International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 05 | May 2020 www.irjet.net p-ISSN: 2395-0072

FPGA Implementation of Image Steganography Algorithms using Generalized Exploiting Modification Direction and Pixel Segmentation Strategy with Indicator Bit

Vijaykumar.T.[1], Suraj.S.[2], Alan.V.[3], John.D.[4]

[1]*Assistant Professor, Dept. of E&CE, Dhanalakshmi College of Engineering, Chennai, Tamil Nadu, India.*
[2,3,4]*Student, Dept. of E&CE, Dhanalakshmi College of Engineering, Chennai, Tamil Nadu, India.*

Abstract - *The generalized exploiting modification direction (GEMD) steganography is an enhancement algorithm of the exploiting modification direction (EMD) algorithm to hide a high payload capacity with maintaining the quality of the stego-image. This paper proposed a developed EMD image steganography algorithm using improved pixel segmentation strategy with an indicator bit (PSS-IB) to overcome the drawbacks of the previous EMD algorithms. The proposed algorithm divides each block of pixels of the cover image into vector of coordinate area (VCA), an indicator bit (IB) and vector modification area (VMA). The VCA vector is kept unchanged, the IB is flipped if required and the VMA vector is modified. The VCA vector is segmented into "n" vectors to carry the secret data. The modified VMA vector is obtained by applying the algorithm on these vectors. Furthermore, the proposed algorithm is designed using MATLAB and FPGA implemented using Xilinx System Generator (XSG) on Spartan 3E Kit. The experimental results with comparison with the previous work, the proposed algorithm has high embedding rate, high embedding payload capacity, less computational complexity and keeping high stego-image quality.*

Key Words: GEMD, EMD, PVD, XSG, PSS-IB, VCA, VMA.

1. INTRODUCTION

Steganography is the science of embedding a secret text in a cover image without leaving a remarkable track on the cover image. The image steganography is a widely acceptable data hiding techniques [1], [2]. The steganography techniques are generally categorized into spatial domain techniques and transform domain techniques. The spatial domain techniques such as LSB based approach, EMD based approaches and Pixel Value Differentiating (PVD) based approaches. The transform domain techniques, such as DCT, DWT and IWT [3] – [5]. The EMD algorithm is a spatial domain technique. There are different EMD algorithms that used to obtain an acceptable stego-image quality as well as embed a large payload [6]. This paper proposed a developed GEMD image steganography algorithm that based on an improved Pixel Segmentation Strategy with Indicator Bit (PSS-IB). The proposed algorithm overcomes the drawbacks of the previous EMD algorithms [7]. The proposed algorithm has two procedures; an embedding procedure and an extracting procedure. During the embedding procedure, the proposed algorithm segments each block of pixels of the cover image into VCA, IB and VMA. The VCA vector of each pixel of the block is kept unchanged, the IB is flipped if required and the VMA vector is modified. The proposed algorithm assigns the overall VCA vector (the vector that contains the VCA of each pixel of the block) and generate "n" vectors to carry the secret text, where n = LVMA−1, and LVMA is the length of overall VMA vector [8] – [11]. During the extracting procedure, the embedded data is extracted from the stego-image pixels. The proposed algorithms are simulated using MATLAB and FPGA implemented using XSG on Spartan 3E Kit [12], [13]. The remaining of this paper is organized as follows; briefly review on different EMD steganography algorithms is presented in Section II. Section III describes the proposed developed GEMD image steganography algorithm based on improved pixel segmentation strategy with indicator bit. Section IV shows the FPGA implementation of the proposed algorithms, Simulation and experimental results are illustrated in Section V. Finally the conclusion remarks are demonstrated in Section.

2. REVIEW ON EMD STEGANOGRAPHY ALGORITHMS

There are several algorithms for image steganography based on the EMD algorithms. In 2006, Zhang et al. proposed a steganography algorithm to convert the secret data from binary into secret digits in (2n+1)- array notational system [6]. Each block of "n" pixels carried one of these secret digits. When each block contains two pixels, the algorithm modifies only one of the two pixels by subtracting 1, adding 1, or unchanging the pixel. In 2007, Lee et al. increases the embedding capacity to 1.5 times that of EMD algorithm by an improved EMD (IEMD) algorithm. This algorithm used only two pixels per block to embed the secret data but cannot used more than two pixels [7]. In 2008, Lee et al. also enhance the payload capacity of EMD by proposing a data hiding based on PSS [8]. However, the space of the VMA was determined to be insufficient and the variable codes needed to be exchanged before extra communication between receiver and transmitter. In PSS, each pixel is segmented into VCA and VMA. Each embedding block consists of two pixels. For the case of two pixels, 16 bits can be segmented into four pieces, which are denoted as VCA1, VMA1, VCA2 and VMA2 respectively. VCA1 and VCA2 correspond to VCA in the first

and the second pixels, VMA1 and VMA2 correspond to VMA in the first and the second pixels. The VCA vector is composed of VCA1 and VCA2. It has (8 – LVMA1) + (8 – LVMA2) bits where LVMA1 and LVMA2 are the bit length of VMA1 and VMA2 respectively. The VMA vector is composed of VMA1 and VMA2. The VCA is the base to generate the generating vectors, while the VMA is a guide to hide data [8]. For example, if the pair of pixels (p1, p2) = (200, 187)10, then (p1, p2) are transformed into their binary stream as (11001000, 10111011)2. If LVMA1 = 2 and LVMA2 = 1, then the VCA vector will be (1100101011101)2 and the VMA vector will be (001)2. If (LVMA1, LVMA2) increases, the mean square error (MSE) increases and the quality of stego image will decrease. In 2013, Kuo et al. proposed a GEMD algorithm. This algorithm embedded (n+1) bits of the secret data in each block of "n" pixels [9]. Kuo et al. also proposed a hybrid GEMD to increase the embedding rate and maintain the stegoimage quality in 2014 [10].

3. THE PROPOSED STEGANOGRAPHY ALGORITHM

This paper proposes a developed GEMD image steganography algorithm based on PSS-IB to improve the quality of stego-image. The proposed algorithm segments each block of pixels of the cover image into VCA, IB and VMA. The VCA vector is kept unchanged, the IB is flipped if required and the VMA vector is modified. The improved PSS is shown in Fig. 1.

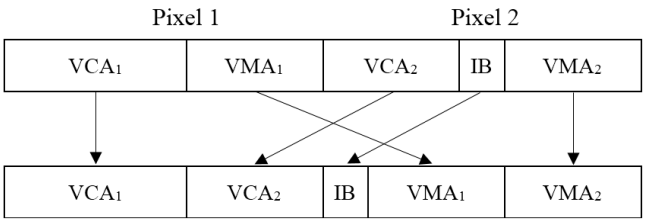


Fig .1

3.1 Embedding Algorithm

The secret data is embedded in the LVMA LSBs, where if there is a change in the LSBs, the distortion of cover image will be less significant rather than that of any change in the MSBs. Furthermore, the modification of the LSBs is not easy to be detected. The extraction function f(g1, g 2, ..., g n) is defined as [9]:

$$f(g_1, g_2, \ldots, g_n) = \left[\sum_{i=1}^{n} g_i . (2^i - 1)\right] \bmod (2^{n+1})$$

The steps of the proposed embedding algorithm are summarized as follow:

Inputs Cover image Ic and binary secret data streams

Outputs Stego image Is

1- Separate the color image into R, G and B channels.

2- For each channel, divide each pixel-pair (p1, p2) into VCA, IB and VMA and determine the value LVMA.

3- Calculate n = (LVMA – 1), to generate the vector of coordinates.

4- Assign the VCA vector and generate (g1, g2, … , gn).

5- Apply the embedding function t = f (g1, g2, ..., gn), calculated by Equation (1).

6- Calculate the difference value d = (s – t).

7- Transform d into binary and Fill it in VMA.

8- Reconstruct stego pixel-pair (p1, p2) by returning blocks back to original positions.

9- Flip the IB and check whether the stego pixel value (p2) is closer to the original value (p2) or not. If yes, the IB is flipped. For example, if given the pixel-pair (p1, p2) = (200, 187) = (11001000, 10111011)2. If LVMA1 = 2 and LVMA2 = 2, i.e. LVMA = 4, and secret data is s = (1101)2. Therefore, the VCA vector will be (11001010111)2, the IB will be (0)2 and the VMA vector will be (0011)2. Then, the stego pixel-pair (p1, p2) = (200, 186)10 as follows:

1- Compute n = LVMA – 1 = 3.

2- Generate 3 vectors, (g1, g2, g3) = (1100, 1010, 111)2 = (12, 10, 7)10.

3- Compute t = f(g1, g2, g3) = (1x12 + 3x10 + 7x7)mod (16) = (11)10.

4- Calculate the difference value d = (s – t) = 13 – 11 = (2)10.

5- Transform d into binary, d = (0010)2, then fill d in VMA = (0010)2.

6- The stego pixel pair will be (11001000, 10111010)2 = (200, 186)10.

7- Flip the IB, the stego pixel value (p2) = (10111110) = (190)10. Here, there is two values for p2.

8- Choose p2 = 186, because it is closer to 187 than 190. So (p1, p2) = (200, 186)10.

3.2 Extracting Algorithm

The secret data can be recovered from the stego-image by following the steps of the extracting algorithm. The

extracting steps divide each pixel-pair of the stego-image into VCA, IB and VMA vectors as in embedding steps.

$$f(g'_1, g'_2, \ldots, g'_n) = \left[\sum_{i=1}^{n} g'_i . (2^i - 1) \right] \bmod (2^{n+1})$$

The steps of the proposed extracting algorithm are summarized as follow:

Inputs Stego-image Is

Outputs Secret data stream S

1- Separate the stego-image into R, G and B channels.

2- For each channel, divide each pixel pair (p'

1, p'

2) into VCA, IB and VMA vectors.

3- Determine the value LVMA.

4- Calculate n = (LVMA – 1), to generate the vector of coordinates.

5- Assign the VCA vector, and generate (g1, g2, …, gn).

6- Compute (g1, g2, … , gn) from (g1, g2, … , gn) using VMA following the GEMD method.

7- Calculate the secret data s = f(g1, g2, … , gn), calculated by Equation (2) and convert it into binary.

8- Concatenate the secret data "s" to obtain the embedded secret data stream S.

For example, if given the stego pixel-pair (p'1, p'2) = (200, 186)10 = (11001000, 10111010)2, for a block, LVMA = 4.

Therefore, VCA = (11001010111)2, IB = (0)2 and VMA = (0010)2, n = LVMA – 1 = 3. Then, the secret data is

recovered, s = (1101)2 as follows:

1- Assign the VCA vector, generate 3 vectors (g1, g2, g3) = ((1100)2, (1010)2, (111)2) = (12, 10, 7)10.

2- Compute (g1, g2, g3) = (11, 11, 7)10 from (g1, g2, g3) = (12, 10, 7)10 when VMA = (0010)2 by following

the GEMD method as shown in Table (1).

3- Calculate the secret data s = f(g1, g2, g3) = f(11, 11, 7) = (13)10.

4- Convert decimal number (13)10 to binary (1101)2.

Table 1: The pixel modified quality depending on "d" [9].

When d < 8	Binary ($b_4b_3b_2b_1$)							
	0000	0001	0010	0011	0100	0101	0110	0111
g'_1	g_1	g_1+1	g_1-1	g_1	g_1	g_1+1	g_1-1	g_1
g'_2	g_2	g_2	g_2+1	g_2+1	g_2-1	g_2-1	g_2	g_2
g'_3	g_3	g_3	g_3	g_3	g_3+1	g_3+1	g_3+1	g_3+1
When d ≥ 8	Binary ($b_4b_3b_2b_1$)							
	1111	1110	1101	1100	1011	1010	1001	1000
	2's complement							
	0001	0010	0011	0100	0101	0110	0111	
g'_1	g_1-1	g_1+1	g_1	g_1	g_1-1	g_1+1	g_1	g_1+1
g'_2	g_2	g_2-1	g_2-1	g_2+1	g_2+1	g_2	g_2	g_2
g'_3	g_3	g_3	g_3	g_3-1	g_3-1	g_3-1	g_3-1	g_3+1

4. FPGA IMPLEMENTATION OF THE PROPOSED ALGORITHMS

The entire implementation of the proposed steganography algorithm based on PSS-IB and GEMD combining method using Simulink and Xilinx blocks is done through three phases. The image pre-processing and image post processing modules are designed using MATLAB software, while the proposed steganography algorithm is FPGA implemented using XSG blocksets [12], [13].

4.1 The image pre-processing phase

The image pre-processing is done using MATLAB that provides the inputs to the Xilinx blocks of the proposed

steganography algorithm as a specific vector array that is suitable for FPGA using system generator as follows:

• **Convert 2-D to 1-D**: The cover image is transformed into a single array of pixels. This array is segmented into

two vectors; vector with odd indices, the other vector with even indices.

• **Text**: The text is prepared to be read by the Xilinx blocks.

4.2 The proposed steganography algorithm

Four different cases according to the number of bits in the VCA and VMA vectors for each block of each channel of the color cover image are proposed. In the first case, (VCA1, VMA1) = (7, 1) and (VCA2, IB, VMA2) = (5, 1, 2), which means that, the VCA vector of each block is 12 bits, the IB is 1 bit and the VMA vector is 3 bits. In the second case, (VCA1, VMA1) = (6, 2) and (VCA2, IB, VMA2) –= (5, 1, 2), which means that, the VCA vector of the each block is 11 bits, the IB is 1 bit and the VMA vector is 4 bits. In the third case, (VCA1, VMA1) = (6, 2) and (VCA2, IB, VMA2) = (4, 1, 3), which means that, the VCA vector of each block is 10 bits, the IB is 1 bit and the VMA vector is 5 bits. In the fourth case, (VCA1, VMA1) = (5, 3) and (VCA2, IB, VMA2) = (4, 1, 3), which means that, the VCA vector of each block is 9 bits, the IB is 1 bit and the VMA vector is 6 bits. The model-based designs using Xilinx blocksets for the previously explained cases. Fig. 2 to Fig. 5 show the FPGA implementation for the first case of the proposed algorithm.

4.3 The image post-processing phase

The blocks of image post-processing are used to concatenate the two stego-signals and convert the resultant vector again into 2-D image as follows:

• **Convert 1-D to 2-D**: The two vectors (one for odd indices and the other for even indices) are concatenated into one vector and then, this 1-D vector is converted into 2-D image, where the stego-image is formed

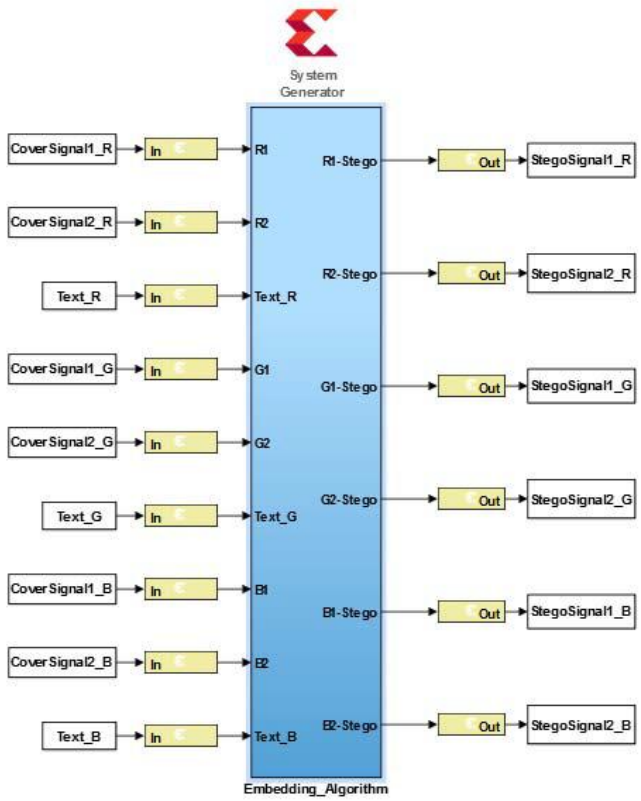


Fig. 2: The proposed XSG based GEMD steganography algorithm (First case).

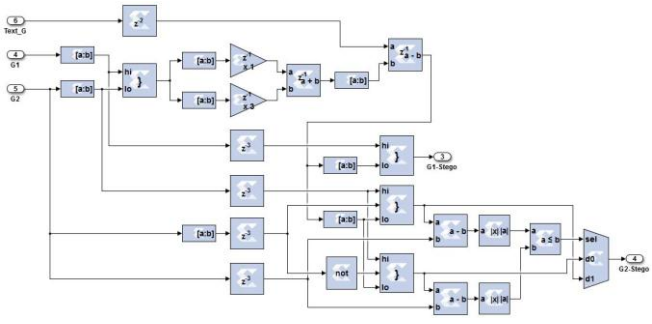


Fig. 3: The proposed XSG embedding algorithm (R channel).

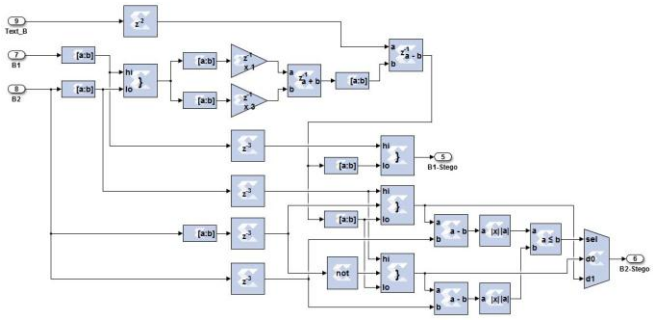


Fig. 4: The proposed XSG embedding algorithm (G channel).

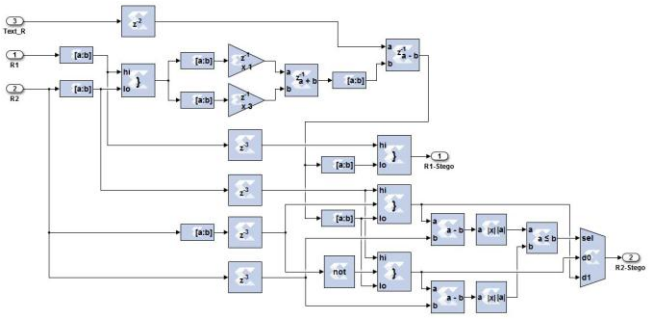


Fig. 5: The proposed XSG embedding algorithm (B channel).

5. EXPERIMENTAL RESULTS AND DISCUSSION

5.1 Database

The proposed steganography algorithm have been applied on various 512x512 BMP cover images. A comparative analysis of the proposed algorithms is investigated. The considered cover images (Airplane, Baboon,

Tiffany, Lena are shown in Fig. 6.

Fig. 6: The color cover images.

5.2 Performance Measurement

The proposed algorithms are performed using MATLAB software on the Intel Core i5 – 4210U CPU computer with 6 GB RAM. Furthermore, the proposed algorithms are designed and FPGA implemented using XSG on Spartan 3E Kit to evaluate the embedding payload capacity and the quality of stego-image. In the original EMD embedding

algorithm, when n = 2, the embedding rate is 3 x log2(5)/2 = 3.48 bpp and the payload capacity is 262,144 x 3 x log2(5)/2 = 912,261 bits. In the IEMD embedding algorithm, when n = 2, the embedding rate is 4.5 bpp and the payload capacity is 262,144 x 4.5 = 1,179,648 bits. In the PSS and EMD embedding algorithm, when a block of pixel-pair is used and LVMA = 3, the embedding rate is log2(2(2 LVMA-1 – 1) + 1)/2 = log2(7)/2 = 1.4 bpp and the payload capacity is 262,144 x 1.4 x 3 = 1,101,004 bits. In the GEMD embedding algorithm, when n = 2, the embedding rate is 3 x 3/2 = 4.5 bpp and the payload capacity is 262,144 x 4.5 = 1,179,648 bits. In the proposed embedding algorithm, when a block of pixel-pair is used and LVMA = 3, then the embedding rate is 0.5 x LVMA x 3 = 0.5 x 3 x 3 = 4.5 bpp and the payload capacity is 262,144 x 4.5 x 3 = 1,179,648 bits, but if LVMA = 6 in the proposed algorithm, then the embedding rate is 0.5 x LVMA x 3 = 0.5 x 3 x 3 = 9 bpp and the payload capacity is 262,144 x 3 x 3 = 2,359,296 bits. The performance evaluation parameters in all experiments are the PSNR, MSE and payload capacity. If the embedding rate is defined as R = S / (M x N), where S refers to the length of the secret data to be hidden and (M x N) is the total number of cover image pixels. The number of hidden secret bits in each pixel and the PSNR are used to compare the performance of the proposed algorithm and the previous EMD algorithms. If IC and IS are the cover image and stego-image respectively, then the MSE and PSNR are calculated using the following equations [14] – [16]:

$$MSE = \frac{1}{MxN}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}[I_S(i,j) - I_C(i,j)]^2$$

$$MSE_{RGB} = \left[\frac{MSE_R + MSE_G + MSE_B}{3}\right]$$

$$PSNR = 10\log_{10}\left(C_{max}^2 / MSE_{RGB}\right)$$

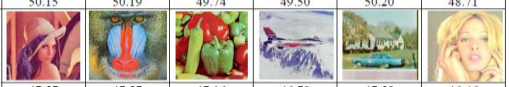
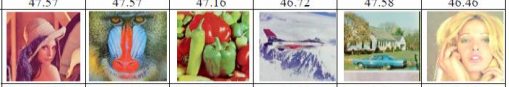
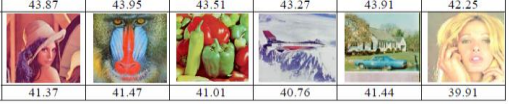
Where M and N are the dimensions of the cover images and Cmax is the maximum value in the original images. Table (2) summarizes a comparison of the embedding rate (bpp) and the PSNR between the previous EMD algorithms and the proposed algorithms, when each block contains a pair of pixels (n=2) in case of color images. As shown, when the embedding rate is 4.5, the PSNR value is 50.15 dB, when the embedding rate is 6, the PSNR value is 47.57 dB, when the embedding rate is 7.5, the PSNR value is 43.87 dB and when embedding rate is 9, the PSNR value is 41.37 dB. Table (3) summarizes a comparison in case of grayscale images, the quality of stego image is about 3.5 dB which is higher than that of the previous algorithms. The corresponding stego-images are shown in Fig. 7. In case of adding attacks to the stego-image, the embedded data cannot be efficiently extracted.

Table 2: Comparison of the embedding rate (bpp) and PSNR for the previous and proposed algorithms applied on color images **[8], [11]**.

Embedding Parameter	Embedding Algorithm							
	EMD	IEMD	EMD with PSS	GEMD	Proposed (Four cases) (L_{VMA1}, L_{VMA2})			
					(1, 2)	(2, 2)	(2, 3)	(3, 3)
Embedding Rate (bpp)	3.48	4.5	4.2	4.5	4.5	6	7.5	9
PSNR (dB)	52.13	50.72	51.0	50.74	50.15	47.57	43.87	41.37

Table (3): Comparison of PSNR for the previous and proposed algorithms applied on grayscale image at the same payload capacity **[8], [10], [17]**.

Embedding Method (L_{VMA1}, L_{VMA2})	Embedding Parameter	EMD-PSS	EMD-PSS-IB	GEMD-PSS	Proposed Algorithm
(1,2)	Payload capacity	393216			
	PSNR	46.36	48.08	46.38	49.83
(2,2)	Payload capacity	524288			
	PSNR	44.14	46.26	44.16	47.31
(2,3)	Payload capacity	655360			
	PSNR	46.38	48.05	40.03	43.51
(3,3)	Payload capacity	786432			
	PSNR	40.75	43.11	37.92	41.19

Embedding Method (L_{VMA1}, L_{VMA2})	Stego-images					
(1,2)						
	50.15	50.19	49.74	49.50	50.20	48.71
(2,2)						
	47.57	47.57	47.16	46.72	47.58	46.46
(2,3)						
	43.87	43.95	43.51	43.27	43.91	42.25
(3,3)						
	41.37	41.47	41.01	40.76	41.44	39.91

The synthesis results of the overall system are summarizing in Table (4). The processing time is 2.62 ms. The proposed steganography algorithm uses few resources of the FPGA; hence space is available for additional logic such as watermarking and image processing applications.

Table 4: The synthesis results of the proposed steganography algorithm.

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	96	9,312	1%	
Number of 4 input LUTs	148	9,312	1%	
Number of occupied Slices	108	4,656	2%	
Number of Slices containing only related logic	108	108	100%	
Number of Slices containing unrelated logic	0	108	0%	
Total Number of 4 input LUTs	149	9,312	1%	
Number used as logic	88			
Number used as a route-thru	1			
Number used as Shift registers	60			
Number of bonded IOBs	103	232	44%	
Number of BUFGMUXs	1	24	4%	
Average Fanout of Non-Clock Nets	1.71			

6. CONCLUSIONS

This paper proposed a developed GEMD image steganography algorithm based on PSS-IB to overcome the drawbacks of the previous EMD algorithms. The stego-image quality and the embedding payload capacity are two important parameters for steganography algorithms. The

proposed algorithms have an embedding rate 3 x 0.5 x LVMA (up to 9 bpp) which is greater than R = 3 x (n + 1)/n of the GEMD method. The proposed algorithms are simulated using MATLAB, designed and FPGA implemented using XSG on Spartan 3E Kit. The simulations and experimental results prove that the proposed algorithm has high embedding payload capacity (up to 2,359,296 bits) and keeping high stego-image quality (up to 50.15 dB).

7. REFERENCES

[1] I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, and T. Kalker, Digital Watermarking and Steganography, London: Elsevier Science & Technology, November 2007.

[2] Y.-H. Chen and H.-C. Huang, A Wavelet-based Image Watermarking Scheme for Stereoscopic Video Frames, 9th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IEEE, pp. 25 – 28, 2013.

[3] A. Cheddad, J. Condell, K. Curran and P. M. Kevitt, Digital Image Steganography: Survey and Analysis of Current Methods, Signal Processing, ELSEVIER, Vol. 90, pp. 727 – 752, 2010.

[4] K. Ntalianis and N. Tsapatsoulis, Remote Authentication via Biometrics: A Robust Video-Object Steganographic Mechanism Over Wireless Networks, IEEE Transactions on Emerging Topics in Computing, Vol. 4, Issue 1, pp. 156 – 174, March 2016.

[5] C. H. Yang, C. Y. Weng, S. J. Wang, and H. M. Sun, Adaptive Data Hiding in Edge Areas of Images with Spatial LSB Domain Systems, IEEE Transactions on Information Forensics and Security, Vol. 3, pp. 488 –497, 2008.

[6] X. Zhang and S. Wang, Efficient Steganographic Embedding by Exploiting Modification Direction, IEEE Communications Letters, Vol. 10, No. 113, pp. 781-783, 2006.

[7] C. F. Lee, Y.R. Wang, C.C. Chang, A Steganographic Method with High Embedding Capacity by Improving Exploiting Modification Direction, Proceedings of the third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP07), IEEE, pp. 497 – 500, 2007.

[8] C. F. Lee, C. C. Chang, and K. H. Wang, An Improvement of EMD Embedding Method for Large Payloads by Pixel Segmentation Strategy, Image and Vision Computing, ELSEVIER, Vol. 26, No. 12, pp. 1670 – 1676, 2008.