IJRRECS/October 2013/Volume-1/Issue-6/1593-1600 **ISSN 2321-5461**

INTERNATIONAL JOURNAL OF REVIEWS ON RECENT ELECTRONICS AND COMPUTER SCIENCE

FPGA IMPLEMENTATION OF LIFTING DWT BASED LSB STEGANOGRAPHY USING MICRO BLAZE PROCESSOR

B.Vidheya Raju[1], B.Vasantha Lakshmi[2]

[1]M.Tech, Dept of ECE, Pragati Engineering College, Surampalem, A.P, India
Email: vidhey15@gmail.com
[2]Associate Professor, Dept of ECE, Pragati Engineering College, Surampalem, A.P, India
Email: vasanthalkshm5@gmail.com

ABSTRACT:

This paper presents Associate in Nursing data concealment technique that utilizes lifting schemes to effectively hide data in pictures. A booming data concealment ought to end in the extraction of the hidden information from the image with high degree of information integrity. Current trends favor victimization digital image files because the cowl file to cover another digital file that contains the key message or data. the smallest amount vital Bit (LSB) embedding technique suggests that information are often hidden within the least vital bits of {the cowl|theduvet|the quilt} image and also the human eye would be unable to note the hidden image within the cover file. This paper explains the LSB Embedding technique and Presents analysis for LSB Steganography by victimization small blaze Processor presents in a very FPGA victimization System C cryptography.

Keywords: Steganography, LSB, Micro Blaze, FPGA.

1. INTRODUCTION:

Cryptography was created as a method for securing the secrecy of communication and lots of totally different ways are developed to inscribe and rewrite information so as to stay the message secret. Unfortunately it's typically not enough to stay the contents of a message secret, it's going to even be necessary to stay the existence of the message secret. The technique wont to implement this, is named steganography. Steganography is that the art of invisible communication by concealing info
inside different info. The term steganography springs from the Greek and virtually suggests that "covered writing" [1]. A steganography system consists of 3 elements: cover-object (which hides the key message), the key message and therefore the stego-object (which

is that the cowl object with message embedded within it.) Given the proliferation of digital images on the net, and therefore the giant redundant bits gift within the digital illustration of a picture, pictures square measure the foremost widespread cowl objects for steganography [2].

A digital image is represented employing a 2-D matrix of the color intestines at every grid purpose (i.e. pixel). Typically, grey pictures use eight bits, whereas colored utilizes twenty four bits to explain the color model, like RGB model. The steganography system that uses a picture because the cowl object is remarked as a picture steganography system [2].

The shift from cryptography to steganography is because of that concealing the image existence as stego-images change to implant the key message to hide pictures. Steganography conceptually implies that the message to be transmitted isn't visible to the informal eye. Steganography has been used for thousands of years to transmit knowledge while not being intercepted by unwanted viewers. It's Associate in nursing art of concealment info within another info. The most objective of Steganography is especially involved with the protection of contents of the hidden info. Pictures are ideal for info concealment [1, 2] as a result of the big quantity of redundant area is formed within the storing of pictures. Secret messages are transferred through unknown cowl carriers in such a way that the terribly existence of the embedded messages is undetectable. Carriers embrace images; audio, video, text or the other digitally diagrammatical code or transmission. The hidden message could also be plaintext, cipher text or something which will be diagrammatical as to a small degree stream.

III. HIDING METHODS IN IMAGE STEGANOGRAPHY

In Image Steganography, There are a variety of methods using which information can be hidden in images. LeastSignificant Bit Replacement Technique: In image steganography almost all data hiding techniques try to alter insignificant information in the cover image. Least significant bit (LSB) insertion is a common, simple approach to embedding information in a cover image. For instance, a simple scheme proposed, is to place the embedding data at the least significant bit (LSB) of each pixel in the cover image [7, 8, 9]. The altered image is called stego-image. Altering LSB doesn't change the quality of image to human perception but this scheme is sensitive a variety of image processing attacks like compression, cropping etc.We will be emphasizing more on this technique for the various image formats.

Moderate Significant Bit Replacement Technique:

The moderate significant bits of each pixel in the cover image can be used to embed the secret message. This method improves sensitivity to modification, but it degrades the quality of stego-image.

Experiments have shown that the length of hidden messages embedded in the least significant bits of signal samples can be estimated with relatively high precision.

IV. THE LSB TECHNIQUE

In the LSB technique of Message hiding, the least significant bit was replaced by the message bit of the secret message. In this paper we evaluated the technique using gray scale images of size 64*64 in which each pixel value was represented with 8 bit

representation.

For Example:

When the number 300, can be which binary representation is 100101100 embedded into the least significant bits of this part of the Cover image. If we overlay these 9 bits over the LSB of the 9 bytes above, we get the following (where bits in **bold** have been changed)

10010101 00001100 11001000
10010111 00001110 11001011
10011111 00010000 11001010

After embedding the message into the cover image, the stegoimage willbe obtained, then this stego image was transformed with the DWT transformation technique so that any hacker can't find where the message was embedded. At the receiver end the inverse DWT is applied, after LSB decryption the original image and message will be obtained.

Proposed Block Diagram:

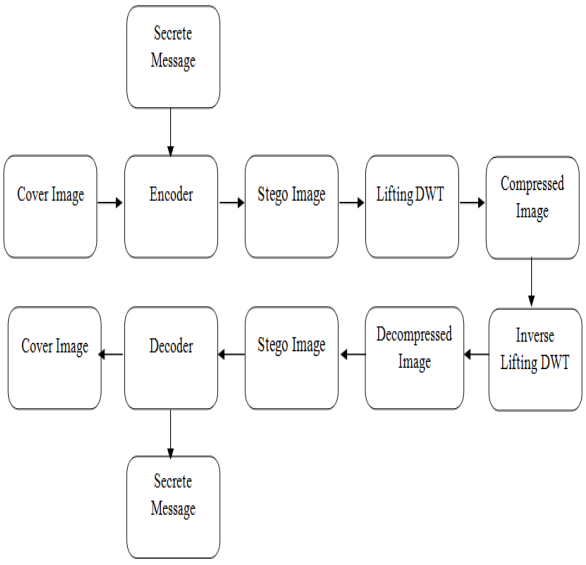


Fig1: Block Diagram of LSB Steganography

WAVELET TRANSFORM

Wavelets are mathematical functions defined over a finite interval and having an average value of zero that transform data into different frequency components, representing each component with a resolution matched to its scale.

The basic idea of the wavelet transform is to represent any arbitrary function as a superposition of a set of such wavelets or basis functions. These basis functions or baby wavelets are obtained from a single prototype wavelet called the mother wavelet, by dilations or contractions (scaling) and translations (shifts). They have advantages over traditional Fourier methods in analyzing physical situations where the signal contains discontinuities and sharp spikes. Many new wavelet applications such as image compression, turbulence, human vision, radar, and earthquake prediction are developed in recent years. In wavelet transform the basic functions are wavelets. Wavelets tend to be irregular and symmetric. All wavelet functions, $w(2kt - m)$, are derived from a single mother wavelet, $w(t)$..

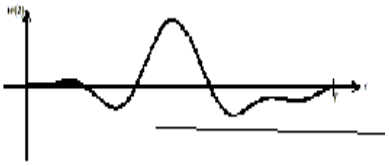
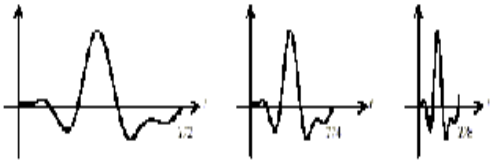


Fig2: Mother wavelet w(t)

Normally it starts at time $t = 0$ and ends at $t = T$. The shifted wavelet $w(t - m)$ starts at $t = m$ and ends at $t = m + T$. The scaled wavelets $w(2kt)$ start at $t = 0$ and end at $t = T/2k$. Their graphs are $w(t)$ compressed by the factor of $2k$ as shown in Fig. 3.3. For example, when $k = 1$, the wavelet is shown in Fig 3.3 (a). If $k = 2$

and 3, they are shown in (b) and (c), respectively.



(a)w(2t) (b)w(4t) (c)w(8t)

Fig 3: Scaled wavelets

The wavelets are called orthogonal when their inner products are zero. The smaller the scaling factor is, the wider the wavelet is. Wide wavelets are comparable to low-frequency sinusoids and narrow wavelets are comparable to high-frequency sinusoids.

2-D TRANSFORM HEIRARCHY

The 1-D wavelet transform can be extended to a two-dimensional (2-D) wavelet transform using separable wavelet filters. With separable filters the 2-D transform can be computed by applying a 1-D transform to all the rows of the input, and then repeating on all of the columns.

LL1	HL1
LH1	HH1

`

Fig4:Sub band Labeling Scheme for a one level, 2-D Wavelet Transform

DWT and Lifting Implementation

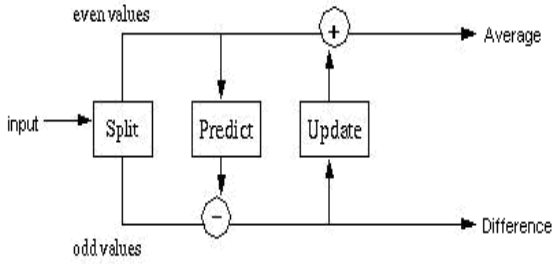


Fig 5: Lifting Scheme Process Flow

It is composed of three basic operation stages:

- **Splitting:** where the signal is split into even and odd
- **Predicting:** Even samples are multiplied by a predict factor
- **Updating :** The detailed coefficients computed by the predict step are multiplied by the update factors and then the results are added to the even samples to get the coarse coefficients

In this procedure Image pixels values are divided into even samples and odd samples then for getting high frequency value= odd-even samples then for low=even + high/2 then this procedure is repeated for each level.

EXPERIMENTAL SETUP

Xilinx Platform Studio

The Xilinx Platform Studio (XPS) is the development environment or GUI used for designing the hardware portion of your embedded processor system. B. Embedded Development Kit Xilinx Embedded Development Kit (EDK) is an integrated

software tool suite for developing embedded systems with Xilinx Micro Blaze and PowerPC CPUs. EDK includes a variety of tools and applications to assist the designer to develop an embedded system right from the hardware creation to final implementation of the system on an FPGA. System design consists of the creation of the hardware and software components of the embedded processor system and the creation of a verification component is optional. A typical embedded system design project involves: hardware platform creation, hardware platform verification (simulation), software platform creation, software application creation, and software verification. Base System Builder is the wizard that is used to automatically generate a hardware platform according to the user specifications that is defined by the MHS (Microprocessor Hardware Specification) file. The MHS file defines the system architecture, peripherals and embedded processors]. The Platform Generation tool creates the hardware platform using the MHS file as input. The software platform is defined by MSS (Microprocessor Software Specification) file which defines driver and library customization parameters for peripherals, processor customization parameters, standard 110 devices, interrupt handler routines, and other software related routines. The MSS file is an input to the Library Generator tool for customization of drivers, libraries and interrupts handlers.

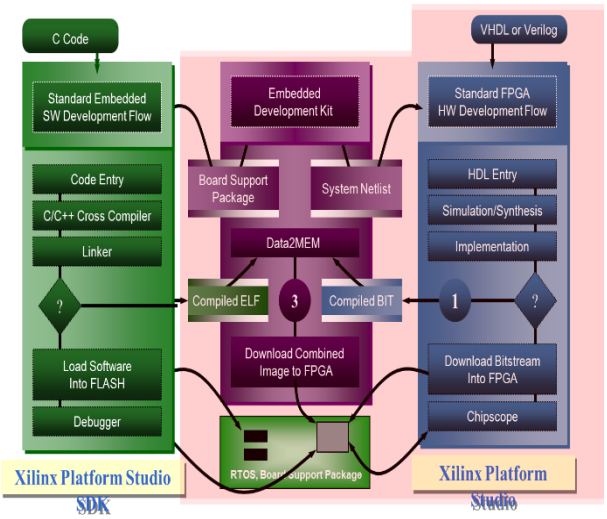


Fig 6: Embedded Development Kit Design Flow

The creation of the verification platform is optional and is based on the hardware platform. The MHS file is taken as an input by the Simgen tool to create simulation files for a specific simulator. Three types of simulation models can be generated by the Simgen tool: behavioral, structural and timing models. Some other useful tools available in EDK are Platform Studio which provides the GUI for creating the MHS and MSS files. Create / Import IP Wizard which allows the creation of the designer's own peripheral and import them into EDK projects. Platform Generator customizes and generates the processor system in the form of hardware net lists. Library Generator tool configures libraries, device drivers, file systems and interrupt handlers for embedded processor system. Bit streaminitializer tool initializes the instruction memory of processors on the FPGA shown in figure2. GNU Compiler tools are used for compiling and linking application executables for each processor in the system

[6]. There are two options available for debugging the application created using EDK namely: Xilinx Microprocessor Debug (XMD) for debugging the application software using a Microprocessor Debug Module (MDM) in the embedded processor system, and Software Debugger that invokes the software debugger corresponding to the compiler being used for the processor. C. Software Development Kit Xilinx Platform Studio Software Development Kit (SDK) is an integrated development environment, complimentary to XPS, that is used for C/C++ embedded software application creation and verification. SDK is built on the Eclipse opensource framework. Soft Development Kit (SDK) is a suite of tools that enables you to design a software application for selected Soft IP Cores in the Xilinx Embedded Development Kit (EDK).The software application can be written in a "C or C++" then the complete embedded processor system for user application will be completed, else debug & download the bit file into FPGA. Then FPGA behaves like processor implemented on it in a Xilinx Field Programmable Gate Array (FPGA) device.

The Algorithm is implemented in Micro blaze Processor and the results are furnished in the tabulation below

Results:

Fig7: Input Image reading through VB

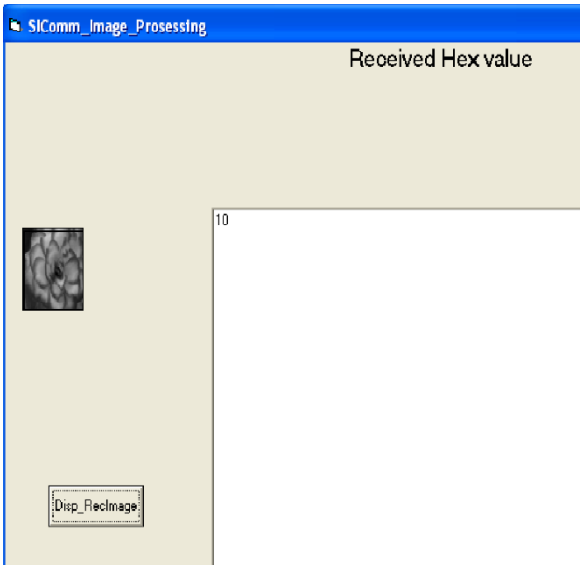


Fig 8: Stego Image

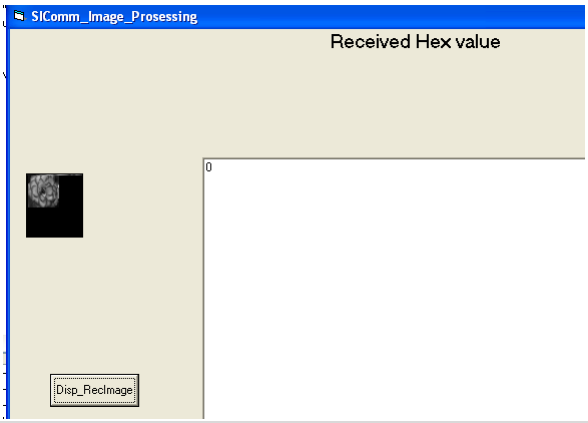


Fig 8: DWT Image

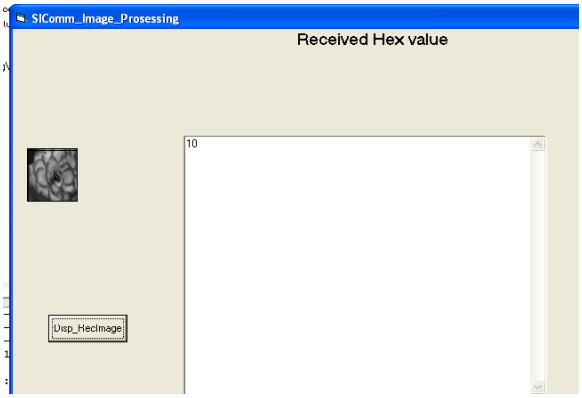


Fig 9: Original Image Reconstructed

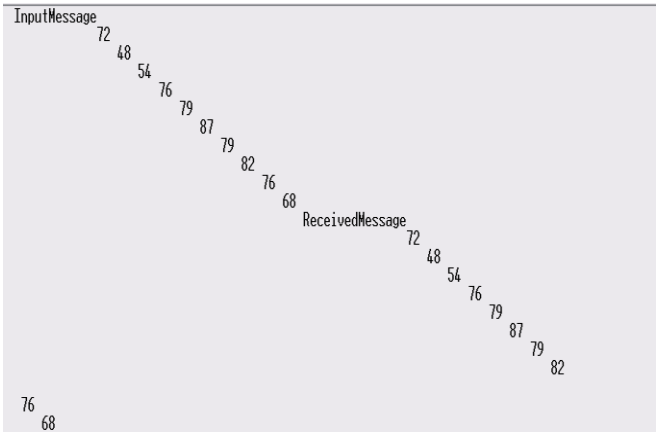


Fig 10: Message display on HyperTerminal

```
Device utilization summary:
---------------------------

Selected Device : 3s200tq144-4

 Number of Slices:                     1880  out of   1920    97%
 Number of Slice Flip Flops:           2118  out of   3840    55%
 Number of 4 input LUTs:               2971  out of   3840    77%
    Number used as logic:              2418
    Number used as Shift registers:     297
    Number used as RAMs:                256
 Number of IOs:                          62
 Number of bonded IOBs:                  62  out of     97    63%
    IOB Flip Flops:                      64
 Number of BRAMs:                         4  out of     12    33%
 Number of MULT18X18s:                    3  out of     12    25%
 Number of GCLKs:                         4  out of      8    50%
 Number of DCMs:                          1  out of      4    25%
```

Fig 11: Synthesis Report

```
Timing Summary:
---------------
Speed Grade: -4

   Minimum period: 15.304ns (Maximum Frequency: 65.342MHz)
   Minimum input arrival time before clock: 6.569ns
   Maximum output required time after clock: 16.181ns
   Maximum combinational path delay: No path found

Timing Detail:
--------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default period analysis for Clock 'sys_clk_pin'
  Clock period: 13.098ns (frequency: 76.348MHz)
  Total number of paths / destination ports: 111905 / 6937
-------------------------------------------------------------------------
```

Fig 12: Timing Analysis

Conclusion:

In this paper we have presented a new method of LSB Steganography using lifting DWT Process. This process was implemented by developing a soft-core processor in the FPGA i.e. Micro Blaze processor which implements the high speed execution to the Application.

Future work can be extended to RGB or color Image processing and can be extended to video processing level also.

REFERENCES:

[1] Pfitzmann Birgit. Information Hiding Terminology, First International Workshop, Cambridge, UK, Proceedings, Computer Science, 1174. pp. 347-350, May–June.

[2] Andreas Westfield and Andreas Pfitzmann, Attacks on Steganographic Systems, Third International Workshop, IH'99 Dresden, Germany Proceeding Computer Science October s, 1768. pp. 61- 76, 1999.

[3] Moerland, T., "Steganography and Steganalysis", *Leiden Institute of Advanced Computing Science*,Silman, J., "Steganography and Steganalysis: An Overview", *SANS Institute*, 2001 Jamil, T., "Steganography: The art of hiding information is plain sight", *IEEE Potentials*, 18:01, 1999

[4] Wei Zhang,ZheJiang,ZhiyuGao, and YanyanLiu,"An efficient VLSI architecture for Lifting based discrete wavelet transform,"IEEETrans.Circuits and systems,vol.59,NO.3,pp. 158-162,Mar.2012.

[5] G. Xing, J. Li, and Y. Q. Zhang, "Arbitrarily shaped videoobject coding by wavelet," IEEE Trans. Circuits Syst. Video Technol., vol. 11, no. 10,pp. 1135–1139, Oct. 2001.

[6] S. C. B. Lo, H. Li, and M. T. Freedman, "Optimization of wavelet decomposition for image compression and feature preservation," IEEE Trans.Med. Imag., vol. 22, no. 9, pp. 1141–1151, Sep. 2003.

[7] K. K. Parhi and T. Nishitani, "VLSI architecture for discrete wavelet transforms," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 1, no. 2, pp. 191–202, Jun. 1993.

[8] X. X. Qin and M. Wang, "A review on detection of LSB matching steganography," *Inf. Technol. J.*, vol. 9, pp. 1725–1738, 2010.

[9] A. D. Ker, "Locating steganographic payload via WS residuals," in*ACM Proc. 10th Multimed. Secur. Workshop*, 2008, pp. 27–31.

[10] A. D. Ker, "A general framework for the structural steganalysis of LSBreplacement," in *Proc. 7th Inf. Hiding Workshop, ser. Springer LNCS*,
2005, vol. 3727, pp. 296–311.

[11] J. Fridrich, M. Goljan, and R. Du, "Detecting LSB steganography in color and grayscale images," *IEEE Multimedia*, vol. 8, no. 4, pp. 22–28, 2001.

[12] S. Dumitrescu, X.Wu, and Z.Wang, "Detection of LSB steganography via sample pair analysis," *IEEE Trans. Signal Process.*, vol. 51, pp.1995–2007, Jun. 2003.

[13] A. Ker, "Steganalysis of LSB matching in grayscale images," *Signal Process. Lett.*, vol. 12, no. 6, pp. 441–444, Jun. 2005.

[14] A. D. Ker, "A fusion of maximum likelihood and structural steganalysis," in *Proc. 9th Inf. Hiding Workshop, ser. Springer LNCS*, 2007, vol. 4567, pp. 204–219.

[15] K. Lee, A. Westfeld, and S. Lee, "Generalised category attack—Improving histogram-based attack on JPEG LSB embedding," *Inf.Hiding'07*, pp. 378–391, 2007.

[16] J. Fridrich and M. Goljan, "On estimation of secret message length in LSB steganography in spatial domain," in *Secur.,Steganogr. WatermarkingofMultimed. Contents VI, ser. Proc. SPIE*, 2004, vol. 5306,pp.23–34.