International Journal of Innovative Research and Scientific Studies, 8(3) 2025, pages: 1793-1810

IJIRSS
International Journal of Innovative Research & Scientific Studies

ISSN: 2617-6548

URL: www.ijirss.com

An innovative LSB image steganography system for multi-image concealment using hardware-software co-design

Ahmad F. Fathil[1*], Harith G. Ayoub[2], Zaid A. Abdulrazzaq[3]

[1,2]*Technical Management institute, Northern Technical University (NTU), Nineveh, Iraq.*
[3]*Department of computer and AI techniques engineering, Northern Technical University (NTU), Nineveh, Iraq.*

Corresponding author: Ahmad F. Fathil (*Email: ahmed.fehr22@ntu.edu.iq*)

Abstract

This paper presents an accelerated image steganography technique utilizing Least Significant Bit (LSB) algorithms implemented on FPGA hardware to conceal and retrieve three hidden images within an RGB cover image. The embedding process uses an XOR operation between the secret bit and the cover pixel's least significant bit (LSB): if the result is zero, the bits are identical, and no change is needed; if the result is one, the secret bit replaces the LSB of the cover pixel. Two system designs were developed to evaluate acceleration: a hardware implementation on a Xilinx ZYNQ-7020 FPGA using XSG programming and a software implementation coded in MATLAB, running on a Core i7-10750H CPU @ 2.60 GHz with 8 GB RAM. The proposed system was evaluated using various performance metrics, including histogram analysis, PSNR, MSE, BER, SSIM, CCR, execution time, operating frequency, and throughput. Experimental results showed clear, accurate image retrieval with significant acceleration: the software execution time was 0.153 seconds, while the FPGA hardware achieved 23.405 microseconds, yielding a speedup factor of approximately 6537×.

Keywords: FPGA, LSB, Multiple images, Steganography.

DOI: 10.53894/ijirss.v8i3.6887
Funding: This study received no specific financial support.
History: Received: 17 March 2025 / **Revised:** 21 April 2025 / **Accepted:** 25 April 2025 / **Published:** 9 May 2025

Competing Interests: The authors declare that they have no competing interests.
Authors' Contributions: All authors contributed equally to the conception and design of the study. All authors have read and agreed to the published version of the manuscript.
Transparency: The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.
Publisher: Innovative Research Publishing

1. Introduction

Digital images are used in many fields and play a role in the transmission of data over public open networks, which can be easily attacked, and this is the main issue [1, 2]. Nowadays, there is a need to safeguard data from falling into the wrong hands. Various technologies are available to address this issue, one of which is steganography [3-5]. The proposed method for concealing text within images (for confidential information) utilizes the least significant bits (LSB) algorithm.

Image steganography has a rich historical background and holds significant importance in the field of data communication. Subramanian [6] proposed an image steganography method using a deep neural network with a long training time. In comparison with this research, MSE and PSNR were not as good as this research. Mandal et al. [3] provide a survey for multiple methods used for steganography; all of the methods have a high computational time, which acts as the main issue in real-time steganography applications. Halboos and Albakry [7] proposed a good steganographic method for text hiding in the cover image and achieved a large PSNR, but without dealing with the time execution point. Abuhmaidan et al. [8] proposed the bit Pattern Steganography method, combining steganography with cryptography for more robustness but without dealing with time execution or cover image MSE or PSNR. Hameed et al. [9] provide a review of multiple research directions in steganography with multiple types such as DNA, network, audio, video, text, image, etc. AbdelRaouf [10] proposed a new steganographic approach using human visual properties with an improved LSB algorithm with improved hiding performance for MSE, PSNR, and SSIM, but the researchers didn't deal with the time of hiding metric for real-time needs. Rahman et al. [4] presented a comprehensive study of different steganography approaches, but overall did not deal with processing time in hardware as this research had. Bandyopadhyay [11] identifies the methods of steganography with MSE, BER, and PSNR metrics without dealing with the processing time factor. Idakwo, et al. [12]; Evsutin, et al. [13] and Zhang, et al. [14] provided a survey of digital image steganography using multiple methods in in the spatial domain or transform domain. Sudha [15] proposed a text steganography method using chaotic maps with good PSNR without BER, MSE, or processing time performance metrics [3, 4, 6-15].

Suhail and Ayoub [16] proposed LSB wave file steganography that depends on a secret data distribution with circular key values between transmitter and receiver, without discussing performance metrics such as MSE, PSNR, and SSIM, also dealing with processing time requirements [16].

However, a few studies have been made about FPGA acceleration, but that work has a huge benefit: first speed, FPGA has effective performance with huge multimedia data such as audio, image, and video [17-19]. The second real-time application has critical timing demands for avoiding latency. Third parallel capability with low power consumption [20-29]. Fourth flexibility: there is no need for hardware redesign. Fifth algorithm testing and optimizing with prototyping. Sixth integration with other systems such as cameras, sensors, and communication devices [30-39].

In this study, an acceleration was introduced by utilizing FPGA hardware, resulting in huge acceleration of embedding and extraction processes. A multiple hardware architecture for three images was proposed. The algorithm and architecture were verified through hardware implementation on the Xilinx ZYNQ 7000 device using VIVADO. The design focused on a 24-bit RGB image. The parallel hardware architecture resulted in an effective hiding/extraction speed with the FPGA-based Xilinx system generator embedding tool. Experimental results show that the authors' proposed design could greatly accelerate the LSB algorithm for multiple images hiding and provide satisfactory operation accuracy.

The paper is organized as follows: Section one presents the introduction, section two presents preliminary concepts, section three presents the proposed methodology, section four presents results and discussion, and section five presents the conclusion and future work.

1.1. Importance of LSB Steganography for Information Hiding

LSB (Least Significant Bit) steganography is a widely used technique in information hiding due to its simplicity, effectiveness, and minimal perceptual impact. Its importance lies in the following aspects:

Data Concealment: It embeds hidden information in the least significant bits of digital media (e.g., images, audio, or video), which are often imperceptible to human senses.

Low Distortion: The changes made to the host media are minimal, ensuring that the cover medium remains visually or aurally indistinguishable from the original.

Efficiency: LSB steganography requires low computational resources, making it suitable for real-time and embedded systems.

1.2. Applications

Covert Communication: For transmitting secret messages without detection.

Digital Watermarking: For copyright protection by embedding ownership information in media.

Secure Authentication: Embedding identifiers in images for secure identity verification.

Importance of LSB Acceleration in Hardware

LSB acceleration in hardware refers to optimizing the processing of LSB-based operations in hardware circuits or platforms like FPGAs, GPUs, or ASICs. Its significance includes:

1.3. The Importance of Timing Calculation

Hardware acceleration improves the speed of LSB embedding and extraction processes, crucial for high-throughput applications like video streaming or real-time communication.

1.4. Energy Efficiency

Optimized hardware implementations consume less power than software-based solutions, which is vital for mobile and embedded systems.

1.5. Security Enhancements

Hardware implementations can integrate additional security layers, such as on-the-fly encryption, making LSB steganography more robust against attacks.

1.6. Scalability

Hardware designs can handle larger datasets or higher resolutions efficiently, enabling the use of LSB techniques in modern high-definition media applications.

1.7. Specialized Applications

Medical Imaging: Embedding metadata like patient details in diagnostic images without altering their quality.

Military and IoT Systems: Securely hiding data in sensor outputs or communication signals for confidentiality.

In summary, while LSB steganography is a cornerstone of information hiding, hardware acceleration of LSB operations amplifies its utility in scenarios requiring high performance, scalability, and security.

1.8. The Main Contribution of this Work Involves

1- Implement the LSB stenography algorithm (Embedding/extraction) in a MATLAB environment for three grayscale secret images and one cover RGB image, then calculate the execution time in software.
2- Build the architecture of the LSB algorithm with a new embedding technique in terms of declaring the hardware operations employed to get the results.
3- Implement the new LSB architecture in the hardware module using the ZYNQ702 evolution board through the VIVADO software tool with high speed and low silicon area, and then evaluate the execution time in the FPGA.
4- Calculating the speed-up ratio between software and hardware to consider the effectiveness of the hardware performance.

2. Preliminary Concepts

2.1. Steganography

Steganography is a technique used for secure communication used for storage of trust information, protecting the change of information and multimedia systems. Steganography is also used for watermarking, which is defined as the operation of protecting ownership of any media, such as film, art, etc. Multiple types in steganography include Least Significant Bit (LSB), Spread Spectrum, Pixel Value Differencing (PVD), and statistical techniques.

Table 1.
Comparison between LSB Methods.

Technique	Robustness	Imperceptibility	Payload capacity	Complexity
LSB	Low	High	High	Low
PVD	High	Medium	Low	Low
Spread spectrum	Medium	Low	Low	Medium
Statistical	Medium	High	Low	Medium

Table 1. The Show LSB algorithm is the best way to determine the robustness point, which can be adjusted by the acceleration factor. The LSB method has a good peak signal-to-noise ratio, a structural similarity index, is easy for hardware implementation, is fast in execution, and is suitable for large amounts of data hiding [40-43].

2.2. Steganography with LSB

The algorithm as in Figure 1, involves hiding all the bits of secret bits in the LSB bit of the cover image since the LSB bit is a small value and cannot affect the image resolution, and it consists of.

1- Embedding system by the sender: exchange the LSBs of the cover image with secret bits of secret data and obtain the Stego image. Embedding system by the sender: exchange the LSBs of the cover image with secret bits of secret data and obtain the Stego image.
2- Extraction algorithm by the receiver: extract the LSB of the secret image and get back the secret data.

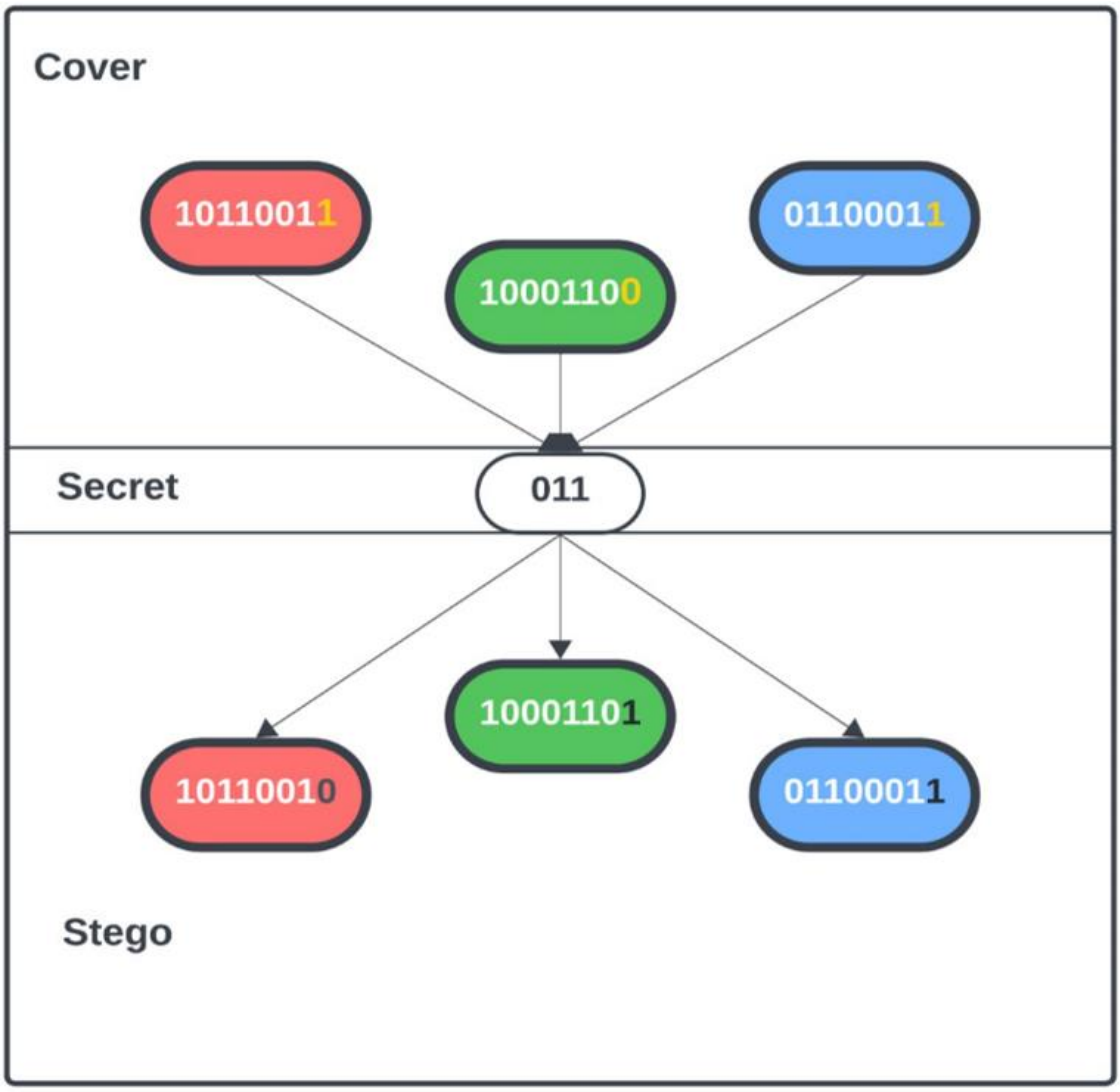

Figure 1.
Stego image using LSB hiding.

2.3. ZYNQ FPGA Implementation using Xilinx System Generator

The Xilinx Vivado Design Suite and the Xilinx System Generator for DSP and HDL work together flawlessly. The most recent nodes are supported by the Xilinx Vivado Design Suite, which is designed for complex designs [44-53]. It integrates hardware and software support and provides system- and device-level design. It comes with toolkits for managing FPGA configuration, synthesis, place, and route. Additionally integrated are specialized tools such as hardware co-simulation, partial reconfiguration, and hardware design languages such as System Verilog, VHDL, or Verilog [54]. Figure 2 shows the design methodology using Xilinx System Generator (XSG). The hardware in this paper is implemented using the VIVADO/XSG environment, VIVADO 2020.2, associated with MATLAB/SIMULINK 2020. They are widely used in various domains such as signal processing, image processing, telecommunications, and more [55-65].

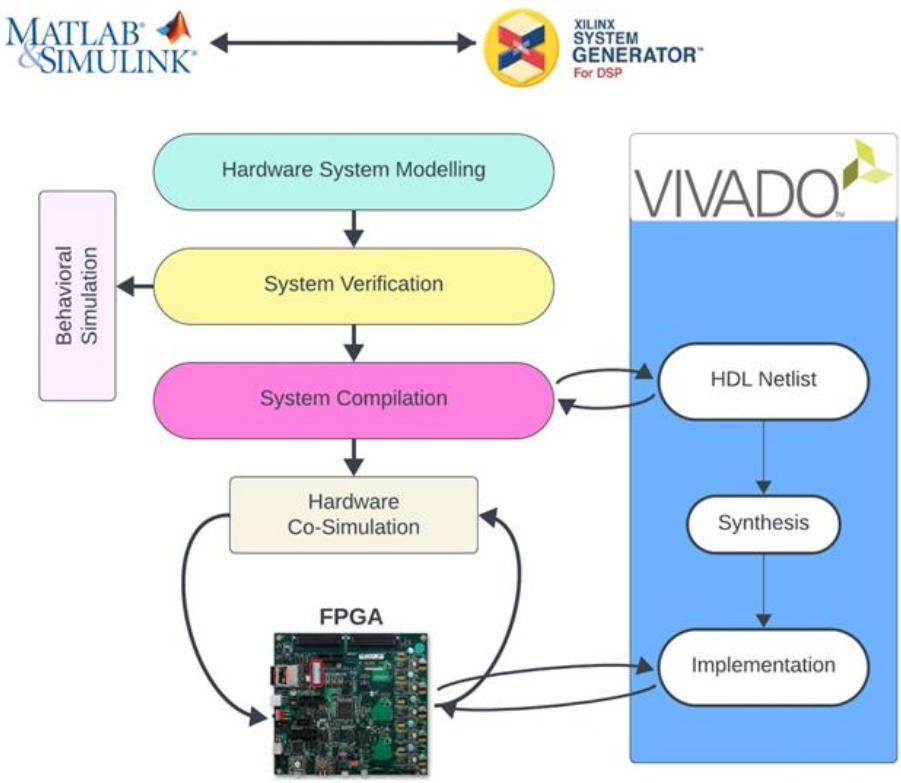

Figure 2.
XSG design flow.

3. Proposed Methodology

The proposed work consists of two parts: software (using MATLAB) and hardware (using FPGA); each part is discussed independently.

3.1. Software

This study was conducted using a Core (TM) i7-10750H CPU @ 2.60 GHz , 2.59 GHz processor with 8 GB RAM hardware. Implementation of the LSB algorithm for three secret images and a cover RGB image involves the implementation of two systems: firstly, the proposed embedding method involved XORing secret bit with least significant bit of cover pixel, if the result was zero logic that means that the two bits are the same, there no need of exchange them else if the result was one logic that means the two bits are different, then the secret bits should take the place of the LSB of cover pixel image. The resulting image of the cover image after replacement is called the Stego image shown in Figure 3. Secondly, the receiver system extracts the image, which begins with converting the Stego image to binary, then takes each LSB bit of each 8 pixels to form one pixel of the secret image. After accumulating all pixels, it can be noticed that the secret image was backed the same as entered into the embedding system shown in Figure 4.

The processing time of the algorithm is measured using a stopwatch timer in MATLAB with the tic command to start the operation of the timer and toc for end. The number of iterations counting time was 50 to be convergent to the most effective value. The researcher calculated the average of these values; the calculating time was equal to 0.153 sec.

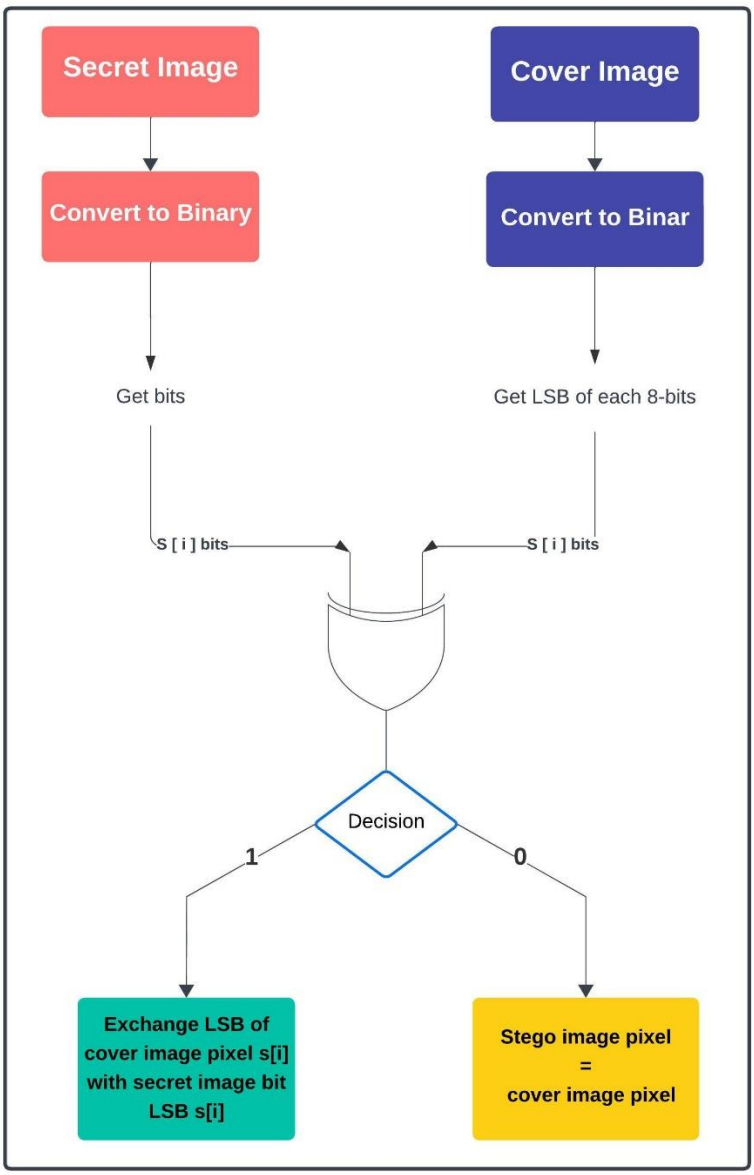

Figure 3.
Embedding LSB.

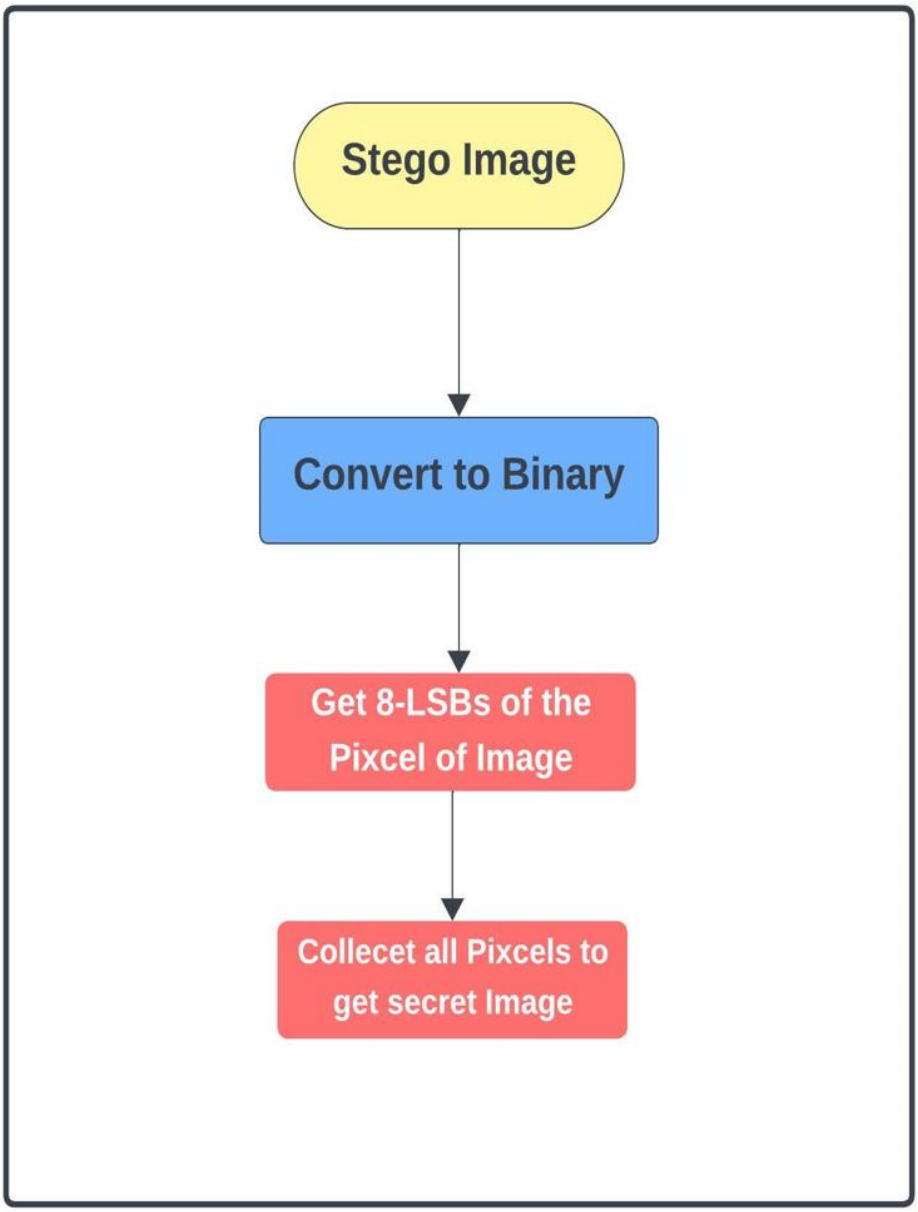

Figure 4.
Extraction LSB.

3.2. Hardware

This part presents a detailed explanation of the hardware system for LSB steganography for three secret images and one RGB cover image, shown in Figure 5, using XSG embedded with Simulink. For testing the input images, the RGB cover image was “peppers.png” with three secret images. Firstly was "cameraman.tif,”) Secondly was "moon.tif,”) and lastly was "coins.png.”).

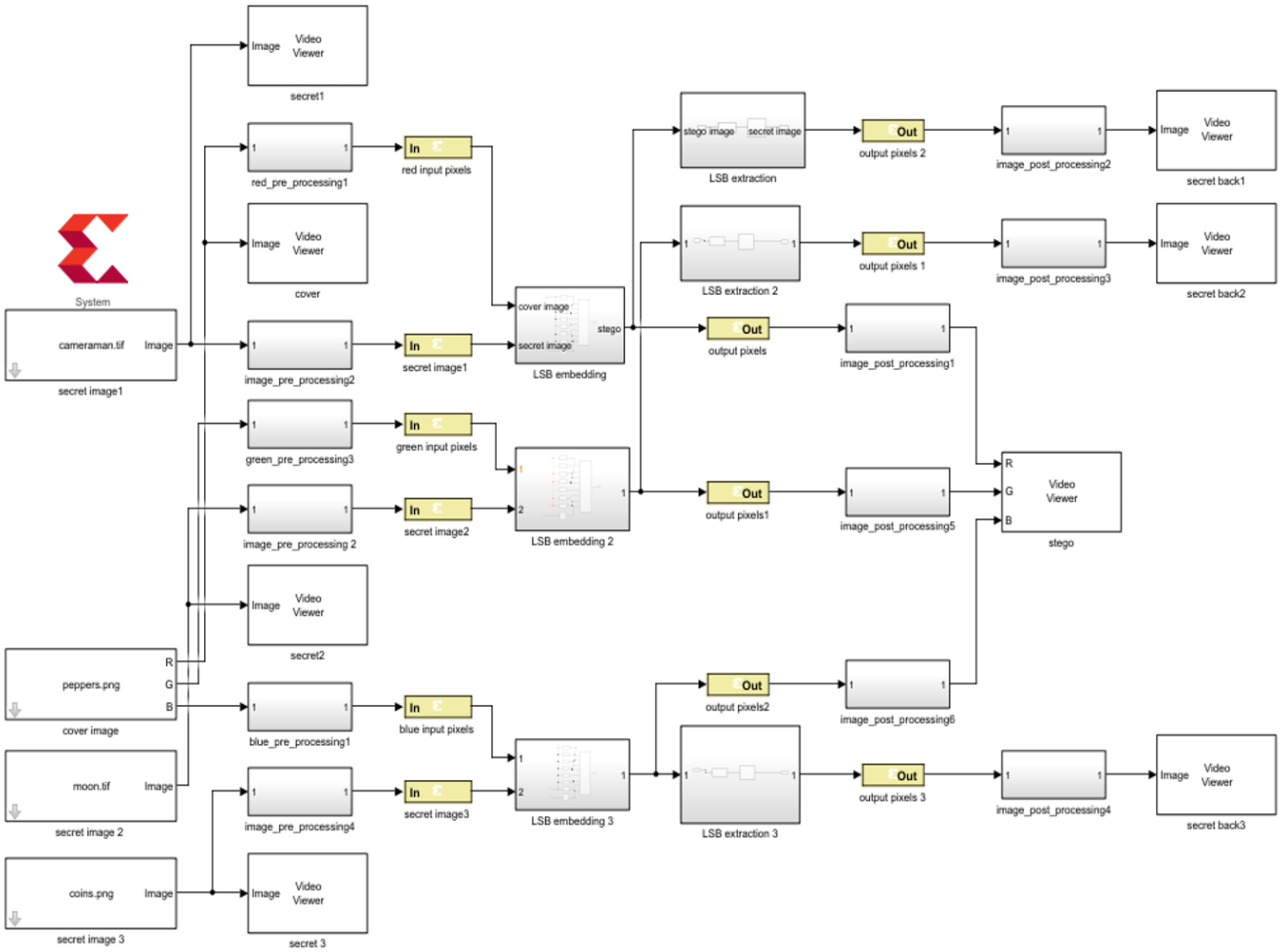

Figure 5.
Overall hardware system of three secret images with one RGB cover image.

Then in the next step, the images need to be entered pixel by pixel into the FPGA environment. That was the role of the image preprocessing subsystem shown in Figure 6, which consists of two parts: First, convert the 2-D array to 1-D array, then convert the 1-D array to scalar values.

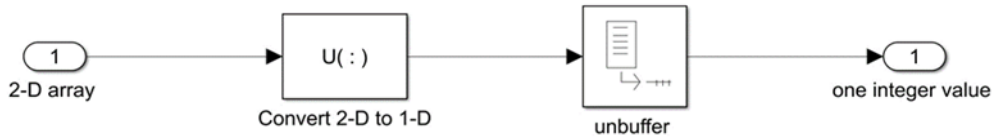

Figure 6.
Image pre-processing subsystem.

The XSG yellow blocks are called gateway-in and gateway-out. the first block used to enter the pixels from the SIMULINK environment to the FPGA environment, the second block did the inverse thing, outing the pixel from FPGA to the SIMULINK to be ready for showing using the SIMULINK block for testing, verifying, and checking the performance of the FPGA design. The pixels output from the FPGA to SIMULINK using the image postprocessing subsystem shown in Figure 7 consist of three parts: first, a buffer to convert the scalar values to one array buffer, then a reshape to convert one array as 2-d as the image size in MATLAB, and the uint8 to convert the data types to 8-bit unsigned pixels for the resulting image.

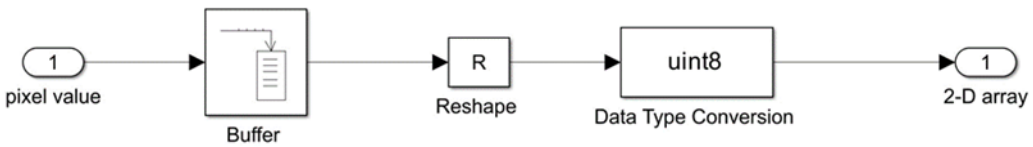

Figure 7.
Image post-processing subsystem.

The rest are three LSB embedding systems and three LSB extraction subsystems, as shown in Figures 8 and 9. Each embedding system, as shown in Figure 10, has two inputs: a pixel of the cover image and a bit of the secret image. The operation will be done with a slice block embedded in XSG to get all the bits of the cover image, except the LSB, which will be replaced by the bit coming from the secret image, then concatenating them together. The output will be a Stego image.

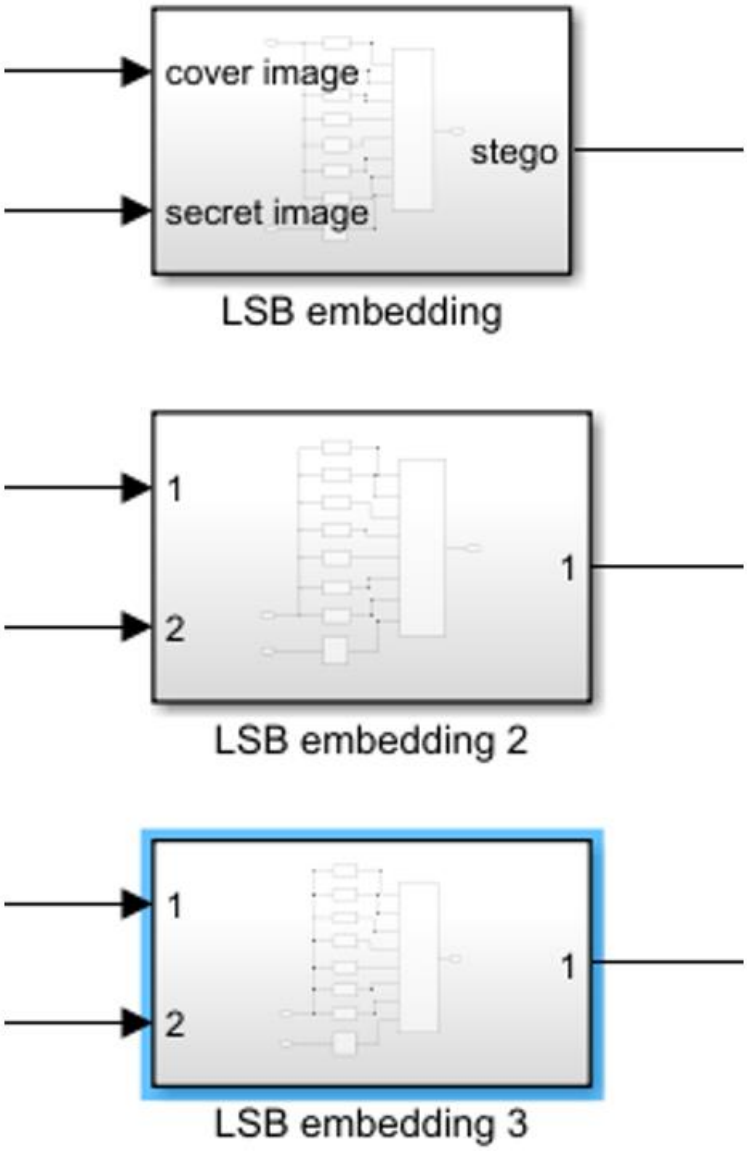

Figure 8.
XSG based LSB embedding subsystem for three secret images hiding.

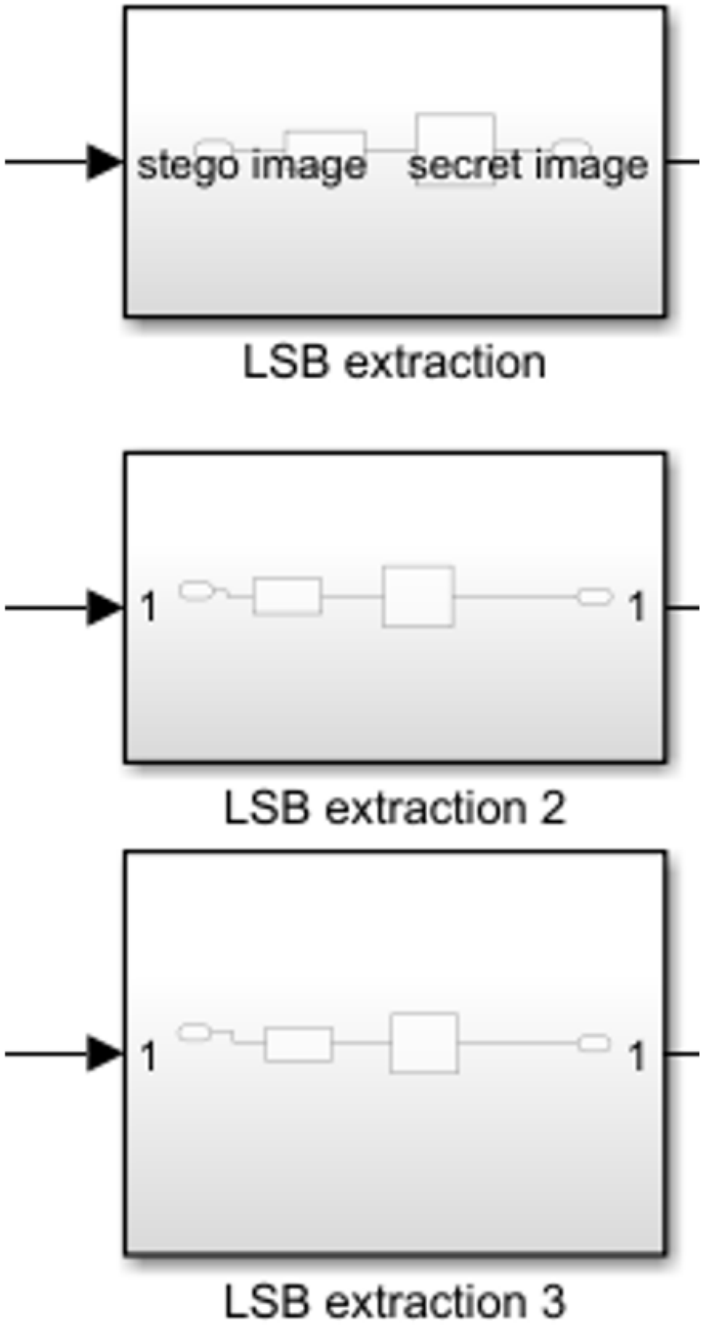

Figure 9.
XSG based LSB extraction subsystem for three secret images hiding.

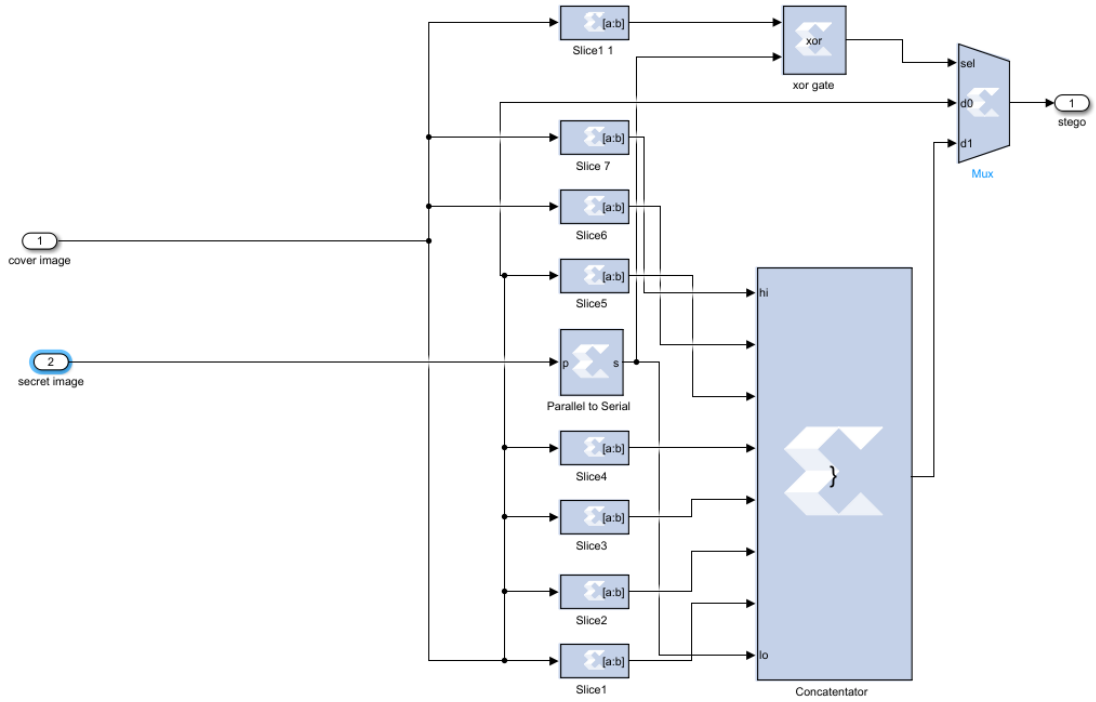

Figure 10.
Hardware design of the embedding subsystem for LSB algorithm.

Each extraction subsystem, as shown in Figure 11, consists of two XSG blocks: the first slice, which is used to get the LSB of Stego that holds the bits of the secret image, and then converting the serial to 8 parallel pixel bits using the serial to parallel block provided by the XSG/SIMULINK environment.

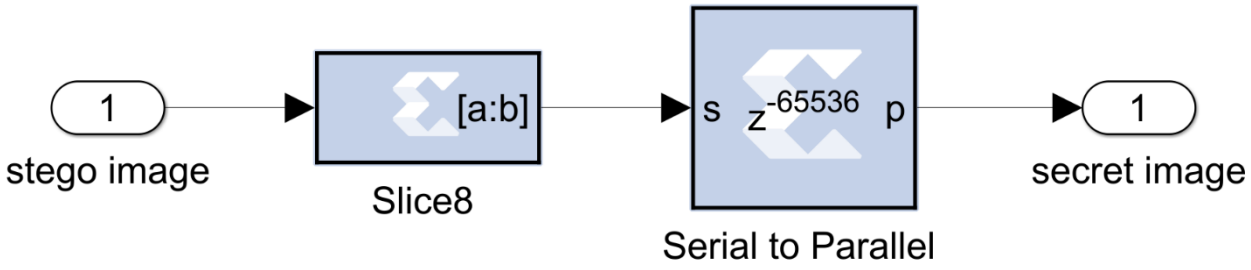

Figure 11.
Hardware design of extracting subsystem for LSB algorithm.

4. Results and Discussion

4.1. Hardware Synthesis, Implementation and Analysis

This step involves compilation of the proposed design to FPGA hardware JTAC co-simulation using the XILINX System Generator/VIVADO tool. The maximum frequency operation of the ZYNQ702 evaluation board is 667 MHZ. When synchronization step over the FPGA, the maximum frequency clock by XSG was 2.8 GHZ, which leads to the throughput of 22.4 Gbps and a processing time for hiding all image operations of 23.405 µs. The acceleration speeds up in the following equations: *Worse negative slack (WNS) =2.651 ns, T=3 ns* from the compilation report of XSG.

$$fmax = 1/(T - WNS) = 2.8 \text{ GHZ} \quad (1)$$

$$harware\ time\ for\ all\ image\ = \frac{256\times256}{fmax} = 23.405\ \mu s \quad (2)$$

$$speed\ up = \frac{software\ time\ (MATLAB)}{hardware\ time\ (FPGA)} = \frac{0.153}{23.405\times10^{-6}} = 6537 \quad (3)$$

Table 2 presents the ZYNQ702 FPGA device summary, which consists of the following:

1- LUT: Represent utilization of Look-Up Tables.
2- LUTRAM: Represent utilization of Look-Up Table RAMS.
3- FF: Represent utilization of Flip-Flops.
4- BRAM: Represent utilization of block RAMs.
5- DSP: Represent utilization of digital signal processing blocks.
6- IO: Represent utilization of input/output buffers.

Table 2.
ZYNQ702 evaluation board summary.

Resource	Utilization	Available	Utilization (%)
LUT	49350	53200	92.7
LUTRAM	15889	17400	91.32
FF	5043	106400	4.7
BRAM	6	140	4.2%
IO	6	200	3
BUFG	5	32	15.6
MMCM	1	4	25

Power report one of the synthesis tools provided by the VIVADO software. Dynamic power consumption was 0.132 W, which is the active power during switching during charging and discharging of the capacitances of the design; static power consumption was 0.105 W, which is the power consumed when the design is also not active, depending on multiple factors: temperature, supply voltage, etc. So, the total was 0.237 W, the overall power report in Figure 11.

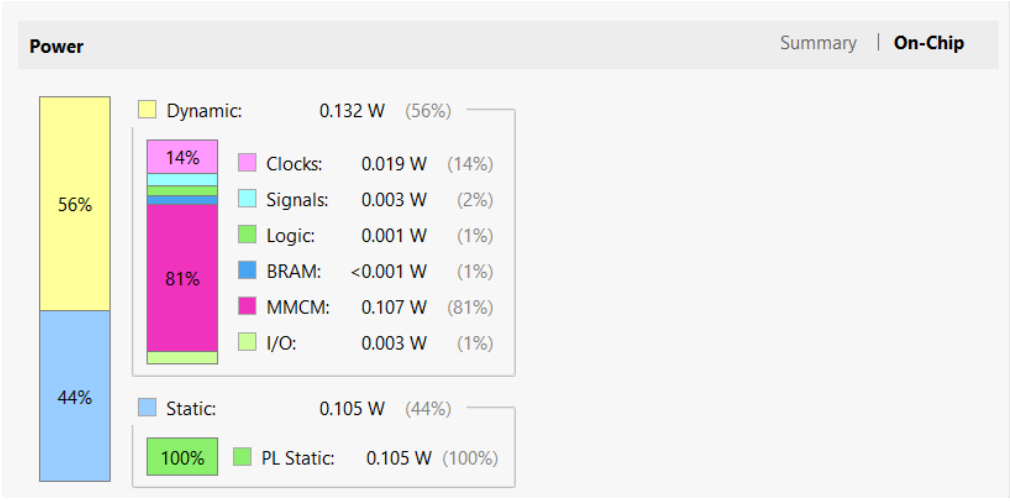

Figure 12.
Power dissipation on ZYNQ702 FPGA device.

Register transfer level schematic (RTL) is the final physical implementation of design, providing more details about the hardware implementation. VIVADO environment for the proposed design contained 10 cells, 2 I/O ports, and 213 nets shown in Figure 12. It can be noticed that complex hardware is required according to the requirements of the proposed algorithm.

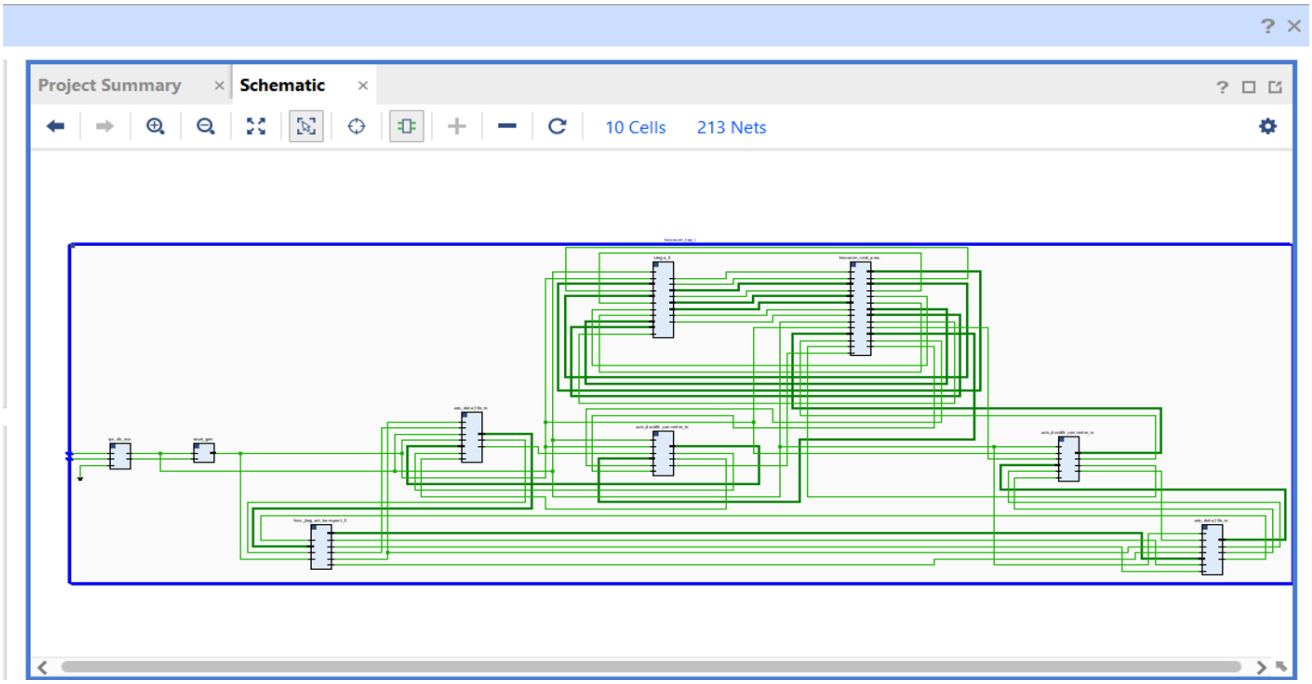

Figure 13.
RTL view of the proposed design.

VIVADO timeline expressed in Figure 13. show the pixel values for red, green, blue cover image with secret images 1,2,3, embedded/XSG enable designers to enter all pixel values of all images in parallel, then process them simultaneously on the same clock faster than software, which deals with each image on multiple clock cycles. It can be observed that the difference between red cover image pixels and red Stego image pixels values of one or 0 means that the LSB of the secret image is exchanged by it.

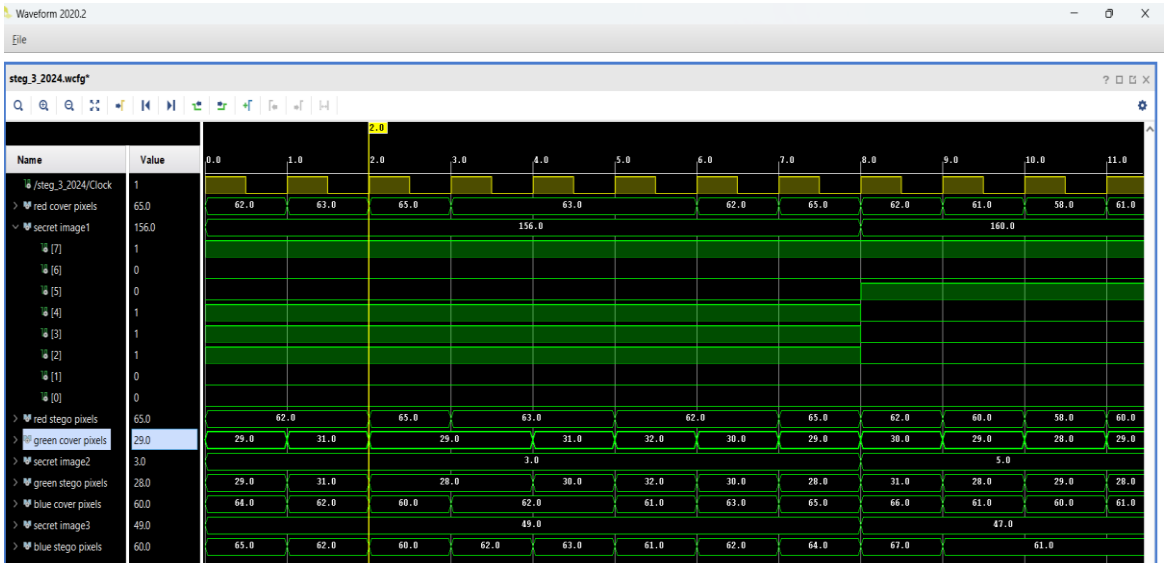

Figure 14.
VIVADO time line cover-secret image pixels.

Hardware co-simulation as shown in Figure 14, represents the final stage of FPGA design, which combines the design model with the ZYNQ702 evaluation board and presents one secret image (“cameraman.tif”) and a cover image (“peppers.png”) with the ZYNQ702 FPGA board.

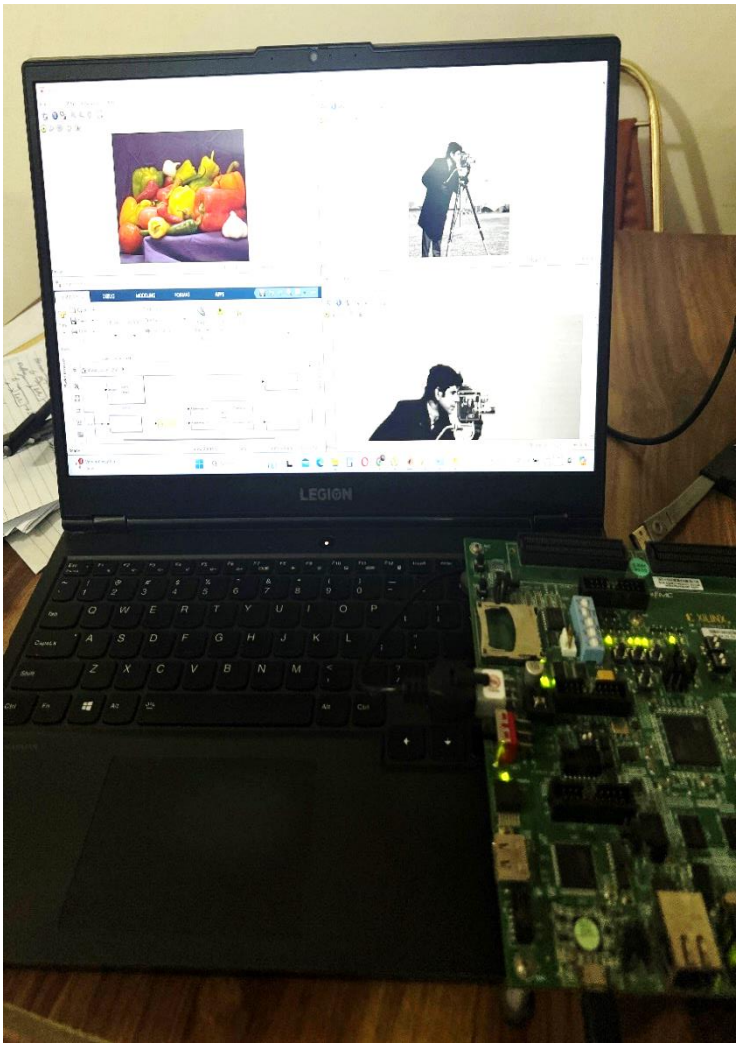

Figure 15.
Hardware co-simulation of the proposed system using ZYNQ702 FPGA.

4.2. Performance Analysis

A number of statistical tests have been run to assess whether the impact of data masking on image quality is within the permitted bounds. Several metrics, including histogram, Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), Cross-Correlation (CCR) and structural similarity.

4.3. Histogram

A graphical illustration conveys the movement of pixels visually by planning the number of pixels at each grayscale level. The cover picture's statistical features were shown to be unaffected by modifying some coefficients, as seen by the histogram of the cover image and the stego image. Therefore, if the histogram of the cover is almost identical to the histogram of the stego-image presented in Figure 16.

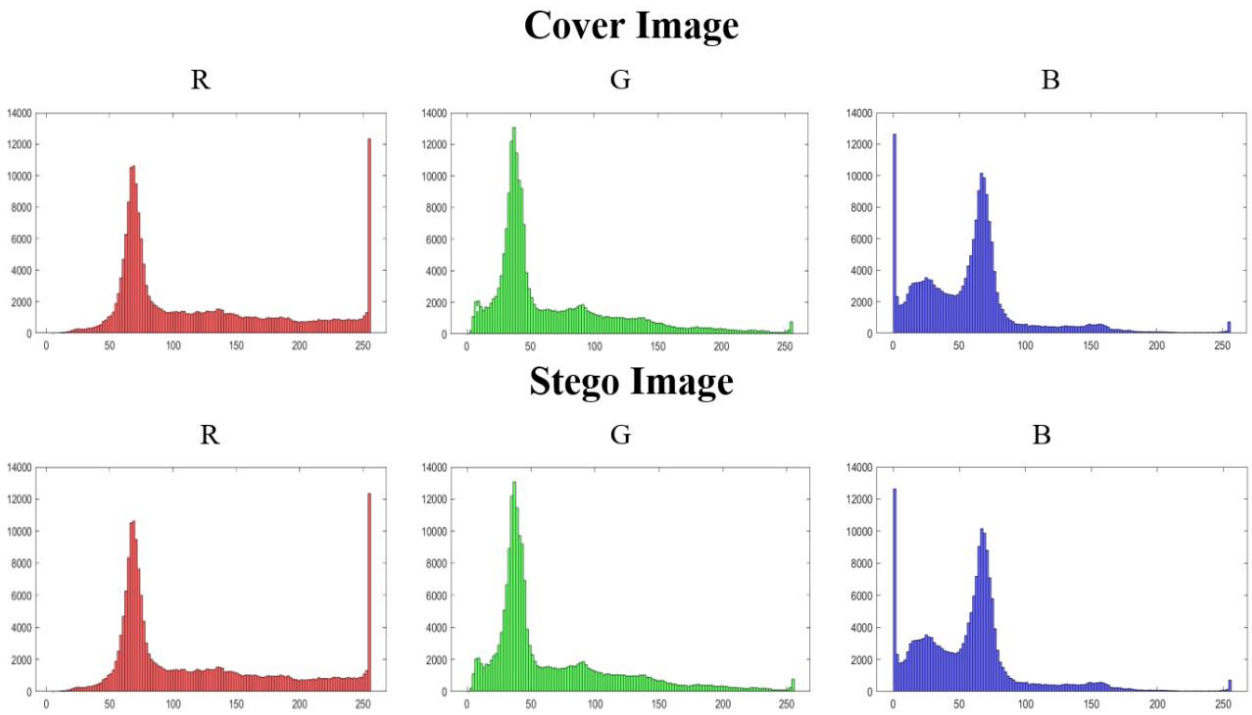

Figure 16.
Histogram Analysis.

4.4. Mean Square Error (MSE)

One of the statistical techniques used to assess the degree of similarity between the stego image and the original image is MSE. When comparing two signals, the error signal is measured after the reference signal has been subtracted, and the mean energy of the error signal is then calculated from Equation 4.

$$MSE = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} (I_{original}(i,j) - I_{encrypted}(i,j))^2 \quad (4)$$

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

It is characterized as the ratio of a signal's maximal power to corrupting noise power. Typically, PSNR is measured using a decibel scale. The PSNR is frequently used to assess how well an image may be recreated. In this instance, the original information is the signal, while the inserted error is the noise. The quality of the image is indicated by the PSNR value.

$$PSNR = 10\ log_{10}\left(\frac{MAX^2}{MSE}\right) \quad (5)$$

4.6. Correlation

Cross-correlation is a mathematical operation commonly used to measure the similarity between two images. In the context of comparing a stego-image with an original image, cross-correlation can be used to identify regions or patterns where the images are similar or dissimilar. The process involves sliding one image over another and computing a measure of similarity at each position.

$$r = n\sum XY - \sum X \sum Y \,(n\sum X^2 - (\sum X)^2) \cdot (n\sum Y^2 - (\sum Y)^2) \quad (6)$$

4.7. Bit Error Rate (BER)

Essential metric for image hiding process for security approaches; it represents the number of bit positions changed in the Stego image. When the value is near 1, it means more bit errors; otherwise, less error bits.

$$BER = Be/Br \quad (7)$$

4.8. Structural Similarity

Quality measurement of image used to measure the similarity index between images. When the value near one means more similarity between images others was means less. Table 3 shows the overall performance hiding metrics of the proposed hardware LSB algorithm.

Table 3.
Performance measurements.

Performance metrics	Proposed work
PSNR	51.1401
MSE	0.500
BER	0.095
SSIM	0.9993
CCR	0.99991
Overall Execution time	23.405 µs
Frequency	2.8 GHZ
Throughput	22.4 Gbps

5. Limitations of the Study

a. Hardware Dependency: The FPGA-based acceleration offered significant speed improvements, but the implementation depends on our hardware (Xilinx ZYNQ702 evaluation board). The results may not achieve using other FPGA devices.

b. Fixed Embedding Technique: The proposed method may not provide high robustness against steganalysis techniques; this may cause it to be vulnerable to detection in spite of the high similarity degree of the Stego and cover image, plus minimum mean square error and peak signal-to-noise ratio.

c. Security Considerations: While the method accelerates the embedding and extraction process, it may need to provide more security levels using encryption algorithm.

d. Software-Hardware Comparison: The study compares software execution on a MATLAB platform with hardware acceleration on an FPGA with a perfect speedup ratio; it may need comparisons with other high-performance computing platforms like GPUs, but the main consideration was to compare software, not hardware.

6. Conclusion and Future Work

In this study, an FPGA has been used to accelerate the process of image steganography for the LSB algorithms in order to hide and recover three secret images in one RGB cover image.

Two designs have been proposed, one implemented in an FPGA chip (Xilinx ZYNQ702 evaluation board) using an XSG programming approach second in MATLAB platform (software coded in MATLAB Core™ i7-10750H CPU @ 2.60 GHz 2.59 GHz processor with 8 GB RAM), achieving 6537 as speed ratio. The proposed design was for accelerating the LSB-Steganography algorithm and the hiding process. The following conclusions were obtained.

1. To embed three secret images within a single cover image, the cover image must be in RGB format, while the secret images should be grayscale, ensuring the cover image is larger than all the secret images combined.
2. It is unnecessary to replace all bits of the secret image with the LSB of the cover image when they are already identical.
3. The XSG methodology for FPGA implementation offers a flexible approach, enabling seamless integration between FPGA hardware-based designs and MATLAB/SIMULINK software tools.
4. Utilizing FPGA hardware through XSG significantly accelerates pixel processing, achieving a speedup factor of up to 6537 compared to software execution on a MATLAB platform running on a Core i7-10750H CPU @ 2.60 GHz with 8 GB RAM.
5. Performance metrics, including histogram analysis, PSNR, MSE, BER, SSIM, CCR, overall execution time, frequency, and throughput, indicate that the LSB algorithm's acceleration is both efficient and effective.

The suggestions for future works are:

1- FPGA-Based pseudo random number generator to random position LSB bit for robustness.
2- FPGA-Based Image Steganography Detection using Artificial Neural Networks.
3- FPGA-Based Pixel value differencing hiding method.
4- FPGA-Based wavelet method for image steganography.
5- FPGA-Based cosine transform method for image steganography
6- Stego-Encryption accelerator system based on FPGA.

References

[1] A. A. Salih, Z. A. Abdulrazaq, and H. G. Ayoub, "Design and enhancing security performance of image cryptography system based on fixed point chaotic maps stream ciphers in FPGA," *Baghdad Science Journal*, vol. 21, no. 5 (SI), pp. 1754-1754, 2024. https://doi.org/10.21123/bsj.2024.10521

[2] H. G. Ayoub, Z. A. Abdulrazzaq, A. F. Fathil, S. A. Hasso, and A. T. Suhail, "Unveiling robust security: Chaotic maps for frequency hopping implementation in FPGA," *Ain Shams Engineering Journal*, vol. 15, no. 11, p. 103016, 2024.

[3] P. C. Mandal, I. Mukherjee, G. Paul, and B. Chatterji, "Digital image steganography: A literature survey," *Information Sciences*, vol. 609, pp. 1451-1488, 2022.

[4] S. Rahman *et al.*, "A comprehensive study of digital image steganographic techniques," *IEEE Access*, vol. 11, pp. 6770-6791, 2023. https://doi.org/10.1109/ACCESS.2023.3210124

[5] M. J. Alhaddad, M. H. Alkinani, M. S. Atoum, and A. A. Alarood, "Evolutionary detection accuracy of secret data in audio steganography for securing 5G-enabled internet of things," *Symmetry*, vol. 12, no. 12, p. 2071, 2020. https://doi.org/10.3390/sym12122071

[6] N. Subramanian, "Image steganography using deep learning methods to detect covert communication in untrusted channels," Master's Thesis, 2021.
[7] E. H. J. Halboos and A. M. Albakry, "Hiding text using the least significant bit technique to improve cover image in the steganography system," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 6, pp. 3258-3271, 2022. https://doi.org/10.11591/eei.v11i6.4249
[8] K. H. Abuhmaidan, M. A. Al-Share, A. M. Abualkishik, and A. Kayed, "Enhancing data protection in digital communication: A novel method of combining steganography and encryption," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 18, no. 6, pp. 1619-1637, 2024.
[9] R. S. Hameed, B. Abd Rahim, M. M. Taher, and S. S. Mokri, "A literature review of various steganography methods," *Journal of Theoretical and Applied Information Technology*, vol. 100, no. 5, pp. 1–10, 2022.
[10] A. AbdelRaouf, "A new data hiding approach for image steganography based on visual color sensitivity," *Multimedia Tools and Applications*, vol. 80, no. 15, pp. 23393-23417, 2021. https://doi.org/10.1007/s11042-021-11365-5
[11] S. K. Bandyopadhyay, *Unseen to seen by digital steganography: Modern-day data-hiding techniques. In Multidisciplinary Approach to Modern Digital Steganography*. IGI Global: Hershey, PA, 2021.
[12] M. Idakwo, M. Muazu, E. Adedokun, and B. Sadiq, "An extensive survey of digital image steganography: State of the art," *ATBU Journal of Science, Technology and Education*, vol. 8, no. 2, pp. 40-54, 2020.
[13] O. Evsutin, A. Melman, and R. Meshcheryakov, "Digital steganography and watermarking for digital images: A review of current research directions," *IEEE Access*, vol. 8, pp. 166589-166611, 2020.
[14] C. Zhang, P. Benz, A. Karjauv, G. Sun, and I. S. Kweon, "Udh: Universal deep hiding for steganography, watermarking, and light field messaging," *Advances in Neural Information Processing Systems*, vol. 33, pp. 10223-10234, 2020.
[15] K. Sudha, "Text steganography using LSB insertion method along with Chaos theory," *arXiv preprint arXiv:1205.1859*, 2012. https://doi.org/10.48550/arXiv.1205.1859
[16] A. T. Suhail and H. G. Ayoub, "A new method for hiding a secret file in several WAV files depends on circular secret key," *Egyptian Informatics Journal*, vol. 23, no. 4, pp. 33-43, 2022.
[17] D. K. Jain, S. Jacob, J. Alzubi, and V. Menon, "An efficient and adaptable multimedia system for converting PAL to VGA in real-time video processing," *Journal of Real-Time Image Processing*, vol. 17, no. 6, pp. 2113-2125, 2020.
[18] E. Alcaín *et al.*, "Hardware architectures for real-time medical imaging," *Electronics*, vol. 10, no. 24, p. 3118, 2021. https://doi.org/10.3390/electronics10243118
[19] S. K. Chatterjee and S. K. Vittapu, "FPGA implementation of EFSME for high efficient video coding standard," *Multimedia Tools and Applications*, vol. 81, no. 23, pp. 34087-34103, 2022.
[20] A. Bhuiyan, D. Liu, A. Khan, A. Saifullah, N. Guan, and Z. Guo, "Energy-efficient parallel real-time scheduling on clustered multi-core," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 9, pp. 2097-2111, 2020.
[21] Ó. Seijo, J. A. López-Fernández, and I. Val, "w-SHARP: Implementation of a high-performance wireless time-sensitive network for low latency and ultra-low cycle time industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3651-3662, 2020.
[22] P. Memari, S. S. Mohammadi, F. Jolai, and R. Tavakkoli-Moghaddam, "A latency-aware task scheduling algorithm for allocating virtual machines in a cost-effective and time-sensitive fog-cloud architecture," *The Journal of Supercomputing*, vol. 78, no. 1, pp. 93-122, 2022.
[23] K. Patil and B. Desai, "A trifecta for low-latency real-time analytics: Optimizing Cloud-based applications with edge-fog-cloud integration architecture," *MZ Computing Journal*, vol. 4, no. 1, p. 1− 12, 2023.
[24] G. A. Ismael, A. A. Salih, A. AL-Zebari, N. Omar, and K. J. Merceedi, "Scheduling algorithms implementation for real-time operating systems: A review," *Asian Journal of Research in Computer Science*, vol. 11, no. 4, pp. 35-51, 2021.
[25] M. Ashjaei, L. L. Bello, M. Daneshtalab, G. Patti, S. Saponara, and S. Mubeen, "Time-sensitive networking in automotive embedded systems: State of the art and research opportunities," *Journal of systems architecture*, vol. 117, p. 102137, 2021.
[26] W. Zhang, "Elf: Accelerate high-resolution mobile deep vision with content-aware parallel offloading," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, 2021, pp. 201-214.
[27] N. Talati, "Prodigy: Improving the memory latency of data-indirect irregular workloads using hardware-software co-design," presented at the IEEE International Symposium on High-Performance Computer Architecture (HPCA). IEEE, 2021
[28] Z. Houssam-Eddine, N. Capodieci, R. Cavicchioli, G. Lipari, and M. Bertogna, "The hpc-dag task model for heterogeneous real-time systems," *IEEE Transactions on Computers*, vol. 70, no. 10, pp. 1747-1761, 2020.
[29] V. Struhár, "Real-time containers: A survey," presented at the 2nd Workshop on Fog Computing and the IoT (Fog-IoT 2020). Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.
[30] M. Živković, "Face detection in Images Using an FPGA," presented at the 2021 Zooming Innovation in Consumer Technologies Conference (ZINC). IEEE, 2021.
[31] A. Leoni, P. Esposito, V. Stornelli, G. Saggio, and G. Ferri, "On the use of field programmable gate arrays in light detection and ranging systems," *Review of Scientific Instruments*, vol. 92, no. 12, p. 121501, 2021. https://doi.org/10.1063/5.0049880
[32] N. Kurata, "High-precision time synchronization digital sensing platform enabling connection of a camera sensor," in *The Twelfth International Conference on Sensor Device Technologies and Applications (SENSORDEVICES 2021) IARIA*, 2021, pp. 98-104.
[33] A. Magyari and Y. Chen, "FPGA remote laboratory using IoT approaches," *Electronics*, vol. 10, no. 18, p. 2229, 2021. https://doi.org/10.3390/electronics10182229
[34] N. Aissaoui, R. Kaibou, and M. S. Azzaz, "Real-time FPGA implementation of digital video watermarking techniques using co-design approach: Comparative study," in *2022 7th International Conference on Image and Signal Processing and their Applications (ISPA)*, 2022: IEEE, pp. 1-6.
[35] K. Dzhanashia and O. Evsutin, "FPGA implementation of robust and low complexity template-based watermarking for digital images," *Multimedia Tools and Applications*, vol. 83, no. 20, pp. 58855-58874, 2024.
[36] O. O. Ordaz-García, M. Ortiz-López, F. J. Quiles-Latorre, J. G. Arceo-Olague, R. Solís-Robles, and F. J. Bellido-Outeiriño, "DALI bridge FPGA-based implementation in a wireless sensor node for IoT street lighting applications," *Electronics*, vol. 9, no. 11, p. 1803, 2020. https://doi.org/10.3390/electronics9111803
[37] V. Kolhapure and V. K. Kodavalla, "Verification of complex multimedia system-on-chip realized in field programmable gate array device," in *2020 International Conference on Industry 4.0 Technology (I4Tech)*, 2020: IEEE, pp. 100-106.

[38] B. Bissendorf, J. Bredereke, J. Brumund, J. Knobloch, S. Pfennig, and N. Seeliger, "FPGA based active camera stabilization for a small satellite-designing a digital control," 2024. https://doi.org/10.26092/elib/2851

[39] S. Sarkar and S. S. Bhairannawar, "Efficient FPGA architecture to implement non-separable fast Fourier transform for image and video applications," *International Journal of Electronics*, vol. 110, no. 4, pp. 631-647, 2023. https://doi.org/10.1080/00207217.2023.2182359

[40] A. S. Ansari, M. S. Mohammadi, and M. T. Parvez, "A comparative study of recent steganography techniques for multiple image formats," *International Journal of Computer Network and Information Security*, vol. 11, no. 1, pp. 11-25, 2019. https://doi.org/10.5815/ijcnis.2019.01.02

[41] S. Solak, "High embedding capacity data hiding technique based on EMSD and LSB substitution algorithms," *IEEE Access*, vol. 8, pp. 166513-166524, 2020. https://doi.org/10.1109/ACCESS.2020.3010639

[42] O. F. A. Wahab, A. A. Khalaf, A. I. Hussein, and H. F. Hamed, "Hiding data using efficient combination of RSA cryptography, and compression steganography techniques," *IEEE Access*, vol. 9, pp. 31805-31815, 2021. https://doi.org/10.1109/ACCESS.2021.3056237

[43] J.-H. Horng, C.-C. Chang, and G.-L. Li, "Steganography using quotient value differencing and LSB substitution for AMBTC compressed images," *IEEE Access*, vol. 8, pp. 129347-129358, 2020. https://doi.org/10.1109/ACCESS.2020.3003302

[44] O. B. Tariq *et al.*, "High-level annotation of routing congestion for Xilinx Vivado HLS designs," *IEEE Access*, vol. 9, pp. 54286-54297, 2021. https://doi.org/10.1109/ACCESS.2021.3073658

[45] M. Alghamdi, *Design space exploration of concurrency mapping to FPGAs in weather and climate applications with Xilinx SDSoC OpenCL, SDSoC C++ and Vivado*. United Kingdom: The University of Manchester, 2022.

[46] P. Agarwal, T. K. Garg, and A. Kumar, *Low-power embedded system design applications using FPGAs* (Embedded Devices and Internet of Things). Boca Raton: CRC Press, 2025.

[47] R. Nigam, S. Thomas, Z. Li, and A. Sampson, "A compiler infrastructure for accelerator generators," in *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, 2021, pp. 804-817.

[48] N. Zhang, X. Chen, and N. Kapre, "Rapidlayout: Fast hard block placement of fpga-optimized systolic arrays using evolutionary algorithm," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 15, no. 4, pp. 1-23, 2022. https://doi.org/10.1145/3501803

[49] A. B. Perina, "Lina: a fast design optimisation tool for software-based FPGA programming," PhD Thesis, Universidade de São Paulo, 2022.

[50] J. M. De Haro *et al.*, "OmpSs@ FPGA framework for high performance FPGA computing," *IEEE Transactions on Computers*, vol. 70, no. 12, pp. 2029-2042, 2021. https://doi.org/10.1109/TC.2020.3010745

[51] D. Koch, N. Dao, B. Healy, J. Yu, and A. Attwood, "FABulous: An embedded FPGA framework," in *The 2021 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2021, pp. 45-56.

[52] S. Ullah, S. S. Sahoo, N. Ahmed, D. Chaudhury, and A. Kumar, "Appaxo: Designing app lication-specific a ppro x imate o perators for fpga-based embedded systems," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 21, no. 3, pp. 1-31, 2022.

[53] J. Cong, "Scheduling and Physical design," in *Proceedings of the 2024 International Symposium on Physical Design*, 2024, pp. 219-225.

[54] J. Cong *et al.*, "FPGA HLS today: successes, challenges, and opportunities," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 15, no. 4, pp. 1-42, 2022. https://doi.org/10.1145/3491427

[55] H. Sarieddeen, M.-S. Alouini, and T. Y. Al-Naffouri, "An overview of signal processing techniques for terahertz communications," *Proceedings of the IEEE*, vol. 109, no. 10, pp. 1628-1665, 2021. https://doi.org/10.1109/JPROC.2021.3084400

[56] Y. Hou *et al.*, "The state-of-the-art review on applications of intrusive sensing, image processing techniques, and machine learning methods in pavement monitoring and analysis," *Engineering*, vol. 7, no. 6, pp. 845-856, 2021. https://doi.org/10.1016/j.eng.2021.05.002

[57] J. A. Zhang *et al.*, "An overview of signal processing techniques for joint communication and radar sensing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 6, pp. 1295-1315, 2021. https://doi.org/10.1109/JSTSP.2021.3082337

[58] X. Dong, D. Thanou, L. Toni, M. Bronstein, and P. Frossard, "Graph signal processing for machine learning: A review and new perspectives," *IEEE Signal Processing Magazine*, vol. 37, no. 6, pp. 117-127, 2020. https://doi.org/10.1109/MSP.2020.3002853

[59] W. Xu, Z. Yang, D. W. K. Ng, M. Levorato, Y. C. Eldar, and M. Debbah, "Edge learning for B5G networks with distributed signal processing: Semantic communication, edge computing, and wireless sensing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 17, no. 1, pp. 9-39, 2023. https://doi.org/10.1109/JSTSP.2023.3242281

[60] I. H. Sarker, "Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions," *SN Computer Science*, vol. 2, no. 6, pp. 1-20, 2021. https://doi.org/10.1007/s42979-021-00637-6

[61] Y. Cui, F. Liu, X. Jing, and J. Mu, "Integrating sensing and communications for ubiquitous IoT: Applications, trends, and challenges," *IEEE Network*, vol. 35, no. 5, pp. 158-167, 2021. https://doi.org/10.1109/MNET.011.2100624

[62] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 18-44, 2021. https://doi.org/10.1109/MSP.2020.3046740

[63] J. Naranjo-Torres, M. Mora, R. Hernández-García, R. J. Barrientos, C. Fredes, and A. Valenzuela, "A review of convolutional neural network applied to fruit image processing," *Applied Sciences*, vol. 10, no. 10, p. 3443, 2020. https://doi.org/10.3390/app10103443

[64] X.-D. Zhang, *Modern signal processing*. Berlin: Walter de Gruyter GmbH & Co KG, 2022.

[65] K. H. Thanoon, A. F. Shareef, and O. A. Alsaif, "Digital processing and deep learning techniques: A review of the literature," *NTU Journal of Engineering and Technology*, vol. 1, no. 3, p. 2071, 2022. https://doi.org/10.3390/sym12122071