Hardware Implementation of Image Steganography Algorithm for Security Purposes

A thesis
submitted in partial fulfillment of the requirements
for the award of degree
of

Master of Technology

in

VLSI Design

Submitted By

Ajay Kumar

Roll No. 601261004

Under the Supervision of

Mrs. Manu Bansal

Assistant Professor, ECED

Thapar University, Patiala

Department of Electronics & Communication Engineering

Thapar University, Patiala-147004

DECLARATION

I hereby certify that the work which is being presented in the thesis entitled, "**Hardware Implementation of Image Steganography Algorithm for Security Purposes**", submitted by me in partial fulfillment of the requirements for the award of degree of Master of Technology in VLSI Design at Thapar University, Patiala, is an authentic record of my own work carried out under the supervision and guidance of Mrs. Manu Bansal.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other University.

Ajay Kumar

Date: 09/07/2014

Ajay Kumar
Roll No: 601261004

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Manu Bansal 9/7/14

Mrs. Manu Bansal
Assistant Professor
ECED, Thapar University
Patiala- 147004

Counter Signed by:

Dr. Sanjay Sharma
Head
ECED
Thapar University,
Patiala-147004

Dr. S.K. Mohapatra
Dean of Academic Affairs
Thapar University,
Patiala-147004

ACKNOWLEDGEMENT

I wish to express my deep gratitude to **Mrs. Manu Bansal, Assistant Professor**, Electronics & Communication Engineering Department for providing her uncanny guidance and support throughout the preparation of the thesis report.

I am also thankful to **Head of the Department, Dr. Sanjay Sharma and PG coordinator, Dr. Kulbir Singh** of Electronics & Communication Engineering Department, for their motivation and inspiration that triggered me for the thesis work.

I would also like to thank all the staff members and my co-students who were always there at the need of the hour and provided with all the help and facilities, which I required for the completion of the thesis.

At last but not the least I would like to thank God for not letting me down at the time of crisis and showing me the silver lining in the dark clouds.

Ajay Kumar
Roll no. 601261004

ABSTRACT

With the advancement of technology, the threats dealt by user have increased exponentially. Hence security of data is required during storage and transmission of data. Image Steganography is best popular techniques now a day. Steganography is a technique of hiding information in some other media like in images, video, text, audio. Image are the popular medium for hiding information because in image different planes available with very little variation of pixel values.

In this thesis work three algorithms are proposed for security of data. In the first algorithm Negative of original image is hidden in cover image for security purposes. During implementation, pixels of negative image are to be substituted in the cover image pixels. In this Even Odd algorithm is proposed for hiding image in the cover image using existing modified LSB algorithm. The second algorithm calculates the Mean Square Error (MSE), Peak Signal Noise Ratio (PSNR) using XORing algorithm and compare with the existing Steganography algorithms in MATLAB. In the third algorithm, first the encrypted data is generated by determining their class. After that encrypted data is hidden in random bits of cover image using XORing method. By doing XORing the probability of pixel variation reduces as compared to replacing method as done in LSB or Modified LSB method.

In this thesis hardware implementation of third proposed Steganography Algorithm is done on Xilinx ISE Design suite and FPGA Spartan 3E kit.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

LSB	Least Significant Bit
MSE	Mean Square Error
PSNR	Peak Signal to Noise Ratio
VHSIC	Very High Speed Integrated Circuits
VHDL	VHSIC Hardware Description Language
ISE	Integrated Software Environment

CHAPTER 1: INTRODUCTION

1.1 History of Steganography

In histories the Steganography was first used during the Golden Age in Greece. An ancient Greek record describes the Steganography. He practiced of melting wax tablets used for writing messages and then write a message in the underlying wood. The wax then reapplied to the wood, giving the appearance of an unused tablet. The tablet then transported without giving attention to anyone. On the other side the tablet again wax to read the message which write in underlying wood [1].

1.2 What is Steganography?

Steganography is an art of invisible communication by hiding of information in some another media. The Steganography term is derived from the Greek words "*stegos*" meaning "cover" and "*grafia*" meaning "writing" and literally means "Cover writing". The Steganography systems consist of three elements:

1. Cover Object
2. The Secret Message
3. The Stego Object

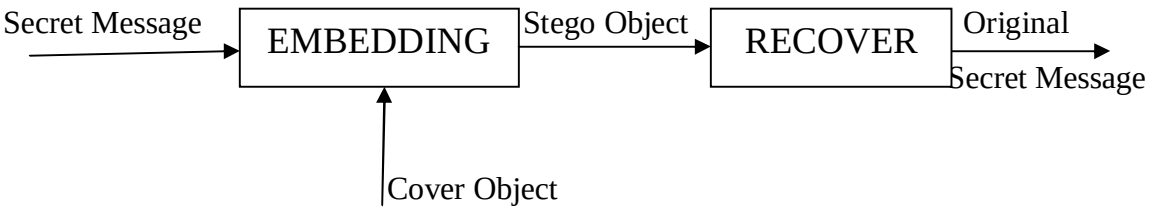


Figure 1.1 BLOCK DIAGRAM OF STEGANOGRAPHY [2]

1. Cover Object

In Steganography the cover objects in which Secret Message is hide. The cover objects are anything like images, audio, videos, text. The most used cover object for hide information is image.

A digital image is described using a 2-D matrix of the color intestines at each grid point (i. e pixels). The gray scale images use 8 bits to describe a pixel on the other hand RGB image uses 24 bits to describe a pixel. In image at each pixels value hide information. Depend upon the image size large no. pixels value available for data hide [3].

2. **The Secret Message**

   In the Steganography the Secret message the message hide in cover image. The Secret message like images, text messages etc.

3. **The Stego Object**

   The Stego object generated after hiding the secret message in cover image. After that Stego Object transmitted. At receiver side processing is done on Stego object to retrieve message from it.

1.3 Need of Steganography over Cryptography

Internet users frequently need to store, send, or receive private information. The most common way to do this is to transform the data into a different form. The resulting data can be understood only by those who know how to return it to its original form. This method of protecting information is known as encryption.

Before the invention of digital means, traditional methods were being used for sending or receiving messages. Before phones, before mail messages were sent on foot. For the messages where privacy was of prime concern, the ways of implementing security were following:

1. Choosing the messenger capable of delivering the message securely.

2. Write the message using such notations that actual meaning of the message was concealed.

3. Hide the message such that even its presence can’t be predicted.

Nowadays Steganography and Cryptography are well known and widely used techniques that manipulate information in order to cipher or hide their existence respectively.

Steganography is the art and science of communicating in a way which hides the existence of the communication. Cryptography scrambles a message so it cannot be understood; the Steganography hides the message so it cannot be seen. Even though both

methods provide security, a study is made to combine both cryptography and Steganography methods into one system for better confidentiality and security.

The distinction between cryptography and steganography is an important one, and is summarized by the following table [3].

Table 1.1

STEGANOGRAPHY	**CRYPTOGRAPHY**
1. Unknown Message Passing.	Known Message Passing.
2. Steganography does not alter the structure of original message.	Cryptography alters the structure of original message.
3. Steganography technology still being developed for certain formats.	Most of the Cryptography algorithm known by all.
4. Steganography prevents discovery of the very existence of communication.	Encryption prevents an unauthorized party from discovery of the contents of a communication.
5. Steganography is a little known technology.	Cryptography is a well known technology.

1.4 Characterization of Steganography

In Steganography techniques a message embed inside a cover image. Various features characterize the strength and weaknesses of a method.

1.4.1 Capacity

The capacity in data hiding indicates the total number of bits hidden and successfully recovered by the Stego system.

1.4.2 Robustness

Robustness refers to the ability of the embedded data to remain intact if the system undergoes transformation like linear and non linear filtering, addition of random noise, rotation, scaling and compression.

1.4.3 Undetectable

The embedded algorithm is undetectable if the image with the embedded message is consistent with a model of the source from which images is drawn. Undetectability is directly affected by the size of the secret message and the format of the content of the cover image.

1.4.4 Invisibility (Perceptual Transparency)

The concept of Invisibility based on the properties of the human visual system. The embedded information is imperceptible if an average human is unable to distinguish between carriers that contain hidden information and others do not. It is important that the embedding occurs without a significant degradation or loss of perceptual quality of the cover.

1.4.5 Security

The embedded algorithm is to be secure if the embedded information is not subject to removal after being discovered by the attacker and it depends on the total information about the embedded algorithm and secret key [4].

1.5 Application of Steganography

1.5.1 Steganography is used for sends information without being censored and without being the fear the information is intercepted and traced back to us.

1.5.2 It is also to use to store information on a location for security purposes. For example, several information sources like some military secrets, banking information, can be stored in a cover image. When we are required to unhide the information from cover image, we can easily reveal the information from cover image.

1.5.3 Steganography has also interesting application in E-commerce. In E-commerce transactions, most users are protected by a username and password, with no real method of verifying that the user. Biometric finger print scanning combined with unique sessions IDs embedded into fingerprint images via steganography, allow for a very secure E-commerce transactions.

1.5.4 Steganography can be used for secure communications to transfer data from one place to other by hiding data in some cover media. Governments are interested in two types of hidden communications: those that support national security and those that do not. Digital steganography provides vast potential for both types. Businesses may have similar concerns regarding trade secrets or new product information.

1.5.5 The transportation of sensitive data is another key use of steganography. A potential problem with cryptography is that eavesdroppers know they have an encrypted message when they see one. Steganography allows to transport of sensitive data past eavesdroppers without them knowing any sensitive data has passed them. The idea of using steganography in data transportation can be applied to just about any data transportation method, from E-Mail to images on Internet websites [1].

1.6 **Outline of Thesis**

The central idea of this thesis is to develop a Secure Steganography algorithm for communication0. Because our algorithm based on spatial domain, so, Chapter 2 gives brief Introduction to Spatial domain that is needed for Steganography. Chapter 3 introduces many existing image Steganography techniques. These techniques embed data in spatial domain. In chapter 4 focuses the techniques proposed and the implementation details of algorithm. In Chapter 5 focuses on conclusions and future work in steganography.

CHAPTER 2: CATEGORIES OF STEGANOGRAPHY TECHNIQUES

There are two Categories of Steganography Techniques.

1. Spatial Domain Technique
2. Frequency Domain Technique

2.1 Overview about Spatial Domain Technique

In this technique the pixels gray level and their color values directly used for encoding the message bits. This technique is the simplest scheme in terms of embedding and extracting the message. The embedding algorithms are applicable mainly to lossless image compression schemes like Tiff images. The major drawback of this technique is amount of additive noise that creeps in the image which directly affects the Peak Signal to Noise Ratio of the image. The most common algorithm belonging to this class of techniques is the Least Significant Bit (LSB) replacement technique in which the least significant bit of the binary representation of the pixel gray levels is used to represent the message bit.

2.2 Overview about Frequency Domain Technique

This technique is used to encode message bits in the transform domain coefficients of the image. This technique realizes large capacity for embedding the message. The transform domain include

1. Discrete cosine Transform (DCT)
2. Discrete wavelet transform (DWT)
3. Discrete Fourier transform (DFT)

By embedding in the transform domain the hidden data resides in more robust areas, spread across the entire image and provide better resistance against signal processing. The transform domain embedding does not necessarily mean generating the transform coefficients on blocks of size 8×8 as done in JPEG compression techniques. It is possible to design techniques which take the transforms on the whole image [5].

2.3 Difference between Spatial Domain and Frequency Domain

Table 2.1

Spatial Domain	Frequency Domain
1. The Spatial Domain technique manipulates the cover image pixel bit values to embed the secret information.	The transform domain techniques embed the message in the frequency domain of the cover image.
2. Spatial Domain techniques are easily detectable.	The Frequency domain technique are difficult to detect and more complex and require much more computation.
3. In Spatial domain technique PSNR value effected by noise.	Frequency based Steganography has higher peak signal to noise ratio and is more secure.

CHAPTER 3: EXISTING IMAGE STEGANOGRAPHY TECHNIQUE

3.1 Least Significant Bit Technique

One of the simplest and very popular techniques of Steganography is Least Significant Bit Technique. In this technique, least significant bits or bits of pixels are replaced by the bits of the data to be hidden.

Original Cover Image Pixels-

Table 3.1

10101010	11101110	11100000	01010111
00110011	00100010	10000111	11101110

Message to hide-10101100

Bits are replaced in LSB of Cover Image pixels.

Stego Image Pixels-

Table 3.2

1010101**1**	1110111**0**	1110000**1**	0101011**0**
0011001**1**	0010001**1**	1000011**0**	1110111**0**

In this technique 1 byte of data required 8 pixels of cover image to hide the data. So data bytes should be $1/8^{th}$ of the cover image. For Example if 256*256 pixels cover image then 32*32 pixels of data can be hiding.

3.2 Modified LSB Technique

In this technique, in place of single bit 2, 3 or up to 4 bits of data is replaced with cover image pixels from LSB side. But when going from LSB side to MSB side the pixels variation increases dynamically like

3.2.1 **2 Bits** of data replace with cover image so maximum 2^2 =4bits variations.

Original Cover Image Pixels-

Table 3.3

10101010	11101110	11100000	01010111
00110011	00100010	10000111	11101110

Message to hide-10101100

Stego Image Pixels Values-

Table 3.4

10101010	11101110	11100011	01010100
00110011	00100010	10000111	11101110

In this technique 1 byte of data required 4 pixels of cover image to hide the data. So data bytes should be $1/4^{th}$ of the cover image. For Example if 256*256 pixels cover image then 64*64 pixels of data can be hiding.

3.2.2 **4 Bits** of data replace with cover image so maximum 2^4 =16 bits variations.

Original Cover Image Pixels-

Table 3.5

10101010	11101110	11100000	01010111
00110011	00100010	10000111	11101110

Message to hide-10101100

Stego Image Pixels Values-

TABLE 3.6

1010**1010**	1110**1100**	11100000	01010111
00110011	00100010	10000111	11101110

In this technique 1 byte of data required 2 pixels of cover image to hide the data. So data bytes should be $1/2^{th}$ of the cover image. For Example if 256*256 pixels cover image then 128*128 pixels of data can be hiding.

3.3 Random Scan Technique

In this technique random bits of pixels are choose to hide the data. There is a large variation in Stego image because of random bits but High security of data because for unauthorized person to difficult in whom pixel data bit stored.

Table 3.7

10101100	11011011	10100000	00010001
01010101	01001100	00011111	10010010

Message to hide is-11001101

Stego Image Pixels Values-

Table 3.8

10101101	11011011	10100000	00010001
01010111	01101100	00011111	10010110

3.4 Raster Scan Principal Technique

This technique is similar to Raster Scan principal apply for TVs. In this, pixels are stored in first row from left to right then pixels are stored from right to left in second row.

Consider original Cover Pixels Values-

Table 3.9

10101100	11011011	10100000	00010001
01010101	01001100	00011111	10010010

Message to hide is-11001101

Stego Image Pixels Values-

Table 3.10

10101101	11011011	10100000	00010000
01010101	01001100	00011111	10010011

3.5 Color Based Technique

In this Scheme one fixed color is used to hide secret data. Intensity values of this fixed color are converted into binary format and the secret information is hidden in this binary data. For ex. consider a gray scale 8 bit image, having intensity values ranging from 0 to 255. Suppose we have fixed a color, whose intensity value is 155. Binary format of this is- 10011011.

We will find total number of pixels from an image, having the same intensity value. Suppose there are 50 pixels found. Then can hide secret information in these 50 pixels, using any data hiding technique like- LSB etc. This technique can be extending by taking more than one fixed color of pixels, from an image [6].

Chapter 4: Implementation Details and Experimental Results

4.1 Implementation Results

In our Experimental Steganography Techniques 2 Cover images and 2 data images of different sizes are used. These images are shown in Appendix B.

4.1.1 **Embedding Algorithm for Hiding Negative of Image using Steganography Even Odd Algorithm for Security Purposes.**

4.1.1.1 XORing Operation

In this technique for hiding the data in cover image, the XORing of data bits with cover pixels bits is done in LSB side because of this the probability in variation in pixel value of cover image reduces. This technique comes in existence because replacing technique in which data bit replace with cover pixels bits is very common.

Table 4.1

10101100	11011011	10100000	00010001
01010101	01001100	00011111	10010010

Message to hide-11001101

Table 4.2

Cover image	10101100	11011011	10100000	00010001
Data bits	00000**01**	00000**11**	00000**00**	00000**11**
Stego Image	**10101101**	**11011000**	**10100000**	**00010010**
	1 bit variation	3 bit variation	No variation	1 bit variation

This technique has an advantage that even when going from LSB side to MSB side the pixels variation in cover image is very less as compared to existing technique.

4.1.1.2 EVEN ODD TECHNIQUE

In this technique data bits split into even and odd parts. After that on different planes of cover image even and odd part hides using XORing technique. So only

parties which communicate know in which part odd and even part is hide. Also at receiver side the even odd part first extract then splited even odd part of data is combined for getting original data.

Table 4.3

Cover Image Red Plane

10101100	11011011	10100000	00010001
01010101	01001100	00011111	10010010

Cover Image Blue Plane

10101110	11011000	10100101	00010101
01010101	01001100	00011111	10010010

Cover Image Green Plane

10101100	11011011	10100000	00010001
01010101	01001100	00011111	10010010

Message to hide-11000101, 10100101

Table 4.4

Cover Image Red Plane

Cover image	10101100	11011011	10100000	00010001
Data bits	00000**01**	00000**01**	00000**00**	00000**11**
Stego Image	**10101101**	**11011010**	**10100000**	**00010010**
	1 bit variation	1 bit variation	No variation	1 bit variation

Cover Image Blue Plane

Cover image	10101110	11011000	10100101	00010101
Data bits	00000**01**	00000**01**	00000**10**	00000**10**
Stego Image	**10101111**	**11011001**	**10100111**	**00010111**
	1 bit variation	1 bit variation	2 bit variation	2 bit variation

Cover Image Green Plane

10101100	11011011	10100000	00010001
01010101	01001100	00011111	10010010

In this Algorithm, for Image hiding is being used existing Image Steganography Algorithm [7, 8].

a) LSB Algorithm
b) Modified LSB Algorithm

Using this Existing algorithm even odd algorithm is proposed.

A. **Even Odd Algorithm**

a) Read Cover image.
b) Obtain the pixel values of cover image and break the pixels value into three planes red plane, Green Plane, Blue Plane.
c) Choose image for hiding.
d) The hiding image size should be $1/8^{th}$ of the original image in using LSB algorithm.
e) The hiding image size should be $1/4^{th}$ of the original image in using modified LSB algorithm.
f) Obtain the negative of image (which used for hiding) and obtain the pixels value and break into three planes.
g) Break each planes pixel value into even and odd part of pixels.
h) Hide the even odd pixels values in different planes of cover image using existing image steganography algorithm like LSB, modified LSB Algorithm.
i) For hiding the image in cover image xor method is used. so
j) After hiding the Stego image is transmitted in communication media.

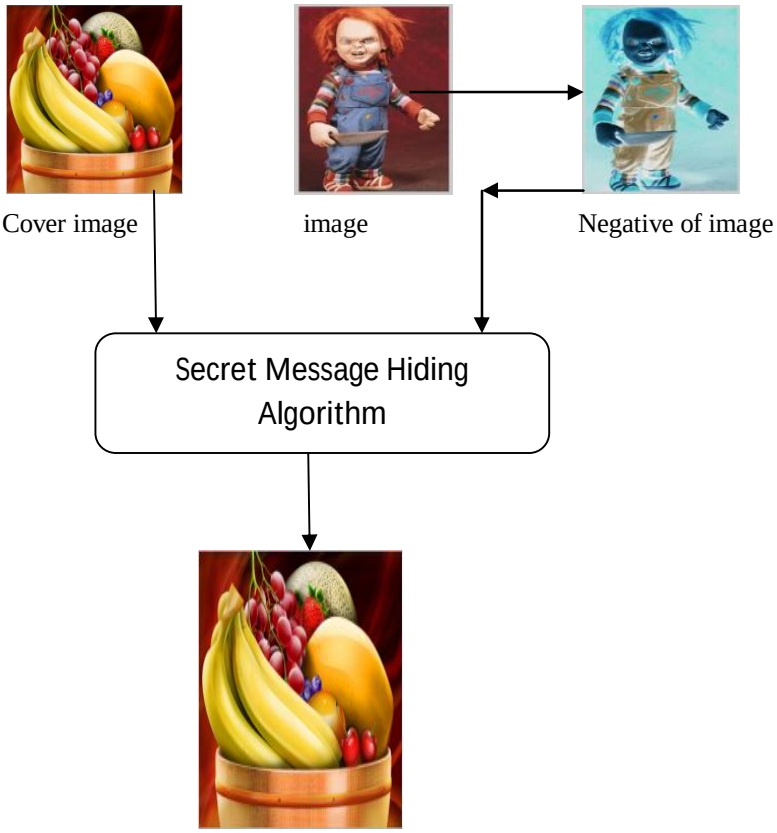


Figure 4.1: BLOCK DIAGRAM FOR HIDING NEGATIVE OF IMAGE USING STEGANOGRAPHY EVEN ODD ALGORITHM

4.1.2 Extracting Algorithm for Hiding Negative of Image using Steganography Even Odd Algorithm for Security Purposes.

1. Receive the Stego image.

2. Extract the different planes of Stego image and Cover image.

3. XORing the Stego image planes with cover image planes.

4. Get the data bits in parts.

5. Combine the data bits after that combine even odd part of data image.

6. Generate the negative of that data image by doing this original data image get.

4.1.3 Embedding Algorithm for Even Odd Algorithm in Image Steganography for Text Data Hiding.

a) Obtain the pixels values of cover image.

b) Let Text Message to hide: “image steganography”

c) Break data into even odd part.

d) Hide the even part of data in one plane and odd part of data in other plane using existing Image Steganography algorithm [7, 8]. Like

a) LSB algorithm

b) Modify LSB algorithm.

e) In place of replacing XORing of text data with cover image pixels bits is done.

4.1.4 Embedding Algorithm for Improving MSE, PSNR using XORing Algorithm in Steganography.

For improving MSE, PSNR using XORing Algorithm following steps are taken

a) Read the cover image.
b) Extract different planes of cover image.
c) Read the data and convert into binary format.
d) XORing of cover image plane with data bits.
e) Mean square error (MSE) is calculated by comparing the stego image with cover image.
f) Peak Signal Noise Ratio (PSNR) is calculated from MSE.
g) Determine the execution time of whole algorithm.
h) After hiding data Stego image is transmitted.
i) At the receiver side XORing of cover image and Stego image is done to extract the original data bits from stego image.

We measure the quality of Steganography images in terms of PSNR (Peak Signal to Noise Ratio) and MSE (Mean Square Error). In ideal case PSNR should be

infinite and MSE should be zero. But it is not possible for Steganography image. So, large PSNR and small MSE are desirable.

1. Mean Squared Error (MSE) is computed by performing byte by byte comparisons of the cover image and stego image. The Computation expressed as

$$\text{MSE}=\frac{1}{M*N}\sum_1^M \sum_1^N (Fij - Gij)^2$$

M: numbers of rows of cover image

N: number of column of Cover Image

Fij: Pixel value from cover image

Gij: Pixel value from Stego Image

Higher value of MSE indicates dissimilarity between Cover image and Stego image.

2. Peak signal to noise ratio (PSNR) measures in decibels the quality of the stego image compared with the cover image. The higher the PSNR better the quality. PSNR is computed using the following equation.

$$\text{PSNR}=20\log_{10} 255 - 10\log_{10} MSE$$

COVER IMAGE
(512*512)

SECRET DATA
(64*64)

STEGO IMAGE
(512*512)

Figure 4.2: 1 bit method using XORing Algorithm

COVER IMAGE
(512*512)

SECRET DATA
(128*128)

STEGO IMAGE
(512*512)

Figure 4.3: 2 bit method using XORing Algorithm

COVER IMAGE
(512*512)

SECRET DATA
(256*256)

STEGO IMAGE
(512*512)

Figure 4.4: 4 bit method using XORing Algorithm

COVER IMAGE
(512*512)

SECRET DATA
(512*512)

STEGO IMAGE
(512*512)

Figure 4.5: 8 bit method using XORing Algorithm

Table 4.5: Comparison Table for MSE

n-bit LSB	MSE for Existing Algorithm[7]	MSE for Proposed Algorithm
1 bit	0.5	0.057
2 bit	2.5	0.29
4 bit	42.7	17.346
8 bit	8640	4789.09

Table 4.6 Comparison Table for PSNR

n- bit LSB	PSNR for Existing Algorithm (dB) [7]	PSNR for XORing Algorithm (dB)
1 bit	51.1	60.57
2 bit	44.1	53.50
4 bit	31.8	35.73
8 bit	8.6	11.32

Examining the result in table, we can make following observations.

1. The Error metrics MSE increases quickly with increases n bit LSB.
2. The image quality indicator PSNR is above 50dB for 1-bit and 2-bit steganography. For images and videos, PSNR ratio up to 30 dB acceptable. Clearly selecting the appropriate n-bit LSB method should carefully balance trade-offs between capacity (i.e. Secret message size) and imperceptibility (i.e. image distortion)

1.1.5 Embedding Algorithm for Random Scan XORing Algorithm for Image Steganography in Scilab

1.1.7.1 RANDOM SCAN TECHNIQUE

In this technique the data byte split into bits after that this bits hide in cover image but at random places using XORing technique. Because of this the probability of variation at random places is very less as compared to existing Random Scan method where cover image pixels bits replaces with data bits.

Table 4.7

10101100	11011011	10100000	00010001
01010101	01001100	00011111	10010010

Message to hide-11000101

Table 4.8

Cover image	10101100	11011011	10100000	00010001
Data bits	00000**01**	0000**01**00	00**00**0000	**11**000000
Stego Image	**10101101**	**11011111**	**10100000**	**11010001**

Algorithm

1. Read the cover image.
2. Read the message.
3. Generate the encrypted message by determining their class like ASCII or EBCDIC type.
4. Read the Encrypted message and convert into binary format.

5. Hide the Encrypted data in Random bits of cover image by doing XORing of cover image with data bits.
6. After hiding data Stego image is transmitted.

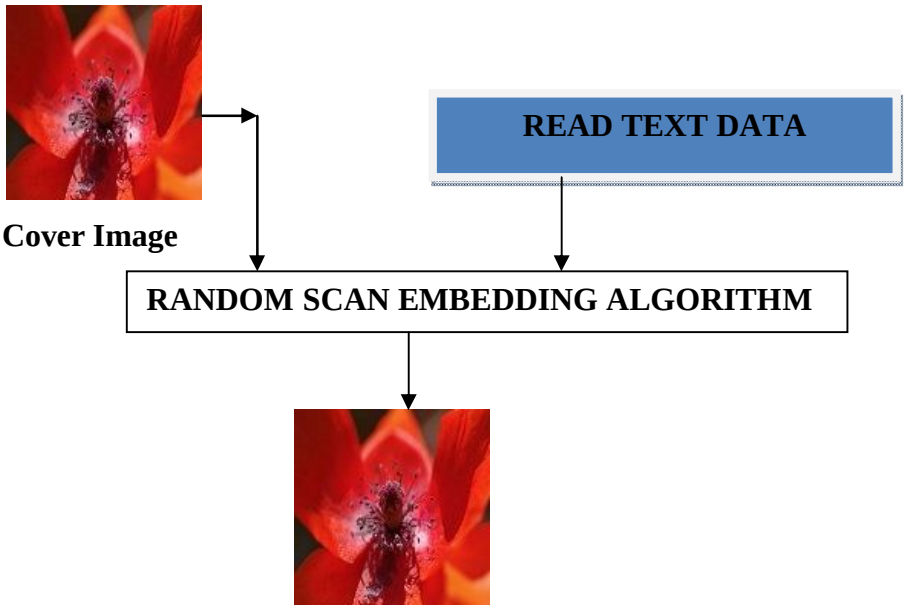


FIGURE 4.6: BLOCK DIAGRAM FOR RANDOM SCAN METHOD

1.1.6 Embedding Algorithm for Hardware implementation of Steganography Technique

For hardware implementation of image Steganography Techniques following steps are taken

1. Read cover image and data in MATLAB.
2. Convert Cover image pixels and data into text file.
3. Read the text file of Cover image and data in VHDL.
4. Break the each byte of data into Random bits.
5. Hide the data bits in cover image using XORing Method.
6. Stego image pixels generated at transmitter side.

4.1.7 Extracting Algorithm for Hardware Implementation of Steganography Technique.

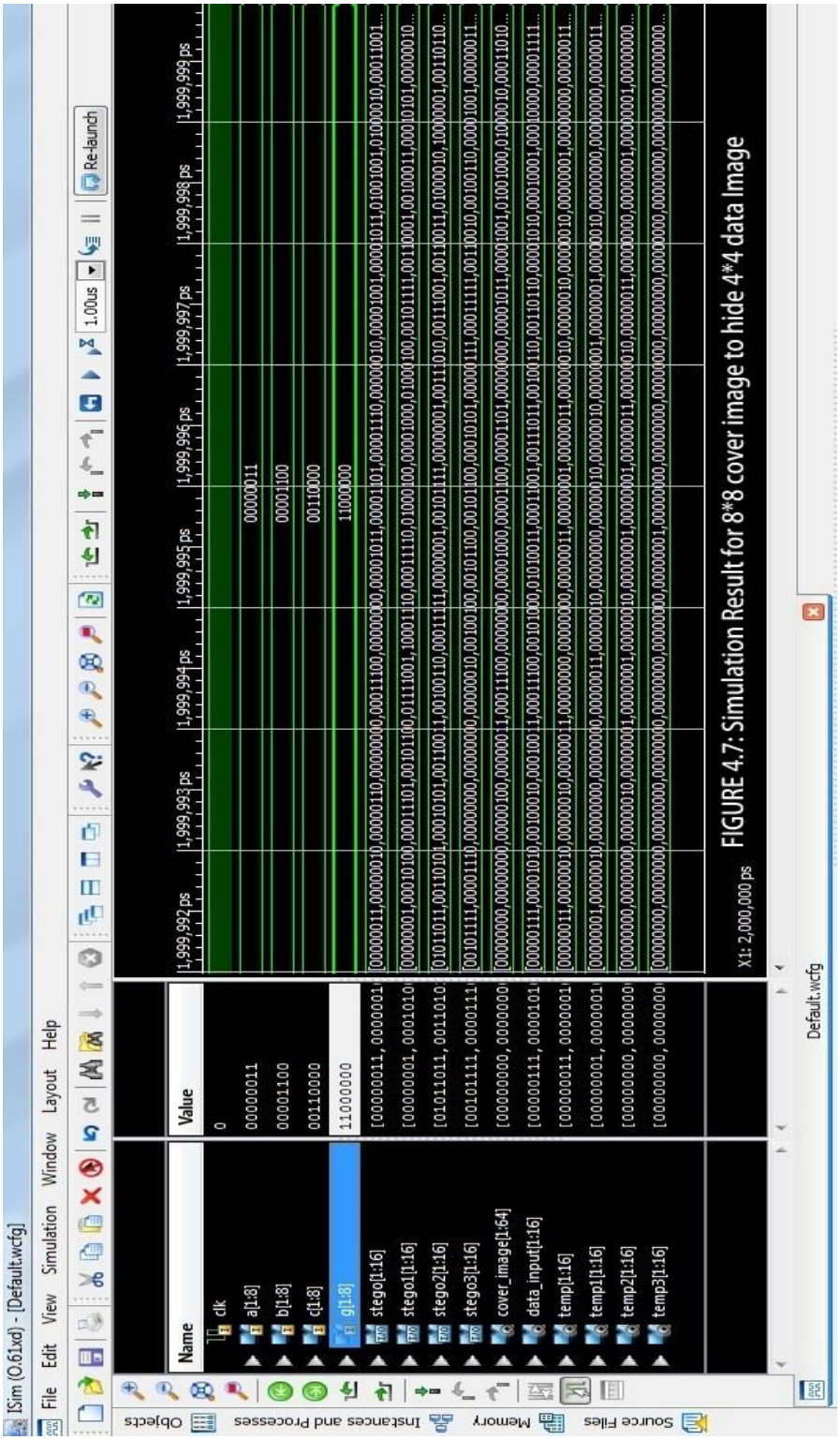
1. Read the text file of Stego image.

2. Xoring of Stego image done with cover image after doing these random bits of data is generated.

3. Then combine the data bits to obtain the original data.

4.2 Simulation and Synthesizable Results

4.2.1 Simulation Results for proposed Algorithm in Xilinx ISE Design Suite.



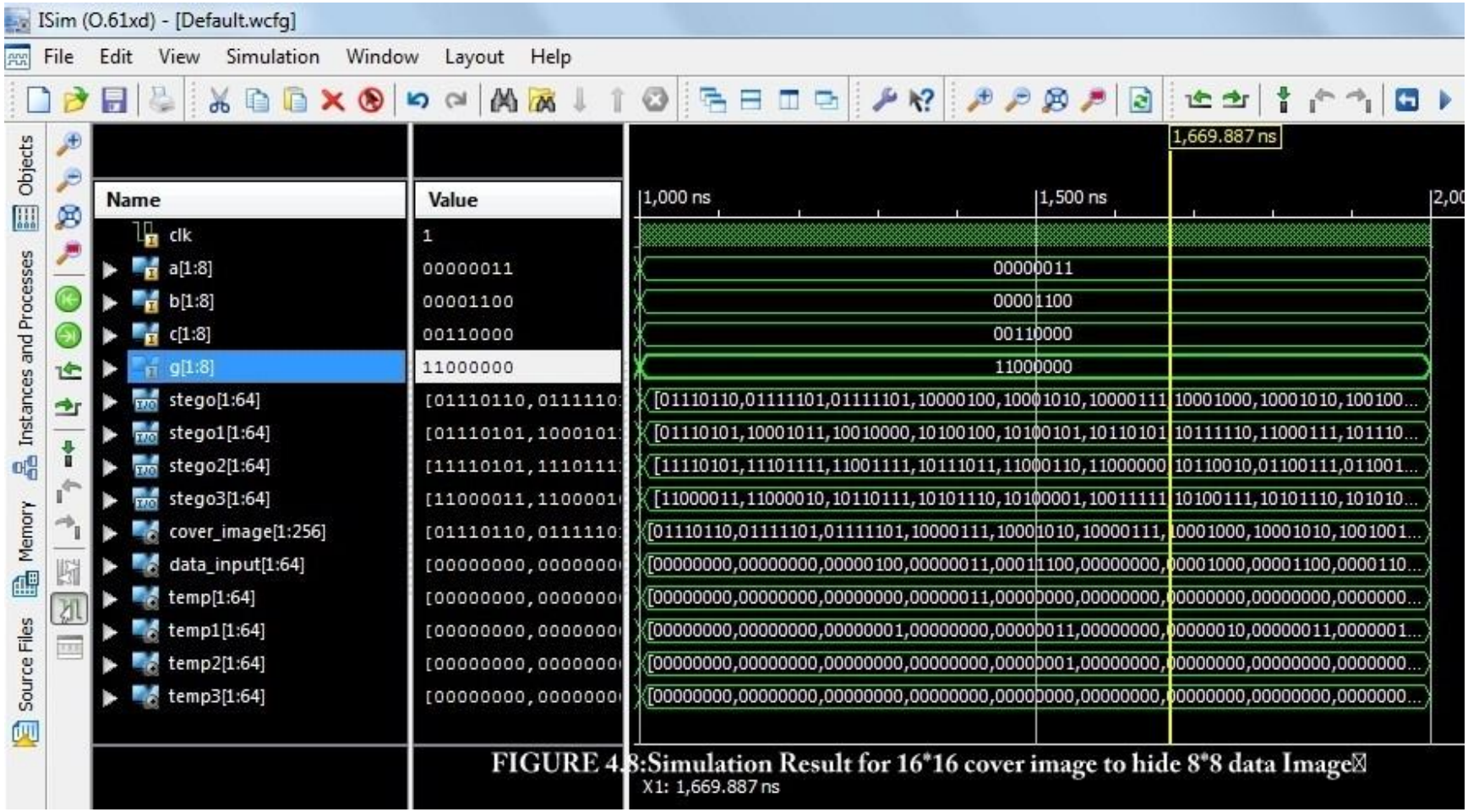


FIGURE 4.8:Simulation Result for 16*16 cover image to hide 8*8 data Image

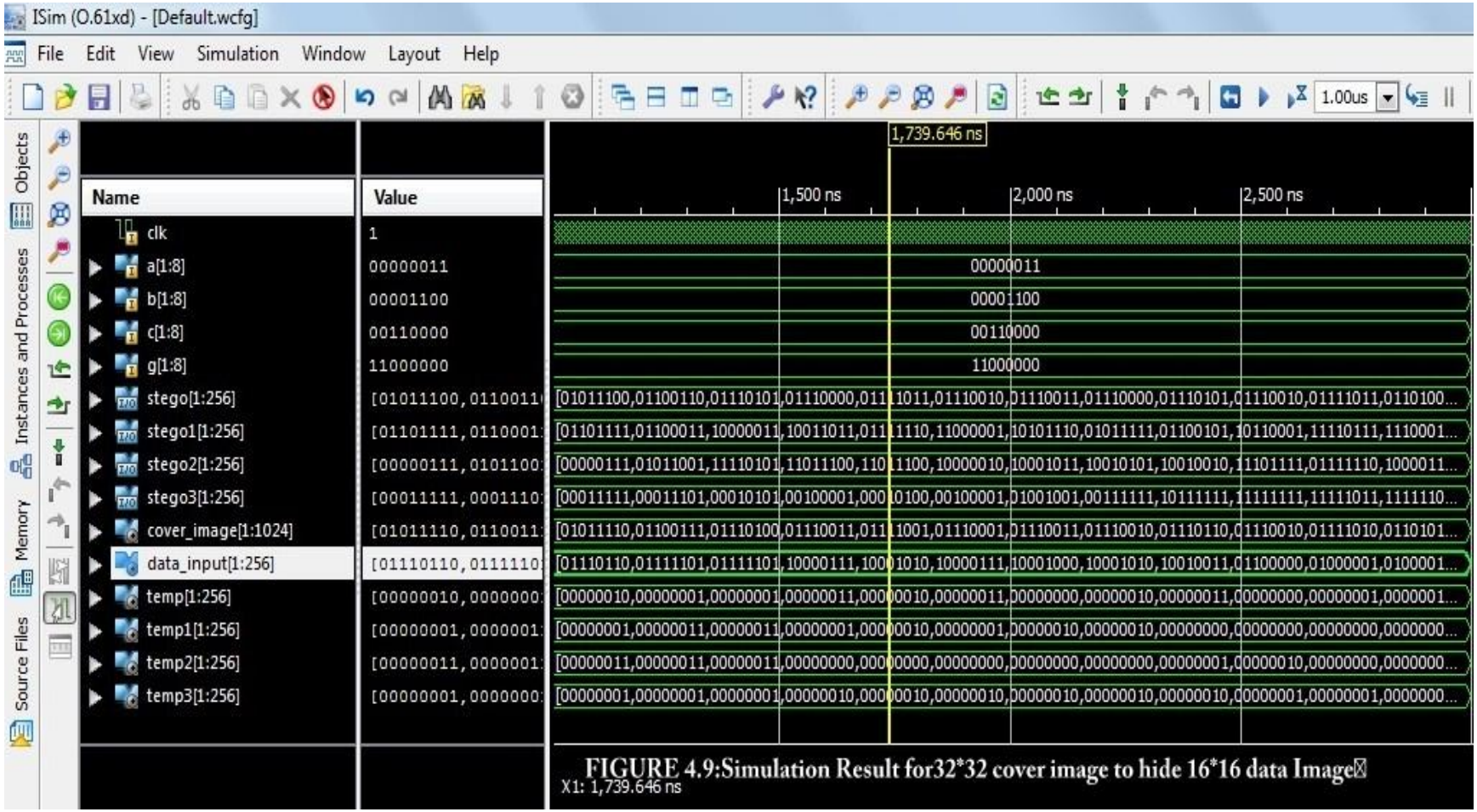


FIGURE 4.9:Simulation Result for32*32 cover image to hide 16*16 data Image

4.1.1 Power Analysis for 8*8 Cover image for 4*4 data image

FIGURE 4.10: POWER ANALYSIS

Synthesizable Result for Steganography Algorithm

4.2.2.1 HDL Synthesis Report for Spartan 3E

Table 4.9

Macro Statistics	For 32*32 Cover Image	For 16*16 Cover Image	For 8*8 Cover Image
# Registers	2048	512	128
# Xors	1024	256	64

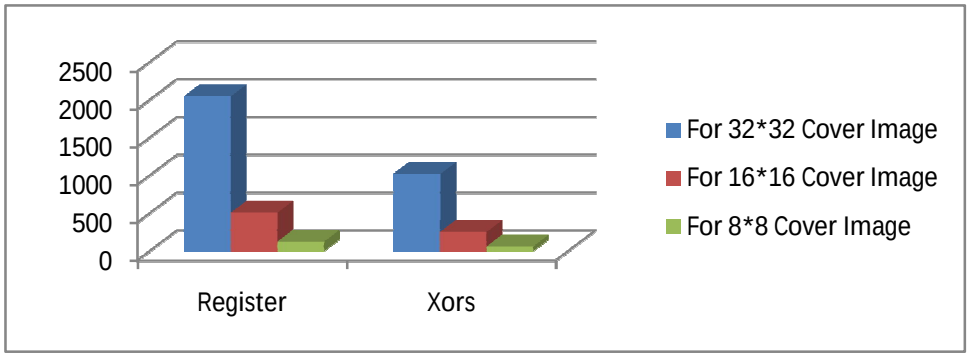


Figure 4.10 HDL Synthesis Report for Spartan 3E

4.2.2.2 Timing Summary for spartan3E:

Table 4.10

Parameters	For 32*32 Cover Image	For 16*16 Cover Image	For 8*8 Cover Image
Minimum period	2.470ns (Maximum Frequency: 404.858MHz)	2.470ns (Maximum Frequency: 404.858MHz)	2.470ns (Maximum Frequency: 404.858MHz)
Minimum input arrival time before clock	1.946ns	1.946ns	1.946ns
Maximum output required time after clock	5.273ns	5.273ns	5.201ns

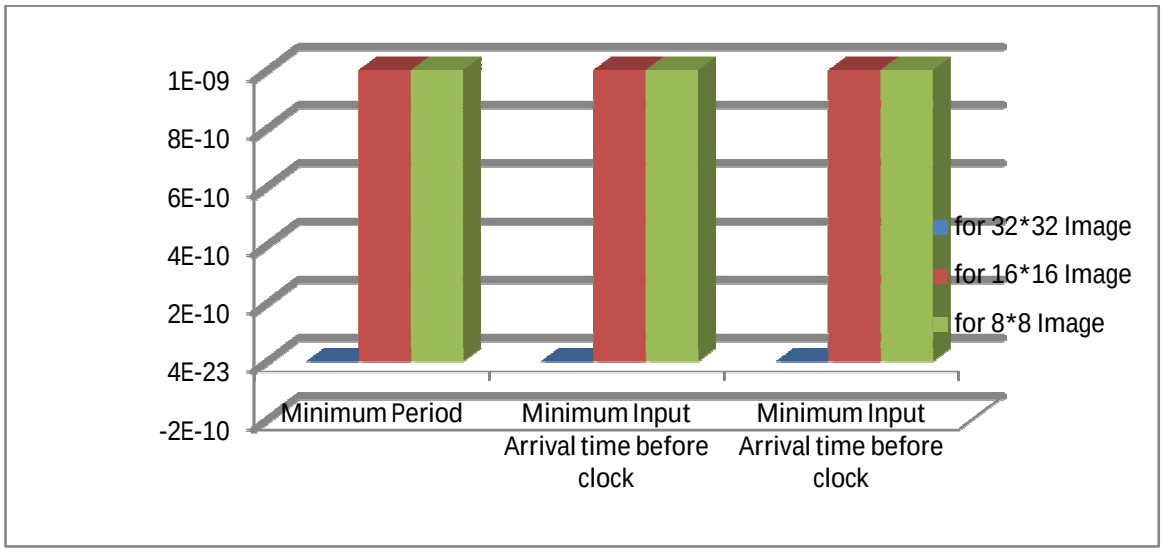


Figure 4.11 Timing Summary for spartan3E

4.2.2.3 Device utilization summary:

Table 4.11

Number of Slices	29 out of 4656 0%	29 out of 4656 0%	24 out of 4656 0%
Number of Slice Flip Flops	45 out of 9312 0%	44 out of 9312 0%	37 out of 9312 0%
Number of 4 input LUTs	15 out of 9312 0%	15 out of 9312 0%	11 out of 9312 0%
Number of IOs	8225	2081	545
Number of bonded IOBs	8213 out of 232 3540% (*)	2069 out of 232 891% (*)	529 out of 232 228% (*)
IOB Flip Flops	6	6	4
Number of GCLKs	1 out of 24 4%	1 out of 24 4%	1 out of 24 4%

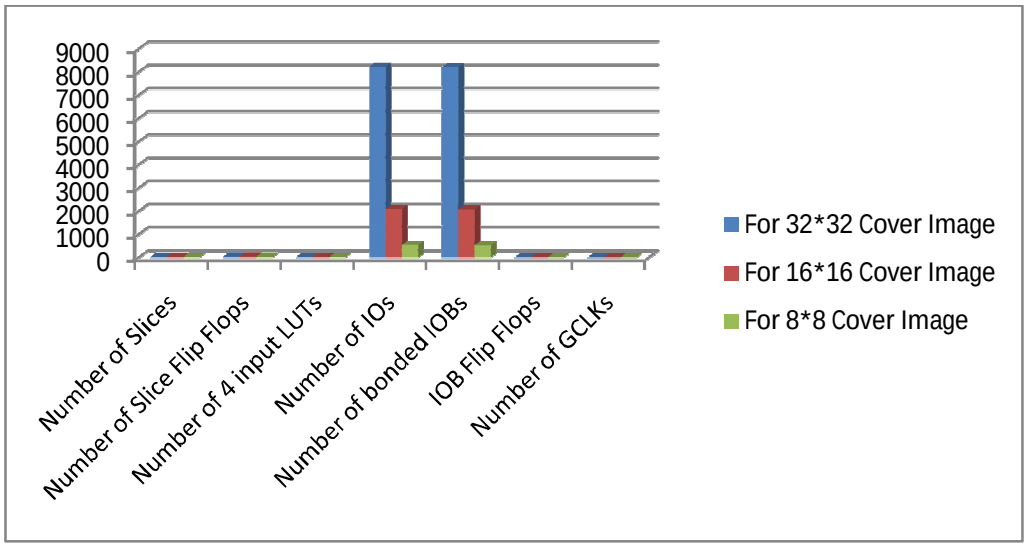


Figure 4.12 Device utilization summaries

Chapter 5: Conclusions and Future Scope of work

5.1 Conclusions

Three different Image Steganography algorithms have been simulated on different tools like MATLAB and Scilab. Hardware implementation of one of the above three Steganography algorithm has been done on Spartan 3E kit using Xilinx ISE Design Suite.

For encryption, XOR operation has been applied on Even Odd Method and Random Scan Method. For embedding data in cover image spatial domain has been used. Proposed Steganography algorithms have the following advantages.

1. Because of XOR operation there is less change in Stego image as compared to replacing method.
2. Improved MSE, PSNR for Stego Image.
3. Random Scan method has high security of data as compared to LSB and Modified LSB method.

5.2 Future Scope of work

Steganography is an emerging research area for protection and authentication for communication. Most of the research is going on this field.

1. The proposed Steganography algorithms can be improved using other techniques to increase the hiding capacity without affecting the imperceptibility of the images.
2. For hiding the data, images are taken as a Cover media but other Medias like video, text, audio can also be used as cover media.
3. Other Arithmetic operations can be applied to steganography algorithms for data hiding.

Reference

[1] Arvind Kumar and Km Pooja, "Steganography- A hiding Technique" *International Journal of Computer Applications*, vol 9, no. 7, November 2010.

[2] Himanshu Gupta , Prof Ritesh Kumar and Dr. Soni Changlani "Enhanced Data Hiding Capacity using LSB-Image Steganography Method" *International Journal of Emerging* Technology *and Advanced Engineering"* vol 3,June 2013.

[3] A. Joseph Raphael and Dr. V. Sundaram "Cryptography and Steganography –an Survey" *International Journal of Computer Technology Application*, vol 2, pp. 626-630, 2012.

[4] Zaidoon Kh. AL-Ani, A.A.Zaidan, B.B.Zaidan and Hamdan.O.Alanazi "Overview-Main Fundamental for Steganography" *Journal of Computing*, vol 2, March 2010.

[5] Ms.G.S.Sravanthi, Mrs.B.Sunitha Devi, S.M.Riyazoddin and M.Janga Reddy "A Spatial Domain Image Steganography Technique Based on Plane Bit Substitution Method" *Global Journal of Computer Science and Technology Graphics & Vision*, vol 12, 2012.

[6] Dr. Diwedi Samidha and Dipesh Agrawal, "Random Image Steganography in Spatial Domain", *International Journal of Computer Science and Information Security*, vol 7, pp. 3, March 2013.

[7] Bassam jamil Mohd,saed Abed and Thaier Al-Haneh,sahel Alouneh, "FPGA hardware of the LSB Steganography method" *International Conference on Computer, Information and Telecommunication Systems (CITS)*, pp 1-4, 2012.

[8] V.Lokeswara Reddy, Dr. A. Subramanyam and Dr. P. Chenna Reddy, "Implementation of LSB Steganography and its evaluation for various file formats" *Int. J. Advanced Networking and Application*,vol. 2,pp. 868-872,2011

[9] Youssef Bassil "Image Steganography Method based on Brightness Adjustment" *Advances in Computer Science and Application (ACSA)*, vol. 2, pp. 2, 2012.

[10] T. Morkel, J. Eloff and M.Olivier, "An overview of Image Steganography" *The fifth Annual Information Security South Africa Conference (ISSA 2005)*, July 2005.

[11] H.Wang,S. Wang, "Cyber warfare: Steganography vs Steganalysis", *Communications of the ACM*, vol. 47, no. 10, pp. 76-82, October 2004.

[12] Mohammad Tanvir Parvez and Adnan Abdul-Aziz Gutub "RGB intersity based variable bits Image Steganography", *IEEE Asia Pacific Services Computing Confernece*,pp 1322-1327, 2008.

[13] Ankita Ganorkar and Sujata Agrawal "Releaving the Hidden Secret with LSB Steganography" *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering*, vol. 1, Issue 3, June 2013.

[14] Deshpande Neeta, Kamalapur Snehal and Daisy Jacobs "Implementation of LSB Steganography and its evaluation for various bits" *IEEE 1st International Conference on Digital Information Management*, India, pp.173-178, December 2006.

[15] M.M Amin, M. Salleh, S. Lbrahim,M.R.K Atmin and M.Z.I Shamsuddin,"information hiding using steganography*", IEEE 4th National Conference on Telecommunication Technology Proceeding, Shah Alam.Malaysia*, pp. 21-25, January 2003.

[16] *www.math.utah.edu/~wright/misc/matlab/matlabintro.html*

[17] *http://en.wikipedia.org/wiki/Scilab*

[18]*www.engrcs.com/tools_programs/Matlab-Scilab-Basic-Diff.pdf*

[19] Xilinx ISE Design Suite-Help

[20] *pratikchauhan.files.wordpress.com/2014/.../vhdl-primer-by-j-bhasker.pdf*

Appendix A: Development Environment for Steganography

For Implementation of Steganography Algorithm the following tools used:

1. **MATLAB**
   1.1 **Typical use of MATLAB**
   1.2 **Advantages of MATLAB**
2. **SCILAB**
   2.1 **Overview about Scilab**
   2.2 **Difference between in Scilab and MATLAB**
3. **Xilinx ISE Design Suite for Hardware Implementation**
   3.1 **Overview about Xilinx ISE Design Suite**
   3.2 **Flow Chart in ISE Design Suite**
   3.3 **Overview about VHDL**

1. MATLAB

MATLAB is widely used in all areas of applied mathematics, in education and research at universities, and in the industry. MATLAB stands for MATrix LABoratory and the software is built up around vectors and matrices. This makes the software particularly useful for linear algebra but MATLAB is also a great tool for solving algebraic and differential equations and for numerical integration. MATLAB has powerful graphic tools and can produce nice pictures in both 2D and 3D. It is also a programming language, and is one of the easiest programming languages for writing mathematical programs. MATLAB also has some tool boxes useful for signal processing, image processing, optimization, etc [16].

1.1 Typical use of MATLAB

1. Math and computation
2. Algorithm development
3. Data acquisition
4. Modeling, simulation, and prototyping
5. Data analysis, exploration, and visualization
6. Scientific and engineering graphics

7. Application development, including graphical user interface building.

1.2Advantages of MATLAB

1. Ease of Use
- Expression typed at the command window
- Built-in integrated editor/debugger
2. Platform Independent
   - Windows 95/98/ME/NT/2000 & Unix
   - Program written on one platform may run on other
   - Data files written may be used on other platform
3. Predefined Functions
   - Extensive Library of Predefined Functions: Arithmetic mean, Standard Deviation, Median (No need to write subroutines)
   - Special Purpose Toolboxes to solve complex problems
4. Visualization
   - Many plotting and Imaging Commands to visualizing technical data
5. Graphical User Interface
   - To design sophisticated data analysis programs that can be operated by relatively inexperienced users.

2. SCILAB

2.1Overview about Scilab

Scilab is a high-level, numerically oriented programming language. The language provides an interpreted programming environment, with matrices as the main data type. By using matrix-based computation, dynamic typing, and automatic memory management, many numerical problems may be expressed in a reduced number of code lines, as compared to similar solutions using traditional languages, such as Fortran, C, or C++. This allows users to rapidly construct models for a range of mathematical problems. While the language provides simple matrix operations such as multiplication, the Scilab package also provides a library of high-level operations such as correlation and complex

multidimensional arithmetic. The software can be used for signal processing, statistical analysis, image enhancement, fluid dynamics simulations, and numerical optimization.

Scilab also includes a free package called Xcos (based on Scicos) for modeling and simulation of explicit and implicit dynamical systems, including both continuous and discrete sub-systems. Xcos is the open source equivalent to Simulink from the Math Works.

As the syntax of Scilab is similar to MATLAB, Scilab includes a source code translator for assisting the conversion of code from MATLAB to Scilab. Scilab is available free of cost under an open source license. Due to the open source nature of the software, some user contributions have been integrated into the main program [17].

2.2Difference between in Scilab and MATLAB [18]

Table A.1

MATLAB	**SCILAB**
MATLAB® is a high-level language and interactive environment for numerical computation, visualization, and programming.	*Scilab* is free and open source software for numerical computation providing a powerful computing environment for engineering and scientific applications
From File menu open and create new source file. Source file has extension ".m"	From File menu open and create new source file. Source file has extension ".sci"
% when function returns a value function [r]= test(a, b)	// when function returns a value function r= test(a, b)

end % when function does not returns a value function []= test(a, b) end	endfunction // when function does not returns a value function test(a, b) endfunction
Write the function name in command window to run	Execute Menu and load into scilab and then Write the function name in command window to run.
mod(x,y)	Pmodulo (x,y)
fprintf (1,'format string', var1, var2,…)	printf ('format string', var1, var2,…)
% comment	// comment

3. Xilinx ISE Design Suite for Hardware Implementation

4.2 Overview about Xilinx ISE Design Suite

The ISE Design Suite is the Xilinx design environment, which allows you to take your design from design entry to Xilinx device programming. With specific editions for logic, embedded processor, or Digital Signal Processing (DSP) system designers, the ISE Design Suite provides an environment tailored to meet our specific design needs [19].

4.3 Flow Chart in ISE Design Suite

3.2.1 Design Entry Overview

Design entry is the first step in the ISE® design flow. During design entry, you create source files to represent your design. The top-level design source file can be any of the following formats:

1. Hardware Description Language (HDL), such as VHDL or Verilog
2. Schematic (SCH)
3. Embedded processor (XMP)
4. EDIF or NGC/NGO file (if you choose to synthesize your design outside of Project Navigator)

3.2.2 Design Synthesis

After design entry and optional simulation, you run synthesis. The ISE® software includes Xilinx Synthesis Technology (XST), which synthesizes VHDL, Verilog, or mixed language designs to create Xilinx®-specific netlist files known as NGC files. Unlike output from other vendors, which consists of an EDIF file with an associated NCF file, NGC files contain *both* logical design data and constraints. XST places the NGC file in your project directory and the file is accepted as input to the Translate (NGDBuild) step of the Implement Design process.

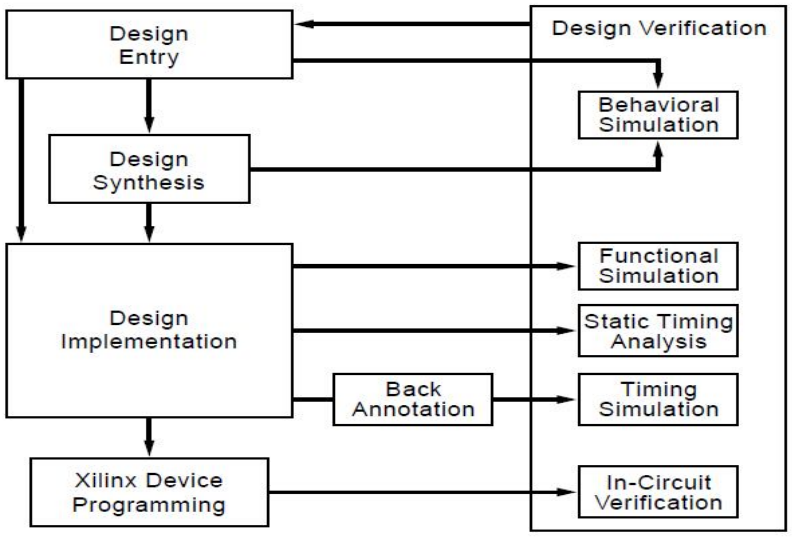


Figure A.1 Flow diagram of Hardware Implementation

3.3.3 Design Implementation

After synthesis, you run design implementation, which comprises the following steps:

1. **Translate -** merges the incoming netlists and constraints into a Xilinx® design file.
2. **Map -** fits the design into the available resources on the target device, and optionally, places the design.
3. **Place and Route -** places and routes the design to the timing constraints.
4. **Generate Programming File -** creates a bit stream file that can be downloaded to the device.

3.3.4 Xilinx Device Programming

After generating a programming file using the **Generate Programming File** process, you can configure your device; create PROM, System ACE™ solution, SVF, XSVF, or STAPL files. You can configure FPGAs or program Xilinx® CPLDs or PROMs in-system, directly from a host-computer using IMPACT with a Xilinx download cable.

3.3 Overview about VHDL

VHDL is an acronym for VHSIC (Very High Speed Integrated Circuits) Hardware Description Language. It is a hardware description language that can be used to model a digital system at many levels of abstraction, ranging from the algorithmic level to gate level. The complexity of the digital system being modeled could vary from that of a simple gate to a complete digital electronic system, or anything in between. The digital system can also be described hierarchically. Timing can also be explicitly modeled in the same description [20].

The VHDL language can be regarded as an integrated amalgamation of the following languages:

1. Sequential Language
2. Concurrent Language
3. Net-list Language
4. Timing Specification
5. Waveform Generation Language

3.3.1 **Capabilities of VHDL**

1. The language can be used as an exchange medium between chip vendors and CAD tool users.
2. The language supports hierarchy; that is , a digital system can be modeled as a set of interconnected components; each component, in turn, can be modeled as a set of interconnected subcomponents.
3. The language supports flexile design methodologies: top down, bottom-up or mixed.
4. It supports both synchronous and asynchronous timing models.
5. The language supports three basic different description styles : structural, dataflow, and behavioral. A design may also be expressed in any combination of these three descriptive styles.
6. Test benches can be written using the same language to test other VHDL models.
7. Nominal propagation delays, min-max delays, setup and hold timing, timing constraints, and spike detection can all be described very naturally in this language.

Appendix B: Cover Images

1. Cover Image used in MATLAB

Figure B.1

2. Cover image used in Scilab

Figure B.2

3. Data Image

Figure B.3

Papers Communicated/Accepted/Published

1. Ajay Kumar, Mrs. Manu Bansal "**Hiding Negative of an Image using Steganography Even Odd Algorithm for Security Purposes**" *IOSR Journal of Computer Engineering*, vol. 16, pp 70-75, Feb 2014.
   (Published in IOSR Journal)
2. Ajay Kumar, Mrs. Manu Bansal "**Even Odd Algorithm in Image Steganography for Text Data Hiding**" *3rd National Conference on Advances in Metrology*, Feb 2014.
   (Published in Conference Proceeding)