# High-Speed Pipelined FPGA Implementation of a Robust Steganographic Scheme for Secure Data Communication Systems

Salah Harb[a], M. Omair Ahmad[b] and M. N. S. Swamy[c]
*Electrical and Computer Engineering Department, Concordia University, 1440 De maisonnueve, Montreal, Canada*

Keywords: Information Hiding, Steganography, Communication Systems, FPGA, Pipelined Architecture, Efficiency.

Abstract: In this paper, we introduce a high-speed and area-efficient hardware design for a novel modulus-based image steganographic scheme, specifically targeting constrained-area steganographic embedded systems. The proposed modulus-based image steganography scheme enhances both image quality and embedding rate while ensuring resilience against PVD histogram analysis, salt-and-pepper noise, and RS analysis attack. The hardware architecture incorporates pipelined registers placed to guarantee balanced-execution paths among computing components. A memory-less finite state machine model is developed to efficiently control the instructions for the steganographic operations. Employing a hardware-software co-design approach, the proposed hardware design is realized as an IP core on the AMD Xilinx Zynq-7000 APSoC platform. It processes concealing operations in just 13 clock cycles, utilizes 148 slices, and operates at 290 $MHz$. This results in a remarkable throughput of 2.32 $Gbps$. The hardware design demonstrates significant improvements in speed, resource utilization, and throughput compared to recent steganographic hardware implementations, making it ideal for resource-constrained, real-time applications ranging from secure embedded communication to advanced IoT data protection.



## 1 INTRODUCTION

In the realm of digital secure data communication, the art of information hiding has become increasingly significant. As digital communication proliferates, enabling the easy distribution of assets like images and videos, it simultaneously raises critical security concerns. Information hiding techniques are pivotal in addressing these concerns, as they encompass methods for both embedding secret data in digital carriers (steganography) and transforming it (cryptography).

Among these techniques, image steganography plays a crucial role. It ensures the covert transmission of confidential data within an image. The core components of this process involve a cover image and a steganographic scheme. The primary function of the steganographic scheme is to conceal secret data within the cover image, creating what is known as a stego image. This process of concealing secret data into the cover image is referred to as the embedding process. Conversely, the process of retrieving the hidden data from the stego image is called the extracting process. To an unsuspecting receiver, the stego image appears ordinary, yet it carries hidden information that can only be extracted by the intended receiver using a shared key. Figure 1 illustrates the main components of the image steganography process.

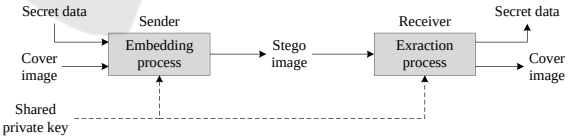



Figure 1: Main components of the image steganography.

Image steganography for data transmission operates either in frequency domain for enhanced security (Kumar and Kumar, 2018; Fakhredanesh et al., 2019; Sukumar et al., 2021; Mandal et al., 2022) or spatial domain, which alter pixels directly. The latter includes interpolation-based (Lu, 2018; Shaik and V, 2019; Hassan and Gutub, 2021), encryption-based (Zhang et al., 2016; Zhang et al., 2019; Wu et al., 2021), and modulus-based schemes (Zhang and Wang, 2006; Chao et al., 2009; Kuo et al., 2016; Liu et al., 2019; Saha et al., 2020; Kumar and Jung, 2020; Leng et al., 2021). Modulus-based schemes are fa-

[a] https://orcid.org/0000-0002-5975-6537
[b] https://orcid.org/0000-0002-2924-6659
[c] https://orcid.org/0000-0002-3989-5476



Harb, S., Ahmad, M. and Swamy, M.
High-Speed Pipelined FPGA Implementation of a Robust Steganographic Scheme for Secure Data Communication Systems.
DOI: 10.5220/0012716700003767

In *Proceedings of the 21st International Conference on Security and Cryptography (SECRYPT 2024)*, pages 397-406
ISBN: 978-989-758-709-2; ISSN: 2184-7711

vored for their balance of image quality, embedding capacity, and resistance to attacks.

The implementation of image steganography can be executed on either software or hardware platforms. While software offers versatility, it often lacks the speed required for high-performance tasks, leading to a shift towards hardware solutions for better efficiency. Traditional CPUs and ASICs, although fast, face design constraints and high costs respectively. FPGA platforms, such as the AMD Xilinx Zynq-7000 APSoC introduced in 2011, provide a good balance with their ability to handle complex steganographic schemes efficiently due to their parallel processing capabilities, combining hardware efficiency with software flexibility.

This paper presents a novel approach to image steganography, implementing a modulus-based steganographic scheme optimized for the AMD Xilinx Zynq-7000 APSoC platform. The proposed scheme conceals large volumes of secret data within small pixel groups, thus enhancing image quality without compromising security. The hardware design, developed and implemented using the AMD Xilinx Vivado 2022.1 suite, adopts a hardware-software co-design approach. This ensures high-throughput processing through optimal pipelining stages.

The rest of the paper is organized as follows. Section 2 provides a comprehensive overview of modulus-based steganographic schemes. Section 3 introduces a novel modulus-based steganographic scheme. Section 4 describes the proposed hardware architecture for the proposed image steganographic scheme. Finally, Section 5 concludes the paper.

## 2 MODULUS-BASED STEGANOGRAPHIC SCHEMES

Modulus-based steganographic schemes, exploiting modification direction (EMD) (Zhang and Wang, 2006) and diamond encoding (DE) (Chao et al., 2009), ensure comparable or better stego image quality with reduced computational complexity than their interpolation and encryption-based peers. This section provides a brief introduction to these two foundational schemes.

### 2.1 EMD and DE Schemes

In 2006, Zhang and Wang introduced a steganographic scheme that adopts a data hiding method using a $(2m+1)$-ary notational system. The method conceals secret data within $m$ consecutive pixels, with only one pixel in the group being adjusted either by incrementing, decrementing, or maintaining its original value. The embedding process in the EMD method starts by calculating the value of the weighted sum modulo function given by

$$F_{EMD} = \sum_{i=1}^{m} (i\ g_i)\ mod\ (2m+1) \tag{1}$$

where $g_i$ denotes the value of the $i$th pixel. The difference between the secret data and $F_{EMD}$ directs the pixel modification as follows. (i) if $s = 0$, no pixels are modified, (ii) if $s \leq m$, the pixel value $g_s$ is incremented by 1, and (iii) if $s > m$, the pixel value $g_{2m+1-s}$ is decremented by 1.

Figure 2 shows all the possible modifications in the pixel values in a group of two pixels ($m = 2$). The concealed digit $E$ is extracted from the modified pixel group using the weighted sum modulo function, defined as

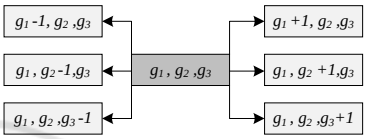



Figure 2: All possible modifications for pixels when $m = 2$.

$$E = F_{\text{EXT\_EMD}} = \sum_{i=1}^{m} i \cdot g_i^{'} \mod (2m+1) \tag{2}$$

In order to improve the embedding capacity provided by the EMD, a steganographic scheme employing diamond encoding (DE) method was introduced by (Chao et al., 2009). Within this DE method, a concealed digit $E$, expressed in the $(2k^2+2k+1)$-ary notational system, is concealed into only two pixels $p$, $q$ of in the cover image. Unlike the EMD scheme, this scheme allows modifications to either one or both pixel values, adjusting them within the range $[-k,k]$ or leaving them intact. The embedding process in the DE method starts by calculating the value of the modulo function defined by

$$F_{DE} = ((2k+1)p+q)\ mod\ (2k^2+2k+1) \tag{3}$$

It should be noted that $F_{DE}$ falls within the range $[0, 2k^2+2k]$. To determine which change from the various possible modifications in the diamond of $S_k$ should be applied to $(p,q)$, the modulus distance $d_k$ between the values of the secret digit $E$ and $F_{DE}$ is calculated using:

$$d_k = (E - F_{DE}) \mod (2k^2+2k+1) \tag{4}$$

The modulus distance $d_k$ will be in the range $[0, 2k^2+2k]$. The set of values for $d_k$, denoted by $D_k$,

can also be arranged in a diamond shape pattern. In this pattern, each $d_k$ value corresponds to a vector in $S_k$, which provides the new values for $(p,q)$. Figure 3 illustrate the distance patterns for $D_2$.

The secret digit $E$ can be extracted from the modified values $(p',q')$ of the stego image by utilizing the modulo function:

$$E = F_{EXT\_DE} = ((2k+1)p' + q') \mod (2k^2+2k+1) \quad (5)$$

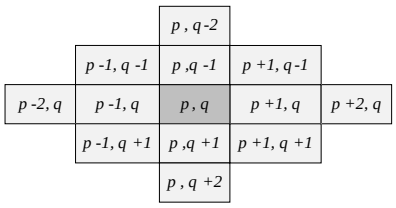



Figure 3: All possible modifications for pixels when $k = 2$.

In the EMD scheme, the choice of $m$ is pivotal for balancing the embedding rate and image quality, as noted in (Zhang and Wang, 2006). At $m = 2$, EMD achieves its peak embedding rate of 1.16 bpp. Larger $m$ values enhance stego image quality but decrease the rate. Conversely, the DE scheme is optimal in image quality at $k = 1$. Using the Lena cover image at a 100% payload yields a PSNR of 52.1 dB and an embedding rate of 1.16 bpp. Increasing $k$ raises the rate but diminishes stego image quality.

# 3 PROPOSED MODULUS-BASED STEGANOGRAPHIC SCHEME

For a given secret data block size, $L$, the EMD scheme provides a stego image with a specific image quality and embedding rate. Similarly, the DE scheme provides a stego image with its own image quality and embedding rate. When $L$ decreases, the EMD scheme delivers a decline in image quality and an uptick in embedding rate. Conversely, with the DE scheme, the trend is the opposite, as $L$ decreases, image quality improves while the embedding rate drops. This behavior is also observed when $L$ increases; however, the roles reverse, the EMD scheme results in enhanced image quality and reduced embedding rate, while the DE scheme experiences decreased image quality and a boosted embedding rate.

Given the above observations, it is essential to consider a steganographic scheme that ensures a balance performance regardless of the secret data block size $L$. Ideally, such a scheme would surpass the maximum embedding rate of the DE scheme while offering superior image quality compared to the EMD scheme. One approach could involve segmenting each $L$-sized secret data block into $L_1$ and $L_2$, embedding them using EMD and DE schemes, respectively. By doing so, we achieve a balance in image quality and embedding rate superior to solely using the EMD or DE scheme.

To further enhance the image quality and embedding rate, two lookup tables are constructed, representing the two segments $L_1$ and $L_2$. The segment $L_1$ is searched within the first lookup table to retrieve two secret digits, $S_0$ and $S_1$, which represent $L_1$. Similarly, the segment $L_2$ is located within the second lookup table to obtain the secret digit $S_2$ that represents $L_2$.

## 3.1 The Algorithm

A secret data block $B$ of size $L$ is divided into two segments of sizes $L_1$ and $L_2$. The first segment is represented by two secret digits, which are concealed by the EMD scheme using the $(2n+1)$-ary notational system. The second segment is represented by one secret digit, and this is concealed by the DE scheme using the $(2k^2+2k+1)$-ary notational system.

Figure 4 depicts the first lookup table for $L_1 = 4$ bits with $m_x = 2$ and $m_y = 2$. Given that $L_1 = 4$ bits, the table comprises 16 $(2^4)$ entries. The values for these entries span the range $[0, 2^4 - 1]$ or $[0000, 1111]_2$. Each value in this table corresponds to the first segment $L_1$ of the secret data block with size $L$. The table is aligned along the $x$- and $y$-coordinate axes, with each axis using its own notational system to denote secret digits. In particular, the $x$-coordinate axis encapsulates all $2m_x$ possible values from the $(2m_x+1)$-ary system, while the $y$-coordinate axis encompasses all $2m_y$ possible values from the $(2m_y+1)$-ary system. As illustrated in Figure 4, the $x$-coordinate axis ranges from [0, 4] for a group of 2 pixels, and similarly, the $y$-coordinate axis also spans [0, 4] for a group of 2 pixels.

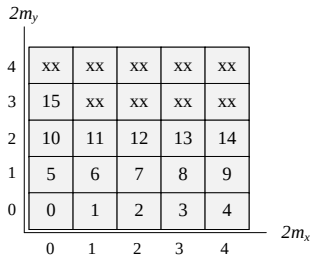



Figure 4: The $x-y$-coordinate lookup table when $L_1 = 4$.

After constructing the $x-y$ coordinate lookup table, the coordinates $(x,y)$ are determined by searching for the value of $L_1$. The values of these coordinates,

$(x,y)$, denote secret digits $S_0$ and $S_1$. Here, $S_0$ corresponds to the $x$-coordinate and $S_1$ to the $y$-coordinate, both representing the value of $L_1$. These secret digits are then embedded in two groups of pixels, $Q_0$ and $Q_1$, using the EMD scheme. Specifically, the $Q_0$ group uses $m_x$ pixels to embed the secret digit $S_0$, and the $Q_1$ group uses $m_y$ pixels to embed $S_1$.

For the segment $L_2$, an additional $z$-coordinate axis is introduced to conceal a third secret digit, $S_2$, which represents $L_2$ using the DE scheme. The $z$-coordinate axis uses the vector set $S_{m_z}$ within the $(2m_z^2+2m_z+1)$-ary notational system. By searching for $L_2$ in the distance pattern $D_{m_z}$, one can determine the specific vector $s_{m_z}$ to be concealed within the third group $Q_2$ of two pixels, denoted as $(p,q)$. Figure 5 illustrates the $z$ coordinate lookup table when $L_2 = 3$. For this segment, the suitable value for the $m_z = 2$. It can be seen from this figure that the $z$-coordinate axis also represents the DE distance pattern $D_2$.

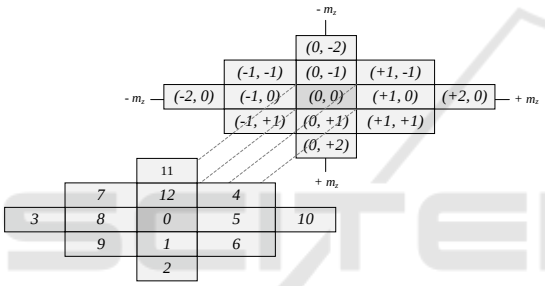



Figure 5: The $z$-coordinate lookup table when $L_2 = 3$.

The three secret digits $S_0$, $S_1$ and $S_2$, represent the secret data block $B$ of size $L$, and three groups of pixels $Q_0$, $Q_1$ and $Q_2$, are utilized to conceal them, respectively. The entire concealing process is summarized in Algorithm 1.

In the EMD scheme, the secret data block of size $L$ can be represented using $\lceil \frac{L}{\log_2(2m+1)} \rceil$ $(2m+1)$-ary secret digits. For example, if $L = 7$, then $m = 6$ and the number of secret digits required is $\lceil \frac{7}{\log_2(2m+1)} \rceil$ or 2 of 13-ary secret digits to represent a 7-bit secret data block. Each secret digit is concealed into a 6-pixel group, thus, 2 of 13-ary secret digits need 12 pixels to conceal the secret data block of 7-bit size. In the DE scheme, the secret data block of size $L$ can be represented using a distance pattern $D_k$ that contains $(2k^2+2k+1)$ secret digits. For example, if $L = 8$, then $k = 11$ and $D_{11}$ has 265 secret digits.

On the other hand, using the proposed scheme, if the size of the secret data block $L = 7$ and the parameters for the $x-y-z$ coordinate lookup table are $m_x = 2$, $m_y = 2$, $m_z = 2$, $L_1 = 4$, and $L_2 = 3$, then the $x$-axis for the $x-y$ coordinate lookup table is $2m_x$ and it requires 2 pixels, and the $y$-axis in the table is $2m_y$ and it requires 2 pixels. Also, 2 pixels are required for the $L_2$ with $k = 2$ and $D_2$ has 13 secret digits. This indicates that our scheme requires a total of 6 pixels to conceal the $L = 7$, which is fewer pixels than the EMD scheme and has smaller distance patterns than the DE scheme.

Algorithm 1: Proposed modulus-based image steganographic scheme.

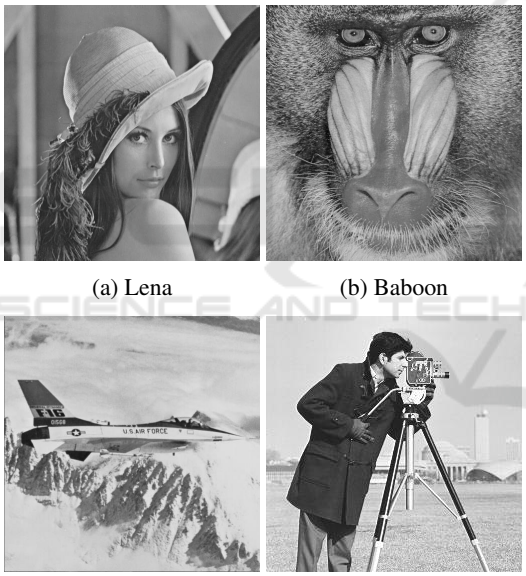
**Input:** Cover image, segments $(L_1, L_2)$, a stream of $L$-bit secret data blocks $B$
**Output:** Stego image
**Initialization:** $x-y$ coordinate lookup table $(0:2m_x, 0:2m_y)$ with entries: $[0:2^{C_{EMD}}-1]$ and $x-y-z$ coordinate lookup table $(-m_z:+m_z, -m_z:+m_z)$ with entries: $[0:2m_z^2+2m_z]$
**while** $Num(B) \neq 0$ **do**
  $Q_0 = m_x$-pixels, $Q_1 = m_y$-pixels, $Q_2 = (p,q)$;
  $(x,y) \leftarrow$ x-y Lookup table$(L_1)$;
  $D_{m_z} \leftarrow$ z Lookup table$(L_2)$;
  $S_0 \leftarrow x, S_1 \leftarrow y, S_2 \leftarrow D_{m_z}$;
  EMD$[S_0]$ using $Q_0$ group $(2m_x+1)$-ary system;
  EMD$[S_1]$ using $Q_1$ group $(2m_y+1)$-ary system;
  DE$[S_2]$ using $Q_2$ group with $(2m_z^2+2m_z+1)$-ary system;

The proposed scheme offers a simplified approach to the conversion of secret data blocks into $l$-ary digits. The conversion process is avoided, which eliminates the need for complex operations like multiplication and division, especially when the cut-offs $L_1$ and $L_2$ are large. Instead, the secret data blocks are stored directly in the lookup tables, and the secret digits $S_0$, $S_1$, and $S_2$ are used for concealing. This approach provides an additional layer of security since the secret data are not directly involved in the concealing process.

The complexity of the proposed scheme is primarily limited to the construction of the lookup table. The construction process becomes more complex as the size of $L$ increases and the cut-off lengths $L_1$ and $L_2$ are increased. Larger cut-off values lead to more entries in the lookup tables, resulting in a more complex construction process. However, this process only needs to occur once at the beginning of the communication session for that specific application. In case the session is expired, or cut-offs are changed, the lookup table can be reconstructed.

## 3.2 Experimental Results

In this section, we present a demonstration and analysis of the results obtained from the experiments using the proposed steganographic scheme. Figure 6 and Figure 7 show a set of commonly used cover images Lena, Baboon, Airplane, Cameraman, and their resulting stego images. A maximum payload is concealed in these cover images. We evaluate our scheme using three performance metrics, which are (i) embedding rate ($bpp$) (ii) image quality, and (iii) robustness against steganalysis attacks. For image quality, we obtain the PSNR/SSIM values of the resulting stego images. The resulting stego images are generated after using the selected parameters $L_1 = 4$, $L_2 = 3$ for the $x - y$ and $z$ coordinate lookup tables. It can be seen from Figure 7 that the imperceptibility is very high in the stego images after concealing the secret data. Higher PSNR/SSIM values indicate that artifacts can not be detected by the human visual system.

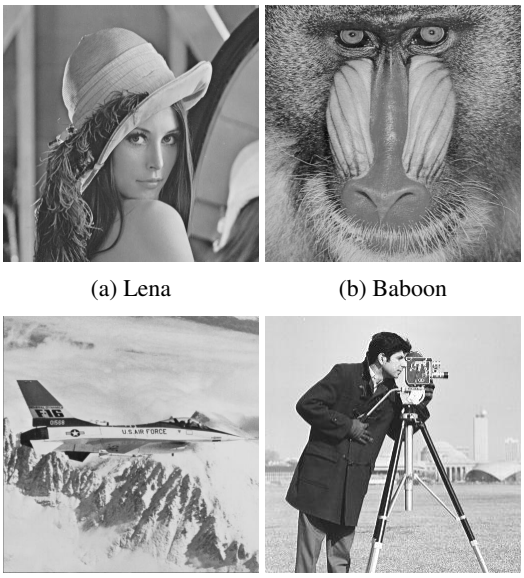


(a) Lena (b) Baboon

(c) Airplane (d) Cameraman

Figure 6: Commonly-used cover images.

In Table 1, we compare the proposed scheme with recent works, all using the same cover images at a 100% payload. The schemes are grouped into interpolation-based, encryption-based, and modulus-based categories. Our proposed scheme, employing $x - y$ and $z$ coordinate lookup tables, surpasses others by achieving the highest PSNR/SSIM values, indicating superior visual quality of the stego images. It also leads in embedding rate. Specifically, using Lena as the cover image with parameters $L_1 = 5$ and $L_2 = 5$, embedding rate achieved by our proposed scheme is 2.15 and 1.35 times higher than that of the EMD and



(a) Lena (b) Baboon

(c) Airplane (d) Cameraman

Figure 7: Resulting stego images: (a) PSNR 55.973 $dB$, (b) PSNR 55.939 $dB$,(c) PSNR 55.890 $dB$, (d) PSNR 55.898 $dB$.

DE schemes, respectively. This demonstrates that our combined steganographic scheme performs more efficiently than the original EMD and DE schemes.

## 3.3 Steganalysis Analysis

The robustness of a stegangraphic scheme can be evaluated by performing steganalysis attack. In this section, we perform such an analysis by considering pixel value difference (PVD) histogram, salt and pepper noise, and regular/singular (RS) analysis attacks.

### 3.3.1 PVD Histogram Analysis

The pixel value difference (PVD) histogram, used in blind steganalysis, detects hidden data in digital images by comparing pixel value differences between a cover image and its stego image. Significant changes in the PVD histogram indicate the presence of secret data. Figure 8 illustrates the average PVD histograms for cover and stego images generated by using the proposed, EMD, and DE schemes with a 100% payload. The proposed scheme shows fewer modifications to the cover image, evidenced by a higher count of zeros in the histogram, implying a lower chance of detecting hidden secret data. This minimal alteration keeps the stego image less suspicious, demonstrating that the proposed scheme is effective against PVD histogram analysis attack.

Table 1: PSNR/SSIM and embedding rate comparisons of other recent works and the proposed scheme.

| Scheme category | Scheme | Cover image | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Lena | | | Baboon | | | Airplane | | |
| | | PSNR (dB) | SSIM | bpp | PSNR (dB) | SSIM | bpp | PSNR (dB) | SSIM | bpp |
| Interpolation based | (Lee and Huang, 2012) (2012) | 22.32 | 0.867 | 1.32 | 21.24 | 0.651 | 1.32 | 23.76 | 0.797 | 1.34 |
| | (Lu, 2018) (2018) | 40 | - | 1.2 | 35.99 | - | - | 40 | - | 1.2 |
| | (Wahed and Nyeem, 2019) (2019) | 46.88 | 0.9352 | 1.87 | 47.19 | 0.9922 | 1.88 | 39.55 | 0.914 | 1.445 |
| | (Shaik and V, 2019) (2019) | 32.24 | - | 1.8 | 22.98 | - | 2.1 | 29.42 | - | 1.2 |
| | (Chen et al., 2020) (2020) | 34.18 | - | 1.41 | 24.69 | - | 2.4 | 32.84 | - | 1.27 |
| | (Hassan and Gutub, 2021) (2021) | - | - | - | 21.53 | - | 2.1 | 28.11 | - | 1.32 |
| Encryption based | (Zhang et al., 2016) (2016) | 36.3 | - | 0.25 | 37.5 | - | 0.25 | 37.8 | - | 0.25 |
| | (Zhang et al., 2019) (2019) | 44 .41 | - | 0 .5 | 27 .11 | - | 0 .5 | 44 .94 | - | 0 .5 |
| | (Wu et al., 2021) (2021) | 48 | - | 2.39 | 22 | - | 0.9 | 23 | - | 2.4 |
| | (Manikandan and Zhang, 2022) (2022) | - | - | - | 28.29 | 0.97 | 0.063 | 45.7 | 0.999 | 0.125 |
| Modulus based | (Zhang and Wang, 2006) (2006) | 51.8 | 0.9971 | 1.16 | 51.8 | 0.9989 | 1.16 | 51.79 | 0.9967 | 1.16 |
| | (Jung and Yoo, 2009) (2009) | 47.92 | 0.9904 | 2.3 | 47.95 | 0.9975 | 2.3 | 47.97 | 0.9897 | 2.3 |
| | (Chao et al., 2009) (2009) | 52.1 | 0.9965 | 1.85 | 46.3 | 0.9962 | 1.85 | 46.7 | 0.996 | 1.85 |
| | (Li and He, 2018) (2018) | 42.74 | - | 2.1 | 36.63 | - | 2.6 | 43.23 | - | 2.1 |
| | (Liu et al., 2019) (2019) | 51.8 | - | 1.16 | - | - | - | 51.54 | | 1.15 |
| | (Kumar and Jung, 2020) (2020) | 39.5 | - | 0.22 | 38.5 | - | 0.22 | 38.5 | - | 0.22 |
| | (Saha et al., 2020) (2020) | 48.66 | - | 2 | 48.52 | - | 2 | 48.52 | - | 2 |
| | (Peng et al., 2020) (2020) | 43.2 | - | 0.75 | 38.5 | - | 0.6 | 46.5 | - | 0.72 |
| | (Leng et al., 2021) (2021) | 43.74 | - | 2.5 | 43.73 | - | 2.5 | 43.74 | - | 2.5 |
| | **Proposed scheme** | **59.987** | **0.9992** | **2.49** | **59.321** | **0.9990** | **2.49** | **59.307** | **0.9994** | **2.49** |

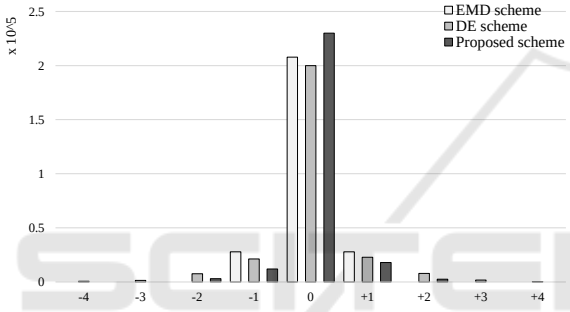



Figure 8: Average PVD histograms between cover images and their stego images generated by the EMD, DE and proposed schemes.

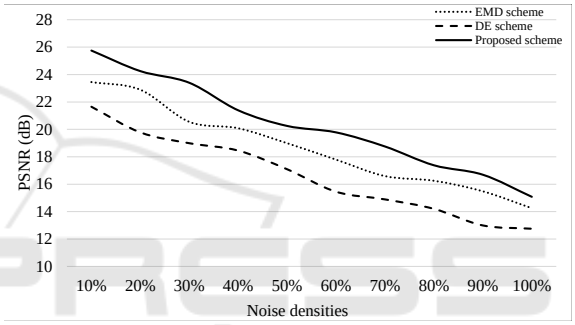



Figure 9: Comparison of the average PSNR for salt and pepper noisy images with noise densities.

### 3.3.2 Salt and Pepper Noise Analysis

The robustness of a steganographic scheme in noisy conditions is tested by introducing salt and pepper noise. This noise alters certain pixels to either black or white and is added to the stego images at varying intensities, ranging from 10% to 100%. Figure 9 shows the PSNR values for the corrupted stego images, where the stego images themselves are obtained using the proposed scheme as well as that by using the EMD and DE schemes. It is seen from this figure that the proposed scheme provides the highest PSNR values at all the levels of the noise indicating that the attacker would be led to believe less that there is a secret data hidden in the stego image produced by the proposed scheme.

The bit error rate (BER) is defined as the proportion of incorrectly extracted secret data bits to the total number of bits embedded. A lower BER value signifies a higher success rate in accurately extracting the secret data, even when the stego image has been compromised by salt and pepper noise. Figure 10 illustrates the BER values for these noise-affected stego images. These images were generated using the proposed scheme and the existing EMD and DE schemes. The data presented in this figure clearly indicates that the proposed scheme consistently achieves the lowest BER values across all levels of noise, highlighting its superior performance in ensuring the accurate retrieval of secret data from corrupted stego images.

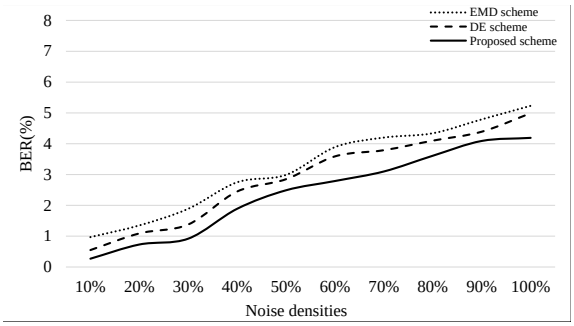



Figure 10: Comparison of the average BER for salt and pepper noisy images with noise densities.

### 3.3.3 RS Analysis Attack

RS analysis attack detects hidden data in images by analyzing the regular $(Rm)$ and singular $(Sm)$ groups of pixels. This analysis evaluates the changes in these groups before and after flipping the least significant bits (LSBs), identifying irregularities that are indicative of steganographic alterations (Fridrich et al., 2001). Figure 11 illustrates the RS analysis results for the proposed scheme applied to the cover image Lena. The analysis suggests that the stego images exhibit no signs of data concealed in their LSBs, as evidenced by the closely matching expected values of the regular $(Rm, R')$ and singular $(Sm, S')$ pixel groups, both with and without flipped LSBs. The ratios $Rm/R'm$ and $Sm/S'm$ demonstrate a similarity characteristic of unaltered images. Consequently, this strongly indicates that the proposed steganographic scheme is effective in concealing data, maintaining the natural balance between the $R$ and $S$ groups (i.e., $Rm \cong R'm$ and $Sm \cong S'm$), thereby securing against RS analysis attack.

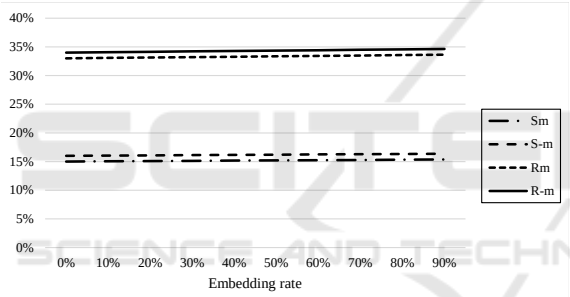



Figure 11: RS-analysis yielded for Lena stego image generated by the proposed scheme.

# 4 FPGA IMPLEMENTATION: RESULTS AND COMPARISONS

Embedded systems within FPGA platforms primarily consist of a processing system (PS) and programmable logic (PL). The PS typically includes a processor, system memory, and built-in peripherals, while the PL provides flexibility for custom hardware implementations. This combination facilitates the development of highly customizable and efficient embedded systems tailored to specific applications. Among the many FPGAs capable of hosting embedded systems, notable options include the Nexys Video Artix-7 FPGA, AMD Xilinx Zynq-7000 SoC ZC706 Evaluation Kit, and the ZedBoard Zynq-7000. Each of these boards has its own strengths, with considerations such as image processing capabilities, I/O expansions, processor performance, and cost-effectiveness playing a crucial role in their selection. Given these criteria, the ZedBoard Zynq-7000 emerges as our chosen platform due to its balance of hardware capability and affordability.

The ZedBoard specifically features the AMD Xilinx Z-7020 Zynq-7000 APSoC, which combines a dual Cortex-A9 ARM processor operating at 866 $MHz$ within its PS, and an Artix-7 FPGA in its PL. This architecture not only supports advanced image processing for steganographic applications but also facilitates efficient communication between the PS and PL via industry-standard AXI connections. The fixed architecture of the PS, including essential peripherals and memory controllers, complements the customizable nature of the PL, where the Artix-7 FPGA allows for the implementation of specialized IP cores.

## 4.1 Overall System Architecture

In this paper, designing and implementing the proposed steganographic scheme within this steganographic embedded system framework is essential. Figure 12 illustrates the overall schematic flow of the proposed reconfigurable Zynq-7000 APSoC for the proposed steganographic scheme.

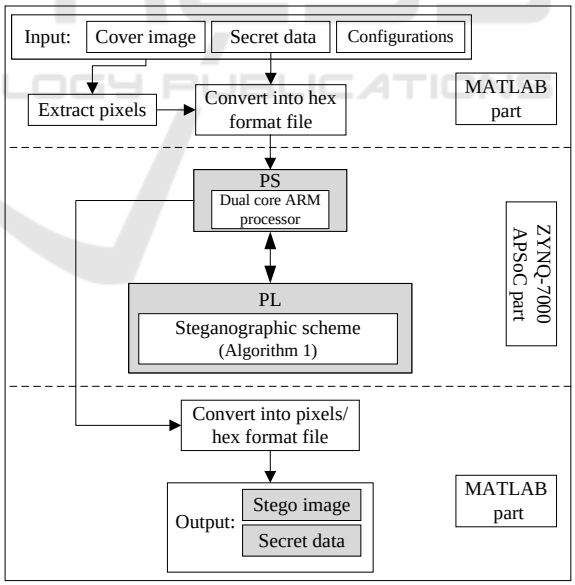



Figure 12: General schematic flow of the proposed reconfigurable Zynq-7000 APSoC (Algorithm 1).

The schematic follows a top-bottom flow, where MATLAB is utilized for image representation and pixel-hex format conversion. The embedding process is as follows. (i) The Zynq-7000 APSoC platform receives the cover image and the secret data as

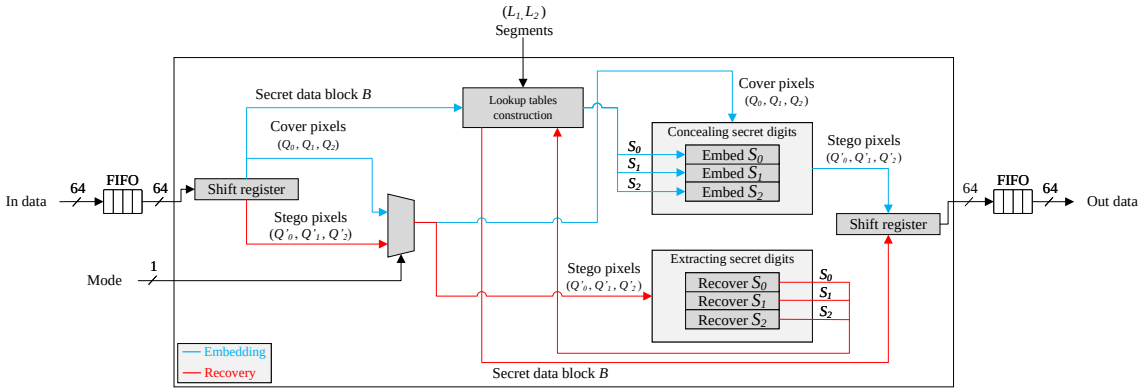



Figure 13: Hardware architecture of the steganographic scheme.

two separate inputs. (ii) MATLAB extracts the pixels from the cover image and converts them into a hex value file, along with the secret data. (iii) The reconfigurable Zynq-7000 embedded system acquires this file, containing the hex values of the image pixels and the secret data, into its PS. (iv) The steganographic scheme conceals the secret data within the hex values of the image pixels. (v) MATLAB is used once again to transform the modified hex values back into pixels, generating the final stego image.

## 4.2 Proposed Hardware Architecture

The hardware architecture of the steganographic scheme is depicted in Figure 13. A data block, comprising hex values of pixels and secret data, is sourced from the DDR3 RAM to feed the IP core. As the input FIFO becomes fully loaded, the embedding process initiates. Once this begins, an interrupt signal is activated to denote that the IP core is currently occupied, preventing further DDR3 RAM reads. Depending on the Mode input signal, a shift register reads from the input FIFO to either perform the embedding (concealing) or the recovery (extracting) of the secret data.

The hardware implementation of the steganographic scheme is carried out by implementing the two processes: embedding and recovery, as well as the two lookup tables. The lookup tables (LUTs) of the Artix-7 FPGA device are utilized to facilitate the weighted sum modulo functions of the EMD and DE schemes. To enhance the performance of the architecture, sub pipelining is employed to minimize the delay in the critical path, a technique especially effective for architectures with iterative behavior. By strategically adding the optimal number of sub pipelining stages, we strike an efficient balance between speed and area. Figure 14 illustrates the trade-off between delay and area for 1 to 4 sub pipelining stages within our design. The optimal trade-off is realized with two stages: the first inserted after the constructed lookup tables and the second inserted after the weighted sum modulo functions. To further refine timing performance, a register balancing (retiming) strategy is integrated into the architecture.

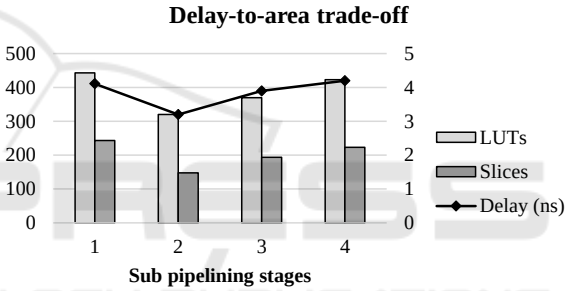



Figure 14: Delay-to-area (slices, LUTs) trade-off using different number of sub pipelining stages.

## 4.3 Place and Route Implementation Results

The development of the steganographic IP core involves three stages: design, implementation, and integration. Initially, RTL Verilog is used in the design stage to craft an effective steganographic approach tailored for FPGA use. This stage includes functional and timing simulations with AMD Xilinx ISim to validate design performance. Next, during implementation, these designs are synthesized and physically realized on the FPGA, with a focus on optimizing resource use, speed, and power efficiency, utilizing the Vivado tool for enhancements like register balancing and logic optimization. A timing constraint ensures an optimal trade-off between size and speed, aiming for zero timing errors.

Table 2 presents the place and rout results of the hardware implementation in comparison with other recent FPGA steganographic implementations. The proposed hardware implementation outperform oth-

Table 2: Comparison of the implemented scheme with other implementations (place and route results).

| Scheme | FPGA | Slices | FFs | LUTs | BRAMs | Freq. (*MHz*) | Throughput (*Mbps*) | Power (*mW*)* | Image quality (PSNR) | Embedding rate (*bpp*) | Processing rate (*fps*) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| EMD-based (Shet et al., 2019) | Artix-7 | 296 | 1890 | 1419 | - | 144.00 | 143.9 | 3807 | 45.6 | 1.16 | 549.00 |
| Lifting-based (Phadikar et al., 2019) | Artix-7 | 476 | 555 | 350 | 128 | 130.14 | 188.8 | 78.45 | 34.79 | 0.004 | 75.00 |
| Parallelism-based (Seyed Dizaji et al., 2021) | Virtex-6 | - | - | - | - | 118.66 | - | - | 42.99 | 0.26 | - |
| Threshold-based (Wei et al., 2021) | Spartan-6 | - | 670 | 627 | 7 | - | - | 200 | 44.43 | 0.25 | - |
| LSB matching (Sinha Roy et al., 2021) | Spartan-3A | - | 647 | 668 | - | - | - | 369.3 | 51.37 | 1.00 | - |
| Chaotic-based (Sun et al., 2023) | Cyclone IV | 1100 | 537 | 1067 | - | - | - | - | 53.24 | 2.00 | - |
| Template-based (Dzhanashia and Evsutin, 2023) | Artix-7 | - | - | - | 1 | 120 | 13.516 | 2450 | 39.66 | 0.00097 | 8.593 |
| Our design | **Artix-7** | **148** | **523** | **320** | **2** | **289.7** | **2318** | **152** | **81.92** | **5.52** | **1054.07** |

* The values given are the powers consumed by only the PL parts of the various hardware implementations.

ers in speed, resource utilization, throughput, and power consumption. The enhanced performance is attributed to the sub pipelining applied in the hardware architecture of the steganographic scheme. Power consumption is assessed using the AMD Xilinx power estimator (XPE) tool, with values in Table 2 representing only the PL part (FPGA device). Notably, our design is ideal for low-power, resource-limited devices such as wearables and RFIDs, given its compact footprint. Efficiency is evident in the operating speed to LUT ratio, at 0.905, surpassing the previous best of 0.10 reported in (Shet et al., 2019). Furthermore, our hardware implementation achieves a throughput to LUT ratio of 7.24, which is higher than the 0.54 reported in (Phadikar et al., 2019).

In the integration phase, the implemented design is packaged as an IP core using the AMD Xilinx IP package integrator. This tool is used to rapidly connect the design, implemented on PL, to PS using the AXI4 interface. The image steganographic system using the Zynq-7000 APSoC has four IP cores connected to PS in the Zynq-7000 APSoC. The steganographic IP core is considered a custom core and can be modified to add more features and support more functions. The UART and I/O IP cores, on the other hand, are produced by AMD Xilinx.

To achieve real-time steganographic performance, modulo arithmetic for data embedding is executed on-the-fly by a dedicated, optimized IP core on the AMD Xilinx Zynq-7000 APSoC. The stego images are generated instantly as cover images and secret data are received, with a library of cover images pre-loaded in DDR3 RAM to eliminate delays. The processing system (PS) utilizes two buffers to manage the flow of images and data to and from the DDR3 RAM, facilitating fast, cyclic processing by the steganographic core. This core also supports parallel processing, enhancing the system's speed. These measures result in a high-performance steganographic system with superior speed, efficiency, and lower power consumption, surpassing other hardware implementations.

# 5 CONCLUSIONS

This paper introduces a novel steganographic scheme implemented on the AMD Xilinx Zynq-7000 APSoC hardware platform. It features an integrated IP core that significantly enhances efficiency, speed, and power consumption. The design demonstrates superior operating speed, resource utilization, and throughput compared to recent hardware steganography solutions, making it ideally suited for real-time processing and secure data communication applications.

# ACKNOWLEDGMENTS

This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada and in part by the Regroupement Strategique en Microelectronique du Quebec (ReSMiQ).

# REFERENCES

Chao, R.-M., Wu, H.-C., Lee, C.-C., and Chu, Y.-P. (2009). A Novel Image Data Hiding Scheme with Diamond Encoding. *EURASIP Journal on Information Security*, 2009(1):658047.

Chen, Y.-q., Sun, W.-j., Li, L.-y., Chang, C.-C., and Wang, X. (2020). An efficient general data hiding scheme based on image interpolation. *Journal of Information Security and Applications*, 54:102584.

Dzhanashia, K. and Evsutin, O. (2023). FPGA implementation of robust and low complexity template-based watermarking for digital images. *Multimedia Tools and Applications*.

Fakhredanesh, M., Rahmati, M., and Safabakhsh, R. (2019). Steganography in discrete wavelet transform based on human visual system and cover model. *Multimedia Tools and Applications*, 78(13):18475–18502.

Fridrich, J., Goljan, M., and Du, R. (2001). Reliable detection of lsb steganography in color and grayscale images. In *Proceedings of the 2001 Workshop on Multimedia and Security: New Challenges*, page 27–30. Association for Computing Machinery.

Hassan, F. S. and Gutub, A. (2021). Efficient Image Reversible Data Hiding Technique Based on Interpolation Optimization. *Arabian Journal for Science and Engineering*, 46(9):8441–8456.

Jung, K.-H. and Yoo, K.-Y. (2009). Improved Exploiting Modification Direction Method by Modulus Operation. *International Journal of Signal Processing, Image Processing and Pattern*, 2:11.

Kumar, R. and Jung, K.-H. (2020). Robust reversible data hiding scheme based on two-layer embedding strategy. *Information Sciences*, 512:96–107.

Kumar, V. and Kumar, D. (2018). A modified DWT-based image steganography technique. *Multimedia Tools and Applications*, 77(11):13279–13308.

Kuo, W.-C., Wang, C.-C., and Hou, H.-C. (2016). Signed digit data hiding scheme. *Information Processing Letters*, 116(2):183–191.

Lee, C.-F. and Huang, Y.-L. (2012). An efficient image interpolation increasing payload in reversible data hiding. *Expert Systems with Applications*, 39(8):6712–6719.

Leng, H.-S., Lee, J.-F., and Tseng, H.-W. (2021). A high payload EMD-based steganographic method using two extraction functions. *Digital Signal Processing*, 113:103026.

Li, Z. and He, Y. (2018). Steganography with pixel-value differencing and modulus function based on PSO. *Journal of Information Security and Applications*, 43:47–52.

Liu, Y.-X., Yang, C.-N., Sun, Q.-D., Wu, S.-Y., Lin, S.-S., and Chou, Y.-S. (2019). Enhanced embedding capacity for the SMSD-based data-hiding method. *Signal Processing: Image Communication*, 78:216–222.

Lu, T.-C. (2018). Interpolation-based hiding scheme using the modulus function and re-encoding strategy. *Signal Processing*, 142:244–259.

Mandal, P. C., Mukherjee, I., Paul, G., and Chatterji, B. N. (2022). Digital image steganography: A literature survey. *Information Sciences*, 609:1451–1488.

Manikandan, V. M. and Zhang, Y.-D. (2022). An adaptive pixel mapping based approach for reversible data hiding in encrypted images. *Signal Processing: Image Communication*, 105:116690.

Peng, F., Zhao, Y., Zhang, X., Long, M., and Pan, W.-q. (2020). Reversible data hiding based on RSBEMD coding and adaptive multi-segment left and right histogram shifting. *Signal Processing: Image Communication*, 81:115715.

Phadikar, A., Maity, G. K., Chiu, T.-L., and Mandal, H. (2019). FPGA Implementation of Lifting-Based Data Hiding Scheme for Efficient Quality Access Control of Images. *Circuits, Systems, and Signal Processing*, 38(2):847–873.

Saha, S., Chakraborty, A., Chatterjee, A., Dhargupta, S., Ghosal, S. K., and Sarkar, R. (2020). Extended exploiting modification direction based steganography using hashed-weightage Array. *Multimedia Tools and Applications*, 79(29):20973–20993.

Seyed Dizaji, S. H., Zolfy Lighvan, M., and Sadeghi, A. (2021). Hardware-Based Parallelism Scheme for Image Steganography Speed up. In *Proc. International Conference on Innovative Computing and Communications*, Advances in Intelligent Systems and Computing, pages 225–236, Singapore. Springer.

Shaik, A. and V, T. (2019). High capacity reversible data hiding using 2D parabolic interpolation. *Multimedia Tools and Applications*, 78(8):9717–9735.

Shet, K. S., Aswath, A. R., Hanumantharaju, M. C., and Gao, X.-Z. (2019). Novel high-speed reconfigurable FPGA architectures for EMD-based image steganography. *Multimedia Tools and Applications*, 78(13):18309–18338.

Sinha Roy, S., Basu, A., Chattopadhyay, A., and Das, T. S. (2021). Implementation of image copyright protection tool using hardware-software co-simulation. *Multimedia Tools and Applications*, 80(3):4263–4277.

Sukumar, A., Subramaniyaswamy, V., Ravi, L., Vijayakumar, V., and Indragandhi, V. (2021). Robust image steganography approach based on RIWT-Laplacian pyramid and histogram shifting using deep learning. *Multimedia Systems*, 27(4):651–666.

Sun, J.-y., Cai, H., Gao, Z.-b., Wang, C.-p., and Zhang, H. (2023). A novel non-equilibrium hyperchaotic system and application on color image steganography with FPGA implementation. *Nonlinear Dynamics*, 111(4):3851–3868.

Wahed, M. A. and Nyeem, H. (2019). Reversible data hiding with interpolation and adaptive embedding. *Multimedia Tools and Applications*, 78(8):10795–10819.

Wei, J., Quan, Z., Hu, Y., Liu, J., Zhang, H., and Liu, M. (2021). Implementing a Low-Complexity Steganography System on FPGA. In *Proc. 9th International Conference on Intelligent Computing and Wireless Optical Communications (ICWOC)*, pages 64–68.

Wu, F., Zhou, X., Chen, Z., and Yang, B. (2021). A reversible data hiding scheme for encrypted images with pixel difference encoding. *Knowledge-Based Systems*, 234:107583.

Zhang, R., Lu, C., and Liu, J. (2019). A high capacity reversible data hiding scheme for encrypted covers based on histogram shifting. *Journal of Information Security and Applications*, 47:199–207.

Zhang, X., Long, J., Wang, Z., and Cheng, H. (2016). Lossless and Reversible Data Hiding in Encrypted Images With Public-Key Cryptography. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(9):1622–1631.

Zhang, X. and Wang, S. (2006). Efficient Steganographic Embedding by Exploiting Modification Direction. *IEEE Communications Letters*, 10(11):781–783.