IJARST
International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

VLSI IMPLEMENTATION OF BLIND IMAGE WATER MARKING SYSTEM USING HYBRID ADDRESS AND MULTIPLIERS

[1]S.Yasaswini, [2]Mrs. U.S.R.L. Sai Kumari, [3]Ch.V.L.S.Tejaswini, [4]G.Vyshnavi, [5]Sk.Hima Bindu.

[2]Asst.Prof, ECE Dept, RISE Krishna Sai Prakasam Group of Institution, Ongole-523001, AP

[1,3,4,5]B.Tech final year students, ECE Dept, RISE Krishna Sai Prakasam Group of Institution, Ongole-523001, AP

(yasaswinisiddabathuni@gmail.com, ramyauppal@gmail.com, Tejaswinichinnu23@gmail.com, vyshnavigudipalli0@gmail.com, Himabindu912103@gmail.com)

Abstract

One of the ways to protect the intellectual property rights of the digital media is digital watermarking. With rapid increase in use of internet and digital media, transmission and reproduction of digital products has become very convenient, but it has some drawbacks also. The digital revolution provides tools to unlimited copying without loss in fidelity. The people can easily steal the digital work of others like image, videos, and audio clips and claim their rights on the stolen things. This situation creates the need of copyright protection of digital media. Digital watermarking can be used for content authentication, detection of illegal duplication and alteration, secret communication as watermarking is very helpful in hiding the secret messages or information in the digital media. Using watermarking the people can keep their work copyrighted. The watermarking algorithm incorporates the watermark in the object, whereas the verification algorithm authenticates the object by determining the presence of the watermark and its actual data bits. The VLSI implementation of digital watermarking system must meet the low area (look up tables, slices, registers, memory elements), delays (path, logical) delays. However, the conventional wavelet transform based VLSI watermarking systems suffers with higher consumption of these attributes due to down sampler. So, this work is focused on implementation of hybrid stationary wavelet transform based watermarking system using VLSI fundamentals.

Keywords: Water Marking, Adders, Multipliers, Image Processing

1. Introduction

The main goal of this thesis is to embed a secret message, also known as a watermark, into an image in such a way that it remains invisible and cannot be removed or altered without significant distortion of the image. Previously, a significant research focus has been given to protect digital rights of images and data. In our work we investigate how image is protected and how it is extracted with matlab and vlsi.It is suitable for applications that require fast and efficient processing of large amounts of data, such as multimedia and security systems. The hybrid adders also improve the speed and efficiency of the system, making it more practical for real-world applications. It is a powerful technique for embedding and extracting hidden messages in images, while maintaining the integrity of the original image. The techniques described in this thesis are based on perceptual research which we extend by our own measurements as required for our applications. The novel contributions of the VLSI based image watermarking system using stationary wavelet transform with hybrid adders project can be summarized as follows: Development of a high-performance watermarking system.The proposed system is designed to achieve high watermarking capacity while maintaining robustness against common image processing attacks. This makes it a more effective and practical solution for protecting the copyright of digital images. Use of the stationary wavelet transform for image decomposition terms of speed, power consumption, and area. This improves the performance of arithmetic operations required for watermarking and makes the system more practical for real-world applications. Implementation of the system in VLSI hardware. The system is implemented in VLSI hardware, making it suitable for real-world applications that require fast and efficient processing of large amounts of data, such as multimedia and security systems. Overall, the novel contributions of this project make it a promising approach to addressing the challenges associated with implementing image watermarking algorithms in VLSI hardware, and it has the potential to contribute to the development of more effective and efficient solutions for protecting digital images in the future. The VLSI based image watermarking system using stationary wavelet transform with hybrid preserving the image quality and integrity. The system uses the stationary wavelet transform to decompose the image into its frequency components, allowing for the selective manipulation of the high-frequency components that are more sensitive to changes in the image. The hybrid adders are used to improve the performance of arithmetic operations required for watermarking. The system is implemented in VLSI hardware, making it suitable for real-world applications that require fast and efficient processing of large amounts of data, such as multimedia and security systems. The system has high watermarking capacity and robustness against common image processing attacks, making it an effective solution for protecting the copyright of digital images. adders is a technique for embedding a secret message, or watermark, into a digital images.

2. Literature Survey

Kumar, M. Pradeep (2021) [1], demonstrated various methods of digital watermarking techniques. In this approach ato watermark the image in the digital camera we can integrate the watermarking chip within framework, to demonstrate that this chip can embedded the watermark directly at the stage of capturing process itself which acts as a real time insertion and reduces the time consuming process of externally adding the watermark to the original image or video after capturing it. Illegal recovering and transmission of multimedia data can be protected via watermark or digital signature or copyright label that authenticates the data. This approach extended the opportunity of protecting the content after the release of content to the open environment.Jayanthi, V. E., Senthil Pitchai, and M. Smitha (2022) [2], proposed an approach is mainly designed for watermarking the images taken from digital cameras of various sizes. The padding technique is used for unequal sizes of the watermark image and original host image. The architecture data path consists of eight and six stages of pipeline capable of watermarking on the pixel-based operation and vector-based operation, respectively. The dual image watermarking architecture data path consists of a 13-stage pipeline. Pipeline and parallelism mechanisms are used to improve throughput. To improve the performance in discrete cosine transform operations at the frequency domain, the shift-add technique replaces the conventional multipliers. The clock gating technique is employed to reduce the power by preventing unnecessary switching in a path.Sakthivel, S. M., and A. Ravi Sankar (2020) [3], presented a computation efficient image watermarking very large-scale integration (VLSI) architecture with improved robustness and throughput. The computational complexity is reduced by the incorporation of integer DCT, whereas the throughput is improved by the implementation of a four-stage pipeline architecture. The secret watermark bits are inserted in the DCT low-frequency coefficients using a mean adaptive threshold value, which not only results in a minimum degradation to the host image quality but also increases the robustness of the proposed algorithm. Penchalaiah, Usthulamuri, and VG Siva Kumar (2020) [4], implemented an image watermarking using LSB technique to hide a secret image, and employed encryption using Advanced Encryption Standard, to enhance the security of the image. An image is a two-dimensional signal, with each pixel value representing the intensity level. The secure transmission of the image along the channel is a challenging task, because of the reason that, any individual can access it, if no security measures are taken. Hardware realization of image watermarking/encryption and dewatermarking/ decryption is implemented using Very Large-Scale Integration. The design is verified by means of co-simulation using MATLAB and Xilinx. Kaibou, Redouane (2021) [5], proposed a new approach for designing invisible non-blind full crypto- watermarking system targeting images security on FPGA platform is presented. This new design is based on the Hardware-Software co-design approach using the High-Level Synthesis (HLS) tool of Xilinx which allows a good compromise between development time and performances. For a better authentication and robustness of the proposed system, the Discrete Wavelet Transform (DWT) is employed. To more enhance the security level, a new chaos-based generator proposed is integrated into a stream cipher algorithm in order to encrypt and decrypt the watermark during the insertion and extraction phases. Thakur, S. (2020) [6], demonstrated an efficient watermarking technique to enhance the performance of DWT-SVD-based approach. The method uses well-known error-correcting code (ECC) and chaotic encryption to reduce the channel noise distortion and improve the security of the technique, respectively. In this method, the cover image is transformed by DWT and the sub-bands are selected for embedding the watermarks. Subsequently, the selected sub-bands are further transformed by SVD. The more robust watermark “patient report” and less robust watermark “patient medical image” is embedded into singular values of the selected DWT sub-bands. The use of transform domain techniques along with hamming code ensures that the approach offers more robustness and reliability. Nagaraj, P. (2020) [7], proposed VLSI implementation of HAAR wavelet- based image compression is proposed and designed. HAAR wavelet transform is one of the easiest methods for image compression because it has coefficients as either 1 or -1. Here software alone is used for the compression together with optimizing it with a continuous optimization algorithm and provides a hardware-free architecture with low cost. The VHDL work is carried out in Xilinx Platform and provides a truncated power architecture for a concrete application. Joshi, Amit M (2017) [8], proposed a method which uses Integer DCT with an error correction concept to improve the robustness of the watermarking system. The performance of proposed algorithm is validated on MATLAB in terms of Cross-Correlation with and without error correction code..Das, Subhajit, Reshmi Maity, and N. P. Maity (2018) [9], proposed VLSI-based digital design and implementation of reversible image watermarking (RIW) architecture using difference expansion (DE). Mathematical simplicity of using a set of linear transformations leads to the choice of DE-based technique for developing hardware design. High-level synthesis approach with resource-constraint design makes the architecture novel that needs only single adder, subtractor, multiplier, and divider along with 20 registers and 14 multiplexers for embedding. Its high performance gain in terms of payload capacity and the visual quality of the watermarked images would make this hardware architecture useful for real-time application on security purpose of medical and military images.Joshi, Amit M. (2018) [10], proposed watermarking algorithm is initially simulated by MATLAB to validate the performance. The architecture of proposed system is subsequently implemented on hardware to measure the real-time performance. The hardware performance of proposed algorithm is demonstrated with FPGA prototyping and ASIC Implementation. This method proposed video watermarking with results on MATLAB and a hardware platform. The hardware results of proposed schemes are covered with field programmable gate array (FPGA) and application specific integrated circuit (ASIC) implementation.Jana, Poulami, and Amit Phadikar (2019) [11], proposed hardware architecture of reversible watermarking for medical images and its implementation in field-programmable gate array (FPGA) is proposed. The difference expansion (DE) method for watermarking in lifting-based DWT domain and channel coding is implemented in hardware to achieve the goal. Maity, Goutam Kumar (2019) [12], demonstrated an experiment which is conducted over a variety of benchmark images and the results establish the superiority of the method. Charles, Subodha, Vincent Bindschaedler, and Prabhat Mishra (2022) [13], proposed a lightweight alternative defense based on digital watermarking techniques. We develop theoretical models to provide security guarantees. Experiments using realistic SoC models and diverse applications demonstrate that our approach can significantly outperform state-of-the-art methods.Dwivedi, Ranjana, and Vinay Kumar Srivastava, (2022) [14], proposed geometrically robust digital image watermarking scheme based on Zernike moments and Features from Accelerated Segment Test (FAST) technique is presented in this paper. FAST technique detect the geometrically robust feature points from the image and circular patches centered around these detected features are used to embed the given watermark data sequence.Prabhu, V. (2022) [15], proposed a model using discrete wavelet transform (DWT) to eliminate the attacks on digital security. Semi-fragile watermarking techniques strive towards establishing the confidentiality of data content.

3. Proposed Mythology

For decades, many conventional signal processing methods have been applied in the image fusion field to extract image features, such as discrete wavelet transform (SWT), contour let transform, shift- invariant shear let transform and quaternion wavelet transform etc. Figure 1 shows the proposed block diagram of blind image watermarking. To overcome these problems optimization based fusion schemes are proposed. Grayscale image watermark embedding and extraction using SWT (Stationary Wavelet Transform) is a popular technique for watermarking digital images. It involves decomposing the image into different frequency bands using the SWT algorithm, and then embedding the watermark into one or more of these frequency bands. The embedding process involves the following steps:

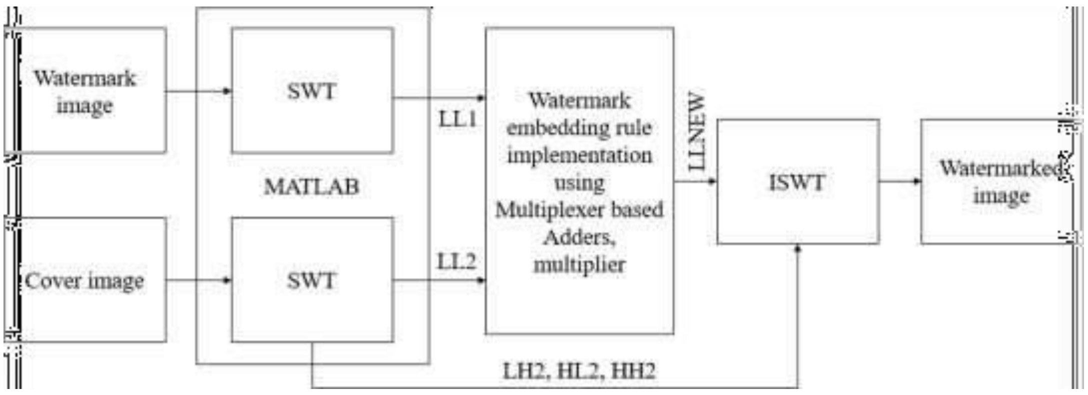


Figure 1. Proposed block diagram.

Step 1: Decomposition: The grayscale image is decomposed into several frequency bands using the SWT algorithm. This results in a set of coefficients representing the image in each frequency band.

Step 2: Selection of coefficients: A subset of coefficients is selected from one or more of the frequency bands for embedding the watermark. The selection of coefficients is based on their sensitivity to changes in the pixel values.

Step 3: Embedding: The selected coefficients are modified to embed the watermark. The modification is performed using a pseudo-random sequence, which is generated using a secret key. The sequence determines the locations and values of the watermark bits that will be embedded in the selected coefficients. Here, the embedding formula is Here, addition, and multiplication is implemented using VLSI based adders and subtractors.

Step 4: Inverse SWT: The modified coefficients are used to reconstruct the watermarked image using the inverse SWT algorithm.

he extraction process involves the following steps:

Step 5: Decomposition: The watermarked image is decomposed into several frequency bands using the SWT algorithm. This results in a set of coefficients representing the image in each frequency band.

Step 6: Selection of coefficients: The same subset of coefficients that were used for embedding the watermark is selected from one or more of the frequency bands.

Step 7: Extraction: The selected coefficients are analyzed to extract the watermark. The extraction process involves applying a correlation operation between the coefficients and the pseudo-random sequence that was used for embedding the watermark. The correlation operation reveals the locations and values of the embedded watermark bits.

Step 8: Verification: The extracted watermark is compared to the original watermark to verify the integrity of the watermark. This is done by computing the correlation coefficient between the extracted and original watermarks.

3.1 SWT

The SWT is a signal processing technique that decomposes a signal into different frequency bands using wavelets. It is like the SWT, but unlike the SWT, the SWT does not down sample the signal at each level of decomposition, which leads to a shift-invariant representation of the signal. Here's a detailed algorithmic analysis of the stationary wavelet transform:

1. Input: A signal x(n) of length N, and a set of wavelet filters h(n) and g(n), where h(n) is the low-pass filter and g(n) is the high-pass filter.

2. Initialize a vector of approximation coefficients V0 = x(n).
3. For each level of decomposition i = 1 to J, do the following:

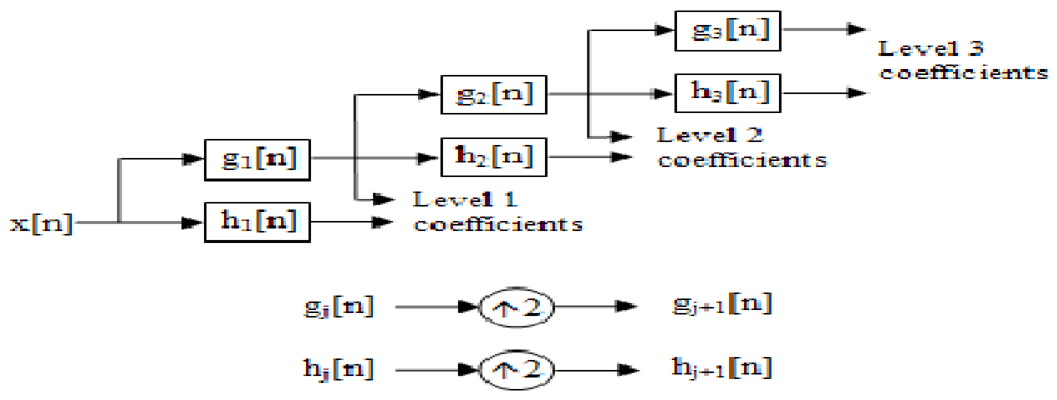


Figure 2. SWT operation

Compute the wavelet coefficients Wj,i(n) and the approximation coefficients Vj,i(n) using the following equations:

$W_{j,i}(n) = (V_{j-1,i} * g(n)) * 2^{(i/2)}$

$V_{j,i}(n) = (V_{j-1,i} * h(n)) * 2^{(i/2)}$

Store the wavelet coefficients Wj,i(n) and the approximation coefficients Vj,i(n) in two separate matrices, W and V, respectively.

output: The wavelet coefficients W and the final approximation coefficients VJ. Let's break down each step-in detail:

Step 1: Input The input to the SWT algorithm is a signal x(n) of length N, and a set of wavelet filters h(n) and g(n), where h(n) is the low-pass filter and g(n) is the high-pass filter.

Step 2: Initialize We start by initializing a vector of approximation coefficients V0, which is just the original signal x(n).

Step 3: Decomposition For each level of decomposition i = 1 to J, we compute the wavelet coefficients and approximation coefficients as follows:

At each level of decomposition, we store the wavelet coefficients Wj,i(n) and the approximation coefficients Vj,i(n) in two separate matrices, W and V, respectively. These matrices will contain the coefficients for each level of decomposition.

Step 4: Output The final output of the SWT algorithm is the wavelet coefficients W and the final approximation coefficients VJ.

The wavelet coefficients represent the high-frequency components of the signal at different scales, while the approximation coefficients represent the low-frequency components of the signal at the final level of decomposition.

3.2 Watermark embedding rule.

The watermark embedding rule is.

LLnew=LL1+alpha. * LL2

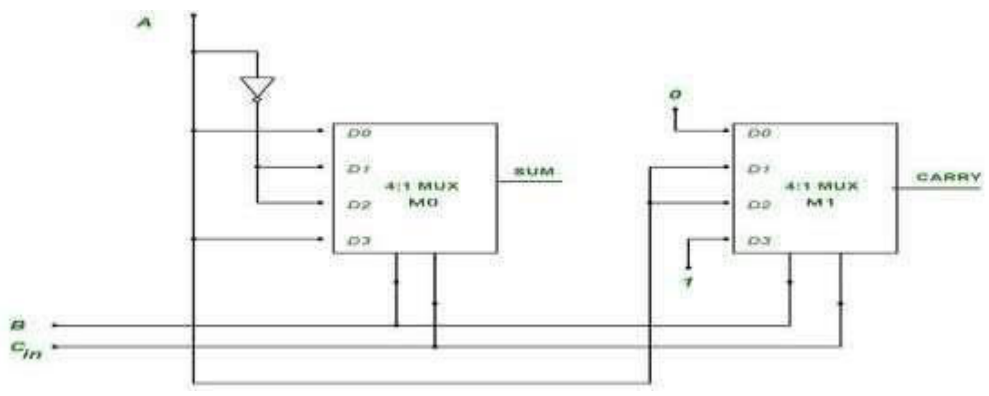


Figure 3. Full adder using Mux.

It is also possible to implement a full adder using only four-to-one muxes. One way to do this is as follows:

Step 1: Use two 4-to-1 muxes to generate the sum output (S):

- Connect A and B to the inputs of one of the muxes.
- Connect Cin and the output of the other mux to the select inputs of the muxes.
- Connect the outputs of the two muxes to the inputs of a third 4-to-1 mux.

- Connect the select inputs of the third mux to the outputs of the first two muxes.
- The output of the third mux is the sum output (S). Step 2: Use another 4-to-1 mux to generate the carry output (Cout):
- Connect the carry input (Cin) and the output of the third mux to two of the inputs of the mux.
- Connect the sum output (S) and the select input of the mux to the other two inputs of the mux.
- The output of the mux is the carry output (Cout).

3.3 Hybrid adder

A hybrid adder is a type of adder that combines multiple adder circuits to improve the speed and efficiency of the addition process. In this analysis, we will discuss the implementation of a hybrid adder using the above full adder that is implemented using an 8-to-1 multiplexer. A hybrid adder can be constructed by dividing the binary number to be added into multiple sections, each of which can be added in parallel. The partial sums generated by each section are then added together using a final adder circuit to obtain the final sum. To implement a hybrid adder using the full adder circuit implemented using an 8-to-1 multiplexer, we can use the following steps:

Step 1: Divide the binary number to be added into multiple sections, each containing n bits.
Let's call them A0, B0, A1, B1, A2, B2, …, An-1, Bn-1.

Step 2: Implement n instances of the full adder circuit using an 8-to-1 multiplexer, one for each pair of bits in the input sections. For example, one full adder circuit can be used to add A0 and B0, another to add A1 and B1, and so on.

Step 3: Connect the carry-out (C-out) output of the first full adder circuit to the carry-in (C-in) input of the second full adder circuit, and so on for all full adder circuits except for the last one.

Step 4: The output of the last full adder circuit will be the final sum of the input sections.

Step 5: If the binary number has an odd number of bits, one extra bit can be added to one of the sections to make the number of bits even, and the final carry-out bit from the last full adder circuit can be added to obtain the final sum.

The hybrid adder can provide faster performance compared to a single full adder circuit by performing addition in parallel. The time taken to obtain the final sum depends on the number of sections and the time taken by the final adder circuit to add the partial sums together. By carefully choosing the number of sections and the type of final adder circuit, the speed and efficiency of the addition process can be optimized.

3.4 Hybrid Multiplier

An array multiplier is a digital combinational circuit used for multiplying two binary numbers by employing an array of full adders and half adders. This array is used for the nearly simultaneous addition of the various product terms involved. To form the various product terms, an array of AND gates is used before the Adder array. Checking the bits of the multiplier one at a time and forming partial products is a sequential operation that requires a sequence of add and shift micro-operations. The multiplication of two binary numbers can be done with one micro-operation by means of a combinational circuit that forms the product bits all at once. This is a fast way of multiplying two numbers since all it takes is the time for the signals to propagate through the gates that form the multiplication array. However, an array multiplier requires many gates, and for this reason it was not economical until the development of integrated circuits. For implementation of array multiplier with a combinational circuit, consider the multiplication of two 2-bit numbers as shown in figure. The multiplicand bits are b1 and b0, the multiplier bits are a1 and a0, and the product is c3c2c1c0. Assuming A = a1a0 and B= b1b0, the various bits of the final product term P can be written as:-

Step 1: P(0)=a0b0

Step 2: P(1)=a1b0+b1a0

Step 3: P(2) = a1b1 + c1 where c1 is the carry generated during the addition for the P(1) term.
Step 4: P(3) = c2 where c2 is the carry generated during the addition for the P(2) term.
For the above multiplication, an array of four AND gates are required to form the various

product terms like a0b0 etc. and then an adder array is required to calculate the sums involving the various product terms and carry combinations mentioned in the above equations to get the final Product bits as shown in Figure 4.

Step 1: The first partial product is formed by multiplying a0 by b1, b0. The multiplication of two bits such as a0 and b0 produces a 1 if both bits are 1; otherwise, it produces 0. This is identical to an AND operation and can be implemented with an AND gate.

Step 1: The first partial product is formed by means of two AND gates.

Step 3: The second partial product is formed by multiplying a1 by b1b0 and is shifted one position to the left.

Step 4: The above two partial products are added with two half-adder (HA) circuits. Usually there are more bits in the partial products, and it will be necessary to use full adders to produce the sum.

Step 5: Note that the least significant bit of the product does not have to go through an adder since it is formed by the output of the first AND gate.

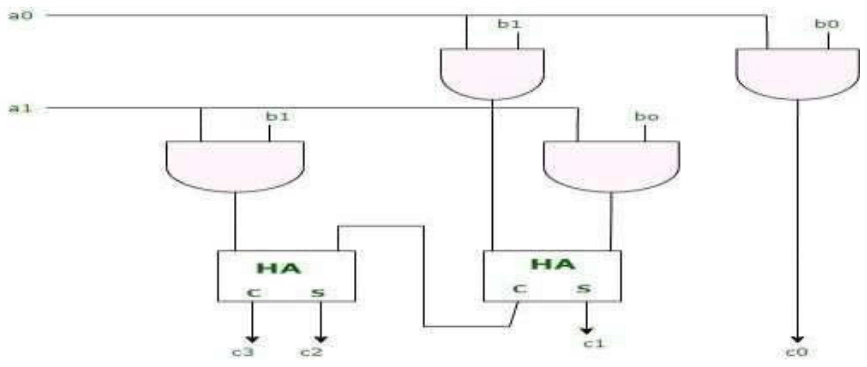


Figure 4. Half adder

A combinational circuit binary multiplier with more bits can be constructed in similar fashion. A bit of the multiplier is ANDed with each bit of the multiplicand in as many levels as there are bits in the multiplier. The binary output in each level of AND gates is added in parallel with the partial product of the previous level to form a new partial product. The last level produces the product. For j multiplier bits and k multiplicand we need j*k AND gates and (j- 1) k-bit adders to produce a product of j+k bits. Array multiplier is the simplest structure of

parallel multiplier. These multipliers using the standard add and shift operation based on 'add and shift' algorithms to perform a multiplication operation. The structure of 4-bit array multiplier is presented in Figure 5. The partial products generator consists of n number of 'AND' gates to multiply the multiplicand with each bit of the multiplier and then these partial products are shifted depending on their order and this summation operation can be performed by using full adder and a half adder. In 4x4 array multiplier, 4x4 AND gates used to generate partial products and 4x (4-2) full adders, and 4 half adders used to generate.

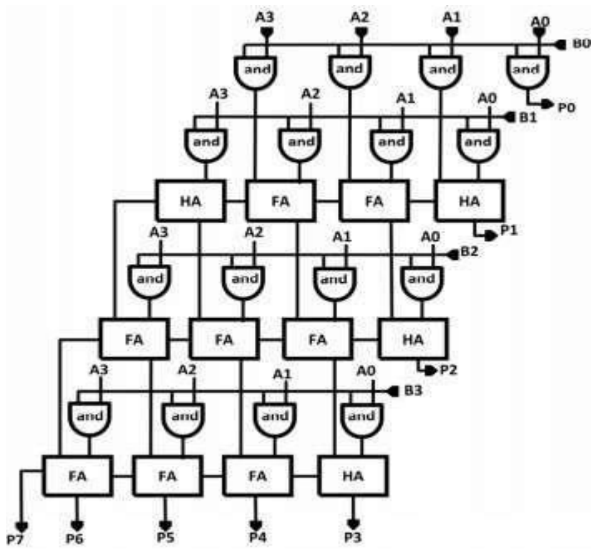
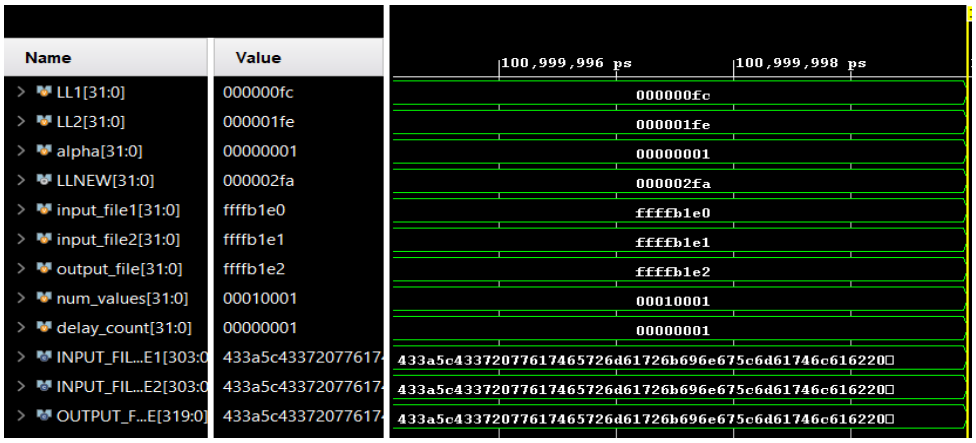


Figure 5. Array multiplier.

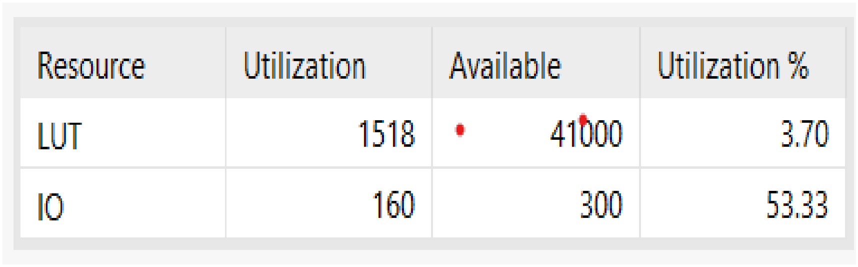
4. Results and Discussion

The simulation results will do by using in Vivado ISE.The timing, power and synthesis reports listed below.



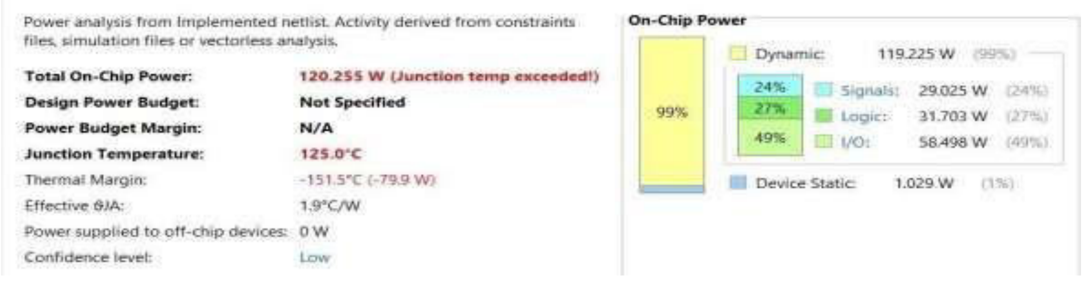
Name	Value	100,999,996 ps – 100,999,998 ps
LL1[31:0]	000000fc	000000fc
LL2[31:0]	000001fe	000001fe
alpha[31:0]	00000001	00000001
LLNEW[31:0]	000002fa	000002fa
input_file1[31:0]	ffffb1e0	ffffb1e0
input_file2[31:0]	ffffb1e1	ffffb1e1
output_file[31:0]	ffffb1e2	ffffb1e2
num_values[31:0]	00010001	00010001
delay_count[31:0]	00000001	00000001
INPUT_FIL...E1[303:0	433a5c43372077617	433a5c43372077617465726d61726b696e675c6d61746c616220
INPUT_FIL...E2[303:0	433a5c43372077617	433a5c43372077617465726d61726b696e675c6d61746c616220
OUTPUT_F...E[319:0]	433a5c43372077617	433a5c43372077617465726d61726b696e675c6d61746c616220

Figure 6. Simulation outcome



Resource	Utilization	Available	Utilization %
LUT	1518	41000	3.70
IO	160	300	53.33

Figure 7. Design summary



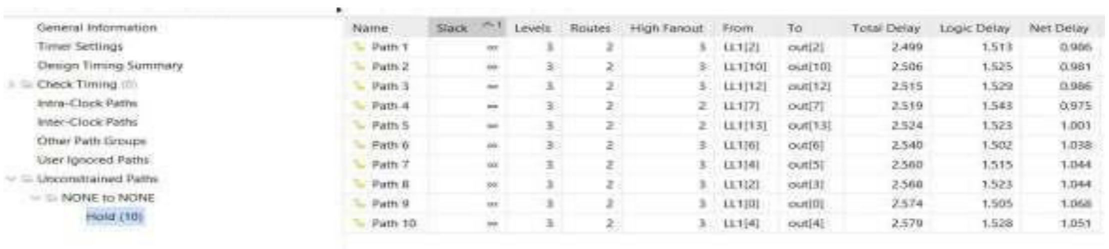
Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	120.255 W (Junction temp exceeded!)
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	125.0°C
Thermal Margin:	-151.5°C (-79.9 W)
Effective θJA:	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

On-Chip Power

🟨	Dynamic:	119.225 W	(99%)
🟦	Signals:	29.025 W	(24%)
🟩	Logic:	31.703 W	(27%)
🟩	I/O:	58.498 W	(49%)
🟦	Device Static:	1.029 W	(1%)

Figure 8. Power summary



General Information
Timer Settings
Design Timing Summary
Check Timing (0)
Intra-Clock Paths
Inter-Clock Paths
Other Path Groups
User Ignored Paths
Unconstrained Paths
NONE to NONE
Hold (10)

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay
Path 1	∞	3	2	3	LL1[2]	out[2]	2.499	1.513	0.986
Path 2	∞	3	2	3	LL1[10]	out[10]	2.506	1.525	0.981
Path 3	∞	3	2	3	LL1[12]	out[12]	2.515	1.529	0.986
Path 4	∞	3	2	2	LL1[7]	out[7]	2.519	1.543	0.975
Path 5	∞	3	2	2	LL1[13]	out[13]	2.524	1.523	1.001
Path 6	∞	3	2	3	LL1[6]	out[6]	2.540	1.502	1.038
Path 7	∞	3	2	3	LL1[4]	out[5]	2.560	1.515	1.044
Path 8	∞	3	2	3	LL1[2]	out[3]	2.568	1.523	1.044
Path 9	∞	3	2	3	LL1[0]	out[0]	2.574	1.505	1.068
Path 10	∞	3	2	3	LL1[4]	out[4]	2.579	1.528	1.051

Figure 9. Time summary.

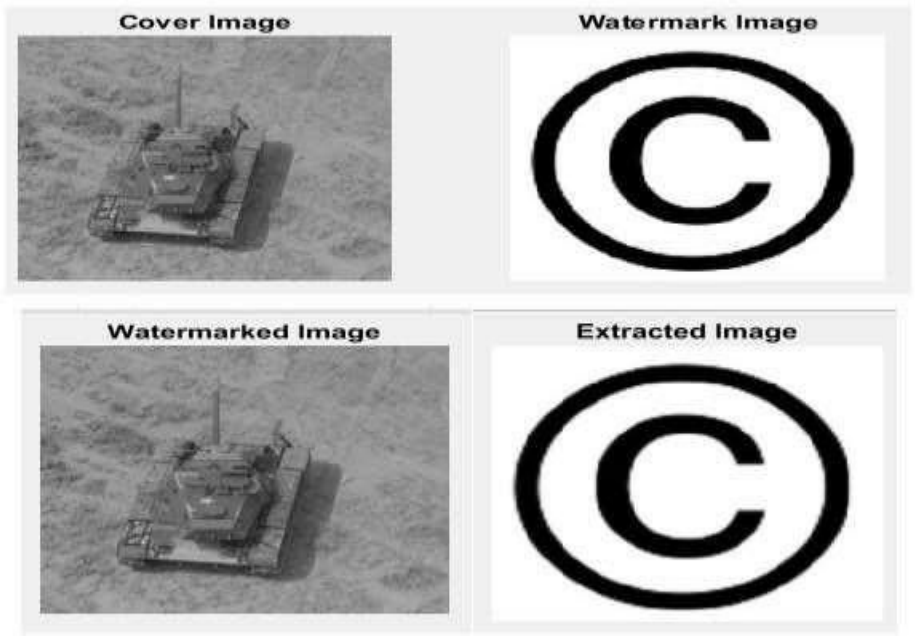


Figure 10. Wateramrking output images.

Table 1. Comparison Table

Parameters	DWT	SWT
Mean Square Error (MSE)	0.01	0.0001
Peak Signal to Noise Power (PSNR)	25 db	37.71 db
Structure Similarity Index Measure (SSIM)	0.97	0.98
Normalized Cross-Correlation (NCC)	0.972	0.999

5. Conclusion

This work implemented the SWT based image watermarking method. Grayscale image watermark embedding and extraction using SWT is a popular technique for watermarking digital images. It involves decomposing the image into different frequency bands using the SWT algorithm, and then embedding the watermark into one or more of these frequency bands. The grayscale image is decomposed into several frequency bands using the SWT algorithm. This results in a set of coefficients representing the image in each frequency band. A subset of coefficients is selected from one or more of the frequency bands for embedding the watermark. The selection of coefficients is based on their sensitivity to changes in the pixel values. The selected coefficients are modified to embed the watermark. The modification is performed using a pseudo-random sequence, which is generated using a secret key. The sequence determines the locations and values of the watermark bits that will be embedded in the selected coefficients. Here, addition, and multiplication is implemented using VLSI based adders and subtractors. The modified coefficients are used to reconstruct the watermarked image using the inverse SWT algorithm.

References

1 Kumar, M. Pradeep, et al. "VLSI implementation of Digital Watermarking Technique for security and authentication of Digital Data." 2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON). IEEE, 2021.

2 Jayanthi, V. E., Senthil Pitchai, and M. Smitha. "Design a hybrid FPGA architecture for visible digital image watermarking in spatial and frequency domain." Journal of Circuits, Systems and Computers 31.01 (2022): 2250020.

3 Sakthivel, S. M., and A. Ravi Sankar. "Computation-efficient image watermarking
architecture with improved performance." Computers & Electrical Engineering 84 (2020): 106649.

4 Penchalaiah, Usthulamuri, and VG Siva Kumar. "Design and Implementation of Low Power and Area Efficient Architecture for High Performance ALU." Parallel Processing Letters 32.01n02 (2022): 2150017.

5 S. V. G. Kumar, M. Vadivel, U. Penchalaiah, P. Ganesan and T. Somassoundaram, "Real Time Embedded System for Automobile Automation," 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), Pondicherry, India, 2019, pp. 1-6, doi: 10.1109/ICSCAN.2019.8878820.

6 Nagaraj, P., et al. "VLSI implementation of image compression using TSA optimized discrete wavelet transform techniques." 2020 Third International Conference on Smart
Systems and Inventive Technology (ICSSIT). IEEE, 2020.

7 Joshi, Amit M., Samar Ansari, and M. Ravikumar. "VLSI architecture of integer DCT based watermarking with error correction capability." TENCON 2017-2017 IEEE Region 10 Conference. IEEE, 2017.

8 Das, Subhajit, Reshmi Maity, and N. P. Maity. "VLSI-based pipeline architecture for reversible image watermarking by difference expansion with high-level synthesis approach." Circuits, Systems, and Signal processing 37 (2018): 1575-1593.

9 Joshi, Amit M. "VLSI implementation of video watermarking for secure HEVC coding standard." Cryptographic and Information Security. CRC Press, 2018. 353- 377.

10 Jana, Poulami, and Amit Phadikar. "Low-power FPGA-based hardware implementation of reversible watermarking scheme for medical image." Computational Advancement in Communication Circuits and Systems, 2019.

11 Maity, Goutam Kumar, et al. "Power-aware VLSI design of reversible watermarking for access control." Microsystem Technologies, 2019.

12 Charles, Subodha, Vincent Bindschaedler, and Prabhat Mishra. "Digital watermarking for detecting malicious intellectual property cores in noc architectures, 2022.

13 Dwivedi, Ranjana, and Vinay Kumar Srivastava. "Geometrically Robust Digital Image Watermarking Based on Zernike Moments and FAST Technique." Advances in VLSI, Communication, and Signal Processing, (2022).