Grenze International Journal of Engineering and Technology, January Issue

Steganography using Verilog by LSB

Chetan Umadi[1], Nagarathana R[2], R. Tushitaa[3], Riya Jain[4], Shreeja Ashok Kumar[5] and Shreya S[6]
[1-6]Dayananda Sagar College of Engineering/Department of Electronics and Telecommunication, Bangalore, India
Email: chetan-tce@dayanandasagar.edu, nagarathana-tce@dayanandasagar.edu, tushitaa03@gmail.com, riyajain201203@gmail.com, shreejaashokkumar31@gmail.com, shreya.shankara7@gmail.com

***Abstract*— This research paper explores using Least Significant Bit (LSB) steganography to embed text within digital images, implemented using Verilog in the Cadence environment. The method ensures secure, imperceptible data hiding by modifying the least significant bits of pixel values, preserving image quality with minimal visual distortion. The process involves data preparation (converting pixels and text to binary using MATLAB), encoding (embedding text in pixel bits sequentially), and decoding (extracting hidden data). Each stage is optimized for computational efficiency. The successful Verilog-based implementation highlights the feasibility of hardware-level data hiding, with future potential for integrating cryptographic techniques to enhance security.**

***Keyword*— Least Significant Bit (LSB) Steganography, Data Hiding, Image Steganography, Binary Conversion, Encoding and Decoding, Secure Data Transmission, Hardware Description Language (HDL).**

I. INTRODUCTION

In an era where data security and privacy are paramount, steganography has become a crucial technique for concealing information within carrier media such as images, audio, and video files. Unlike encryption, which makes data unreadable, steganography hides data within a carrier, making it undetectable under casual observation. Among various methods, Least Significant Bit (LSB) steganography stands out for its simplicity and efficiency. By modifying the least significant bits of image pixel values, it embeds secret data with minimal visual distortion, making it a preferred method for image-based data hiding due to low computational requirements. This research focuses on implementing LSB steganography using Verilog within the Cadence design environment, enabling precise bit-level manipulation. Verilog's integration with Cadence's robust simulation tools allows for an efficient, scalable hardware-based solution. The primary objective is to embed text data within digital images without visible changes while ensuring accurate and error-free extraction.

The project comprises several stages: data preparation, encoding, decoding, testing, and quality assessment. MATLAB converts text and image data into binary formats compatible with LSB encoding. In Verilog, pixel bits are sequentially modified to embed text data, and decoding reverses this process to extract the hidden message. Testing within Cadence ensures functionality across various image sizes and text lengths. Quality assessment involves evaluating PSNR and SSIM to confirm minimal impact on visual quality. Additionally, code optimization within Cadence reduces resource usage and processing time. Results show that Verilog-based LSB steganography achieves imperceptible data embedding and efficient resource utilization. Future enhancements could include integrating cryptographic techniques, opening new possibilities for secure hardware-based data transmission in applications requiring discreet communication and data privacy.

Grenze ID: 01.GIJET.11.1.562

II. Literature Review

This paper[1] review examines the evolution of image steganography techniques, focusing on methods designed to counter sophisticated detection systems. It covers LSB embedding, adaptive steganography, and CNN-based steganalysis resistance, assessing their effectiveness across various image formats and applications. The paper highlights AI-enabled algorithms, adaptive embedding strategies, and encryption integrations that enhance covert communication and security. It also explores preprocessing methods like noise reduction and discusses the trade-offs between efficiency, robustness, and security, providing insights into each technique's limitations and vulnerabilities. This paper[2] review focuses on LSB-based image steganography, systematically analyzing studies from 2016 to 2020. The paper evaluates a range of LSB modifications, detailing methods that enhance capacity, security, and invisibility. Techniques like one-bit and multi-bit LSB, combined with encryption algorithms (e.g., RSA, AES) and transformation tools (e.g., wavelet transforms), are highlighted for their role in securing embedded information. The paper also discusses recent hybrid approaches that combine LSB with other algorithms to improve resilience against detection and boost efficiency. This paper[3] provides an extensive review of contemporary methods for embedding and detecting hidden data within images. Covering a range of spatial and transform domain techniques, it evaluates the security and robustness of popular approaches like LSB embedding, adaptive techniques, and advanced AI-based detection. Key focus areas include balancing embedding capacity with security and resilience against attacks, alongside challenges in maintaining imperceptibility. The review also outlines future directions for enhancing steganographic defenses to keep pace with evolving detection technologies. This paper [4] offers a comprehensive overview of image-based data hiding methods, with a focus on the Least Significant Bit (LSB) technique. It examines LSB's strengths in terms of simplicity and minimal impact on image quality, along with its limitations, including vulnerability to detection and security risks in certain applications. The survey also compares LSB to alternative methods, assessing each on criteria such as security, embedding capacity, and visual quality, ultimately highlighting LSB's practicality in specific low-risk scenarios and applications. This paper[5] proposes a novel enhancement to the traditional Least Significant Bit (LSB) method by integrating a pseudo-random number generator (PRNG) to select pixels for data embedding. This randomization minimizes predictability, making the process more secure against detection. By carefully controlling the pixel selection, the method improves the imperceptibility of the hidden data, resulting in reduced visual distortion.

III. Methodology

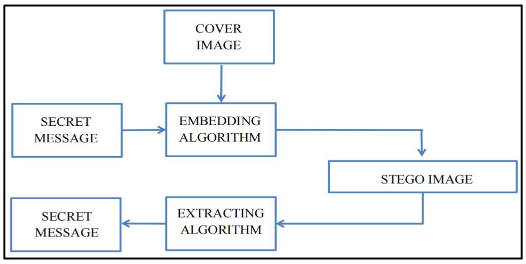

Fig 3.1: Block Diagram

This research paper utilizes the Least Significant Bit (LSB) steganography technique to hide text data within images, implemented through MATLAB for data preparation and Verilog in Cadence for encoding and decoding. The methodology involves multiple stages, including data preparation, encoding, decoding, testing, quality assessment, code optimization, and final documentation. Each stage is detailed below to provide a comprehensive view of the process.

A. Steganography Technique

LSB Steganography: The LSB technique is chosen for its simplicity and effectiveness in hiding information within images. This method alters the least significant bits of image pixel values to embed secret data, in this case, text, with minimal visual impact on the image. LSB is well-suited for digital images where small bit modifications are imperceptible to the human eye, making it an ideal choice for this project.

B. Data Preparation

- Binary Conversion in MATLAB: To embed text into an image, both the text and the image must first be converted into binary format.

- Image Conversion: Each pixel in the image is represented as an 8-bit binary value in MATLAB, facilitating bit-level manipulation required for LSB encoding.
- Text Conversion: Text data is converted into its ASCII binary representation, where each character is represented by a standard 8-bit ASCII code.

C. Encoding Process in Verilog

- Bit Masking and Embedding: Using Verilog, the LSB encoding process manipulates the binary image data by masking the least significant bit of each pixel.
- Sequential Embedding: Each bit of the text (ASCII binary) is embedded sequentially into the LSB of each pixel. This sequential embedding continues until the entire text message is embedded into the image.

D. Decoding Process

- Extraction of LSBs: The decoding process begins by extracting the LSBs from the modified image pixels. These bits collectively represent the binary form of the hidden text.
- Binary Reconstruction: The extracted binary data is then reconstructed into a binary stream.
- ASCII Conversion: The binary stream is converted back into readable text by mapping the binary data to ASCII characters. This allows for the hidden message to be retrieved in its original form.

E. Testing and Verification

- Module-Level Testing: Each component (e.g., encoding and decoding modules) is tested individually in Cadence to verify correct functionality.

F. Quality Assessment

- PSNR (Peak Signal-to-Noise Ratio):PSNR is calculated to measure the difference between the original image and the modified image with embedded data. Higher PSNR values indicate lower levels of distortion and higher quality.
- SSIM (Structural Similarity Index): SSIM is also measured to assess the perceptual quality of the modified image. SSIM scores close to 1.0 suggest minimal perceptual differences between the original and modified images.
- These metrics ensure that the embedded message does not significantly impact the visual quality of the image.

G. Code Optimization

- Efficiency Enhancements in Cadence: The encoding and decoding code is optimized to reduce the number of operations, making it computationally efficient.
- Memory Management: Efficient memory usage is prioritized, especially during the binary conversions and LSB modifications. This ensures minimal memory overhead, crucial for handling large images.

H. Final Documentation

- Performance Metrics: The final report includes detailed metrics such as encoding time, decoding time, and memory usage to assess the efficiency of the implementation.
- Quality Scores: PSNR and SSIM scores are documented to provide evidence of the technique's effectiveness in maintaining image quality.

IV. RESULTS

A. Outputs from Matlab

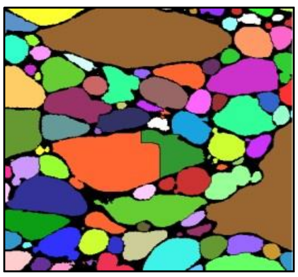

Fig 4.1: Input image

The dimensions of the image (width x height) are stored to determine the number of pixels available for embedding.

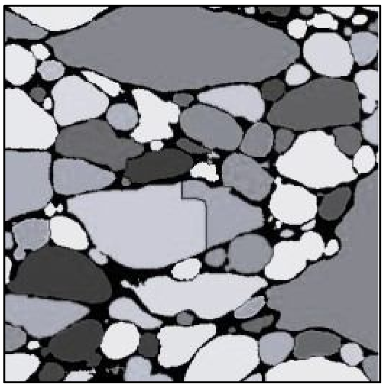

Fig 4.2: Grayscale image

The binary values are stored in 2D array representing the entire image where each pixel values corresponds to a row in a array.

```
001001010000001000000001
100001111000011010000001
100010011000100010001100
100010011000100110001001
100010011000100110001001
100010101000100110001001
100010011000101010000100
011110011000110110000111
000111000000001000101101
101111101100010101101000
```

Fig 4.3: Image to pixel

The binary representation of the image is then flattened into a 1D array, as we need to access each individual bit for modification during embedding.

```
01101000 01100101 01101100 01101100 01101111
```

Fig 4.4: Message in binary form

The secret text is converted into its binary form by character to its ASCII equivalent and then representing that ASCII value as an 8-bit binary number.

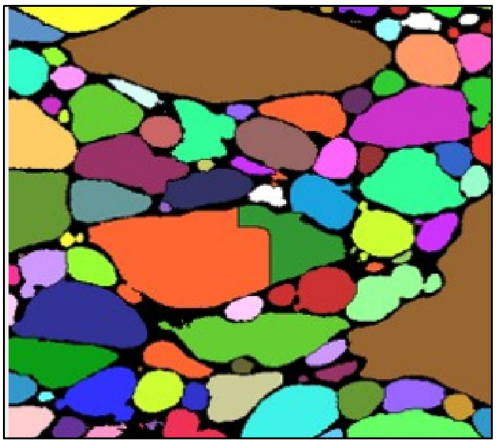

Fig 4.5: Modified image with message hidden

After encryption the message is hidden in the image

B. Outputs from Verilog

Embedding of Message

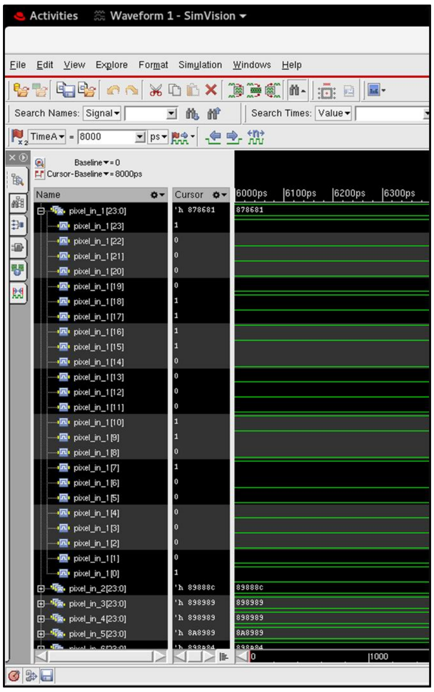

Fig 4.6: Input pixel 1 waveform

The embedding starts for the first pixel which consist of 24bits.

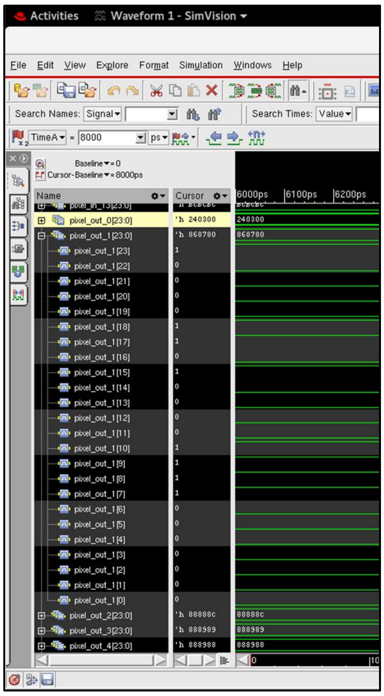

Fig 4.7: Output pixel 1 waveform

The message is hidden in 0,8 and 16th bit in 1st pixel

Recovery of Message

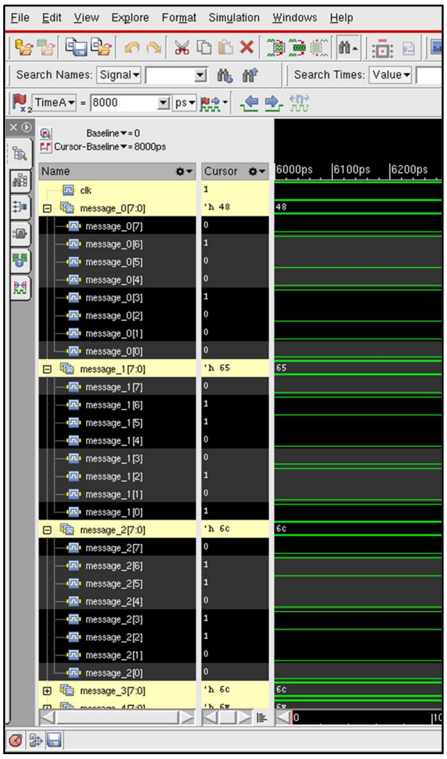

Fig 4.8: Input message [H-E-L] waveform

The algorithm iterates through the binary array of the secret message and embeds each bit in the LSB of the corresponding pixel of the cover image for H-E-L.

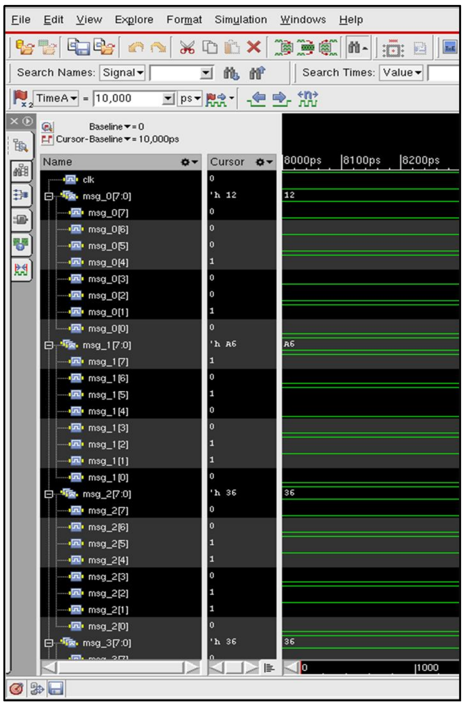

Fig 4.9: Recovered message [H-E-L] waveform

Original message(H-E-L) is recovered and is verified using MATLAB

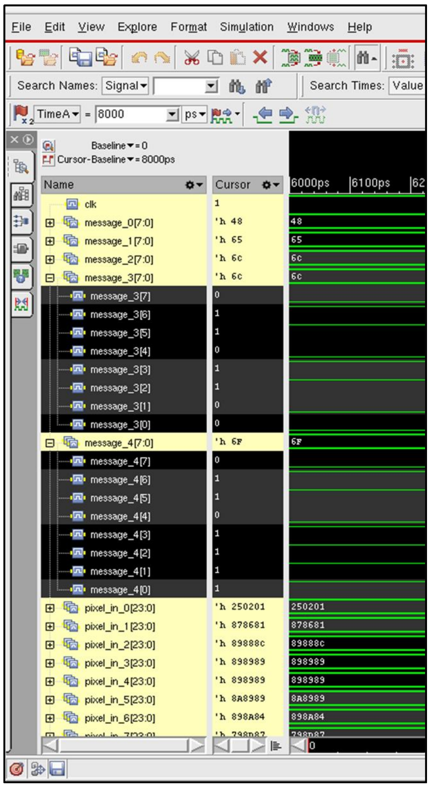

Fig 4.10: Input message [L-O] waveform

The algorithm iterates through the binary array of the secret message and embeds each bit in the LSB of the corresponding pixel of the cover image for L-O.

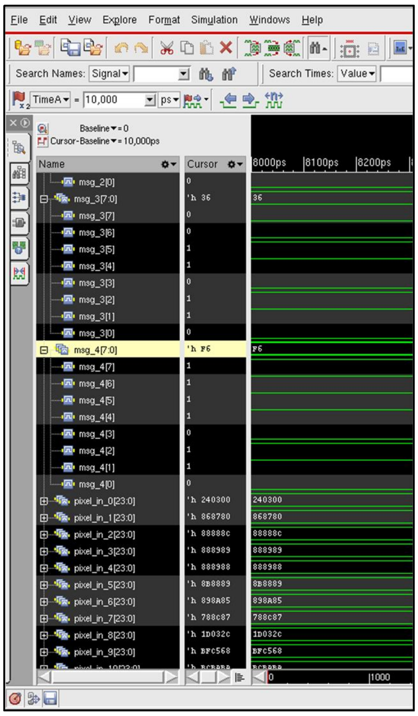

Fig 4.11: Recovered message [L-O] waveform

Original message(L-O) is recovered and is verified using MATLAB

V. Conclusion and Future Scope

This project successfully implements the Least Significant Bit (LSB) steganography technique using Verilog, demonstrating the feasibility of embedding secret messages into images without noticeable visual changes. The primary objective creating a secure and efficient method for hiding sensitive information was achieved. The technique ensures that the hidden message remains imperceptible to the human eye, validating its practicality for secure data transmission and storage. This approach highlights the potential of hardware-based steganography solutions in enhancing data confidentiality, particularly in digital communication systems. Overall, the project lays a strong foundation for utilizing LSB steganography as a viable solution for covert communication.

Several enhancements can be made to expand this project's utility and robustness. First, increasing the message capacity by utilizing a larger number of pixels or optimizing data embedding algorithms could enable the hiding of longer messages. Second, integrating cryptographic techniques alongside LSB steganography would enhance security, making the hidden messages more resistant to unauthorized access or tampering. Third, implementing error detection and correction mechanisms would ensure message integrity, particularly in environments where image quality may degrade due to compression or transmission noise. Additionally, exploring adaptive LSB techniques and applying machine learning for dynamic encoding could further improve performance and security, making the system more resilient to steganalysis.

References

[1] Abid Yahya- 2018-Steganography Techniques for Digital Images.

[2] Gandharba Swain -2018-Advanced Digital Image Steganography Using LSB, PVD, and EMD: Emerging Research and Opportunities.

[3] Jessica Fridrich,-2009- Steganography in Digital Media: Principles, Algorithms, and Applications

[4] Thomas O'Neil -2019- Voynich Morse Code Steganography Cipher

[5] H. Shi, X.-Y. Zhang, S. Wang, G. Fu, and J. Tang, ‘‘Synchronized detection and recovery of steganographic messages with adversarial learning,’’ in Proc. Int. Conf. Comput. Sci. Cham, Switzerland: Springer, 2019, pp. 31–43. https://ieeexplore.ieee.org/document/9335027

[6] Blessing, L.T.M. and Chakrabarti, A. (2009), DRM, a Design Research Methodology. London: Springer https://www.researchgate.net/publication/358680915_Image_Steganography_using_Least_Significant_Bit_LSB_-_A_Systematic_Literature_Review

[7] N. F. Johnson and S. Jajodia, ‘‘Exploring steganography: Seeing the unseen,’’ Computer, vol. 31, no. 2, pp. 26–34, Feb. 1998 https://ieeexplore.ieee.org/document/9952432

[8] Vaishnavi, A.; Pillai, S. Cybersecurity in the Quantum Era-A Study of Perceived Risks in Conventional Cryptography and Discussion on Post Quantum Methods. J. Phys. Conf. Ser. 2021, 1964,2002. https://www.mdpi.com/2076-3417/13/21/11771

[9] Cheddad, A., Condell, J., Curran, K., Mc Kevitt, P.: Digital image steganography: survey and analysis of current methods. Signal Process. 90(3),727–752,2010. https://joems.springeropen.com/articles/10.1186/s42787-019-0061-6

[10] Deepak Singla, and Rupali Syal. "International Journal of Computational Engineering Research." Citeseerx.ist.psu.edu. N.p., Mar-Apr. 2012. Web https://globaljournals.org/GJCST_Volume10/gjcst_vol10_issue_1_paper8.pdf

[11] Niels Provos, and Peter Honeyman. “Hide and Seek: An Introduction to Steganography.” IEEE Security & Privacy Magazine, May-June 2013. Web. https://assets.cambridge.org/97805211/90190/frontmatter/9780521190190_frontmatter.pdf

[12] Amanpreet Kaur;Gaurav Soni 2017 9th International Conference on Computational Intelligence and Communication Networks (CICN) https://ieeexplore.ieee.org/document/8319378

[13] Sumeet Kaur; Savina Bansal; R. K. Bansal -2014 -https://ieeexplore.ieee.org/document/6828087

[14] Research on Image Stegnography using based on Suduko matrix and LSB Value with multi-level encryption-2022 https://ieeexplore.ieee.org/document/9972616