

Chaotic Spreading Spectrum System Development and Implementation Using FPGA/VHDL for Secure Wireless Communications

Group One

Date: October 15, 2002

Technical Advisor:

Professor Yu-Dong Yao

Signatures:

Team Members:

Anthony T. Bianco

John Choi

Ivan Quizhpi

Azamat Villar

Rhys dela Cruz

We pledge our honor that we have abided by the Stevens Honor Code.

Table of Contents

I.) Abstract

II.) Project Proposal Plan

- 1.) Introduction**
- 2.) Design Requirements**
- 3.) Design Approaches**
- 4.) Financial Budgets**
- 5.) Project Schedule**

III.) Conclusion

IV.) References

V.) Appendix

- 1.) Gantt Chart**
- 2.) LRC Chart**

I.) Abstract

This project will address chaotic spread-spectrum based system as well as secure wireless systems using FPGA and VHDL. There will be many subjects to learn and resolve. For example, after studying the dynamics of chaotic system, one problem to be addressed and worked on during the project is generating a chaotic signal. Following this, another problem is developing a communication system with chaotic carrier signals. Our group will further have to learn the uses and advantages of FPGA and VHDL to develop and implement the entire system. Programming two FPGA and VHDL boards will thus be the key in approaching this project. One board will be used as the transmitter and the other will be the receiver. Before working on the goals and requirements of our project, we will implement digital phase modulation or PSK to show that our hardware and entire setup is valid. The final outcome of this and the entire project will be a small setup consisting of two laptops and the two boards demonstrating that secure communication using the demonstrated methods is indeed possible.

II.) Project Proposal Plan

The present state of the project we are working on is in the beginning stages. This is the first time that Chaotic Spread Spectrum (SS) will be implemented in secure wireless communications. Our group has a great task in front of us. With the help of our professors, peers, the school, and other sources, we will be successful in this project. We are looking forward to learning more about wireless communications. We would like to achieve a greater understanding about FPGA and VHDL. We believe that this project will give us great hands on experience that a class room can not provide. We are looking forward to completing this project. This project is a feasible project that just needs researching and reading for our part.

Approaches are also discussed in the latter part of this document. If the approach seemed to be problematic, a group meeting will be held. Once the group is at a consensus, we then would speak to Professor Yao and have his input in the problem. The evaluation of our various approaches is time. We are given a short time span. Any parts we are able to cut, that are not crucial to the success of our project, will be cut. We can not let ourselves focus on the parts of the project that has insignificant importance to the project.

Costs to this project are minimal. Stevens has a state of the art Wireless Laboratory. The Director of Computer and Electrical Engineer has already given us permission to use the test equipment, which would be the most expensive parts of the project. By researching through the Internet, we were able to find the most affordable products in our product list.

II - 1.) Introduction

Chaotic spread-spectrum system development and implementation using FPGA/VHDL for secure wireless communications is a unique project in that it has not been attempted before. There have been advances and much research in the individual fields of spread-spectrum communications, chaotic signals, wireless security, and of course FPGA and VHDL. But to combine all of these components into this one project is a first, and this is a timely engineering challenge. This group decided to work on such a project because it will prove to be worthy and useful for homeland securities, since chaotic spread-spectrum is very secure for wireless systems. This will also be the first test bed for Wireless Systems for Stevens Institute of Technology.

One can see that some areas of this project have been researched individually very extensively. For example, one can consult Communication Systems, The Essential

Guide to Wireless Communications Applications, The Essential Guide to Telecommunications, Principles of Wireless Networks, Wireless Communications – Principles and Practice, Introduction to Spread Spectrum Communications, and A VHDL Primer. These are all good resources for one to bring him up to date in the different areas of this project in an effective manner. These are the same resources the group intends to refer to in learning and applying the different aspects of this project. These books will be used intensively by the group. Professor Yao will be giving us other forms of publications to assist us.

There are different ways in which our group adds new features and improved performance parameters to enhance the resultant system performance of our project. For example, we will use FPGAs as opposed to Digital Signal Processors or even microprocessors. This aspect of our design will greatly reduce the overall cost. Also, we will use chaotic spread-spectrum for wireless security as opposed to other forms of security. Chaotic spread-spectrum is very secure and will thus be really useful and effective in our project. These can be seen as improvements in previous methods to provide security in our project area.

This project will operate in any area that requires the study of spread-spectrum communications, chaotic signals, wireless security, and FPGA and VHDL. Our advances in applying chaotic spread-spectrum by using FPGA boards and VHDL language will be a great asset to any company that wishes to work with wireless communication. Our project in its final form will save a company time and money in that the company will not have to spend manpower and the time that our group has already spent in developing this project. For example, we will work on generating a code in VHDL for chaotic spread-spectrum. A company will save time and resources on not having to work on this code. Further, our entire project could be extended to new users such as colleges that want to research and instruct in the areas that this project covers, thus broadening the application of this project.

II - 2.) Design Requirements

Our design is based on Chaotic Spread Spectrum (SS) theory for encryption in communication for Homeland Security. Chaotic Spread Spectrum is based on the chaos theory which is a deterministic, non-period, non-converging and bounded random-like process that is dependent upon its parameter and initial condition.

We will apply the chaos theory by means of a programmable logic device called Field Programmable Gate Array (FPGA) and by using VHSIC Hardware Description Language (VHDL) as our means of implementing the chaos theory. VHDL is

programming development software used for digital circuits. We will also be using VHDL in the Xilinx WebPACK. That is the recommended language used for programming FPGAs.

During our first semester, we will first approach our project using a different well known method of encryption for transmitting and receiving digital signals. The method that will be implemented is phase modulation which is way of encoding information into a carrier wave by variation of its phase in accordance with an input signal. A simple method of phase modulation is called Phase Shift Keying (PSK).

PSK is a technique with only two states use to encrypt digital signals by shifting the period of a wave. The range of the carrier phase is $0 \leq \theta < 2\pi$.

$$u_m(t) = Ag_T(t)\cos(2\pi f_c t + 2\pi m/M), \quad m = 0,1,\dots,M-1$$

This equation is the general form of a set of M carrier-phase modulated signal waveforms where $g_T(t)$ determines the spectral characteristics of the transmitted signal, A is the amplitude, and M determines the phase modulation. $M = 2^k$ for M -ary phase modulation, where k is the number of information bits per transmitted symbol. For example, for a binary phase modulation $M = 2$ where for every transmitted symbol there is 1 bit of information.

Once we have learned to develop the VHDL source code for implementing the PSK equation, we can verify it's validity by sending data across two FPGA Spartan II based boards. After our tests pass we can start working on the chaos theory code and implement it into our FPGA boards and test the code for performance and accurate encryption.

II - 3.) Design Approaches

There are three possible design approaches we can take. We looked at three different ways we can approach this project. These approaches have been carefully examined. By talking to Professor Yao, our technical advisor, we were able to come up with a better and more reliable approach. Our final approach is finally established. While researching our project, we also realized that there will be basic steps that have to be performed. These steps are essential to the completion of our project. These steps are:

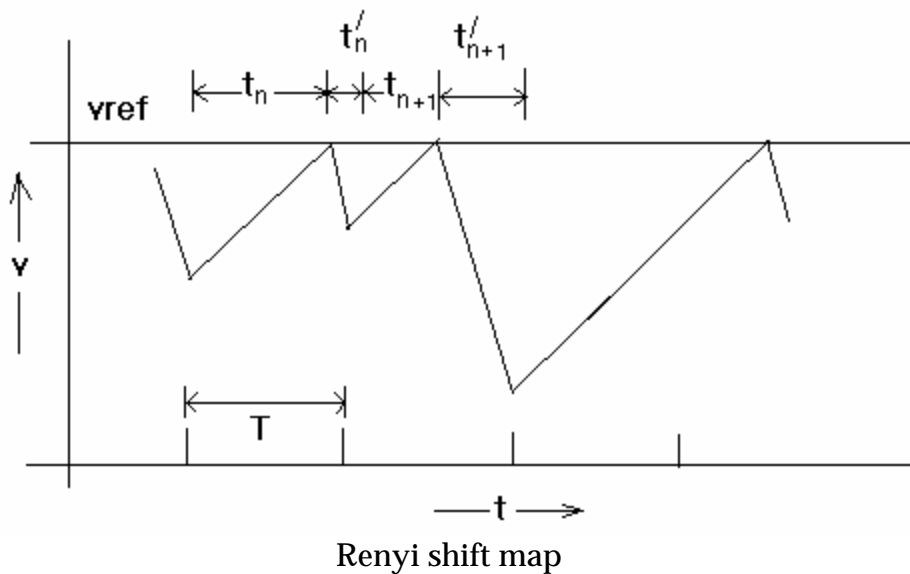
- ❖ Define the Project
- ❖ Identify the Product
- ❖ Order and Receive
- ❖ Research Chaotic Spread Spectrum (SS)
- ❖ Generate Chaotic SS Code
- ❖ Test Chaotic SS

These steps will be in all three approaches that we have come up with.

Approach One	
Step	Description
Define the Project	By this time, we would like to really identify our project and the project plan. This will be the time to really do more research on our project to fully understand the project.
Identify the Product	By this time, we would like to fully identify the items we need for this project. We would also like to order the items by this time also.
Order and Receive	We have to place an order on the types of boards we will be using. We will also begin to start to order the Parts in our parts list.
Research Chaotic SS	We will be receiving more information from Jin Yu. He is a graduate student that was assigned by Prof. Yao to help us work on the Chaotic SS part of the project. We will also be researching throughout the break.
Build Chaos Generator	By using the Renyi Map, we would be able to build a Chaos Generator.
Generate Chaotic Code	Generate a code in VHDL for Chaotic SS.
Test Chaotic SS	This is the actual project. We will be testing the Chaotic SS by using an FPGA board. The Language is VHDL.

Approach one is a feasible result. Although, we realized that there were many hidden items that has to be dealt with. We had two concerns for this approach. The first concern is testing. We realized that once we arrive at the testing stage, we had no way of actually testing for our results. Chaotic SS is still a developing technology. To our knowledge, no one has done anything on actual implementation of chaos SS into wireless communications. Therefore, once we have an actual output, we will never really be certain of our out put is correct. Another problem with testing is time. Doing all these tests might not allow us ample time to produce an end result.

The next problem with Approach one is the building of the Chaos Generator. With the implementation of the Renyi shift map, we can build a chaos generator. Learning to implement the Renyi shift map and achieving a chaos generator would become an issue of time. The Renyi shift map can be seen below. There is documentation that talks about creating a chaos generator. Although, we realized that this would become a major point of our project. If we continued in making this generator, it would totally take over our project. We felt it unnecessary to create the chaos generator because we did not want this to be the focus of our project.



Approach Two	
Step	Description
Define the Project	By this time, we would like to really identify our project and the project plan. This will be the time to really do more research on our project to fully understand the project.
Identify the Product	By this time, we would like to fully identify the items we need for this project. We would also like to order the items by this time also.
Order and Receive	We have to place an order on the types of boards we will be using. We will also begin to start to order the Parts in our parts list.
Research Chaotic SS	We will be receiving more information from Jin Yu. He is a graduate student that was assigned by Prof. Yao to help us work on the Chaotic SS part of the project. We will also be researching throughout the break.
Generate Chaotic Code	Generate a code in VHDL for Chaotic SS.
Test Chaotic SS	This is the actual project. We will be testing the Chaotic SS by using an FPGA board. The Language is VHDL.

Approach two takes the building of the chaos generator. This will give us ample time to work on the Chaotic SS. By taking this major task out, we have time to concentrate on the actual project definition. We decided we can simulate a chaos generator through the Xilinx WebPACK we download from the Xilinx website. We are all taking Digital System Design. In that class, we are talking about FPGA and VHDL. We are hoping that the class will help us understand VHDL and FPGA in a more comprehensive way. This class told us about the WebPACK. By buying the same FPGA board that the class is using, we can use this class to our advantage. When looking through the WebPACK, we realized that we can create a chaos generator. This cuts the time in having to design one by ourselves. This design would have included a circuit design. This circuit would then have to be integrated into the boards. By having a ready made generator, we are able to save time.

Approach two also has a problem. So far we look like we just jumping into the project without fully researching and testing our results. We encounter the same problem as approach one. This problem is the testing problem. We still have no way to test our results. If we can not test our results, we will be unable to make certain if our end results are accurate and reliable.

Approach Three	
Step	Description
Define the Project	By this time, we would like to really identify our project and the project plan. This will be the time to really do more research on our project to fully understand the project.
Identify the Product	By this time, we would like to fully identify the items we need for this project. We would also like to order the items by this time also.
Order and Receive	We have to place an order on the types of boards we will be using. We will also begin to start to order the Parts in our parts list.
Research PSK	Learn more about phase shift keying (PSK)
Generate Source Code	Generate a code in VHDL for PSK.
Test PSK	We will be using a PSK based system to test our FPGA board.
Research Chaotic SS	We will be receiving more information from Jin Yu. He is a graduate student that was assigned by Prof. Yao to help us work on the Chaotic SS part of the project. We will also be researching throughout the break.
Generate Chaotic Code	Generate a code in VHDL for Chaotic SS.
Test Chaotic SS	This is the actual project. We will be testing the Chaotic SS by using an FPGA board. The Language is VHDL.

Approach three is our final decision. By adding a section of Phase shift keying (PSK), we will be able to test the accuracy of our results. We needed a way to test our FPGA board setup. PSK is currently used in wireless communications. Since this system is already in use, and well documented, we can test our project against the results that the PSK as already given. Once we have the results of our project, we can compare our results to that of previously documented in PSK systems. Once we are successful with the PSK system, we can then research on the Chaotic SS system. With the knowledge gained from our classes and the testing of the PSK, we will be able to complete the chaotic SS in the given time limit. Having prior knowledge of FPGA and VHDL will assist us greatly in the Chaotic SS system, instead of just jumping into it.

II - 4.) Financial Budgets

Preliminary Budget:

<u>Category</u>	<u>Budgets Cost</u>
<i>Materials/Hardware</i>	
2 XSA-100 Spartan II (FPGA BOARD)	600
2 XST-1 XStend V1.3.2 Prototyping Extended Board	200
MINIDIN-6 PS/2 Male-to-Male Cable	20
Laptop (Donated by Rhys dela Cruz)	N/A
Lab Equipment (Provided by Stevens Institute)	N/A
<u>Miscellaneous</u>	<u>80</u>
Subtotal	\$ 900
<i>Research</i>	
Internet Costs	0.00
Books	0.00
IEEE Membership	0.00
<u>Magazines</u>	<u>0.00</u>
Subtotal	\$ 0.00
<i>Labor</i>	
Research by Group	0.00
Design by Group	0.00
Assembly by Group	0.00
<u>Testing by Group</u>	<u>0.00</u>
Subtotal	0.00
<i>Documentation</i>	
<u>Reports by Group</u>	<u>0.00</u>
Subtotal	\$ 0.00
<i>Overhead</i>	
Rent (Dorm/School Wireless Lab)	0.00
Utilities	0.00
<u>Miscellaneous</u>	<u>0.00</u>
Subtotal	\$ 0.00
Total Project Cost	\$ 900

II - 5.) Project Schedule

Steps	Duration	Start Date	End Date
Define the Project	32d	Wed 8/28/02	Thu 10/10/02
Identify the Product	32d	Wed 9/4/02	Thu 10/17/02
Order and Receive	12d	Wed 10/16/02	Thu 10/31/02
Research PSK	24d	Wed 10/9/02	Mon 11/11/02
Generate Source Code	15d	Tue 11/5/02	Mon 11/25/02
Test PSK	15d	Mon 11/25/02	Sun 12/15/02
Research Chaotic SS	45d	Mon 12/16/02	Fri 2/14/03
Generate Chaotic Code	20d	Mon 2/17/03	Fri 3/14/03
Test Chaotic SS	35d	Mon 3/17/03	Fri 5/2/03
Project Proposal	12d	Mon 9/30/02	Tue 10/15/02
Proposal Report	22d	Fri 11/15/02	Mon 12/16/02
Proposal Presentation	11d	Mon 12/2/02	Mon 12/16/02
Design Report	23d	Mon 3/31/03	Wed 4/30/03
Final Presentation	11d	Wed 4/16/03	Wed 4/30/03
Learn VHDL	76d	Mon 9/2/02	Mon 12/16/02

These are the steps that have to be taken to complete the project. It is similar to the approach; however, it is more detailed. A Gantt chart can be seen in Appendix 1. These steps have to be accomplished in order for the whole project to be successful. The Gantt chart shows our progress as the semester passed by. The Spring Semester is also included in this chart.

Along with the Gantt chart, there will be an LRC Chart. This chart breaks down each group member's tasks for this project. This chart can be seen in Appendix 2. This chart shows how each member has a particular task. Each group member has a particular responsibility. With that responsibility, the individual has other group members assisting him. Sections of the project are cut into various parts. Each member has a primary responsibility.

III.) Conclusion

The idea of this project is to arrive at a test set for chaotic spread spectrum by using an FPGA Board. As students, we intend to have a more hands on approach to engineering problems. We expect to have an operational testing bed with all the programs and codes that are needed for this project. From the engineering work described above, we will learn valuable knowledge as well as gain hands on experience. Once the approach of the project is successfully implemented, testing other various codes that once was used only in Matlab can be tested in the testing bed. Users can actually use the test bed to work out Matlab projects.

By using the Xilinx WebPACK, users can also work out more complicated problems through this test bed. The test bed will be a great addition to any institution that does various testing. We would also expect to have a better homeland security system once the project is complete. By using chaotic theory, it will become harder to break encrypted messages. This test bed will lay the foundation for better, more secure homeland security system. Once completed it will also be the first test bed that uses FPGA and VHDL with Chaotic Spread Spectrum in Stevens Institute of Technology.

IV.) References

Books:

Haykin, Simon. Communication Systems. New York: John Wiley & Sons Inc., 2001

Dornan, Andy. The Essential Guide to Wireless Communications Applications.
New Jersey: Prentice Hall PTR, 2001

Dodd, Annabel Z. The Essential Guide to Telecommunications.
New Jersey: Prentice Hall PTR, 2000

Peterson, Roger L., Rodger E. Ziemer and David E. Borth. Introduction to Spread
Spectrum Communications. New Jersey: Prentice Hall PTR, 1995

Pahlavan, Kaveh and Prashant Krishnamurthy. Principles of Wireless Networking.
New Jersey: Prentice Hall PTR, 2002

Bhasker, J. A VHDL Primer. New Jersey: Prentice Hall PTR, 1992

Rappaport, Theodore S. Wireless Communications Principles and Practices.
New Jersey: Prentice Hall PTR, 2002

Electronic Sources:

www.xess.com

www.xilinx.com

V.) Appendix

1.) Gantt Chart

2.) LRC Chart