A Thesis Report

On

"Design and Development of a Steganographic System based on a Modern, Steganalysis Resistant LSB Algorithm on FPGA"

Dissertation Submitted towards partial fulfilment of requirement for the award of Degree of

Master of Technology
In
VLSI Design

Submitted by:

Kunjan Pathak

Roll No. 601461017

Under the supervision of:

Mrs Manu Bansal

(Assistant Professor, ECED)

Department of Electronics & Communication Engineering

Thapar University, Patiala-147004 (Punjab)

July 2016

CERTIFICATE & DECLARATION

I hereby declare that the work which is being presented in the dissertation titled **"Design and Development of a Steganographic System based on a Modern, Steganalysis Resistant LSB Algorithm on FPGA "** in the partial fulfilment of the requirement for the award of the degree of Master of Technology in VLSI Design submitted in Electronics and Communication Engineering Department of Thapar University, Patiala is an authentic record of my study carried out as under guidance of **Mrs Manu Bansal** (Assistant Professor, ECED) during 2014-2016.

Pathak K.S.

Date: 14-7-16

Pathak Kunjan Sanjaykumar

Roll No: 601461017

It is certified that the above statement made by the student is correct to the best of my knowledge and belief.

Manu Bansal

(MRS MANU BANSAL) 14/7/16

Assistant Professor, ECED

Thapar University,

Patiala-147004

Countersigned by:

Head

ECED, Thapar University,

Patiala-147004

Dean of Academic Affairs

Thapar University,

Patiala-147004

ACKNOWLEDGEMENT

This report is completed with prayers of many and love of my family and friends. However, there are a few people that I would like to specially acknowledge and extend my heartfelt gratitude who have made the completion of this report possible. With the biggest contribution to this report, I would like to thank **Mrs Manu Bansal** had given me her full support in guiding me with stimulating suggestions and encouragement to go ahead in all the time of the report.

I am also thankful to **Dr. Sanjay Sharma**, Professor and Head, Electronics and Communication Engineering Department, for providing us with the adequate infrastructure for carrying out the work.

I am also thankful to **Dr. Amit Kumar Kohli**, P.G. Coordinator, Electronics and Communication Engineering department, for the motivation and inspiration that triggered me for the work.

Lastly, I would also like to thank my parents for their years of unyielding love and encourage. They have always wanted the best for me and I admire their determination and sacrifice.

Kunjan Pathak

ABSTRACT

Steganography is the science and art of covert communication, without letting anyone know that there is any message hidden in the cover object. Practically all sorts of digital multimedia objects can be utilised as cover objects for steganography. But digital images have been the most lucrative option for cover media. In images, spatial domain algorithms are most suitable for hardware implementation due to the less complex nature of operations and ability to provide better performance. Hardware implementations of steganographic algorithms are identified as a major research gap in this work. This thesis work makes an attempt to implement a modern spatial domain steganographic algorithm, which is resistant to many recent steganalysis attacks, on FPGA based hardware. Many operations and elements in original one third probability algorithm have been found either not fully optimised or unsupported for hardware implementation.

This thesis has optimised some of operations. Some of unsupported functionality has been replaced with their synthesizable equivalents. Some basic functions like major change, minor change and LSB functions have been refined for FPGA implementation. Elements like two dimensional packed arrays, floor function and intermediate floating point values have been avoided and their suitable replacements have been reinvented wherever possible. Especially, the structure of change matrix ‘C’ has been thoroughly redesigned. A system level design consisting of image storing RAMs and A FSM for embedding operations has been developed. This design based on modified algorithm has been implemented on Xilinx SP605 board (Spartan 6 series XC6SLX45T FPGA). The design is found out to be working correctly with minimum period of 9.526ns, which translates to the maximum frequency of 104.971 MHz. Resultant stego images, produced by implemented design, have been analysed for their qualitative correctness and quantitative parameters, both of which have been found at par with or marginally better than original algorithm .

Keywords

Steganography, Data security, Spatial domain algorithm, One third probability LSB algorithm, Optimization for hardware, FPGA implementation

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

1. LSB: Least Significant Bit
2. FPGA: Field Programmable Gate Array
3. TCP-IP: Transmission Control Protocol-Internet Protocol
4. HCF-COM: Histogram Characteristic Function-Centre of Mass
5. DCT: Discrete Cosine Transform
6. DFT: Discrete Fourier Transform
7. FSM: Finite State Machine
8. MSE: Mean Square Error
9. PSNR: Peak Signal to Noise Ratio
10. RAM: Random Access Memory

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Definition

The term steganography was first coined by an occultist, namely Trithemius. Steganography, coming from two Greek words *stegos,* meaning roof or covered and *graphia* which means writing, is the art and science of hiding the fact that communication is taking place. [1]

1.2 Historical Background

Steganography sounds like a modern technology but the fact is quite contrary. The first known application as ancient as Greek antiquity, when messages were tattooed on shaved heads of slaves and then their hair would allowed to be grown to keep message undiscoverable. Another method employed wax tables as a cover source. Messages were written on the underlying wood and they were covered with a new wax layer. The tables appeared blank so they passed inspection without inviting suspicion.

In the 20th century, invisible inks were a widely employed. In the Second World War, milk, vinegar, fruit juices and urine were very popular to write secret messages. When heated, these fluids become darker and the message was readable. Eventually, the Germans designed microdot technique. They were photographs with the size of a printed period but have the clarity of a standard typewritten page. Microdots were then printed inside a letter or on an envelope. Due to their tiny size, they could easily be sent unnoticed.

Recently, the United States government made a claim about Osama Bin Laden and the al-Qaeda organization using steganography to send messages through websites, emails and newsgroups. However, until now, no substantial evidence has been found supporting these accusations, so either al-Qaeda had really good steganographic algorithms, or the claim is probably not true.[2][3]

1.3 General Overview of Steganography

Steganographic systems are meant to effectively hide data termed as 'secret message'. Main key words used in the steganographic systems are: the cover medium, secret message, secret key and embedding algorithm. The cover medium, such as image, video, audio or text, is the carrier for the message. Secret message is the information which has to be hidden in the suitable digital cover media. The secret key is used to embed secret message depending on the hiding algorithms. Embedding algorithm is the way or the idea that is implemented to embed the secret information in the cover message.

Almost all digital file formats can be used as cover medium in steganography, but more suitable formats are the ones which have a high degree of redundancy. Redundancy can be defined as the bits of an object that provide far more accuracy than actually needed for display or utilization. The redundant bits of a cover medium are those which can be changed without the modification being perceived easily. Image and audio files specially fulfil this obligation, while research has also explored other file formats that can be employed for steganography.

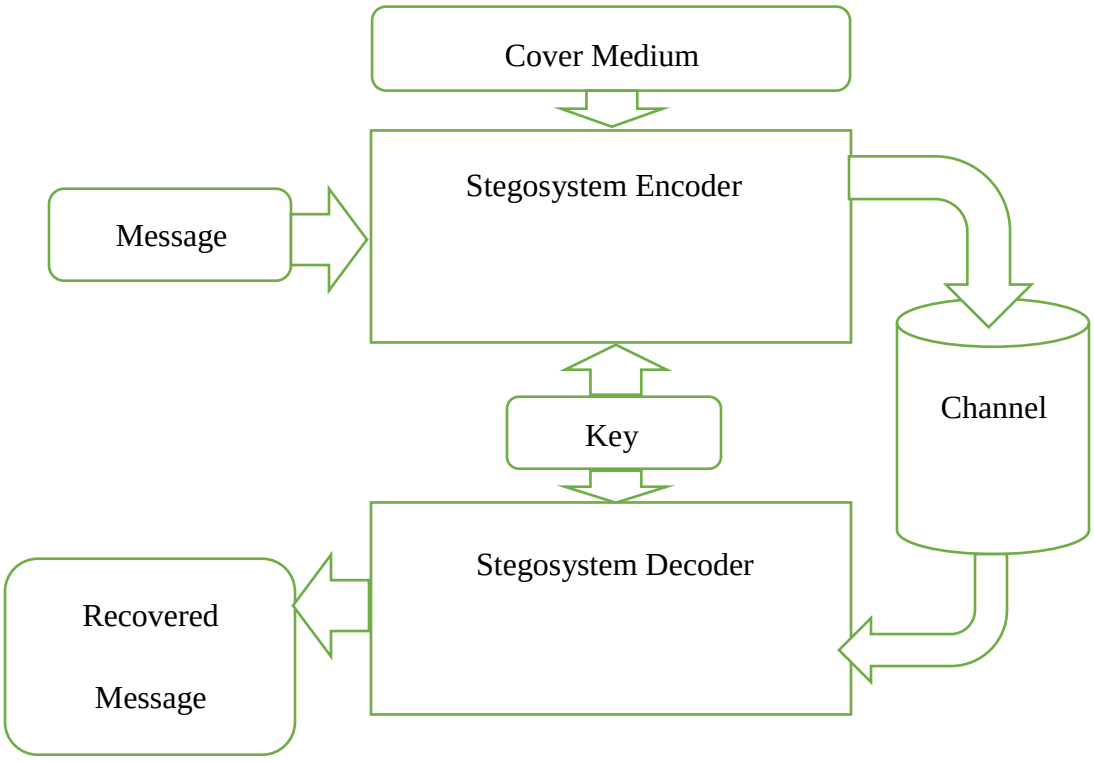

Fig. 1. Basic block diagram of steganographic system

The figure 1 shows basic block diagram that how steganographic system works. It is explained as follows:

- Cover medium is the carrier for secret message such as image, video, audio, text or any other digital media. Secret message is the information which is needed to be concealed in suitable digital cover medium.
- The secret key is employed to embed the message depending on the hiding algorithms. The embedding algorithm is the way or the idea that usually use to embed the secret information in the cover message.

Steganographic system may use any kind of digital file as cover file and secret message as well. However cutting edge techniques use redundancies in networking protocols as carrier media. Thus the carriers can be classified as below [2] [3]:

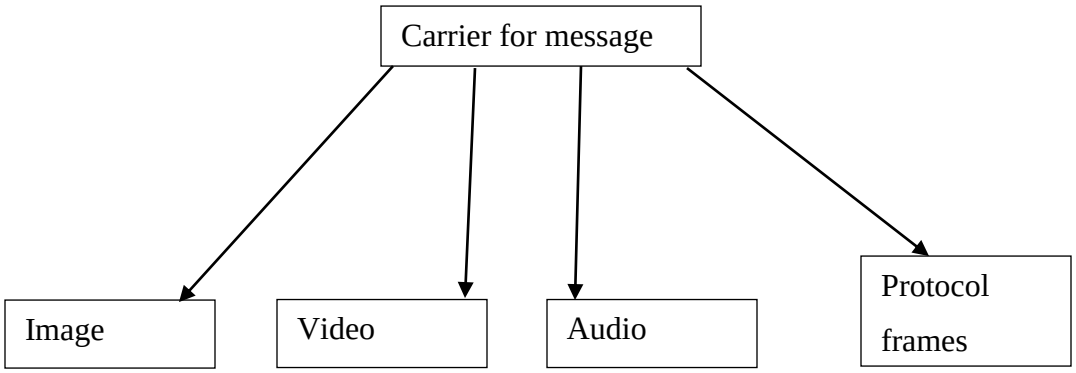

Fig. 2. Type of carriers in steganography

Images:

Images have been most popular cover medium to implement Steganography. A lot of research has already been done to develop and implement Steganographic algorithms involving images. Most of the work has been done to implement these algorithms on software level using image processing tools like MATLAB. A verity of formats are adopted for digital images like .jpg, .gif, .tiff, .bmp etc. Images are also preferred due to their Omni presence on recent social

networking platforms and other websites as well. However, there has been fewer hardware implementation of all these algorithms has been done.

Broadly, steganographic techniques involving image can be classified in these categories:

(i) Spatial Domain Techniques:

In spatial domain methods a steganographer modifies the secret data and the cover medium in the spatial domain, which involves encoding at the level of the LSBs. This method although simpler, has a larger impact compared to the other two types of methods.

Least significant bit (LSB) replacement is a common, simple approach to embedding in formation in a cover image. The LSBs of some or all of pixels inside an image is changed to bits of secret message. When using a true colour image, a bit of each of the red, green and blue colour components can be used, since they are each having size of 8 bit (total 24 bits for image itself). Thus, 3 bits can be stored in each pixel. So, a 600 × 400 pixel image, can store a total amount of 720,000 bits or 90,000 bytes of message data. For example an array of 3 pixels from a 24-bit image is be as follows:

(11011101 10111100 00001100)
(11100110 11110100 00111100)
(00010010 01011101 11110011)

When the number 210, which binary representation is 11010010, is embedded into the least significant bits of this part of the image, the resulting grid is as follows:

(1101110**1** 1011110**1** 0000110**0**)
(1110011**1** 1111010**0** 0011110**0**)
(0001001**1** 0101110**0** 1111001**1**)

Although First 8 bytes were sufficient for embedding the number, only the 5 underlined bits are needed to be changed according to the embedded message. On average, only half of the bits in given cover image will undergo a change to embed secret message. Since there are 256 possible intensity levels of each colour plane, altering the LSB of a pixel results in insignificant changes in the intensity in the colour planes. These deviations cannot be perceived by the human eye. So, the secret message is successfully concealed. With a well-chosen image, one can even hide the message in the LSB as well as second to LSB and still differences are not perceptible.

Major drawback of this method is that spatial domain algorithms are well known. Thus any steganalysis attack aimed to bit manipulation will easily detect secret information. One implementation has been developed using MATLAB code implementing LSB modification to illustrate the results of LSB steganography.

(ii) Transform Domain:

A more complex approach in steganography employs usage and modification of discrete cosine transformations' coefficients. Discrete cosine transformations (DCT), are used by the JPEG compression scheme to transform consecutive 8 x 8 pixel blocks of the image, into 64 DCT coefficients for each block. Each DCT coefficient F (u, v) of image f(x, y)'s 8 x 8 block of pixels is:

$$F(u,v) = \frac{1}{4}C(u)C(v)\sum_{\substack{0\le i\le 7\\ 0\le j\le 7}} f(x,y) * \cos\frac{(2x+1)u\pi}{16} * \cos\frac{(2y+1)v\pi}{16} \quad (1)$$

Where C(x) = 0.707 when x equals 0 and C(x) = 1 otherwise. After calculating the coefficients, the quantizing operation is performed as:

$$F^Q(u,v) = Floor(\frac{F(u,v)}{Q(u,v)}) \quad (2)$$

Where Q (u, v) is a 64-element quantization table. They are distinct according to camera and various proprietary compression schemes. This quantization coefficients can be manipulated the same way as explained in the LSB alteration spatial technique.

Although a change in a DCT coefficient would likely to affect all 64 pixels, the LSB of the quantized DCT coefficients can be used to embed secret message. Images compressed using lossless techniques would be vulnerable to visual modifications when the LSBs are changed. This has not been the case with the above mentioned method, as it takes place in the frequency domain of the image, instead of the spatial domain. So there would be no visual modifications to the cover image.

(iii) Masking and Filtering:

Masking and filtering techniques, usually restricted to 24 bits RGB or grayscale images, take a different method to conceal the message. These techniques are quiet comparable to paper

watermarks, creating patterns in the cover image. One of the ways is to modify the luminance component from parts of the image. While masking transforms the visible properties of an image, it has to be done in such a way that the human eye is not able to differentiate between cover and stego images.

Since masking utilizes visible features of the image, it is more robust than LSB modification. It is more immune to compression, cropping and different kinds of digital image processing. The information is not concealed at the ”noise” level but is inside the visible portion of the image, which makes it more optimum choice as compared to LSB modifications when a lossy compression scheme such as JPEG is being used.

- **Audio:**

Hiding information inside audio files can be done in many possible methods. Employing simple LSB replacement is possible, as alterations will usually not create audible modifications to the sounds. Another Technique involves exploiting limitations of human hearing system. Using inaudible range of frequencies, secret message is embedded in that bands. Using any frequencies exceeding 20 kHz, messages are embedded inside sound files and would go undetected by human checks.

Also, a message can be encoded using musical tones with a substitution scheme. For instance, an ‘X’ is tone would stand for a 0 and a ‘Y’ tone would represent a 1. A tune can now be composed around the secret message or an existing piece can be selected together with an encoding scheme. That would denote a message [2].

- **Video:**

Video files are generally an assortment of images and sounds, so most of above mentioned techniques prescribed on images and audio can be applied to video files as well. The great advantages of video are the large embedding capacity of data that can be hidden. The fact that video is a moving stream of images and sounds also helps. Any small but otherwise noticeable distortions, might go by unnoticed by human perception because of the uninterrupted flow of information [2].

- **Network Protocol Frames:**

Contrary to typical steganographic techniques that use digital media such as images, audio and video files to hide data, network steganography utilizes communication protocols' control elements and inherent function. Thus, such techniques are tougher to detect and remove.

Typical network steganography techniques employ altering the features of a single network protocol. These modifications can be applied to the PDU (Protocol Data Unit), to the time relationships amongst the exchanged PDUs or both. [4]

Network steganography covers a vast range of methods, which are:

(i) **Steganophony** is embedding messages in Voice-over-Internet Protocol conversations, for example, the engagement of delayed or corrupted VoIP packets that would be ignored by the receiver under normal operation (this is called LACK — Lost Audio Packets Steganography). Another approach is hiding message in unused header fields.

(ii) **WLAN Steganography** is transmission of secret message in Wireless Local Area Networks. A practical example of WLAN Steganography is the HICCUPS system (Hidden Communication System for Corrupted Networks) [5][6]

1.4 Comparison of Steganography, Watermarking and Encryption

The trio- steganography, Watermarking and Encryption serves the same purpose to secure data and to hide it from unauthorized access. However there are some stark differences, which can be summarized as followed [6]:

TABLE I COMPARISON OF STEGANOGRAPHY, WATERMARKING AND ENCRYPTION

Method / Criterion	Steganography	Watermarking	Encryption
Carrier	Any digital media	Usually image or audio files	Mostly text based, with some support to image files
Secret Data	Secret message(any form)	Watermark	Plain text
Key	Optional	N/A	Compulsory
Input File	2	1	1
Objective	Secrete communication	Copyright preserving	Data protection
Concern	Delectability or capacity	Robustness	Robustness
Type of Attack	Steganalysis	Image processing	Cryptanalysis
Detection	Usually Blind	Mostly achieved by cross correlation	Full recovery of data
Visibility	Never	It is removed/replaced	De-ciphered
Flexibility	Free to choose any suitable cover	Cover choice is limited	N/A

1.5 Performance Evaluation for Various Steganographic Techniques

All the above mentioned techniques for steganography have different strengths and weaknesses. It is significant to ensure that only the most suitable algorithm is employed for the application. All steganographic algorithms have to fulfil some basic requirements. The most essential

condition is that a steganographic algorithm has to be unnoticeable .A set of parameters mentioned below further describe the imperceptibility of an algorithm. [2] [3] [7]:

(i) **Payload capacity:** Steganography aims at hidden communication and thus call for sufficient embedding payload capacity. In The real time systems the payload capacity directly refers to embedding rate for information, which is critical for any real time system,

(ii) **Invisibility:** The invisibility of a steganographic algorithm is the most vital prerequisite, since the strength of steganography lies in its capability to be undetected by the human perception and various attacks after that. The moment it is observed that a cover media has been tampered, the algorithm is compromised.

(iii) **Robustness:** Secret message should be recoverable from a stego message even after coming across noise or attacks. Statistical steganalysis is the method of detecting secret message through employing statistical tests on target data. Many steganographic methods leave a 'signature' while embedding. Detecting them is easy using statistical analysis. A steganographic algorithm must not leave a mark in the image which is statistically significant.

(iv) **Computational complexity**: Processing time required to hide whole message or recovery of the same should not be too long and it should be suitable according to the goal of the steganography scheme. It becomes especially important when steganographic system has to be implemented on the hardware.

(v) **Independence from cover media file format**: It might attract suspicion that only certain types of file formats are being communicated between sender and receiver. Thus most powerful steganographic algorithms usually have the capability to embed message in any type of file format for given type of media (i.e. image, audio etc.). This also leads to the solution of the problem that a suitable cover may not be available at the right moment and in the right format to be employed as a cover media.

(vi) **Security**: Embedded secret messages should be secured itself, and the cover media must not have any indication of their existence. A secret key can be used to increase

the steganography security level. It can be a randomization seed about selecting order of cover pixel or a key for additional cryptographic layer itself.

2. LITERATURE SURVEY

Steganography has been undergone a large amount of research in recent times. Numerous papers on steganographic techniques, attacks and hardware implementation has been studied. The following section gives a brief review of studied literature.

2.1 Abbas Cheddad *et al.* [7]:

The paper focuses mainly reviews and analyses various steganographic algorithms on digital images. Some common standards and guidelines for steganography has also been discussed. There are several recommendations suggested in the paper. It advocates use of Object Oriented Programming methods. Even though not being primary focus of the paper, steganalysis has been discussed in brief. In The techniques focused on DCT, DWT and adaptive methods, image distortion is very less because information is hidden inside coefficients of certain transformation. This fact is significant especially when hidden message is relatively smaller. .These newer methods don't have embedding rate as higher as those achieved by spatial domain methods. Higher embedding rates have remained Achilles heel for steganographic algorithms. Digital video files are emerging as a good alternative to image files due to much higher embedding capacity and better imperceptibility. However, major challenges for video files are embedding issues in highly intercorrelated images and the fact that video files undergo manipulation in compressed forms.

2.2 Andreas Westfeld [9]:

This paper has been accepted as a benchmark in the field of steganography which uses transform domain coefficients as cover media. It is one of the most cited and implemented transform domain steganography algorithms. There are steganographic algorithms that are resistant of attacks while having lesser embedding capacity. While algorithms with greater embedding rate have been found to be weak against statistical and visual attacks. F5, the algorithm presented in the paper, combines both- higher embedding rate and greater resistance against visual and statistical attacks. For equalization of embedding rate and reduction in required number of changes in coefficients, matrix encoding and permutative straddling

techniques have been employed. F5 achieves embedding capacity of 13% of cover JPEG file size. However, it has not undergone any hardware implementation till date.

2.3 Wojciech Mazurczyk *et al.* [10]:

This paper provides a detailed survey of modern steganographic techniques targeting smartphones. Major attention has been given to the methods developed during the period 2005 to the second quarter of 2014. All diverse methods have been classified according to the subsystem of the device targeted to hide message. Three covert channels, according to the authors, are local, object and network. Also, it reviews the relevant approaches used to detect and overcome steganographic attacks or threats. Last of all, it cites some of the most popular software applications for steganography and possible future scope. This paper covers most of modern steganographic techniques including approximately 150 sources. These techniques require low computation power and they can be realized on hardware with low power consumption. Most of techniques are yet to be implemented on any kind of hardware.

2.4 Y. F Huang *et al.* [11]:

This paper is one of foundational paper on network steganography. The paper dissipates a newer steganography algorithm with higher embedding capacity, in which data hiding has been accomplished using the inactive frames of low bit rate audio streams. Voice over Internet Protocol (VoIP) utilizes such kind of audio frames in an extensive fashion. The work shows that inactive VoIP frames can achieve higher embedding capacity while maintain similar imperceptibility as compared to the active audio frames, which is in contrast with normal assumptions. The results from experimental analysis shows that proposed method achieves a greater embedding capacity while distortion of the original speech maintained at imperceptible levels. The paper also establishes that the hiding message in in active audio frames is more suitable than in active audio frames of VoIP using proposed steganographic algorithm.

2.5 Haider Shahadi Ismael *et al.* [12]:

This paper emphasizes on hardware implementation of an audio steganography method. Most steganography methods does not qualify for real time communication. Through parallel processing capacities provided by Field Programmable Gate Array (FPGA), speed of a steganographic system can improve .There is much less explored in this field. In this paper a parallel hardware-architecture for dual-mode audio steganography (DMAS) based on FPGA

hardware has been described. Reconfiguration of the same hardware blocks in embedding as well as recovery processes reduces the hardware utilization. Successful implementation on a Xilinx XC6SLX16 FPGA board yields utilization of 97 slices. Data processing at an operating frequency of up to 58.82 MHz and full message recovery at an embedding rate of 25% with an audio quality above 45 dB in terms of PSNR has been achieved concurrently. Any kind of steganographic payload like compressed images, audio files and ASCII text can be embedded in cover audio file by the virtue that present design is a generic audio processor. Another application of the design lies in hiding speech messages for secure real time communication where cover audio file have standard sampling frequency and secret message frequency may lie in a specific range.

2.6 Saeed Sarreshtedari *et al.* [13]:

The paper explores a new spatial domain LSB algorithm providing capacity of one bit per pixel. Without compromising with the embedding capacity, proposed one third probability embedding method reduces probability of modifying a pixel value to one third. It results in better resistance against many LSB steganography detectors and also improves imperceptibility. A pixels carries secret message bits according to a function of three neighbouring cover. The proof about no effective improvements being achieved by increasing pixel sequence length has also been provided. A derivation of a closed form equation for change in terms of number of pixels in pixel group is also given. Proposed algorithm also demonstrates histogram equalization properties to compensate changes in cover image histogram whenever it is possible. Results presented in the paper shows that proposed algorithm performs better than histogram compensating version of LSB matching algorithm. The stego image pixels are more similar to the corresponding cover pixels due to reduction of the probability of change per pixel, while maintaining the same capacity. Due to this feature imperceptibility and protection against well-known steganalysis methods.

2.7 Ozdemir Cetin *et al.* [14]:

In addition to still images, this paper aims to utilize steganographic technique for video streams. For an operational and successful hiding process the selection of target pixels, which are going to been used for storing the secret message is especially crucial whenever stenographical methods are applied to video carries. Improper pixel selection may lead to undesired temporal

and spatial perception issues occurring in the resultant video. Two new steganographic algorithms using similar histograms and dissimilar histograms are proposed in this paper. Both algorithms are based on selecting appropriate pixel approaches by perceptibility and capacity parameters of the cover video. While offering relatively higher data embedding capacity, these algorithms also improves spatial and temporal perception of stego video. Using a user defined HCV parameter, this method has been implemented for a target application. In dissimilar histogram methods, block based approach yields better result than similar frame based approaches.

3. IDENTIFIED GAPS IN RESEARCH AND PROBLEM STATEMENT

3.1 Analysis of Research Gaps in Steganography Techniques

Following table summarises several papers from reputed journals along with parameters on which research gaps have been discovered in this study.

TABLE II ANALYSIS OF RESEARCH GAPS IN STEGANOGRAPHY TECHNIQUES

Parameter / **Paper**	**New algorithm**	**Software Implementation**	**Performance analysis**	**Hardware Implementation**
Abbas Cheddad *et al.*	Review paper	No	Yes	
Andreas Westfeld	Yes (F5 , Jsteg also mentioned)	Yes	Yes	
Wojciech Mazurczyk *et al.*	Review paper	No	Yes (Only for some algorithms)	
Yong Feng Huang *et al.*	Yes	Yes	Yes	
Haider Ismael Shahadi *et al.*			Yes	Yes
Sarreshtedari *et al.*	Yes	Yes	Yes	
Ozdemir Cetin *et al.*	Yes	Yes	Yes	

3.2 Problem Statement based on Gaps in Research

- Hardware implementation of steganographic techniques has been found comparatively rare as per identified in the survey. Thus Hardware implementation is main objective as it improves real time performance of the steganographic system.
- In literature survey, numerous steganographic techniques targeting a range of cover media have been studied. They have been examined for their performance parameters like imperceptibility, robustness, embedding capacity.
- The principal objective of this thesis work has been to design a hardware based image steganographic system which can embed a secret message (which is also another image in stored in system) into desired cover image employing the best embedding and extraction algorithm possible.
- Spatial domain steganography techniques, like LSB substitution, offer better embedding capacity. This improvement in embedding capacity is achieved using operations which are relatively simpler than those used by the frequency domain techniques. However, weakness of such techniques against statistical attacks like chi-square test has motivated development of LSB matching.
- Improving over LSB matching, more recent one third probability steganography has reduced probability of change in pixel value is considered as 0.5 to 1/3. One third probability algorithm has also been found to be resistant against statistical steganalysis attacks, all of which makes it an ideal choice for proposed system.
- One third probability algorithm has been implemented on image processing tools of MATLAB. Performance parameters of a steganographic algorithm like embedding capacity and imperceptibility have been evaluated.
- Motivation for this work lies in the fact that most of the operations explained in one third probability algorithm are not optimized or well supported for hardware implementation. Purpose of this work has also been to modify the algorithm for hardware implementation and then implement the modified algorithm on FPGA. A FSM containing details of hardware implementation has been designed.
- The system has been designed to contain memories for image storage and histogram compensating operations. The scheduling of reading-writing cycles in those memories

and order of operations in the embedding process is determined by a finite state machine (FSM).

- Design has to be coded in any HDL targeting a FPGA board using suitable tools for FPGA design flow. After implementation, the design must be evaluated for various parameters of area, power and static timing performance.

4. ONE THIRD PROBABILITY EMBEDDING ALGORITHM AND ITS OPTIMIZATION FOR HARDWARE IMPLEMENTATION

Considering various Performance Parameters identified in the literature survey, algorithm given by Sarreshtedari *et al.* [13] has been selected for hardware implementation. Various Performance Parameters like Embedding Capacity, Resistance against Steganalysis and Invisibility has been considered while picking the algorithm. The following section covers details of one third probability steganography algorithm for images. Firstly, operations from original algorithm has been described. They are followed by various proposed revisions to optimize or modify some of the operations to achieve optimum hardware design.

4.1 One Third Probability Embedding Algorithm

4.1.1 Introduction:

Simple most algorithm of spatial domain image steganography is LSB replacement method. The LSB of a pixel is simply replaced by secret message bit. Now odd valued pixel remains the same or decremented by one grey level. Similarly even vales either remains the same or become incremented by '1'. Eventually histogram of resultant stego image develops pair wise structures because every two consecutive grey level of histogram converges to the same grey level value. These pair wise formations are easily noticeable in stego image histograms making steganalysis easier and comprising on imperceptibility.

An improved algorithm called LSB matching (LSBM) tries to resolve this major weakness [15]. In LSB matching, pixel value is increased or decreased in random fashion whenever there is mismatch in cover pixel LSB and secret message bit. To crack LSBM, Steganalysis methods based on centre of mass of histogram characteristics function (HCF-COM) have been proposed by Harmsen *et al*. [16]. However, they are more complicated than previous steganalysis methods.

LSBM assumes that message is encrypted and incoming bit stream is purely random sequence. So, each message bit may or may not match with cover pixel LSB with exact one half probability. Thus, probability of change per pixel can be concluded as 0.5 in LSB matching.

Mielikainen [17] has proposed an improved version of LSBM, called revisited LSB matching (LSBMR) Proposed LSBMR improves probability of change per pixel to 0.37 from 0.5.

4.1.2 Basics and Definitions:

Minor Change: Minor change function adds or subtracts '1' from any pixel value 'x' to keep ⌊x/2⌋ unchanged.

Mathematically,

$$Minor\ Change(x) = \left\lfloor \frac{x}{2} \right\rfloor * 4 + 1 - x \tag{3}$$

Major Change: Minor change function adds or subtracts '1' from any pixel value 'x' to make ⌊x/2⌋ changed.

Defining it mathematically,

$$Major\ Change(x) = \left\lfloor \frac{x+1}{2} \right\rfloor * 4 + 1 - x \tag{4}$$

A group of 3 pixel is taken at a time for embedding message. Embedding process of the message is done in such a fashion that, corresponding resultant stego image pixels satisfy equations stated below:

$$m_{i=} f\left(y_{i,} y_{i+1}\right) = LSB(y_i + \left\lfloor \frac{y_{i+1}}{2} \right\rfloor) \tag{5}$$

$$m_{i+1=} f\left(y_{i+1,} y_{i+2}\right) = LSB(y_{i+1} + \left\lfloor \frac{y_{i+2}}{2} \right\rfloor) \tag{6}$$

$$m_{i+2=} f\left(y_{i+2,} y_{i}\right) = LSB(y_{i+2} + \left\lfloor \frac{y_{i}}{2} \right\rfloor) \tag{7}$$

Where y_i, y_{i+1} and y_{i+2} are values of pixel from resultant stego image.

For decision making, triple result vector 'r' is defined as:

$$r = [r_i, r_{i+1}, r_{i+2}] = xor([m_i, m_{i+1}, , m_{i+2}], [f\left(x_{i,} x_{i+1}\right), f\left(x_{i,} x_{i+1}\right), f\left(x_{i,} x_{i+1}\right)]) \quad (8)$$

4.1.3 Embedding Algorithm:

The algorithm can be described step by step as stated below:

Step 1 Initialization:

In algorithm given by Sarreshtedari *et al.*[13], the cover image 'x' is copied into the stego image 'y'. Pixels having extreme values of N − 1 and 0 are decreased and increased by one, correspondingly. The N × N matrix 'C' where N − 1 is the maximum value allowed for pixel values is defined and all of its entries are set to zero.

Step 2 First Scan:

The cover image is divided into groups, each consisting three pixels (x_i, x_{i+1}, x_{i+2}). These groups are called embedding units. A secret key governs selection order of pixels from cover image. Thus security of the embedding process is assured.

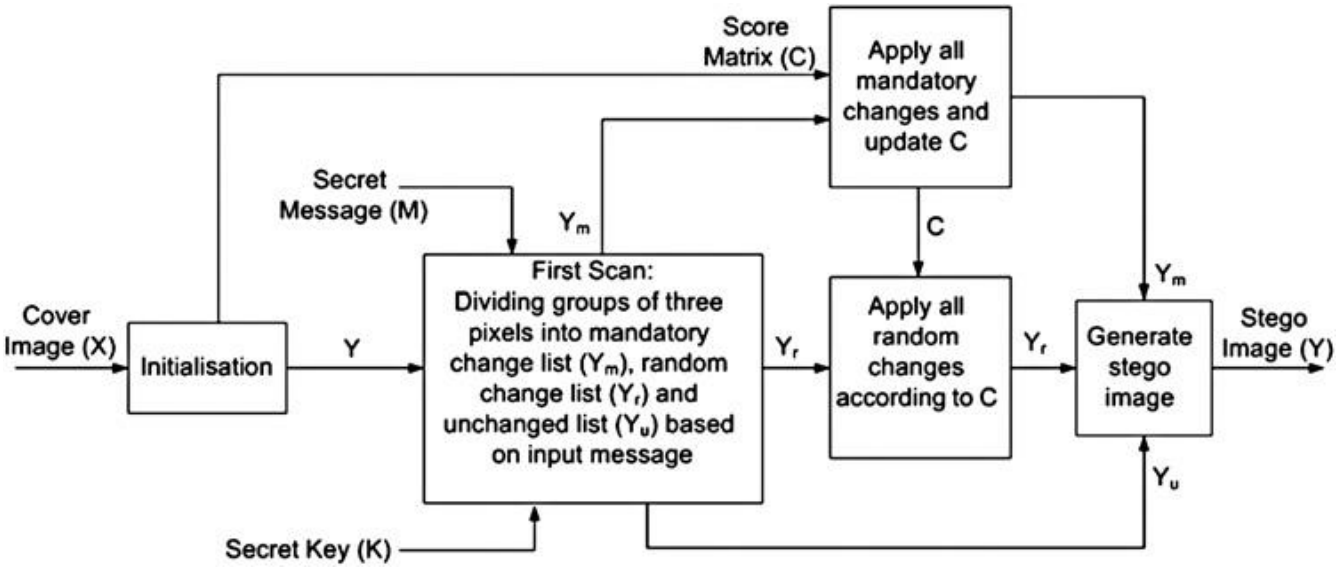

Fig. 3. Top level block diagram of embedding algorithm [13]

For each embedding unit, corresponding secret message bits are drawn from message source. Resultant stego pixels (y_i, y_{i+1},y_{i+2}) would hold values after data embedding.

The result vector is calculated as per given equation (8). These computations are continued until all of the result vectors, corresponding to all embedding units, have been derived. Based on them, the embedding units are separated into three distinct groups:

1. If r = [0, 0, 0], the embedding unit is added to no change list. These pixel groups would not undergo any changes.
2. Mandatory list consists of pixel embedding groups whose result vector consists of at least one or two '1's in them. According to specific result vector, pixel group must undergo a major or minor change
3. If r = [1, 1, 1], the corresponding embedding unit's pixel would face random changes as per 'score' calculation.

Step 3 Mandatory Embedding Phase:

According to drawn pixel values and secret message bits, cover pixel value would undergo corresponding change as mentioned in the table 3.

TABLE III MANDATORY CHANGE PHASE COMPUTATIONS [13]

r vector	Resultant Stego Pixel	Corresponding Change in Score Matrix 'C'
[1, 0, 0]	$y_i = Minor(x_i)$	$C(y_i, x_i) = C(y_i, x_i) + 1$ $C(x_i, y_i) = C(x_i, y_i) - 1$
[0, 1, 0]	$y_{i+1} = Minor(x_{i+1})$	$C(y_{i+1}, x_{i+1}) = C(y_{i+1}, x_{i+1}) + 1$ $C(x_{i+1}, y_{i+1}) = C(x_{i+1}, y_{i+1}) - 1$
[1, 1, 0]	$y_{i+1} = Major(x_{i+1})$	$C(y_{i+1}, x_{i+1}) = C(y_{i+1}, x_{i+1}) + 1$ $C(x_{i+1}, y_{i+1}) = C(x_{i+1}, y_{i+1}) - 1$
[0, 0, 1]	$y_{i+2} = Minor(x_{i+2})$	$C(y_{i+2}, x_{i+2}) = C(y_{i+2}, x_{i+2}) + 1$ $C(x_{i+2}, y_{i+2}) = C(x_{i+2}, y_{i+2}) - 1$
[1, 0, 1]	$y_i = Major(x_i)$	$C(y_i, x_i) = C(y_i, x_i) + 1$ $C(x_i, y_i) = C(x_i, y_i) - 1$

[0, 1,1]	$y_{i+2} = Major(x_{i+2})$	$C(y_{i+2}, x_{i+2}) = C(y_{i+2}, x_{i+2}) + 1$ $C(x_{i+2}, y_{i+2}) = C(x_{i+2}, y_{i+2}) - 1$

The score matrix (C) is delivered to the next random embedding stage. Approximately 7/8 amount of message embedding is accomplished in this stage.

Step 4 Random Embedding Phase:

For each embedding sets resulting in r = [1, 1, 1], random changes have to be made. In random phase three different change sets can satisfy equations (5) to (7). To determine best possible change set regarding histogram compensation, a 'score' has to be calculated. Score calculation includes summation of signs of entries from 'C' matrix for each possible change set. An example would give better understanding of the procedure. If there are 2 bit wide pixel registers, then intensities can vary from 0 to 3. After finishing mandatory embedding phase, resultant change matrix is,

$$C=\begin{pmatrix} 0 & 4 & 0 & 0 \\ -4 & 0 & 1 & 0 \\ 0 & -2 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

For example, message embedding set m= [0, 1, 1] and cover pixel set x= [1, 1, 2].

So,

A= LSB ([(1 + LSB (⌊1/2⌋)), (1 + LSB (⌊2/2⌋)), (2 + LSB ⌊1/2⌋)]) = [1, 0, 0]

Vector r = [0, 1, 1] xor [1, 0, 0] = [1, 1, 1]

Here one of three pixels has to undergo a minor change and one of them has to face major change. The third pixel remains unchanged. Score can be calculated for each case and best possibility gets picked up.

1. Major(x1):1 to 2, Minor(x2):1 to 0 ⇒ Score s1 = sign(C(1, 2)) + sign(C(1, 0)) = 0
2. Major(x2):1 to 2, Minor (x3)) :2 to 3⇒ Score s2 = sign(C(1, 2)) + sign(C(2, 3)) = 2
3. Major(x3):2 to 1, Minor (x1):1 to 0⇒ Score s3= sign(C(2, 1)) + sign(C(1; 0)) = −2

Thus the best choice is second case.

As grey level ‘1’ and ‘2’ are facing changes here, entry (2,1) and (3,2) are incremented and entry (1,2) and (2,3) are decremented by one in C matrix.

$$C=\begin{pmatrix} 0 & 4 & 0 & 0 \\ -4 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

For every change, two entries in ‘C’ matrix are manipulated which is similar to the mandatory embedding phase. If two change sets are found with same score than change set with maximum addition value of corresponding entries in ‘C’ matrix is selected.

Step 5 Stego Image Generation:

Embedding process in all three list is accomplished and pixels are relocated to their original places to generate stego image.

4.1.4 Extraction Algorithm:

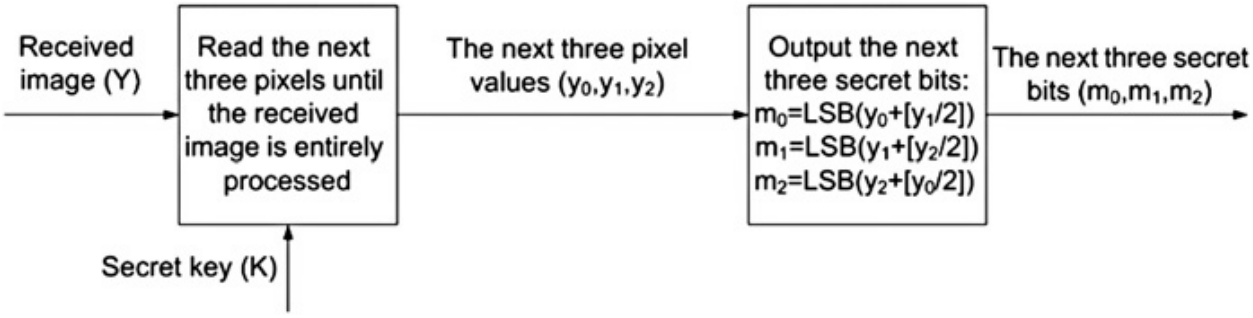

Fig. 4. Top level block diagram of extraction algorithm [13]

Embedded units (y_i, y_{i+1}, y_{i+2}) are derived from stego image, which are basically group of three adjacent pixels. The same secret key, which was used in embedding process, is reused here to decide selection order for extraction process. The secret message bits are extracted using equation (5) to equation (7). Extraction process halts when all message bits are detected.

4.2 Modified One Third Probability LSB Algorithm

As explained earlier, motivation for this work lies in the fact that most of the operations explained in one third probability algorithm are not optimized or well supported for hardware implementation. Purpose for this section is to modify the algorithm for hardware implementation.

4.2.1 Basic Definitions

Minor Change: More optimized version in terms of hardware design is to consider the grey pixel value. If a grey pixel value is an even number, then minor function increments it by one and decrements it by one if number is odd.

$$Minor(x) = x + 1 \qquad (when\ x\ is\ even) \tag{9}$$

$$Minor(x) = x - 1 \qquad (when\ x\ is\ odd) \tag{10}$$

Major Change: Similar to the minor change function, the major change function can also be optimized. If a grey pixel value is an even number, then minor function decrements it by one and increments it by one if number is odd.

$$Major(x) = x - 1 \qquad (when\ x\ is\ even) \tag{11}$$

$$Major(x) = x + 1 \qquad (when\ x\ is\ odd) \tag{12}$$

Observing equation (3) and (4), one can easily conclude that major and minor functions require many complex operations. Each of them may require an addition, a subtraction, a multiplication, a division operation and one application of floor function. Each of these operation require comparatively greater hardware resources, especially multiplication and division operations. Floor function may not be supported by many synthesizer tools. Even if there is any kind of support for floor function, it would lead to a larger chunk of hardware resources.

Whereas equation (9) to (12) shows that all of them can be combined with a simple multiplexer (to decide if given number is even or odd) accompanied by an adder and a subtractor. Multiplexers are available in most of FPGA families as an atomic unit in a similar fashion with adders-subtractor. Thus It would lead to more efficient design at the end.

LSB Function: LSB function can also be optimized as:

$$f'\left(x_{i,}\, x_{i+1}\right) = LSB(x_{i,} + (1\ place\ arithmaticaly\ shifted\ value\ of\ x_{i+1})) \quad (13)$$

It is noteworthy that an arithmetic right shift by 1 place produces the same result as division and the floor function combined in equation (8).

For example, let's say a=1, b=3.

Thus, f' (a, b) = LSB (1+floor (3/2)) = LSB (1+floor (1.5)) = LSB (2) = 0.

Using equation (13),

f'(a, b) = LSB (1+(arithmetically one place shifted value of 3_{10}=$(00000011)_2$)

= LSB (1 + ($(00000001)_2$= 1_{10})) = LSB (2) = 0.

In LSB function a simple observation can be made that floor function and division operations are replaced with a single arithmetic right shift. Hardware equitant of arithmetic right shift is generally a simple shift register, which is much more inexpensive than floor function and division operations combined.

Result Vector:

To determine operation on pixel group, result vector is defined as:

$$r = [r_i, r_{i+1}, r_{i+2}] = xor([m_i, m_{i+1}, , m_{i+2}], [f'\left(x_{i,}\, x_{i+1}\right), f'\left(x_{i,}\, x_{i+1}\right), f'\left(x_{i,}\, x_{i+1}\right)]) \quad (14)$$

Where f(x, y) is hardware optimized version of LSB function.

Optimization of all three functions, which are used repeatedly, ensures least hardware consumption by eliminating costly operations like floor function, division, multiplication as well as excess number of additions and subtractions.

Use of floor function and division operation may require end-result or intermediate variable to be a floating point number, which will require a separate class support, precision values. It may lead to a 64 bit register for floating number with required precision in place of 8

bit unsigned integer value. Additionally, limited or no synthesis support is available for floating point type [20].

4.2.2 Embedding Steps

Modified one third probability algorithm follows these steps to embed secret message in cover picture.

Step 1 Initialization:

As per directed in original algorithm, cover image is copied in stego image. But Structure of change matrix ‘C’ has been replaced with lower and upper triangular arrays ‘lt’ and ‘ut’.

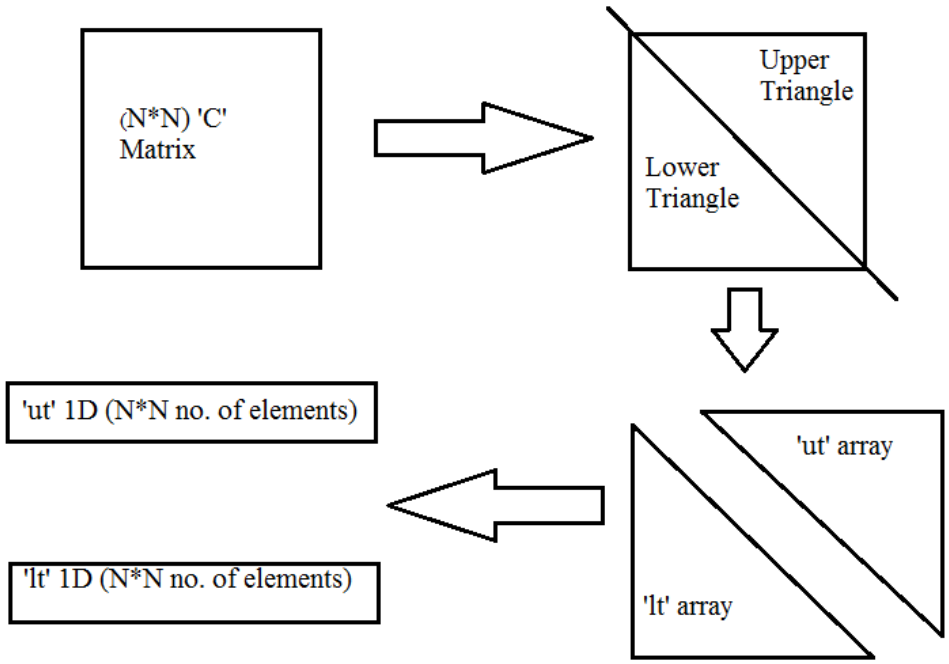

Fig. 5. Development of ‘lt’ and ‘ut’ arrays from change matrix ‘C’

Major reason being the fact that synthesis of two dimensional array consisting signed integers has not been supported by majority of synthesis tools to create RTL level description. Thus, this array is converted in two single dimensional array with control signals to write changes in pixel weather in upper or lower triangle as shown in the figure. Each, lower triangular array “lt” and upper triangular array “ut” both do have (N*N) no. of elements. They all are initialized as zero.

Step 2 First Scan:

The cover image is divided into groups of three pixels (x_i, x_{i+1}, x_{i+2}) which are called embedding units ((x_i, x_{i+1}, x_{i+2}) are the same as (y_i, y_{i+1}, y_{i+2}) at the beginning.

For each embedding unit, three secret bits (m_i, m_{i+1}, m_{i+2}) are fetched from the input. Embedded unit (y_i, y_{i+1}, y_{i+2}) would contain values for corresponding pixels in the stego image after data embedding.

The result vectors are calculated as given in equation (14). This calculation is repeated until the result vectors are derived for all embedding units. The changes in embedding units are performed according to the table 2.

If there is a change in C(x,y) or C(y,x) element according to the mandatory phase calculations as described in original one third probability algorithm, corresponding change in lower or upper triangle matrix would take place as described below:

1. For an entry in C matrix C(a, b), if a>b, then corresponding change would happen in lower triangular matrix.
2. For an entry in C matrix C(a, b), if a<b, then corresponding change would happen in upper triangular matrix.
3. For an entry in C matrix C(a, b), if a=b, then corresponding change happens in diagonal elements of C matrix. It would correspond to no change case when r = [0, 0, 0]. So no action is required.

Step 3 Mandatory Phase:

For each embedding unit in cover image, pixel values and corresponding secret message bits are drawn. The pixel values are transformed according to the following 8 cases. Last two rows shows modifications with respect to ‘lt’ and ‘ut’ arrays.

TABLE IV MANDATORY EMBEDDING PHASE COMPUTATIONS IN MODIFIED ALGORITHM

r vector	**Resultant stego pixel after change**	**Change if y>x**	**Change if y<x**

[1, 0, 0]	$y_i = Minor(x_i)$	$lt(y_i * x_i) = lt(y_i * x_i) + 1$ $ut(x_i * y_i) = ut(x_i * y_i) - 1$	$ut(y_i * x_i) = ut(y_i * x_i) + 1$ $lt(x_i * y_i) = lt(x_i * y_i) - 1$
[0, 1, 0]	$y_{i+1} = Minor(x_{i+1})$	$lt(y_{i+1} * x_{i+1}) = lt(y_{i+1} * x_{i+1}) + 1$ $ut(x_{i+1} * y_{i+1}) = ut(x_{i+1} * y_{i+1}) - 1$	$ut(y_{i+1} * x_{i+1}) = ut(y_{i+1} * x_{i+1}) + 1$ $lt(x_{i+1} * y_{i+1}) = lt(x_{i+1} * y_{i+1}) - 1$
[1, 1, 0]	$y_{i+1} = Major(x_{i+1})$	$lt(y_{i+1} * x_{i+1}) = lt(y_{i+1} * x_{i+1}) + 1$ $ut(x_{i+1} * y_{i+1}) = ut(x_{i+1} * y_{i+1}) - 1$	$ut(y_{i+1} * x_{i+1}) = ut(y_{i+1} * x_{i+1}) + 1$ $lt(x_{i+1} * y_{i+1}) = lt(x_{i+1} * y_{i+1}) - 1$
[0, 0, 1]	$y_{i+2} = Minor(x_{i+2})$	$lt(y_{i+2} * x_{i+2}) = lt(y_{i+2} * x_{i+2}) + 1$ $ut(x_{i+2} * y_{i+2}) = ut(x_{i+2} * y_{i+2}) - 1$	$ut(y_{i+2} * x_{i+2}) = ut(y_{i+2} * x_{i+2}) + 1$ $lt(x_{i+2} * y_{i+2}) = lt(x_{i+2} * y_{i+2}) - 1$
[1, 0, 1]	$y_i = Major(x_i)$	$lt(y_i * x_i) = lt(y_i * x_i) + 1$ $ut(x_i * y_i) = ut(x_i * y_i) - 1$	$ut(y_i * x_i) = ut(y_i * x_i) + 1$ $lt(x_i * y_i) = lt(x_i * y_i) - 1$
[0, 1,1]	$y_{i+2} = Major(x_{i+2})$	$lt(y_{i+2}, x_{i+2}) = lt(y_{i+2} * x_{i+2}) + 1$ $ut(x_{i+2} * y_{i+2}) = ut(x_{i+2} * y_{i+2}) - 1$	$ut(y_{i+2}, x_{i+2}) = ut(y_{i+2} * x_{i+2}) + 1$ $lt(x_{i+2} * y_{i+2}) = lt(x_{i+2} * y_{i+2}) - 1$
[0, 0,0]	No Change	No Change	No Change
[1,1,1]	Save to the random list	No Change	No Change

Step 4 Random Phase:

After completion of mandatory embedding phase, random embedding phase consists of three different possible change sets similar to original algorithm. In modified algorithm too, the

‘score’ has to be calculated to examine best possible change set that compensates changes in histogram due to mandatory round of embedding procedure.

For same cover pixel embedding set and secret message embedding set described in step-4 of section 4.4.1, contents in lower and upper triangular arrays can be obtained as mentioned below. Note that diagonal elements have been excluded because they corresponds to no change operations.

lt[1:15] = [0 C(2*1) C(3*1) c(4*1) 0 0 C(3*2) 0 C(4*2) 0 0 0 C(4*3) 0 0 0]

= [0 -4 0 0 0 -2 0 0 0 0 0 -1 0 0 0]

ut[1:15] = [0 C(1*2) C(1*3) c(1*4) 0 0 C(2*3) 0 C(2*4) 0 0 0 C(3*4) 0 0 0]

= [0 4 0 0 0 2 0 0 0 0 0 1 0 0 0]

For embedding unit x= [1, 1, 2] and m= [0, 1, 1]

r = [0, 1, 1] xor LSB ([(1 + LSB (⌊1/2⌋)), (1 + LSB (⌊2/2⌋)), (2 + LSB ⌊1/2⌋)]) = [1, 1, 1]

1. Major(x1):1 to 2, Minor(x2):1 to 0 ⇒ Score(1) = sign(ut(2*3)) + sign(lt(2*1)) = 0
2. Major(x2):1 to 2, Minor (x3)) :2 to 3⇒ Score(2) = sign(ut(2*3)) + sign(ut(3*4)) = 2
3. Major(x3):2 to 1, Minor (x1):1 to 0⇒ Score(3) = sign(lt(3*4)) + sign(lt(2*1)) = −2
   Thus the best decision is second case.

As grey level ‘1’ and ‘2’ are facing changes here, entry (3,2) and (4,3) are incremented and entry (2,3) and (3,4) are decremented by one in C matrix.

$$C=\begin{pmatrix} 0 & 4 & 0 & 0 \\ -4 & 0 & 1 & 0 \\ 0 & -10 & 0 & \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Corresponding changes in lt and ut arrays would be:

lt = [0 -4 0 0 0 -1 0 0 0 0 0 0 0 0 0] ; ut = [0 4 0 0 0 1 0 0 0 0 0 0 0 0 0].

Thus for all pixel sets, which had been stored in random change list undergo such modifications and stego image is finally complete. It is noteworthy that this result is similar to the one obtained by original algorithm in section 4.4.1.

4.2.3 Extraction Steps

Extraction procedure is driven by fetching three pixel group of stego image which are called embedded units. Secret message is extracted according to the equations given below. It is notable that hardware optimized version of LSB function is used here.

$$m_{i=} f'\left(y_{i,} y_{i+1}\right) = LSB(y_i + \left\lfloor \frac{y_{i+1}}{2} \right\rfloor) \quad (15)$$

$$m_{i+1=} f'\left(y_{i+1,} y_{i+2}\right) = LSB(y_{i+1} + \left\lfloor \frac{y_{i+2}}{2} \right\rfloor) \quad (16)$$

$$m_{i+2=} f'\left(y_{i+2,} y_i\right) = LSB(y_{i+2} + \left\lfloor \frac{y_i}{2} \right\rfloor) \quad (17)$$

Extraction process continues until all embedded units have undergone extraction procedure. Similar to the original algorithm, embedding unit selection order has to be determined by the secret key.

5. SYSTEM DESIGN FOR FPGA IMPLEMENTATION

Present section consists of the details regarding steganographic system meant for transforming a predefined resolution cover image into stego image. Various components of the system have been described in subsequent sections.

5.1 Top Level System Architecture

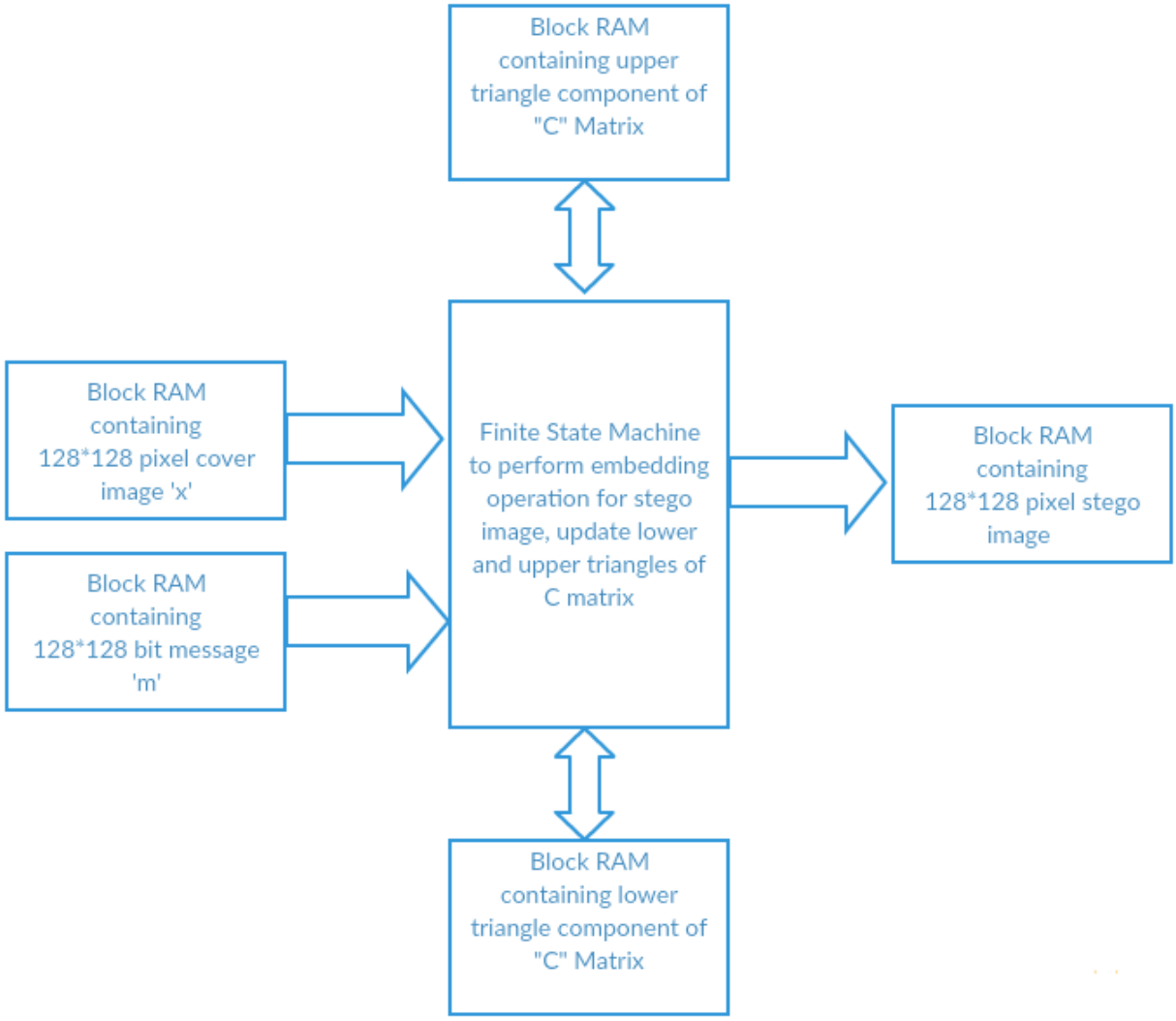

Fig. 6. Top level block diagram of steganographic system based on modified algorithm

The figure shows top level system architecture of the steganographic system. The block RAM is loaded with cover image ‘x’ and secret message ‘m’. Due to the constraints of resources available on target FPGA, image size is set at 128×128 pixels and secret message has also been chosen to be of 128*128 bits. Embedding in the image has been done at its full capacity of 1 bit per pixel.

To perform the functions of lower and upper triangular arrays, which assist in all over histogram compensating characteristics of one third probability algorithm, two block RAM with 256*256 no. of signed numbers has been allocated. They are initialized as all zeroes in each element.

Resultant stego image ‘y’ has 128*128 pixel resolution. It has also been assigned a block RAM and initialized as all zeroes.

Individual components i.e. block RAMs and FSM will be explained in subsequent sections.

5.2 Block RAM

Block RAMs are available in most families of FPGAs as atomic units. However, Synthesizer tools may infer to block RAM or distributed RAM (RAM made with logic slices’ elements instead of dedicated RAM). But creation of a block RAM is always prioritized. Present system design uses single port block RAMs in no change mode.

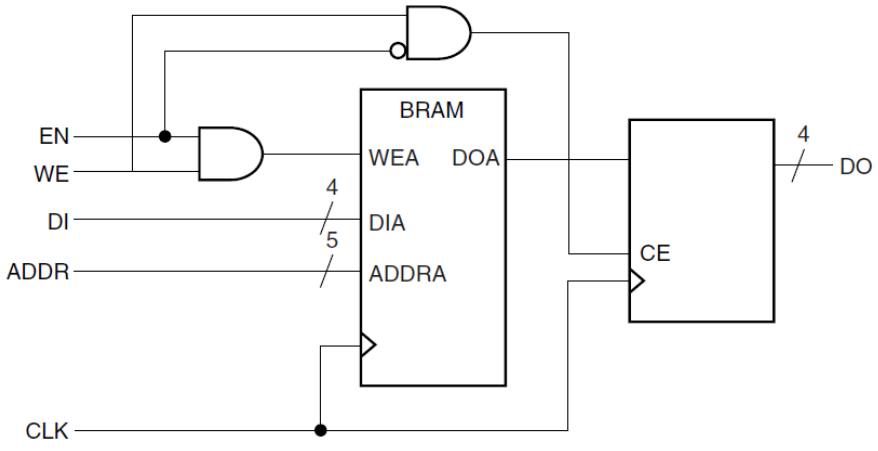

Fig. 7. Synthesized RTL schematic of a single port block RAM in no change mode

Present schematic shows a RAM with 4 bit cells and uses 5 bit wide address line to address all 32 4-bit cells. Proposed design uses 8 bit or 10 bit wide cells with 16 bit wide address lines. The signals from above schematic are as follows:

TABLE V SINGLE-PORT RAM IN NO-CHANGE MODE SIGNAL DESCRIPTIONS

Signal Name	Functionality
EN	Clock Enable
WE	Write Enable
DI	Data Input
DO	Data Output
ADDR	Address for data
CLK	Positive edge triggered clock for system

Working principal of such single port block RAM can be explained as: when positive edge of clock arrives, if signals EN and WE both are high then data at port DI will be written at the memory cell with address provided by signal ADDR. When signal WE is kept low, data at port DOA is reflected through latch at port DO. When EN is low, last data output is latched at port DO.

Storing intended data in these block RAMs is accomplished through initialization using binary or hexadecimal files. This method is most suitable for storing an image while implementing a design on FPGA. Images and secret message is converted in suitable binary or hex file using appropriate tools and block RAM is initialized through memory read commands of Verilog HDL. Cover image, secret message block RAMs have been initialized according to their respective binary file. Lower and upper triangular arrays, stego image block RAMs have been initialized using binary files containing all zeroes. Processing of data in RAM ‘x’ and ‘m’ is

completed using FSM during subsequent clock cycles and processed data will be written in RAMs representing stego image and arrays ‘lt’-‘ut’.

5.3 FSM based on Modified embedding algorithm

The Finite State Machine, which is responsible for executing embedding process, has been explained here. All transitions take place at the positive edge of system clock. Functionality of each state has been described in brief.

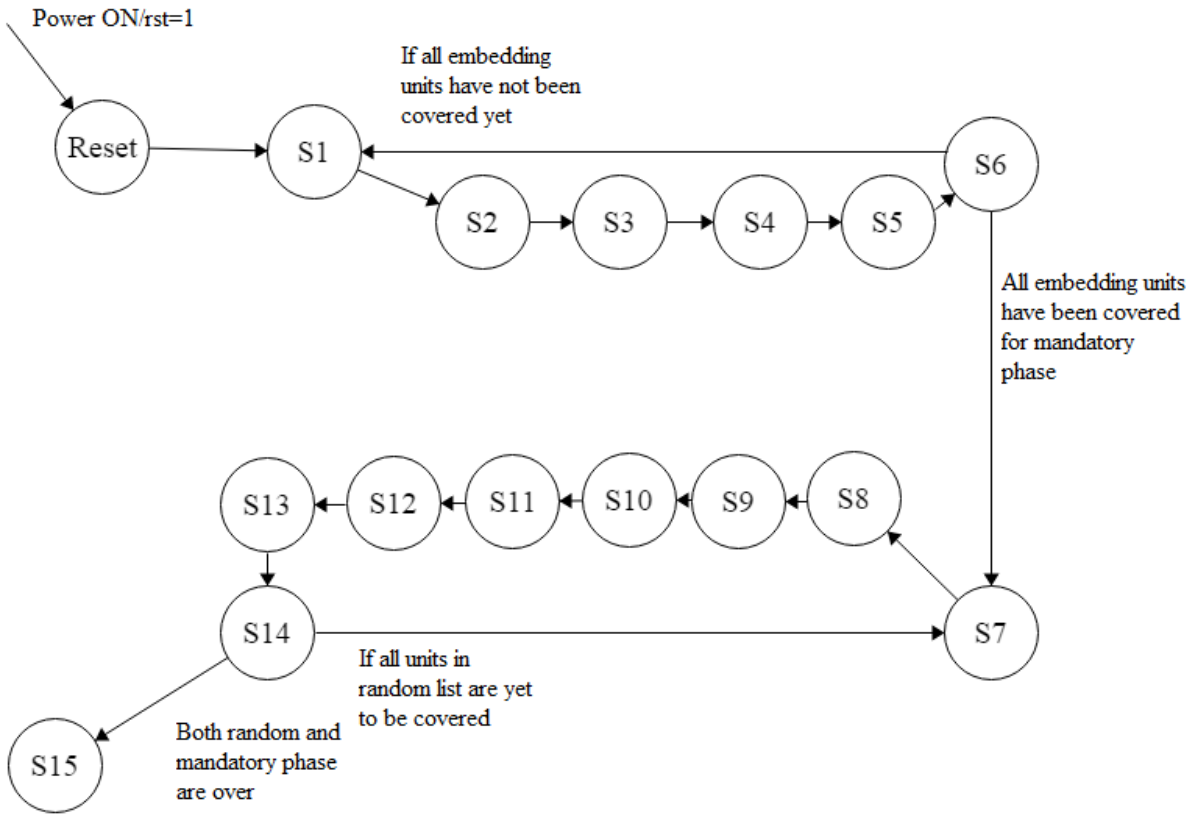

Fig. 8 Finite State Machine designed for embedding operations

1. **Reset**: FSM enters this state at power ON or when reset input is asserted. Address for all memories are initialized at 0.
2. **S1**: Values of cover image embedding unit and message embedding unit are copied into temporary registers reserved for each corresponding pixel.

3. **S2**: Mandatory embedding cases are applied according to table 4; resultant stego embedded unit and addresses for upper and lower triangle arrays are generated.
4. **S3**: Stego unit is written at its appropriate address in RAM designated for storing stego image.
5. **S4**: Entries from ‘lt’ and ‘ut’ arrays are copied in temporary registers using their addresses generated in state S2.
6. **S5**: Increment and decrement operations according to table are carried on the entries. Address of ‘x’ and ‘m’ are also incremented.
7. **S6**: If all the embedding units are finished with the mandatory phase embedding process then, FSM goes ahead to state s7, in which random phase of embedding starts. Otherwise process of mandatory phase continues for rest of units. Processed entries for ‘lt’ and ‘ut’ are also written back in their respective RAMs at appropriate addresses.
8. **S7**: Random embedding phase starts. All units with r = [1, 1, 1] are lead through state s8 and ahead.
9. **S8**: Addresses for ‘lt’ and ‘ut’ are calculated for all major and minor changes described in random phase in step 4 of section 4.2.2.
10. **S9** to **S12**: Contents from all addresses are fetched and stored in temporary registers for one action set in each embedding unit.
11. **S13**: All computation of random phase are carried out including score calculation. Based on the score, decision making about optimum action set is accomplished.
12. **S14**: Resultant values are updated in RAM for stego image ‘y’. If all the embedding units have been finished with the process then, FSM goes ahead to state S15, otherwise process continues until all units have not undergone random phase of embedding process.
13. **S15**: End state.

6. IMPLEMENTATION, RESULTS AND DISCUSSION

This section mostly covers the details of FPGA implementation including tools and methodologies used. Various parameters of FPGA implementation like device utilization summary, static timing analysis, simulation results and RTL diagrams have been included in the section. Following them, quantitative and qualitative analysis of obtained stego images has been described which includes images along with their histogram and qualitative parameters like PSNR, correlation coefficient etc.

6.1 Implementation Procedure

- After through design of top level system architecture, which has been explained in chapter 5, design is needed to be converted in synthesizable code in any HDL (Verilog or VHDL). Verilog HDL has been used for present design.
- In various stages, verilog coding of present design has been completed. The design follows bottom to top approach. It includes design and verification of basic functions like major and minor change, followed by memory design and initialization tasks.
- FSM has been coded through hybrid coding style for Melay-Moore state machines. All transitions have been designed to be performed at the positive edge of system clock.
- A testbench containing system clock, input embedding units and other verification functionalities has been designed and implemented to test the design at an embedding unit level.
- After functional verification of design, binary files for inputs and initialization have been applied to the design. These binary files are derived from some standard MATLAB test images. Output Stego images from their corresponding block RAMs have been transformed in to suitable binary format.
- These extracted binary files are imported in MATLAB. Subjective appearance of the image including their histograms have been derived for qualitative analysis.
- Qualitative analysis of these images includes computation of MSE, PSNR, correlation coefficient and entropy for each stego image.

- All the results have been summarised and comparatively analysed with the original one third probability steganography algorithm results.

6.2 FPGA Implementation Results

Present design has been developed using Xilinx ISE tool's version 13.2. Target FPGA family is Spartan 6. SP605 board, which includes Spartan 6 series XC6SLX45T FPGA, has been utilized for implementation. All of stages for FPGA implementation like synthesis, mapping, placement & routing and burning have been accomplished through Xilinx ISE tools with suitable constraints. Summary of the implementation has been given below.

6.2.1 Device Utilization Summary

TABLE VI DEVICE UTILIZATION SUMMARY

Slice Logic Utilization	Used	Available	Utilization
Number of Slice Registers	214	54,576	1%
Number of Slice LUTs	322	27,288	1%
Number of slice LUT used as Memory	0	6,408	0%
Number of occupied Slices	160	6,822	2%
Number of LUT Flip Flop pairs used	392		
Number of bonded IOBs	171	296	57%
Number of RAMB16BWERs	112	116	96%
Number of DSP48A1s	9	58	15%
Average Fanout of Non-Clock Nets	5.52		

Summary shows that design utilises minimal amount of available FPGA resources. Thus design has achieved efficiency. However, block RAM is a notable exception because storage of 3 (128*128) images and two (256*256) element arrays require significant amount of memory.

6.2.2 RTL Diagrams

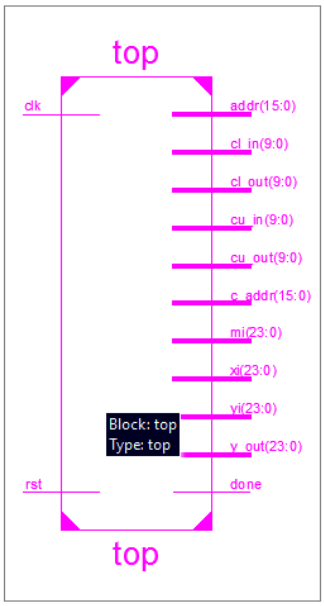

Fig. 9. Black box representation of steganographic system

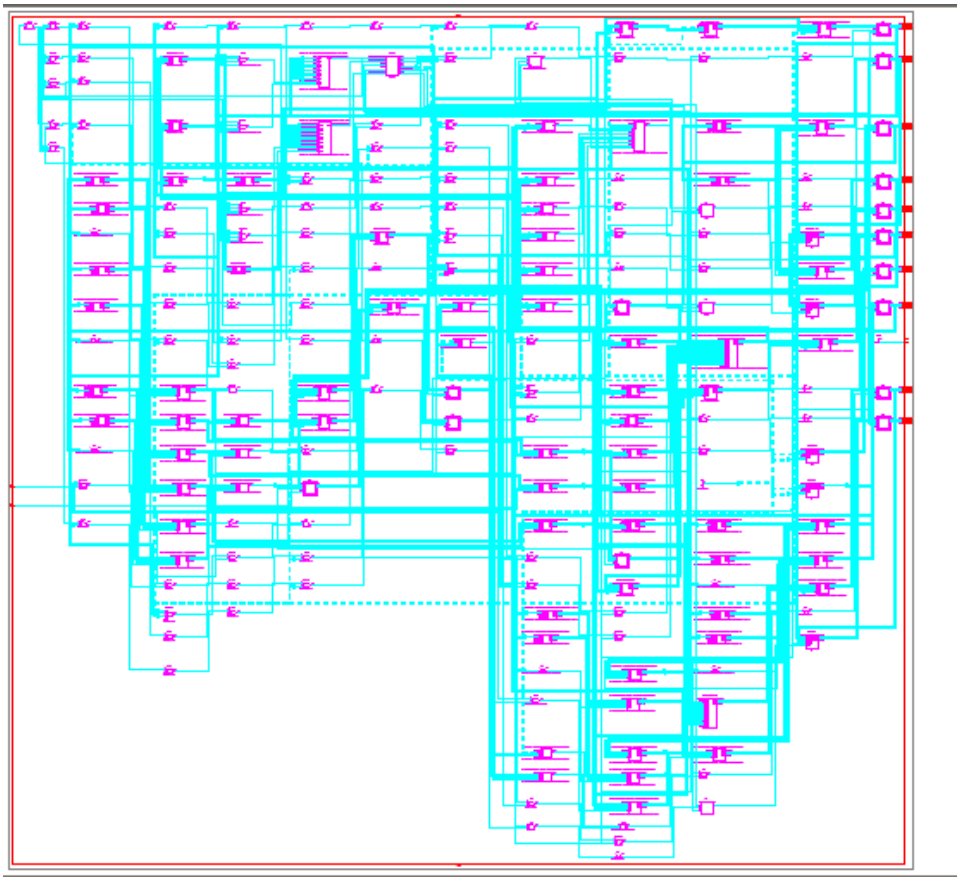

Fig. 10. Detailed RTL level representation of the system

6.2.3 Simulation Results for a Single Embedding Group

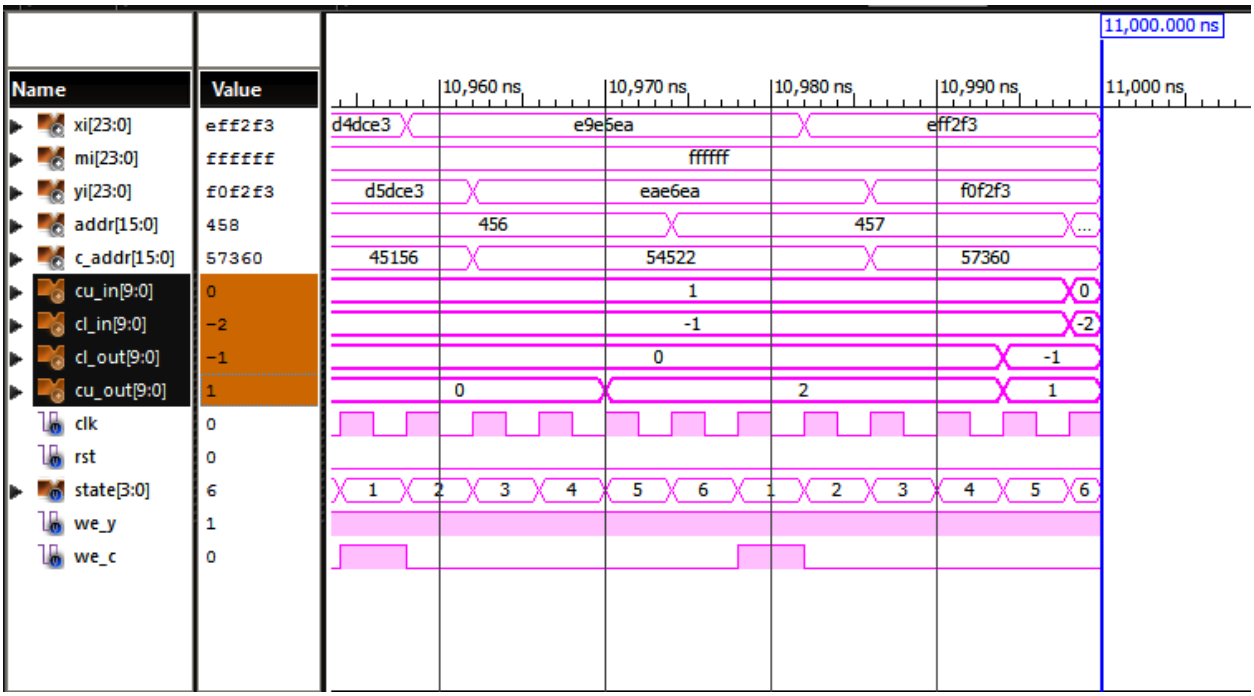

Fig. 11. Simulation results showing embedding foe a pixel group in mandatory phase

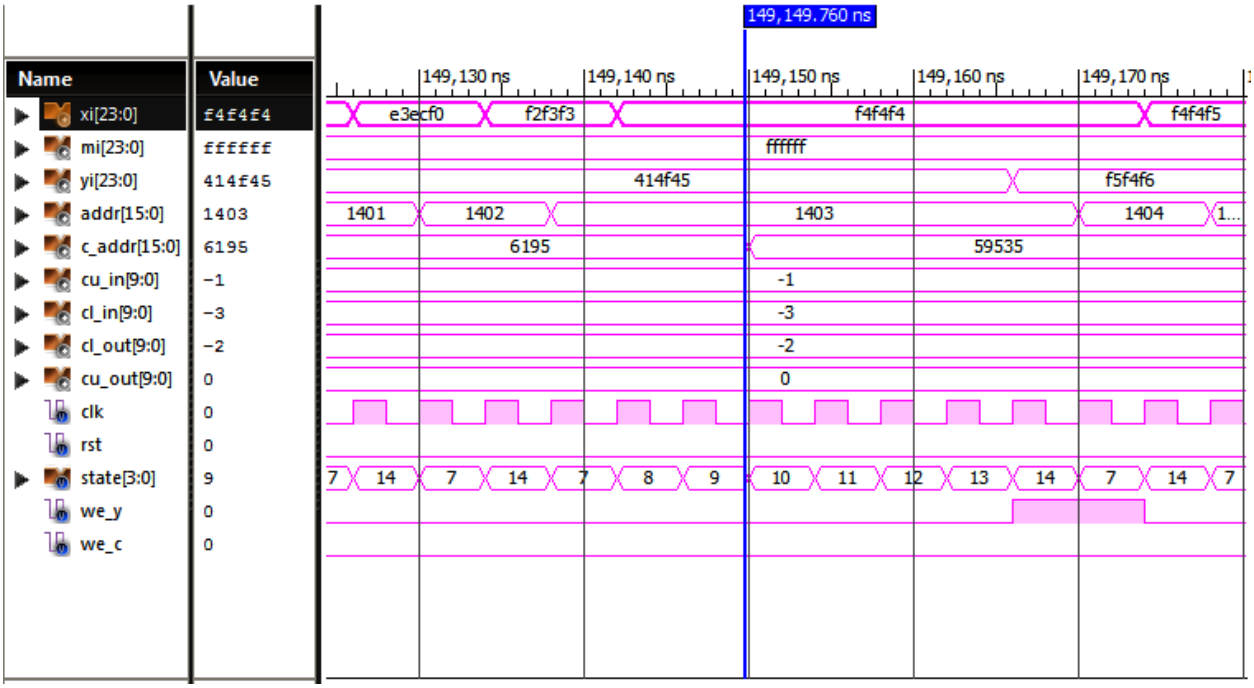

Fig. 12. Simulation results showing embedding foe a pixel group in mandatory phase

6.2.4 Timing Summary

- Minimum period: 9.526ns (Maximum Frequency: 104.971MHz)
- Minimum input arrival time before clock: 4.614ns
- Maximum output required time after clock: 4.743ns

6.3 Qualitative Analysis of Images

Qualitative analysis of obtained stego images includes visual examination for subjective correctness and checking for any visible alterations in the image. It also includes histograms comparison for checking resistance against HCF-COM attacks.

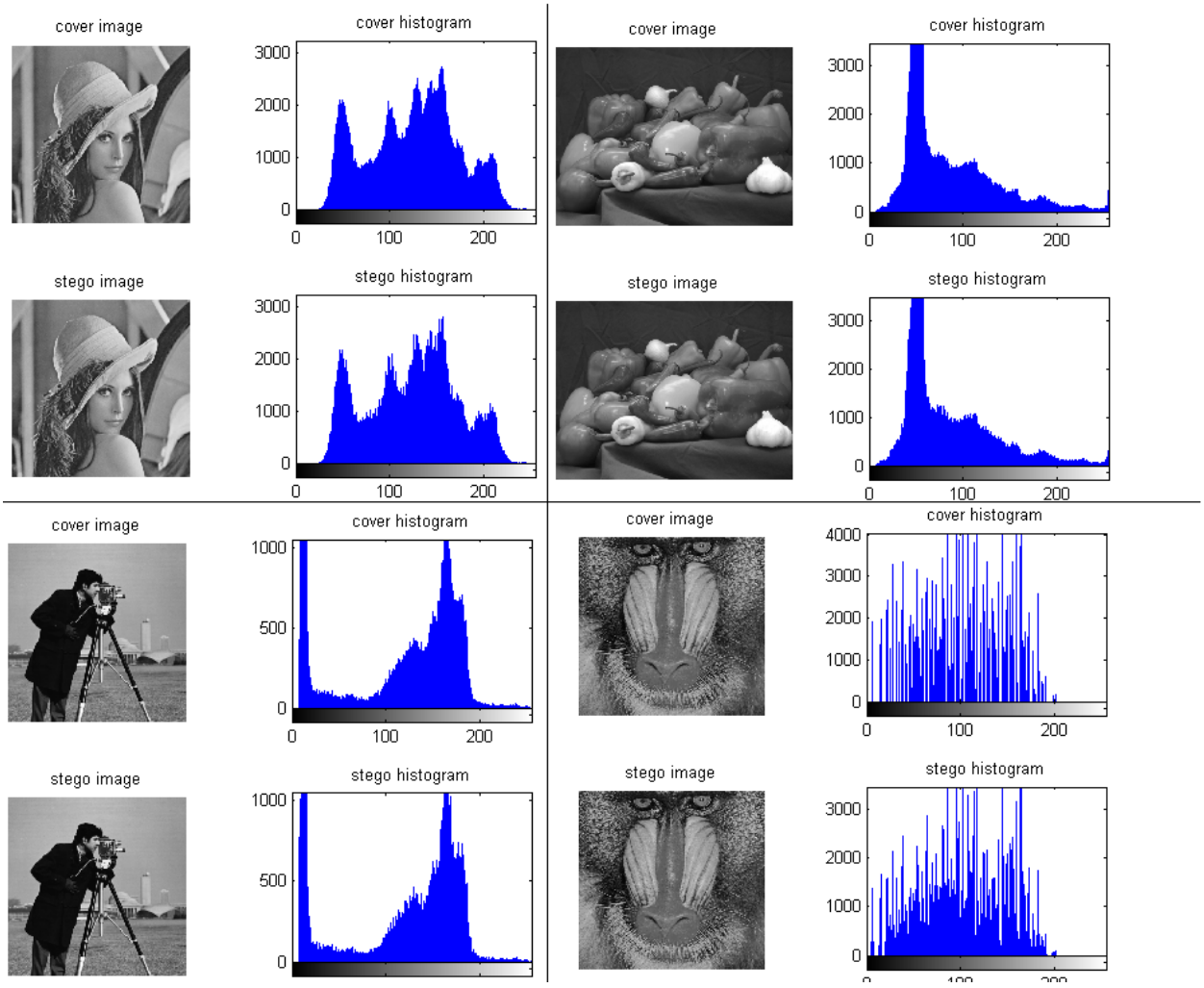

Fig. 13. Standard cover images and resultant stego images along with their respective histograms. From top right clockwise (i) lenna (ii) peppers (iii) baboon (iv) cameraman

6.4 Quantitative Analysis of Images

Several statistical and qualitative parameters must be included while analysing processed stego images. Present section includes some of performance parameters to analyse stego images for various requirements of steganography.

• **Mean Squared Error (MSE):** It is computed by performing pixel by pixel squared differences of the cover and stego-image. The computation can be expressed as follows:

$$Mean\ Square\ Error = \frac{1}{m*n}\sum_m \sum_n (x_{mn} - y_{mn})^2 \quad (18)$$

m: number of rows of cover image

n: number of column in cover image

x_{mn}: pixel value from cover image

y_{mn}: pixel value from stego image

Higher value of MSE indicates dissimilarity between cover image and stego image.

• **Peak signal-to-noise ratio (PSNR):** The quality of the stego-image when compared with the cover. It is measured in decibels. The higher the PSNR, the better the quality of resultant image. PSNR is computed using the following equation:

$$PSNR = 10log_{10}\frac{255^2}{MSE} \quad (19)$$

Where 255 is maximum grey level in an 8 bit image pixel.

Correlation Coefficient: It reflects statistical similarity between resultant stego and cover image. It should be 1 ideally. It is calculated by given formulae.

$$r = \frac{\sum_m \sum_n (x_{mn} - \bar{x})(y_{mn} - \bar{y})}{\sqrt{(\sum_m \sum_n (x_{mn} - \bar{x}))^2} * \sqrt{(\sum_m \sum_n (y_{mn} - \bar{y}))^2}} \quad (20)$$

Here, 'x' and 'y' are cover and stego images respectively with 'm' rows and 'n' columns. $\bar{x}, \bar{y}$ are arithmetic means for both images.

Entropy in Image: Image entropy is a quantity which is used to describe the `business' of an image, i.e. the amount of information which must be coded by any image processing algorithm. Ideally entropy for both cover and stego image should be equal.

$$e = -\sum_i p_i \log_2 p_i \quad (21)$$

Where p_i indicates probability associated with grey level 'i'.

Table 7 is the collection of various stats for standard test images of MATLAB.

TABLE VII QUANTITATIVE PERFORMANCE ANALYSIS OF MODIFIED EMBEDDING ALGORITHM

Image	**MSE**	**PSNR(dB) (From Original Algorithm) [13]**	**PSNR(dB)**	**Correlation coefficient**	**Cover entropy**	**Stego image entropy**
lenna	0.3334	52.9024	52.935	0.9999	7.4451	7.4437
peppers	0.3331	52.8968	52.9389	0.9999	6.9917	6.9914
cameraman	0.347	52.9310	52.9188	1	7.0097	7.0161
baboon	0.333	52.8968	52.94	0.9999	6.6777	7.1406

As it is evident from the results that correlation between cover and stego image is almost equal to 1, which is nearly ideal. Entropy of stego image is not lesser than cover image in any case, which suggests that information in cover image does not reduce by embedding algorithm. MSE and consequent PSNR levels are nearly equal or marginally better than results from Sarreshtedari et al., Which suggests subjective indifference between cover and stego images and suggests that stego image is imperceptible.

7. CONCLUDING REMARKS AND FUTURE SCOPE

7.1 Concluding Remarks

- The FPGA implementation of modified one third probability algorithm shows optimized implementation of a steganographic system.
- It has been one of the few examples of system level implementation of a novel spatial image steganographic algorithm, which is resistant to many newer steganalysis attacks. In modified algorithm, many operations and elements of original algorithm have been optimized or modified.
- The proposed system has been implemented on Xilinx SP605 board (Spartan 6 series XC6SLX45T FPGA).The qualitative and quantitative analysis of the results from the implemented system has been done. The performance of the system has been found to be at par with the original algorithm.

7.2 Future Scope

- Further hardware implementations for onion steganographic methods can be developed, which includes cryptography and other additional security layers.
- This thesis work presents an image based steganographic system implementation. However, steganographic system development based on audio, video or network protocols steganography techniques present vast opportunities.
- In the same implementation, power saving and performance enhancing techniques like pipelining and parallelism can be introduced.

REFERENCES

1. Halenár, Robert. "STEGANOGRAPHY USED FOR COPYRIGHT PROTECTION IN MATLAB ENVIRONMENT." *European Journal of Science and Theology* 10, no. 1 (2014): 253-262.
2. Johnson, Neil F., and Sushil Jajodia. "Exploring steganography: Seeing the unseen." *Computer* 31, no. 2 (1998): 26-34.
3. Krenn, Robert. "Steganography and steganalysis." *Retrieved September* 8, no. 2007 (2004): 2.
4. Leung, H. Y., L. M. Cheng, L. L. Cheng, and Chi-Kwong Chan. "Hardware realization of steganographic techniques." In *Intelligent Information Hiding and Multimedia Signal Processing, 2007. IIHMSP 2007. Third International Conference on*, vol. 1, pp. 279-282. IEEE, 2007.
5. Murdoch, Steven J., and Stephen Lewis. "Embedding covert channels into TCP/IP." In *International Workshop on Information Hiding*, pp. 247-261. Springer Berlin Heidelberg, 2005.
6. Mazurczyk, Wojciech, and Krzysztof Szczypiorski. "Steganography of VoIP streams." In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pp. 1001-1018. Springer Berlin Heidelberg, 2008.
7. Cheddad, Abbas, Joan Condell, Kevin Curran, and Paul Mc Kevitt. "Digital image steganography: Survey and analysis of current methods." *Signal processing* 90, no. 3 (2010): 727-752.
8. Kessler, G.,'Steganography: Hiding Data Within Data', accessed 21.03.2015, http://www.garykessler.net/library/steganography.html
9. Steganalysis, High Capacity Despite Better, and Andreas Westfeld. "F5—A Steganographic Algorithm." In *Information Hiding: 4th International Workshop, IH 2001, Pittsburgh, PA, USA, April 25-27, 2001. Proceedings*, vol. 2137, p. 289. Springer Science & Business Media, 2001.
10. Mazurczyk, Wojciech, and Luca Caviglione. "Steganography in modern smartphones and mitigation techniques." *IEEE Communications Surveys & Tutorials* 17, no. 1 (2015): 334-357.

11. Huang, Y. F., Tang, S., Yuan, J., ‘Steganography in Inactive Frames of VoIP StreamsEncoded by Source Codec’, *IEEE Transactions On Information Forensics And Security*, **6**,(2), June 2011
12. Shahadi, Haider Ismael, Razali Jidin, and Wong Hung Way. "Concurrent hardware architecture for dual-mode audio steganography processor-based FPGA." *Computers & Electrical Engineering* 49 (2016): 95-116.
13. Sarreshtedari, Saeed, and Mohammad Ali Akhaee. "One-third probability embedding: a new±1 histogram compensating image least significant bit steganography scheme." *IET image processing* 8, no. 2 (2014): 78-89.
14. Cetin, Ozdemir, and A. Turan Ozcerit. "A new steganography algorithm based on color histograms for data embedding into raw video streams." *computers & security* 28, no. 7 (2009): 670-682.
15. Ker, A.D.: ‘Improved detection of LSB steganography in grayscale images’. Lecture Notes in Computer Science, 2005, vol. 3200,pp. 583–592
16. Harmsen, Jeremiah J., and William A. Pearlman. "Steganalysis of additive-noise modelable information hiding." In *Electronic Imaging 2003*, pp. 131-142. International Society for Optics and Photonics, 2003.
17. Mielikainen, Jarno. "LSB matching revisited." *IEEE signal processing letters*13, no. 5 (2006): 285-287.
18. Mohanapriya, S. "Design and implementation of steganography along with secured message services in mobile phones." *International Journal of Emerging Technology and Advanced Engineering, ISSN* (2012): 2250-2459.
19. Schaathun, H., Chapter 2, Cryptography versus Steganography, “Steganography and Steganalysis”, Summary and Notes, Wiley-IEEE Press.
20. Xilinx Inc., (2016). *XST User Manual*. Retrieved from http://www.xilinx.com/itp/xilinx10/books/docs/xst/xst.pdf

LIST OF PUBLICATIONS

1. Pathak K., Bansal M. “A FPGA based Steganographic System implementing a Modern Steganalysis Resistant LSB Algorithm” , *Defence Science Journal* (communicated, reviewed, revised and sent back) as on 10th July, 2016

TURNITIN ORIGINALITY REPORT

601461017

by Kunjan Pathak

FILE	KUNJAN_PLAG.DOCX (972.3K)		
TIME SUBMITTED	10-JUL-2016 02:56PM	WORD COUNT	9034
SUBMISSION ID	688749658	CHARACTER COUNT	49320

601461017

ORIGINALITY REPORT

24%	14%	20%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	people.cs.uu.nl Internet Source	4%
2	Sarreshtedari, Saeed, and Mohammad Ali Akhaee. "One-third probability embedding: a new ±1 histogram compensating image least significant bit steganography scheme", IET Image Processing, 2014. Publication	3%
3	www.ukessays.com Internet Source	2%
4	Shahadi, Haider Ismael, Razali Jidin, and Wong Hung Way. "Concurrent hardware architecture for dual-mode audio steganography processor-based FPGA", Computers & Electrical Engineering, 2016. Publication	1%
5	www.krenn.nl Internet Source	1%
6	www.m-hikari.com Internet Source	1%
7	www.abbascheddad.net	