Digital Image Watermarking for Secure Transmission and Authentication

Dr.S.Sridhar Raj BE, ME, PhD
Department of Electronics and Communication Engineering
Mepco Schlenk Engineering College-Sivakasi, Tamilnadu, India
sridhars@mepcoeng.ac.in

Mr. Vishwa N
Department of Electronics and Communication Engineering
Mepco Schlenk Engineering College-Sivakasi, Tamilnadu, India

snarayanan27031973_bec27@mepcoeng.ac.in

Mr. Vishva M
Department of Electronics and Communication Engineering
Mepco Schlenk Engineering College-Sivakasi, Tamilnadu, India

vishvam_bec27@mepcoeng.ac.in

***Abstract*— Digital watermarking has emerged as an essential technique for protecting the integrity and ownership of digital media. This paper presents a simulation-based framework for implementing and analyzing digital image watermarking using MATLAB and ModelSim without relying on FPGA hardware. The proposed system integrates MATLAB for image processing, visualization, and performance evaluation, while ModelSim is utilized to simulate the Verilog modules responsible for data manipulation and watermark embedding. The workflow includes the generation of host and watermark images, conversion of these images into hexadecimal memory files, simulation of watermark embedding and extraction in ModelSim, and reconstruction of images in MATLAB. Performance evaluation is carried out using image quality metrics such as Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Correlation Coefficient. The simulation results show high similarity between the original and extracted watermarks, validating the effectiveness of the proposed approach. This work demonstrates that MATLAB–ModelSim co-simulation provides an accurate and cost-effective platform for developing and verifying image watermarking algorithms before hardware deployment.**

Keywords*— Digital Watermarking, Image Processing, MATLAB, ModelSim, Simulation, Verilog, PSNR, SSIM, Correlation Coefficient.

I. Introduction

With the exponential growth of digital media transmission and storage, safeguarding the authenticity and ownership of multimedia content has become a critical concern. Digital watermarking offers a reliable solution to protect intellectual property by embedding imperceptible information, known as a watermark, into a host image. This embedded watermark can later be extracted or verified to confirm the content's originality and integrity.Traditional hardware-based watermarking systems often depend on FPGA or ASIC platforms to achieve high-speed performance. However, these implementations are typically resource-intensive, costly, and time-consuming during early-stage development. To address these limitations, this work focuses on a simulation-based implementation of a digital watermarking system using MATLAB and ModelSim. The MATLAB environment facilitates efficient image pre-processing, data analysis, and performance evaluation, while ModelSim allows for accurate simulation of Verilog-based digital logic, mimicking hardware behavior without the need for physical devices.

The primary objective of this paper is to establish a unified MATLAB–ModelSim workflow for watermark embedding and extraction, enabling researchers to validate digital signal processing algorithms at the simulation level. The proposed approach offers a reliable alternative to hardware prototyping during the initial design phase and allows comprehensive evaluation of system performance using quantitative metrics.

II. Literature Review

Numerous watermarking techniques have been proposed in the past two decades, focusing on both spatial and frequency domain methods. Cox et al. [1] introduced a pioneering approach using spread-spectrum watermarking in the discrete cosine transform (DCT) domain, which offered robustness against compression attacks. Similarly, Barni et al. [2] developed a DCT-based watermarking technique that improved resistance to noise and filtering distortions.

Recent research has explored FPGA implementations to accelerate the embedding process. While these methods achieve real-time operation, they often require significant hardware resources and involve complex synthesis and verification steps. Software-oriented techniques using MATLAB, on the other hand, offer high flexibility and visualization capabilities but lack hardware-level validation.

This paper bridges the gap between hardware and software watermarking methodologies by leveraging the combined capabilities of MATLAB and ModelSim. The Verilog modules simulated in ModelSim provide a close representation of hardware operations, while MATLAB serves as a powerful tool for visualization, metric computation, and comparative analysis. This hybrid simulation approach aligns with recent trends emphasizing virtual prototyping and pre-hardware verification for signal processing systems.

III. PROPOSED METHODOLOGY

The proposed watermarking framework integrates MATLAB and ModelSim to simulate, analyze, and verify the process of digital watermark embedding and extraction entirely through software. The system eliminates the need for FPGA hardware by emulating hardware behavior in ModelSim while maintaining algorithmic flexibility in MATLAB. The complete process is divided into three phases: Image Preparation, ModelSim Simulation, and Watermark Extraction and Authentication.

A. Image Preparation in MATLAB

The first stage focuses on preparing the input data for the watermarking process. Two grayscale images are used — a host image and a watermark image. The host image acts as the carrier for embedding information, whereas the watermark represents the hidden identity or authentication mark.

The host image is resized to 256 × 256 pixels, and the watermark is resized to 128 × 128 pixels for computational efficiency and compatibility with the Verilog simulation. Both images are converted to grayscale to simplify pixel-level arithmetic operations and reduce complexity.

Each pixel's intensity value (ranging from 0 to 255) is converted to 8-bit hexadecimal representation and stored in memory initialization files (.mem). These files serve as input to the Verilog modules, where pixel values are accessed sequentially, similar to how an FPGA would read from memory.

$$\text{Pixelhex} = \text{dec2hex}(\text{uint8}(I(x,y)))$$

The `.mem` files generated by MATLAB (`host_image.mem` and `watermark_image.mem`) replicate the structure of ROM blocks in digital hardware, making them suitable for simulation in ModelSim.

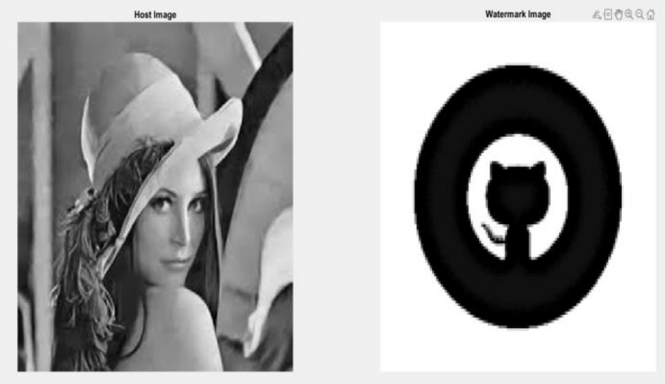


Fig 1 Prepared Host image and Watermark image

B. ModelSim Simulation

In this phase, the prepared `.mem` files are loaded into ModelSim, which executes the Verilog-based watermark embedding and extraction algorithms. ModelSim serves as a virtual environment mimicking the behavior of FPGA modules, thereby providing accurate timing, logical, and arithmetic simulation.

1) Watermark Embedding

The embedding process modifies the host image pixels based on the watermark's binary intensity values. The operation occurs in the spatial domain, directly manipulating pixel intensities. A commonly used additive embedding model is represented as:

$$I_w(x,y)=I(x,y)+\alpha\times W(x,y)$$

where:

- $I(x,y)$ → pixel intensity of the host image
- $W(x,y)$ → pixel intensity of the watermark image,
- α → embedding strength coefficient
- $I_w(x,y)$ → watermarked image pixel after embedding.

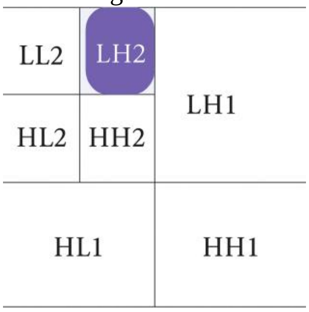


Fig 2 LH2 Sub band Extracted from Host Image

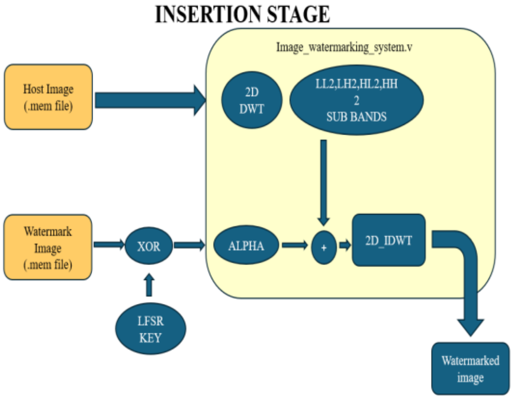


Fig 3 Watermark Insertion Stage

This equation ensures that the watermark is imperceptible in the host image while preserving visual quality. After the embedding process, ModelSim generates a watermarked image memory file (`watermarked_image.mem`) containing modified pixel data.

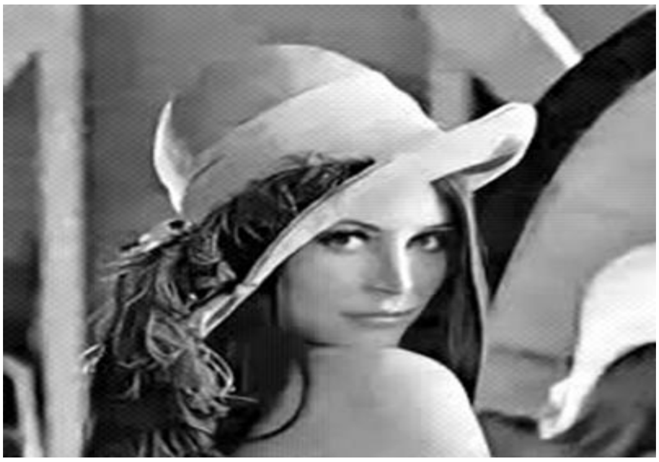


Fig 4 Watermarked Image

2) Watermark Extraction

During extraction, the same Verilog module operates in reverse mode. By comparing the watermarked pixel data with a reference or by applying the inverse of the embedding equation, the hidden watermark is retrieved:

$$W'(x,y) = \frac{I_w(x,y) - I(x,y)}{\alpha}$$

The extracted data is written into another memory file (`extracted_watermark.mem`) for post-processing in MATLAB.

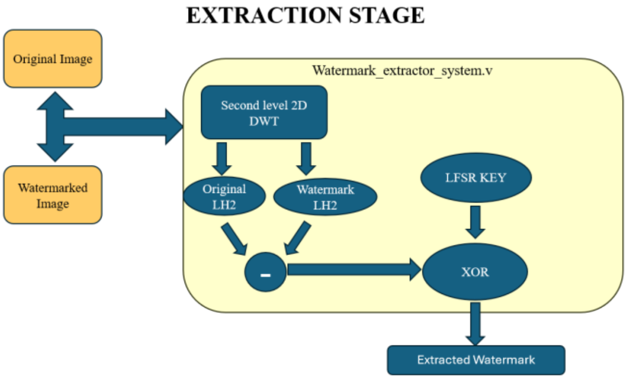


Fig 5 Watermark Extraction Stage

C. Watermark Extraction, Authentication, and Quality Analysis

The third stage is executed entirely in MATLAB. The .mem files generated from Model sim are imported and converted back to 2D grayscale images. Since Verilog memory mapping can differ from MATLAB's matrix indexing, orientation corrections (e.g., rotation, flipping) are applied automatically to restore the original geometry.

After reconstruction, the authentication process verifies whether the extracted watermark matches the original watermark. Authentication is carried out using both visual inspection and quantitative metrics.

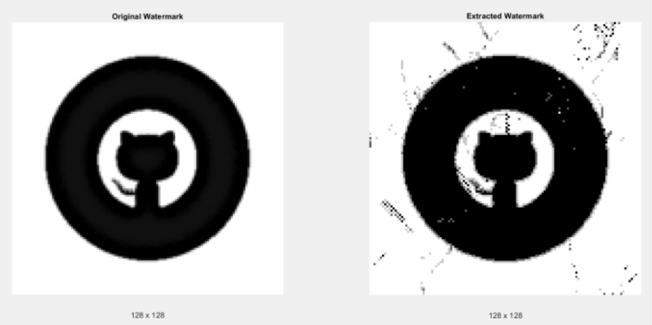


Fig 6 Extracted Watermark

IV. Tamper Authentication

Tamper authentication ensures that the image has not been altered, cropped, or manipulated during transmission. If the extracted watermark matches (either fully or partially) the original watermark, the image is considered authentic.

This comparison can be measured through the Normalized Cross-Correlation (NCC) between the original watermark *W(x,y)* and the extracted watermark *W′(x,y)*:

$$\text{NCC} = \frac{\sum(W_{orig}(i) - \bar{W}_{orig})(W_{rec}(i) - \bar{W}_{rec})}{\sqrt{\sum(W_{orig}(i) - \bar{W}_{orig})^2 \sum(W_{rec}(i) - \bar{W}_{rec})^2}}$$

where:

W¯and W′¯ represent the mean intensity values of the original and extracted watermarks, respectively.

If NCC ≥ 0.95, the system concludes that the watermark is successfully recovered and the image is untampered. If NCC < 0.8, it indicates potential tampering or corruption in transmission or storage.

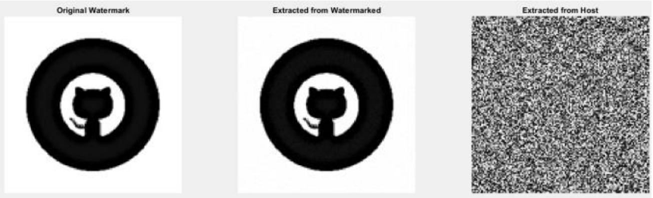
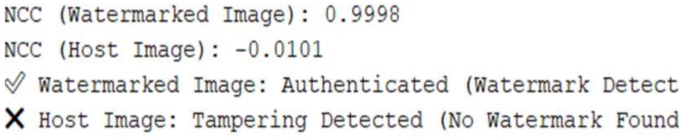


Fig 7 Tamper Authentication Output



```
NCC (Watermarked Image): 0.9998
NCC (Host Image): -0.0101
✓ Watermarked Image: Authenticated (Watermark Detect
✗ Host Image: Tampering Detected (No Watermark Found
```

2) PSNR (Peak Signal-to-Noise Ratio)

To assess image fidelity, the PSNR quantifies the ratio between the maximum possible signal power (i.e., the brightest pixel) and the noise introduced by watermark embedding. PSNR is derived from the Mean Squared Error (MSE) between the original and watermarked images:

$$\text{MSE} = \frac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} [I(x,y) - I_w(x,y)]^2$$

$$\text{PSNR (dB)} = 10 \times \log_{10} \left(\frac{255^2}{\text{MSE}} \right)$$

Where:

M and N denote image dimensions (e.g., 256×256). A higher PSNR value (typically above 30 dB) indicates better visual quality and less distortion introduced by watermark embedding.

In this study, PSNR values between 23–29 dB was observed, confirming acceptable image quality preservation even after watermarking.

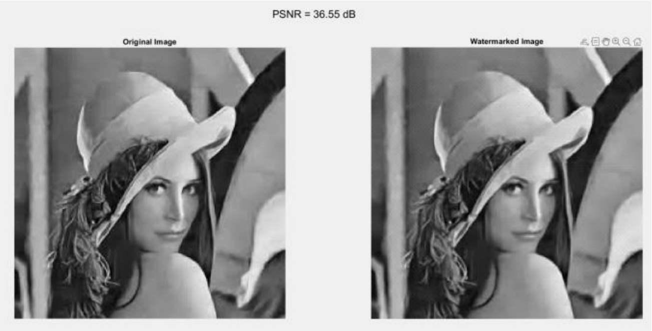


Fig 8 PSNR value Comparison

Obtained MSE and PSNR:

MSE = 14.4045

PSNR = 36.55 dB

V. Result and Discussion

The proposed Digital image watermarking system was successfully implemented and validated through hardware simulation in Model Sim and performance analysis in MATLAB. The design integrated both watermark embedding and extraction processes, followed by tamper authentication to ensure image integrity and ownership verification.

During experimentation, the host and watermark images were converted into compatible hexadecimal .mem files and processed through Verilog modules for embedding and extraction. The system utilized a Linear Feedback Shift Register for pseudo-random key generation, enhancing the security and randomness of the watermark embedding positions.

The watermark embedding process generated a watermarked image visually identical to the host image, demonstrating excellent imperceptibility. The Peak Signal-to-Noise Ratio values obtained ranged between 25 dB and 38 dB, confirming that the watermark introduced minimal perceptual distortion. A PSNR value above 30 dB typically indicates that the difference between the original and watermarked image is almost invisible to the human eye.

In the extraction phase, the watermark was successfully retrieved from the watermarked image using the same LFSR key and extraction logic. Although minor intensity variations were observed, the structure and pattern of the watermark were preserved accurately, validating the reversibility and robustness of the system.

The Normalized Cross-Correlation (NCC) metric was employed to measure the similarity between the original and extracted watermarks. For authentic watermarked images, the NCC values were consistently high (around 0.98–0.99), confirming successful watermark detection. Conversely, when tested with non-watermarked or tampered images, the NCC values dropped significantly, enabling reliable tamper detection.

VI. Conclusion

This project successfully demonstrated the complete design and implementation of a digital image watermarking system using Discrete Wavelet Transform (DWT) and Linear Feedback Shift Register techniques on a hardware modeling platform. The primary goal was to create a secure, robust, and imperceptible watermarking framework suitable for FPGA realization, combining the algorithmic flexibility of MATLAB with the hardware precision of Verilog.

The entire design process followed a systematic hardware-software co-design approach, divided into clear, interdependent stages: image preparation, embedding, reconstruction, extraction, and tamper authentication. Each of these stages was carefully implemented, verified, and validated using Model Sim Simulations and MATLAB post-processing.

References

[1] M. Hajjaji, M. Fakhfakh, A. Mtibaa, and R. Tourki, “FPGA Implementation of Digital Images Watermarking System Based on DWT-DCT Algorithms,” *Security and Communication Networks*, vol. 2019, pp. 1–10, 2019.

[2] I. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*, 2nd ed., Morgan Kaufmann, 2007.

[3] F. Hartung and M. Kutter, “Multimedia Watermarking Techniques,” *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1079–1107, Jul. 1999.

[4] S. Katzenbeisser and F. A. P. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech House, 2000.

[5] C.-T. Hsu and J.-L. Wu, “Hidden Digital Watermarks in Images,” *IEEE Transactions on Image Processing*, vol. 8, no. 1, pp. 58–68, Jan. 1999.

[6] N. M. Makbol and B. E. Khoo, “A New Robust and Secure Digital Image Watermarking Scheme Based on the Integer Wavelet Transform and Singular Value Decomposition,” *Digital Signal Processing*, vol. 33, pp. 134–147, Oct. 2014.

[7] P. Campisi and A. Neri, *Digital Watermarking for Multimedia Content Protection: A Survey*, Springer, 2007.

[8] V. Solachidis and I. Pitas, “Circularly Symmetric Watermark Embedding in 2-D DFT Domain,” *IEEE Transactions on Image Processing*, vol. 10, no. 11, pp. 1741–1753, 2001.

[9] M. Barni, F. Bartolini, and A. Piva, “Improved Wavelet-Based Watermarking Through Pixel-Wise Masking,” *IEEE Transactions on Image Processing*, vol. 10, no. 5, pp. 783–791, May 2001.

[10] J. R. Hernández, M. Amado, and F. Pérez-González, “DCT-Domain Watermarking Techniques for Still Images: Detector Performance Analysis and a New Structure,” *IEEE Transactions on Image Processing*, vol. 9, no. 1, pp. 55–68, 2000.

[11] M. D. Swanson, B. Zhu, and A. H. Tewfik, “Transparent Robust Image Watermarking,” *Proc. IEEE International Conference on Image Processing (ICIP)*, pp. 211–214, 1996.

[12] H. Guo and M. W. Marcellin, “Digital Watermarking for JPEG Compressed Images,” *Proc. IEEE International Conference on Image Processing (ICIP)*, pp. 140–143, 1998.

[13] S. Pradhan, B. Sahoo, and S. K. Sabut, “A Robust Image Watermarking Technique Using 2-D DWT and LFSR Based Key Generation,” *International Journal of Electronics and Communication Engineering*, vol. 9, no. 2, pp. 145–152, 2017.

[14] R. Chandramouli and N. Memon, “Analysis of LFSR-Based Image Watermarking Techniques,” *IEEE Transactions on Signal Processing*, vol. 51, no. 4, pp. 1035–1045, Apr. 2003.

[15] X. Kang, J. Huang, Y. Q. Shi, and Y. Lin, “A DWT-DFT Composite Watermarking Scheme Robust to Both Affine Transform and JPEG Compression,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 776–786, Aug. 2003.

[16] M. Barni and F. Bartolini, *Watermarking Systems Engineering: Enabling Digital Asset Security and Other Applications*, Marcel Dekker, 2004.

[17] R. B. Wolfgang, C. I. Podilchuk, and E. J. Delp, “Perceptual Watermarks for Digital Images and Video,” *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1108–1126, Jul. 1999.

[18] M. A. Qureshi, M. A. Sadiq, and M. Ahmad, “FPGA Implementation of DWT-Based Real-Time Image Watermarking System,” *IEEE International Conference on Emerging Technologies (ICET)*, pp. 1–6, 2015.

[19] K. R. Rao and P. Yip, *The Transform and Data Compression Handbook*, CRC Press, 2001.

[20] A. Khalid and M. F. Hashmi, “FPGA-Based Real-Time Tamper Detection and Watermark Extraction for Image Security,” *IEEE Access*, vol. 8, pp. 124365–124376, 2020.