International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 04 | Apr-2018 www.irjet.net p-ISSN: 2395-0072

VLSI Based RDH Technique with Efficient Embedding Capacity

Sukanya P[1], Rajkumar S[2]

[1]*M.Tech.VLSI Design, Dept. of Electronics and Communication Engineering, NCERC, Pampady, Kerala, India*
[2]*Professor, Dept. of Electronics and Communication Engineering, NCERC, Pampady, Kerala, India*

Abstract - *Reversible data hiding (RDH) is one kind of information hiding technique with the characteristics such that not only the secret message needs to be precisely extracted, but also the cover image itself should be restored losslessly. In this project, complexity measurement is calculated to smoothen the image pixels and lossless compression technique uses run length coding which is simple to implement in FPGA and further to increase the embedding capacity the image is compressed before embedding and boundary extension is carried out. A VHDL code can be generated using algorithms which can be applied on an FPGA and can be operated with multimedia device. For secret data security level is enhanced by encryption. Embedding capacity and PSNR are improved thus, comparatively results in high performance.*

Key Words: *Reversible data hiding, Run length coding, FPGA*

1.INTRODUCTION

Reversible data hiding is the secret data has to be embedded into an original image and the receiver has to extract the secret data as well as the original image. Audio, video etc, embedding data in an image is a technological challenge. The size of the embedded data should not increase the size of original image. . In general, for a given embedding capacity (EC), one expects to minimize the embedding distortion measured by PSNR of the marked image with the original one. In a digital image, one can select some expandable difference values of pixels, and embed one bit in them. To extract the embedded data and restore the original values, the decoder needs to know which difference values have been selected for the DE [2]. DE technique to reversibly embed a payload into digital images. Both the payload capacity limit and the visual quality of embedded images of the DE method are among the best. According to the optimal value transfer matrix, the auxiliary information is generated and the estimation errors are modified. Also, the host image is divided into a number of subsets and the auxiliary information of a subset is always embedded into the estimation errors in the next subset [3]. Coltuc (2007) introduced a new approach to reversibly embed data based on simple transforms with low mathematical computation. Modification on prediction errors (MPE) only modifies less error values for embedding fewer data bits, thus high quality stego image is obtained [4]. Later by adaptive embedding and pixel selections helped in improving the embedding capacity and lesser distortion [5]. Reversible Data Hiding (RDH) with different PEE using histograms[1].In the C-PEE(Conventional Prediction Error Expansion) the prediction error e_i can be either shifted or expanded.−1 and 0 are the bits are used as expansion for embedding process. The original image x_i is modified to generate marked image. In A-PEE (Adaptive Prediction Error Expansion) is created for image quality. Here the complexity measurement is calculated for every x_i .The complexity measurement should satisfy the threshold value (T).The other values of the pixels are unmodified. Since threshold value should satisfy the complexity value it is that the embedding capacity will increase. In O-PEE (Optimized Prediction Error Expansion) only the bins more than 1 or smaller than −1 are only shifted and 0 is unmodified. Where, distortion is reduced from N −H to N −2H.For embedding capacity distortion, parameter a should be lesser than b. In AO-PEE (Adaptive and Optimal Prediction Error Expansion) adaptive and optimal PEE methods are combined together. By using these methods the distortion after embedding would be less. Both the methods combined it should satisfy the three parameters like a,b,T. These existing papers depend on arithmetic coding for lossless compression and thus it can be only implemented in matlab with lesser embedding capacity. In this paper, the proposed method consists of two divisions of an image namely shadow pixel and blank pixel. Embedding process is done first in shadow pixels and then in the blank pixels. This process is done through rhombus prediction. Lossless compression technique uses run length coding instead of arithmetic coding for simple and easy to implement in FPGA. In this technological world even if a secret data is embedded there is lot of cyber attacks, and to overcome this situation the secret data has to be encrypted to enhance the security level. Therefore, results in higher performance, security and improved embedding capacity.

2. RDH Embedding and Extraction

The histogram modification technique is about generating histogram and finding the peak point and the zero point and shifting histogram bins to embed bits. It uses a pair of peak and zero points to embed the secret messages. For each histogram sequence h_n, there are two bins a_n, b_n will be expanded and the other bins larger than b_n or smaller than a_n will be shifted. The original pixel x_i can be taken as its complexity measurement n_i . Thus the prediction error e_i can be modified to $\tilde{e_i}$ and can be embedded using,

$$\hat{e}_i = \begin{cases} e_i \ , & \text{if } a_n < e_i < b_n \\ e_i + \mathrm{m}, & \text{if } e_i = b_n \\ e_i - \mathrm{m}, & \text{if } e_i = a_n \end{cases}$$

Where also,

$$\hat{e}_i = \begin{cases} e_i + 1, \text{if } e_i > b_n \\ e_i - 1, \text{if } e_i < a_n \end{cases} \quad [1]$$

Here to avoid underflow and overflow 0 is changed to 1 and 255 is changed to 254. Run length coding is used for lossless compression. The complexity measurement n_i is computed for each consecutive pixels of the original pixel. After embedding the secret data into the shadow pixels the auxiliary information are given to the blank pixels which consists of the parameters a_n, b_n, compressed location map, and LSB bit values. The extraction and image restoration is opposite to the embedding process. By using,

$$e_i = \begin{cases} \hat{e}_i \ if \ a_n \le \hat{e}_i \le b_n \\ \hat{e}_i - 1 \text{ if } \hat{e}_i > b_n \\ \hat{e}_i + 1, \text{if } \hat{e}_i < a_n \end{cases} \quad [2]$$

1. For, a_n and n the embed one bit data would be $a_n - 1, n$.

2. For, b_n and n the embed one bit data would be $b_n + 1, n$.

3. When, e is lesser than an the shifting values would be $e - 1, n$.

4. When, e is greater than bn the shifting values can be $e + 1, n$.

5. When, $a_n < e < b_n$ are unmodified and remains the same.

Embedding procedure is that the image is partitioned into two layers. First is the shadow pixel and second is the blank pixel. The secret data has to be embedded first in shadow region one by one and after the other. Completion of shadow region tends to move to the next blank region. Where blank region has to be embedded with auxiliary information's to recover the cover image from the secret image with less distortion.

2.1 Rhombus prediction

To improve the accuracy of prediction, the rhombus prediction is proposed to generate the prediction errors from the original image. The four pixels {v1,v2,v3,v4}are used in rhombus prediction and the twelve pixels {v2,v3,v4,w1,...,w9} are used as a context x_i to compute the complexity measurement.

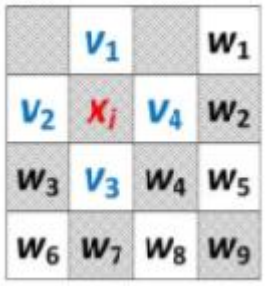

Fig -1: Rhombus prediction

For prediction in which the cover image is divided into two sets denoted as "shadow" and "blank". Embedding need to be processed to cover the whole image and the prediction of blank pixels is processed only after the embedding of shadow pixels is completed. And, in the decoding phase, first extract the embedded message and realize image recovery for blank pixels, and then extract the embedded message and realize image recovery for shadow pixels.

2.2 Lossless Compression Technique

The lossless compression technique uses Run length coding instead of arithmetic coding. Run length coding uses a sequence of 0s encoding with 1.Run length coding is a technique works by reducing the image size of a repeating string of characters. While, repeating strings are called "runs". Encoding can be processed with 2 bytes,

1. First byte represents the number of characters in the run and is called run count.

2. An encode run contain 128 or 256 characters.

3. Run count only access 127 or 255 characters.

4. Second byte is the value of the characters range (0-255 or 0-128).

The Run length can be separated into groups. The group number will decide the location of the bit received and the code word will be generated. To find the group number,

$$G_n = [log_2(l+3)] - 1 \quad [3]$$

If the group number is n both improved code and code sequence 2 will have n bits, and final code will have 2n bits.

2.3 Embedding capacity

To increase the embedding capacity the image should be compressed before embedding and uses run length coding. Also boundary extension is done. Thus the PSNR value also improves and results in efficient working of the principle. When PSNR (peak signal to noise ratio) is high it is obtained that this image has higher quality even after embedding and

finally for the security the secret data has to be authenticated using encryption key with an XOR function.

3. Experimental result

Table I: COMPARISON OF PSNR (IN DB) BETWEEN THE PROPOSED METHOD AND THE EXISTING METHOD

Image	PSNR (DB) [1]	PSNR(DB) [Proposed] [After embedding]	PSNR(DB) [After extraction]
Lena	70.7	80.2	78.2

From table-I the result shown is observed that the PSNR value of an existing Lena image has 70.7 db and the proposed method has a PSNR value of 80.2 db after embedding and a PSNR value of 78.2 db. This does not give much difference and from these PSNR values it is observed that the quality of the image is not degraded much. This calculation is undertaken with the help of MATLAB.

Table II: COMPARISON OF EMBEDDING CAPACITY BETWEEN THE PROPOSED METHOD AND THE EXISTING METHOD

Image	EC bits[1]	EC bits [Proposed]
Lena	43702	44069

From table –II the embedding capacity is calculated. The existing embedding capacity generates a value of 43702 bits while in the proposed method the embedding capacity has to be increased. To increase the embedding capacity boundary extension is done by adding 1 to row and column. since we only use 0-255 range of an image, instead of 0 it s used as 1 and 255 is changed to 254. If boundary extension is done, the pixels can be used from the beginning and embedding can be done from the first pixel of the image. This does not bring any underflow or overflow and results in 44069 bits.

4. Output waveforms

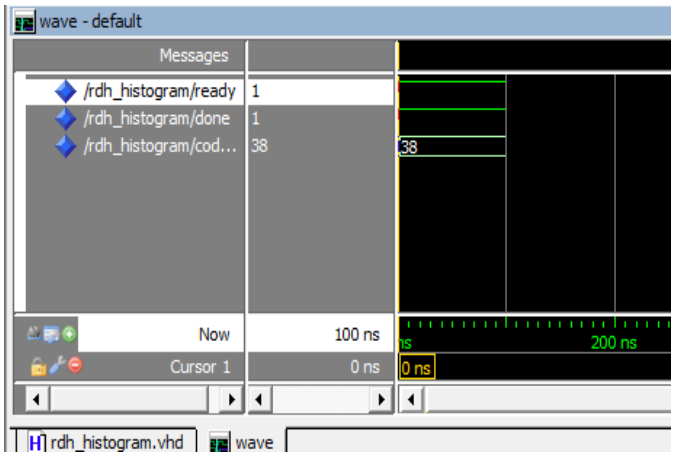

Fig-2: Waveform obtained from MODELSIM after embedding.

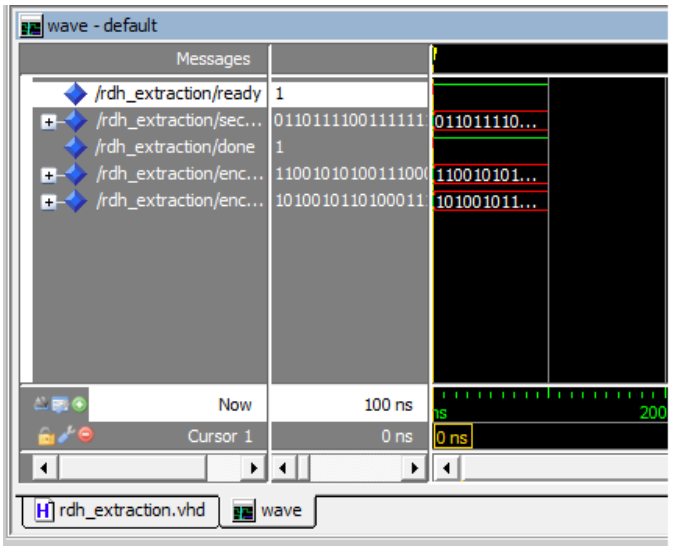

Fig-2: Waveform obtained after extraction.

From Fig-1 and fig-2 shows the output waveform generated in MODELSIM using VHDL coding. It obtains a run length of 38 and extraction consists of the encrypted message and the encrypted key which is XORed to get a secret message.

5. CONCLUSION

For a lossless compression run length coding is used and it results in high performance. For a Lena image the embedding capacity is 44069 bits and the existing method has an embedding capacity of 43702 bits. The proposed method has embedding capacity improved with 1000 bits comparing but, for other images like bird the embedding capacity is obtained less but the PSNR value is improved on both images. Results shows that the recovered image after extraction gives a PSNR value of 78.2 db and the PSNR value after embedding gives a value of 80.2 db . Thus from the values calculated it is clear that the image has not degraded and produces enough image quality. Further for improved embedding capacity for other images like scenario, thermal images also have to be improved with other lossless compression method. Thus this created VHDL code can be implemented in an FPGA kit for embedding and PSNR values can be calculated. This FPGA can be used with the help of a multimedia and then embedding process can be done with the FPGA language System C.

REFERENCES

[1] Xiaolong Li, Weiming Zhang, Xinlu Gui, and Bin Yang, “Efficient Reversible Data Hiding Based on Multiple Histograms Modification” IEEE Trans.Inf. Forensics Security . Vol.10.no.9.sept 2015.

[2] J.Tian, Reversible data embedding using a difference expansion,” IEEE Trans. Circuits Syst. Video Technol., vol. 13, no.8, pp. 890–896, Aug. 2003.

[3] X. Zhang, "Reversible data hiding with optimal value transfer," IEEE Trans. Multimedia,vol.15,no.2, pp. 316–325, Feb. 2013.

[4] W. Hong, T.-S. Chen, and C.-W. Shiu, "Reversible data hiding for high quality images using modification of prediction errors," J. Syst. Softw., vol. 82, no. 11, pp. 1833–1842, Nov. 2009.

[5] B. Ou, X. Li, Y. Zhao, R. Ni, and Y.-Q. Shi, Pairwise prediction- error expansion or efcient reversible data hiding, IEEE Trans. Image Process., vol. 22, no. 12, pp.50105021, Dec. 2013.

[6] J. Wang and J. Ni, A GA optimization approach to HS based multiple reversible data hiding, in Proc. IEEE WIFS, Nov. 2013, pp. 203208.

[7] X. Li, B. Yang, and T. Zeng, Efcient reversible watermarking based on adaptive prediction-error expansion and pixel selection, IEEE Trans. Image Process., vol. 20, no. 12, pp.35243533, Dec. 2011.

[8] S.Raveendra Reddy, and S.M. Sakthivel , "A FPGA implementation of data hiding using LSB matching method," IJRET,vol.04,issue.03,mar.2015.