Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN: 2581 - 611X)
VOLUME:9, ISSUE:1, JANUARY 2026, PAPER NUMBER:25

FPGA-BASED HARDWARE ACCELERATOR USING DWT FOR DIGITAL IMAGE WATERMARKING

Subasri Kamalakannan.ME[1], Revathy M.PhD[2]

[1], Students,and [2] Faculty
Dept. of Electronics and Communication Engineering(VLSI),
PSNA CET, Dindigul, Tamilnadu.
srikannan242001@gmail.com

Abstract: *This paper presents a hardware-accelerated image watermarking system implemented on a Field-Programmable Gate Array (FPGA) using the Discrete Wavelet Transform (DWT). The proposed architecture enhances the imperceptibility and robustness of watermark embedding while optimizing hardware resource usage and power consumption. Designed and simulated using the Xilinx Vivado toolchain on a Spartan-7 FPGA, the system embeds watermark data into selected frequency sub-bands to ensure high visual fidelity and resistance to signal degradation. Experimental results demonstrate excellent image quality with a Structural Similarity Index Measure (SSIM) of 0.9999, Peak Signal-to Noise Ratio (PSNR) of 54.52 dB, and Mean Squared Error (MSE) of 0.000004. The efficient hardware design confirms the suitability of FPGA-based implementations for real-time, secure multimedia processing.*
Keywords: *FPGA, Digital Watermarking, Discrete Wavelet Transform, SSIM, PSNR, MSE, Xilinx Vivado.*

1. INTRODUCTION:

Digital image watermarking is a crucial technology for ensuring the security of multimedia content, enabling applications such as copyright protection and content verification. To meet the real-time and low-power requirements of modern systems, hardware acceleration using FPGAs has become an increasingly attractive solution. This work presents an FPGA-based DWT watermarking system, targeting robust watermark embedding with minimal distortion.

2.DIGITAL WATERMARKING TECHNIQUES

Digital watermarking is a critical technology used to ensure the security and integrity of multimedia content. It involves embedding a piece of information, known as a watermark, into a digital medium such as an image, audio, or video. The embedded watermark can be later extracted or detected to verify the authenticity and ownership of the multimedia product. Depending on the application, digital watermarking techniques are categorized into several types based on different criteria, including the working domain, visibility, and robustness.

2.1 Classification Based on Working Domain

Spatial Domain Techniques embed watermark data directly into the pixels of the multimedia file. Common techniques include modifying the least significant bit (LSB) of image pixels. While spatial domain methods are computationally simple and efficient, they are less robust against various attacks such as compression or noise addition. Transform Domain Techniques embed the watermark by modifying the frequency coefficients of the multimedia data. Techniques such as Discrete Wavelet Transform (DWT), Discrete Cosine Transform (DCT), and Discrete Fourier. Transform (DFT) falls under this category. Transform domain techniques provide higher robustness and better imperceptibility

compared to spatial domain methods.

2.2 Classification Based on Visibility

In visible watermarking, the watermark is perceptible to the human eye, often displayed as a logo or text on the multimedia content. This technique is commonly used for branding or copyright claims in digital images and videos. Here Invisible Watermarking, The watermark is embedded in such a way that it is imperceptible to the human eye but can be detected or extracted using specific algorithms. Invisible watermarking is widely used for security and authentication purposes.

2.3 Classification Based on Robustness

Robust Watermarking techniques aim to ensure that the watermark remains intact even after the multimedia content undergoes common operations like compression, cropping, or noise addition. Robust watermarking is essential for applications such as copyright protection.
Fragile watermarks are designed to become distorted or destroyed if the content is tampered with. These are used to verify the integrity of the content.
Semi-fragile watermarking is designed as a middle ground between robust and fragile watermarking, these techniques can tolerate minor modifications, such as compression, while detecting more significant changes.

3.TRANSFORM DOMAIN TECHNIQUES:

Transform domain techniques are widely used in digital watermarking due to their robustness and ability to embed watermarks in a way that is imperceptible to the human eye. Unlike spatial domain methods, which directly manipulate pixel values, transform domain techniques work by converting the multimedia data into the frequency domain. By embedding watermark information into specific frequency components, these methods ensure that the watermark remains resistant to common image processing operations such as compression, filtering, and noise addition.
Transform domain techniques rely on mathematical transformations to analyze and modify the frequency components of an image. Commonly used transformations include:
DCT divides an image into different frequency components, concentrating most of the energy in low-frequency regions. Watermarks are embedded in these low or mid-frequency coefficients to achieve a balance between robustness and imperceptibility.
DFT represents an image in terms of its sinusoidal components. It is resilient to certain geometric transformations, such as rotation or scaling, making it suitable for specific applications.
DWT decomposes an image into multiple sub-bands, each representing different frequency components. This hierarchical decomposition provides excellent localization in both spatial and frequency domains, making it a preferred choice for digital watermarking.

4. FOCUS ON DISCRETE WAVELET TRANSFORM (DWT)

The Discrete Wavelet Transform (DWT) is a powerful transform domain technique that has gained significant popularity in digital image watermarking due to its unique properties. DWT decomposes an image into four sub-bands: Low-Low (LL), Low-High (LH), High-Low (HL), and High High (HH). These sub-bands represent different levels of detail in the image:

LL Sub-Band contains the approximation or coarse details of the image. This sub-band is most visually significant and is often targeted for watermark embedding to ensure robustness.

LH and HL Sub-Bands contain horizontal and vertical edge details, respectively. These are moderately significant and can also be used for embedding less critical watermark information.

HH Sub-Band contains diagonal edge details and represents high-frequency components, often less significant for human perception.

5. FIELD-PROGRAMMABLE GATE ARRAYS (FPGAS)

Field-Programmable Gate Arrays (FPGAs) have become an essential technology for accelerating computation-intensive tasks in various domains, ranging from digital signal processing to machine learning and cryptography. Their ability to provide hardware-level parallelism, flexibility, and energy efficiency makes them a preferred choice for tasks requiring high-speed computation. Unlike traditional general purpose processors, FPGAs allow for custom hardware design tailored to specific algorithms, significantly improving performance and reducing power consumption. Tasks such as image filtering, feature extraction, and video compression benefit greatly from the parallelism and low latency of FPGAs. FPGA accelerators are used for deep learning inference, especially in edge computing scenarios where power efficiency is critical. High-speed encryption and decryption algorithms are often implemented on FPGAs for secure communications. FPGAs are widely used in wireless communication systems for tasks such as modulation, coding, and packet processing. FPGAs can perform multiple operations simultaneously, enabling faster execution of complex algorithms. Unlike CPUs or GPUs, FPGAs allow the design of application-specific hardware architectures optimized for a particular task. FPGAs consume significantly less power compared to traditional processing units, making them ideal for energy-sensitive applications. The deterministic nature of FPGA hardware ensures consistent and low-latency performance, essential for real-time applications.

6.DISCRETE WAVELET TRANSFORM AND ITS SIGNIFICANCE

The Discrete Wavelet Transform (DWT) is a powerful mathematical tool used for analyzing and processing signals, including images. It provides a representation of a signal in both the time (or spatial) and frequency domains, offering better localization of frequency information compared to other transforms like the Discrete Fourier Transform (DFT) or Discrete Cosine Transform (DCT). This property makes DWT particularly useful in image processing tasks, including compression, denoising, and digital watermarking.

6.1 Discrete wavelet transform

The DWT works by decomposing a signal or image into sub bands that represent various levels of detail. It applies a pair of filters: A low-pass filter that captures the coarse approximation of the signal (low-frequency components). A high-pass filter that extracts finer details (high-frequency components).

After filtering, the signal is downsampled to reduce the data size, effectively separating it into different scales. For images, this process results in four sub-bands at each decomposition level. LL (Low-Low) represents the approximation of the image, containing coarse details and most of the energy. LH (Low-High) captures horizontal edge information. HL (High Low) Captures vertical edge information. HH (High-High) contains diagonal edge details. This decomposition can be performed iteratively on the LL sub-band to create a multi level, hierarchical representation of the image.

6.2 Steps of dwt in image processing

The input image is passed through low-pass and high-pass filters along rows and columns, generating the LL, LH, HL, and HH sub-bands. Watermark data is embedded into one or more sub-bands (commonly the LL or LH sub-band) to achieve a balance between robustness and imperceptibility. After watermark embedding, the image is reconstructed using the inverse DWT (IDWT) by combining the modified sub bands.

7 PYTHON: PROGRAMMING LANGUAGE

Python is one of the most popular programming languages in the modern era due to its simplicity, versatility, and wide range of applications. It has gained immense recognition in various fields, including web development, data science, artificial intelligence, and embedded systems. This paper provides an overview of Python, discussing its features, applications, and advantages, along with its impact on the software industry. Python is a high-level, interpreted programming language known for its easy syntax and readability. Developed by Guido van Rossum and released in 1991, Python was designed to emphasize code readability and efficiency. It has grown significantly over the years and is widely used across multiple domains due to its extensive libraries and community support.

8 Proposed Methodology

In this project, the Discrete Wavelet Transform is adopted to transform the given image from the spatial domain to the frequency domain, allowing watermark data to be embedded into specific sub-bands. The performance of the DWT-based watermarking is evaluated using standard metrics:

8.1. SSIM (Structural Similarity Index)

The Structural Similarity Index (SSIM) is a metric used to assess the perceptual similarity between two images, focusing on aspects such as luminance, contrast, and structure. SSIM values range from 0 to 1, with values closer to 1 indicating higher similarity and minimal perceptual distortion. In the context of image watermarking, a high SSIM value between the original and watermarked images suggests that the watermarking process has preserved the visual quality of the original image, ensuring the watermark remains imperceptible to human observers. This makes SSIM a valuable tool for evaluating the effectiveness of watermarking techniques in maintaining image integrity.

$$SSIM(x,y) = \frac{(2\mu x\mu y + C1)(2\sigma xy + C2)}{(\mu_x^2 + \mu_y^2 + C1)(\sigma_X^2 + \sigma_Y^2 + C2)}$$

8.2. Peak Signal-to-Noise Ratio (PSNR)

PSNR quantifies The Structural Similarity Index (SSIM) is a metric used to assess the perceptual similarity between two images, focusing on aspects such as luminance, contrast, and structure. SSIM values range from 0 to 1, with values closer to 1 indicating higher similarity and minimal perceptual distortion. In the context of image watermarking, a high SSIM value between the original and watermarked images suggests that the watermarking process has preserved the visual quality of the original image, ensuring the watermark remains imperceptible to human observers. This makes SSIM a valuable tool for evaluating the effectiveness of watermarking techniques in maintaining image integrity.

The ratio between the maximum possible power of the image signal and the power of the distortion introduced by watermark embedding. It is measured in decibels (dB).

A higher PSNR value indicates better image quality, with less distortion caused by watermark embedding. It helps balance the trade-off between imperceptibility and robustness. Commonly used in image processing tasks to evaluate distortion or quality degradation.

$$PSNR = 10\ log_{10}(\frac{MAX^2}{MSE})$$

8.3 Mean Squared Error (MSE)

MSE measures the average squared difference between pixel intensities in the original and watermarked images. It quantifies the level of distortion introduced. A lower MSE value indicates minimal distortion due to watermark embedding. It complements PSNR by providing raw distortion measurements.

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}[I(i,j) - W(i,j)]^2$$

9 HOW PYTHON IS USED FOR THE SIMULATION AND EVALUATION OF THESE METRICS

Python, with its extensive libraries and tools, is a powerful platform for implementing and simulating digital watermarking algorithms. Here is how Python is utilized in this project.

9.1 Implementation of Watermarking Algorithms

Python provides libraries like NumPy, OpenCV, and PyWavelets for efficient image processing and mathematical computations:

NumPy: Used for handling image matrices and performing mathematical operations. OpenCV: Facilitates image reading, writing, and preprocessing. PyWavelets: Implements the Discrete Wavelet Transform (DWT) and Inverse DWT for embedding and extracting watermarks.

9.2 Metric Calculation

Python's flexibility allows direct calculation of SSIM, PSNR, and MSE. SSIM Libraries such as scikit-image or OpenCV provide built-in functions to compute SSIM. Python From skimage.metrics import structural_similarity as Tim

ssim_value = ssim(original_image, watermarked_image)

PSNR can be calculated using mathematical formulas directly or via libraries. Python import cv2 psnr_value = cv2.PSNR(original_image, watermarked_image)

MSE can be computed using simple NumPy operations. Python import numpy as

np mse_value = np.mean((original_image - watermarked_image) ** 2)

9.3 Visualization

Python libraries like Matplotlib are used to visualize original and watermarked images, as well as performance results:

Compare images side by side.

Display plots of metric values across different embedding strengths.

9.4 Performance analysis

The calculated metrics are analyzed to assess:

Imperceptibility: Using SSIM and PSNR.

Distortion: Using MSE.

Trade-offs: Between robustness and imperceptibility by experimenting with different embedding parameters.

Fig 1. Performance Output

This paper presents a DWT-based watermarking algorithm and evaluates its performance using key quality metrics, including Peak Signal-to-Noise Ratio (PSNR), Mean Squared Error (MSE), and Structural Similarity Index Measure (SSIM). The results demonstrate a high level of imperceptibility and robustness, with a PSNR of 54.52 dB, an MSE of 0.000004, and an SSIM of 0.999947. The watermarking algorithm successfully embedded the watermark while maintaining excellent image quality. These results confirm that the proposed approach ensures effective watermark embedding with minimal perceptual degradation.

The high PSNR and SSIM values indicate that the watermarking method preserves image quality while maintaining robustness.

10 WATERMARKED IMAGE QUALITY ASSESSMENT RESULTS

This image serves as an excellent test subject for digital image watermarking techniques due to its rich textures, intricate details, and diverse color gradients. The complexity of the scene allows for evaluating watermarking robustness, imperceptibility, and preservation of visual quality after embedding.

Fig 2. Original image

Fig 3. Watermark image

PSNR:54.52dB, indicating high imperceptibility. MSE: 0.000004, showing minimal differences between the original and watermarked image. SSIM: 0.999947, confirming that the structural and perceptual quality is nearly unchanged. These results demonstrate that the watermark is successfully embedded without degrading the image quality, ensuring that the watermarking process is both effective and secure.

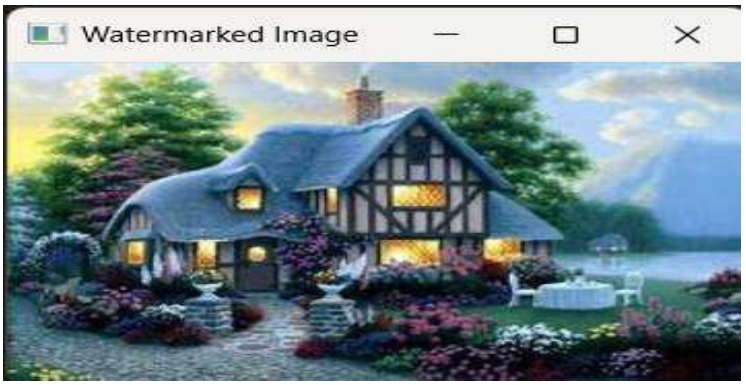

Fig 4. Watermarked image

11 OVERVIEW OF FPGA

Field Programmable Gate Arrays (FPGAs) are integrated circuits designed to be configured by users after manufacturing. Unlike traditional fixed-function chips, such as Application-Specific Integrated Circuits (ASICs), FPGAs offer a high degree of flexibility, allowing developers to create custom hardware solutions tailored to specific applications. FPGAs are composed of programmable logic blocks, configurable interconnect, and I/O ports, enabling the implementation of a wide variety of digital circuits. FPGAs are widely used in fields such as telecommunications, automotive systems, medical imaging, and digital signal processing, thanks to their ability to process computation intensive tasks in real-time. The architecture of an FPGA makes it suitable for parallel processing, where multiple computations can be carried out simultaneously, significantly reducing processing time.

12 BENEFITS OF FPGA

TECHNOLOGY 12.1 Reconfigurability

FPGAs can be reprogrammed multiple times, allowing designers to modify or upgrade functionality without changing the physical hardware. This feature is particularly useful for prototyping and testing applications, as it reduces time and costs associated with hardware redesign.

12.2 Parallel Processing

FPGAs support parallel execution of tasks by allowing multiple logic blocks to operate concurrently. This parallelism provides a significant speed advantage for computation-heavy applications, such as image processing, digital signal processing, and machine learning.

12.3 Real-Time Processing

FPGAs enable real-time data processing with low latency, making them ideal for time-critical applications like watermark embedding, video streaming, and industrial control systems.

12.4 Hardware-Level Optimization

Developers can design custom logic circuits to optimize for specific tasks, achieving higher performance and efficiency compared to general-purpose processors.

12.5 Cost-Effective for Custom Solutions

While FPGAs may have higher initial costs than traditional processors, their reconfigurability and flexibility make them cost-effective for applications that require frequent updates or customization.

13 DESIGN METHODOLOGY FOR IMPLEMENTING DWT ON FPGA

The implementation of the Discrete Wavelet Transform (DWT) on an FPGA requires a carefully planned methodology to ensure efficient performance, accuracy, and hardware resource utilization. The following design methodology outlines the key steps and considerations for this implementation.

13.1 Algorithm Analysis and Partitioning

The DWT decomposes an image into approximation and detail components (low-pass and high-pass sub-bands) by applying wavelet filters. This involves repeated convolution and down sampling operations.

2D DWT For Image watermarking, a 2D DWT is applied. The rows and columns of the image matrix are processed sequentially to generate sub-bands (LL, LH, HL, and HH). These sub-bands represent

different frequency components of the image.

The DWT operation is divided into smaller computation blocks (e.g., convolution, down-sampling) to leverage FPGA's parallel processing capability.

13.2 Hardware Architecture Design

The DWT is implemented using hardware modules for the low-pass and high-pass filtering operations. Separate modules are designed for convolution, down-sampling, and control logic.

A pipelined architecture is used to overlap the processing stages, increasing throughput. For example, while one row is being processed, another can be fetched and prepared.

Rows or blocks of the image can be processed in parallel to exploit the FPGA's parallelism. This reduces the overall computation time.

13.3 FPGA-Specific Optimizations

Efficient mapping of operations to FPGA resources, such as DSP slices for multipliers, BRAM for intermediate storage, and logic elements for control, is critical.

The FPGA's on-chip memory (BRAM) is used to store intermediate results, ensuring low-latency data access. External memory (e.g., DDR) is used only for large images.

Fixed-point representation is employed instead of floating point arithmetic to reduce computational complexity and resource usage without significantly affecting accuracy.

13.4 Xilinx Vivado

The DWT design is modeled in HDL (VHDL/Verilog) or using High-Level Synthesis (HLS) in C/C++. Xilinx Vivado provides simulation tools to verify the correctness of the DWT design at both functional and timing levels.

The design is synthesized to generate a netlist and implemented on the FPGA fabric by mapping it to the available resources.

The design is synthesized to generate a netlist and implemented on the FPGA fabric by mapping it to the available resources. Constraints are applied to meet timing requirements. The Vivado Design Suite offers tools for identifying and resolving timing bottlenecks.

13.5 Parallel Computing Strategies for DWT Acceleration

Row and Column Processing in Parallel For a 2D DWT, rows, and columns of the image matrix can be processed in parallel to reduce latency.

This involves duplicating hardware modules for simultaneous operation. A pipelined design is implemented to ensure continuous data flow. As one stage completes processing, the next stage begins operation without waiting for the entire operation to finish. Block-Based Process for Large images are divided into smaller blocks, which are processed in parallel.

This reduces memory requirements and speeds up computation. Wavelet Filter Bank Duplication for Multiple instances of the wavelet filter bank are deployed to handle several rows or blocks concurrently.

13.6 Performance Metrics and Validation

FPGA resource utilization, including logic elements, DSP slices, and BRAM, is monitored to ensure the design fits within the FPGA's capacity. The design's processing speed is evaluated by measuring latency for a given image size.

FPGA's clock frequency and pipeline efficiency play a significant role in determining performance. The output of the hardware-accelerated DWT is compared with software-based DWT implementations (e.g., Python) to ensure correctness.

14 POWER ANALYSIS OF FPGA IMPLEMENTATION

Efficient power management is crucial for FPGA-based implementations, particularly in applications requiring high computational intensity, such as digital image watermarking. This study presents a power analysis of an FPGA implementation utilizing a Xilinx Spartan-7 device, focusing on power distribution among logic components, DSP units, I/O, and signals. The analysis provides insights into dynamic and static power consumption, contributing to the optimization of FPGA-based watermarking systems. FPGAs offer a balance between power efficiency and computational performance, making them suitable for image processing applications. However, understanding power dissipation is necessary to optimize system performance and ensure thermal stability. This study evaluates the power consumption of an FPGA-based watermarking design using Xilinx Vivado's power analysis tool.

14.1 Power Analysis Results

The power analysis was performed on an implemented netlist, considering constraints, simulations, and vectorless estimations. The total on-chip power was measured at 7.634 W, comprising 7.518 W (98%) dynamic power and 0.116 W (2%) static power. The distribution of dynamic power across major FPGA components is as follows:

Signals: 0.584 W (8%)

Logic elements: 0.378 W (5%)

DSP blocks: 2.268 W (30%)

I/O interfaces: 4.287 W (57%)

The FPGA junction temperature reached 43.6°C, with an ambient temperature of 25.0°C and an effective thermal resistance (θJA) of 2.4°C/W. The confidence level for switching activity estimation was classified as low, indicating potential inaccuracies due to unspecified constraints or missing simulation activity

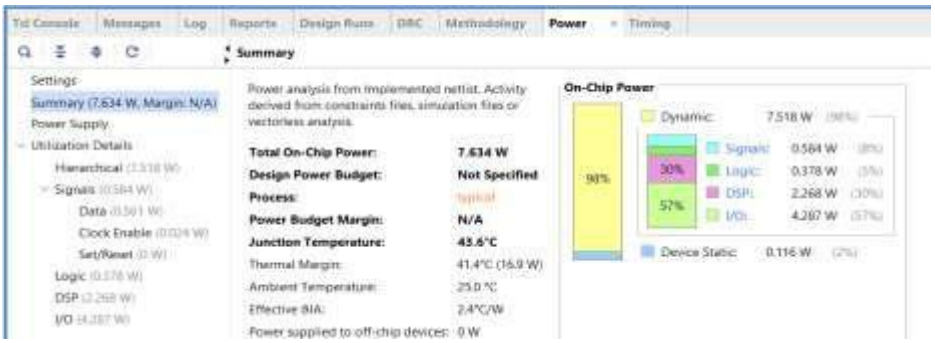

Fig 4. Power Analysis Output

14.2 Discussion and Optimization Strategies

The power consumption analysis reveals that the I/O interfaces and DSP units contribute significantly to the total power dissipation, accounting for 57% and 30% of dynamic power, respectively. This suggests that optimization efforts should focus on: Reducing I/O power consumption by minimizing high-frequency data transfers and optimizing signal toggling rates. Enhancing DSP efficiency by

leveraging parallelism and low-power arithmetic operations. Lowering static power dissipation through dynamic voltage and frequency scaling (DVFS)

Further refinement using the Power Constraint Advisor in Xilinx Vivado is recommended to improve power estimation accuracy and identify unnecessary switching activity.

14.3 Dataflow Architecture

The dataflow architecture of the proposed FPGA-based watermarking system is designed to maximize efficiency and parallelism. The clock signal is managed using a global buffer to synchronize processing units.

Two DSP48E1 blocks are utilized for processing the DWT coefficients, one dedicated to high-frequency component extraction (high_out1) and the other for low-frequency component extraction (low_out1).

The high-frequency and low-frequency components are processed simultaneously, improving computational speed. The computed coefficients are mapped to output ports, each carrying 16-bit data streams for further processing or storage.

14.4 Hardware Implementation and Simulation

The FPGA implementation leverages the Spartan-7 platform with the following hardware specifications:

DSP blocks optimized for DWT calculations. 44 registers for intermediate data handling. LUT Utilization: 50 LUTs for logic operations. 68 ports facilitating data exchange.

The developed architecture was simulated and verified using Xilinx Vivado, confirming accurate DWT coefficient computation and high-performance execution.

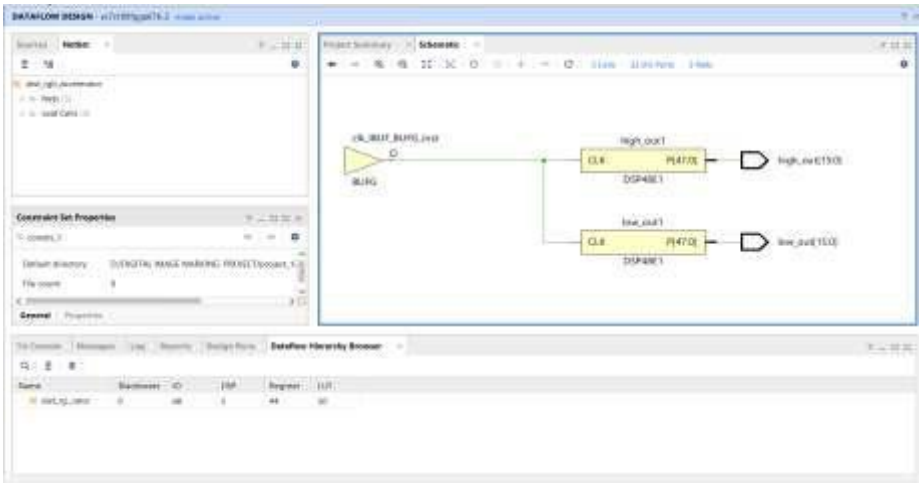

Fig 5. Data flow Output

15 CONCLUSION

SSIM, PSNR, and MSE are critical metrics for evaluating the effectiveness of digital watermarking. By leveraging Python's computational capabilities, these metrics can be efficiently implemented, analyzed, and visualized, ensuring that the watermarking system achieves an optimal balance between imperceptibility, robustness, and security. The design and implementation of DWT on FPGA using Xilinx Vivado leverage the parallel computing capabilities and reconfigurable nature of FPGA hardware. Key strategies, such as pipelining, block-based processing, and resource optimization, enable

high-speed and low-power computation. The efficient use of FPGA resources makes it an ideal platform for real-time computation-intensive tasks like image watermarking. This approach ensures significant improvements in speed and power efficiency compared to traditional software-based implementations.

16 REFERENCES

[1] Ramyashree, p. S. Venugopala,s. Raghavendra and b. Ashwini "cryptic care: a strategic approach to telemedicine security using LSB and DCT steganography for enhancing the patient data protection" Volume 12, 2024

[2] Shahid Rahman, Jamal Uddin Hameed Hussain, Ayaz Ali Khan, Muhammadzakarya, Aftab Ahmed, Andmuhammadhaleem “A Comprehensive Study of Digital Image Steganographic Techniques” Volume 11, 2023

[3] Zhong-xun Wang, Kai-Yue Sha, and Xing-long Gao “Digital Watermarking Technology Based on LDPC Code and Chaotic Sequence” Volume 10, 2022

[4] Juntao Ma, Jie Chen, and Gang Wu “Robust Watermarking via Multi-domain Transform Over Wireless Channel: Design and Experimental Validation" Volume 10,2022

[5] Ferda Ernawan, Dhani Ariatmanto, and Ahmadfirdaus "An Improved Image Watermarking by Modifying Selected DWT-DCT Coefficients" Volume 9,2021

[6] Wafahamdanalshoura, Moatsum alawida, Zurinahni Zainol, Jesen Teh And Abdulatif Alabdulatif “Hybrid SVD-Based Image Watermarking Schemes: A Review” Volume 9, 2021

[7] Yanpeng Cao, Feng Yu, and Yongming Tang "A Digital Watermarking Encryption Technique Based on FPGA Cloud Accelerator" Volume 8, 2020 [8] Oleg Evsutin, Anna Melman, and Romanmeshcheryakov “Digital Steganography and Watermarking for Digital Images: A Review of Current Research Directions” Volume 8, 2020

[9] Mohamed Ali Hajjaji, Mohamed Gafsi Abdessalem Ben Abdelali and Abdellatif Mtibaa (2019) “FPGA Implementation of Digital Images Watermarking System Based on Discrete Haar Wavelet Transform” Article ID 1294267 Hindawi

[10] Alejandra Menendez-Ortiz, Claudia Feregrino-Uribe, Rogelio Hashimoto-Beltran, and Jose Juan Garcia-Hernandez "A Survey on Reversible Watermarking for Multimedia Content: A Robustness Overview" Volume 7, 2019

[11] Sung-Woo Byun, Heui-su Son, and Seok-Pil Lee “Fast and Robust Watermarking Method Based on DCT Specific Location” Volume 7, 2019