A FPGA IMPLEMENTATION OF DATA HIDING USING LSB MATCHING METHOD

S. Raveendra Reddy[1], Sakthivel S M[2]

[1]*M. Tech (VLSI Design), Department of Sense, VIT University, Chennai, India*
[2]*Asst. Professor (Sr.G), Department of Sense, VIT University, Chennai, India*

Abstract

Data Hiding is one of the familiar methodologies used nowadays to authenticate and resolve the copyrights of the digital data. This paper discusses the verilog based FPGA implementation of data hiding in grayscale image using the concept of the LSB Matching technique. In this verilog based data hiding process the secret digital signature is hidden in the host image then analyzed with the PSNR value and Payload capacity.

Keywords: *Data hiding, Least-Significant-bit matching method, Reversible, Gray Scale Image, Real Time LSB Embedding, FPGA- Implementation etc..*

----------***----------

1. INTRODCUTION

The originality of the digital data is altered during the process of transmission and sharing over internet [1]. Hence there exists a problem in sending the data in a imperceptible manner [2]. Data hiding is one of the methods to hide the secret data in the host content and transmit in an effective manner [3]. The digital data can be an image, audio and video etc [4,5]. The data hiding process is said to be reversible if the original data can be reconstructed just by reversing the algorithm. Here the host is considered as a gray scale image and watermark as a sequence of binary data bits. Using LSB Matching methodology the secret data bits are hidden inside an image as secret signature.

All the data hiding algorithms till date are of software implementation only, a few are in hardware. If the data hiding operation is carried out by the Field Programmable Gate Array (FPGA) Chip, the process is quite fast and area, power and timing optimization is possible due to the presence of custom circuitry [6-8]. The FPGA based data hiding is a real time implementation whereas the software based approaches are doing the data hiding in offline mode of processing [6-8].

The proposed data hiding approach is the FPGA implementation of LSB Matching by J.Mielikainen[9] in real time using Verilog HDL language. Generally the data hiding process has to be efficient if it has a high payload capacity with good degree of imperceptibility. In this work the data hiding process is evaluated using the values of PSNR (i.e. to measure the Imperceptibility) and hiding capacity (i.e. no. of. bits hidden).

2. LSB MATCHING

This Section explains the process of data hiding in a grayscale image using the concept of LSB Matching technique. In the normal LSB embedding method simply we replace the LSB bits of the host content directly whereas the LSB Matching based data embedding involves the checking of the four conditions and finally the secret data signature is hidden in the host content.

In this LSB matching based data hiding process the two pixel pairs are selected for the secret data embedding and checked for the four conditions stated in the four different cases as below. The same process is repeated for the entire pairs of the pixels in the host image are processed. In case of the general LSB embedding technique the embedding is carried out by direct pixel replacement here it is processed by four checking process and then embedding. Hence this methodology will preserve the host image quality when compared to ordinary LSB embedding technique. The entire LSB matching based data hiding process is represented using a flow chart form in the Fig-1.

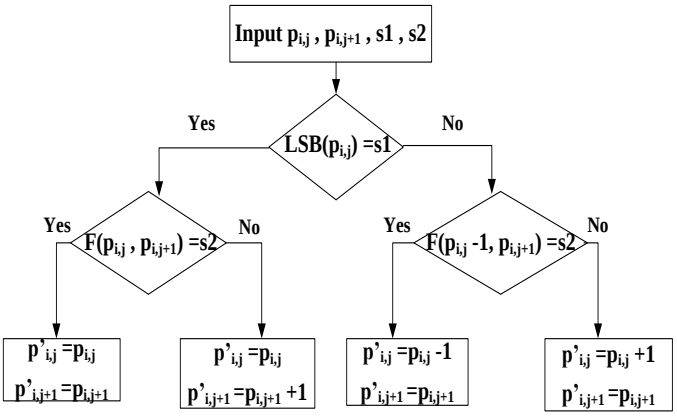

Fig.1. Strategy of Data Hiding

In the data hiding process two pixels$p_{(i,j)}$and$p_{(i,j+1)}$are selected first and the four case stated below are checked for the digital signature hiding. Here the secret data bits are considered as the$s_1$ands_2. Similarly$p'_{(i,j)}$and$p'_{(i,j+1)}$are the

two modified pixels after the data hiding process. The four cases for the embedding action is given below as follows.

Case-1: When $LSB(p_{(i,j)}) = s_1$ and $F(.) = s_2$, the pixel pair $p_{(i,j)}$ and $p_{(i,j+1)}$ does not need modification.

Case-2: When $LSB(p_{(i,j)}) = s_1$ and $F(.) \neq s_2$, the pixel $p_{(i,j)}$ does not change, and $p'_{(i,j+1)} = p_{(i,j+1)} + 1$.

Case-3: When $LSB(p_{(i,j)}) \neq s_1$ and $F(.) = s_2$, the pixel $p'_{(i,j)} = p_{(i,j)} - 1$, and $p_{(i,j+1)}$ does not change.

Case-4: When $LSB(p_{(i,j)}) \neq s_1$ and $F(.) \neq s_2$, the pixel $p'_{(i,j)} = p_{(i,j)} + 1$, and $p_{(i,j+1)}$ does not change.

For all the cases consecutive pixel are taken as $p_{(i,j)}$ and $p_{(i,j+1)}$ and the functional LSB bits of $p_{(i,j)}$ are calculated.

In the Case-1 the criteria is checked in such a way that if the LSB bit of $p_{(i,j)}$ is same as the secret bits s_1 and the calculated functional LSB bit F(.) bit using the equation -1 is same as the secret bit S_2 then the pixels $p_{(i,j)}$ and $p_{(i,j+1)}$ doesn't need any modification

$$F(p_{(i,j)}, p_{(i,j+1)}) = LSB\left(\left\lfloor \frac{p_{(i,j)}}{2} \right\rfloor + p_{(i,j+1)}\right) \quad (1)$$

The Case-2 analysis holds the situation if the LSB bit of $p_{(i,j)}$ is same as the secret bit S_1 and the F(.) is not equal to S_2 then pixel $p_{(i,j+1)}$ is replaced by $[p_{(i,j+1)} + 1]$ simply. Whereas the pixel $p_{(i,j)}$ is maintained as such.

The Case-3 is applicable to data hiding if the LSB of $p_{(i,j)}$ is not equal to S_1 and F(.) is equal to S_2 then the pixel $p_{(i,j)}$ undergoes a modification as $p_{(i,j)} = p_{(i,j)} - 1$ whereas the $p_{(i,j+1)}$ is not disturbed.

For the Case-4 analysis if the LSB of $p_{(i,j)}$ is not equal to S_1 and the F(.) is not equal to S_2 then the pixel $p_{(i,j)}$ is only replaced with $p_{(i,j)} = p_{(i,j)} + 1$.

The entire process of data hiding is explained here with an example sample pair of pixels. Consider the pixel pairs $p_{(0,0)} = 49, p_{(0,1)} = 44$ and confidential informations $= (10)_2$ as an example. When LSB (49) =1 and LSB (49) =1, then $p_{(0,0)} = 49$ it is substituted in Eq. (1) to give F(49,44)=LSB(68)=0 and F(49,44)=0. The case-1 condition are met, and this results in modified pixels as $p'_{(0,0)} = 49$ and $p'_{(0,1)} = 44$. In the same way the four cases are iteratively verified for data hiding row wise and column wise of the image until all the pixels pairs are processed. By this LSB matching concept with the secret bits and modification in the pairs of the pixel the data hiding is carried out effectively. The same can be used for the any type of grayscale images. As compared to ordinary LSB embedding the data hiding is happened based upon the matching process we can able to preserve the host image quality as such in this process.

Fig.2. Input image before Data Hiding

Fig.3. Output image after Data Hiding

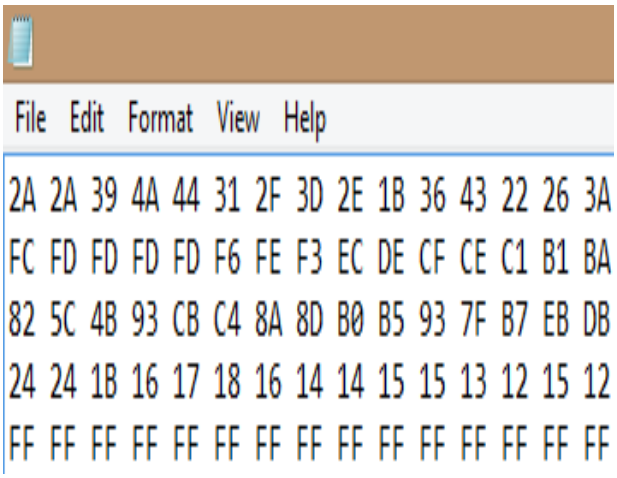

Fig.4. Image hexadecimal values

3. RESULTS AND DISCUSSIONS

This section gives the simulation results of the LSB matching process of data hiding using Matlab and the Xilinx for the FPGA implementation. During the Matlab simulation of data embedding we have taken an image size of 160 X 160 in Fig-2. The entire strategy is modeled in Matlab and the secret signature is embedded. Here the signature is considered as a random pattern. The input image before and after data hiding are shown in Fig- 2 & 3 respectively. From the Matlab simulation results it is found that the embedding capacity for this kind of LSB matching based embedding is of 12800bits with a high PSNR ratio of 52dB.

For the Verilog simulation of the LSB based data hiding the image pixel intensity values are converted into hexa decimal values as in Fig-4. Then the converted Hexa decimal values are stored in a RAM of size as the host image size. Then the data embedding strategy is modeled as a top level entity using Verilog HDL. Then the same entity is simulated using Xilinx ISE 14.3 with inputs of image hex values and secret signature from the RAMs. Here the secret signature is also stored in a RAM. The simulation results for the data hiding process are shown in the Fig-5. Finally the output images are also stored in the output Ram as hexa decimal values. For the reconstruction of the image the image hexa decimal values are wrote into the text file and watermarked image is converted using the Matlab. From the simulation of the verilog results, a comparison has been made with Matlab outputs and found that both are same in nature. (i.e. there is no deviation for Matlab and FPGA simulation results).

After verifying the simulated results the top level entity is synthesized using the Xilinx ISE 14.3 using the device family Artix 7 family with a speed grade of -3. In the Artix family the device XC7A100t is used with a package of CSG324. The device utilization summary for the data hiding strategy after the synthesis is shown in the Table-II. The table-I shows the embedding capacity and PSNR ratio value for the data embedding process. From the synthesis results it is found that there is no critical path in the design. The RTL technology level schematic is shown in the Fig-6.

The data hiding process is evaluated by the values of the embedding capacity and how effectively the secret signature is embedded. Here the above two metrics are evaluated by the values of embedding capacity and PSNR ratio respectively. The embedding capacity is indicated by bits per pixel (bpp) and the imperceptibly is given by the values of PSNR in dB.

Table –1 Embedding capacity and PSNR Value

Embedding Capacity (bpp)	PSNR (dB)
0.50 bpp	52 dB

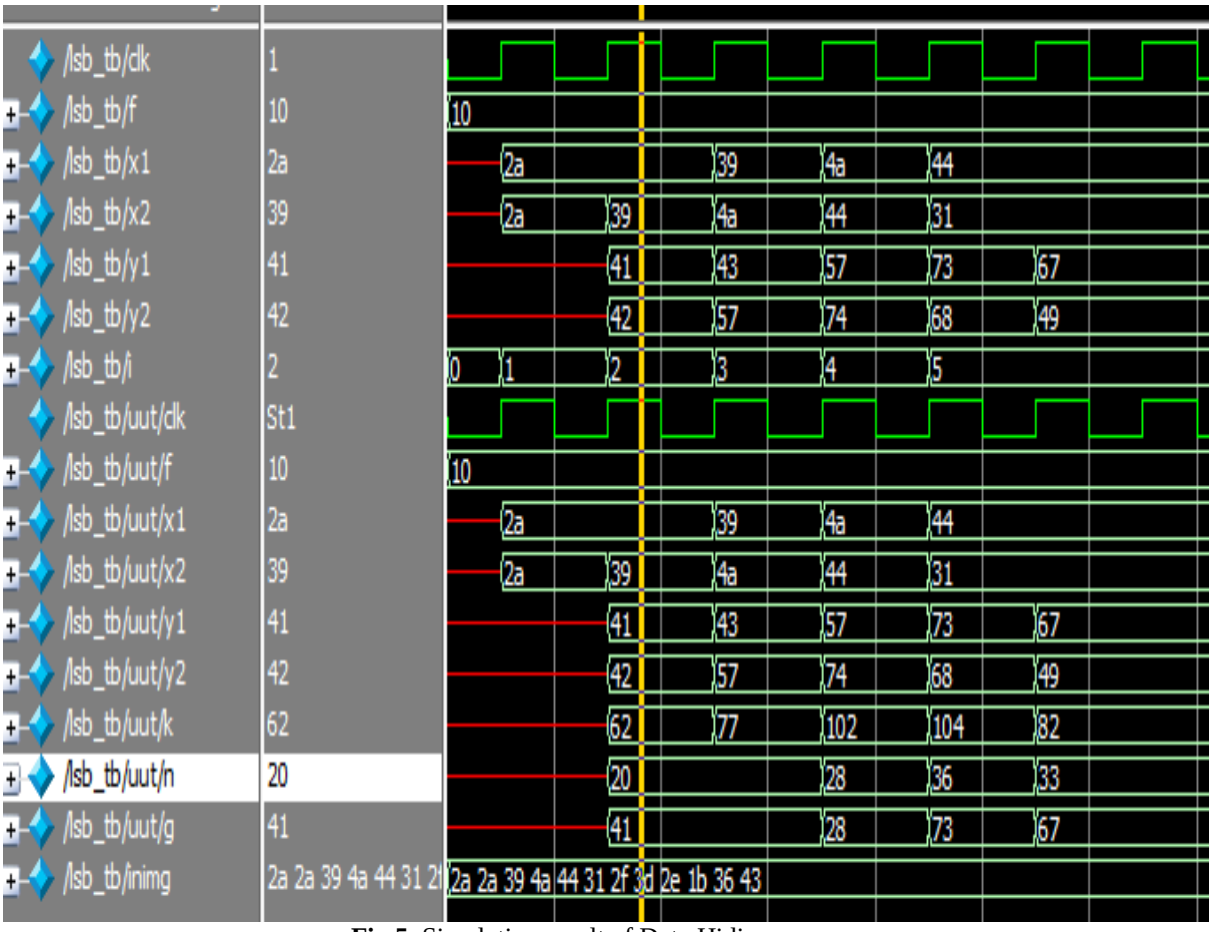

Fig.5. Simulation result of Data Hiding process

Table – 2 Device utilization Summary for Data Hiding

Parameters	LSB Method
Number of slice LUTs	23
Number of fully used LUT-FF pairs	0
Number of bonded IOBs	35
Number of BUFG / BUFGCTRLs	1

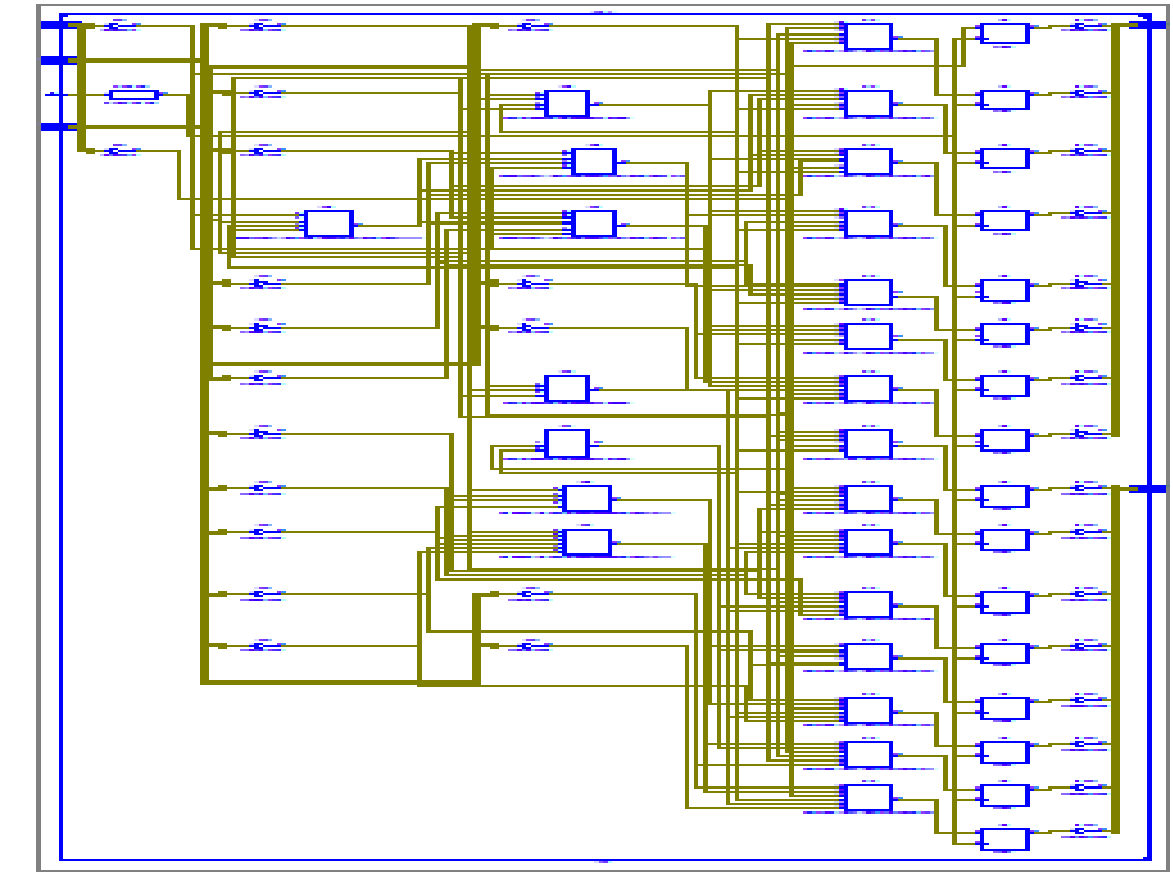

Fig.6. RTL Schematic of Data Hiding process

4. CONCLUSION

In this paper a FPGA implementation of watermarking in real time using LSB matching process is done.Our algorithm does embedding in the host image in an invisible manner. As it is a FPGA chip it can be incorporated a watermarking coprocessor inside a digital camera to carry out the watermarking, in the instant of capturing the picture itself.

REFERENCES

[1] Cox, I.J., Miller, M.L., Bloom, J.A., Fridrich, J., Kalker, J . : 'Digital watermarking and steganography' (Morgan Kaufmann Publishers,2008).

[2] C.T. Li, Digital fragile wateramrking scheme for authentication of JPEG images, in: Proceedings – Vision, Image and Signal processing, vol. 151(6), 2004, pp. 460-466.

[3] Neil F. Johnson, Sushil Jajodia, Exploring steganography: seeing the unseen, Computer Practices (1998) 26–34.

[4] R.J. Anderson, F.A.P. Petitcolas, On the limits of steganography, IEEE Journal on Selected Areas in Communications 16 (1998) 474–481.

[5] F.A.P. Petitcolas, R.J. Anderson, M.G. Kuhn, Information hiding—a survey, Proceedings of the IEEE, special issue on protection of multimedia content 87 (7) (1999) 1062–1078.

[6] Saraju P. Mohanty, N. Ranganathan and Ravi K. Namballa, "VLSI Implementation of Visible Watermarking for a Secure Digital Still Camera Design", IEEE Proceedings of the 17th International Conference on VLSI Design (VLSID'04)

[7] S.P. Mohanty, N. Ranganathan, K. Balakrishnan, "A dual voltage-frequency VLSI chip for image watermarking in DCT domain", IEEE Transactions on Circuits and Systems II (TCAS-II) 53 (5) (2006),pp. 394–398.

[8] P.Karthigaikumar,K.Baskaran, FPGA and ASIC implementation of robust invisible binary image watermarking algorithm using connectivity preserving criteria, Microelectronics Journal 42 (2011),pp.82-88.

[9] Mielikainen, LSB matching revisited, IEEE signal Processing Lett, 13(5)(2006) 285-287.
[10] D.C.Lou, C.L.Chou, H.Y.Wei,H.F.Hung, Active steganalysis for interpolation error based reversible data hiding, Pattern Recognit. Lett. 34(9) (2013) 1032-1036.
[11] X.Zhang and S.Wang, Steganography using multiplebase notational system and human vision sensitivity,IEEE Signal Process. Lett., vol.12, no. 1, pp. 67–70, Jan. 2005.
[12] A.M. Alattar, Reversible watermark using the difference expansion of a generalized integer transform, IEEE Trans. Image Process.13 (2004) 1147-1156.
[13] A. Ker, Improved detection of LSB steganography in Grayscale images, in*Proc.* Information Hiding Workshop, vol. 3200, Springer LNCS, 2004, pp. 97–115.
[14] T. Sharp," An implementation of key-based digital Signal steganography, in *Proc*. Information Hiding Workshop, vol. 2137, Springer LNCS, 2001, pp. 13–26.
[15] Gil-Je Lee, Eun-Jun Yoon, Kee-Young Yoo, A new lsb based digital water marking scheme with random mapping function, IEEE Computer Society (2008).

BIOGRAPHY

S. Raveendra Reddy pursuing M.TECH degree in VLSI Design from the Vellore Institute of Technology, Chennai, (2013-2015) and the B.TECH degree in Electronics and communication engineering from the R.V.R institute of Engg & technology, Hyderabad, in 2012.