

Wavelet Data Hiding using Achterbahn-128 on FPGA Technology

Mohamed I. Mahmoud, Moawad I. M. Dessouky, Salah Deyab, and Fatma H. Elfouly

Abstract— A data hiding technique is proposed for embedding a significant amount of data in digital Images while retaining high perceptual quality. The scheme employs digital communication techniques to achieve high robustness to standard image processing operations. Information is embedded in the wavelet domain by modifying selected wavelet coefficients of the host image. In the past work, the embedded signature data size or the original host image is fixed and predetermined for both transmitter and receiver. In this paper the data size is variable with predefined data limit and the embedded signature data is extracted without knowing the original host image. The signature data is encrypted using Achterbahn-128 stream cipher before embedding it in the wavelet coefficients. This paper intends to propose, the implementation of the data hiding system on FPGA technology. From the simulation results which is applied to the matlab program in order to see the output image (stego image), we can say that the human eyes cannot observe the difference between the original image and stego image.

Keyword: data hiding, wavelet transform, FPGA, Achterbahn-128

I. INTRODUCTION

In the present era of computers and fast communication, one needs to protect communicated information (message or plain text) from unauthorized user, while sending it through any electronic media. One such technique to protect the data is Steganography. Data hiding is also known as steganography (from the Greek words stegano for "covered" and graphos, "to write"). The Steganography consists of techniques to allow the communication between two persons. It hides not only the contents but also the existence of the communication in the eyes of any observer. These techniques use a second perceptible message, with meaning disjointed by the secret message. This second message works as a "Trojan horse" and is a container of the first one [1,2,3]. The new technologies and in special way, the information networks require more and more sophisticated strategies in order to prevent the message privacy. In this context, digital images and audio is excellent candidate to turn into containers of the messages, since the bits of a secret text message can be superimposed, as slight noise, to the bits employed for coding a digital image. Historically, the first instance of the use of Steganography is found when the Greeks received warning of Xerxes hostile intentions from a message underneath the wax of a writing tablet [4]. The Chinese practiced Steganography by embedding a code

Ideogram at a prearranged place in a dispatch while medieval Europe utilized grill systems in which a paper or wooden template would be placed over a seemingly innocuous text, highlighting an embedded secret message. The scientific study of Steganography can be traced to Simmons who in 1983 formulated it as the "Prisoners Problem". In this scenario, two prisoners (Alice and Bob) wish to devise an escape plan. However, all there communications pass through the warden (Willie) and if he detects any encrypted messages he will frustrate their plan by throwing them into solitary confinement. So they must find some way of hiding their cipher text in an innocent looking cover text. Also they must ensure their plan does not get destroyed through transmission channel.

In this chapter a data embedding technique is proposed for embedding a significant amount of data in digital images while retaining high perceptual quality. The scheme employs digital communication techniques to achieve high robustness to standard image processing operations. Information is embedded in the wavelet domain by modifying selected wavelet coefficients of the host image. The embedded data is encrypted by using Achterbahn-128 stream cipher.

II. MAIN SPECIFICATIONS OF ACHTERBAHN-128

The keystream generator of ACHTERBAHN-128 consists of thirteen binary primitive nonlinear feedback shift registers of

lengths between 21 and 33 and a Boolean combining function $F : F_2^{13} \rightarrow F_2$. The function F combines the output sequences of the thirteen feedback shift registers to produce the keystream $\zeta = (z_0, z_1, \dots)$. Throughout this proposal we shall use the capital letters $A_j, j = 0, 1, \dots, 12$, to designate the primitive FSR's and, in a slight abuse of notation, also to designate the feedback functions of the shift registers. The length of the shift register A_j is denoted by N_j . We have [5].

$$N_j = 21 + j \quad j = 0, 1, \dots, 12.$$

Let the initial state of the shift register A_j prior to encryption be given by the Row vector

$$r_0 = (r_0, r_1, \dots, r_{N_j-1}) \in F_2^{N_j}$$

The row vector r_0 is derived from the secret key K and the initial value IV using the key-loading algorithm to be described in Section 3.5. The key-loading algorithm ensures that r_0 will not be the zero vectors no matter which (K, IV) pair is used for initialization.

The output of the keystream generator at time t , denoted by $S(t)$, is the one of the Boolean combining function F with the inputs corresponding to the output sequences of the NLFSRs correctly shifted i.e. $S(t) = F(x_0(t), \dots, x_{12}(t))$. The Boolean combining function F is given by [5]:

$$\begin{aligned} F(x_0, x_1, \dots, x_{12}) = & x_0 + x_1 + x_2 + x_3 + x_4 + x_5 + x_7 + x_9 + x_{11} + x_{12} + x_0x_5 + x_2x_{10} + x_2x_{11} + \\ & x_4x_8 + x_4x_{12} + x_5x_6 + x_6x_8 + x_6x_{10} + x_6x_{11} + x_6x_{12} + x_7x_8 + x_7x_{12} + x_8x_9 + x_8x_{10} + x_9x_{10} + \\ & x_9x_{11} + x_9x_{12} + x_{10}x_{12} + x_0x_5x_8 + x_0x_5x_{10} + x_0x_5x_{11} + x_0x_5x_{12} + x_1x_2x_8 + x_1x_2x_{12} + x_1x_4x_{10} \\ & + x_1x_4x_{11} + x_1x_8x_9 + x_1x_9x_{10} + x_1x_9x_{11} + x_1x_9x_{12} + x_2x_3x_8 + x_2x_3x_{12} + x_2x_4x_8 + x_2x_4x_{10} + \\ & x_2x_4x_{11} + x_2x_4x_{12} + x_2x_7x_8 + x_2x_7x_{12} + x_2x_8x_{10} + x_2x_8x_{11} + x_2x_9x_{10} + x_2x_9x_{11} + x_2x_{10}x_{12} + \\ & x_2x_{11}x_{12} + x_3x_4x_8 + x_3x_4x_{12} + x_3x_8x_9 + x_3x_9x_{12} + x_4x_7x_8 + x_4x_7x_{12} + x_4x_8x_9 + x_4x_9x_{12} + \\ & x_5x_6x_8 + x_5x_6x_{10} + x_5x_6x_{11} + x_5x_6x_{12} + x_6x_8x_{10} + x_6x_8x_{11} + x_6x_{10}x_{12} + x_6x_{11}x_{12} + x_7x_8x_9 + \\ & x_7x_9x_{12} + x_8x_9x_{10} + x_8x_9x_{11} + x_9x_{10}x_{12} + x_9x_{11}x_{12} + x_0x_5x_8x_{10} + x_0x_5x_8x_{11} + x_0x_5x_{10}x_{12} + \\ & x_0x_5x_{11}x_{12} + x_1x_2x_3x_8 + x_1x_2x_3x_{12} + x_1x_2x_7x_8 + x_1x_2x_7x_{12} + x_1x_3x_5x_8 + x_1x_3x_5x_{12} + x_1x_3x_8x_9 \\ & + x_1x_3x_9x_{12} + x_1x_4x_8x_{10} + x_1x_4x_8x_{11} + x_1x_4x_{10}x_{12} + x_1x_4x_{11}x_{12} + x_1x_5x_7x_8 + x_1x_5x_7x_{12} + \\ & x_1x_7x_8x_9 + x_1x_7x_9x_{12} + x_1x_8x_9x_{10} + x_1x_8x_9x_{11} + x_1x_9x_{10}x_{12} + x_1x_9x_{11}x_{12} + x_2x_3x_4x_8 + x_2x_3x_4x_{12} \\ & x_2x_3x_5x_8 + x_2x_3x_5x_{12} + x_2x_4x_7x_8 + x_2x_4x_7x_{12} + x_2x_4x_8x_{10} + x_2x_4x_8x_{11} + x_2x_4x_{10}x_{12} + x_2x_4x_{11}x_{12} \\ & x_2x_5x_7x_8 + x_2x_5x_7x_{12} + x_2x_8x_9x_{10} + x_2x_8x_9x_{11} + x_2x_9x_{10}x_{12} + x_2x_9x_{11}x_{12} + x_3x_4x_8x_9 + x_3x_4x_9x_{12} \\ & x_4x_7x_8x_9 + x_4x_7x_9x_{12} + x_5x_6x_8x_{10} + x_5x_6x_8x_{11} + x_5x_6x_{10}x_{12} + x_5x_6x_{11}x_{12} \end{aligned}$$

The combining function F has the following properties:

- (i) F is balanced;
- (ii) F has algebraic degree 4;
- (iii) F is correlation immune of order 8;
- (iv) F has nonlinearity 3584;
- (v) F has algebraic immunity 4;
- (vi) Each variable of F appears in at least one monomial of degree 4 such that the Shift register lengths corresponding to the variables in that monomial are pairwise Relatively prime.
- (vii) F has an efficient hardware implementation [5].

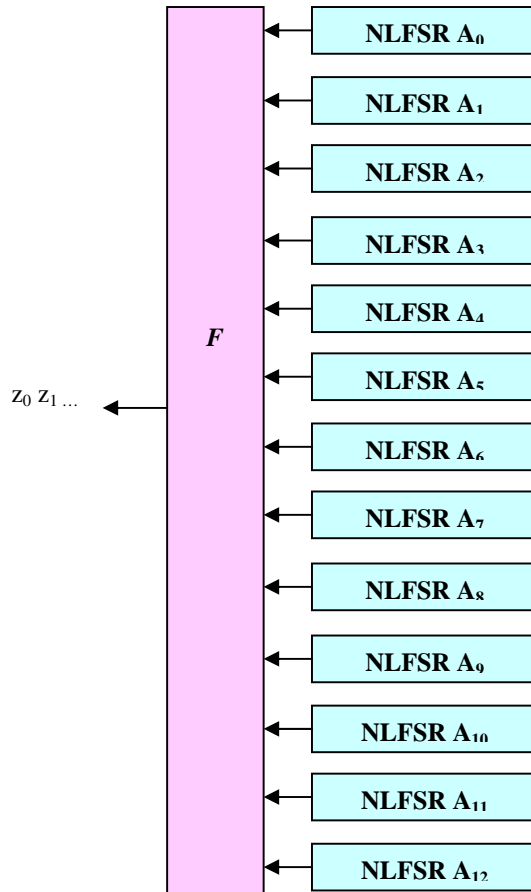


Fig. 6.1 The keystream generator of ACHTERBAHN-128

A. The key-loading algorithm

ACHTERBAHN-128 accommodates all key lengths between 40 and 128 and all IV -lengths between 0 and 128 that are multiples of eight. We shall use the letters k and l to denote the key and IV length, respectively. Assume that the secret key k is given as the bit string $K = u_0 u_1 \dots u_{k-1}$, and that the initial value (or initial vector) is given as $IV = v_0 v_1 \dots v_{l-1}$. The key and IV -loading algorithm is defined as follows [5]:

Step1. The memory cells $D_0, D_1, \dots, D_{N_j-1}$ of the shift register A_j are filled with the first N_j key bits $u_0 u_1 \dots u_{N_j-1}$. This is done for all thirteen shift registers in the keystream generator of ACHTERBAHN-128.

Step2. Into each shift register A_j the remaining $K - N_j$ key bits $u_{N_j} u_{N_j+1}, \dots, u_{k-1}$ are introduced, one after the other, according to Fig.6.2.

Step3. Into each shift register A_j all l initial value bits v_0, v_1, \dots, v_{l-1} are introduced in the same way as already described for the key bits in step2.

Step4. Each shift register A_j emits one bit. The thirteen shift register bits are then compressed by the Boolean combining function F into one output bit. This output bit is immediately fed back into each shift register as depicted in Fig.6.2. The same output bit is fed into all thirteen shift register. This operation is repeated 32 times.

Step5. The content of the memory cell D_0 in each shift register A_j is overwritten with a 1. This operation makes sure that none of the shift registers gets initialized with the all zero state.

Step6. Each shift A_j is clocked 64 times without emitting any output bit (warm-up).

The states of the shift register A_j at the end of step6 define the *initial state* of the keystream generator [5].

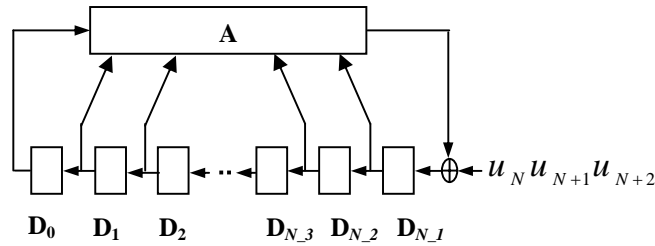


Fig.6.2: Bitwise introduction of key or IV -bits into a shift register

III. WAVELET TRANSFORM TECHNIQUE

Wavelet domain techniques are becoming very popular because of the developments in the wavelet stream in the recent years. A one dimensional discrete wavelet transform is a repeated filter bank algorithm [6]. The input is convolved with a high pass filter and a low pass filter. The low-pass filter branch generates the average DWT coefficients of the signal, while the high-pass branch generates the detail DWT coefficients. As the filter pair processes the signal, the output is decimated by a factor of two. Filtering the signal controls the resolution of the signal, while the subsampling process controls the scale. Scale and frequency are inversely proportional such that higher frequencies correspond to lower (i.e. finer) scales, while lower frequencies correspond to higher (i.e. coarser) scales. Because the filters separate the frequency bandwidth, the filter pairs produce different resolutions, or levels, of detail. Down sampling the filter output allows the output to be stored in the original signal space. The average coefficients are stored in the first half of the space, and the detail coefficients are stored in the latter half. The average coefficients are then processed again through the same set of filters producing a second set of average and detail coefficients. This DWT decomposition of the signal continues until the desired scale is achieved [6].

Two-dimensional signals, such as images, are transformed using the two-dimensional DWT. The two-dimensional DWT operates in a similar manner, with only slight variations from the one-dimensional transform. Given a two-dimensional array of samples, the rows of the array are processed first with only one level of decomposition. This essentially divides the array into two vertical halves, with the first half storing the average coefficients, while the second vertical half stores the detail coefficients. This process is repeated again with the columns, resulting in four subbands within the array defined by filter output. Fig.1 shows a one level decomposition using the two-dimensional DWT. these steps result in four classes of coefficients: the (HH) coefficients represent *diagonal* features of the image, whereas (HL and LH) reflect *vertical* and *horizontal* information. At the coarsest level, we also keep low pass coefficients (LL) [6].

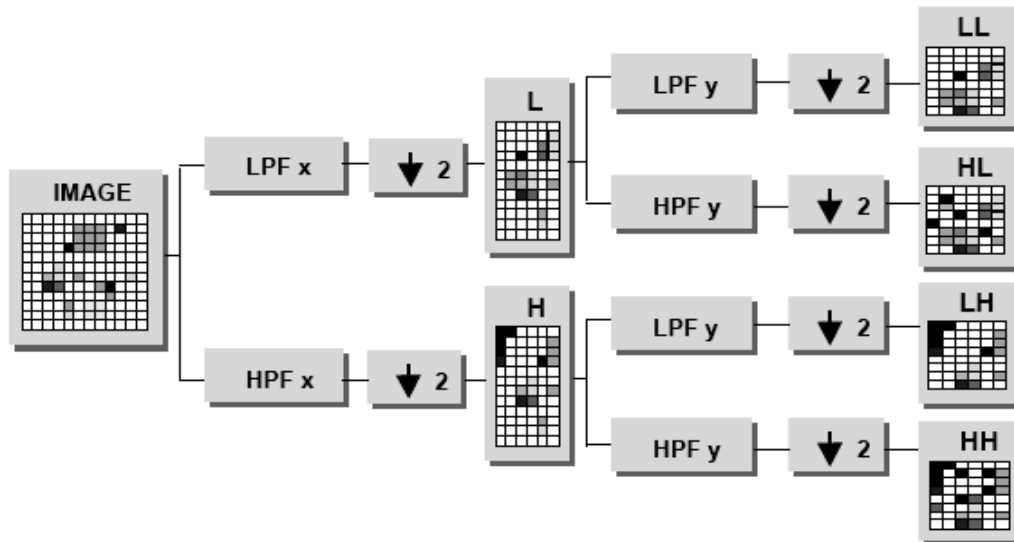


Fig. 1: One-level decomposition using the two-dimensional DWT, where LPF x Represents low-pass filtering of the image rows, HPF x represents high-pass filtering of Image rows, LPF y represents low-pass filtering of image columns, and HPF y represents high-pass filtering of image columns.

Since the discrete wavelet transform allows independent processing of the resulting components without significant perceptible interaction between them, hence it is expected to make the process of imperceptible embedding more effective [6].

This research applies wavelet transform by using Haar wavelet. In level one choose the three subbands to host the data, the three subbands which are (HH, LH, and HL), because the human eyes are not sensitive to the small changes in the edges and textures of an image but very sensitive to the small changes in the smooth parts of an image, the subband(LL)[6].

A. Algorithm for Embedding Data in Wavelet Transform

Step1: we need first to convert the secret message into a 1D bit stream. Of course the details of this step will depend on the particular message type. For example, in the case that the message is in text form, we can form the bit stream by simply converting the ASCII code of each character into an 8-bit binary representation, and then concatenating them as a sequence. The information bits are encrypted using ACHTERBAHN-128 stream cipher before embedding them in the elements of the host.

Step2: Before modification of coefficients, pseudorandom permutation of secret message is used for increasing security of embedded message. The idea behind the permutation is that the permutation generator uses the stego key and produces as output different sequences of the set $\{1, 2, 3, \dots, \text{length}(\text{message})\}$. Nobody can guess the generated random sequence without knowing the secret key. This ensures that only recipients who know the corresponding secret key will be able to extract the message from a stego-object.

Step3: Decompose image by using Haar wavelet transform.

Step4: The data sequence should be inserted into the least significant bit (LSB) of the wavelet subband coefficients starting from *HH* to *HL* according to the data length, since the receiver must know the data length in order to extract the data. So we insert not only the data sequence but also the data length sequence N .

Step5: after the embedding process ends the stego image is produced by applying the *Inverse of the Wavelet Transform (IWT)* on the modified coefficients.

B. Algorithm for Extracting Data in Wavelet Transform

Step1: Decompose image by using Haar wavelet transform.

Step2: The proposed scheme is blind since with the data length (N) only, the original cover-image is not needed to recover the embedded secret message from the received stego-image. So we need to extract the data length sequence N from the wavelet coefficients.

Step3: Extract the embedded data bits from the N LSB's of the wavelet coefficients.

Step4: The proposed scheme is considered secure. That is; without knowing the stego-key a passive warden can't extract the secret message. In addition, without knowing the decryption key we cannot retrieve the information bits.

IV. DATA HIDING ON FPGA

A wavelet based data hiding system will have a 2-D DWT core implemented on FPGA. Two-dimensional discrete wavelet transform is implemented as a cascade combination of two one-dimensional wavelet transforms (which described in chapter5), along with a set of memory buffers between the two stages. The memory buffers store intermediate results between the stages of the two-dimensional discrete wavelet transform. The image is input line by line to the serial processor. This computes the DWT along the rows, storing the coefficients in memory. Once all the lines have been input, the coefficients stored in memory are transposed to column major format and the DWT is computed along the columns. This direct method has the advantages of high efficiency and low complexity.

Recent advances in FPGA technology not only provide a significant increase in resources available for implementing logic, but also furnish a significant amount of flexible internal RAM. The internal RAMs provide the advantage of on-chip storage for storage of intermediate results and eliminate time-consuming external memory access operations. Taking advantage of this increased flexibility, the embodiments described herein are optimized for various wavelet filters and decomposition trees. In this way, the invention provides a designer with increased freedom and flexibility in choosing the order of the decomposition filters and the tree structure. Therefore the embodiments described herein can be implemented on FPGAs such as the cyclone FPGA from Altera [].

The implementation of the embedding and extraction modules is presented in the block diagram as shown in fig. 2.

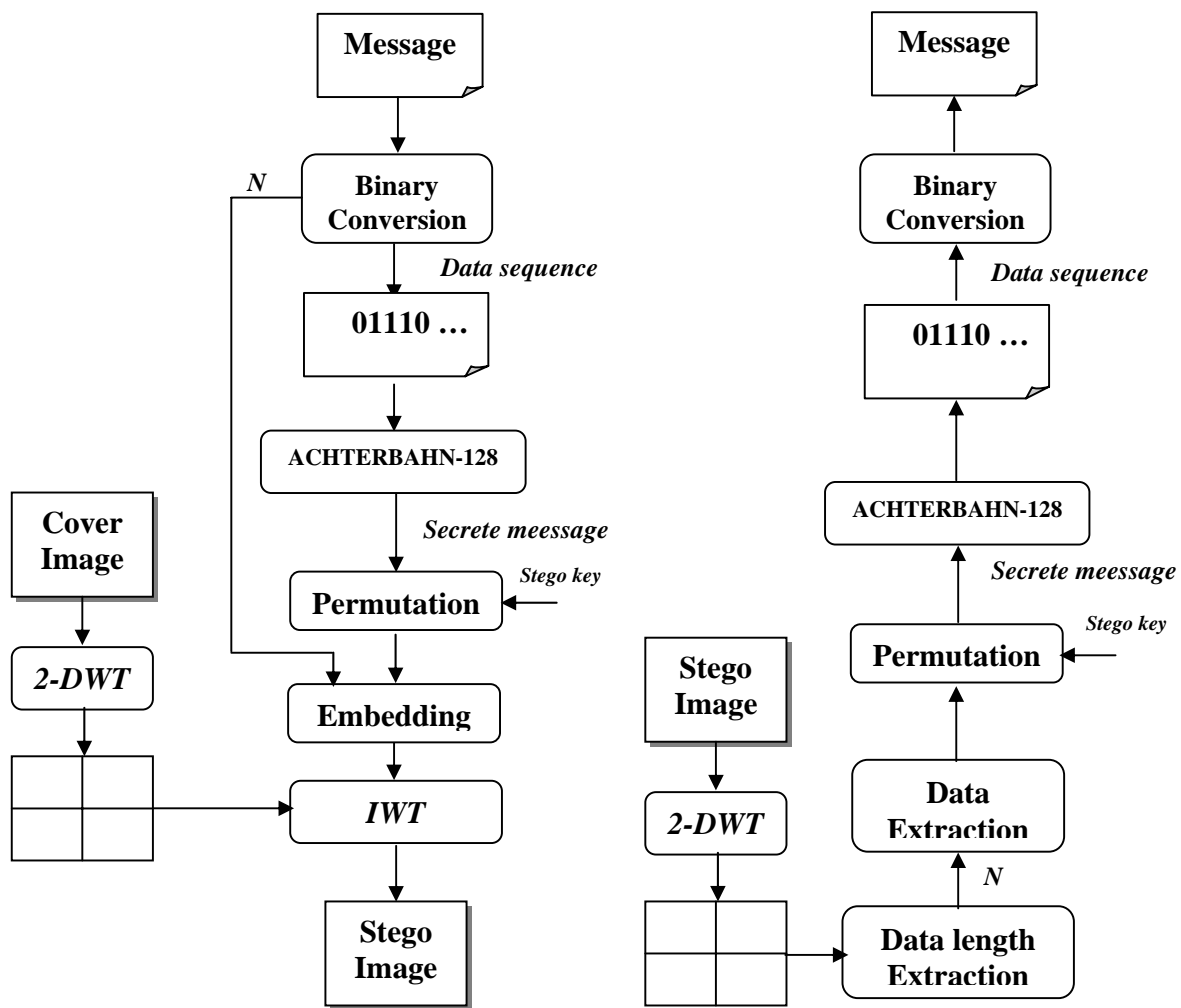


Fig. 2: Block diagram for hiding binary data in the wavelet coefficients of an image
 (a) The embedding module, (b) The extraction module.

V. SIMULATION OF DATA HIDING ON FPGA

To illustrate the functionality of the designed data hiding system through the simulation, the designed test bench has been run, the digital image and message applied as the inputs of the data hiding with clock period equal to 200 ns for image and 800 ns for message. The test bench results are shown in Fig 3. The data output results from the simulation which represent the stego image and the original image are applied to the matlab program in order to see them as in Fig. 4. From the two images we can say that the human eyes cannot observe the difference between them.



(a)



(b)

Fig.4. (a) original image (b) stego image

VI. SYNTHESIS RESULTS

In the case of satisfied simulation results, it is not a guarantee that the real FPGA will also function, the synthesis phase will started. A synthesis tool is used to include the propagation delay of the real scheme using the delay of each element that used in the design including the internal wiring connections. This time of propagation is calculated from any input to any output to detect the path that has the long propagation time or it is normally called the (critical path). The designer must takes into consideration these critical paths, which in some cases block the action of a desired function. Normally the design is an iterative process to compromise among many parameters or criteria. This means that; we may go through the design steps as many as we can to improve and optimize the result of our design. We have implemented the design using Altera FPGA device, EP1C12Q240C8. This device contains 12060 logic elements

VII. CONCLUSION

This paper proposed an efficient implementation of the data hiding system with a new algorithm for embedding and extracting data in wavelet transform. The suggested design is tested. The synthesis result of the suggested design is presented. Based on the obtained results, we can say that the human eyes cannot observe the difference between the stego image and original host image. Since the designed system hides not only the message contents but also the existence of the communication in the eyes of any observer. The implementation of the data hiding system on FPGA gives a fast and reliable realization.

REFERENCES

- [1]. Anil Kumar and Navin Rajpal, 2006. Application of T-Code, Turbo Codes and Pseudo-Random Sequence for Steganography. *Journal of Computer Science* 2 (2): 148-153.
- [2]. Bender, W., D. Gruhl, N. Morimoto and A. Lu, 1996. Techniques for data hiding. *IBM Systems J.*, 35: 313-336.
- [3]. Johnson, N.F. and S. Jajodia, 1998. Exploring steganography: Seeing the unseen. *IEEE Computer*, 31: 26-34.
- [4]. Anderson, R.J. and F.A.P. Petitcolas, 1998. On the limits of steganography. *IEEE J. Selected Areas in Communication*, 16: 4.
- [5]. Berndt Gammel, Rainer G'ottfert, Oliver Kniffner, 2006. ACHTERBAHN-128/80.
- [6]. Jonathan B. Ballagh, 2001. An FPGA-based Run-time Reconfigurable 2-D Discrete Wavelet Transform Core. Thesis submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering.