DOI: 10.21917/ijme.2018.0098

FPGA IMPLEMENTATION OF LSB-MR BASED STEGANOGRAPHY ALGORITHMS

K. Nandhini and S. Arivazagan
Department of Elecctronics and Communication Engineering, Mepco Schlenk Engineering College, India

Abstract

In network security Image Steganography is one of the crucial data hiding technique. In this paper a new architecture was proposed for with and without pipelining technique for LSB Matching Revisited steganography algorithm and it is implemented in FPGA using Verilog HDL. The design motto is to increasing the speed, reducing the clock cycle by performing embedded and fetching operation parallel, reduces the complexity in both transmitting and receiving side and also we can send more confidential message due to the large cover image (128×128×8) size which we have used here. In the transmitting side data hiding is performed and receiving side extraction is performed. Whole entire process is takes place in both hardware and software, finally, result was analysed for both software and hardware level. From the outcome investigation, Pipelined model will give a superior outcome while contrasting with non-pipelining mode as far as less inserting time contrasted with the non-pipelined mode.

Keywords:
Verilog HDL, LSBMR, FPGA, Pipelining

1. INTRODUCTION

Steganography is the art of hiding message, nowadays internet surfing is crucial in our daily life. Transmitting an information through internet should be Encrypted. Different information concealing strategies are, for example, Cryptography, Steganography and watermarking which are referred to beneath. In Cryptography, encryption is done only by changing over unique message into another through encoding procedure thus the message can't read by the unapproved party. Steganography came in to the picture when one needs to hide the large content (secret message size). Simply steganography performs by replacing or inserting bits or bit of confidential message behind host medium The cover medium are image/video, text, audio and file. By choosing the host medium quality of the stego image is increased. The goal of steganography is to secure the communication and replacing the LSBs or LSB of the cover-image with the message bits. Due to this simple replacing technique there exists an imbalance in the embedding distortion in the stego image. So least significant bit (LSB) method was modified and the new modified LSBMR method improved imperceptibility. Our work is organized in the following sections. The section II shows the previous work done by the early researchers and section III shows our proposed system in detailed manner about implementation of LSBMR steganography algorithms with and without pipelining technique in FPGA. Result and discussion and the conclusions are in section IV and V respectively.

2. EXISTING METHOD

According to Ling et al. [1] when implementing extensive computer vision applications in image processing data hiding platforms is still a challenging task. Field-programmable gate arrays (FPGAs) provide a suitable technology to accelerate image processing by customized hardware. Pixel-based modules are prefer for simple pre-processing tasks while image is processed by FPGA. LSB replacement method involves a message to be embedded behind the host medium and replacing the LSBs of the cover image with message bits. This method performs increment or decrement pixel values by one for entire image pixel, while odd values are decreased by unity. Due to this, the existence of an imbalance in the embedding distortion within the stego image is visible. Introduction of structural asymmetry reduces the difficulty to detect and identify the existence of hidden message. In this way LSB matching method modify the cover images. This LSB matching method perform matching operation between secret message and cover image based on that, random addition or subtraction of one is done, if there is any mismatch between the message bit and LSB of the cover image from the cover pixel value [2]. After Luo et al. [3] introduced Edge Based LSB matching method for gray scale images. The test is carried out on 6000 natural image dataset. The overall performance measures of Edge Based LSB matching is correlated with Luo et al. [3], LSB Matching (LSBM), LSB Matching Revisited (LSBMR) [4], PVD [5], improved version of PVD (IPVD) [6], adaptive edges with LSB (AE-LSB) [7], and hiding behind corners (HBC) [8]. The satisfactory of stego image is discern out the usage of parameter peak signal-to-noise ratio (PSNR) cost. Embedding efficiency can be stepped forward by way of embedding at the rate of 1bit LSB matching like Mielikainen [9], which reduced the ENMPP with the equal message period from 0.5 to 0.375. The desire of whether or not to add or subtract one to/from a pixel fee in their technique is predicated on both the original pixel values and a couple of two consecutive secret bits. However, this approach of embedding can't be implemented on saturated pixels (i.e. Pixels with values 0 and 255), which is one of the drawbacks of this technique. Then, the method of LSB matching is proposed by Li et al. [10] with the equal ENMPP for the equal embedding charge the use of sum and distinction covering set (SDCS). Another approach became proposed by using Zhang et al. [11] to enhance the drubbing efficiency of LSB matching, the usage of a mixture of binary codes and moist paper codes, The embedding efficiency of this technique can reap the upper sure of the generalized ±1 embedding schemes.

3. PROPOSED METHOD

Among available steganography techniques, the simplest method is LSB steganography. Advanced steganography schemes exploit the LSB method for embedding and extraction. A simple LSB result contains an imbalance in the embedding side and so the stego image get distorted. To predict this imbalance problem, a new method was introduced, where LSBMR method cannot simply is replacing the LSB bit in cover image of all entire pixels as does by LSB algorithm. Here four conditions can be checked whether cover image LSB bit are equal with message bit. Although, high level programming languages, such as, C, C++, Matlab etc. are used to simulate steganography algorithm in software and the throughput achieved from these schemes are inadequate. Therefore, hardware implementation of steganography algorithms is the right choice in order to achieve high system throughputs.

3.1 METHODOLOGY

3.1.1 Embedding Phase:

This phase explains about the process of LSBMR (Least significant bit matching revisited) data hiding in a gray scale image. In normal LSB steganography method simply LSB bit of the host can be replaced by comparing with the message bit, whereas the LSBMR based data embedding involves the checking of four conditions based on that secret data can hide in the host image. In this LSB matching revisited based data hiding process the two pixel pairs are selected for the data embedding, although checked four below stated conditions. Similarly perform this process for all the entire pairs of the pixels in the host image. Therefore this methodology will retain the host image quality when compared to the ordinary LSB embedding technique. The LSBMR steganography algorithm is given as follows:

Input: A pair of cover image pixel x_i, x_{i+1}

Two message bits: m_i, m_{i+1}

Output: A pair of stego image pixels y_i, y_{i+1}

If m_i = LSB (x_i)

If $m_{i+1} \neq f(x_i, x_{i+1})$

$y_{i+1} = x_{i+1} \pm 1$

else

$y_{i+1} = x_{i+1}$

end

$y_i = x_i$

else

if $m_{i+1} = f(x_{i-1}, x_{i+1})$

$y_i = x_i - 1$

else

$y_i = x_i + 1$

end

$y_{i+1} = x_{i+1}$

end

In case-1 the 2-criteria is checked in such a way that if the LSB bit of x_i is same as the secret message bit m_i and followed by check m_{i+1} is same as the calculated functional LSB bit f(.) bit using the Eq.(1), if LSB f(.) and m_{i+1} is not equal then stego image pixel y_{i+1} is replaced by randomly adding ±1 to x_{i+1}. If m_{i+1} is same as the calculated functional LSB f (.) then stego image pixel is replaced by y_{i+1} is same as cover image pixel x_{i+1}, for both the case y_i is same as x_i does not need any modification.

In case-2 again 2-criteria is checked in such a way that if the LSB bit of x_i is not same as the secret message bit m_i and followed by check m_{i+1} is same as the calculated functional LSB bit f(.) bit using the Eq.(1), if LSB f(.) and m_{i+1} is not equal then stego image pixel y_i is replaced by adding +1 to x_i. If m_{i+1} is same as the calculated functional LSB f (.) then stego image pixel y_i is replaced by subtracting -1 to x_i, for both the case y_{i+1} is same as x_{i+1} does not need any modification. The whole procedure of information is clarified with illustration. Consider match of pixel sets $x(0,0) = 49$, $x(0,1) = 44$ and classified data $m_i = 1$ for instance. Whenever LSB (49) = 1 and LSB(44) = 0, then both $x(0,0)$ and $x(0,1)$ is substituted in Eq.(1) to give $f(49,44) = 0$. The case-1 are met, and this result is modified pixels as $x_{(0,0)} = 49$ and $x_{(0,1)} = 44$. Similarly the two cases are verified iteratively for data embedding in row wise and column wise of entire image pixels pair are processed. Pair of pixel modification is takes place in LSBMR concept to embedded data. Examples of embedding and extracting message bit pair into cover image pixels pairs is shown in Table.1 and Table.2 respectively.

$$f(x_i, x_{i+1}) = LSB\left(\left\lfloor \frac{x_i}{2} \right\rfloor + x_{i+1}\right) \quad (1)$$

3.1.2 Extracting Phase:

In this segment the implanted information is recouped from stego picture. Mystery picture can be gotten by utilizing connection which is given as follows:

Input: Stego image bits: y_i, y_{i+1}.

Output: Secret image pixels s_i, s_{i+1}.

S_i = LSB (y_i)

$S_{i+1} = f(y_i, y_{i+1})$

Table.1. Example of embedding message bit pairs (m_i, m_{i+1}) into cover image pixel pairs (x_i, x_{i+1}), resulting in stego image pixel pairs (y_i, y_{i+1})

x_i	x_{i+1}	m_i	m_{i+1}	y_i	y_{i+1}
1	1	0	0	2	1
1	1	0	1	0	1
1	1	1	0	1	0 or 2
1	1	1	1	1	1
1	2	0	1	2	2
1	2	1	0	1	2
1	2	1	1	1	1 or 3

Table.2. Example of extracting message bit from stego image pixel pairs (y_i, y_{i+1})

y_i	y_{i+1}	$LSB(y_i)$	$f(y_i, y_{i+1})$
162	151	0	0
162	150	0	1
161	150	1	0
163	150	1	1

3.2 HARDWARE IMPLEMENTATION

3.2.1 Architecture of Embedding Side:

An In this proposed method first cover and secret image pixel is converted into binary format using MATLAB and it can be stored as a .coe file and stored image as a binary in a CIP and SIP BRAM using an IP core generator. This phase contains counter module, which is acting as an address generator to access pixels which is stored in CIP and SIP BRAM, clock is used to activate logical operation block to perform logical operations, functional block to perform the function which is in the Eq.(1). This block performs a pair of pixel operation using enable signal, contemporary logical and functional operation can be performed using same enable signal. The 3 bit down counter is used to decrement the address in the SIP (Stego image pixel) BRAM, which is initially high after getting output from logical block it can be decreased, if down counter reach 0^{th} address it enable the 10bit counter to increment the address of whole SIP BRAM as shown in Fig.1. This process is performed for the entire all pixels. The crucial factor is single clock cycle only required for fetching and embedding operation due to this clock cycle get reduced processor speed get increased. Note,

- CIP BRAM: Cover Image Pixel Block RAM
- SIP BRAM: Secret Image Pixel Block RAM
- STIP BRAM: Stego Image Pixel Block RAM

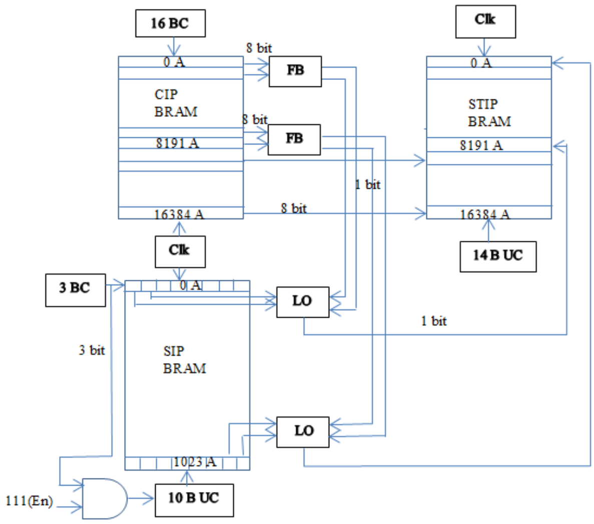


Fig.1. Architecture of Embedding phase

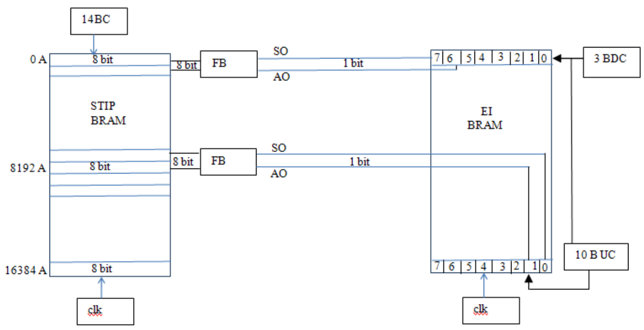


Fig.2. Architecture of Extracting phase

3.2.2 Architecture of Extracting Side:

In this extracting phase counter module is used to generate address to access the pixel value which is stored in STIP BRAM. Functional operation is performed; using enable signal input is access from the memory, contemporarily 3 bit down counter is in progress when down counter reach 0^{th} address it will enable the 10 bit counter to increase the address of whole memory, similarly operation is performed for entire all image pixels as shown in Fig.2.

3.2.3 Functional Block Diagram:

Functional block containing divider and adder circuit to perform equation.1function. This module is performing flooring using shifting operation followed by adder circuit is used, to perform addition. Input which is given to shifter is taken from 0^{th} address in CIP BRAM which means image 1^{st} pixel and the output of the sifter is added to 2^{nd} pixel of cover image which is taken from the 1^{st} address in CIP BRAM.

The output from shifter is denoted as shifted output and output from adder from is denoted as adder output. These two outputs is act as an input to logical block for further operation. Here not overlapping operation is performed i.e., 0^{th} bit of 1st two image pixel is used to perform functional operation for further operation it use 3^{rd} and 4^{th} pixel value 0^{th} bit is taken which is shown in Fig.3.

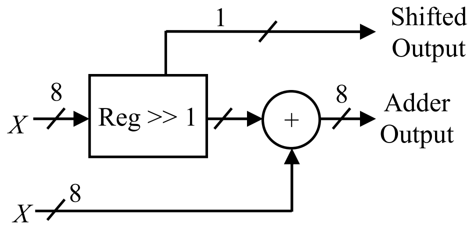


Fig.3. Functional Block diagram

3.2.4 Block Diagram of Logical Operation:

The shifted output and 7^{th} bit of secret image 1^{st} pixels are given as an input to xnor gate, then adder output and 6^{th} bit of secret image 1^{st} pixels are given as an input to the another xnor gate, output from both xnor gate is act as an select line to choose a mathematical operation which is shown Fig.4.

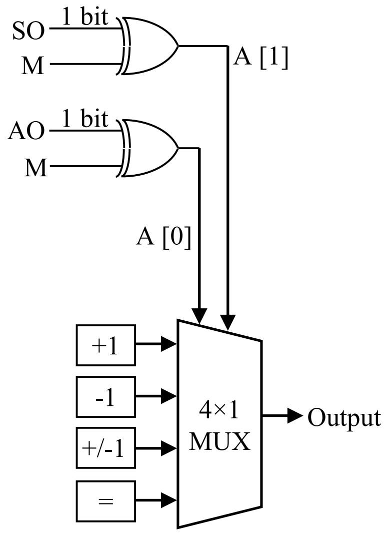


Fig.4. Block diagram of logical operation

3.3 PIPELINING

To improve the processor performance pipelining technique is used. Basic concept of pipelining is fetching, decoding and execution, two clock cycles are required for fetching and decoding process. Pipelining concept is used to speed up the process but area get increased due to using more number of register. While increasing the stages of pipelining clock cycle get increased due to this frequency also get increased. This method propose a new architecture to perform embedding and fetching operation in single clock cycle for paired pixel for entire cover image.

4. EXPERIMENTAL RESULTS AND ANALYSIS

The Verilog code for the proposed model is written and the code is simulated using ModelSim Altera 6.4a starter Edition and Xilinx Design suite 14.6. This method has been conducted on several grayscale images. The performance measures of the method have been evaluated and compared on the basis of two parameter with and without pipelining technique and the parameters are: frequency and delay

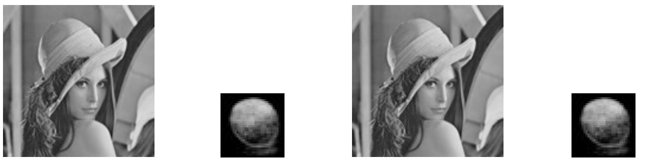
4.1 VERILOG RESULTS FOR VARIOUS IMAGES

Experiments are conducted with 5 different cover images such as Lena, Baboon, Barbara, Boat and Cameraman. Steganography LSBMR algorithm is implemented using Verilog HDL on each cover image, using 5 different secret images, such as, bone, skull, CT scan image, finger print and MRI scan images and thereby result 25 stego images are created, among them sample result, which are shown in Fig.5.

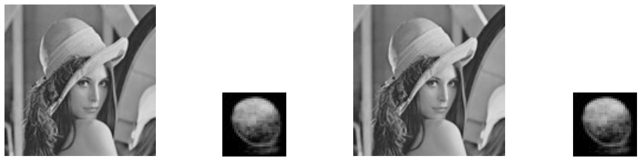
(a) Cover image Lena (b) Secret image Bone (c) Stego image Lena (d) Extracted Secret Image Bone

(a) Cover image Lena (b) Secret image skull (c) Stego image Lena (d) Extracted Secret Image Skull

(a) Cover image Lena (b) Secret image CT Scan (c) Stego image Lena (d) Extracted Secret Image CT Scan



(a) Cover image Lena (b) Secret image Fingerprint (c) Stego image Lena (d) Extracted Secret Image Fingerprint



(a) Cover image Lena (b) Secret image MRI Scan (c) Stego image Lena (d) Extracted Secret Image MRI Scan

Fig.5. Verilog results for various images

Also, Fig.6 shows the extracted secret images are of size 128×128 while the secret images are of size 36×36. PSNR value for each and individual stego and extracted secret images are evaluated using MATLAB and also delay and frequency was analysed for with and without pipelining mode in embedding and extraction side using Xilinx software.

4.2 SIMULATION RESULT OF EMBEDDING PHASE

The inputs are two images, one is cover and secret. Using enable signal pair of pixel operation is performed, if enable signal is high it perform shifting operation else it perform addition. Contemporarily logical operation is performing, output of functional block is considered as one of the input to functional block. Variable state [7:0] showing the output there is small variation in the decimal value while compared to doutc[7:0] and the remaining variables are which are assigned as internal variable for temporary storage.. Simulation result of Embedding is shown in Fig.6.

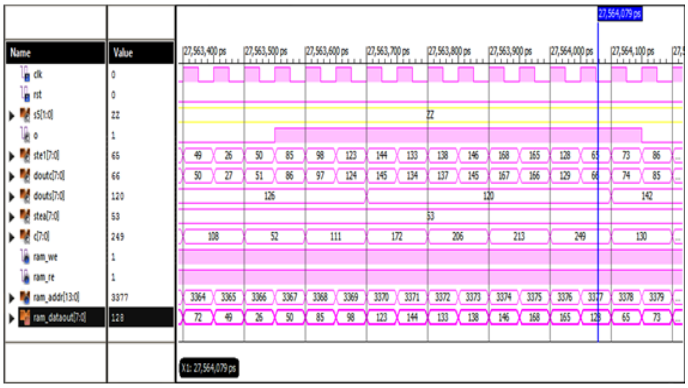


Fig.6. Simulation result for Embedding phase

4.3 SIMULATION RESULT OF EXTRACTING PHASE

The waveform for Embedding algorithm is presented in Fig.7. This module performs extraction operation in bit wise which is hide in cover medium LSB (0^{th} bit). 1^{st} 8 byte (1-8 image pixel value which is present in memory) in STIP contain secret image 1^{st} pixel value. To store that, temporary single bit memory is used its write depth is 8 initially it is in high then address is decrement to store output in a particular address. Variable out [7:0] represent the extracted output and the remaining variables are which are assigned as internal variable for temporary storage.

Fig.7. Simulation result for Extracting phase

Table.3. Performance measures of embedding and extracting using LSBMR algorithm

Cover image (.png) (128×128)	Secret image (.png) (36×36)	PSNR of Stego Image (128×128)	PSNR of extracted secret image (36×36)
Lena	Bone image	48.1311	48.1309
	Skull image	48.1311	48.1309
	CT scan image	48.1311	48.1309
	Finger print	48.1311	48.1309
	MRI scan image	48.1454	48.1444
Baboon	Bone image	48.1311	48.1309
	Skull image	48.1311	48.1309
	CT scan image	48.1350	48.1332
	Finger print	48.1311	48.1309
	MRI scan image	48.1595	48.1587
Barbara	Bone image	48.1311	48.1309
	Skull image	48.1311	48.1309
	CT scan image	48.1311	48.1309
	Finger print	48.1311	48.1309
	MRI scan image	48.1473	48.1465
Boat	Bone image	48.1311	48.1309
	Skull image	48.1311	48.1309
	CT scan image	48.1364	48.1356
	Finger print	48.1311	48.1309
	MRI scan image	48.1390	48.1374
Camera man	Bone image	48.4470	48.4469
	Skull image	48.1311	48.1309
	CT scan image	48.1311	48.1309
	Finger print	48.1355	48.1349
	MRI scan image	48.1478	48.1466

4.4 PERFORMANCE COMPARISION

The Table.4 and Table.5 respectively shows the performance comparison with and without pipelining in embedding and extraction side for frequency and delay. Using pipelining technique delay, which means embedding and extracting process get increased, in addition parameter viz. frequency get varied.

Table.4. Performance comparison in Embedding side

Cover image no. of Bytes	Secret image no. of Bytes	Method	Frequency	Delay
16384 (Bytes)	1024 (Bytes)	Pipelining	340.38MHz	3.407ns
		Without Pipelining	348.208MHz	3.445ns

Table.5. Performance comparison in Extracting side

Stego image no. of Bytes	Secret image no. of Bytes	Method	Frequency	Delay
16384 (Bytes)	1024 (Bytes)	Pipelining	271.30MHz	3.439ns
		Without Pipelining	279.438MHz	3.579ns

5. CONCLUSION

The project addresses the computation delay reduction and frequency for the LSBMR based image steganography. This is achieved by implementing the Embedding and Extraction schemes of LSBMR steganography algorithm in device such as FPGA. This paper developed new architectures; each module designed adopts high degrees of pipelining schemes. Embedding and extraction process have various modules and it was realized using Xilinx SPARTAN-6 device using Verilog HDL coding. Before hardware implementation process proposed architecture were analysed using model-sim to confirm the proposed concepts.

REFERENCES

[1] Michael Schaeferling et al., "ASTERICS An Open Toolbox for Sophisticated FPGA-Based Image Processing", *Proceedings of International Embedded World Conference*, pp. 1-8, 2015.

[2] M.S. Sutaone and M.V. Khandare, "Image based Steganography using LSB Insertion Technique", *Proceedings of IET International Conference on Wireless, Mobile and Multimedia Networks*, pp. 146-151, 2008.

[3] Weiqi Luo, Fangjun Huang and Jiwu Huang, "Edge Adaptive Image Steganography Based on LSB Matching Revisited", *IEEE Transactions on Information Forensics and Security*, Vol. 5, No. 2, pp. 201-124, 2010.

[4] J. Mielikainenm, "LSB Matching Revisited", *IEEE Signal Processing Letters*, Vol. 13, No. 5, pp. 285-287, 2006.

[5] Da-Chun Wu and Wen-Hsiang Tsai, "A Steganographic Method for Images by Pixel-Value Differencing", *Pattern Recognition Letters*, Vol. 24, No. 9-10, pp. 1613-1626, 2003.

[6] Cheng-Hsing Yang, Chi-Yao Weng, Shiuh-Jeng Wang and Hung Min Sun, "Adaptive Data Hiding in Edge Areas of Images with Spatial LSB Domain Systems", *IEEE Transactions on Information Forensics and Security*, Vol. 3, No. 3, pp. 488-497, 2008.

[7] Kh. Manglem Singh, L. Shyamsudar Singh, A. BubooSingh and Kh. Subhabati Devi, "Hiding Secret Message in Edges of the Image", *Proceedings of International Conference on Information and Communication Technology*, pp. 238-241, 2007.

[8] Kathryn Hempstalk, "Hiding behind Corners: using Edges in Images for Better Steganography", *Proceedings of Computing Women's Congress*, pp. 1-3, 2006.

[9] J. Mielikainen, "LSB Matching Revisited", *IEEE Signal Processing Letters*, Vol. 13, No. 5, pp. 285-287, 2006.

[10] X. Li, B. Yang, D. Cheng and T. Zeng, "A Generalization of LSB Matching", *IEEE Signal Processing Letters*, Vol. 16, No. 3, pp. 69-72, 2009.

[11] Z. Weiming, Z. Xinpeng and W. Shuozhong, "A Double Layered Data Embedding Scheme", *IEEE Signal Processing Letters*, Vol. 14, No. 11, pp. 848-851, 2007.

[12] Carlos Gonzalez et al., "Use of FPGA or GPU-based Architectures for Remotely Sensed Hyper Spectral Image Processing", *Integration the VLSI journal*, Vol. 46, No. 2, pp. 89-103, 2013.