An FPGA Implementation of Secured Steganography Communication System

Dr. Ahlam Fadhil Mahmood
Computer Engineering
University of Mosul

Nada Abdul Kanai
Computer Engineering
University of Mosul

Sana Sami Mohmmad
Computer Center
University of Mosul

Abstract

Steganography is the idea of hiding secret message in multimedia cover which will be transmitted through the Internet. The cover carriers can be image, video, sound or text data. This paper presents an implementation of color image steganographic system on Field Programmable Gate Array and the information hiding/extracting techniques in various images. The proposed algorithm is based on merge between the idea from the random pixel manipulation methods and the Least Significant Bit (LSB) matching of Steganography embedding and extracting method.

In a proposed steganography hardware approach, Linear Feedback Shift Register (LFSR) method has been used in stego architecture to hide the information in the image. The LFSRs are utilized in this approach as address generators. Different LFSR arrangements using different connection unit have been implemented at the hardware level for hiding/extracting the secret data. Multilayer embedding is implemented in parallel manner with a three-stage pipeline on FPGA.

This work showed attractive results especially in the high throughputs, better stego-image quality, requires little calculation and less utilization of FPGA area. The imperceptibility of the technique combined with high payload, robustness of embedded data and accurate data retrieval renders the proposed Steganography system is suitable for covert communication and secures data transmission applications.

Keyword: Steganography, Steganalysis, LFSR, LSB matching, FPGA.

تنفيذ نظام تضمين للاتصالات السرية باستخدام البوابات القابلة للبرمجة حقليا

الخلاصة

أن إخفاء المعلومات هي فكرة لإخباء رسالة سرية في تغطية من الوسائط المتعددة أذ يتم إرسالها ضمنيا عبر الإنترنت. ويمكن أن تكون التغطية ضمن الصور ،الفيديو ،الصوت أو النص البياني. هذه الورقة، تقدم معمارية مطورة لإخفاء والاسترجاع في الصور الملونة على رقاقة البوابات القابلة للبرمجة حقليا . الخوارزمية المقترحة تستند على أساس الدمج بين فكرة تحديد عنصر الصورة بصورة عشوائية و مطابقة البت الاوطئ لاخفاء والاسترجاع.

في النظام المادي المقترح ، استخدام سجل الازاحة ذي التغذية الخلفية الخطية كمولد للعناوين . القيم الابتدائية والترتيبات المختلفة لسجلات الازاحة طبقت لتوليد العناوين العشوائية للاخفاء والاسترجاع .التضمين بطبقات متعددة نفذ كي يعمل على التوازي بثلاث وحدات. هذا العمل أعطى نتائج جيدة خصوصا في الانتاجية العالية ، جودة الصور ، يَتطلّبُ حسابَات قليلة ويشغل مساحة قليلة من رقاقة البوابات القابلة للبرمجة حقليا. غموض التقنية مع الحمولةِ ،

العاليةِ والمناعة في البياناتِ المُضَمَّنةِ وإسترجاعِ البياناتِ الدقيقِ يُجعلُ نظام الاخفاء والاسترجاع المُقتَرَحَ مناسبَ للإتصالاتِ السريةِ و تطبيقاتِ تحويلِ المعطيات الآمنةِ.

Introduction

The Internet has revolutionized the modern world and the numerous Internet based applications that get introduced these days add to the high levels of comfort and connectivity in every aspects of human life [1]. As the modern world is gradually becoming "paperless' with huge amount of information stored and exchanged over the Internet, it is imperative to have robust security measurements to safeguard the privacy and security of the underlying data.

Cryptography was created as a technique for securing the secrecy of communication and many different methods have been developed to encrypt and decrypt data in order to keep the message secret [2]. Unfortunately it is sometimes not enough to keep the contents of a message secret, it may also be necessary to keep the existence of the message secret. The technique used to implement is called steganography, meaning *covered writing*, dates back to ancient Greece [3]. It is the art and science of invisible communication. This is accomplished through hiding information in other information, thus hiding the existence of the communicated information.

Steganalysis is the reverse process of steganography, it is the mechanism of detecting the presence of hidden information in the stego media and it can lead to the prevention of disastrous security incidents [4].

Steganography techniques require two files: cover media, and the data to be hidden [5]. One of the commonly used techniques is the LSB replacement *where* the least significant bit of each pixel is replaced by bits of the secret till secret message finishes [1],[5],[6]. The risk of information being uncovered with this method is susceptible to all 'sequential scanning' based techniques [1], which is threatening its security.

LSB matching (also known as 1 embedding) is a minor modification of LSB replacement: if the secret data bit does not match the LSB of cover image, 1 is randomly either added to or subtracted from cover pixel value [2, 4]. Clearly, LSB matching avoids the asymmetry in LSB replacement; then as a consequence, it is much harder to detect. In [7], Xiaolong proposed the G-LSB-M scheme to improve the conventional LSB matching method.

The random pixel manipulation technique attempts at overcoming the sequential scanning problem, where pixels, which will be used to hide data are chosen in a random fashion based on a stego-key. However, this key should be shared between the entities of communication as a secret key [6].

In this paper, we propose an improved Steganography/ Desteganography algorithm to emebed/Extract spatial domain least significant bit (LSB) matching. The choice of embedding positions within a cover image mainly depends on a pseudorandom number generator with considering the relationship between the image content itself and the secret message. The hardware Steganography and it's Extract system are present, it is able to operate in real network environment.

This paper is organized as follows: Section 2 includes the LSB matching steganography algorithm. In section 3 the proposed LSB steganography is described in more detail, followed by detailed discussion on color space conversion. Section 4 includes the hardware FPGA design for steganography encoder/decoder based on random numbers logic. In section 5 Matlab experimental results to check the strength of proposed methods are tabulate. Section 7 includes conclusion and future work directions.

The LSB Steganography algorithm

LSB is a commonly used technique in image based steganography that it use the images as the cover media. There are two kinds of LSB Steganography: LSB replacement and LSB matching[1]. LSB replacement embeds a secret message into the cover image by replacing the LSB with message bit. LSB Matching does not simply replace the LSB of the cover image, if the bit must change, ±1 is added to the pixel value. Whether to use '+' or'-' is chosen randomly and has no effect on the hidden message. The extraction of the secret message for both LSB replacements

and LSB Matching work the same way: the LSB for each selected pixel is the hidden bit.

The original LSB matching algorithm can be formally described as follows:

$$P_s = \begin{cases} P_c + 1, \text{if } b \neq LSB(P_c) \text{and} (k > 0 \text{ or } P_c = 0) \\ P_c - 1, \text{if } b \neq LSB(P_c) \text{and} (k < 0 \text{ or } P_c = 255) \\ P_c \quad , \text{if } b = LSB(P_c) \end{cases} \quad \ldots\ldots (1)$$

Where Ps (resp. Pc) denotes a pixel value in the stego image (resp. cover image), b is the message bit to be hidden, and 1(is an i.i.d. random variable with uniform distribution on {-1, +1}. This process can be applied to all pixels in the image or only for a pseudo-randomly chosen portion.

The Proposed Steganography algorithm

In color image steganography algorithms, each pixel in an image is represented as a 24-bitmap value, composed of 3 bytes representing the R, G and B values for the three primary colors Red, Green and Blue respectively. Every color can be specified as a weighted sum of a red, green, and a blue component, the R,G,B byte have the same eye sensitive. If the color information is stored in the intensity and color format, some of the processing steps can be made faster. As a result, Cb and Cr provide the hue and saturation information of the color and Y' provides the brightness information of the color. Because eye is less sensitive to Cb and Cr, the proposed Steganography algorithm start from the idea of less eye sensitive to two Chrominance images then Luminance one, so it can embed secrete message in the less sensitive layer. A color in the R'G'B' color space is converted to the Y'CrCb color space using the following equation:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}$$

While the inverse conversion can be carried out using the following equation:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.164 & 0 & 1.596 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} + \begin{bmatrix} -222.9 \\ -135.6 \\ -276.8 \end{bmatrix}$$

The second idea in the proposed data hide procedure is the random pixel manipulation technique. However in this technique, a Pseudo Random Number Generator (PRNG) is used to select some pixels of the cover image. Then, the secret will be hided randomly in the channel of the selected pixels.

In our system six different (PRNG) are used, two for each channel (Y,Cb,cr) to set the pixel locations in three layers . In each channel pixel coordinates are defined by two integer random numbers, one specifying the row and other for the column. To ensure secure data transmission all six LFSRs are design to operate with different initial value (seed) numbers and different feedback switch connection. Linear Feedback Shift Register, three channel adderss generators are realised by using a shift register and linear feedback performed by several XOR (or XNOR) gates[8], as shown in Figure 1.

The Embedding Secret Message Procedure

After randomly selected Luminance embed pixel, compare it's LSB with secrete bit , if its equal embed '0' it in the 1-LSB of Cb Chrominance image of first otherwise embed '1' in Cb-LSB. At the same time the second secret bit is compare with first random select position in Cr-2LSB. The proposed algorithm can be described as follows:

$$P_s = \begin{cases} \text{if } b = LSB(Y - P_c), & flag = 0, 1 - LSB \text{ in } Cb = 0 \\ \text{if } b \neq LSB(Y - P_c), & flag = 1, 1 - LSB \text{ in } Cb = 1 \\ \text{if } b = 2 - LSB(Cr - P_c), & flag = 0, 1 - LSB \text{ in } Cr = 0 \\ \text{if } b \neq 2 - LSB(Cr - P_c), & flag = 1, 1 - LSB \text{ in } Cr = 1 \end{cases} \ldots(4)$$

So the high sensitive image remained unchanged, and only some of LSB of the Chrominance image are varied according to the flag bit in random manner to overcome sequential scanning in LSB replacing method. The flow chart of embed method are shown in Figure 2.

The Extracting Secret Message Procedure

The flowchart of extracting secret message is shown in Figure 3. There are five steps in this procedure. Now, they are described as follows:

Step 1. Use same Six LFSR which it have same seed and connection unit.

Step 2. Compute the value (*X*, *Y*) coordinate of needed pixel.

Step 3. Extract the first flag bit from Cb channel, read it if have zero value, extract secrete bit from luminance. Otherwise complement extract secret bit.

Step 3. Extract the second bit of random pixel of Cr channel.

Step 4. Complete the secret character.

Step 5. Take complete message and convert to Ascii by computing function *Ascii*.

Proposed FPGA Based LSB Steganography Embedder and Extractor.

The architecture of the steganography Embedder/Extractor system is shown in Figure 4. It consists of six parts: Address Generator (AG), Color Space Converter (CSC), block memory, Steganography Embedd Unit (SEU), Steganography Extract Unit (DSEU) and main controller (MCU). AG is responsible for calculating the addresses which are used to access the block memory using six nine bit LFSRs units. SEU is the kernel of this embedder, where LSB algorithm is implemented with a 3-stage pipeline. The block memory is a three single-port RAM. It is used for storage of three channel image pixels, and the data width is 8 bits. The MCU is an counters with conversion element that is used to synchronize the operation of units.

Address Generator(AG)

AG generates addresses to access the cover randomly for message bit hiding. This generator is composed of a six LFSRs operates in parallel , LFSRs-based bit generators are realised by using a D flip flop shift register and linear feedback performed by several XOR gates, as shown in Figure 1. The feedback network controlled by a several switching circuit. All Six LFSR operate in parallel and the important parameters are the number of bits, the XOR switch connection and the seed number. Table 1 shows the various values used.

Any one of these parameters are change the address are absolutely different, even a single bit change in the initial loading value there will be ultimate change in the following values. This property of LFSRs is vital in implementing the rich randomness in information hiding in cover images. In order to cover all the pixel locations of an image a six 9-bit LFSR can be used for embedding data in a 512x 512 cover color image two for each layer address. Figure 4 presents the hardware simulation result of six 9 – bit LFSR using wave scope unit in ISE10.1 software.

***Color Space Conversion Unit** (CSC)*

System Generator supports a black box block that allows VHDL code into Simulink and co-simulated with either ModelSim or Xilinx ISE Simulator. The VHDL code of the multiplierless architecture present in [9], which is based on distributed arithmetic (DA) principles are used to RGB ↔ YCbCr.

Block Memory

Most System Generator blocks that include memory or storage provide options to expose the reset and clock enable ports. In the proposed system three on-chip signal-Port block RAM are used one for each channel, each of the memories can afford write then read operations. Four externals memories, three Group for store three output Steganography images and last one to store the output detected message.

Steganography Embedd Unit (SEU)

The SEU acquires one memory byte form each channel and two bits of secrete message compare first bit with the Y-LSB bit if equal put flag to '0' otherwise '1' into Cb-LSB and complete the procedure mentioned in Equation 4 to embed second bit in Cr. The VHDL code are written and simulate using black box.

DeSeganography Embedd Unit (DSEU)

The opposed operation of SEU are perform on this unit, it is extract the secret message from three image layers and give the result to external memory.

Experimental Result

The proposed embed and extract algorithms are implemented using MATLAB, and have successfully applied it to standard test images varying from complex (Baboon) to smooth (boat). The proposed experiment, will take a sequential bit string as the secrete message and adopt peak signal-to-noise ratio (PSNR) value as measurements of image quality and embedding capacity, respectively. The steganography versions of test images (sized 512x512, 24-bit color scale) are placed as examples in Figure 5, where the visual qualities of them are satisfactory.

FPGA Implementation Results

The Xilinx XtremeDSP kit is a versatile FPGA development tool useful for rapid prototyping Xilinx FPGA realized digital signal processing (DSP) applications. The XtremeDSP kit consists of a Nallatech Ben-ONE main board, a Nallatech Ben-ADDA daughter card, and is fitted with a Xilinx Virtex-II XC2VP30 FPGA chip. The XtremeDSP hardware supports Matlab/Simulink based logic synthesis employing the Xilinx System Generator (XSG) hardware description language (HDL) synthesis tool. The simulation Block diagram is shown in Figure 6.

The synthesis results are shown in Table 2. It can be seen that the architecture may process more than 300 million pixels per second.

The tests of the proposed algorithm implemented in software that served as the basis of this architecture were carried out in a computer with a intel(R) core(TM)2 Duo CPU at 2 GHz and 2.96 GB of RAM memory, Windows XP operating system with Matlab version 7.4. As it can be seen in Table 3, the software implementation runs on average in 2.6727seconds per image, whereas the hardware architecture achieves an average of 0.00418 seconds per image. That means, the hardware architecture perform 639 times faster than the software implementation.

Conclusions

In this paper, a proposed reversible steganography scheme has been presented, different from the latest schemes using adaptive LSB method and LFSR techniques. To preserve the statistical and visual features in cover images, the proposed embed the secret message into the less sensitive regions adaptively according to sensitivity of the human visual system. It uses (Cr and Cb) only for embed flag, so it remained luminance channel without any change.

The proposed model is more secure against attacks because it depends on a list of security parameters. These security parameters are the six different LFSR that is used for generate the random embed address, it have six different seed number and multifarious connection unit as well as the embedding data is the flag not the absolute data. These parameters must be known to hackers to extract secret message from the cover image, it extract in the receiver side that it used same FPGA system only.

According to the experimental results, the proposed reversible scheme provides a higher capacity of the performance of multilayer embedding and achieves better image quality for steganography images. In addition, the computational cost of the proposed scheme is small, according to the operating frequency it can be used to embed and extract message in video file

References

1. N. Meghanathan1 and L. Nayak", Steganalysis Algorithms for Detecting the Hidden Information in Image, Audio and Video Cover Media" , International Journal of Network Security & Its Application (IJNSA), Vol.2, No.1, January 2010, PP. 43-55.

2. W. Luo, Member, IEEE, F. Huang, Member, IEEE, and J. Huang, Senior Member, IEEE ," Edge Adaptive Image Steganography Based on LSB Matching Revisited", IEEE Transactions on Information Forensics and Security, VOL. 5, NO. 2, JUNE 2010, PP.201-214.

3. C.Lin, C. Chang, W. Lee, and J. Lin," A Novel Secure Data Hiding Scheme Using a Secret Reference Matrix", Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kyoto, Japan, 12-14 September 2009.

4.B. Xia, X. Sun, J. Qin," Steganalysis Based on Neighbourhood Node Degree Histogram for LSB Matching Steganography", International Conference on Multimedia Information Networking and Security, IEEE Computer Society Washington, USA ,18-20 November , 2009, PP.79-82.

5.H.B.Kekre, A.A. Athawale , S.A. Patki ," Improved Steganalysis of LSB Embedded Color Images based on Stego-Sensitive Threshold Close Color Pair Signature", International Journal of Engineering Science and Technology (IJEST), Vol. 3 No. 2 Feb 2011, PP. 836-842.

6. X. Yu, and N. Babaguchi ," An Improved Steganalysis Method of LSB Matching", International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IEEE Computer Society Washington, , USA , 15-17 Aug. 2008 ,PP. 557-560.

7. X. Li, B. Yang, D. Cheng, and T. Zeng," A Generalization of LSB Matching", IEEE Signal Processing Letters, VOL. 16, NO. 2, February 2009 , PP. 69-72.

8. R. Mita, G. Palumbo and M. Poli , " Pseudo-random sequence generators with improved inviolability performance", IEE Proc.-Circuits Devices Syst., Vol. 153, No. 4, August 2006, PP.375-382.

9. A. F. Mahmood and A.M. salih " Implementation of Multiplierless Architectures for Color Space Conversions on FPGA" , accepted in Al-Rafidian Engineering journal.

Table 1: LFSRs Parameters

LFSR	Bit No.	closed Switchs	seed
LFSR1	9	1,5	3F
LFSR2	9	1,2,3	4F
LFSR3	9	1,2,8	33
LFSR4	9	1,2,3	31
LFSR5	9	1,6	54
LFSR6	9	1,5,7	34

Table 2: Results of synthesis in FPGA

Slices	2,411 out of 13,696 17%
Frequency(MHZ)	107.75
Throughput(Mbps)	3 pixel /Clock cycle
Latency	0.065μsec

Table 3: Comparison of times (in Seconds) of the Software and FPGA Hardware implementations

Image	Size	Software	Hardware	Ratio
babbon	512*512	2.6727	0.00418	639.4
pepper	512*512	2.6727	0.00418	639.4
boat	256*256	1.3421	0.00209	642.1
sailboat	512*512	2.6727	0.00418	639.4

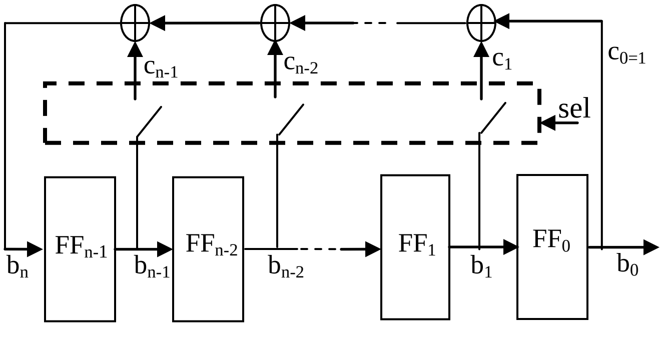


Figure 1: Block diagram of an n-bit Linear Feedback Shift Register (LFSR)

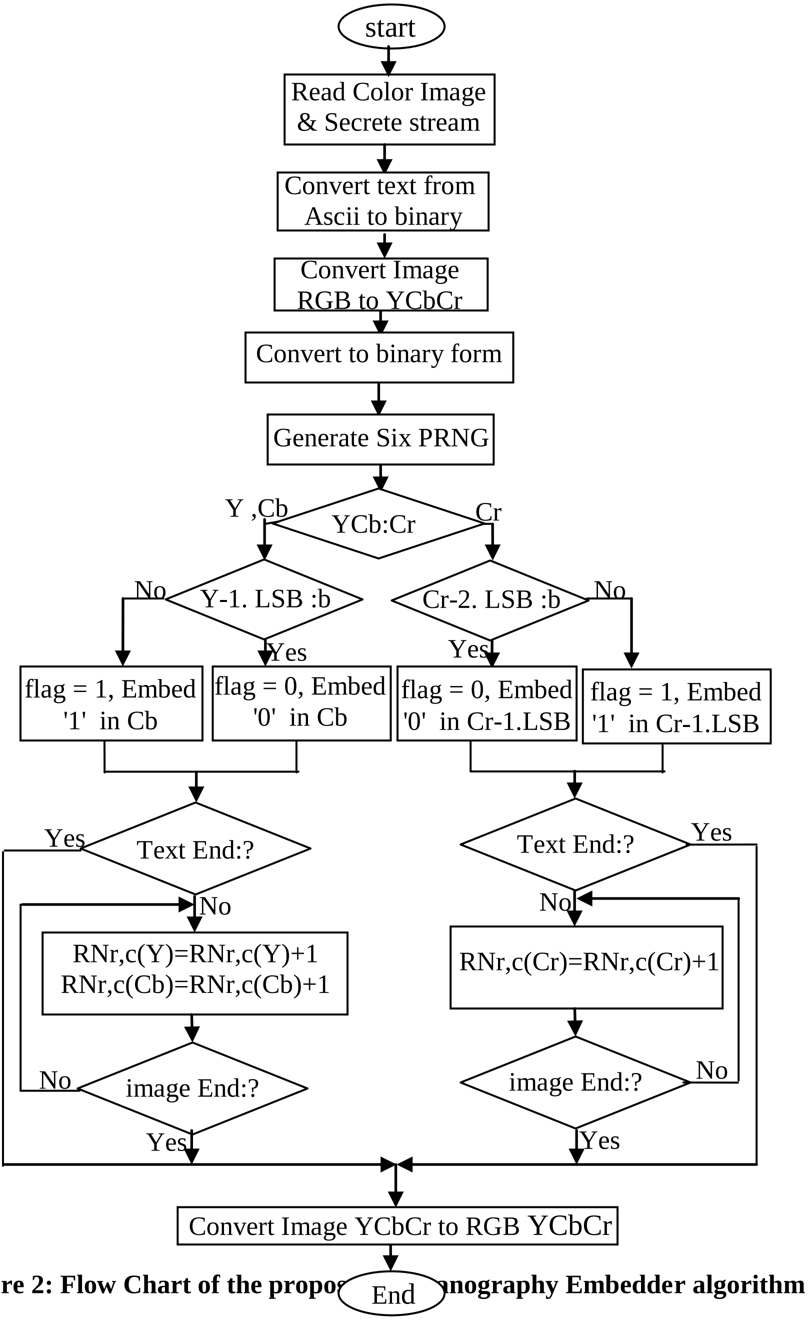

start
Read Color Image & Secrete stream
Convert text from Ascii to binary
Convert Image RGB to YCbCr
Convert to binary form
Generate Six PRNG
Y ,Cb
YCb:Cr
Cr
No
Y-1. LSB :b
Yes
Cr-2. LSB :b
Yes
No
flag = 1, Embed '1' in Cb
flag = 0, Embed '0' in Cb
flag = 0, Embed '0' in Cr-1.LSB
flag = 1, Embed '1' in Cr-1.LSB
Yes
Text End:?
No
Text End:?
Yes
No
RNr,c(Y)=RNr,c(Y)+1
RNr,c(Cb)=RNr,c(Cb)+1
RNr,c(Cr)=RNr,c(Cr)+1
No
image End:?
Yes
image End:?
No
Yes
Convert Image YCbCr to RGB YCbCr
End

Figure 2: Flow Chart of the propo nography Embedder algorithm

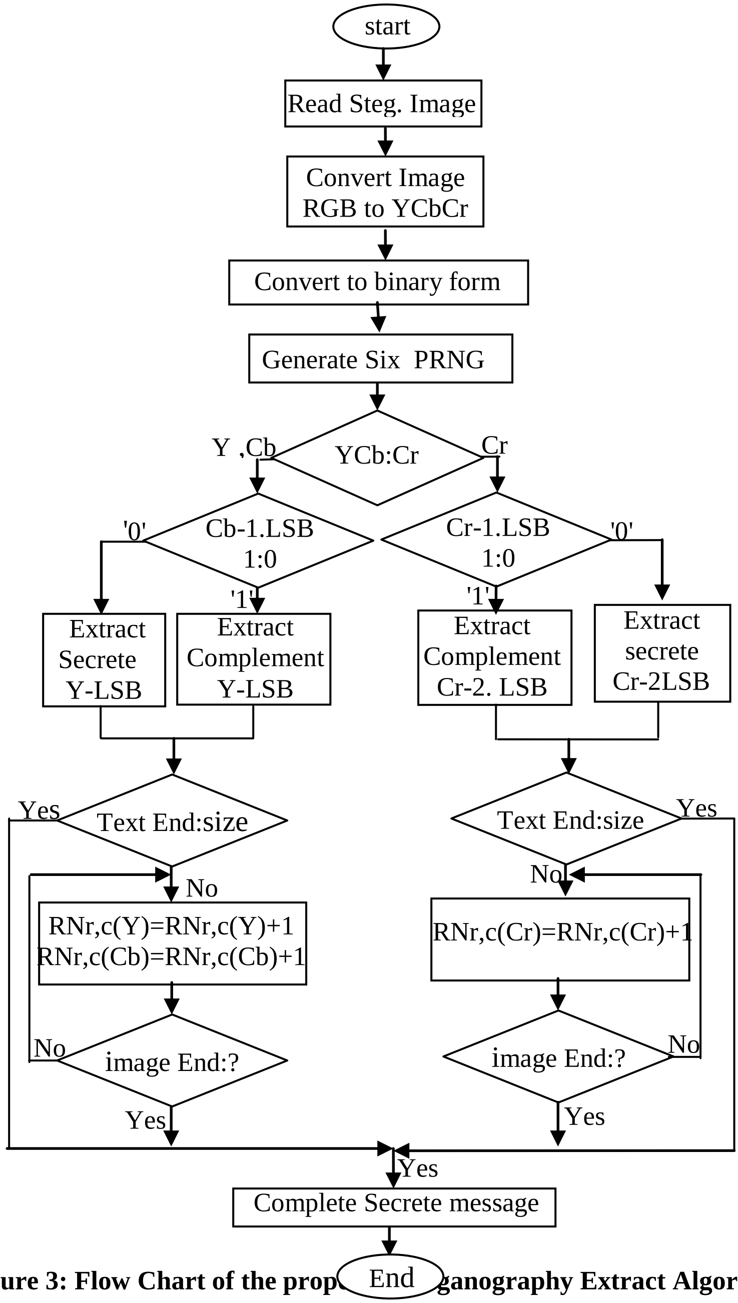


Figure 3: Flow Chart of the prop ganography Extract Algorithm

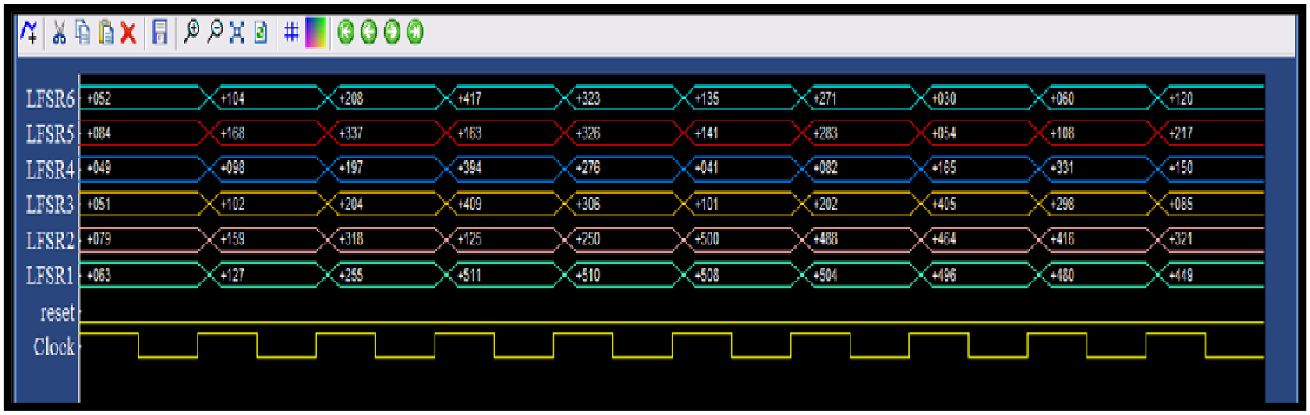

LFSR6 +052 +104 +208 +417 +323 +135 +271 +030 +060 +120
LFSR5 +084 +168 +337 +163 +326 +141 +283 +054 +108 +217
LFSR4 +049 +098 +197 +394 +276 +041 +082 +165 +331 +150
LFSR3 +051 +102 +204 +409 +306 +101 +202 +405 +298 +085
LFSR2 +079 +159 +318 +125 +250 +500 +488 +464 +416 +321
LFSR1 +063 +127 +255 +511 +510 +508 +504 +496 +480 +449
reset
Clock

Figure 4: Simulation results of Six parallel operation LFSR with different initial value(seed) and connection network

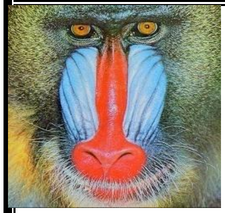
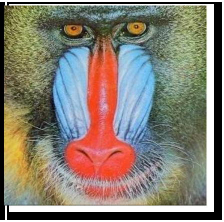
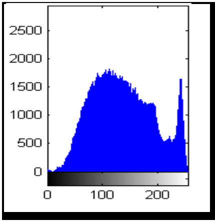
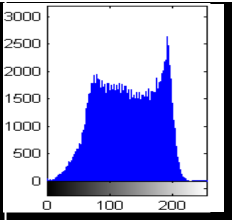
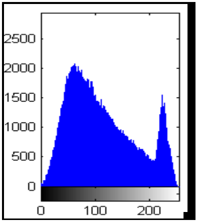
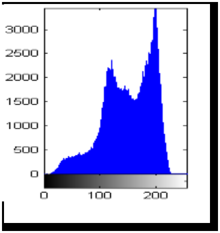
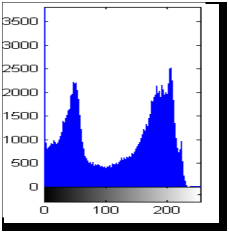
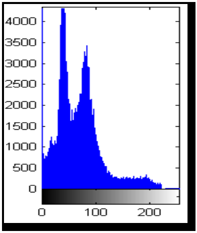
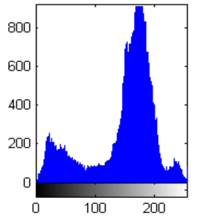
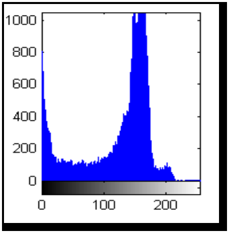
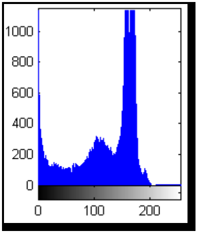
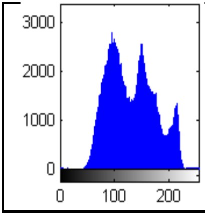
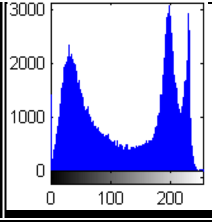
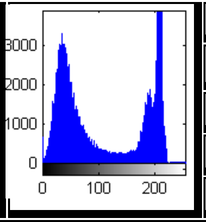
Cover Images	Steg-Images	Steg-Histogram			Pay-load	PSNR (dB)
		Red Channel	Green Channel	Blue Channel		
					10%	51.3
					30%	50.8
					50%	49.4
					70%	48.6
					100%	47.9
					10%	52.1
					30%	51.9
					50%	51.6
					70%	50.4
					100%	49.7
					10%	51.1
					30%	50.7
					50%	50.1
					70%	49.1
					100%	48.2
					10%	51.5
					30%	51.1
					50%	50.8
					70%	49.9
					100%	49.1

Figure 5: Steganography System Results

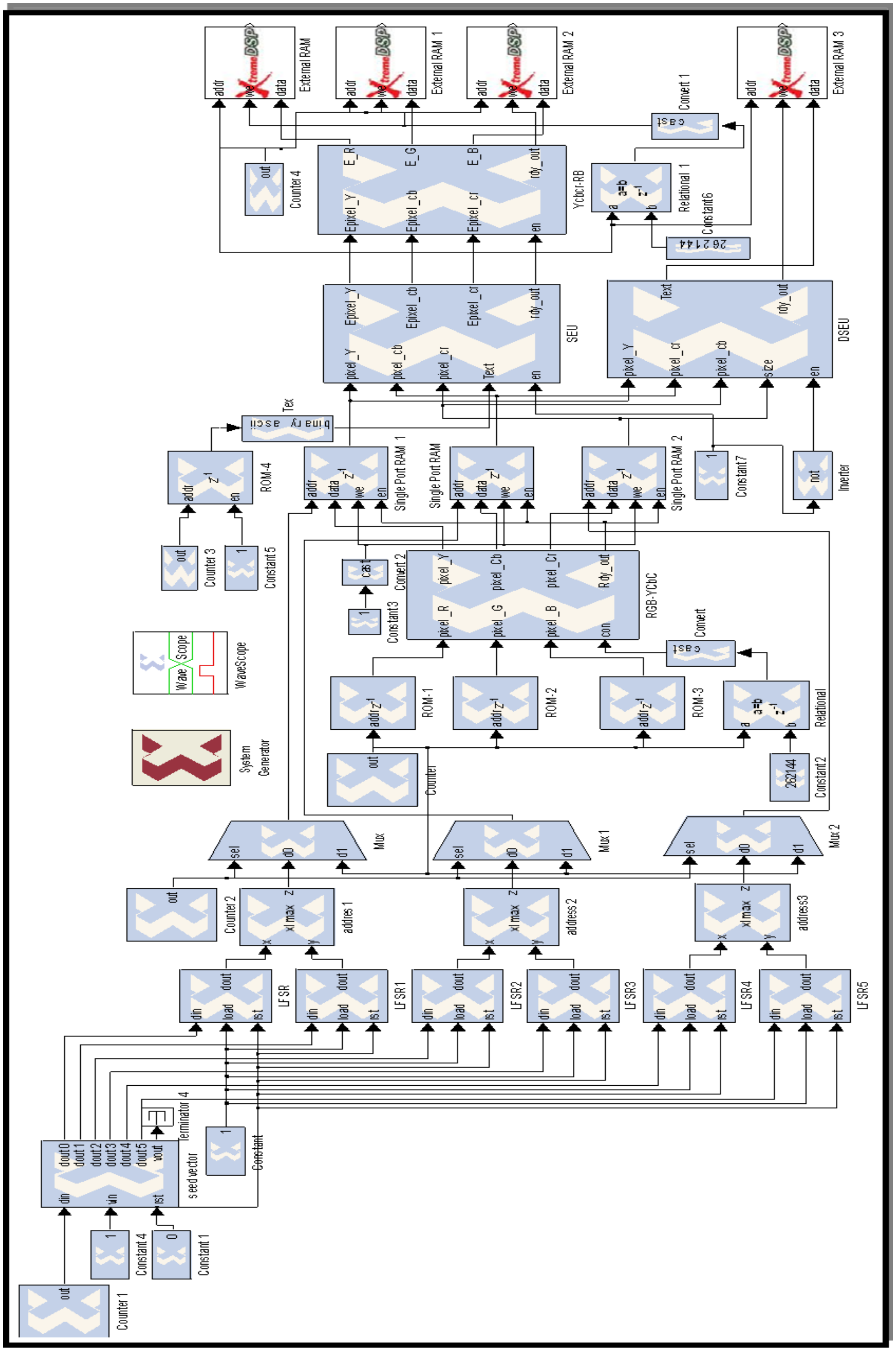


Figure 6: Architecture of the Steganography System