards, which will be explained in detail throughout this chapter.

5.2 Block Spreading Communication System

Fig. 5.1 shows a block diagram of the transmitter and receiver systems. The transmitter design is constituted by the following subsystems, Preamble, sync-sequence known-data (see chapter 4, section 4.6.2), user data generator, user data spreading, adder and Manchester-encoder.

The receiver design is as follows, sync-detection (see chapter 4, section 5.6.3), clock-recovery (see chapter 4, section 4.2.2), Manchester-decoder, data-recovery and system clock.

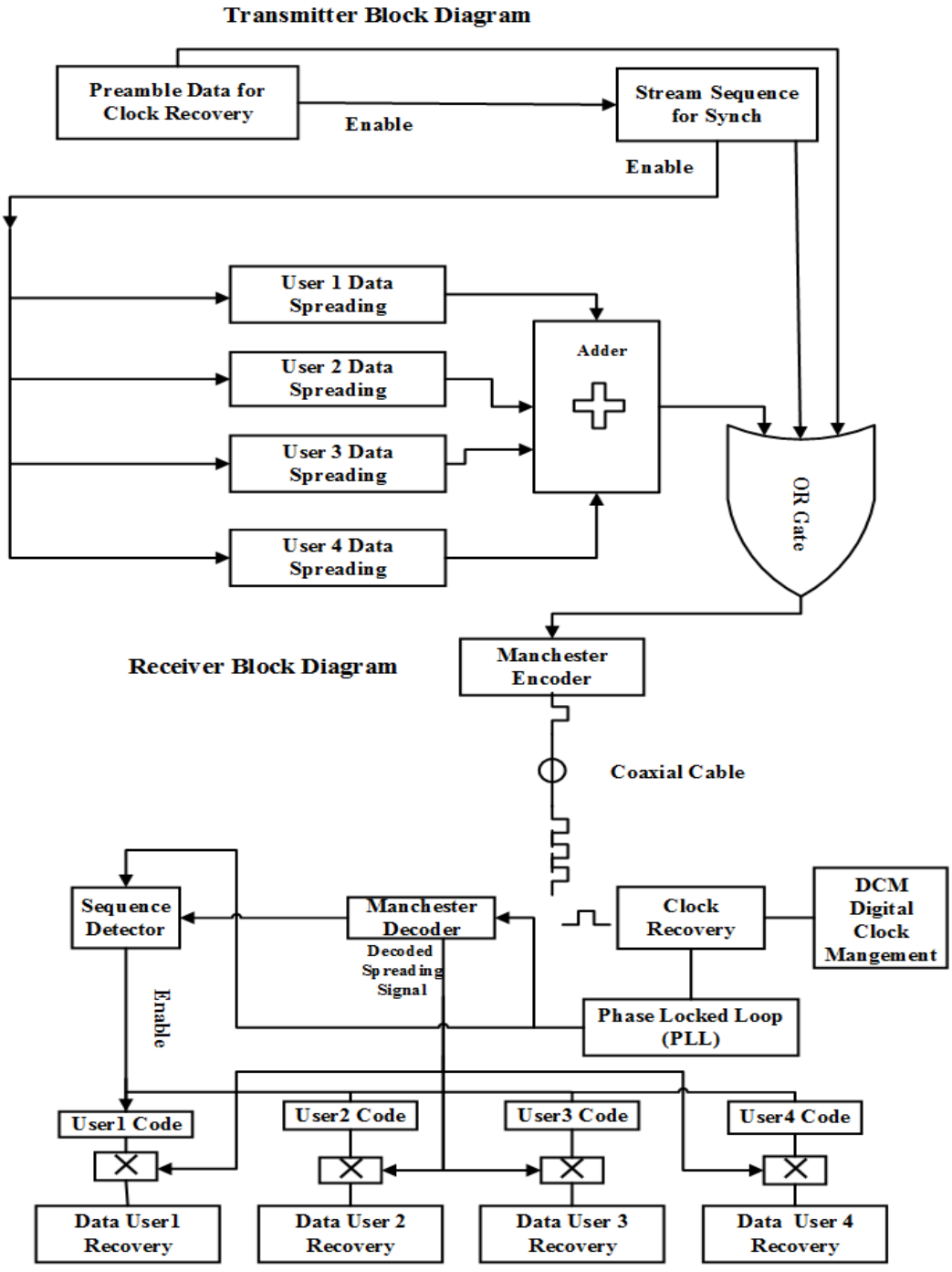

Fig. 5.1 .Four-user Block – spreading communication system.

5.3 Transmitter System

Fig. 5.2 shows the block diagram of the transmitter design, along with the process of the transmitter block-spreading communication system.

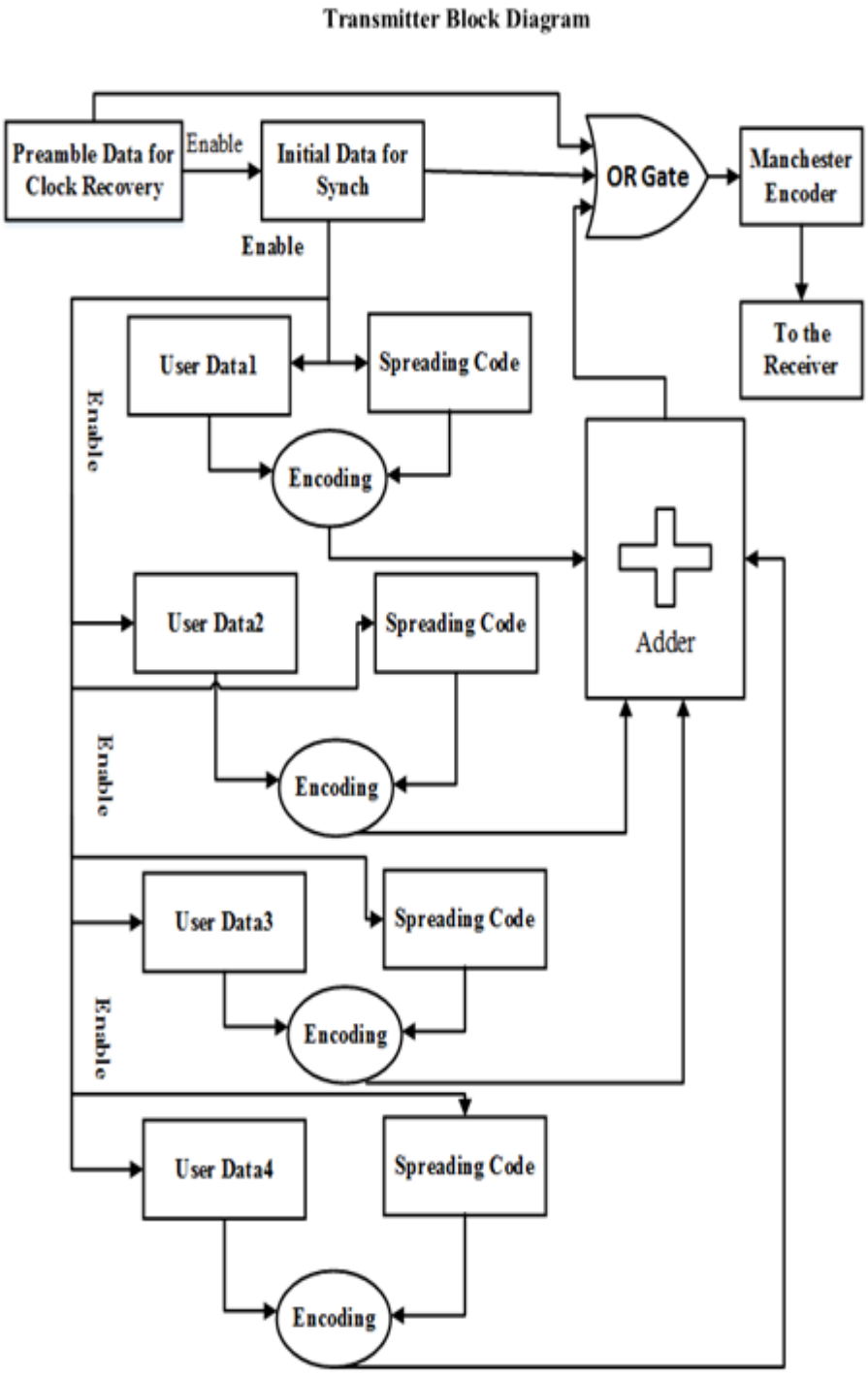

Fig. 5.2. A block diagram of the four-user spreading communication system.

Hardware properties of the digital communication system based on chaotic block spreading are presented in Table 5.1.

Main Clock Frequency	20 MHz oscillator
Modulation	Spread Spectrum (SS)
Spreading Code	32-bit
Spreading User Data Frequency	65 MHz
User Data Frequency	2 MHz
Data Rate	2 Mbps
FPGA Development Board	SP605
FPGA Family	Xilinx® Spartan 6
FPGA (IC) Model Number	xc6slx45t-3fgg484
Number of User Data Streams Generated and Encrypted	Four
Method of User Data Stream Production	Linear Feedback Shift register (LFSR)

Table 5.1. Fixed Properties of the Digital Communication System Based on Chaotic Block Spreading.

The transmitter has been designed based on the Xilinx System Generator®. The design contains the subsystems of preamble, “sync”-sequence, user data generating and user data spreading are shown in Fig. 5.3. The design shows that the gateway ‘out’ (user data) was used to compare the transmitted user data with the retrieved user data for bit-error calculations purposes. Adder blocks are used to sum all four user data. The convert block is used to output only one bit stream by using fixed point precision with number of bit is 1 and binary point is 0. The OR gate is used to sequential the process of the system, where the first step is to transmit the preamble system and then the sync sequence is started. Whereas the last step is the user data transmission.

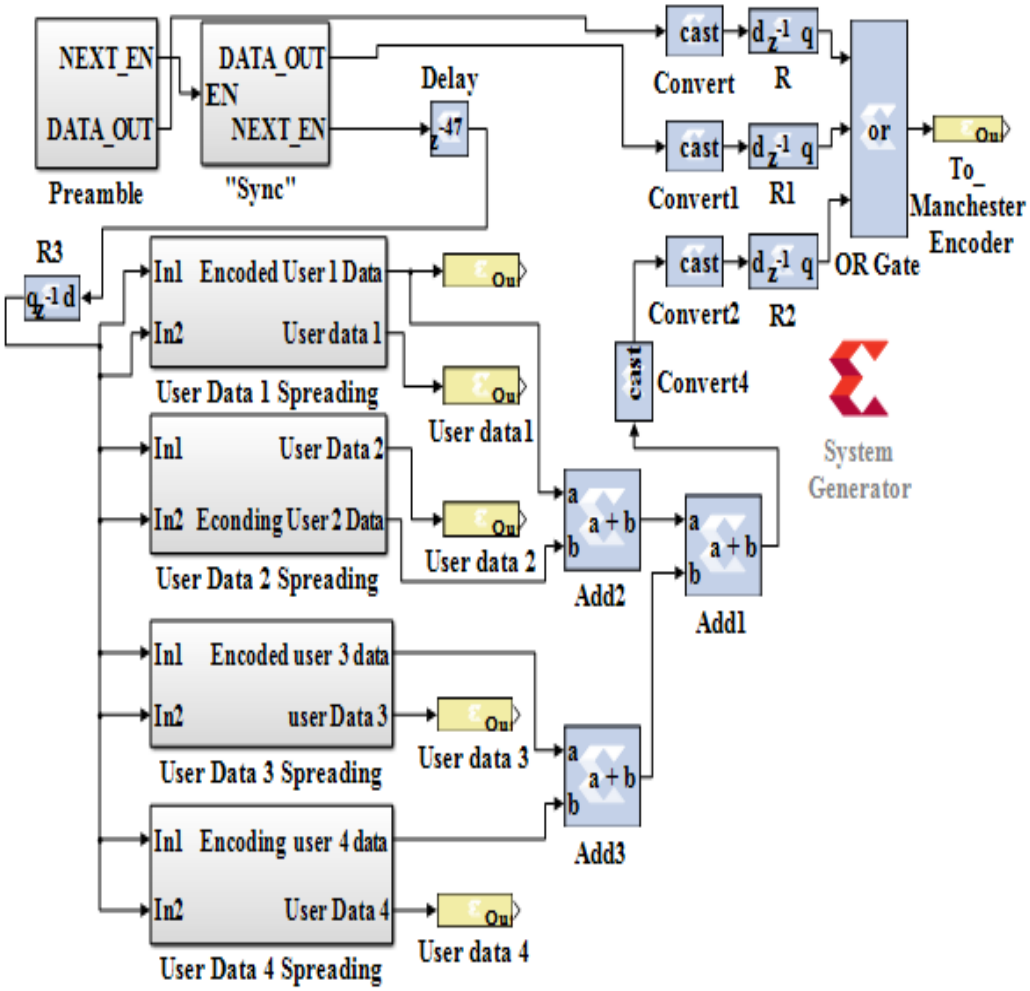

Fig. 5.3. Four-user spreading system, viewed in the Xilinx System Generator®.

5.3.1 Preamble subsystem

The preamble is used in the transmitter model for clock recovery purposes. The 256-bits of known data are transmitted to the receiver. The preamble subsystem is as follows:

- Counter block is used to count from 0 to 255.
- ROM block of depth of 256 with known data.
- Relational block with comparison with constant value of 255 is used to enable the next subsystem as well as to reset the ROM when it reaches the set value. When the ROM is reset, its output is zero.

- Convert block is used to convert unsigned data type to Boolean data type. Fig. 5.4 shows the preamble subsystem.

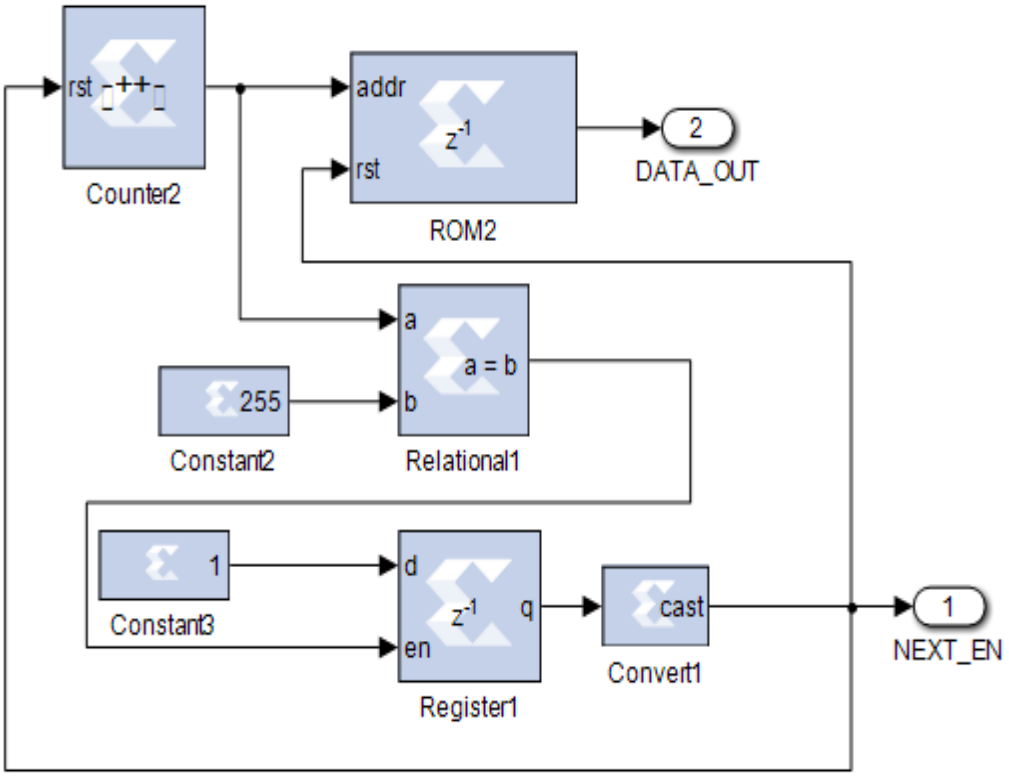

Fig. 5.4. The preamble subsystem, as viewed in the Xilinx System Generator®.

5.3.2 User data generator

In practice the user generates his/her own system but here we are generating a pseudo random data to represent the user data. User data are generated using a Xilinx Liner Feedback Shift Register (LFSR). The structure chosen of the LFSR is Fibonacci with an XOR gate. The Fibonacci type specifies the structure of the feedback. It has one XOR gate at the beginning of the register chain that XORs the taps together with the result going into the first register. The XOR is used to specify the feedback signal. The number of bits in LFSR specifies the number of registers in the LFSR chain. The initial value specifies the initial seed value where the LFSR begins its repeating sequence. A feedback polynomial specifies the tap points of the feedback chain, and the value must be entered in hex format. In order to generate different bit sequences for each user as data, a different polynomial chain is chosen for each user [139].

5.3.3 User data spreading

The user data is encoded using a multiplication code. Fig. 5.5 shows the user data spreading. Fig. 5.6 illustrates the design of user spreading code multiplied by user data. Each single bit of user data is spread by 32-bits from the chaotic signals. Fig. 5.6(a) shows the user spreading code repeating every 32 bits. Fig. 5.6(b) shows the user data generated using linear LFSR. Fig. 5.6(c) shows the user data has spread. Fig. 5.7 demonstrate the real-time results of the user data spreading.

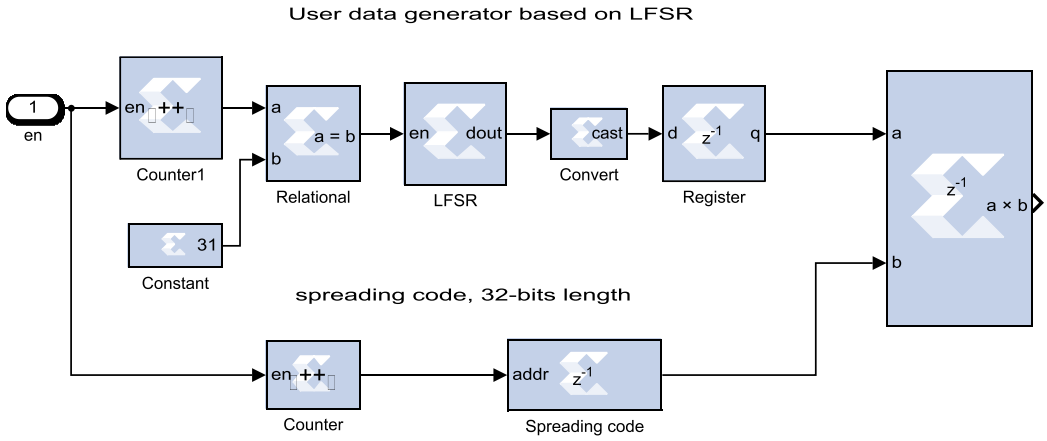

Fig. 5.5. User data spreading.

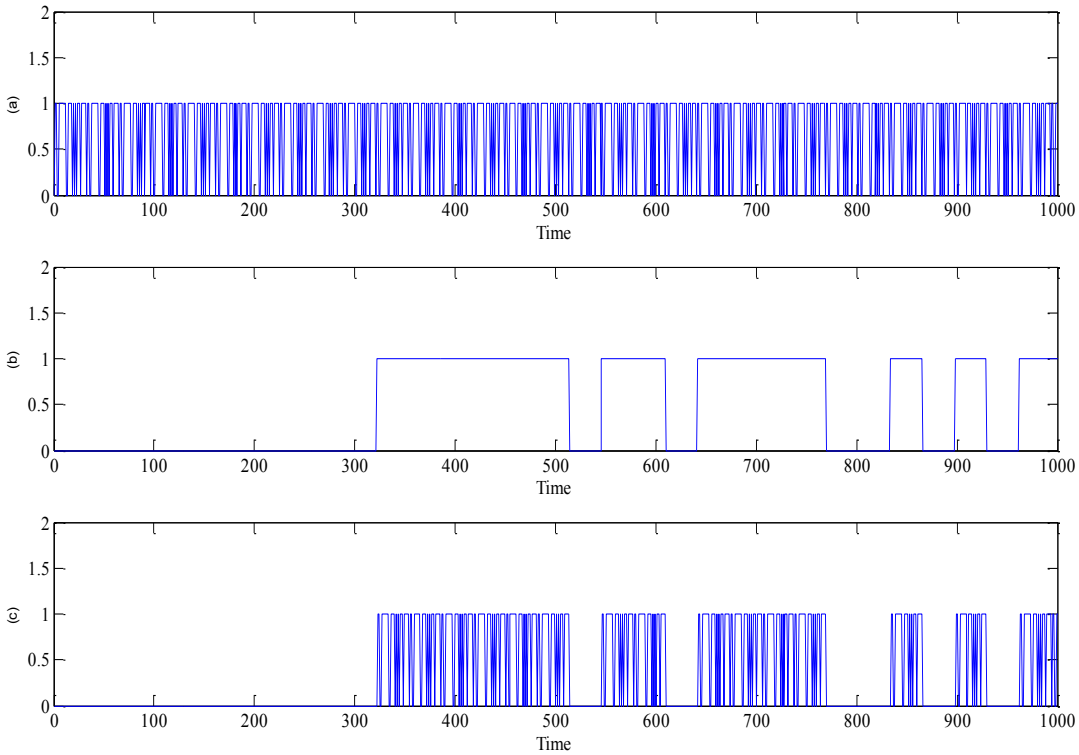

Fig. 5.6. Simulation test. (a) User spreading code (32-bits fixed), (b)User data and (c) Spreading user data.

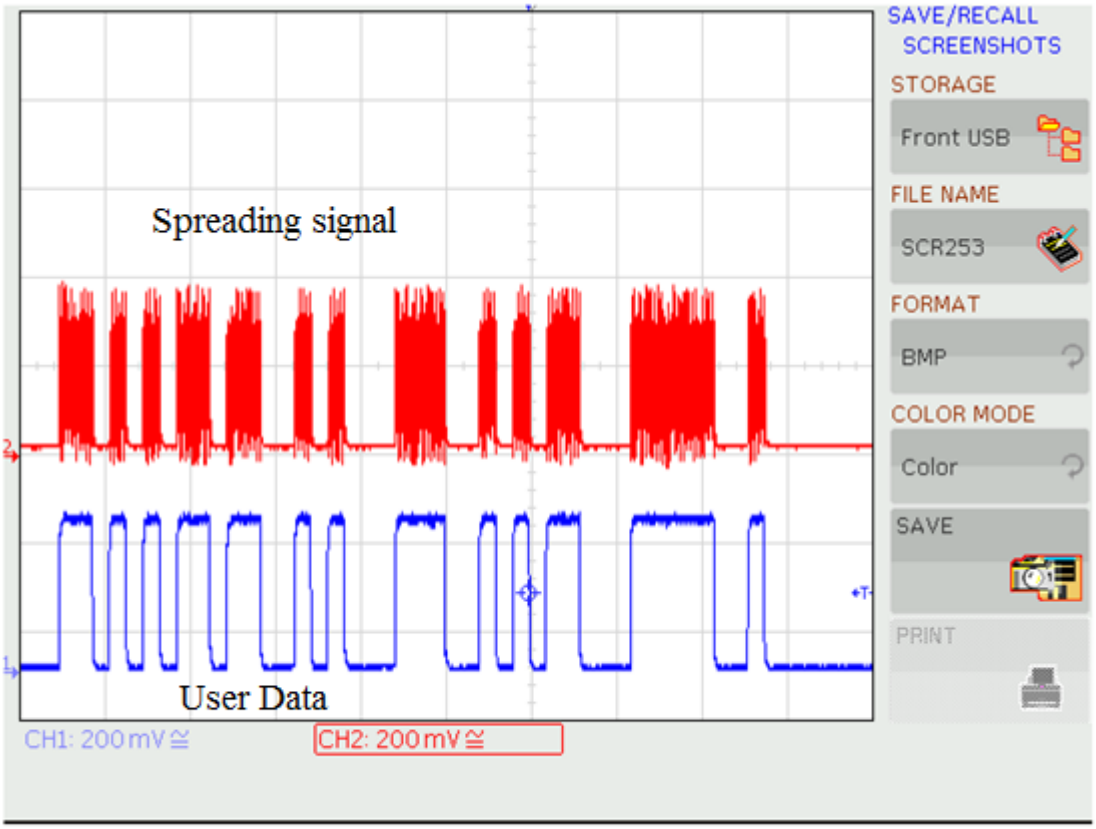

Fig. 5.7. Real-time result of the user data spreading, visualized by the oscilloscope.

5.3.4 Adder

In order to sum the contribution of the four users, it is necessary to use the Xilinx Add/Sub blocks as shown in Fig. 5.8 . Fig. 5.9 shows the simulated result of the four user spreading block are combined. Fig. 5.10 shows that the four user data are combined.

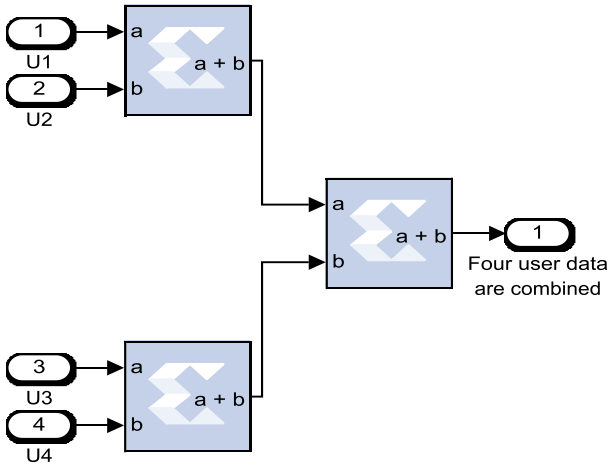

Fig. 5.8. Adder.

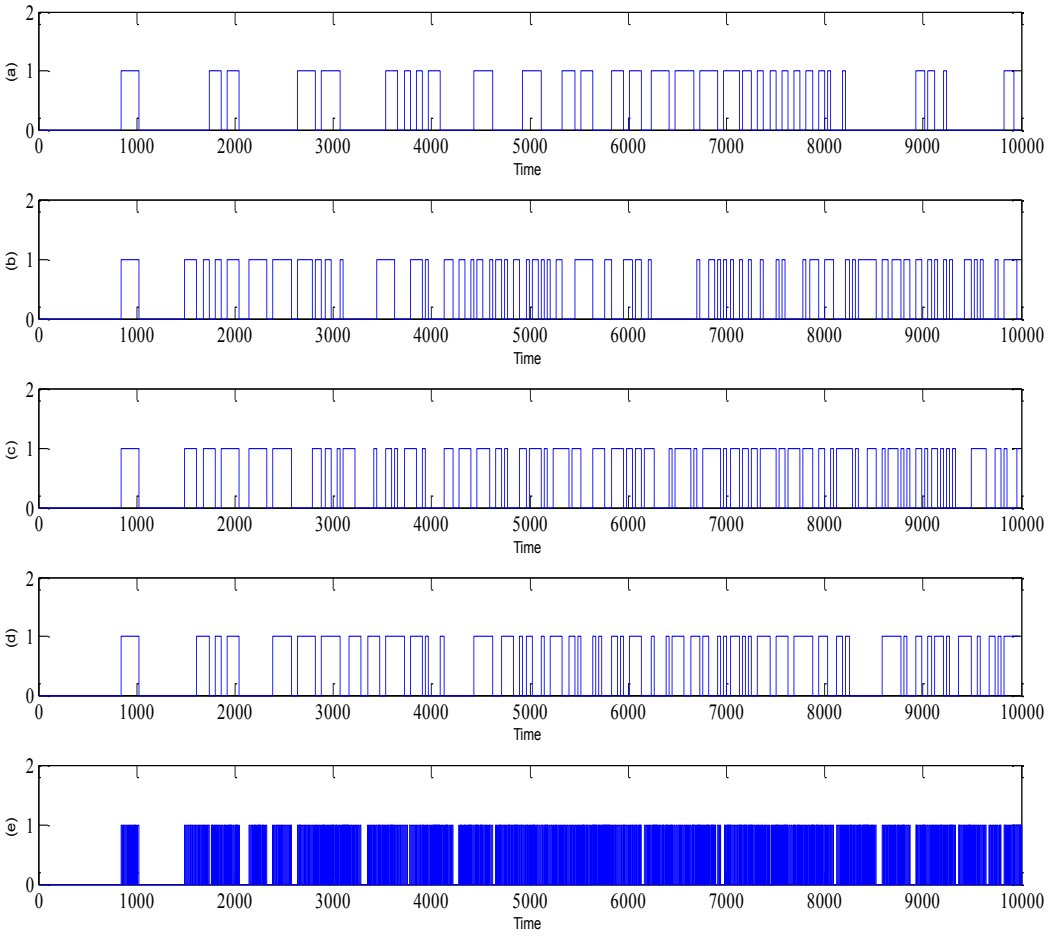

Fig. 5.9. Simulated results, (a) User 1, (b) User 2, (c) User 3 (d) User 4 and (e) four user data are combined.

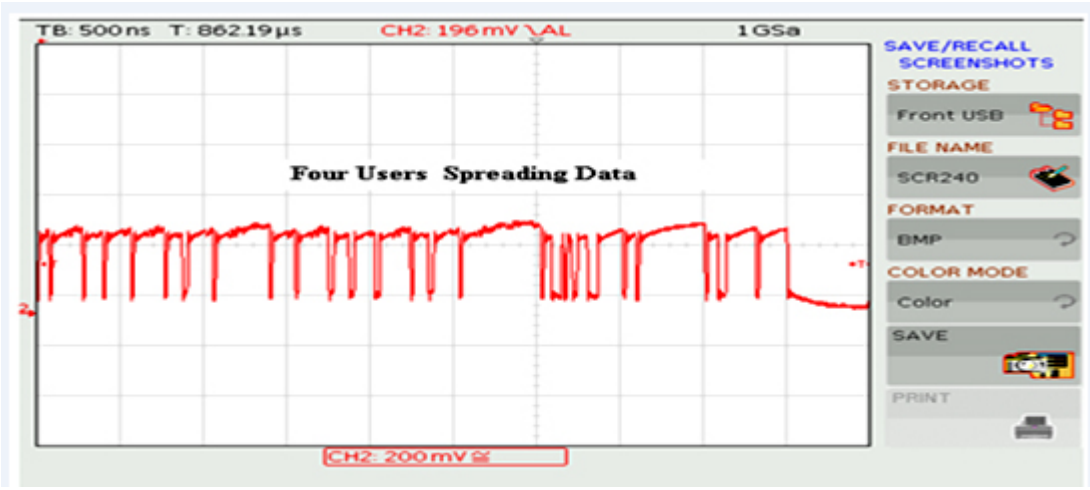

Fig. 5.10. Four users spreading data combined in real time, visualized by the oscilloscope.

5.3.5 Manchester encoder

The Manchester encoding is used in data transmission to allow the receiver to easily synchronize with the transmitter. It splits each bit period into two and ensures that there is always a transition between the signal levels in the middle of each bit. The Manchester encoder is established by using VHDL code, and the operation is performed by XOR of the data with the clock. Fig. 5.11 shows the data has been encoded.

Fig. 5.11. User data encoded with the Manchester encoder in real time, visualized by the oscilloscope.

5.3.6 System clock rates

A clock receives a frequency as an input from the source and can distributes that frequency or generates new frequencies to fulfil the application requirements within the system. This means the clock is able to convert the input source frequency into higher (multiplier) or lower frequency (divide). This process can be done using PLL.

The main clock used in the transmitter is a single-ended, oscillator socket Low Voltage Complementary Metal Oxide Semiconductor (LVCMOS) running at 20MHz. Two output clock rates have been generated to satisfy the transmitter design requirement using Digital Clock Management (DCM). A 65 MHz clock rate and the 130 MHz clock rate. The 65 MHz clock has been used for spreading the user data while 130 MHz has been used for the register.

5.3.7 Integrated Synthesis Environment (Xilinx® ISE)

The ISE Project Navigator is a high-level software manager for FPGA design. The transmitter design contains all files related to the project as follows.

- Digital clock management, provided by using the CORE Generator IP and Architecture Wizard IP.
- Manchester encoder based on Very High Speed Integrated Circuit Hardware Description Language (VHDL) code.
- Transmitter design based on the System Generator®.
- Register (to prevent overshoot) based on the System Generator®.
- User Constraints File (UCF).

Fig. 5.12 displays the top-level Register-Transfer Level (RTL) graphical representation of the transmitter design.

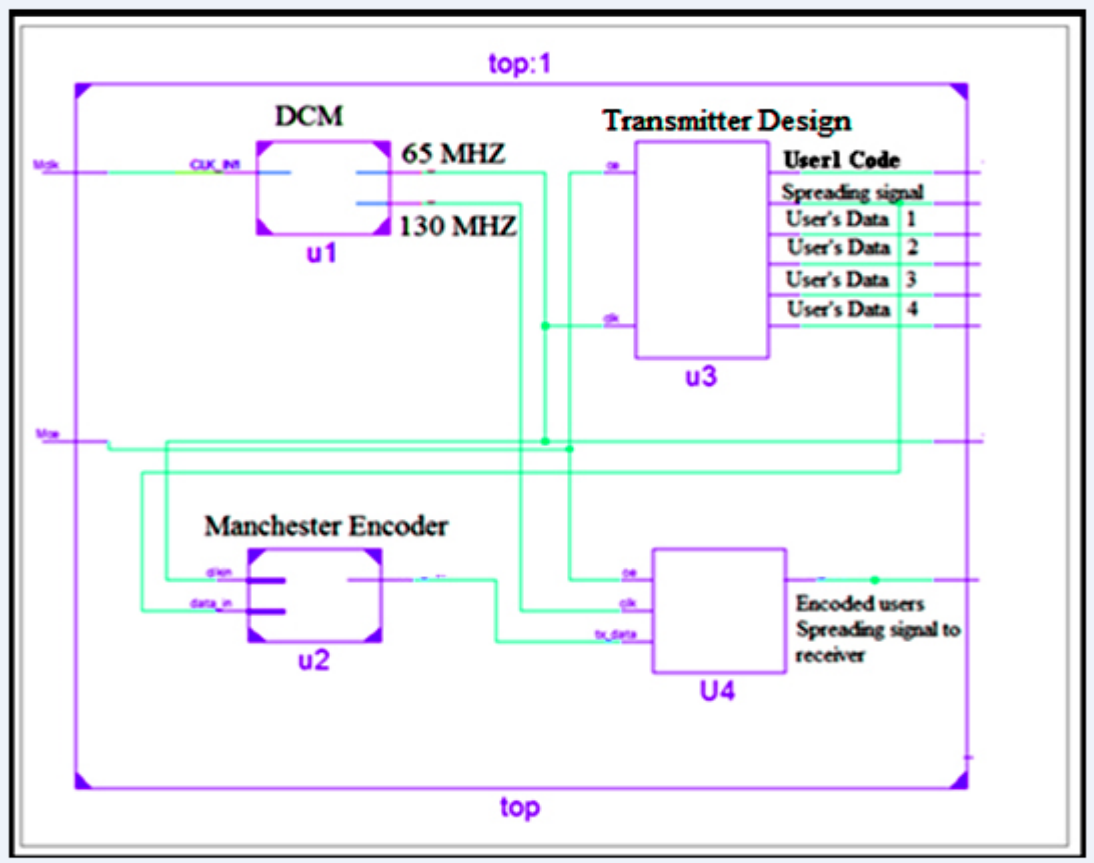

Fig. 5.12. RTL schematic of the transmitter design.

The signals are sent to the receiver through an SMA connector pin 'A3'. The SP605 board includes a 6-pin single-inline (SIP) male pin header (J55). We used these pins

to compare the data transmitted with the received data. All four user data are assigned to the SIP header connections pin (G7, H6, D1 and R7) receptively. The master clock is assigned to AB13. The Xilinx plan ahead provides pin assignments that makes easy for the user to fully automatic or semi-automated I/O ports to physical package pins. Thus, we used plan ahead to assign desired pins. Fig. 5.13 shows the plan ahead- pre-synthesis.

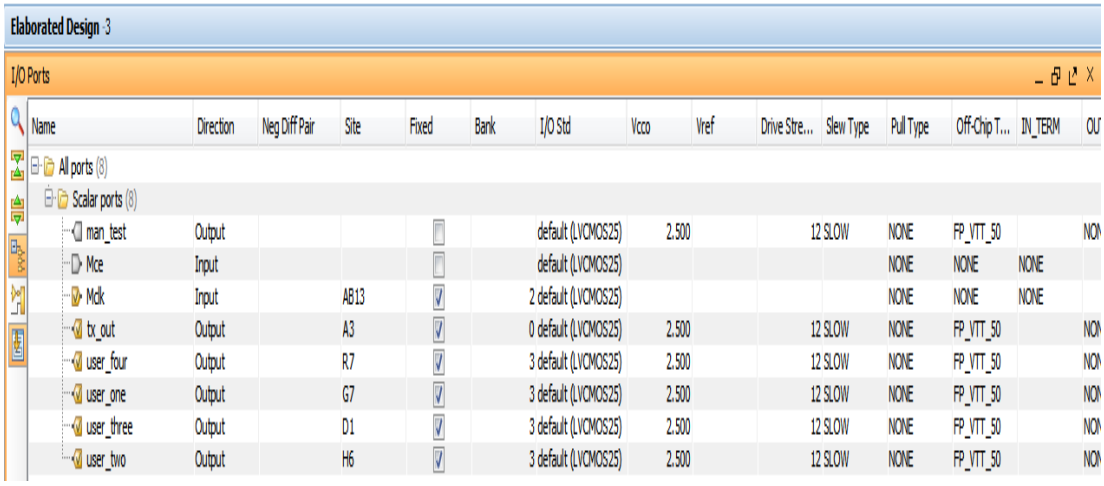

Fig. 5.13. I/O pin (plan ahead).

5.3.8 Device Utilisation Summary

Table 5. *2* presents the device utilisation summary of the target device 'xc6slx45t-3fgg484'. From this summary, we can determine that the transmitter design can be implemented in a small chip. The table shows that only 1% is consumed for slice registers and Look Up Table (LUT).

Slice Logic Utilisation	Used	Available	Utilisation
Number of Slice Registers	130	54,576	1%
Number Used as Flip Flops	130		
Number Used as AND/OR logics	0		
Number of Slice LUTs	111	27,288	1%
Number Used as Logic	76	27,288	1%
Number Used as Memory	13	6,408	1%
Number Used Exclusively as Route-Thrus	22		
Number of Occupied Slices	68	6,822	1%
Number of MUXCYs Used	100	13,644	1%
Number of Fully-Used LUT-FF Pairs	100	131	76%

Number of BUFG/BUFGMUXs	3	16	18%
Number of DSP48A1s	1	58	1%
Number of PLL_ADVs	1	4	25%
Number of BUFIO2FB/BUFIO2FB_2CLKs	1	32	3%
Number of LOCed IOBs	5	10	50%

Table 5. 2. Device Utilisation Summary of the Target Device.

5.4 Receiver System

Fig. 5.14 shows the block diagram of the receiver. The received signal is decoded by the Manchester decoder. The 'enable' signal is then received from the sequence detector block, instructing the user codes to begin. The cross product function is used, multiplying the replica of the user spreading code with the user code at the receiver. The 32-bit length buffer is used to accumulate the data. The fixed threshold for each user data is applied to discriminate the user data from others.

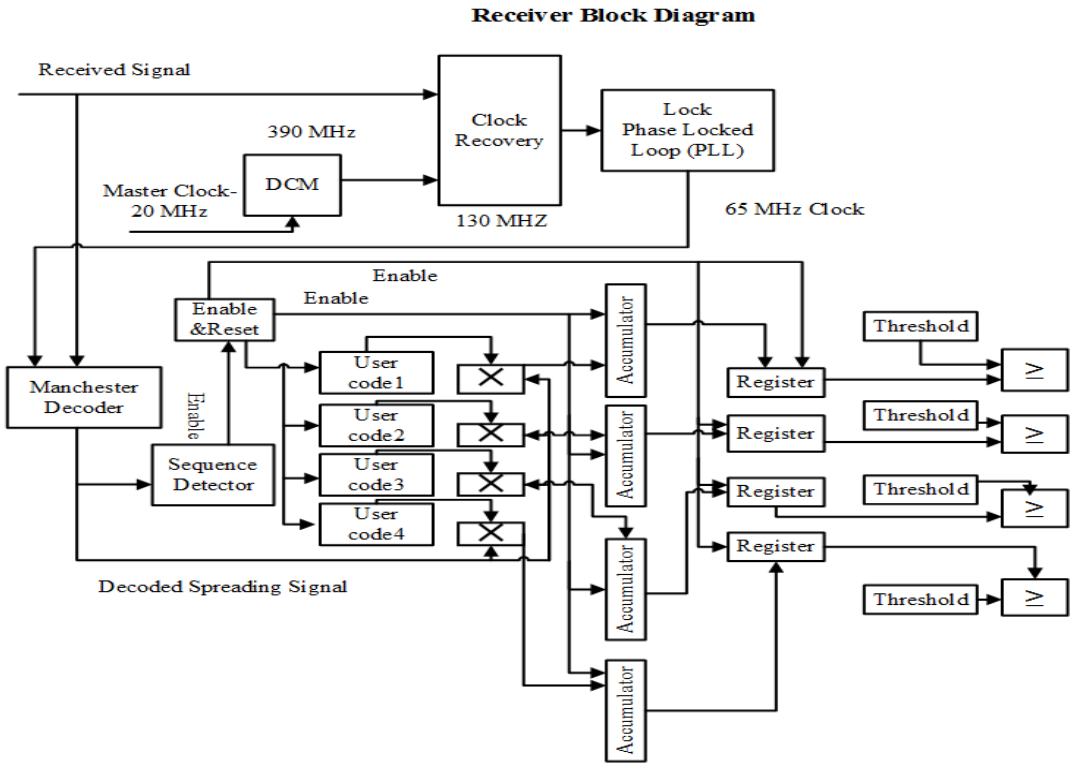

Fig. 5.14. A block diagram of the receiver design.

The system consists of two System Generator® designs, clock recovery and receiver system that are wrapped up later by using ISE tool. The System Generator® designs

are shown in Fig. 5.15. Gateway 'outs' are used to trace and visualize the signals via the oscilloscope. The Finite State Machine (FSM) "sync" detector (based on VHDL code) is performed by using a Black Box. Mux blocks are used in this design to start only when the "sync" signal is received. Otherwise, the user code is set to zero.

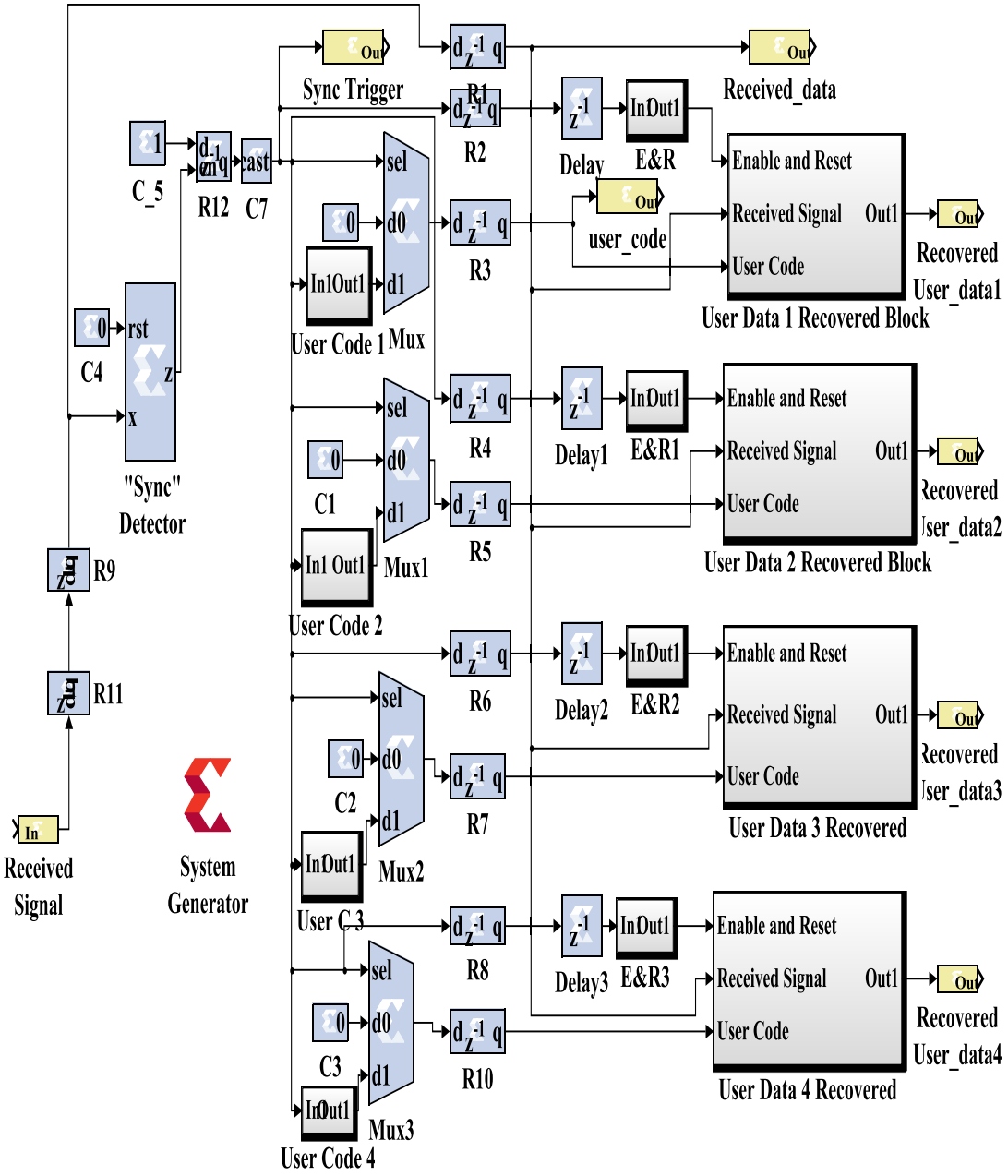

Fig. 5.15. The receiver system, viewed in the Xilinx System Generator®.

5.4.1 Manchester decoder

The RTL model of the Manchester decoder shows the Exclusive OR gate with two inputs. The first input is the recovered clock which is 65 MHz, and the second input

is the encoded spreading signal. The output signal of the encoder model is the decoded spreading signal. The RTL model of the Manchester decoder is displayed in Fig. 5.16. The decoded user spreading data can be seen in Fig. 5.17.

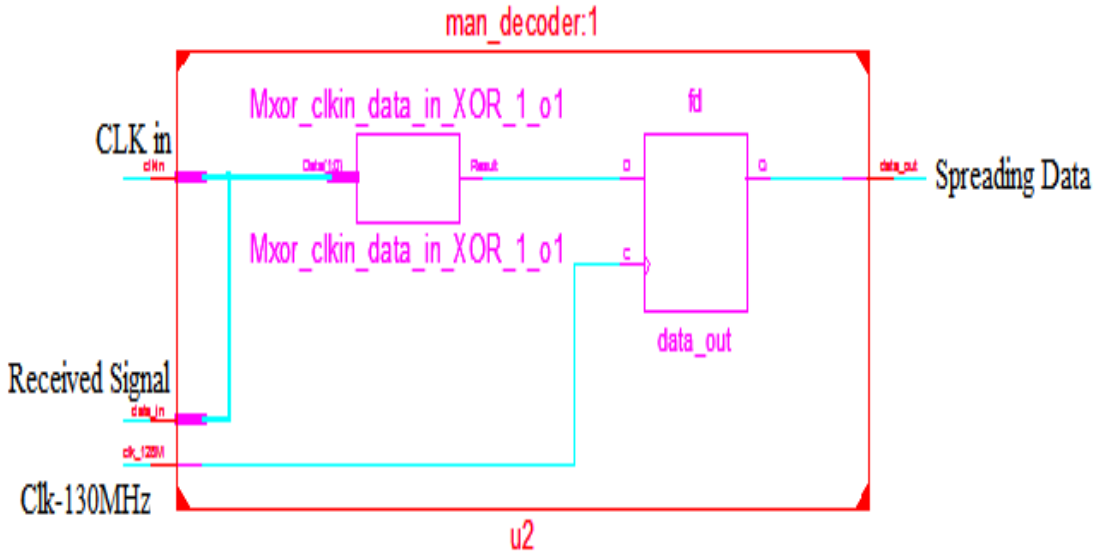

Fig. 5.16. Manchester decoder model.

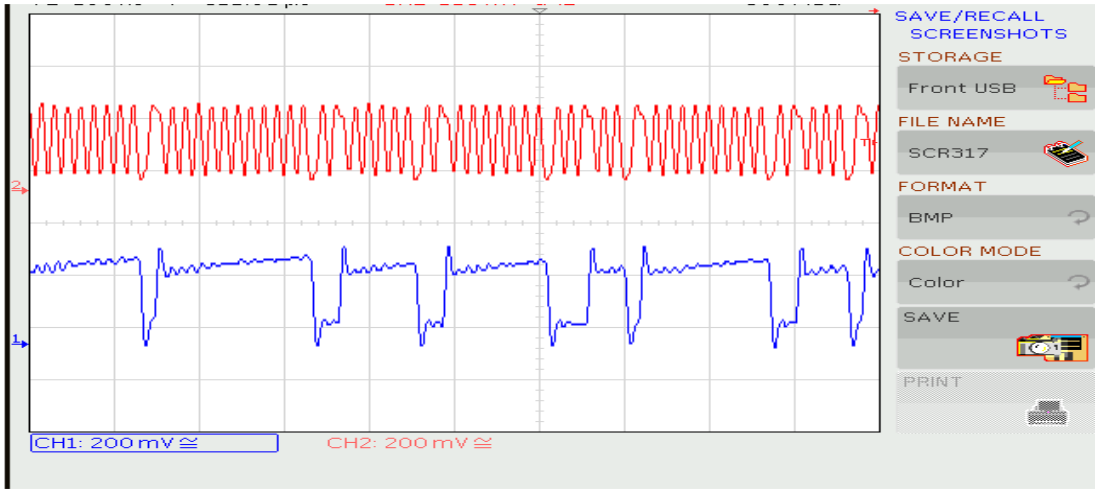

Fig. 5.17. Decoded user data spreading signal in real time, visualized by the oscilloscope. The blue signal is user data, whereas the red is encoded signal.

5.4.2 Block spreading synchronisation

Once the "sync" sequence is detected, the enable signal for user codes is initiated to synchronise the spreading user data with the user code. Fig. 5.18 demonstrates this occurrence.

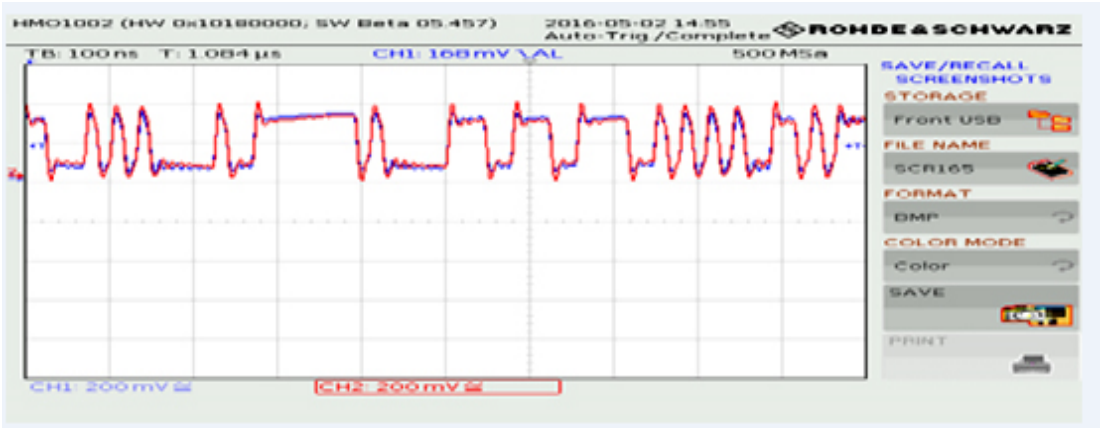

Fig. 5.18. Real-time synchronisation of transmitted spreading signals and user block signals, visualized by the oscilloscope.

5.4.3 Data recovery

Fig. 5.19 demonstrates the block-spreading communication system for four users.

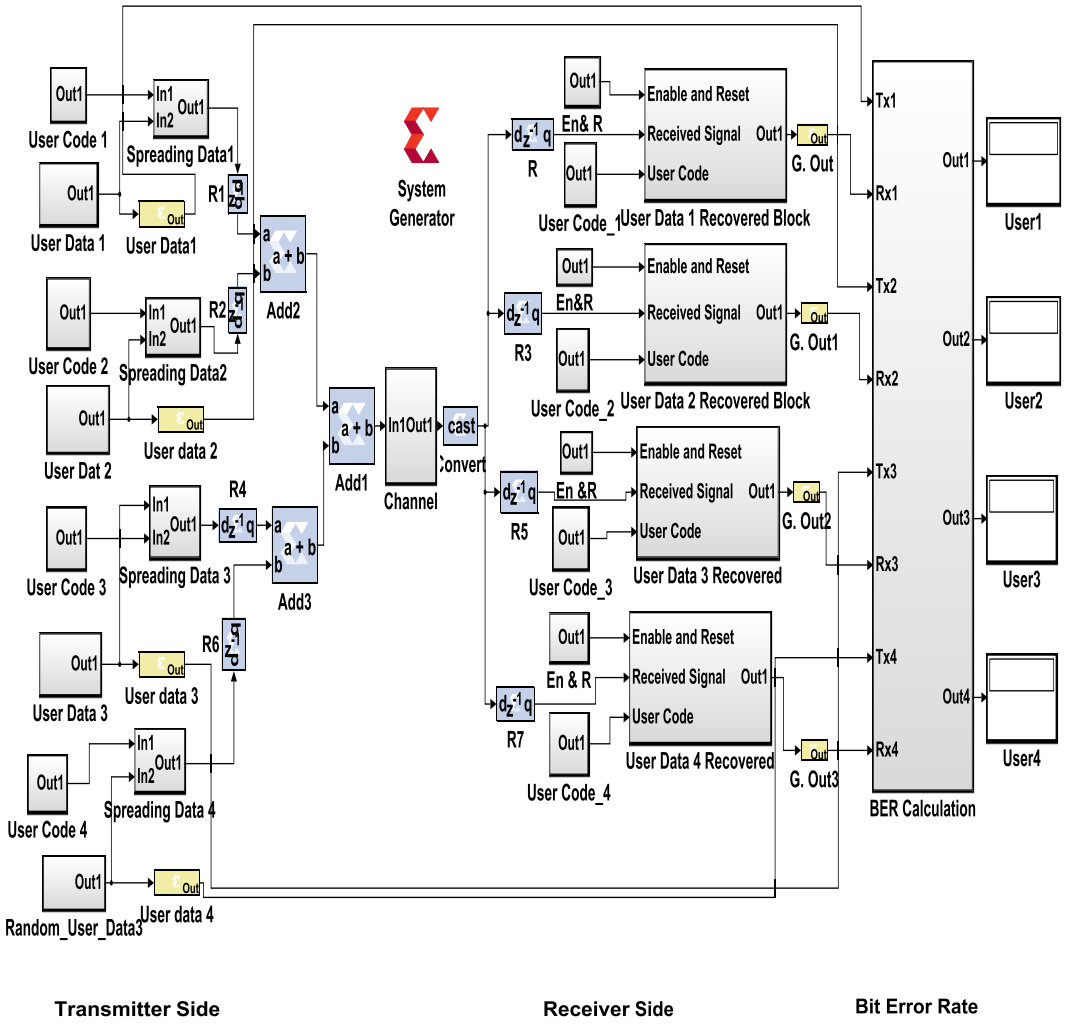

Fig. 5.19. Block-spreading communication system of four users.

Fig. 5.20 illustrates the simulated results of the transmitted and recovered data of the four users. The recovered user data is delayed compared to the transmitted user data because of the processes used for transmitting and recovery.

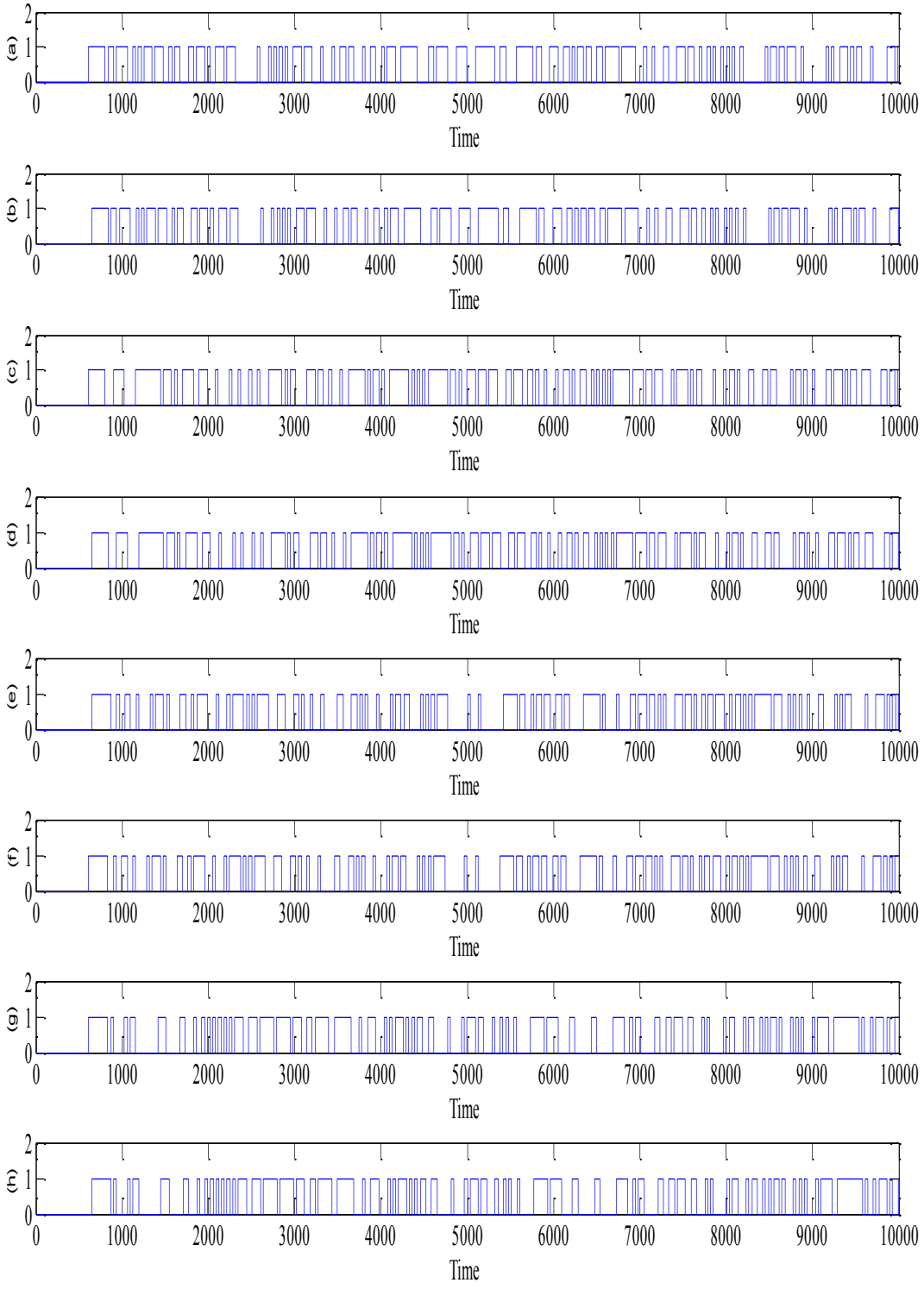

Fig. 5.20. Simulation test for four-user data recovery. (a) User data 1, (b) User data 1 recovery, (c) User data 2, (d) User data 2 recovery, (e) User data 3, (f) User data 3 recovery, (g) User data 4 and (h) User data 4 recovery.

The four user data are recovered well as shown in Fig. 5.21, Fig. 5.22, Fig. 5.23, Fig. 5.24.

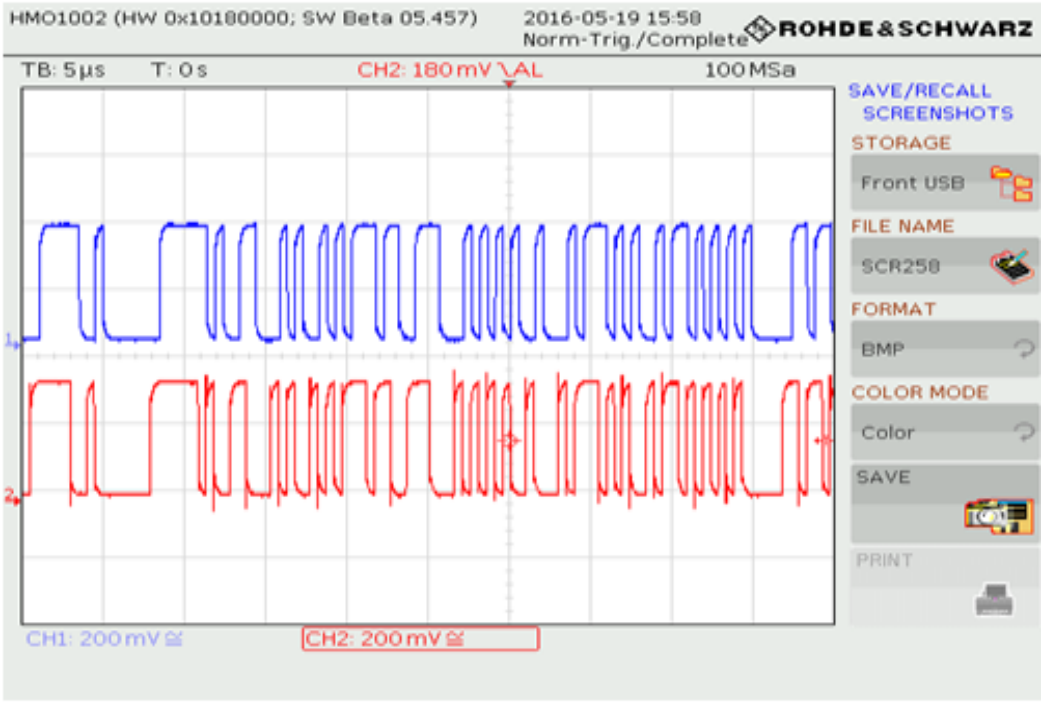

Fig. 5.21. User 1 data recovery in real time, visualized by the oscilloscope. The red signal is transmitted user data and blue signal is the recovered user data.

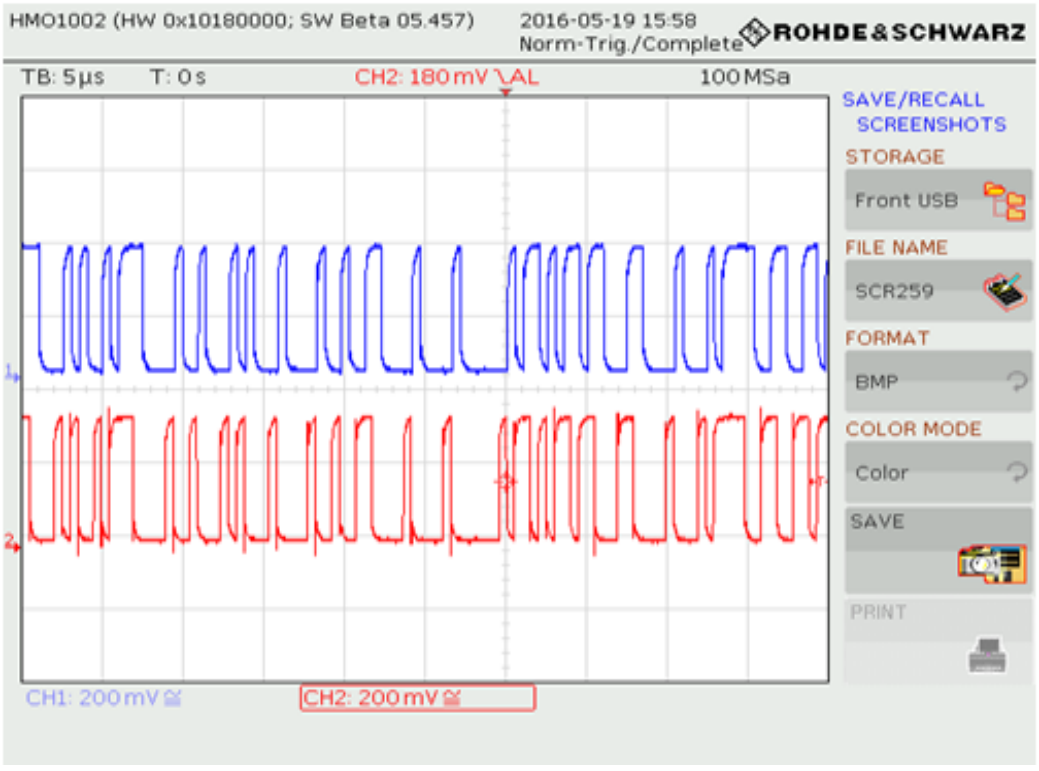

Fig. 5.22. User 2 data recovery in real time, visualized by the oscilloscope. The red signal is transmitted user data and blue signal is the recovered user data.

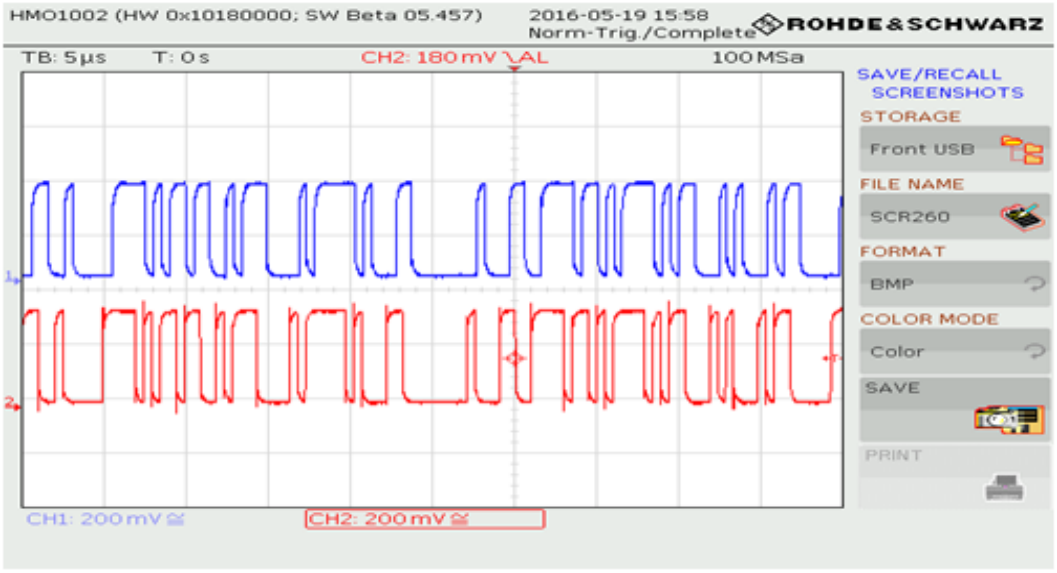

Fig. 5.23. User 3 data recovery in real time, visualized by the oscilloscope. The red signal is transmitted user data and blue signal is the recovered user data.

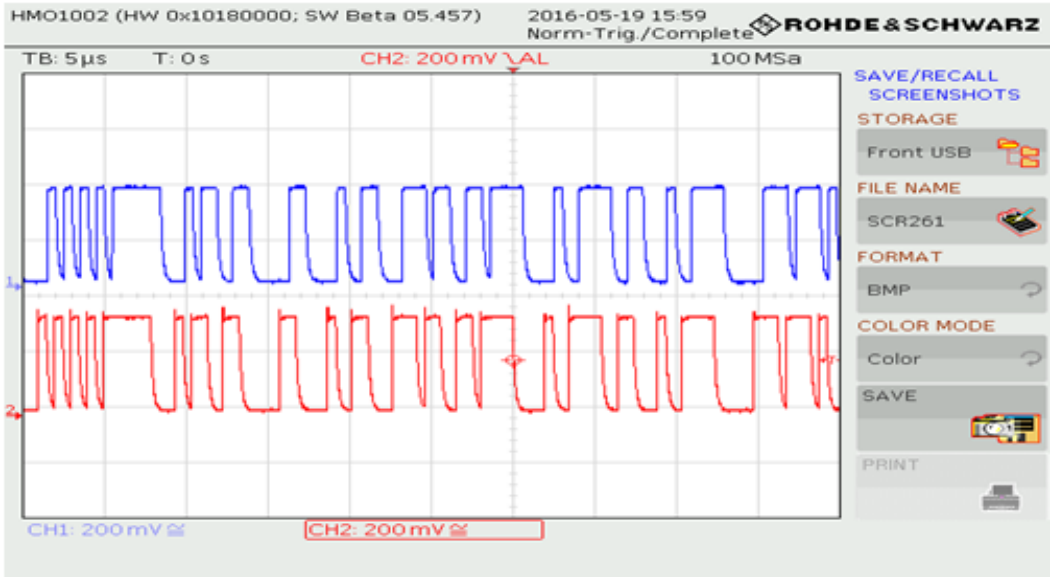

Fig. 5.24. User 4 data recovery in real time, visualized by the oscilloscope. The red signal is transmitted user data and blue signal is the recovered user data.

5.4.4 Integrated synthesis environment (Xilinx® ISE)

The receiver design contains all files related to the project as follows.

- Digital clock management is provided by using IP (Core generator and Architecture Wizard).
- Clock recovery.

- Phase Locked Loop (PLL).
- Manchester decoder based on VHDL code.
- Receiver design based on System Generator®.

Fig. 5.25 shows the top-level register-transfer level (RTL) graphical representation of the receiver design.

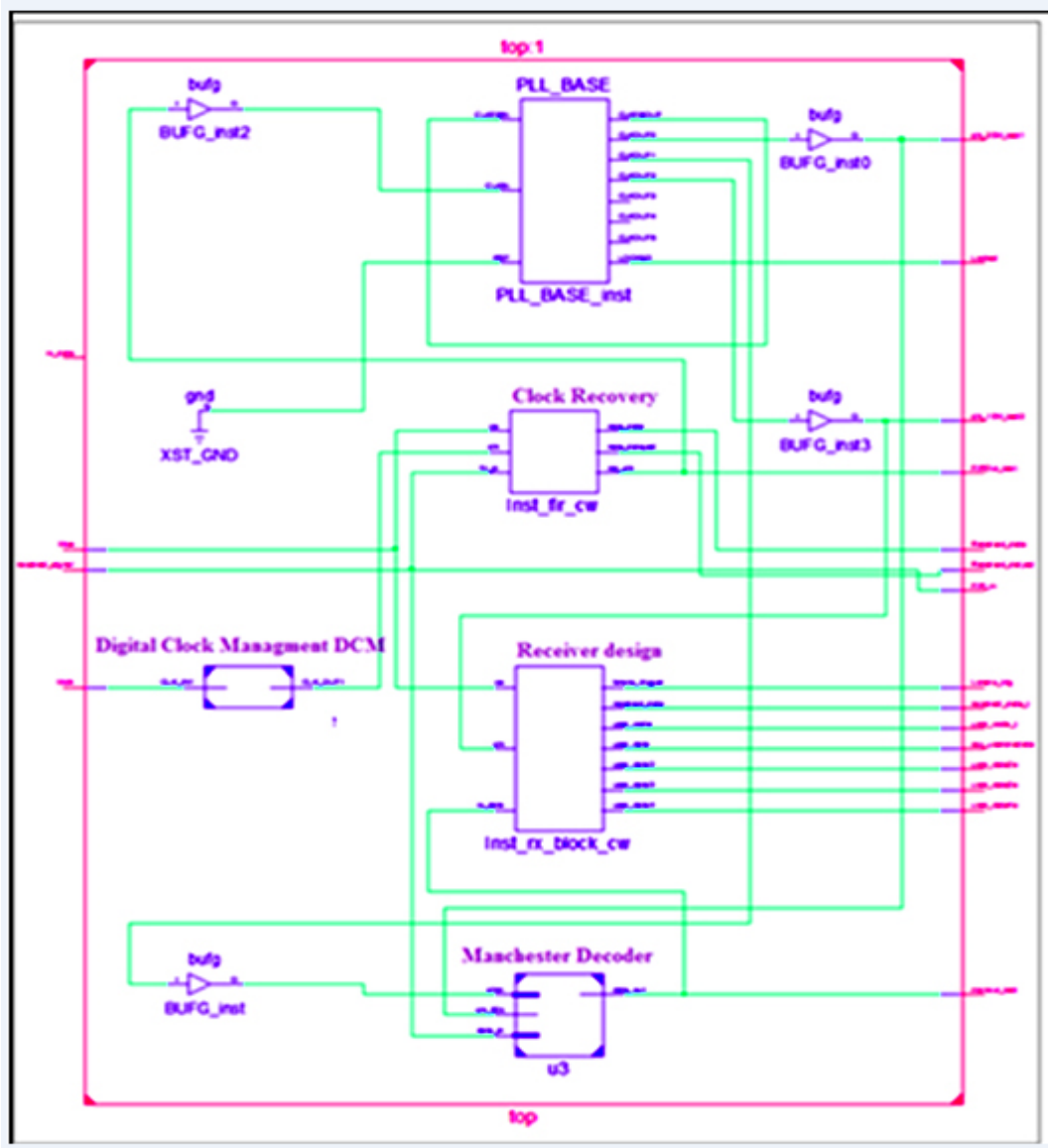

Fig. 5.25. RTL schematic of the receiver design.

5.4.5 Device utilisation summary

Table 5. *3* provides a device utilisation summary of the target device 'xc6slx45t-3fgg484'. From this summary, we can determine that the receiver design can be implemented in a smaller chip.

Slice Logic Utilisation	*Used*	*Available*	*Utilisation*
Number of Slice Registers	563	54,576	1%
Number Used as Flip Flops	563		
Number Used as AND/OR Logics	0		
Number of Slice LUTs	576	27,288	2%
Number Used as Logic	389	27,288	1%
Number Used as Memory	93	6,408	1%
Number Used Exclusively as Route-Thrus	94		
Number of Occupied Slices	215	6,822	3%
Number of MUXCYs Used	256	13,644	1%
Number of Fully-Used LUT-FF Pairs	395	657	60%
Number of BUFG/BUFGMUXs	6	16	37%
Number of DSP48A1s	24	58	41%
Number of PLL_ADVs	2	4	50%
Number of BUFIO2FB/BUFIO2FB_2CLKs	1	32	3%
Number of LOCed IOBs	4	17	23%

Table 5. 3. Device Utilisation Summary of the Target Device.

5.5 Conclusion

This chapter presents a digital communication system based on a Lorenz block-spreading communication system. A digital spreading communication system with four users has been implemented using two Spartan 6 FPGA boards. The aim of this chapter is to develop reliable method of synchronisation and data recovery. Codes that have been used in this work are extracted from the Lorenz generators. Each user code length is 32-bits. The method that has been used to retrieve the user data transmitted is based on cross-product and summation. The transmitter design has consumed only 1% of the slice registers and Look Up Table (LUT). Thus, it can be implemented in a smaller chip. On the other hand, the receiver design has consumed 1% of the slice registers and LUT which means that the system can be implemented in a smaller chip. The data transmitted for all four users are recovered perfectly at the receiver FPGA board. The communication system achieved a data rate of 2 Mbps.

Chapter 6

FPGA IMPLEMENTATION OF COMMUNICATION SYSTEMS USING CHAOTIC STREAM CIPHER

6.1 Introduction

In this chapter, we are explaining the implementation of the stream cipher. The challenges of this implementation are clock recovery and synchronisation of two chaotic generators.

High-speed processing of modern computers, powerful field-programmable gate array (FPGA) based boards and graphics processing unit (GPU) based boards help hackers attack communication systems at high frequencies [1]. There is always a need to improve security and develop systems secured by a perfect cipher that can defeat existing attacks. The present chapter presents such an approach.

Many implementation methods have been proposed to enhance system security [78, 97, 100, 101, 140, 141]. Several communication systems based on chaotic encryption have been reported that claim security against electronic attacks, especially brute force attacks. However, most of these systems have security weaknesses [104, 142, 143]. Further, most of the reported schemes do not explain well the major security properties of the cryptosystem, such as the level of security, performance and ease of implementation.

These properties play a major role in the cryptosystem evaluation. Any limitation in any one of these properties of the proposed cryptosystem implementation will negatively affect the estimation of the system reliability [103].

Chaos-based communication systems are well-ranked with regards to these properties, which makes them suitable for cryptography applications [6, 9, 11, 54, 90, 142]. However, the dynamics of the basic chaotic system are not completely secure, and they may be vulnerable to attack if designed with weaknesses [99], such as fixed parameters, finite precision, fixed point arithmetic, short cycle length and non-ideal

distribution. This work explains how these weaknesses can be overcome and highly secure system can be designed.

6.2 The Cryptosystem

The cryptosystem has been explained in detailed in chapter 3 (section 3.2.2). The SIMULINK design of the cryptosystem has been converted into System Generator® blocks. Fig. 6.1 shows the block diagram of the encryption process.

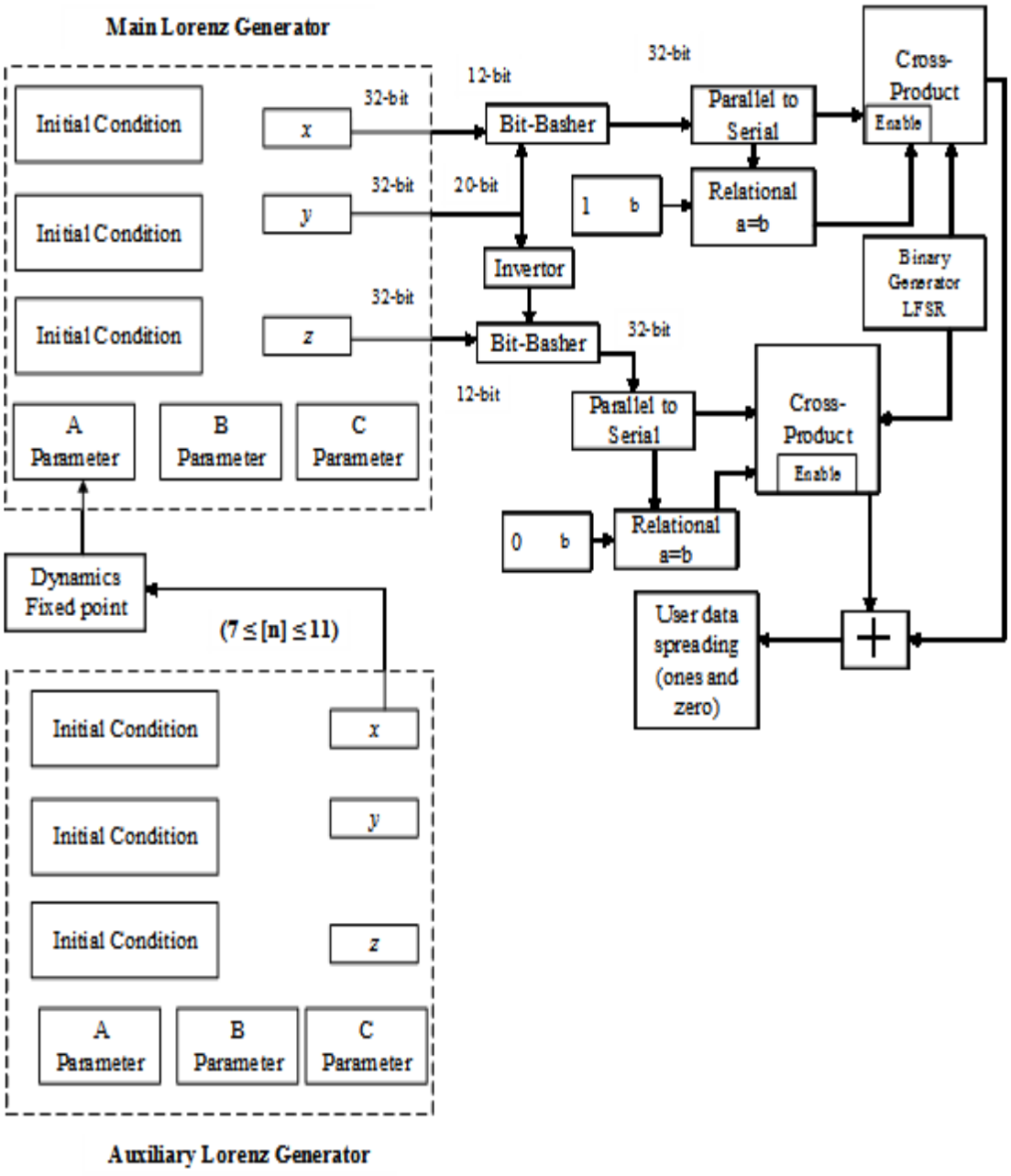

Fig. 6.1. The Block Diagram of the encryption process at the transmitter.

To increase security, the user data (binary stream) is spread using two of the three output signals of the Main Lorenz Generator. If the current bit at the user data stream (b[n]) represents a binary one, the Bit-Basher will concatenate the first 12-bits of the output *x*[n] with the last 20-bits of the output *y*[n]. In the case that the current bit represents a binary zero, the Bit-Basher will concatenate the first 12-bits of the output *z*[n] with the *inverted* last 20-bits of the output *y*[n]. The spectrum spreading is implemented by a 32-bit multiplier that produces the product of the current bit (b[n], representing either a one or a zero) with the 32-bits output of the Bit-Basher. Fig. 6.2 shows the block diagram of the scrambling process.

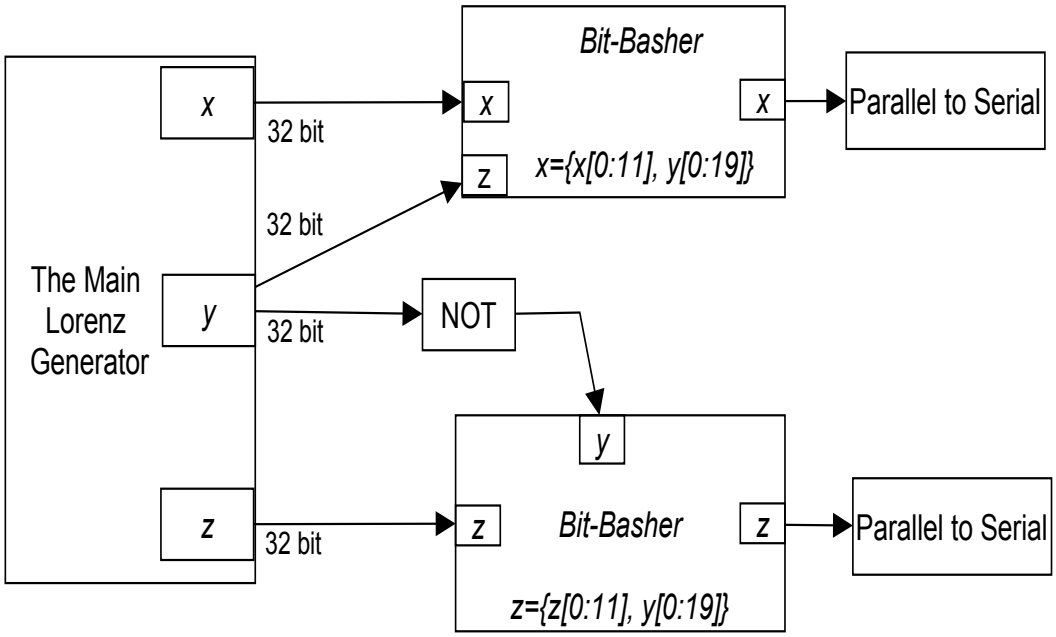

Fig. 6.2. The Block Diagram of the scrambling process.

To test the system, a Linear Feedback Shift Register (LFSR) is used to generate a binary user data stream (b[n]). The Bit-Basher operates to scramble the bit stream of the Main Lorenz Generator so as to produce an encrypted binary stream that satisfies the randomness test.

6.3 Cryptosystem Implementation Overview

Fig. 6.3 shows the block diagram of the complete system process. Fig. 6.4 shows the system flow chart of one user stream cipher based on the Lorenz model. The transmitter side constitutes the following subsystems, Preamble subsystem, Sync

sequence of known data subsystem, User data generator subsystem, Main Lorenz model, Auxiliary Lorenz model, “ones” user data spreading subsystem, “zeros” user data spreading subsystem and Manchester encoder subsystem. The receiver constitutes the following subsystems, clock recovery, Manchester decoder, PLL, sequence detector, Main Lorenz model, Auxiliary Lorenz model, data recovery and system clock.

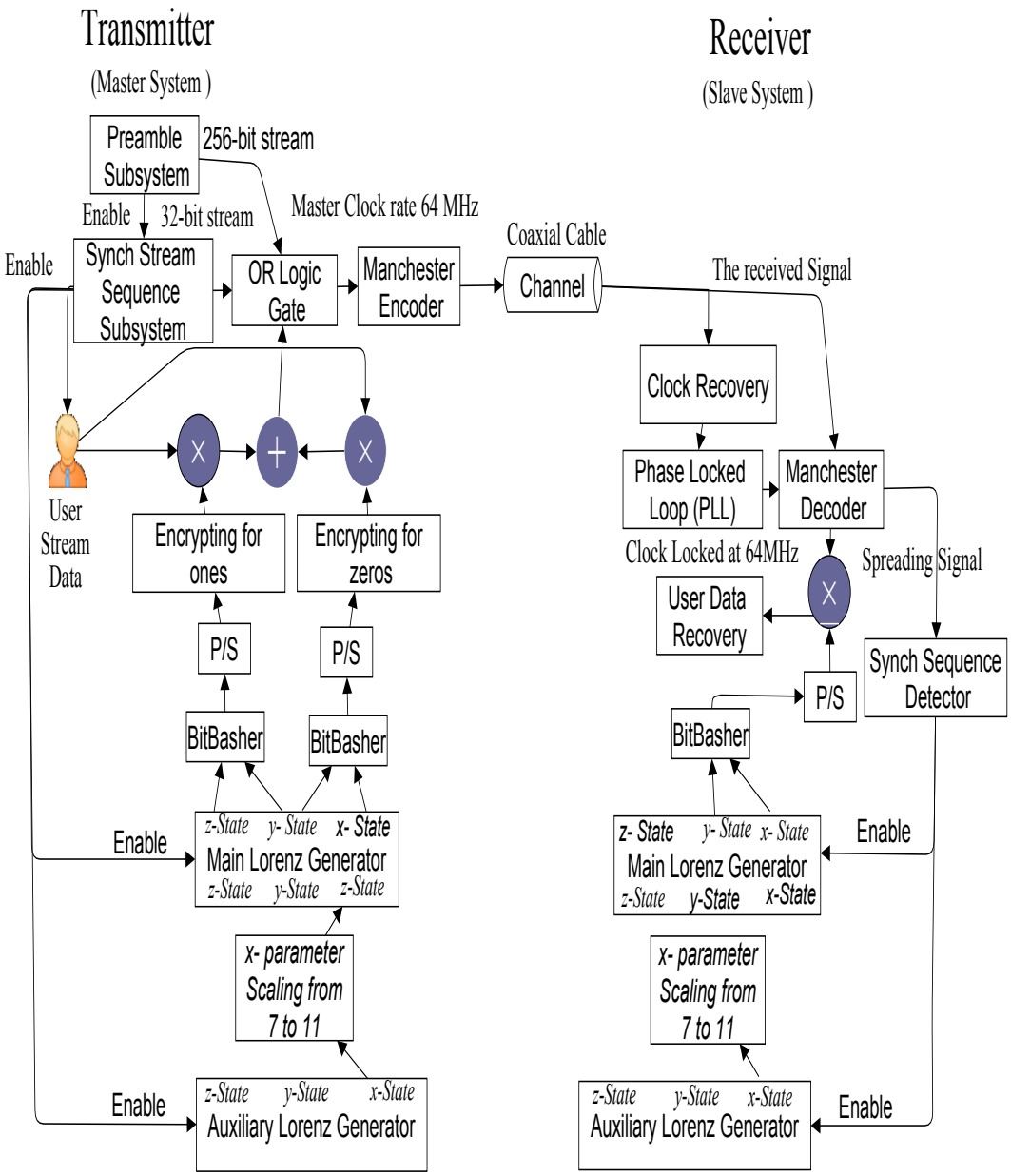

Fig. 6.3. The block diagram of the complete system process.

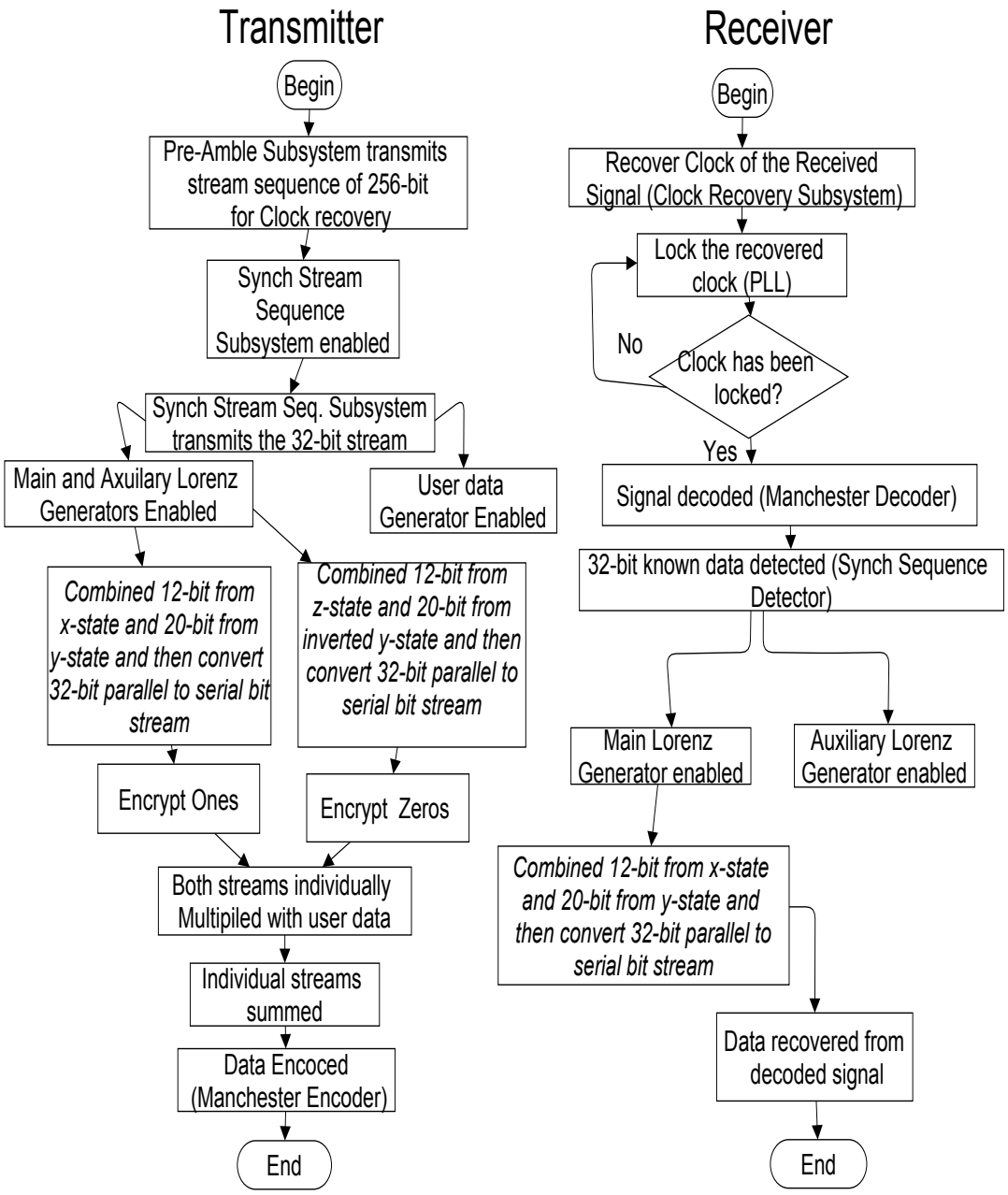

Fig. 6.4. System flow chart of one user stream cipher based on Lorenz model.

6.4 Implementation of the Lorenz Model

The Lorenz generator (Simulink simulation model discussed in chapter 3) is implemented using the Xilinx System Generator®. The operational signals are represented by a 32-bit fixed-point fractional data-type with 25 fractional bits (the only exception being the user data stream b[n]).

As per the Lorenz equations actualized by the implementations of the model (Main Lorenz Generator and Auxiliary Lorenz Generator), there are three system parameters and three initial conditions.

In order to visualize the chaotic signal outputs of the Main Lorenz Generator, a "black box" constituting VHDL code is used to convert the 32-bit fixed-point fractional to a 12-bit integer for a Digital to Analog Converter (DAC) IC on the FPGA board. The analogue output produced from the *x*, *y*, *z* outputs of the Lorenz generator are thus represented by a voltage ranging between 0 and 2.5 V and this is connected to an oscilloscope to visualize the operation. The Digilent® Peripheral Module Interface Digital to Analog Convertor (PMODDA) module is used with the FPGA board, which interfaces the 12-bit Serial Peripheral Interface (SPI) 8-Channel DAC IC (Analog Devices AD5628). Fig. 6.5 depicts the Main Lorenz Generator. Fig. 6.6 depicts Lorenz chaotic signals. Fig. 6.7, Fig. 6.8 and Fig. 6.9 show the Lorenz attractors. Fig. 6.10 depicts real time signal of the Lorenz chaotic signals and Fig. 6.11 shows the Lorenz attractor.

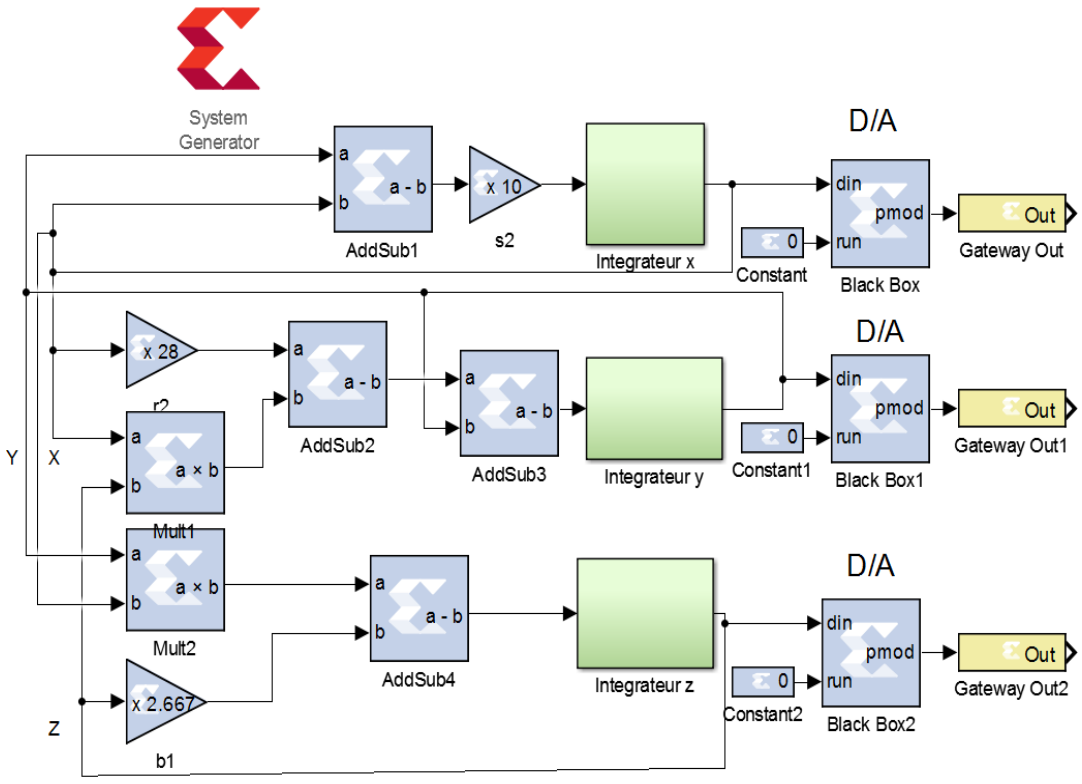

Fig. 6.5. The Lorenz Model, as viewed in the Xilinx System Generator®. This model is implemented as the Main Lorenz Generator and the Auxiliary Lorenz Generator.

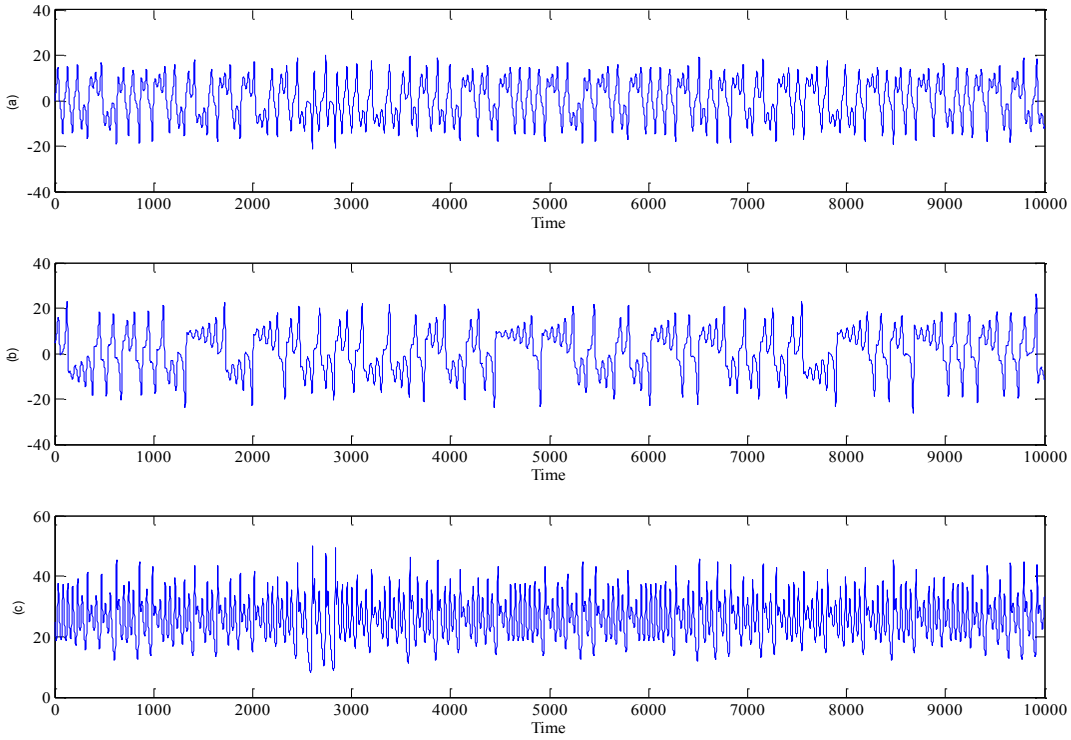

Fig. 6.6. Lorenz chaotic signals, (a) *x* signal, (b) *y* signal and (c) *z* signal.

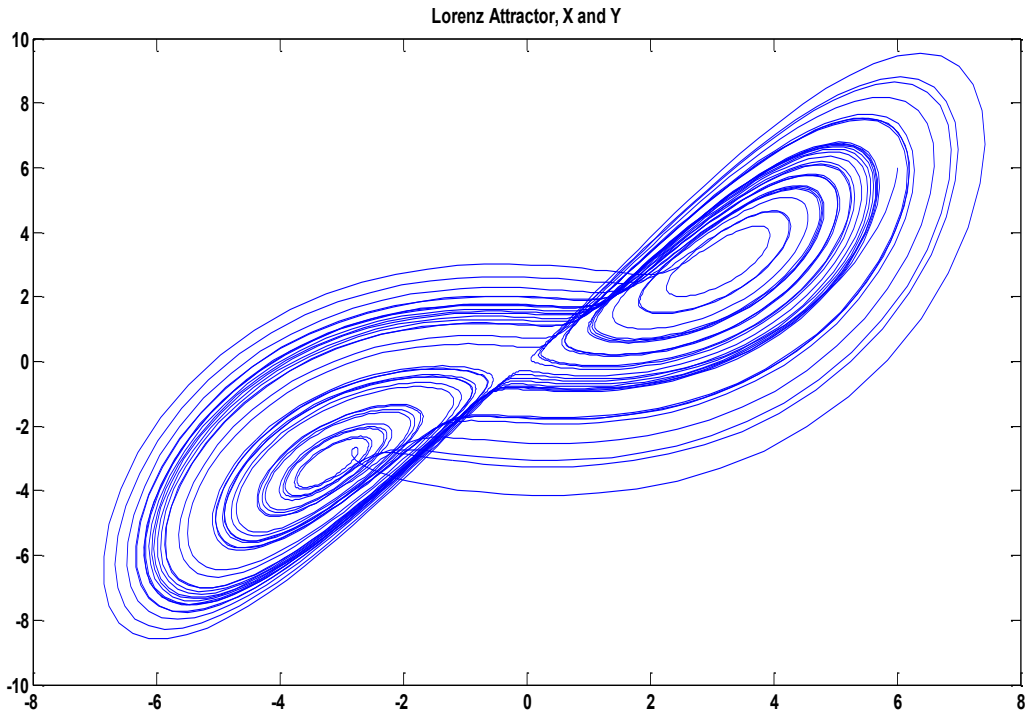

Fig. 6.7. The *x*-*y*- attractor.

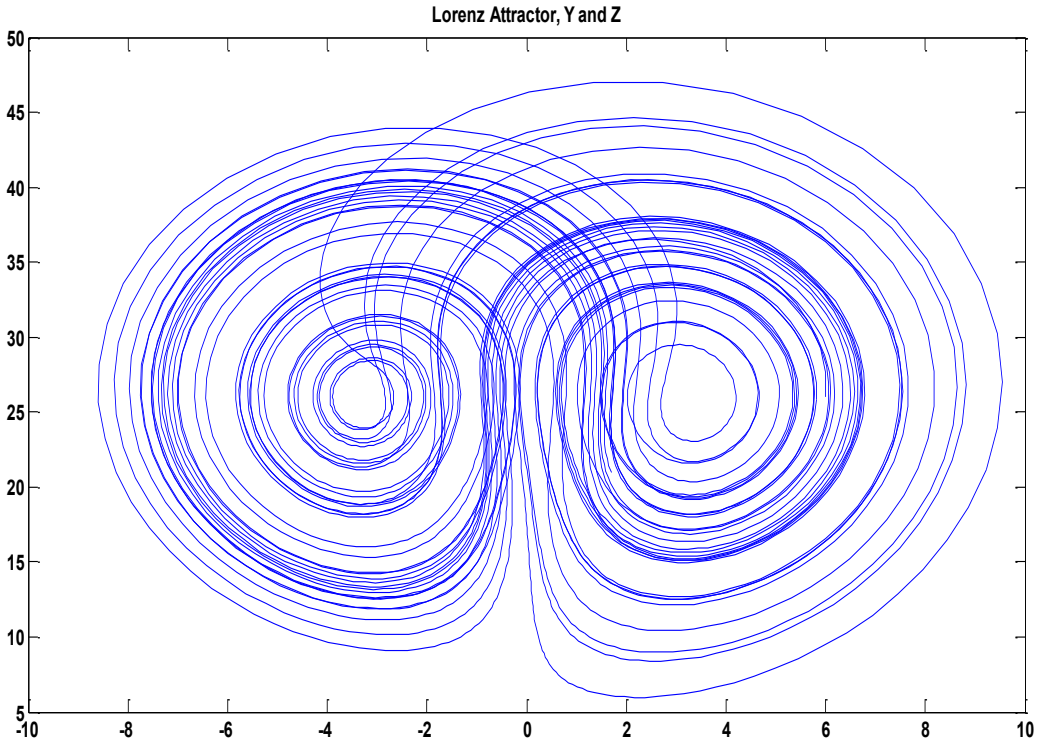

Fig. 6.8. The *y*-*z* attractor.

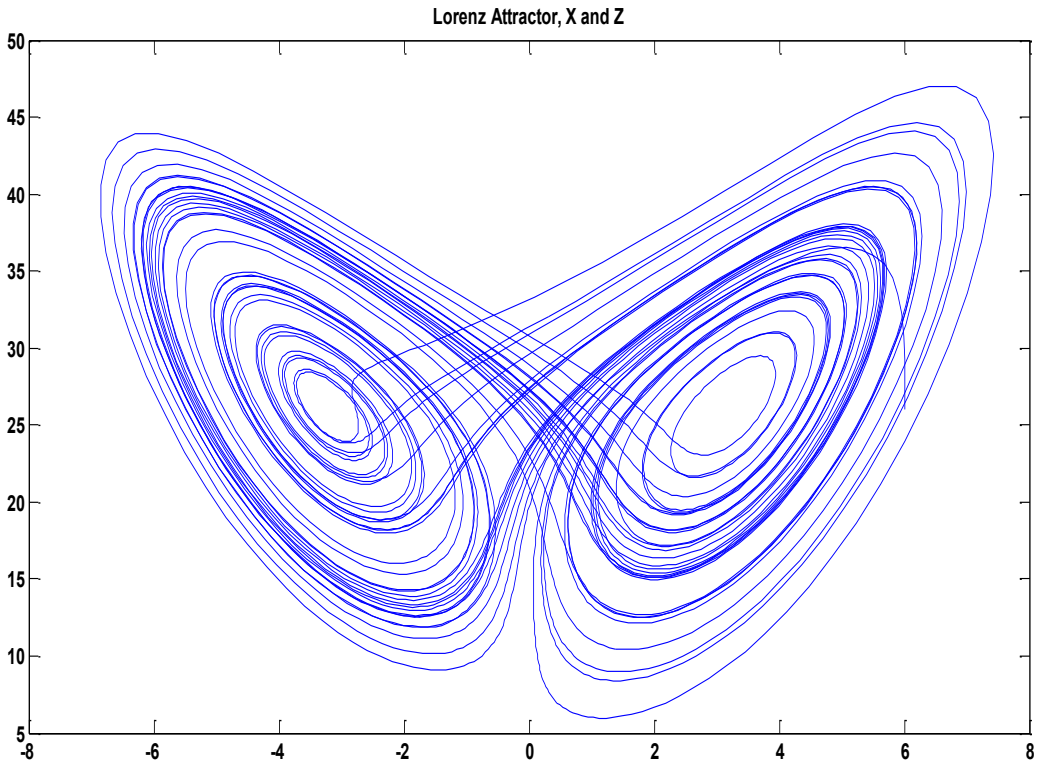

Fig. 6.9. The *x*-*z* attractor.

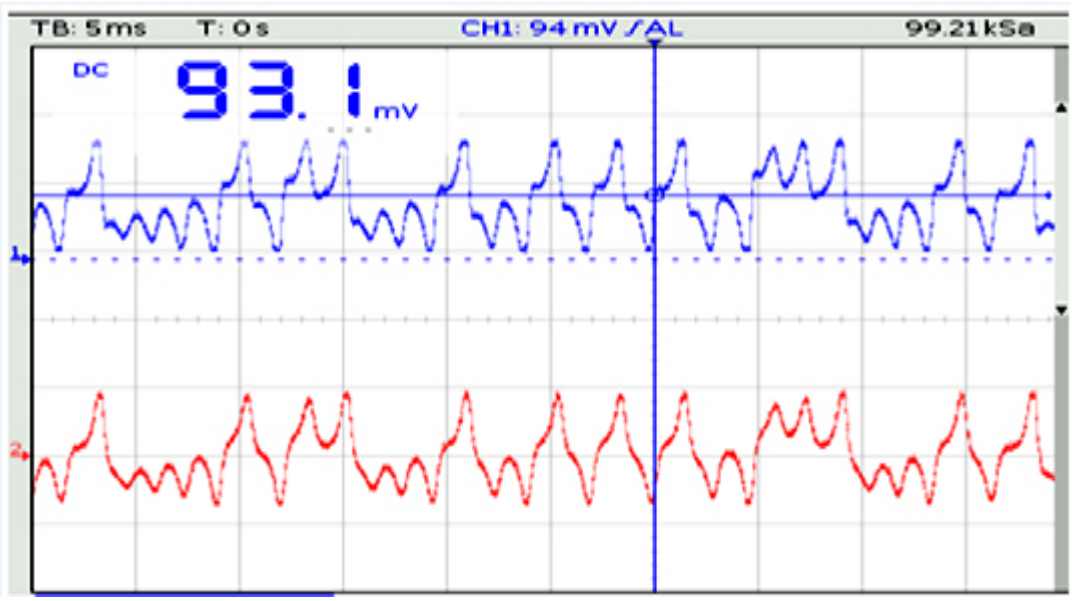

Fig. 6.10. Chaotic signal of the Lorenz model in real-time, as visualized by the oscilloscope (*x* is represented by the blue signal, *y* is represented by the red signal).

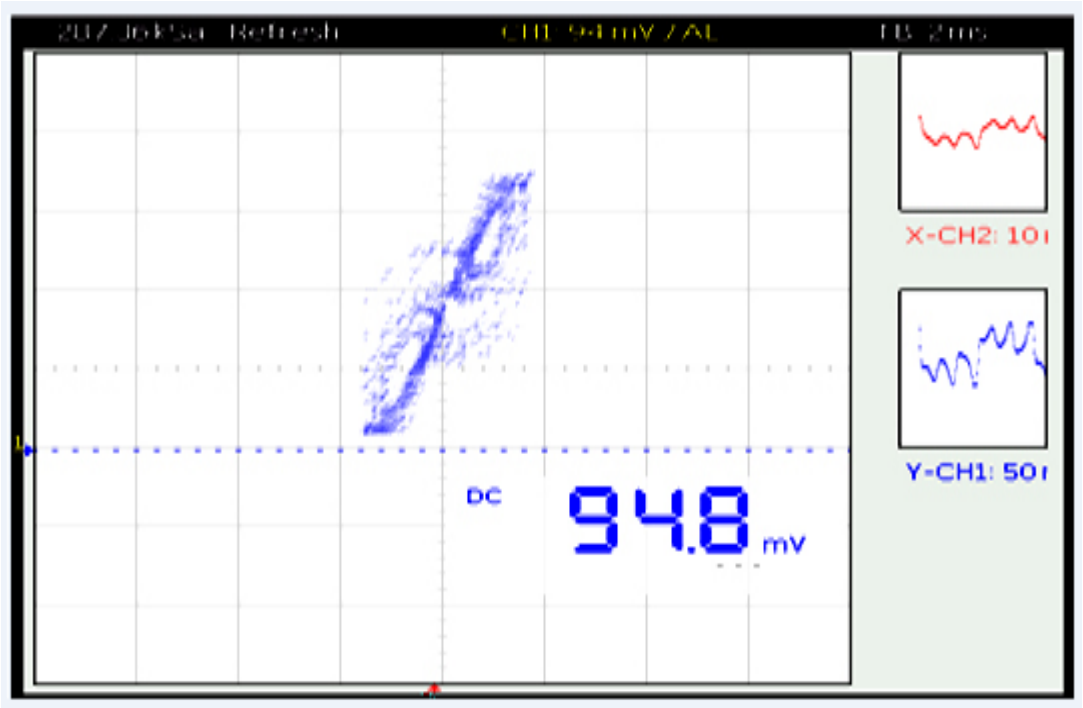

Fig. 6.11. The *x*-*y* attractor in real-time, as visualized by the oscilloscope.

6.5 Implementation of the Transmitter

Hardware properties of the cryptosystem are presented in Table 6. 1.

Main Clock Frequency	20 MHz oscillator
Modulation	Spread Spectrum (SS)
Spreading Code	32-bit
Spreading User Data Frequency	64 MHz
User Data Frequency	2 MHz
Data Rate	2 Mbps
FPGA Development Board	SP605
FPGA Family	Xilinx® Spartan 6
FPGA (IC) Model Number	xc6slx45t-3fgg484
Number of user data streams generated and encrypted	One
Method of user data stream production	Linear Feedback Shift register (LFSR)

Table 6. 1. Fixed properties of the cryptosystem implementation.

The transmitter consists of four System Generator® designs, which are wrapped up later by using Xilinx ISE ®. These are preamble and sync sequence generator, user data encryption and spreading, Stream cipher based on two Lorenz generators and parallel to serial convertor. Fig. 6.12, Fig. 6.13, Fig. 6.14 and Fig. 6.15 show the System Generator® designs.

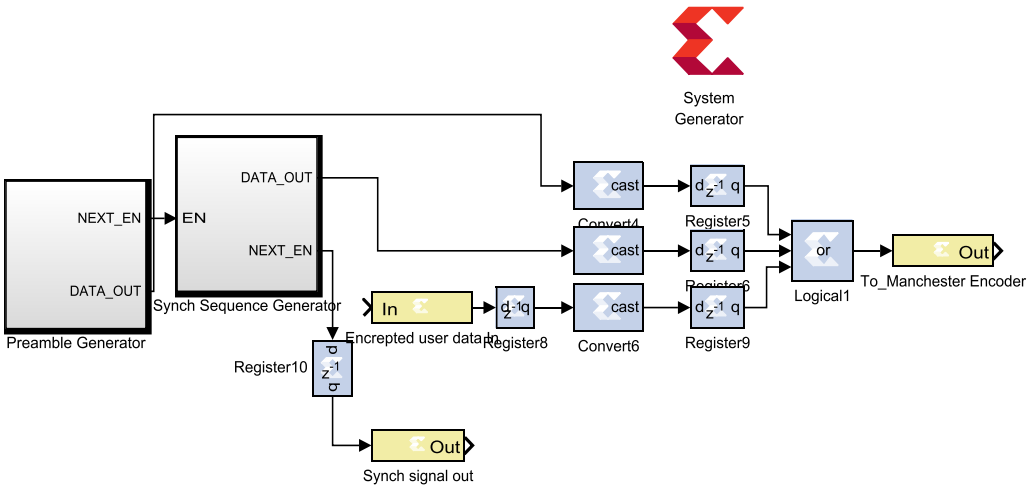

Fig. 6.12. Preamble and sync sequence generator, as viewed in the Xilinx® System Generator®.

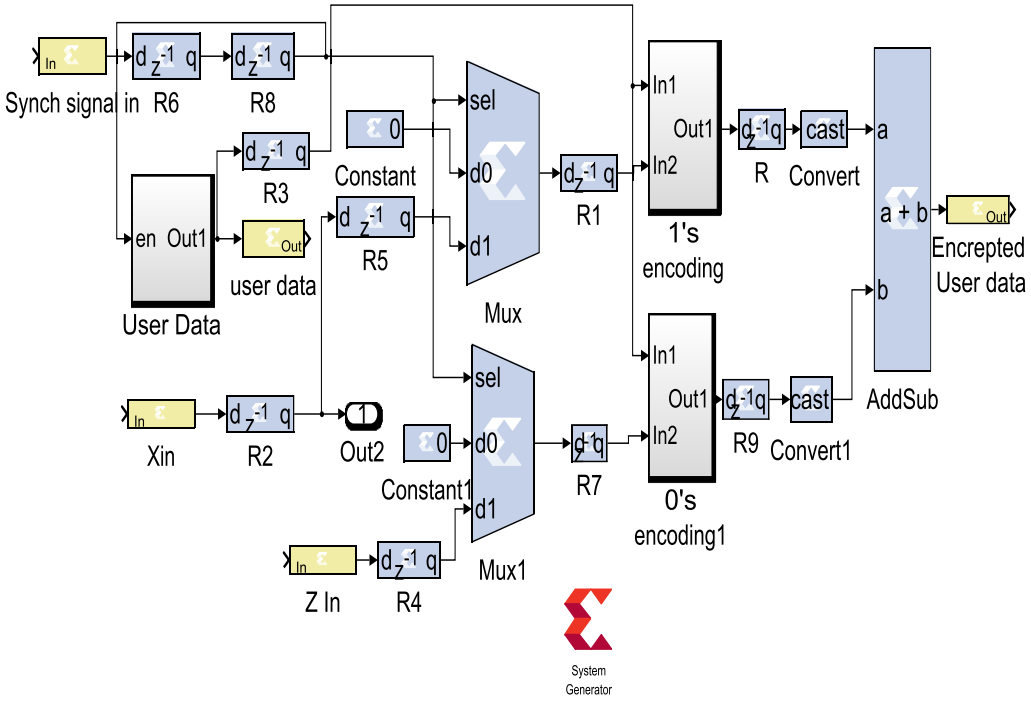

Fig. 6.13. User data encryption and spreading, as viewed in the Xilinx® System Generator®.

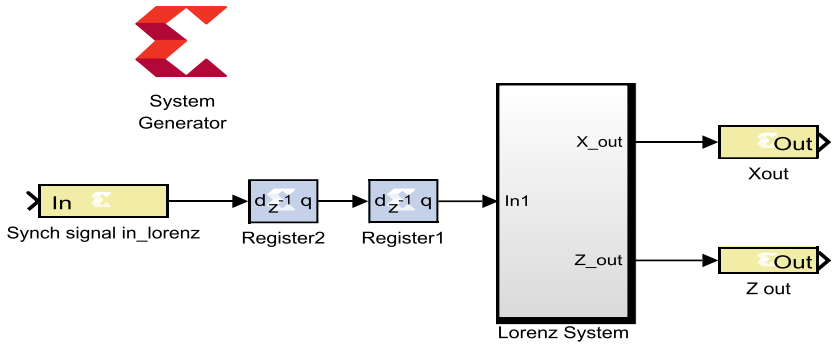

Fig. 6.14. Stream cipher based on two Lorenz generators, as viewed in the Xilinx System Generator®.

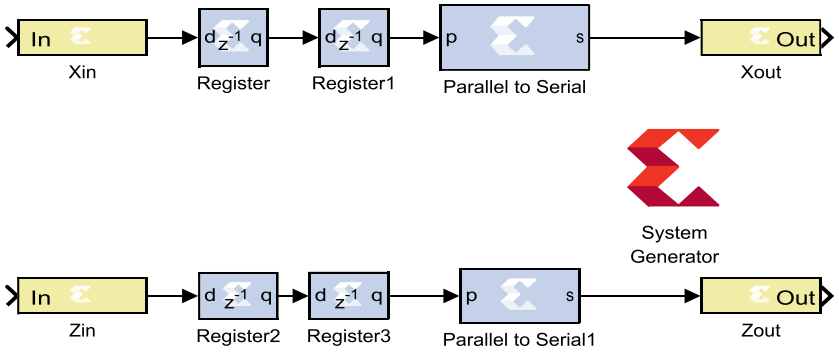

Fig. 6.15. Parallel to Serial convertor, as viewed in the Xilinx System Generator®.

Fig. 6.16 shows that the register transfer level (RTL) graphical representation of the transmitter design.

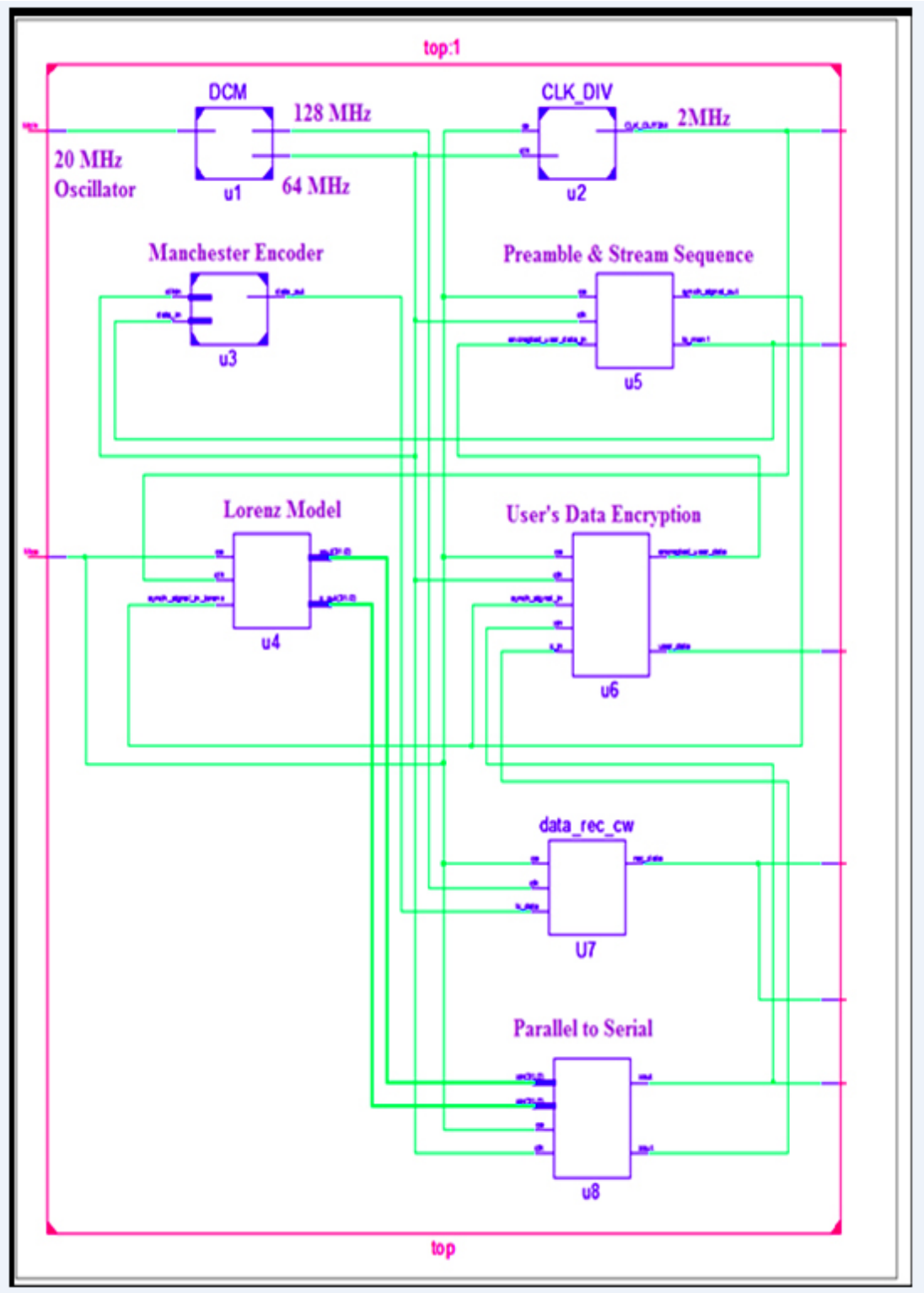

Fig. 6.16. RTL of the transmitter design.

6.5.1 User data spreading

Spread Spectrum technique is used to map the binary user data by using multiplication. A 32-bit spreading code used. Then, the two spreading signals are combined together to generate a secure bit stream. The task of the system model is to encrypt the "ones" and "zeros" user data based on the Lorenz chaotic signal of *x*-state and *z*-state. The multiplexers are used with two inputs. The first input is zero constant, and the second input is chaotic signal. The multiplexer output is zero until the select (SEL) receives the "ones" signal, and then, the multiplexer output is the chaotic signal. Fig. 6.17 shows the System Generator® model for user data encryption. Fig. 6.18 shows the simulation test of "ones" user data encrypted by using the *x*-state chaotic signal. Fig. 6.19 shows the simulation test of "zeros" user data encrypted by using the *z*-state chaotic signal.

Fig. 6.20 shows the two spreading signals combined together. Fig. 6.21show a real-time test of the user data spreading of ones, and Fig. 6. 22. Fig. 6.23 shows the two spreading signals are combined together.

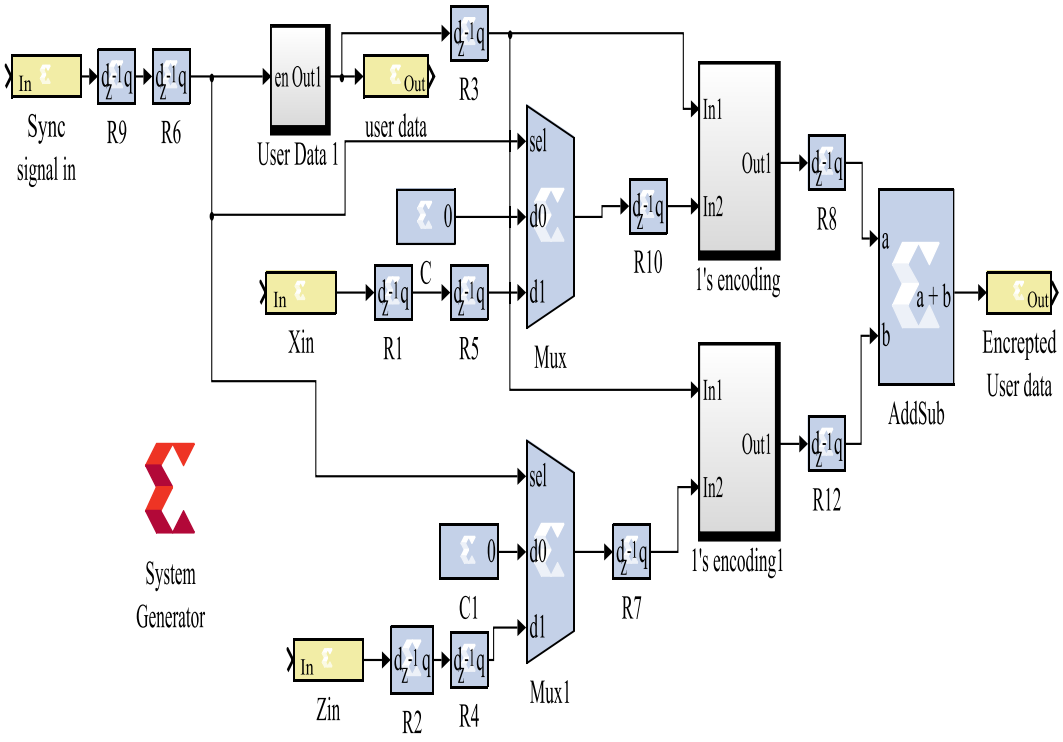

Fig. 6.17. User data encryption, as viewed in the Xilinx System Generator®.

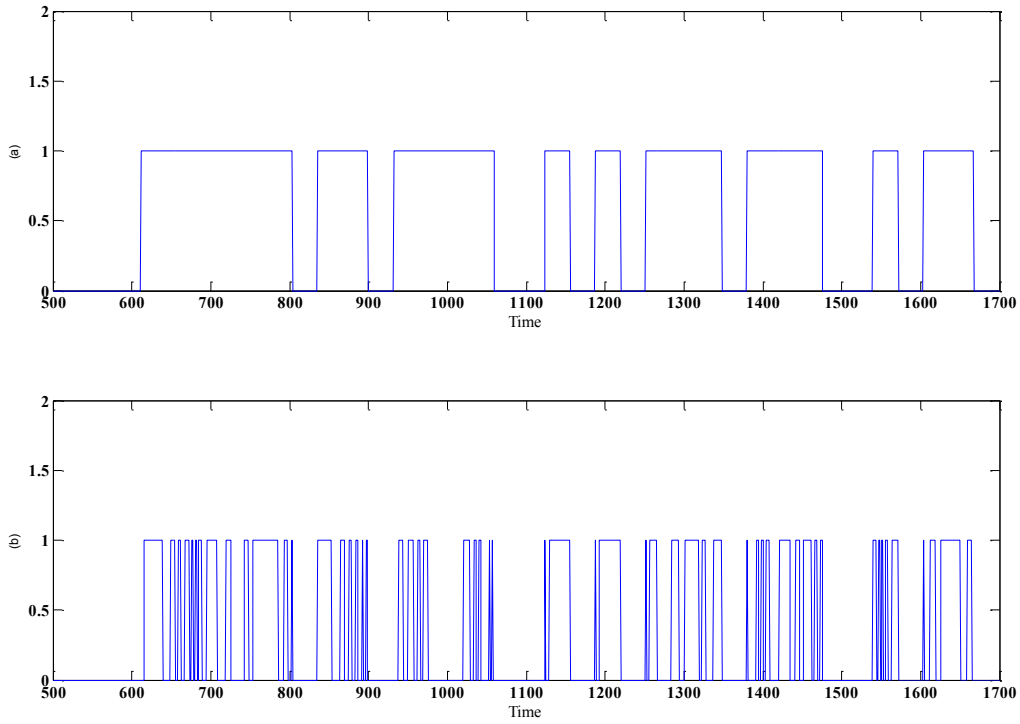

Fig. 6.18. Simulation test. (a) User data and (b) Encoded user data, only ones.

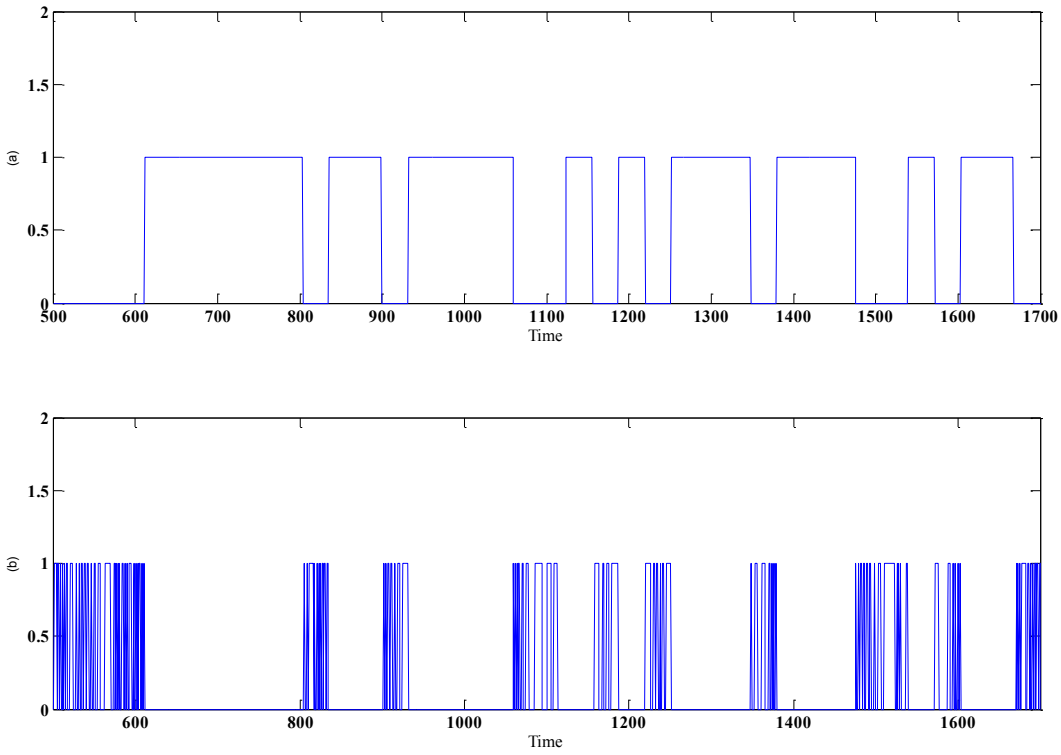

Fig. 6.19. Simulation test. (a) User data and (b) Encoded user data, only zeros.

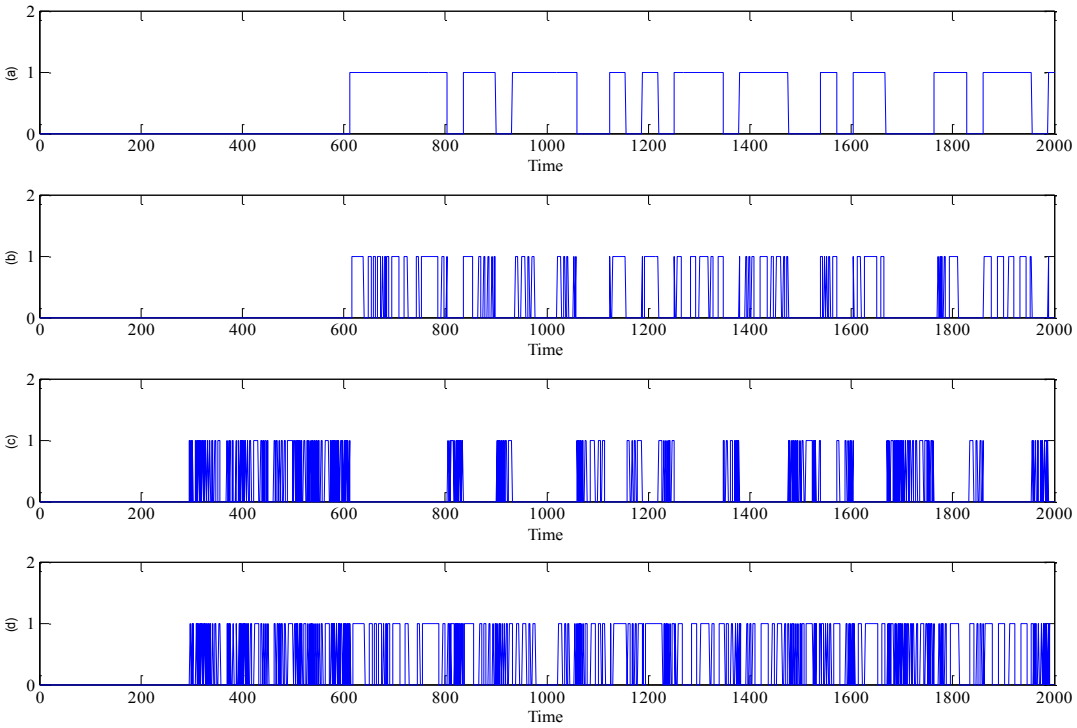

Fig. 6.20. Simulation results. (a) User data, (b) User data encoded ones, (c) User data encoded zeros and (d) Ones and zeros are combined.

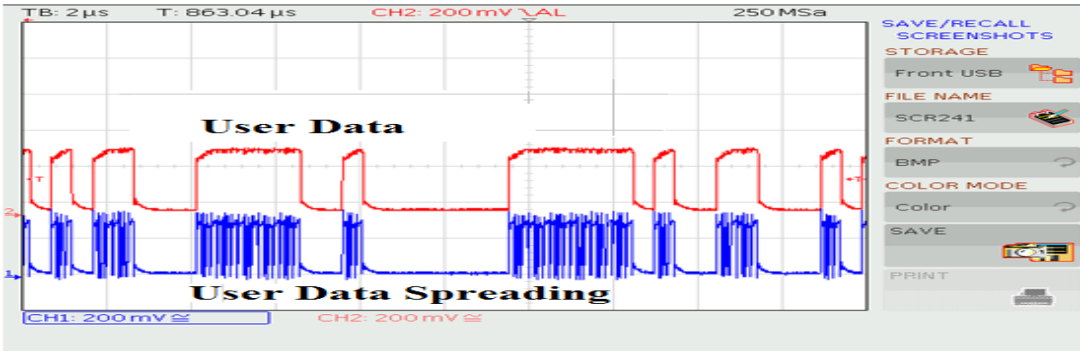

Fig. 6.21. User data has spread using 32-bits.

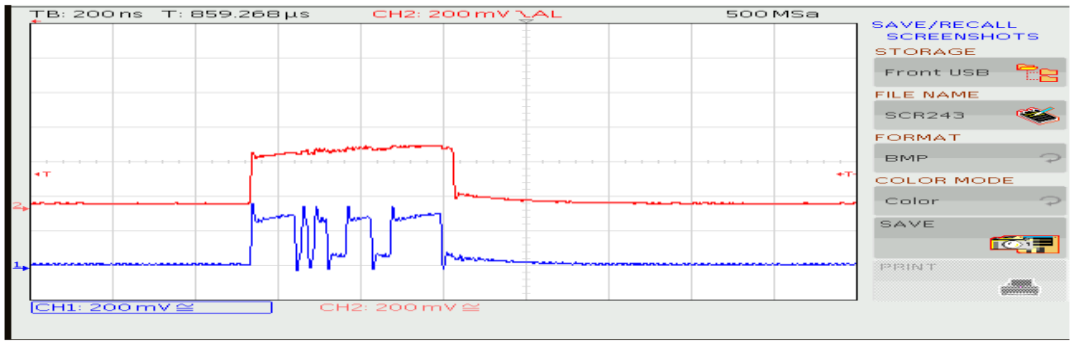

Fig. 6. 22. User data spreading using 32-bits length.

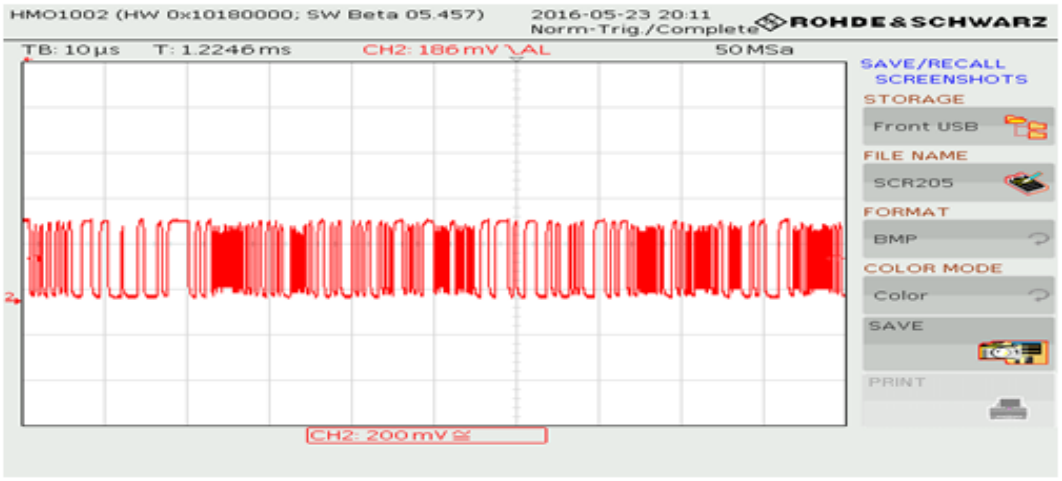

Fig. 6.23. Real-time test of the user data spreading of ones and the two spreading signals are combined.

6.5.2 A stream cipher where the encryption key is continuously changing

The Auxiliary Lorenz model is used to change the one of the parameters of the main Lorenz model with time. In order to keep the chaotic signal output response of the Lorenz model, a scaling model is added. Thus, the output signal of the scaling model is fluctuating between 7 and 11. The scaling model consists of multiplier block, adder and constant value. Fig. 6.24 shows the auxiliary Lorenz model and a scaling model.

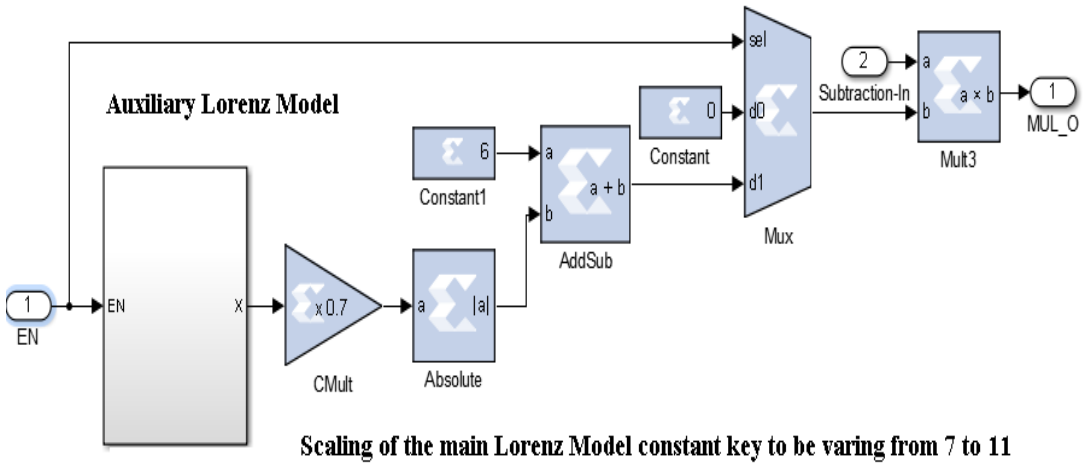

Fig. 6.24. Auxiliary Lorenz generator, as viewed in the Xilinx System Generator®.

6.5.3 Parallel to serial

Although the output is fixed point, with a binary point at 20, the Lorenz model is represented by 32 bits in parallel. The parallel-to-serial System Generator® block is used to serialize the chaotic signal.

6.5.4 Manchester encoder

The Manchester encoder is used in the transmitter design for clock recovery. The Manchester encoding is used in data transmission to allow the receiver to easily synchronize with the transmitter. It splits each bit period into two and ensures that there is always a transition between the signal levels in the middle of each bit. The Manchester encoder is established by using VHDL code, and the operation is performed by XOR of the data with the clock.

Fig. 6.25 shows the encoding of the spreading signal by using the Manchester encoder at the transmitter and receiver.

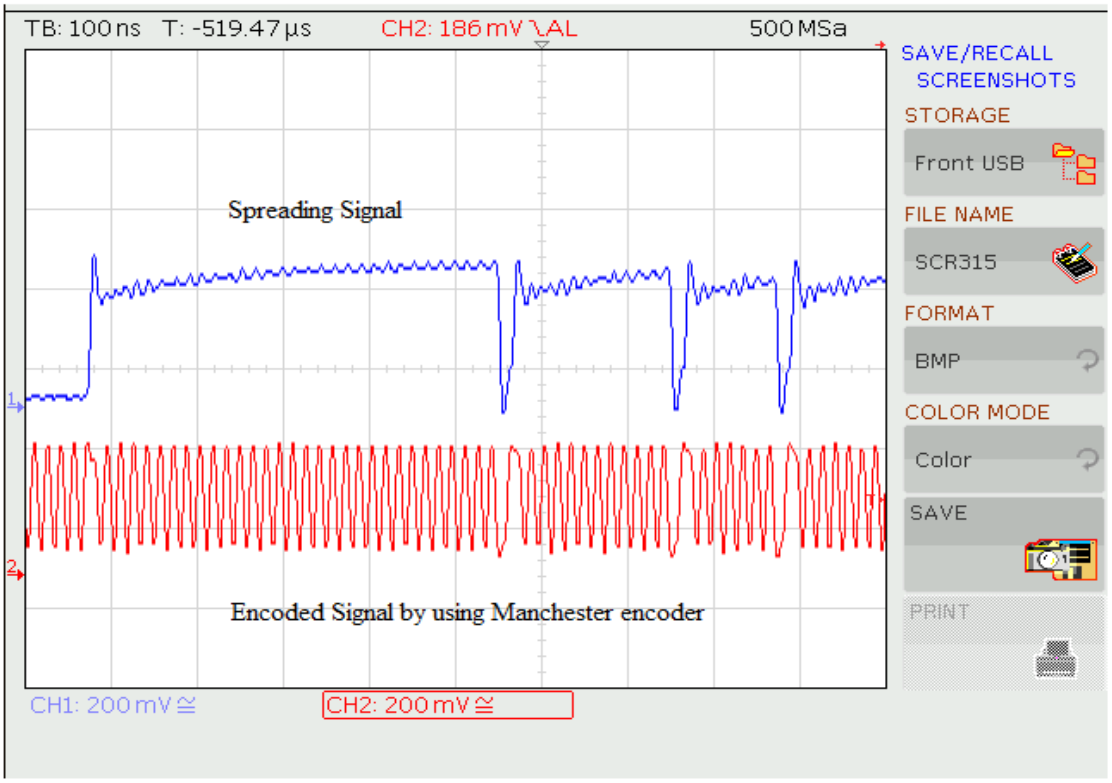

Fig. 6.25. The spreading signal is encoded by using Manchester encoder.

6.5.5 System clock rates

The main clock used in transmitter design is an oscillator socket single-ended (see section 5.3.6). A 64 MHz clock rate has been used for all the transmitter subsystems except the clock divider that runs on the clock rate of the 8 MHz. Fig. 6.26 shows the clock distribution of the transmitter design.

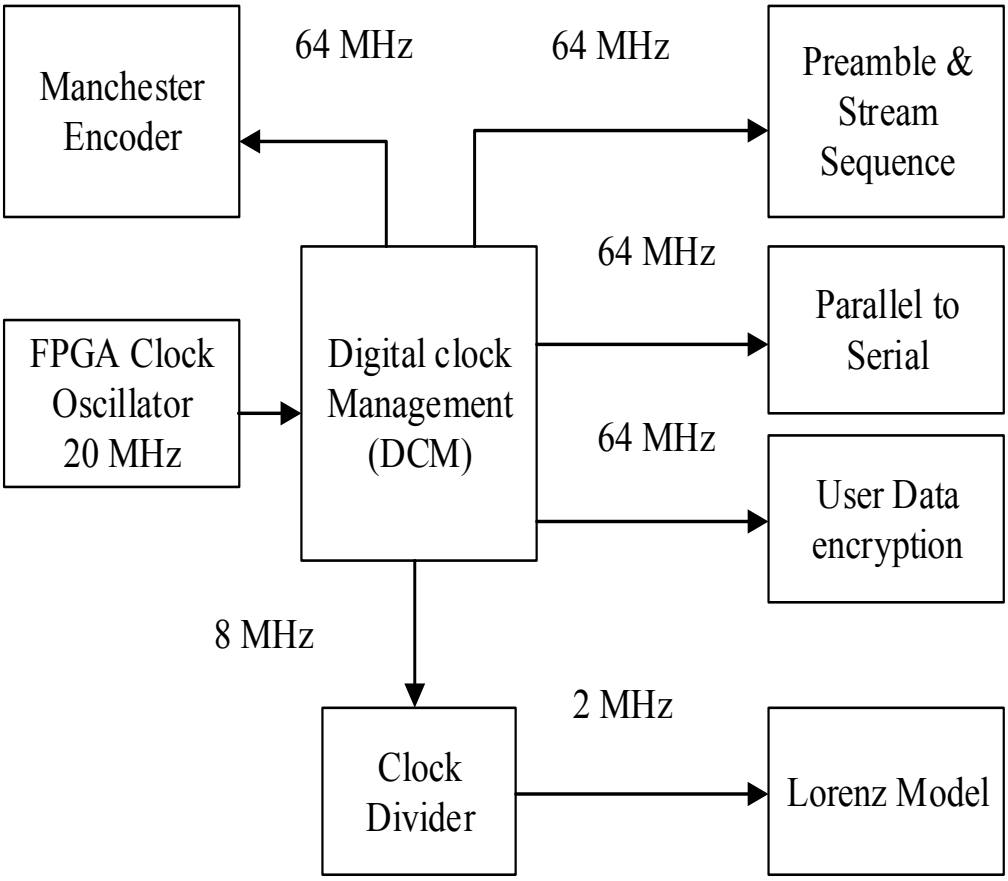

Fig. 6.26. Clock distributions of the transmitter.

6.6 Integrated Synthesis Environment (Xilinx ISE) of the transmitter

The transmitter design, which contains all files related to the project, is as follows, Digital clock management is provided by using IP (Core generator and Architecture Wizard), Clock Divider based on VHDL, Lorenz Model, Manchester decoder based on VHDL code, Receiver design based on System Generator® and parallel to serial, user data encryption model, preamble and Stream Sequence subsystem and Users Constraint File (UCF).

6.6.1 Device utilisation summary

Table 6. 2 shows the device utilisation summary of xc6slx45t-2fgg484. As a result, we can use smaller chip for the design.

Slice Logic Utilisation	Used	Available	Utilisation
Number of Slice Registers	3,843	54,576	7%
Number used for Flip Flops	463	-	-
Number used for AND/OR logics	3,380	-	-
Number of Slice LUTs	13,526	27,288	49%
Number used as Logic	13,435	27,288	49%
Number used as Memory	3	6,408	1%
Number used exclusively as route-thrus	88	-	-
Number of occupied slices	4,074	6,822	59%
Number of MUXCYs used	8,968	13,644	65%
Number of fully used LUT-FF pairs	3,744	13,531	27%
Number of BUFG/BUFGMUXs	4	16	25%
Number of DSP48A1s	0	58	0%
Number of PLL_ADVs	1	4	25%
Number of BUFIO2FB/BUFIO2FB_2CLKs	1	32	3%
Number of LOCed IOBs	5	8	62%

Table 6. 2. Device utilisation summary.

6.7 Implementation of the Receiver

The receiver is constructed to recover the transmitter clock first and then achieve synchronisation between two the chaotic signals. After that, the data are extracted by using the correlation between the received signal and a synchronized replica of the spreading chaotic sequence (see chapter 4).

Since the receiver design has multi-clock speed, the transmitter consists of four System Generator® designs, which are wrapped up later by using Xilinx® ISE. Fig. 6.27, Fig. 6.28, Fig. 6.29 and Fig. 6.30 show the System Generator® designs of the clock recovery, sync detector, the Lorenz generator and parallel to serial.

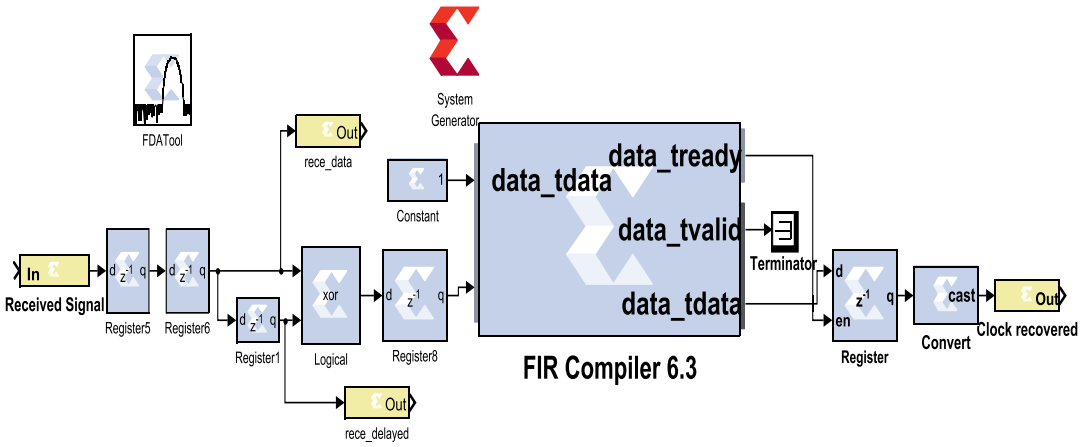

Fig. 6.27. Clock recovery, as viewed in the Xilinx System Generator®.

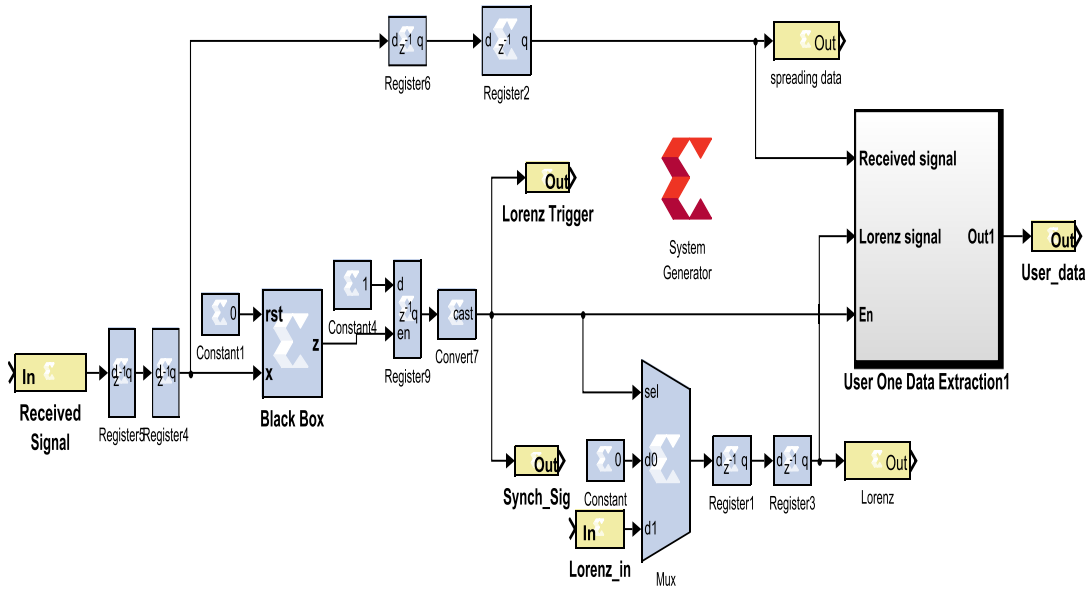

Fig. 6.28. Sync detector, as viewed in the Xilinx System Generator®.

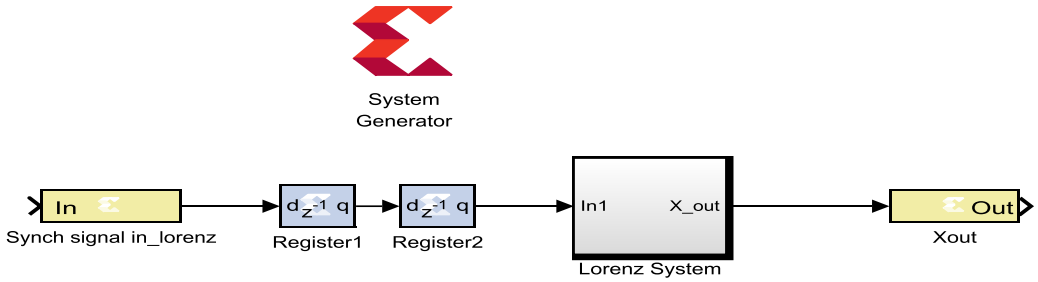

Fig. 6.29. Lorenz generator *x*-state, as viewed in the Xilinx System Generator®.

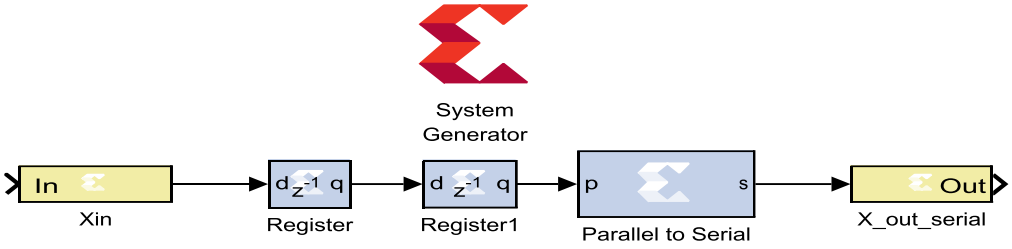

Fig. 6.30. Parallel-to-serial convertor, as viewed in the Xilinx System Generator®.

The receiver design contains the following subsystems, sync detector, which was discussed in chapter 4, clock Recovery, which was discussed in chapter 4, Manchester decoder and data recovery, which was discussed in chapter 4.

6.7.1 Clock recovery

The clock recovery was discussed in Chapter 4. The FIR design is described as follows:

- Sampling Frequency (Fs) = 380 MHz
- Fstop1 = 93 MHz
- Fpass1 = 127 MHz
- Fpass2 = 128 MHz
- Fstop2 = 163 MHz
- Attenuation on both sides of the passband = 118 dB
- Pass band ripple = 1.
- Filter's order is 38.

6.7.2 System clock rates

The main clock used in the receiver design is an oscillator socket single-ended LVCMOS at 20MHz (see section 5.3.6). A 380MHz has been used to feed the FIR filter while the rest of the subsystems run on the clock rate of 64MHz. The Lorenz model runs on the clock rate of 2MHz since a lower clock rate cannot be generated using PLL design. Fig. 6.31 shows the block diagram of the receiver system clock.

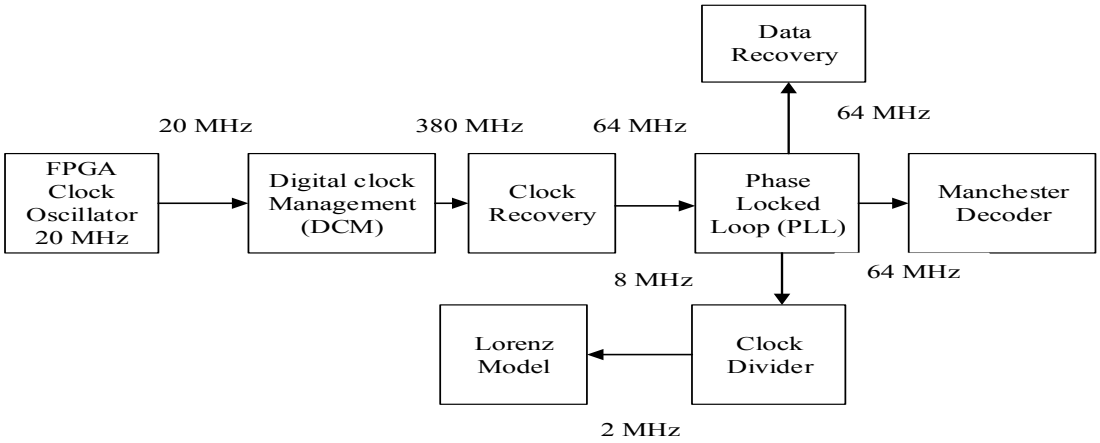

Fig. 6.31. System clock distributions of the receiver.

6.7.3 Manchester decoder

The Manchester decoder is established by using VHDL code, and the operation is performed by XOR of the data with clock. Fig. 6.32, Fig. 6.33 and Fig. 6.34 show the spreading signal have been decoded using the Manchester decoder.

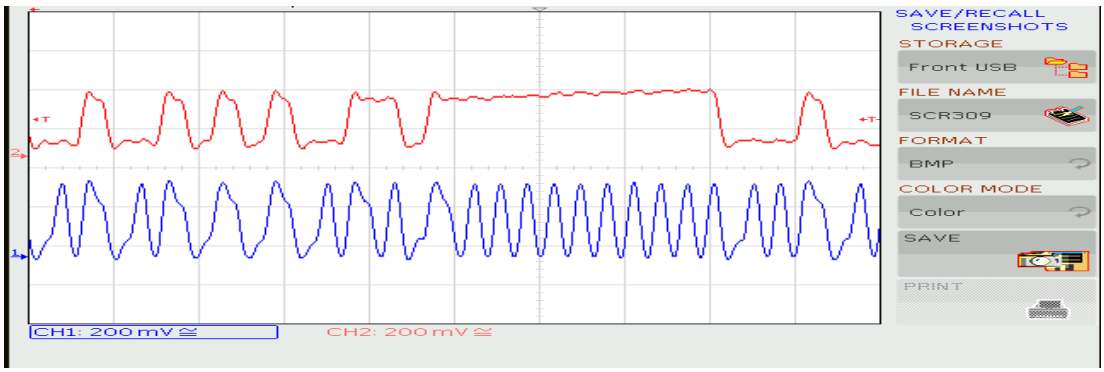

Fig. 6.32. Real-time spreading signal decoded using Manchester decoder, as visualized by the oscilloscope.

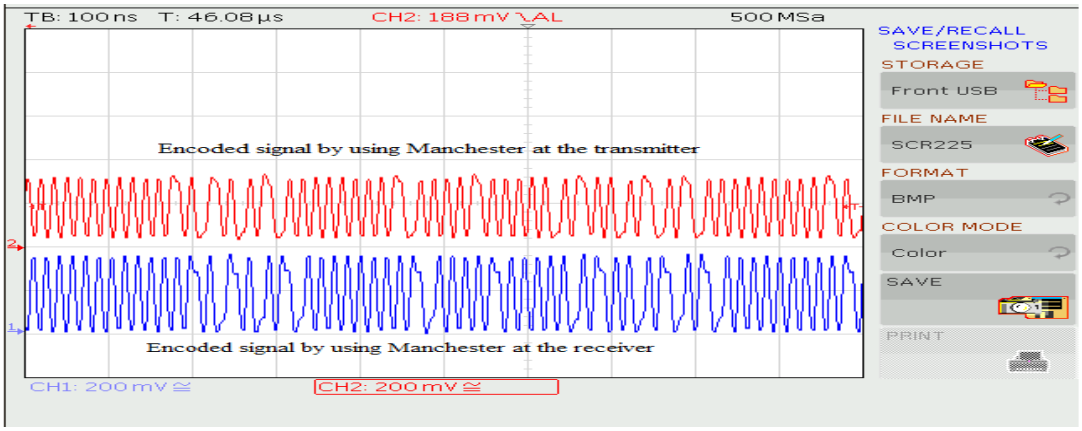

Fig. 6.33. Real-time encoded signal at the transmitter and receiver, as visualized by the oscilloscope.

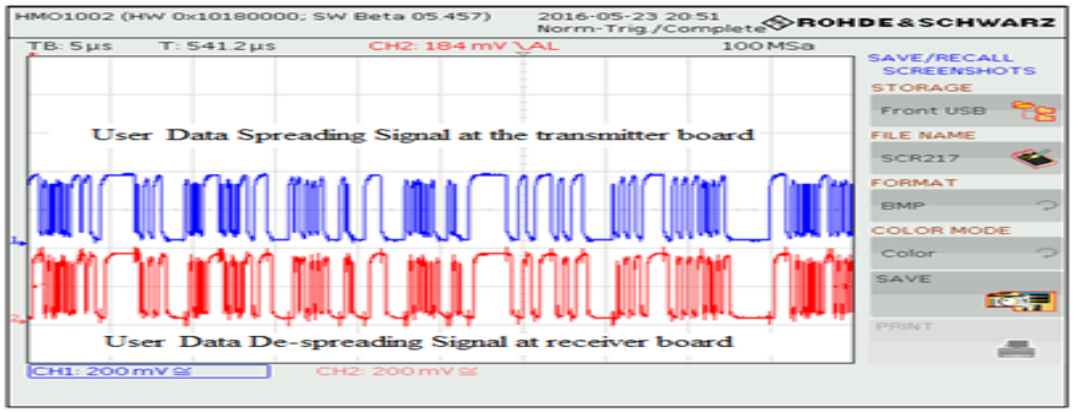

Fig. 6.34. Real-time test of the user data spreading at transmitter and de-spreading at the receiver, as visualized by the oscilloscope.

6.7.4 User data recovery results

The receiver is constructed to extract the data by using correlation between the received signal and a synchronized replica of spreading chaotic sequence. Two signals (received signal and replica of the transmitter binary stream at the receiver) are multiplied using cross-product block. Only one binary stream (x [n] or y[n]) is needed to extract the user data. The result of multiplication is then accumulated every 32 samples. After that, the register is used to show the last value of the accumulation process which is every 32 bit cycle where the enable signal of the register is synchronous with the received signal. This value is compared with a threshold value using the relational block (≥) to retrieve the user data. Fig. 6.35 shows the block diagram of the decryption process. Fig. 6.36 shows the user data recovery process.

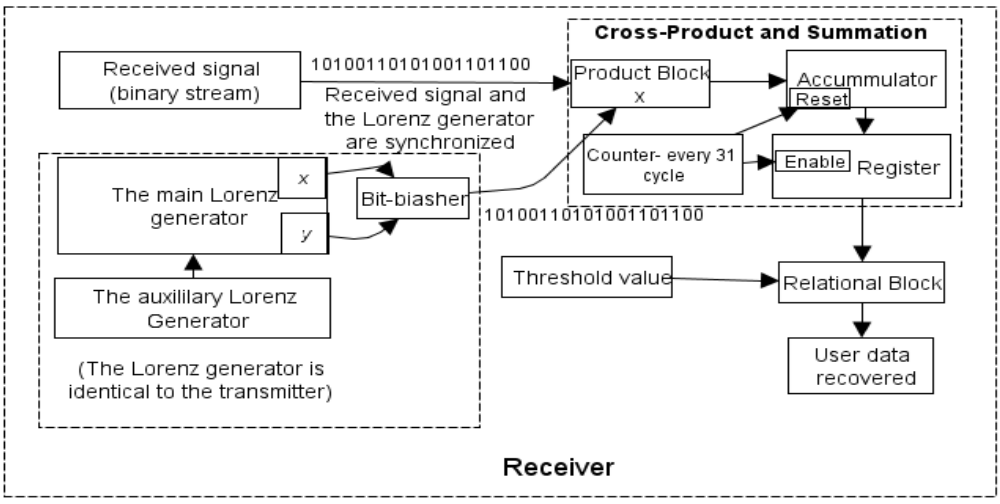

Fig. 6.35. Decryption process.

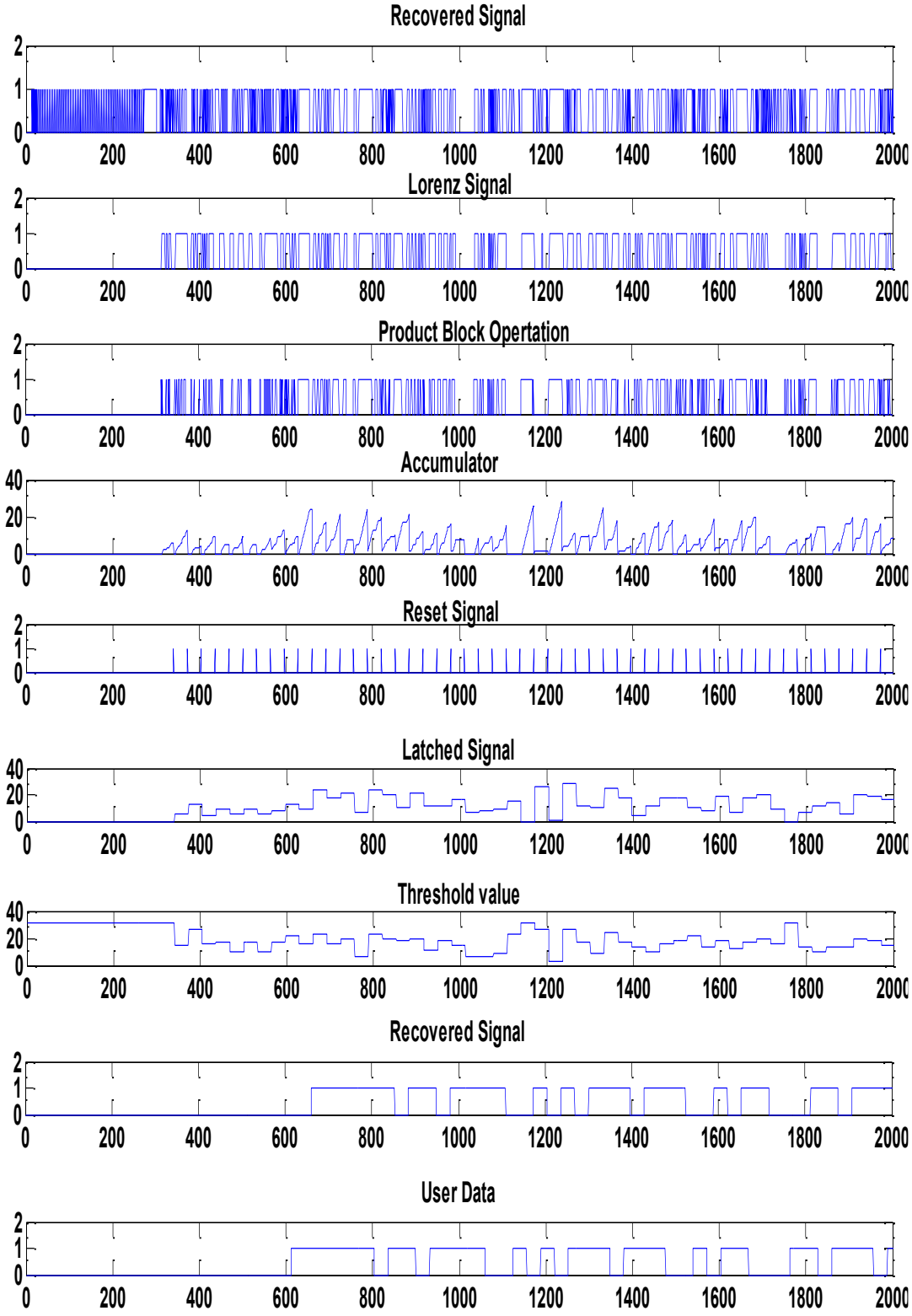

Fig. 6.36. Data recovery process.

The receiver system recovers the user data. Fig. 6.37 and Fig. 6.38 show the simulation test and real-time test of the data recovery of the user data.

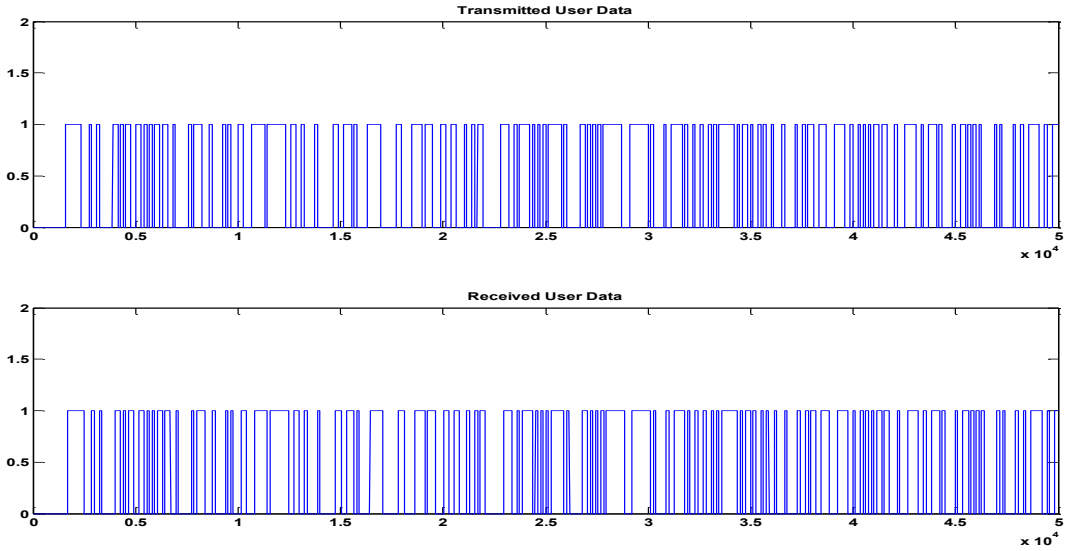

Fig. 6.37. Transmitted and Recovered User data.

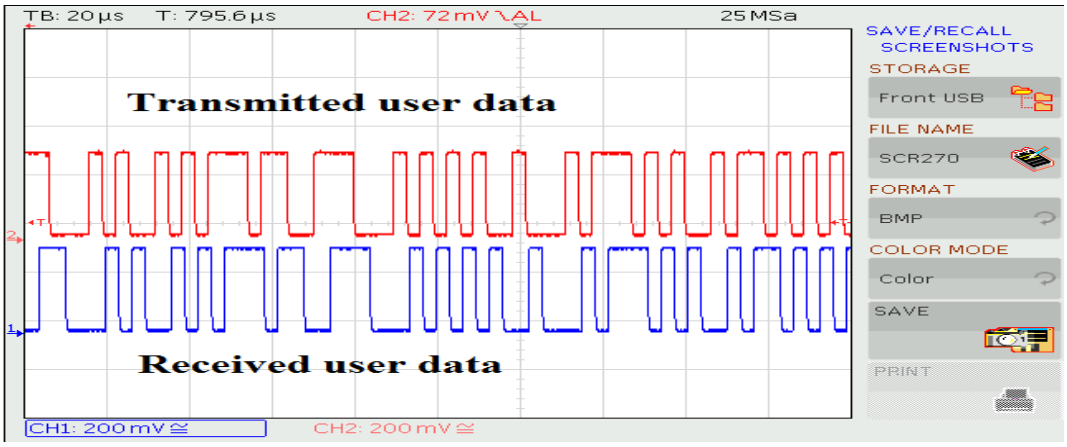

Fig. 6.38. User data recovery.

6.8 Integrated Synthesis Environment (Xilinx ISE) of the receiver

The ISE Project navigator is a high-level manger that allows organisation of the transmitter design, which contains all files related to the project as follows, digital clock management is provided by using IP (Core generator and Architecture Wizard), Phase Locked Loop (PLL), Manchester decoder based on VHDL code, clock recovery, data recovery, clock divider, Parallel to serial, Lorenz Model and Users Constraint File (UCF).

Fig. 6.39 shows a register transfer level (RTL) graphical representation of the transmitter design.

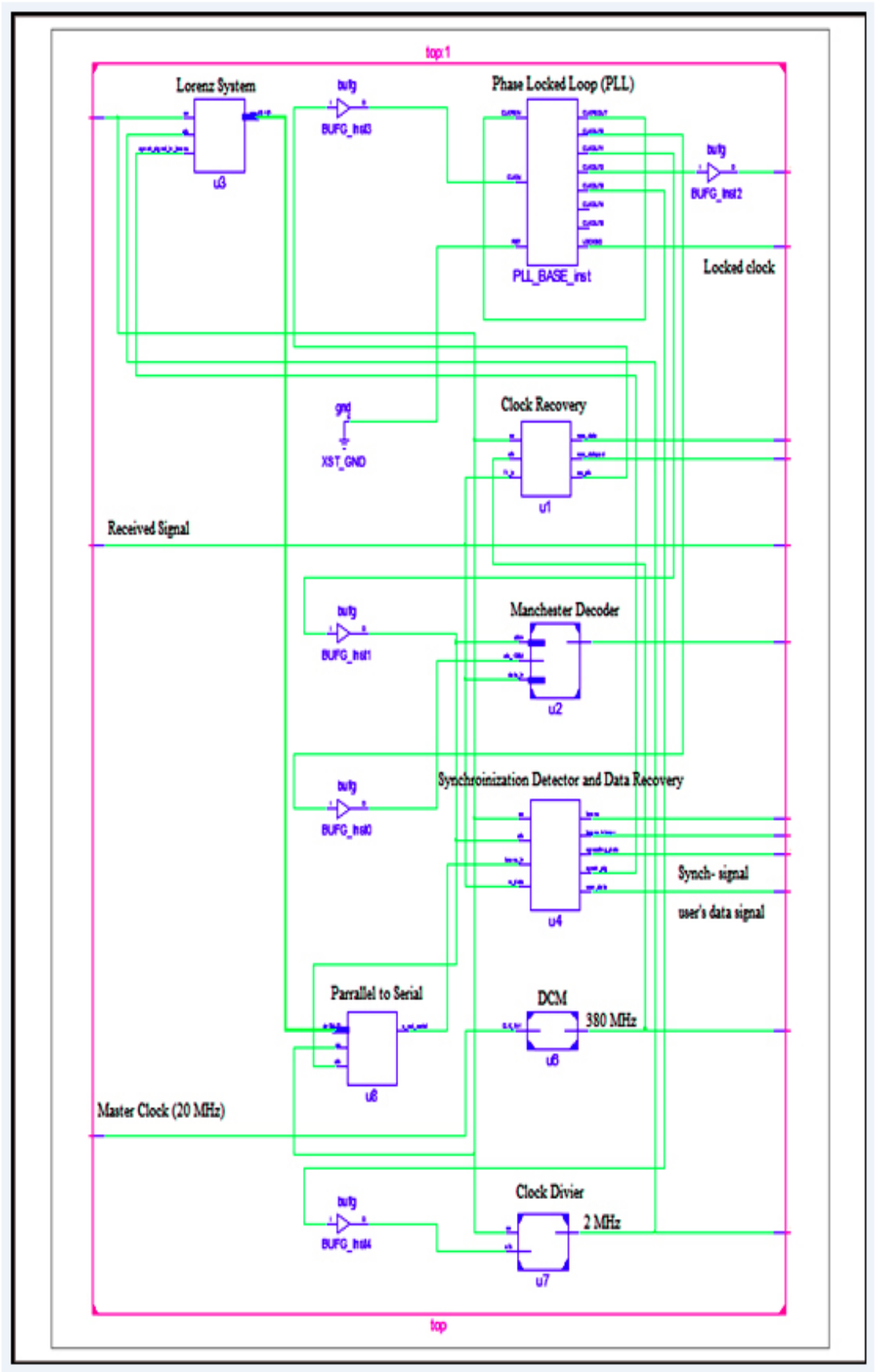

Fig. 6.39. RTL of the receiver design.

6.8.1 Device utilisation summary

Table 6.3 shows that the device utilisation summary of the xc6slx45t-2fgg484. As a result, we can use smaller chip for the design.

Slice Logic Utilisation	*Used*	*Available*	*Utilisation*
Number of Slice Registers	5,003	54,576	9%
Number used for Flip Flops	674	-	-
Number used for AND/OR logics	3,380		
Number of Slice LUTs	13,705	27,288	50%
Number used as logic	13,464	27,288	-
Number used exclusively as route-thrus	101	-	-
Number of occupied slices	4,364	6,822	63%
Number of MUXCYs used	9,756	13,644	71%
Number of fully used LUT-FF pairs	3,842	13,649	32%
Number of BUFG/BUFGMUXs	8	16	50%
Number of DSP48A1s	20	58	34%
Number of PLL_ADVs	2	4	50%
Number of BUFIO2FB/BUFIO2FB_2CLKs	1	32	3%
Number of LOCed IOBs	4	14	42%
Number of BUF/BUFGMUXs	8	16	50%

Table 6.3. The device utilisation summary.

6.9 Cryptanalysis of the Stream cipher

6.9.1 Randomness test of the Lorenz stream cipher

In this experiment, 100 binary sequences with a size of 1,000, 000 bits each are generated by the Lorenz generator. It is clear from the test results that the chaotic generator has successfully passed the 13 statistical randomness tests. The results are shown in Tables 4. The P-value is explained in chapter 3, section 3.1.5.

Statistical Test	**Status**	**P-value**
Frequency	Pass	0.911413
Block Frequency	Pass	0.911413

CUSUM-Forward	Pass	0.739918
CUSUM-Reverse	Pass	0.739918
Runs	Pass	0.350485
Long Runs of Ones	Pass	0.008879
Rank	Pass	0.534146
FFT Test	Pass	0.350485
Non-overlapping	Pass	0.534146
Overlapping	Pass	0.911413
Approximate Entropy	Pass	0.739918
Linear Complexity	Pass	0.534146
Serial	Pass	0.122325

Table 4: the device utilisation summary.

6.9.2 Sensitivity of mismatched key.

We tested the sensitivity of the system to the accuracy of the key. Each of the multiplier parameters (A, B and C) and initial conditions is in turn changed by a factor of 10^{-9} in the transmitter to test the sensitivity of the system. In each case the data could not be retrieved in the receiver. No signal was recovered and the results are shown in Fig. 6.40. Fig. 6.41 shows how the chaotic spreading sequence in the receiver started to vary from that of the transmitter. This is the cause of the error in the received signal.

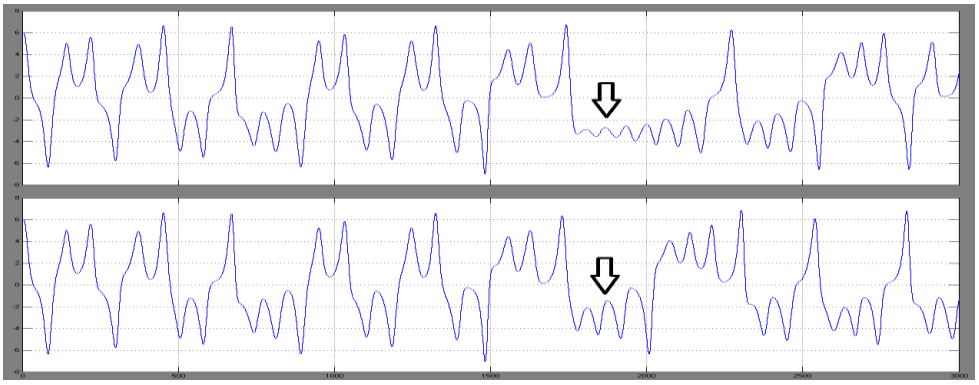

Fig. 6.40. The two Lorenz Generators are out of synch after the time indicated by the arrows.

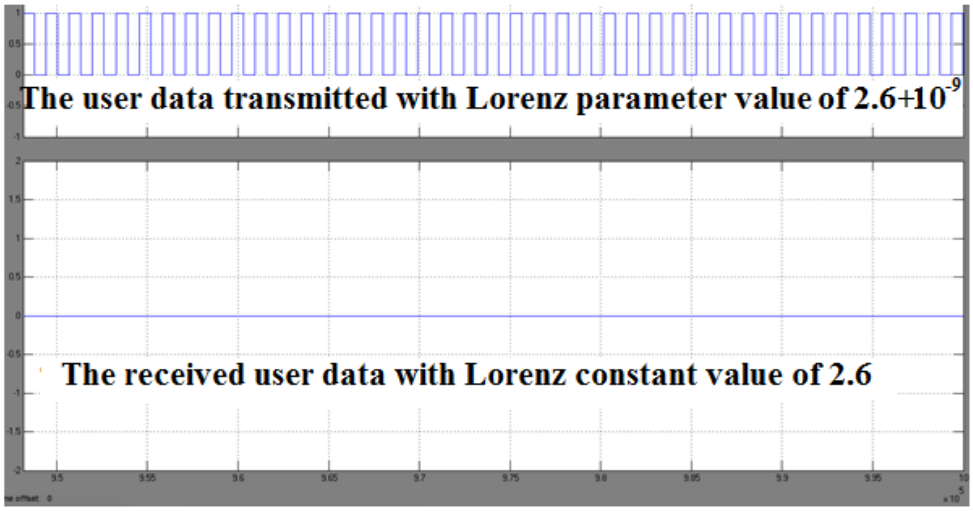

Fig. 6.41. User data recovery when two Lorenz parameters are mismatched.

6.10 Conclusion

This chapter presents a digital communication system with high security based on Lorenz stream cipher. The system is implemented using two separate Spartan 6 FPGA boards. User data encryption method using two Lorenz chaotic systems, in which the encryption key various continuously, is working perfectly. User data threshold tracking subsystem is able to track the proper threshold value in order to retrieve the user data. User data transmitted is recovered perfectly at the receiver. The Lorenz stream bits have passed the randomness test. Each of the multiplier parameters (A, B and C) and initial conditions is in turn changed by a factor of 10^{-9} in the transmitter to test the sensitivity of the system. In each case the data could not be retrieved in the receiver. The system has a high degree of security compared to other communication systems in which the total key length is $2^{(12*32)} = 2^{384}$. The data rate achieved is 2 Mbps.

In future works, the system performance will be tested in the presence of noise. Further, the data rate can be increased by using digital modulation. Optical fibres may also be used instead of coaxial cables.

Chapter 7

Conclusion and future work

In this dissertation, the conclusion from the research is presented as shown below:

We described a digital communication system with high immunity and security based on the digitization of a Lorenz chaotic stream cipher. This technique was built on digitizing two Lorenz chaotic models to increase the security level. Spread-spectrum technology was used for user data spreading. A new cryptosystem approach based on Lorenz chaotic systems is presented for secure data transmission. The system uses a stream cipher, in which the encryption key varies continuously. Furthermore one or more of the parameters of the Lorenz generator is controlled by an auxiliary chaotic generator for increased security. The data encryption uses a symmetric cipher which a key length of 576-bits. This is a key space of the system is 2^{576}. The scrambling scheme was developed and Lorenz stream cipher binary stream passed the NIST randomness test successfully. In addition, the system output signal has a high sensitivity to small changes in any parameter. Moreover, the auto-correlation and cross correlation for 32-bits have good results. The maximum auto-correlation and cross-correlation functions for (32, 64, 128 and 256-bits) have shown good results. The auto-correlation of 32-bits was 32 and the cross-correlation value was 8. Moreover, when word length is longer, the auto-correlation and cross correlation are improved. In this chapter, the communication system was designed and tested for 32-bits. The aim of choosing the 32-bit was to reduce hardware resources consumed and to increase the data rate. The security level and noise performance with a word length of 32 are both very good. Even high security level and noise immunity could be obtained by increasing the word length. On the other hand, we have compared a cross-correlation of one Lorenz generator in terms of *x* and *y*, *x* and *z* and *y* and *z* and with two Lorenz generators with different parameters in terms *x* and *y*, *x* and *z, y* and *z*, we found that both of them are the same and result in low correlation.

Three different methods have been developed to extract the data at the receiver. De-spreading based on cross-correlation, on cross product and summation and de-spreading on dot product and summation. All three methods have shown the ability to extract the user data transmitted successfully. However, we used product and

summation method because it is easily converted to Xilinx ® System Generator blocks. In contrast, de-spreading based on cross correlation method is hard to be converted to Xilinx System Generator® because some SIMULINK blocks were not yet available.

The communication system was designed, and we obtained simulation results, as well as performance results and the bit error rate (BER) of the system on a noisy channel. The performance results were evaluated in terms of Signal to Noise Ratio (SNR) of the CDMA system for four users. Results have been evaluated and compared to standard accepted BER of 10^{-6}. The system performance were shown a good results. At -2.974 signal to noise ratio, the system achieved no bit error with $1e^{6}$ bits transmitted for four users. The system has achieved a good performance based on the results obtained and compared to other communication systems.

The study of the communication system security and the accompanying results are provided. Our cryptosystem has been compared with existing symmetric cryptography systems such as DES, AES, blowfish and One-Time Pad in term of the key length and key space. Security analysis shows the system to have a high degree of security compared to other communication systems. Our approach is to provide a cryptosystem that can be compared to a One-Time-Pad.

The clock-recovery technique was tested with the Simulink tool and validated in real-time hardware tests with FPGA boards. The implementation shows that the desired frequency was recovered and locked. Hardware results are described. Data recovery was also tested in both simulation and real-time implementation, the results of which are described.

We discuss and detail a practical system for synchronising two chaotic generators used in the digital Code Division Multiple Access (CDMA) method. The simulated results of the clock recovery technique are obtained. The clock recovery SIMULINK blocks are then converted to System Generator® blocks. The simulated test shows that the clock is recovered well from the random data generator (Bernoulli generator). The clock recovery is then implemented in real time by using two separate Spartan 6 FPGA boards. The clock rate of 65 MHz is recovered well at the receiver. The PLL is locked the desired frequency at clock rate of 65 MHz. The data recovery based on a System Generator® is presented. The simulated results are obtained. We have described a practical system for synchronising two chaotic generators used in a digital CDMA.

The techniques are based on a sync stream that is transmitted to the receiver in order to trigger the Lorenz chaotic generator in the same time step. The chaotic synchronisation is well maintained at the receiver. Both the receiver and transmitter are implemented using two separate Spartan 6 FPGA.

Our digital communication system with high immunity and high security was implemented using Spartan 6 boards. A detailed explanation of the implementation method, including the sequential process of the transmitter subsystem, clock-recovery subsystem, user data generator, user data spreading and Manchester encoder are presented in Chapter 6. In addition, a Lorenz chaotic signal was implemented and chaotic signal attractors were visualized in an oscilloscope, employing the Spartan 6 boards and a Pmod4 SPI protocol. The randomness test results of the Lorenz chaotic model output were obtained with the NIST statistical test suite.

We also obtained hardware implementation results of the block-spreading communication system for four users via Spartan 6 FPGA boards. These results, as well as a detailed explanation of the implementation method of the transmitter, which included the sequential process of these subsystems: preamble, clock-recovery, user data generator, user data spreading and Manchester encoder are presented. The aim of the implementing a digital communication system is to develop reliable method of synchronisation and data recovery. Codes that have been used in this work are extracted from the Lorenz generators. Each user code length is 32-bits. The method that has been used to retrieve the user data transmitted is based on cross-product and summation. The transmitter design has consumed only 1% of the slice registers and Look Up Table (LUT). Thus, it can be implemented in a smaller chip. On the other hand, the receiver design has consumed 1% of the slice registers and LUT which means that the system can be implemented in a smaller chip. The data transmitted for all four users are recovered perfectly at the receiver FPGA board. The communication system achieved a data rate of 2 Mbps.

A digital communication system with high security based on Lorenz stream cipher. The system is implemented using two separate Spartan 6 FPGA boards. User data encryption method using two Lorenz chaotic systems, in which the encryption key various continuously, is working perfectly. User data threshold tracking subsystem is able to track the proper threshold value in order to retrieve the user data. User data transmitted is recovered perfectly at the receiver. The Lorenz stream bits have passed

the randomness test. Each of the multiplier parameters (A, B and C) and initial conditions is in turn changed by a factor of 10^{-9} in the transmitter to test the sensitivity of the system. In each case the data could not be retrieved in the receiver. The system has a high degree of security compared to other communication systems in which the total key length is $2^{(12*32)} = 2^{384}$. The data rate achieved is 2 Mbps.

For the Future work, we have the following parts need further research.

In the real time implementation of the Lorenz Generator implemented for a clock rate of 2MHz without in time violence issue. However, the Lorenz Generator need further optimization to be synchronise and to avoid any issue regarding the time violence when the clock rate increased above 2MHz.

The advantage of a synchronisation method is that there is no need to inject a signal into the dynamics of the slave system, which affects the channel efficiency. In addition, this method is not affected by a high noise environment. However, the disadvantage of this method is that when the synchronisation signal is affected, the synchronisation between the two systems will be lost. Thus, this synchronisation method needs to be developed to address the case of a synchronisation bit stream that is affected by noise.

The system performance will be tested in the presence of noise. Further, the data rate can be increased by using digital modulation. Optical fiber may also be used instead of coaxial cables.

Reference

[1] S. Kumar, C. Paar, J. Pelzl, G. Pfeiffer, and M. Schimmler, "Breaking ciphers with COPACOBANA–a cost-optimized parallel code breaker," in International Workshop on Cryptographic Hardware and Embedded Systems, pp. 101-118, 2006

[2] R. Clayton and M. Bond, "Experience using a low-cost FPGA design to crack DES keys," in International Workshop on Cryptographic Hardware and Embedded Systems, pp. 579-592, 2002.

[3] R. Kharel, "Design and Implementation of secure chaotic communication system," PhD thesis, March 2011.

[4] H. Williams, "A modification of the RSA public-key encryption procedure (Corresp.)," IEEE Transactions on Information Theory, vol. 26, pp. 726-729, 1980.

[5] T. Kean and A. Duncan, "DES key breaking, encryption and decryption on the XC6216," in FPGAs for Custom Computing Machines, 1998. Proceedings. IEEE Symposium on, pp. 310-311, 1998.

[6] M. Itoh, "Spread spectrum communication via chaos," International Journal of Bifurcation and Chaos, vol. 9, pp. 155-213, 1999.

[7] G. Heidari-Bateni, C. McGillem, and M. Tenorio, "A novel multiple-address digital communication system using chaotic signals," in Communications, 1992. ICC'92, Conference record, SUPERCOMM/ICC'92, Discovering a New World of Communications., IEEE International Conference on, pp. 1232-1236, 1992.

[8] B. Fan and L.-R. Tang, "A new five-dimensional hyperchaotic system and its application in DS-CDMA," in Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on, pp. 2069-2073, 2012,

[9] G. Heidari-Bateni and C. McGillem, "Chaotic sequences for spread spectrum: An alternative to PN-sequences," in Wireless Communications, 1992. Conference Proceedings., 1992 IEEE International Conference on Selected Topics in, pp. 437-440, 1992

[10] A. P. Kurian, S. Puthusserypady, and S. M. Htut, "Performance enhancement of DS/CDMA system using chaotic complex spreading sequence," IEEE Transactions on wireless communications, vol. 4, pp. 984-989, 2005.

[11] G. Jakimoski and L. Kocarev, "Chaos and cryptography: block encryption ciphers based on chaotic maps," IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 48, pp. 163-169, 2001.

[12] J. Yu and Y.-D. Yao, "Detection performance of chaotic spreading LPI waveforms," IEEE Transactions on wireless communications, vol. 4, pp. 390-396, 2005.

[13] L. Kong, G. Kaddoum, and M. Taha, "Performance analysis of physical layer security of chaos-based modulation schemes," in Wireless and Mobile Computing, Networking and Communications (WiMob), 2015 IEEE 11th International Conference on, pp. 283-288, 2015.

[14] A. S. Mansingka, M. A. Zidan, A. Radwan, and K. Salama, "Secure Ds-CDMA spreading codes using fully digital multidimensional multiscroll chaos," in Circuits and Systems (MWSCAS), 2013 IEEE 56th International Midwest Symposium on, pp. 1334-1338, 2013.

[15] F. Agnelli, G. Mazzini, R. Rovatti, and G. Setti, "A first experimental verification of optimal MAI reduction in chaos-based DS-CDMA systems," in Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on, pp. 137-140, 2001.
[16] C. Paar and J. Pelzl, Understanding cryptography: a textbook for students and practitioners: Springer Science & Business Media, 2009.
[17] S. H. Strogatz, Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering: Westview press, 2014.
[18] J. Y. Stein, Digital signal processing: a computer science perspective: John Wiley & Sons, Inc., 2000.
[19] A. E. R. Shehata, "Secure Computer Communications and Databases Using Chaotic Encryption Systems," Doctoral thesis, University of Kent, 2002.
[20] G. Heidari-Bateni and C. D. McGillem, "A chaotic direct-sequence spread-spectrum communication system," IEEE Transactions on communications, vol. 42, pp. 1524-1527, 1994.
[21] G. Kaddoum, F.-D. Richardson, and F. Gagnon, "Design and analysis of a multi-carrier differential chaos shift keying communication system," IEEE Transactions on communications, vol. 61, pp. 3281-3291, 2013.
[22] M. Kennedy, R. Rovatti, and G. Setti, Chaotic electronics in telecommunications: CRC press, 2000.
[23] G. Mazzini, G. Setti, and R. Rovatti, "Chaotic complex spreading sequences for asynchronous DS-CDMA. I. System modeling and results," IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 44, pp. 937-947, 1997.
[24] M. Sobhy and A. Shehata, "Chaotic radar systems," in Microwave Symposium Digest. 2000 IEEE MTT-S International, pp. 1701-1704, 2000.
[25] M. I. Sobhy, M. A. Aseeri, and A. E. Shehata, "Real time implementation of continuous (Chua and Lorenz) chaotic generator models using digital hardware," in Proc. of the Third International Symposium on Communication Systems Networks and Digital Processing, pp. 38-41, 1999.
[26] M. I. Sobhy and A.-E. Shehata, "Chaotic algorithms for data encryption," in Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on, pp. 997-1000, 2001.
[27] G. Kaddoum, "Wireless chaos-based communication systems: A comprehensive survey," IEEE Access, vol. 4, pp. 2621-2648, 2016.
[28] H. Dedieu, M. P. Kennedy, and M. Hasler, "Chaos shift keying: Modulation and demodulation of a chaotic carrier using self-synchronizing Chua's circuits," IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 40, pp. 634-642, 1993.
[29] U. Parlitz, L. O. Chua, L. Kocarev, K. Halle, and A. Shang, "Transmission of digital signals by chaotic synchronization," International Journal of Bifurcation and Chaos, vol. 2, pp. 973-977, 1992.
[30] G. Kolumbán, M. P. Kennedy, and L. O. Chua, "The role of synchronization in digital communications using chaos. I. Fundamentals of digital communications," IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 44, pp. 927-936, 1997.
[31] C. Tse and F. Lau, "Chaos-based digital communication systems," Operating Principles, Analysis Methods and Performance Evaluation (Springer Verlag, Berlin, 2004), 2003.

[32] M. P. Kennedy, G. Kolumbán, G. Kis, and Z. Jákó, "Performance evaluation of FM-DCSK modulation in multipath environments," IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 47, pp. 1702-1711, 2000.
[33] C. Shannon, "A mathematical theory of communication, bell System technical Journal 27: 379-423 and 623–656," Mathematical Reviews (MathSciNet): MR10, 133e, 1948.
[34] C. E. Shannon, "Communication theory of secrecy systems," Bell Labs Technical Journal, vol. 28, pp. 656-715, 1949.
[35] C. E. Shannon, "Communication in the presence of noise," Proceedings of the IRE, vol. 37, pp. 10-21, 1949.
[36] L. Chua, "Dynamic nonlinear networks: State-of-the-art," IEEE Transactions on Circuits and Systems, vol. 27, pp. 1059-1087, 1980.
[37] A. Pande and J. Zambreno, "A chaotic encryption scheme for real-time embedded systems: design and implementation," Telecommunication Systems, pp. 1-11, 2013.
[38] R. Vali, S. Berber, and S. K. Nguang, "Accurate derivation of chaos-based acquisition performance in a fading channel," IEEE Transactions on wireless communications, vol. 11, pp. 722-731, 2012.
[39] R. Vali, S. M. Berber, and S. K. Nguang, "Analysis of chaos-based code tracking using chaotic correlation statistics," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 59, pp. 796-805, 2012.
[40] S. Berber and S. Feng, "Chaos-based physical layer design for WSN applications," in 17th WSEAS Int. Conf. on Communications, Rhodes, Greece, pp. 157-162, 2013.
[41] G. Mazzini, R. Rovatti, and G. Setti, "Interference minimisation by autocorrelation shaping in asynchronous DS-CDMA systems: chaos-based spreading is nearly optimal," Electronics Letters, vol. 35, pp. 1054-1055, 1999.
[42] L. Gong and L. Shaoqian, "Chaotic spreading sequences with multiple access performance better than random sequences," IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 47, pp. 394-397, 2000.
[43] T. Yang and L. O. Chua, "Chaotic digital code-division multiple access (CDMA) communication systems," International Journal of Bifurcation and Chaos, vol. 7, pp. 2789-2805, 1997.
[44] V. Lynnyk and S. Čelikovský, "On the anti–synchronization detection for the generalized Lorenz system and its applications to secure encryption," Kybernetika, vol. 46, pp. 1-18, 2010.
[45] Y. Xia, C. Tse, and F. C.-M. Lau, "Performance of differential chaos-shift-keying digital communication systems over a multipath fading channel with delay spread," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 51, pp. 680-684, 2004.
[46] G. Mazzini, R. Rovatti, and G. Setti, "Sequence synchronization in chaos-based DS-CDMA systems," in Circuits and Systems, 1998. ISCAS'98. Proceedings of the 1998 IEEE International Symposium on, pp. 485-488, 1998
[47] B. Jovic, C. Unsworth, G. S. Sandhu, and S. M. Berber, "A robust sequence synchronization unit for multi-user DS-CDMA chaos-based communication systems," Signal Processing, vol. 87, pp. 1692-1708, 2007.

[48] G. Kaddoum, D. Roviras, P. Chargé, and D. Fournier-Prunaret, "Robust synchronization for asynchronous multi-user chaos-based DS-CDMA," Signal Processing, vol. 89, pp. 807-818, 2009.
[49] T. Yang, "A survey of chaotic secure communication systems," International Journal of Computational Cognition, vol. 2, pp. 81-130, 2004.
[50] K. M. Cuomo and A. V. Oppenheim, "Circuit implementation of synchronized chaos with applications to communications," Physical review letters, vol. 71, p. 65, 1993.
[51] J.-c. Feng and K. T. Chi, "On-line adaptive chaotic demodulator based on radial-basis-function neural networks," Physical Review E, vol. 63, p. 026202, 2001.
[52] T. Yang and L. O. Chua, "Secure communication via chaotic parameter modulation," IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 43, pp. 817-819, 1996.
[53] S. Sadoudi, M. S. Azzaz, and C. Tanougast, "Novel experimental synchronization technique for embedded chaotic communications," in Control, Decision and Information Technologies (CoDIT), 2014 International Conference on,pp. 669-672, 2014.
[54] F. Dachselt and W. Schwarz, "Chaos and cryptography," IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 48, pp. 1498-1509, 2001.
[55] M. S. Azzaz, C. Tanougast, S. Sadoudi, A. Bouridane, and A. Dandache, "An FPGA implementation of a Feed-Back Chaotic Synchronization for secure communications," in Communication Systems Networks and Digital Signal Processing (CSNDSP), 2010 7th International Symposium on, pp. 239-243, 2010.
[56] S. Sadoudi, C. Tanougast, and M. S. Azzaz, "First experimental solution for channel noise sensibility in digital chaotic communications," Progress In Electromagnetics Research C, vol. 32, pp. 181-196, 2012.
[57] Z. Ying-Qian and W. Xing-Yuan, "A parameter modulation chaotic secure communication scheme with channel noises," Chinese Physics Letters, vol. 28, p. 020505, 2011.
[58] L. Shujun, M. Xuanqin, and C. Yuanlong, "Pseudo-random bit generator based on couple chaotic systems and its applications in stream-cipher cryptography," Progress in Cryptology—INDOCRYPT 2001, pp. 316-329, 2001.
[59] O.-C. Yue, "Spread spectrum mobile radio, 1977-1982," IEEE Transactions on Vehicular Technology, vol. 32, pp. 98-105, 1983.
[60] B. Goldberg, "Applications of statistical communications theory," IEEE Communications Magazine, vol. 19, pp. 26-33, 1981.
[61] R. Scholtz, "The origins of spread-spectrum communications," IEEE Transactions on communications, vol. 30, pp. 822-854, 1982.
[62] M. A. Abu-Rgheff, Introduction to CDMA wireless communications: Academic Press, 2007.
[63] R. L. Peterson, R. E. Ziemer, and D. E. Borth, Introduction to spread-spectrum communications vol. 995: Prentice Hall New Jersey, 1995.
[64] P. I. Martoyo, A. Susanto, E. Wijanto, H. Kanalebe, and K. Gandi, "Chaos codes vs. orthogonal codes for CDMA," in Spread Spectrum Techniques and Applications (ISITA), 2010 IEEE 11th International Symposium on, pp. 189-193, 2010.

[65] R. C. Hilborn, Chaos and nonlinear dynamics: an introduction for scientists and engineers: Oxford University Press on Demand, 2000.
[66] G. Kaddoum, M. Coulon, D. Roviras, and P. Chargé, "Theoretical performance for asynchronous multi-user chaos-based communication systems on fading channels," Signal Processing, vol. 90, pp. 2923-2933, 2010.
[67] W. M. Tam, F. C.-M. Lau, and C. Tse, "A multiple access scheme for chaos-based digital communication systems utilizing transmitted reference," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 51, pp. 1868-1878, 2004.
[68] N. X. Quyen, V. Van Yem, and T. Q. Duong, "Design and analysis of a spread-spectrum communication system with chaos-based variation of both phase-coded carrier and spreading factor," IET Communications, vol. 9, pp. 1466-1473, 2015.
[69] A. Abel and W. Schwarz, "Chaos communications-principles, schemes, and system analysis," Proceedings of the IEEE, vol. 90, pp. 691-710, 2002.
[70] G. Setti, R. Rovatti, and G. Mazzini, "Synchronization mechanism and optimization of spreading sequences in chaos-based DS-CDMA systems," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. 82, pp. 1737-1746, 1999.
[71] R. Rovatti, G. Setti, and G. Mazzini, "Chaotic complex spreading sequences for asynchronous DS-CDMA. Part II. Some theoretical performance bounds," IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 45, pp. 496-506, 1998.
[72] D. Leon, S. Balkır, M. Hoffman, and L. Perez, "Pseudo-chaotic PN-sequence generator circuits for spread spectrum communications," IEE Proceedings-Circuits, Devices and Systems, vol. 151, pp. 543-550, 2004.
[73] N. Rahnama and S. Talebi, "Performance comparison of chaotic spreading sequences generated by two different classes of chaotic systems in a chaos-based direct sequencecode division multiple access system," IET Communications, vol. 7, pp. 1024-1031, 2013.
[74] R. Rovatti, G. Setti, and G. Mazzini, "Chaos-based spreading compared to m-sequences and Gold spreading in asynchronous CDMA communication systems," in Proc. European Conference on Circuit Theory and Design, 1997.
[75] G. Cimatti, R. Rovatti, and G. Setti, "Chaos-based spreading in DS-UWB sensor networks increases available bit rate," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 54, pp. 1327-1339, 2007.
[76] T. Kohda, A. Tsuneda, and A. J. Lawrance, "Correlational properties of Chebyshev chaotic sequences," Journal of time series analysis, vol. 21, pp. 181-191, 2000.
[77] T. Kohda and A. Tsuneda, "Pseudonoise sequences by chaotic nonlinear maps and their correlation properties," IEICE Transactions on Communications, vol. 76, pp. 855-862, 1993.
[78] H. Nejati, A. Beirami, and W. H. Ali, "Discrete-time chaotic-map truly random number generators: design, implementation, and variability analysis of the zigzag map," Analog Integrated Circuits and Signal Processing, vol. 73, pp. 363-374, 2012.
[79] T. K. Ksheerasagar, S. Anuradha, G. Avadhootha, and K. S. R. Charan, "Performance analysis of DS-CDMA using different chaotic sequences," in

Wireless Communications, Signal Processing and Networking (WiSPNET), International Conference on, pp. 2421-2425, 2016.
[80] W. M. Tam, F. C.-M. Lau, C. Tse, and A. J. Lawrance, "Exact analytical bit error rates for multiple access chaos-based communication systems," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 51, pp. 473-481, 2004.
[81] R. Matthews, "On the derivation of a “chaotic” encryption algorithm," Cryptologia, vol. 13, pp. 29-42, 1989.
[82] Z. Li, K. Li, C. Wen, and Y. C. Soh, "A new chaotic secure communication system," IEEE Transactions on communications, vol. 51, pp. 1306-1312, 2003.
[83] M. Ahmad and O. Farooq, "Chaos based PN sequence generator for cryptographic applications," in Multimedia, Signal Processing and Communication Technologies (IMPACT), 2011 International Conference on, pp. 83-86, 2011.
[84] S.-L. Chen, S.-M. Chang, W.-W. Lin, and T. Hwang, "Digital secure-communication using robust hyper-chaotic systems," International Journal of Bifurcation and Chaos, vol. 18, pp. 3325-3339, 2008.
[85] M. I. Sobhy and A.-E. Shehata, "Methods of attacking chaotic encryption and countermeasures," in Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on, 2001, pp. 1001-1004.
[86] B. Jovic and C. Unsworth, "Performance comparison of multi-user chaos-based DS-CDMA synchronisation unit within AWGN and Rayleigh fading channel," Electronics Letters, vol. 43, pp. 988-989, 2007.
[87] K. Klomkarn, A. Jansri, and P. Sooraksa, "A design of stream cipher based on multi-chaotic functions," in Communications and Information Technology, 2004. ISCIT 2004. IEEE International Symposium on, pp. 931-935, 2004.
[88] X. Wang, X. Wang, J. Zhao, and Z. Zhang, "Chaotic encryption algorithm based on alternant of stream cipher and block cipher," Nonlinear Dynamics, vol. 63, pp. 587-597, 2011.
[89] W.-k. Wong, L.-p. Lee, and K.-w. Wong, "A modified chaotic cryptographic method," in Communications and Multimedia Security Issues of the New Century, ed: Springer, pp. 123-126, 2001.
[90] M. Baptista, "Cryptography with chaos," Physics Letters A, vol. 240, pp. 50-54, 1998.
[91] M. A. Aseeri, M. I. Sobhy, and P. Lee, "Lorenz chaotic model using filed programmable gate array (fpga)," in Circuits and Systems, 2002. MWSCAS-2002. The 2002 45th Midwest Symposium on, pp. I-527, 2002.
[92] G. R. Goslin, "A Guide to Using Field Programmable Gate Arrays (FPGAs) for Application-Specific Digital Signal Processing Performance. Xilinx," Inc., http://www. xilinx. com, 1995.
[93] K. M. Cuomo, A. V. Oppenheim, and S. H. Strogatz, "Synchronization of Lorenz-based chaotic circuits with applications to communications," IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 40, pp. 626-633, 1993.
[94] L. Cong and W. Xiaofu, "Design and realization of an FPGA-based generator for chaotic frequency hopping sequences," IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 48, pp. 521-532, 2001.

[95] M. Azzaz, C. Tanougast, S. Sadoudi, and A. Dandache, "Real-time FPGA implementation of Lorenz's chaotic generator for ciphering telecommunications," in Circuits and Systems and TAISA Conference, 2009. NEWCAS-TAISA'09. Joint IEEE North-East Workshop on, 2009, pp. 1-4.
[96] D. Majumdar, R. Moritz, H. Leung, and J. M. Brent, "An enhanced data rate chaos-based multilevel transceiver design exploiting ergodicity," in MILITARY COMMUNICATIONS CONFERENCE, 2010-MILCOM 2010, pp. 1256-1261, 2010.
[97] P. Giard, G. Kaddoum, F. Gagnon, and C. Thibeault, "FPGA implementation and evaluation of discrete-time chaotic generators circuits," in IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society, pp. 3221-3224, 2012.
[98] L. Merah, A. Ali-Pacha, N. H. Said, and M. Mamat, "A pseudo random number generator based on the chaotic system of Chua's circuit, and its real time FPGA implementation," Applied Mathematical Sciences, vol. 7, pp. 2719-2734, 2013.
[99] S. Liu, J. Sun, Z. Xu, and Z. Cai, "An improved chaos-based stream cipher algorithm and its VLSI implementation," in Networked Computing and Advanced Information Management, 2008. NCM'08. Fourth International Conference on, pp. 191-197, 2008.
[100] S. Sadoudi, C. Tanougast, and M. S. Azzaz, "A new robust additive hyperchaos masking algorithm for secure digital communications," in Control, Decision and Information Technologies (CoDIT), 2013 International Conference on, pp. 501-504, 2013.
[101] Y. Wu, Y. Zhou, and L. Bao, "Discrete wheel-switching chaotic system and applications," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 61, pp. 3469-3477, 2014.
[102] A. A. Khare, P. B. Shukla, and S. C. Silakari, "Secure and Fast Chaos based Encryption System using Digital Logic Circuit," International Journal of Computer Network and Information Security, vol. 6, p. 25, 2014.
[103] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," International Journal of Bifurcation and Chaos, vol. 16, pp. 2129-2151, 2006.
[104] G. Alvarez, F. Montoya, M. Romera, and G. Pastor, "Cryptanalyzing a discrete-time chaos synchronization secure communication system," Chaos, Solitons & Fractals, vol. 21, pp. 689-694, 2004.
[105] L. R. Knudsen, Block ciphers-analysis, design and applications: Citeseer, 1994.
[106] N. I. o. S. a. Technology, "Random Number Generation and Testing," ed, 2012.
[107] T. Li, J. Ren, Q. Ling, and A. Jain, "Physical layer built-in security analysis and enhancement of CDMA systems," in Military Communications Conference, 2005. MILCOM 2005. IEEE, pp. 956-962, 2005.
[108] M. Electronics, "One Time Pad Encryption," http://www.cryptomuseum.com/manuf/mils/files/mils_otp_proof.pdf.
[109] B. Schneier, "Applied Cryptography: Protocols, Algorithms, and Source Code in C. Wiley," New York, 1996.
[110] N. J. Croft and M. S. Olivier, "Using an approximated one-time pad to secure short messaging service (SMS)," in Proceedings of the Southern African

Telecommunication Networks and Applications Conference. South Africa, pp. 26-31, 2005.
[111] M. Suneel, "Electronic circuit realization of the logistic map," Sadhana, vol. 31, pp. 69-78, 2006.
[112] C. F. Lam, D. T. Tong, M. C. Wu, and E. Yablonovitvh, "Experimental demonstration of bipolar optical CDMA system using a balanced transmitter and complementary spectral encoding," IEEE Photonics Technology Letters, vol. 10, pp. 1504-1506, 1998.
[113] Mathworks, "Uniform Noise Generator (UNG) " https://uk.mathworks.com/help/comm/ref/uniformnoisegenerator.html.
[114] B. Jovic and C. P. Unsworth, "Chaos-based multi-user time division multiplexing communication system," IET Communications, vol. 1, pp. 549-555, 2007.
[115] N. Instrument, "Introduction to FPGA Technology: Top 5 Benefits," 2012.
[116] K. Underwood, "FPGAs vs. CPUs: trends in peak floating-point performance," in Proceedings of the 2004 ACM/SIGDA 12th international symposium on Field programmable gate arrays, pp. 171-180, 2004.
[117] X. company, https://www.xilinx.com/products/design-tools/vivado/integration/sysgen.html.
[118] X. company, Xilinx company https://www.xilinx.com/products/design-tools/ise-design-suite/ise-webpack.html.
[119] W. D. Lautenschläger and S. Schabel, "Clock recovery," ed: Google Patents, 2002.
[120] J. R. Francis and A. Gupta, "Phase detector for high speed clock recovery from random binary signals," ed: Google Patents, 1998.
[121] R. D. Roberts, "Apparatus for clock recovery from digital data," ed: Google Patents, 1989.
[122] P. K. Hanumolu, G.-Y. Wei, and U.-K. Moon, "A wide-tracking range clock and data recovery circuit," IEEE journal of solid-state circuits, vol. 43, pp. 425-439, 2008.
[123] B. Razavi, "Challenges in the design high-speed clock and data recovery circuits," IEEE Communications Magazine, vol. 40, pp. 94-101, 2002.
[124] D. L. Hershberger, "Nrz clock and data recovery system employing phase lock loop," ed: Google Patents, 1992.
[125] M. Belkin, "Phase tolerant bit synchronizer for digital signals," ed: Google Patents, 1983.
[126] M. Cooperman and P. L. Andrade, "Data communication system," ed: Google Patents, 1993.
[127] C. Lee and R. Venkata, "Clock data recovery with double edge clocking based phase detector and serializer/deserializer," ed: Google Patents, 2008.
[128] M. U. A. Belorkar and D. S. Ladhake, "Design of low power phase locked loop (PLL) using 45nm VLSI Technology," International journal of VLSI design & Communication Systems (VLSICS), vol. 1, 2010.
[129] A. Aggarwal, A. Satija, and T. Nagpal, "FIR filter designing using Xilinx system generator," International Journal of Computer Applications, vol. 68, 2013.
[130] L. M. Pecora and T. L. Carroll, "Synchronization in chaotic systems," Physical review letters, vol. 64, p. 821, 1990.

[131] J. Suykens, P. Curran, and L. Chua, "Master-slave synchronization using dynamic output feedback," International Journal of Bifurcation and Chaos, vol. 7, pp. 671-679, 1997.
[132] M. E. Yalcin, J. A. Suykens, and J. Vandewalle, "Master–slave synchronization of Lur'e systems with time-delay," International Journal of Bifurcation and Chaos, vol. 11, pp. 1707-1722, 2001.
[133] J. M. Muñoz-Pacheco, E. Zambrano-Serrano, O. Félix-Beltrán, L. C. Gómez-Pavón, and A. Luis-Ramos, "Synchronization of PWL function-based 2D and 3D multi-scroll chaotic systems," Nonlinear Dynamics, vol. 70, pp. 1633-1643, 2012.
[134] N. Jiang, W. Pan, B. Luo, S. Xiang, and L. Yang, "Bidirectional dual-channel communication based on polarization-division-multiplexed chaos synchronization in mutually coupled VCSELs," IEEE Photonics Technology Letters, vol. 24, pp. 1094-1096, 2012.
[135] J. Lu, D. W. Ho, J. Cao, and J. Kurths, "Exponential synchronization of linearly coupled neural networks with impulsive disturbances," IEEE Transactions on Neural Networks, vol. 22, pp. 329-336, 2011.
[136] X. Yang, J. Cao, and J. Lu, "Stochastic synchronization of complex networks with nonidentical nodes via hybrid adaptive and impulsive control," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 59, pp. 371-384, 2012.
[137] R. Rovatti, G. Setti, and G. Mazzini, "Statistical features of chaotic maps related to CDMA systems performance," in Proc. MTNS'98, 1998.
[138] G. Setti, G. Mazzini, and R. Rovatti, "Gaussian characterization of self-interference during synchronization of chaos based DS-CDMA systems," in Electronics, Circuits and Systems, 1998 IEEE International Conference on, pp. 231-234, 1998.
[139] M. Shala, "Optical Communication," Doctoral thesis, 2010.
[140] M. Delgado-Restituto and A. Rodriguez-Vazquez, "Mixed-signal map-configurable integrated chaos generator for chaotic communications," IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 48, pp. 1462-1474, 2001.
[141] F. Pareschi, G. Setti, and R. Rovatti, "Implementation and testing of high-speed CMOS true random number generators based on chaotic systems," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 57, pp. 3124-3137, 2010.
[142] G. Alvarez, F. Montoya, M. Romera, and G. Pastor, "Cryptanalysis of a chaotic secure communication system," Physics Letters A, vol. 306, pp. 200-205, 2003.
[143] G. Alvarez, F. Montoya, M. Romera, and G. Pastor, "Keystream cryptanalysis of a chaotic cryptographic method," Computer Physics Communications, vol. 156, pp. 205-207, 2004.