FPGA Implementation of the Non-Redundant Error-Correcting Scheme Based on Chaotic Dynamics

Hikmat N. Abdullah and Sarab K. Mahmood

Department of Electrical Engineering, Al-Mustansiryah University, Bab Al-Moadham, Baghdad 10001, Iraq
hikmat_04@yahoo.com

Abstract: This paper presents the design procedure and implementation results of a non- redundant error correcting scheme based on chaotic dynamics using Altera Cyclone III family FPGA board. The scheme is first designed using MATLAB-Simulink then converted to VHDL codes for implementation purposes using the modern tool "Simulink HDL Coder". The simulation waveforms have been obtained using ModelSim Altera 6.5b.Synthesis reports and board programming files have been obtained using the QUARTUS II 9.1 package. ALTERA-Cyclone III FPGA board has been used as target devices for implementation purpose. The hardware test results show that the coding scheme functions successfully.

[Abdullah HN. and Mahmood SK. **FPGA Implementation of the No-Redundant Error-Correcting Scheme Based on Chaotic Dynamics.** *J Am Sci* 2014;10(11):25-30]. (ISSN: 1545-1003). http://www.jofamericanscience.org. 4

Keywords: Analogue channel coding; Chaotic signals; noncoherent detection; Field programmable gate arrays.

1. Introduction

Digital communication systems using chaos attracts many researchers for their spread spectrum nature (Kolumbán *et al.*, 1996) and (Seyyedrezae, 2014). Chaotic sequences obtained from a certain class of difference equations are non-periodic and sensitive to initial conditions, and it is difficult to predict their future behavior from past observations. Chaos modulation schemes have been proposed for communications such as chaos shift keying (CSK) (Long *et al*, 2012). However, many researchers have focused on the development of non-coherent detection systems that do not need to recover basis signals (unmodulated carriers) at the receiver.

The optimal detection proposed by Schimming and Hasler (2000) is well known as a non-coherent receiver. However, if a chaotic sequence length N becomes long, calculation becomes very complicated and the signal detection becomes difficult. Shintaro (2009) proposed a method of detecting symbols from the calculation value of the shortest distance between received signals and chaotic map, i.e., a suboptimal receiver (Schimming and Hasler, 2002) .Instead of calculating the PDF, the suboptimal receiver approximates the PDFs by calculating the shortest distance between the received signals and the chaotic map.

In this paper, the designed suboptimal receiver that combined chaos based non-coherent modulation and non-redundant error correcting coding is designed as MATLAB Simulink blocks and implemented on FPGA board without the need to learn VHDL or even other FPGA design entries. The designed system uses two successive chaotic sequences from the same chaotic map; the initial value of the second sequence is used as the end value of the first sequence. This feature gives the receiver additional information to correctly recover the information data. The written MATLAB code/Simulink blocks should take into ahead what is so called fixed point arithmetic notations (Abdullah and Valenzuela, 2008).

Although the non-redundant error correcting scheme based chaotic dynamics is analogue in nature, it is also interested to investigate its equivalent digital version. In this case the system will lose some of its features like wideband spectrum but will gain other features like implementation flexibility, ease of re-programming using different chaotic maps and the use as alternative channel coding scheme for existing digital communication systems.

2. The system structure

The block diagram of the chaotic system with channel coding is shown in Figure 1. In the transmitter side, the data are CSK modulated by selecting one out of two chaotic carrier signals according to the value of input data (0 or 1).

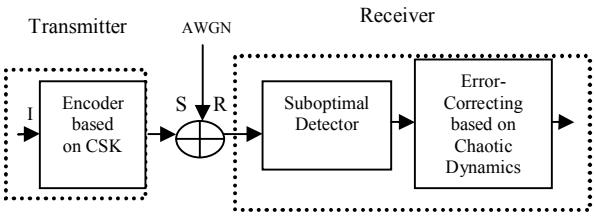


Figure 1. Discrete-time binary CSK communication system with error correcting coding.

The modulated data are simultaneously encoded during modulation process by creating a statistical relation between each two successive chaotic signals

corresponds to two successive data bits. The receiver side contains a suboptimal detector based on short distance calculation and an error correcting decoder based on the dynamics of statistically related chaotic signals. The decoder will decide whether the received signal is 0 or 1 according to decoding criteria depends on short distance measures provided by the suboptimal detector and analyzing the dynamics of statistical relations of incoming sequences.

3. The mathematical model of the system

A chaotic sequence is generated using chaotic map in order to encode the binary data. Logistic map is used due to its simplicity. The Logistic chaotic map is defined as:

$$x_{n+1} = ax_n(1 - x_n) \qquad 0 \le x \le 1 \qquad (1)$$

where a is the parameter of this map which is a positive real constant. CSK modulation is done using two chaotic sequences, each with different generation map. The modulator generates the chaotic sequence depending on the value of the binary data. If the data symbol "1", chaotic sequence given by Equation (1) is used. If the data symbol "0", chaotic sequence given by the reverse function of equation (1) is used as shown in Figure 2.

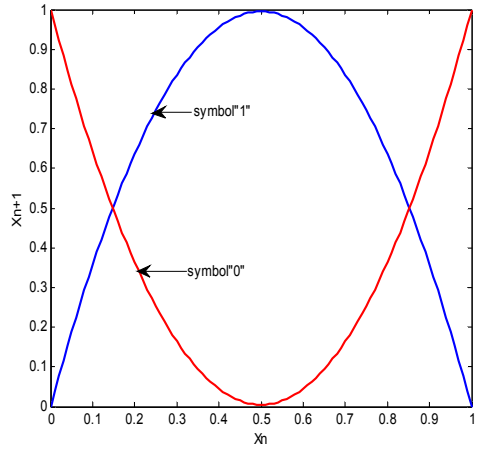


Figure 2. Logistic chaotic map.

When K bit information is transmitted and N chaotic signals are generated from identical chaotic map for each bit, the amount of the data becomes $K \times N$, where N is the chaotic sequence length. At the beginning, the initial value is chosen randomly in order to generate a chaotic sequence and is different for symbol "1"(x_0) and symbol "0" (y_0).

4. The detection algorithm

The suboptimal receiver approximates the Probability Density Function PDF by calculating the shortest distance between the received signal and the chaotic map in N_d-dimensional space (N_d : 2,3,.....) using N_d successive received signals (N_d must be $\le N$). Figure 3 shows the 2-dimensional space of the Logistic map whose coordinates correspond to the two successive received signals R=(r_i,r_{i+1}) where i=1,2.....N-1.

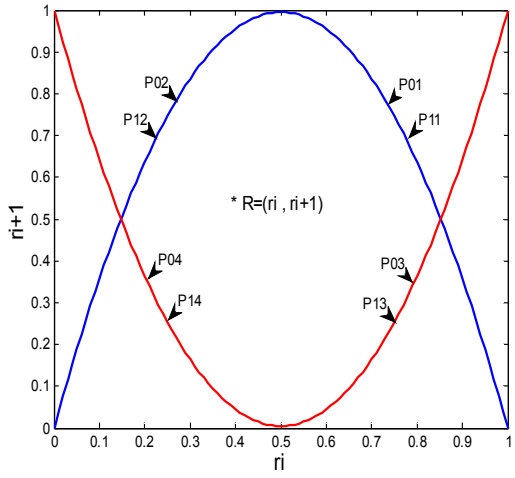


Figure 3. Two-Dimensional Space of (r_i ,r_{i+1}).

To determine which map is closest to point R in the 2- dimensional space in Figure 3, the shortest distance between the point and the map is calculated. The receiver calculates the shortest distance using the scalar product. Two vectors are required for a scalar product. Any two points of $\mathbf{P_0}$= (x_o, y_o) and $\mathbf{P_1}$= (x_1, y_1) are chosen from each curved line in the space of Figure 3, as shown in Figure 4.

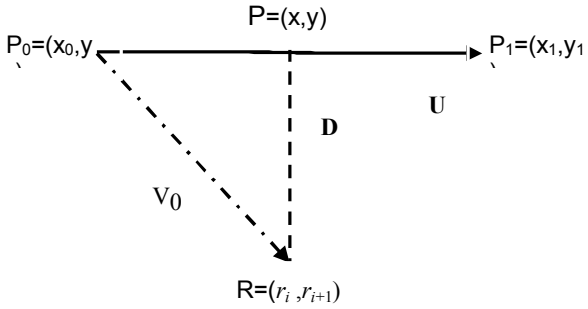


Figure 4. Calculation of the shortest distance.

From Figure (4), the closest point **P** =(X, Y) to R and accordingly the shortest distance D are calculated using the following equations [5-7]:

$$A = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} \qquad (2)$$

$$u = (l, m) = \left(\frac{x_1 - x_0}{A}, \frac{y_1 - y_0}{A}\right) \qquad (3)$$

$$vo = (r_i - x_0, r_{i+1} - y_0) \qquad (4)$$

$$T = l(r_i - x_0) + m(r_{i+1} - y_0) \qquad (5)$$

$$P = (x, y) = (Tl + x_0, Tm + y_0) \qquad (6)$$

$$D = \sqrt{(x - r_i)^2 + (y - r_{i+1})^2} \quad (7)$$

For the 2-dimensional case, there are two curved lines in the space for symbol "1" (number of curved lines $=2^{Nd-1}$). So, the minimum value in two distances is decided as the shortest distance D_1 for symbol "1". In the same way, D of symbol "0" is decided as D_0. Both $D_1$and $D_0$are calculated for all k and their summations $\sum D_1$ and $\sum Do$ are found. Finally, the decoded symbol is decided as 1 (or 0) for $\sum D_1 < \sum Do$(or$\sum D_1 > \sum Do$).

5. The error-correction algorithm

The error-correction algorithm is based on chaotic dynamics and it is initiated immediately after suboptimal detection as shown in Figure 5.

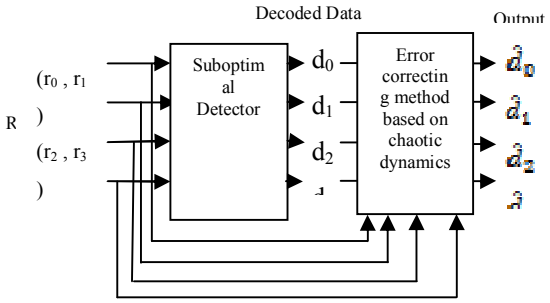


Figure 5. Block diagram of the error-correction algorithm.

The processing steps of the error correction algorithm can be described as follows: at first, the receiver arranges the received samples according to the decoded symbols and produces two large sequences; *Seq*"1"(.) and *Seq*"0(.). Next, the receiver apply the suboptimal detection algorithm, i.e., the calculation of the shortest distance between *Seq*"1"(.) and the chaotic map for symbol "1" described by D_1(*Seq*"1"(.)) and the calculation of the shortest distance between *Seq*"0"(.) and the chaotic map for symbol "0" described by D_0(*Seq*"0"(.)) as shown in Figure 6a. In addition, a reference distance D_{Base} is defined for analyzing the chaotic dynamics as:

$$D_{Base} = D_1\big(Seq"1"(.)\big) + D_0\big(Seq"0"(.)\big) \quad (8)$$

Then, a new comparison distances are defined to analyze the chaotic dynamics as $D_R(m)$ where (m= 0,1, … K-1) and the subscript R means the first character of "Reverse". $D_R(m)$ means the shortest distance between arranged sequence when the m-th decoded symbol is reversed and the chaotic map corresponding to their sequences as shown in Figure 6b.

If the receiver can recover the information symbols and arrange the received samples correctly, then the values of $D_R(m)$ becomes larger when compared with D_{Base}. If an error is occurred when the receiver detects symbols, then one of $D_R(m)$ (i.e. $D_R(0)$, …. $D_R(K-1)$) becomes smaller as compared with D_{Base} and other $D_R(m)$. As shown in Figure 6b when an error is occurred at R_2, the value of $D_R(2)$ becomes smaller as compared with D_{Base} and other $D_R(m)$. In this case, the receiver can locate that the error occurs at R_2 and correct it.

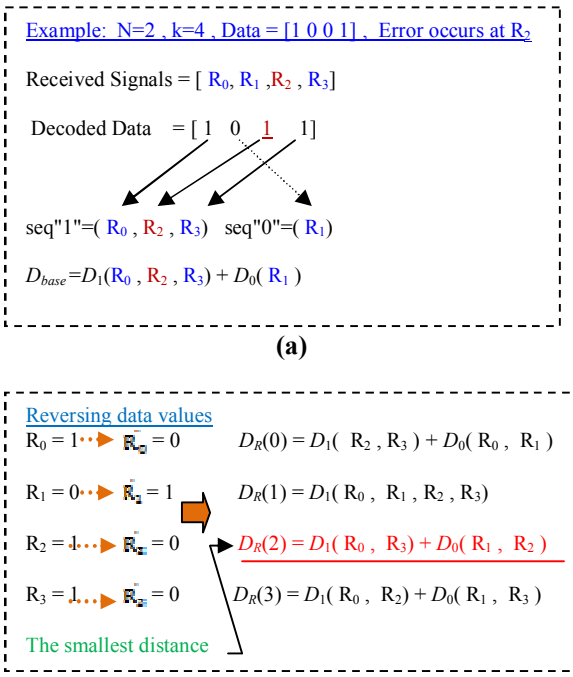


Figure 6. Analysis of Chaotic dynamics using suboptimal detection.(a) Calculation of D_{Base} ,(b) Calculation of $D_R(m)$.

6. Design flow of the system

Figure 7 shows the proposed design flow used to implement our non-redundant error correcting system using FPGA. In order to implement the non-redundant error-correcting system, its parameters must be set first. Our system has the following parameters: k=4, N=2, N_d=2, the initial conditions are x_0=1, and y_0=0.875. The design of our non-redundant error-correcting system is carried out using MATLAB Simulink to efficiently implement it using FPGA technology (The MathWorks Inc., 2014) The information obtained from MATLAB-Simulink implementation would help the designer to correctly optimize the cost and speed in FPGA implementation.

Simulink HDL coder is a new tool comes with MATLAB-Simulink software package can be used to create hardware description language (HDL) code based on Simulink® models and Stateflow® finite-state machines. Simulink HDL coder compatibility checker can be used to examine MATLAB-Simulink

model blocks for HDL code generation compatibility. The coder generates VHDL code that builds the design included in the model. The test bench together with HDL simulation tools can be used to run the generated HDL code and evaluate its behavior.

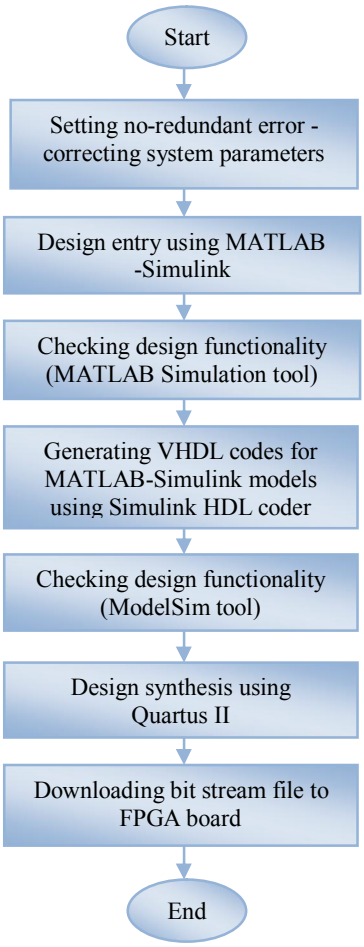


Figure 7. Flow design of the error-correcting system.

The functionality of the designed system can be verified using Altera/Mentor Graphics ModelSim (6.5b) simulation tool. Therefore, the test bench codes are compiled and simulated using the generated scripts by the HDL coder. The synthesis operation ,which is the transformation of the high-level VHDL language that describes the circuit at the Register Transfer Level (RTL) into a lower level of logic abstraction, produces a bit stream file which can be downloaded in the FPGA board. The bit stream file of the designed system has been successfully downloaded to Altera-Cyclone III EP3C120 FPGA board. Figure 8 shows the photo of the implemented system. The internal clock available in to Altera-Cyclone III EP3C120 FPGA board (25 MHz accessed with pin AH15) is used to drive the system.

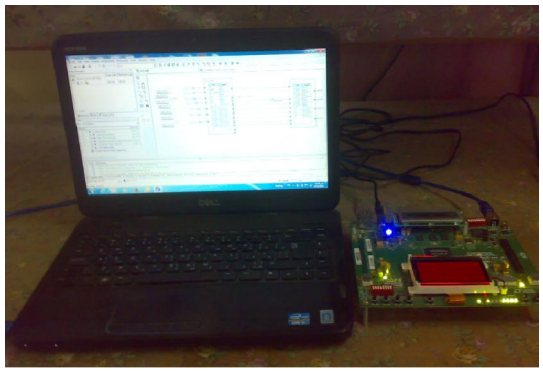


Figure 8. Photo of the implemented system using Cyclone III FPGA board.

7. Implementation results

The performance results of our designed coding scheme in AWGN channel using Matlab-Simulink simulation will be published in other paper (Abdullah and Mahmood, 2014). Here we will present the results of FPGA implementation of the system in the absence of noisy channel. These results are divided into three categories: synthesis reports summary, ModelSim simulation and hardware test results.

- ***Synthesis Reports***

The compilation report provides a lot of information that may be of interest to the designer. Table 1 shows the table of the summary of the compilation report of the designed system.

Table (1) The compilation report summary of the designed system.

Total logic elements	2.804/119.088 (2%)
Total combinational function	2.801/119.088 (2%)
Total pins	12/532(2%)
Total PLLs	0/4 (0%)
fmax	27.26 MHz
***tco* (worst case)**	75.928 ns

- ***ModelSim Simulation Results***

Figures 9 and 10 show the transmitted samples vectors and the decoded symbols when the information symbol is [1001] and [1101] respectively. It can be seen from this figures that the decoded bit streams (the waveforms named: subsystem1_tb/out2) is exactly the same of the transmitted ones (named: subsystem1_tb/in1). However, there is some delay between the transmitted and decoded bits which is normal due to the required processing time at both transmitter and receiver.

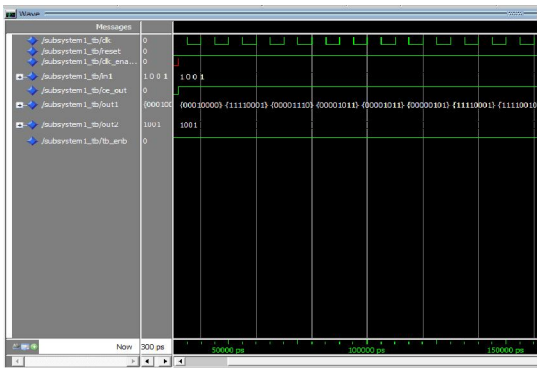


Figure 9. The Transmitted and the decoded symbols when the information symbol is [1 1 0 1].

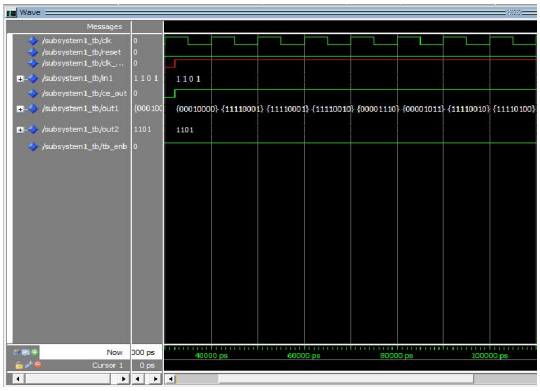


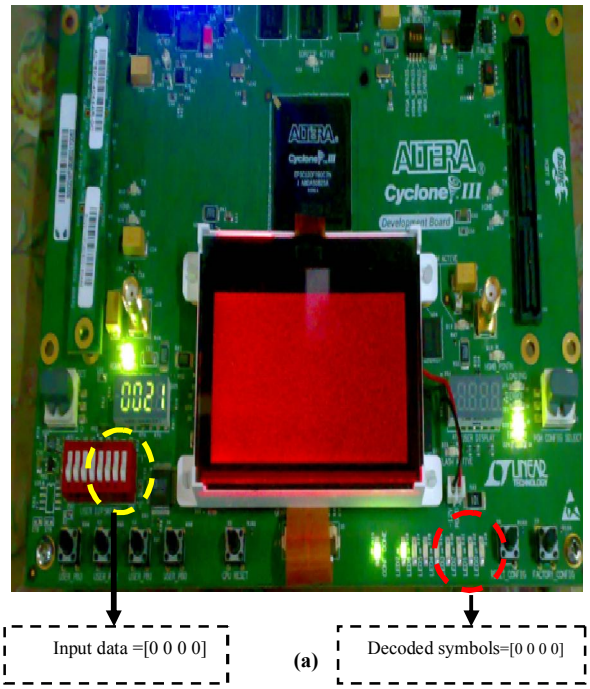
Figure 10. The Transmitted and the decoded symbols when the information symbol is [1 1 0 1].

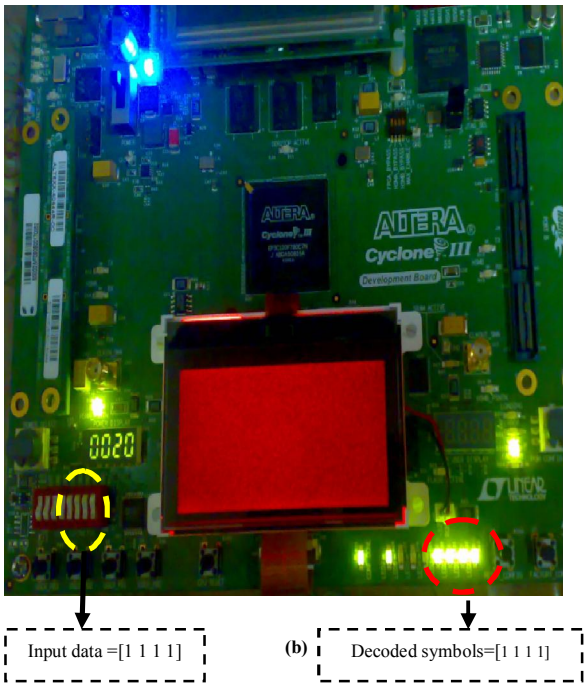
- ***Experimental Test Results***

After compiling the VHDL code by using Quartus II and downloading the bit streams successfully to Cyclone III kit, the final step is to test the real hardware action. To do that, the utilities available on the kit like switches and light emitting diodes (LEDs) are used as test measures. The inputs of the transmitter are connected to a four bush bottom switches out of eight ones available in the board such that the status of these switches (ON/OFF) represents the input data values ("1"/"0"). Similarly, the decoded data are connected to a four LEDs out of eight ones available in the board such that their status represents the decoded data values. The switches and LEDs used for experimental test are marked by dashed circle as shown in Figure 11. In this figure different data streams are considered. In all cases, it can be seen that the decoded data are exactly the same as the input data which proves the successful decoding process.

8. Conclusions

The designed no redundant error-correcting scheme could be implemented using FPGA. This approach is flexible, economical and easy in implementation and satisfies the reprogramming requirement capability of the designed system. Simulink HDL coder automates the hardware implementation process and represents a flexible design tool. Simulink HDL coder does not support all





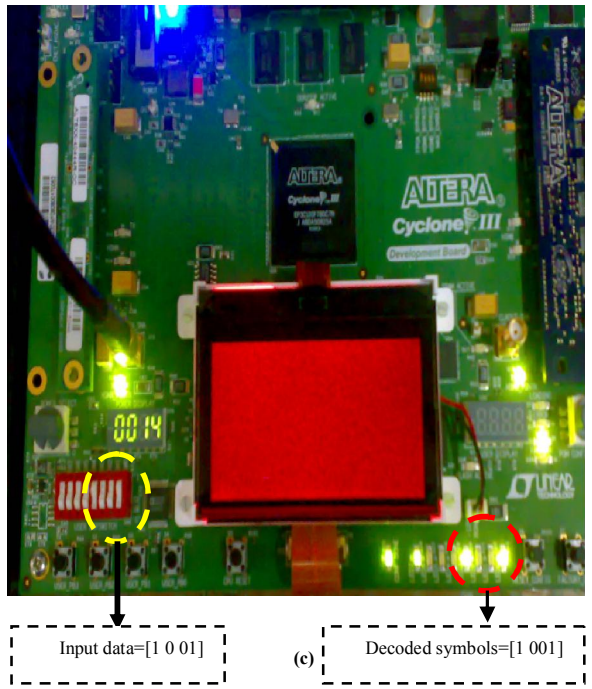


Figure 11. Experimental test results for different inputs data values.

MATLAB-Simulink blocks, so to generate VHDL codes for such blocks they must be redesigned according to their basic design blocks supported by Simulink HDL coder.

References

1. Kolumbán G, Vizvari G, Schwarz W and Abel A. Differential chaos shift keying: A robust coding for chaos communication. Proceedings of International Workshop on Nonlinear Dynamics NDES, June 1996: 87-92.
2. Seyyedrezae S. The application of chaos/complexity theory in classroom teaching, task design and lesson planning. Journal of Language Sciences & Linguistics, 2014: 2 (2):27-32.
3. Long M, Peng F, and Chen Y. Bit error rate improvement for chaos shift keying chaotic communication systems. IET Communications, 2012: 6 (16): 2639-2644
4. Hasler, M. and Schimming T. Chaos communication over noisy channels. International. Journal of Bifurcation and Chaos, 2000: 10(4): 719-736.
5. Shintaro A, Yoshifumi N and Takaya Y. Error-correcting scheme without redundancy code using chaotic dynamics for non-coherent chaos communications. Proceedings of International Conference on Nonlinear Circuit Networks, Tokushima 2009.
6. Shintaro A, Yoshifumi N. Suboptimal receiver using shortest distance approximation method for chaos shift keying. Journal of Signal Processing, 2009: 13(2): 161- 169.
7. Hasler M. and Schimming T. Optimal and suboptimal chaos receivers. Proceedings of IEEE, 2002: 90(5):733-746.
8. Abdullah H and Valenzuela A. A Joint Matlab/FPGA design of AM receiver for teaching purposes. EMNT2008 conference, Munich University of Technology, Germany, 2008.
9. SIMULINK® HDL Coder™ user's guide. The MathWorks, Inc. 2014.
10. SIMULINK communication systems simulation for MATLAB. The Mathwork, Inc. 2014.
11. SIMULINK block library of the communication toolboxes reference. The Mathwork, Inc., 2014.
12. Abdullah H. and Mahmood S. Performance evaluation of non-redundant error correcting scheme using logistic chaotic map. To be published in Engineering and development Journal, ISSN 1813-7822, December 2014.

9/21/2014