Comprehensive Survey of Image Steganography Systems based on FPGA Implementation

Ali Y. Al-Ashwal (*,1)

Waled Hussein Al-Arashi [1]

Abdul-Malik H. Y. Saad [2]

Mohammed Al-Shadadi [1]

[1] Department of Electronics Engineering, Faculty of Engineering, University of Science and Technology, Sana'a, Yemen

[1] College of Engineering, University of Buraimi, Al Buraimi, 512, Oman

* Corresponding author: ebn_alshahel@hotmail.com

https://doi.org/10.59222/ustjet.2.1.3

Comprehensive Survey of Image Steganography Systems based on FPGA Implementation

Abstract:

Steganography is the art of concealed communication. The basic idea of steganography is to hide secret data inside a cover media in an undetectable manner. Cover media and Secret data can be any popular digital media such as image, video, audio, or text. Various types of image steganography methods have been proposed according to the cover image domain, i.e. spatial domain and transform domain. The spatial domain methods are easy and simple while the transform domain methods are more complex. However, the transform domain methods are more robust against image processing operations. Besides, they are more secure and less detectable in an unsecured channel. The recent steganography methods are based on sophisticated algorithms that include several computational time-consuming tasks, and hence they are unable to embed and extract the hidden data in real-time. Therefore, the trend is to implement the steganography methods in the hardware to speed up their processing time and so improves the efficacy of steganography techniques. FPGA is used in various fields in the modern era due to lower development cost, flexibility, and reconfigurability. Most of the published steganography techniques carried out on FPGA are based on the spatial domain. In this paper, it has comprehensively studied and reviewed various existing implementations of image steganography on FPGAs. This includes studying their general operations, requirements, and performance evaluations. This review would assist researchers in finding the research gaps in FPGA implementation of steganography methods for real-time applications.

Keywords: steganography, information hiding, FPGA, hardware implementation, LSB, DWT.

https://doi.org/10.59222/ustjet.2.1.3

دراسة شاملة لأنظمة إخفاء المعلومات في الصور استناداً إلى التنفيذ في مصفوفة البوابات المنطقية القابلة للبرمجة

الملخص:

علم اخفاء المعلومات (Steganography) هو فن الاتصال المخفي، حيث تقوم الفكرة الأساسية لعلم إخفاء المعلومات على إخفاء البيانات السرية داخل وسائط الغلاف بطريقة لا يمكن اكتشافها. ويمكن استخدام أي وسائط رقمية شائعة مثل الصور أو الفيديو أو الصوت أوٍ النص كوسائط للغلاف. هناك أنواع مختلفة من تقنيات إخفاء المعلومات في الصور الرقمية وفقً لمجال صورة الغلاف، مثل تقنيات المجال المكاني وتقنيات مجال التحويل. تعتبر تقنيات وطرق المجال المكانيٍ الأكثر استخدام بسبب سهولة وبساطة تنفيذها، في حين أن طرق مجال التحويل أكثر تعقيداً. ومع ذلك، فإن طرق مجال التحويل أكثر قوة ومتانة في مواجهة عمليات معالجة الصور الرقمية المختلفة. علاوة على ذلك، فإن طرق مجال التحويل أكثر أمانًا وأقل قابلية للاكتشاف في قنوات الاتصال الغير آمنة. تعتمد طرق اخفاء المعلومات الحاليةٍ على خوارزميات معقدة تتضمن العديد من العمليات والمهام الحسابية التي تستغرق وقتًا طويلاً، وبالتالي فهي غير قادرة على تضمين البيانات السرية واستخراجها في الوقت الفعلي او الحقيقي. لذلك، فإن الاتجاه هو لتنفيذ طرق اخفاء المعلومات في الأجهزة أو العتاد لتسريع وقت معالجتها وبالتالي تحسين فعالية وكفاءة تقنيات اخفاء المعلومات. في العصر الحديث تُستخدم مصفوفة البوابات المنطقية القابلة للبرمجة (FPGA) في العديد من التطبيقات نظرًا للعديد من المميزات مثل انخفاض تكلفة التطوير والمرونة وإمكانية إعادة التكوين. ويلاحظ أن معظم تقنيات اخفاء المعلومات الحالية التي يتم إجراؤها على مصفوفة البوابات المنطقية القابلة للبرمجة تعتمد على المجال المكاني. في هذه البحث، تم عمل مسح ومراجعة شاملة للعديد من تقنيات اخفاء المعلومات الحالية والمنفذة على مصفوفة البوابات المنطقية القابلة للبرمجة. حيث يتضمن هذا دراسة عملياتها العامة ومتطلباتها وتقييم أدائها. ستساعد هذه المراجعة الباحثين في العثور على فجوات البحث المتعلقة بتقنيات إخفاء المعلومات والمستندة على مصفوفة البوابات المنطقية القابلة للبرمجة لمختلف التطبيقات التي تتطلب أن يكون التنفيذ في الوقت الفعلي.

الكلمات المفتاحية: إخفاء المعلومات، مصفوفة البوابات المنطقية القابلة للبرمجة، التنفيذ العتادي، البت الأقل أهمية (LSB)، تحويل المويجات المنفصلة.

https://doi.org/10.59222/ustjet.2.1.3

1. Introduction

Since the electronic communication is vulnerable to malicious interference and eavesdropping, the security and privacy issues are more important than ever [1-6]. In order to address the information security, several methods have been suggested in the field of security systems. These methods can be categorized into two main fields: information encryption and information hiding [7-13] as illustrated in Figure 1. In cryptography, the secret data is encoded and therefore the message's content is protected. However, during eavesdropping on the communication media, it will be clear that there is a secret communication [14-19].

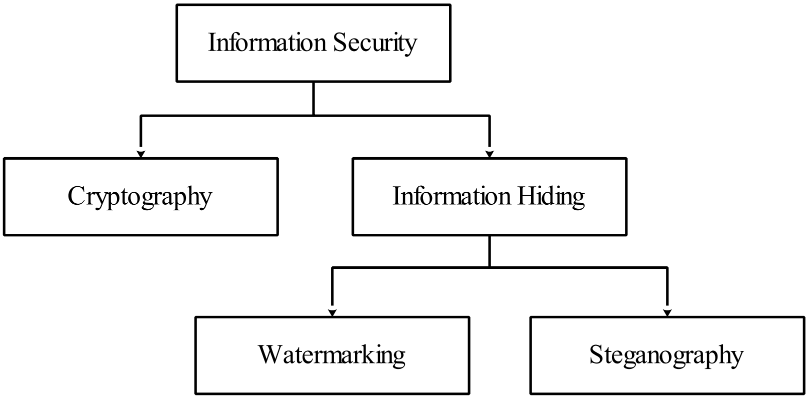

Figure 1: Information security techniques

However, the information hiding as a generic term embeds the secret data in another digital media in an undetectable manner. Hence, the secret data is imperceptible or its existence remains a secret [9, 10, 15, 20]. Steganography is a type of information hiding, which has cover media and secret data. The cover can be any popular digital media such as image, video, audio or text. The resulting embedded cover media is known as stego-media [9, 10, 14, 20-23]. As a result, many applications of steganography are being considered in industrial electronics, enabling the circulation of secret information for intelligence agencies, hidden personal information on smart cards, online voting, and invisible patient details for healthcare applications [7]. Although the secret data is invisible, the steganalysis aims to discover the existence of the secret data and tries to extract them [9, 10, 24, 25]. The image is the most common medium who is used in steganography because it contains a high frequency of redundant data [22, 23, 25, 26].

https://doi.org/10.59222/ustjet.2.1.3

Various types of image steganography methods have been proposed according to cover image domain, i.e. spatial domain and transform domain. Spatial domain techniques exploit the pixel intensity value of the cover image directly or indirectly to conceal the secret message bits. They are easy and simple way of data embedding. On the other hand, they have less robustness. This means that they are more affected by attacks such as compression, cropping, scaling and rotation [9, 10, 26-28].

In the transform domain, the cover image pixels are first converted to other forms, and then the secret data is embedded into the cover image coefficients. Most of the transform domain techniques are more robustness compared to spatial domain techniques. Thus, the transform-based systems are more effective in preserving the stego-image quality. They are also more secure and less detectable in an unsecured channel. However, the processes of embedding and extraction the secret data are more complex compared to the spatial domain [7, 10, 27-30].

The efficiency of a steganographic system is evaluated using various criteria such as imperceptibility, security, capacity of hiding information, robustness, etc. However, there is a trade-off between these factors, for instance; the increase of secret data leads to distort the stego-image. Therefore, all the criteria must be maintained at an optimum level [10, 31-34].

Recently, the steganography methods have been developed based on sophisticated algorithms that include several difficult computation steps [16, 35-40]. Most of the previous steganography algorithms are implemented in software platforms. They are often slow and incapable of real time applications such as ATM, online Stock trading, mobile wallets, and many other real-time electronic transactions applications must are supported by cryptographic and steganographic security systems. In addition, the computation complexity and hence the delay is increasing by the increase with the payload capacity. Thus, the trend is to implement the steganography methods in hardware to speed up the execution of algorithms and to improve the effectiveness of steganography techniques [41-44]. Garcia-Hernandez et al. [45] was able to speed up the processing time by 160x when implementing their developed method in hardware compared to the software implementation.

Few articles have been published on implementing the steganography algorithms on hardware platforms. However, the hardware implementation of steganography algorithms is characterized by many features, among which are portability, ability to connect to other systems, increased speed of processing, flexibility, low power usage, reliability, and reconfigurability [35, 46-49].

To implement steganography on hardware platforms, Digital Signal Processor (DSP), Application-Specific Integrated Circuit (ASIC) and Field Programmable Gate Array (FPGA) are used. Real-time performance is achieved by using DSP for the implementation of steganography algorithms. However, parallelism and parallel processing are not fully exploited. The implementation using ASIC has several advantages over FPGA such as decreased silicon area, decreased power consumption and improved performance. However, these facilities are obtained with higher cost and more time for fabrication [42, 48, 50-53]. Thus, FPGAs are more suitable to implement different steganography algorithms across different architectures. It provides high execution of steganography algorithms with parallelism [42, 53-55]. Furthermore, FPGA platform offers perfect shield against infectious attacks, lower development cost, greater flexibility and reconfigurability [29, 35, 42, 48, 50, 56, 57]. Reconfigurable architectures based on FPGA are used in various fields in modern era. Designing these architectures and producing results in real time is still a pressing problem [42, 50, 58, 59]. Figure 2 shows the steganography system using FPGA.

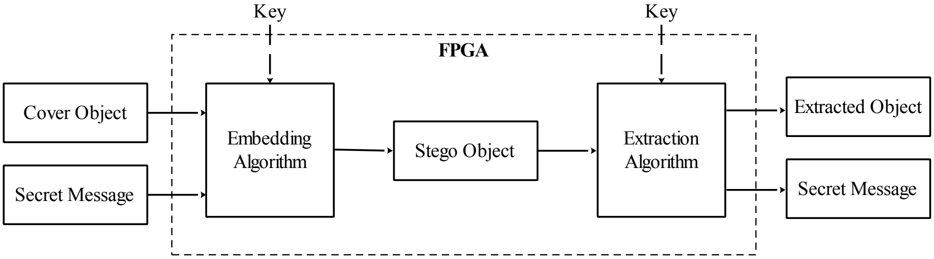

Figure 2: Steganography system using FPGA

The rest of this paper is organized as follows. Section 2 presents the performance evaluation techniques. Section 3 and Section 4 discuss the steganography implementations on FPGA based on spatial domain and transform domain, respectively. Section 5 provides the results comparison and discussion. Finally, Section 6 presents concluding remarks.

2. Performance Evaluation Metrics

Various metrics are used to evaluate different aspects of image steganography systems. The metrics can be classified into two categories. The first category is used to evaluate the image steganography algorithm itself. The second category is utilized to evaluate the efficiency of the steganography algorithm implementation on hardware. A trade-off between the different performance metrics should be taken into account during designing and implementing the steganography algorithms. The different performance metrics are described as follows.

2.1 Evaluation of Image Steganography Algorithms

The image steganography algorithms are evaluated using various metrics such as capacity of hiding information, visual quality, security, robustness, and computational complexity.

2.1.1 Embedding Capacity

An effective steganographic system is always aimed at transmitting maximum information using a minimum cover media. The embedding capacity or payload capacity is the number of secret data that is embedded (in bits) relative to the size of the cover image. It is normally expressed in bits per pixel (bpp). Embedding capacity is calculated using the Equation (1) [2, 9, 10, 60, 61].

$$\text{bpp} = \frac{\text{Number of secret bits that are embedded}}{\text{Number of pixels in the cover image}} \qquad (1)$$

2.1.2 Visual Quality

The secret data is embedded by any steganographic methods, which alter the visual quality of the cover image. It is not easily noticeable with the human eye. The changes have to be analyzed using standard measurement techniques. Maintaining higher payload capacity while maintaining visual quality and other evaluation parameters is a major challenge in steganography. There are several types of metrics to measure the visual quality of steganography (i.e. MSE, PSNR, and SSIM). The most used techniques are MSE and PSNR which stand for Mean Squared Error and Peak Signal to Noise Ratio, respectively.

MSE is calculated by performing pixel-by-pixel squared differences between the cover and stego-images. The MSE value should be as small as possible. A higher MSE value indicates a big difference between the cover image and the stego-image. MSE is measured using the Equation (2) [9, 10, 22, 62].

https://doi.org/10.59222/ustjet.2.1.3

$$\mathrm{MSE} = \frac{1}{\mathrm{M} \times \mathrm{N}} \sum_{x=1}^{\mathrm{M}} \sum_{y=1}^{\mathrm{N}} (\mathrm{C}_{xy} - \mathrm{S}_{xy})^2 \qquad (2)$$

Where x and y are the image coordinates, M and N represent the number of rows and columns in the cover image, and Cxy and Sxy are the pixels values of the cover image and stego-image, respectively, at (x,y) coordinate.

PSNR is one of the popular and high-level metrics, which used to measure the quality of the stego-image compared to a cover image. The high value of PSNR indicates better quality of stego-image. PSNR is calculated using Equation (3) [9, 10, 22, 62, 63].

$$\mathrm{PSNR(dB)} = 10 \times \log_{10}\left(\frac{\mathrm{Max}^2}{\mathrm{MSE}}\right) \qquad (3)$$

Where Max is a maximum pixel intensity value that is 255 in grey level with 8-bits. PSNR is measured in decibels.

2.1.3 Security/Undetectability

Attackers are attracted to detect or retrieve the presence of secret data from the stego-image. Therefore, while transmitting secret data through a communication channel, data access should be avoided by unauthorized persons. The steganography technique is regarded as secure if the secret data is not detectable by statistical means or steganalysis attacks. In addition, the secret data cannot be extracted after being detected by the attacker. The most famous steganalysis attacks are histogram analysis, bit plane analysis, Chi-Square, and RS analysis. In histogram analysis, the histograms of both cover image and stego-image are compared to determine the distribution of pixels or unusual shapes observed due to the embedding algorithm. Using bit plane analysis, a bit plane of the cover image is usually correlated with other neighboring bit planes. After applying the embedding algorithm, this correlation may change and then can be used as evidence of embedding secret data. On the other hand, the chi-square analysis is based on a statistical analysis of pairs of values that are exchanged during secret data embedding. The method was developed to detect embedding methods based on the Least Significant Bit (LSB). In the RS analysis, the method was developed to detect the hiding data based on the LSB embedding. The method utilizes the small alteration in LSB of pixels and the discrimination function to classify these pixels into regular and singular groups. Then using the frequency of these groups, the RS method estimates the length of the secret message hidden in the stego-image. Security or undetectability in steganography is considered a critical evaluation parameter [2, 9, 10].

2.1.4 Robustness

During the transmission of a stego-image through a communication channel, there is a possibility of image corruption by a third hand using image processing operations such as noise addition, sharpening, blurring, scaling, rotations, cropping, etc. The term robustness refers to the ability of a stego-image to maintain the secret data even if it is processed through various image-processing operations To measure the robustness of the steganography methods, different methods are used. However, the most popular methods are Bit Error Rate (BER) and Normalized Coefficient (NC). BER is used to measure the percentage of the number of different bits between the original secret message and the extracted secret message. The lower value of BER indicates better robustness of the steganography method. On the other hand, NC is used to measure the similarity between the original secret message and the extracted secret message. The higher value of NC indicates better robustness of the steganography method [9, 10, 64].

2.1.5 Computational Complexity

The term computational complexity in steganographic approaches is the amount of time, operations, space, etc. required to implement the steganography algorithms. It refers to the efficiency of steganography embedding and extraction algorithms. The low complexity of computation is considered as the ideal. In general, steganographic methods based on the spatial domain are often simple and computationally less expensive. Whereas most of the existing methods which are based on transform domain or artificial intelligence require extensive computational capacity [9].

2.2 Evaluation of Hardware Implementation

The most used metrics that are used to evaluate the efficiency of the steganography algorithms implemented on hardware platforms are throughput and efficiency.

Throughput is considered the basic performance metric to measure the performance of steganography implementation on FPGA. Throughput is expressed in pixels per second (pps) and is calculated using the Equation (4) [35, 48, 65].

$$\text{Throughput} = \frac{(\text{data input}) \times (\text{max. frequency})}{\text{clock cycles}} \qquad (4)$$

Efficiency is the other basic performance metric, which is used to measure the efficiency of implementation regarding the use of hardware. Efficiency is defined as the ratio of the throughput to the area. It is computed as in Equation (5) [35, 48, 65].

$$\text{Efficiency} = \frac{\text{Throughput}}{\text{Area}} \qquad (5)$$

3. Steganography Implementation on FPGA based on Spatial Domain

Spatial domain techniques are more desirable for hardware implementation using FPGA due to their features such as high embedding capacity, high imperceptibility, and lower computational complexity. So, most of the proposed steganography algorithms which have been implemented in the FPGA platform are based on the spatial domain [48, 54, 62, 66]. Most of FPGA implementations have been implemented with the help of Altera or Xilinx FPGAs [42, 54]. These techniques can be categorized as follows:

3.1 LSB Replacement

In LSB replacement, the n-bits of the secret data are altered by n-LSBs of image pixels. The LSB based steganography provides the simplest way of embedding secret data into the LSBs of image pixel values with high embedding capacity [42, 67]. However, the LSB technique is fragile against attacks like low pass filtering and compression [22].

The study [56] has used Xilinx ISE to perform the LSB steganography algorithm using Matlab software and FPGA-based hardware. The RGB color image is used as a cover image. The performance speed of the steganography algorithm in the two implementations is compared. The speed of the embedding and extraction process on Xilinx was much faster. Where the embedding and the extraction process in Matlab were 0.174 and 0.059 sec, respectively, the embedding and the extraction process in Xilinx ISE were 250 ns each of them.

In [68], two image steganography algorithms based on generalized chaotic maps are presented. One chaotic map is used in the first algorithm to index the one-dimensional memory. In the second algorithm, two maps were used to index the two-dimensional memory. Generalized chaotic maps are used to

https://doi.org/10.59222/ustjet.2.1.3

randomize messages during the embedding process. Thus, the security level of the proposed algorithms is improved. Various performance measures are used to compare both methods. The hardware designs for the two algorithms were implemented on Xilinx Virtex 7 FPGA board with 350 MHz frequency.

In study [69], data security is enhanced by a combination of cryptography and steganography. Firstly, the data is encrypted with a key, and this key is generated using the Diffie–Hellman key exchange algorithm. Then, the encrypted data is hidden in the image file using the LSB steganography method. The data hiding capacity is improved by increasing more than one LSB of the image. The quality of the stego image is measured by calculating the PSNR and MSE. The embedding and the extraction modules were implemented using the Xilinx Vivado tool, using Verilog code for the proposed Architecture and MATLAB for image characterization and key generation.

In [70], the data is embedded and extracted with and without pipelining technique. The proposed algorithm is realized in FPGA using the Xilinx Virtex-V device. The data hiding method of 4 bits is based on LSB steganography. The performance comparison for a pipelined and non-pipelined mode of data is performed for delay and memory usage parameters. The data embedding and extraction using pipelining mode produce better results than in non-pipelined mode in terms of processing time. This makes it more acceptable for real-time processing.

Mohd et al. [35] have implemented several steganography methods using Altera and Xilinx FPGAs. The methods are i-bit LSB, mix-bit LSB, random i-bit LSB, and texture-based algorithms. The steganography methods are evaluated using different performance metrics. The normalized data shows that Altera FPGA (Cyclone II) employs 31% more LUT than Xilinx FPGA (Virtex-6). The power results of Altera are on average twice the power results of Xilinx; this is due to the difference in platform technology. The designs with minimum resource utilization are 4-bit LSB (for Altera) and 2-bit LSB (in Xilinx). The 4-bit LSB and random 4-bit LSB have superior throughput and energy results. For good image quality, the best are mix_332 and 2-bit LSB. The study shows that the mix_332 consistently provides good results across all metrics.

Shete et al. [22] have introduced a steganography system utilizing the LSB replacement approach. The developed method has been implemented in both Matlab and FPGA device. LSB algorithm replaces every 2-bits of secret data by 2-LSBs of cover image pixels. The proposed system uses the Xilinx Spartan 3A for hardware implementation, which provides a faster response in term of processing time compared with Matlab. The proposed FPGA design takes 5.4 ns, while Matlab takes about 8.6 sec to perform the same operations. The performance of the LSB algorithm is evaluated based on PSNR, MSE, Bit Error Rate (BER), and processing time. The 2-LSBs technique showed adequate results in terms of PSNR, MSE, and BER.

In [42], high-speed reconfigurable architectures for LSB/multi-bit based image steganography algorithm adapted FPGAs/ASICs implementation is designed. The proposed system gives a competent throughput. This is because of integrating high pipeline and parallel operations at the module level. The architectures are realized in the Xilinx Virtex-II Pro FPGA device and work at a rate of 183.48 frames/second in real-time. The proposed algorithm is tested by varying embedding bit size and cover image resolution, which remain in good quality. Figure 3 shows the embedding and extraction methods suitable for FPGA implementation.

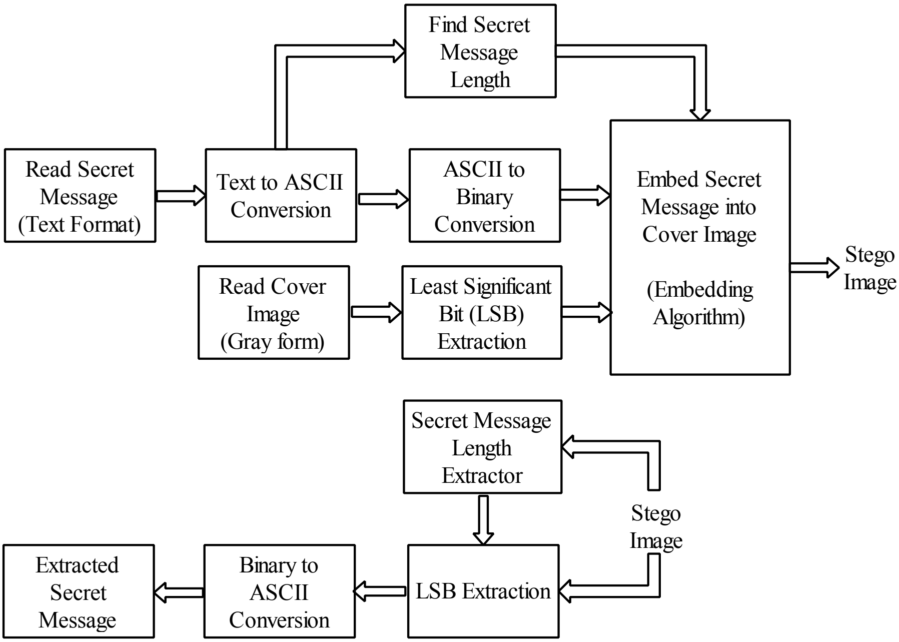
Find Secret Message Length
Read Secret Message (Text Format)
Text to ASCII Conversion
ASCII to Binary Conversion
Embed Secret Message into Cover Image (Embedding Algorithm)
Stego Image
Read Cover Image (Gray form)
Least Significant Bit (LSB) Extraction
Secret Message Length Extractor
Stego Image
Extracted Secret Message
Binary to ASCII Conversion
LSB Extraction

Figure 3: Embedding and extraction methods suitable for FPGA implementation for [42]

Farouk et al. [71] have proposed a secret key steganographic micro-architecture using FPGA. The secret message is dispersed on the cover in a specific way that is based on a secret key known only to the sender and receiver. The secret key is used to increase obscurity. Hence, if the warden detects the existence of the message, he cannot retrieve it. The micro-architecture approach provides an acceptable degree of data hiding with minimal cover distortion.

Farouk et al. [72] have introduced an improvement to micro-architecture FPGA implementation. Hybrid Hiding Encryption Algorithm (HHEA) is suggested. The proposed algorithm is based on steganography and cryptography for packet-level encryption. This design overcomes the limitations of pre-designed micro-architecture, the non-exploitation of the ability of parallel bit replacement, and the serial input plaintext encryption. The modified micro-architecture comes with five basic modules. These modules are; the message cache, the message alignment module, the key cache, the comparator, and the encryption module.

In [73], a stego architecture based on Different Linear Feedback Shift Registers (LFSRs) is proposed. Therefore, different LFSR arrangements using different polynomial expressions have been implemented for hiding the secret data. The LFSRs are used as random addresses generators to conceal the secret data in the cover image pixels. LFSRs are a good approach for an ideal security key. For the implementation of stego architecture, Altera Cyclone II FPGA is used.

Huang et al. [74] have suggested an FPGA implementation model for the Optimal Pixel Adjustment Process (OPAP) algorithm of image steganography. In the proposed scheme, the first k-bits of the secret message are embedded into each cover image pixel by the LSB substitution method, followed by executing associated OPAP calculations to reconstruct a stego pixel. Xilinx Spartan 3E FPGA is used to develop the proposed algorithm, programmed with Verilog language. The hardware implementation provides faster execution time compared to software implementation, which reaches to 15 times.

In [75], two effective techniques for designing an image based steganography system on the chip are presented. The algorithms are LSB insertion and watermarking. In the LSB algorithm, the LSBs of the cover image pixels are replaced by the bitstream of the text to be hidden with the aid of a key to provide greater protection. In the watermarking algorithm, RGB image is converted into YCbCr image for embedding data. The luminance component is (Y) represents the brightness of the image, while the (Cb and Cr) are chrominance components, which store color difference information. Thus, the secret message will be hidden in the Cb and Cr components of the image. The Altera FPGA and Quartus II design software are used to perform the proposed techniques. The watermarking algorithm produces lower PSNR and is more complicated than LSB insertion. However, the data is well protected. Therefore, the LSB insertion is more suitable for FPGA implementation.

Elshazly et al. [76] have proposed an algorithm that embeds data on each component of color image. The sender's signature and the length of the secret message are hidden in Red component, while the secret message is hidden in the Green and Blue components. Four different cases of LSB are considered; LSB-1, LSB-2, LSB-3 and LSB-4. The proposed algorithms are executed using MATLAB and implemented on FPGA using Xilinx system generator (XSG) based on Hardware/Software Co-simulation. In the proposed algorithm, a large secret text (up to 98,304 characters in a color cover image of size 512x512 pixels) can be embedded while maintains better PSNR. Figure 4 shows the embedding algorithm.

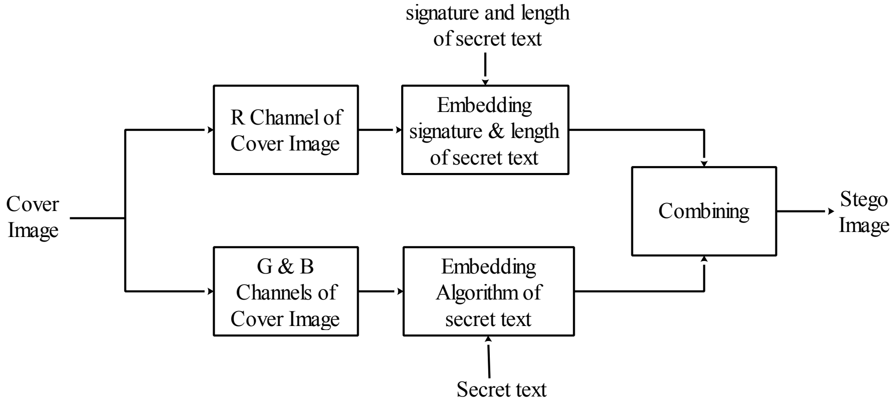
signature and length of secret text
R Channel of Cover Image
Embedding signature & length of secret text
Cover Image
Combining
Stego Image
G & B Channels of Cover Image
Embedding Algorithm of secret text
Secret text

Figure 4: The proposed embedding algorithm for [76]

Mohd et al. [66] have introduced a hardware design of 2/3 LSB steganography technique in a cyclone II FPGA of the Altera family. The 2/3-LSB technique provides good image quality, which is between 2-bits and 3-bits LSB. The proposed design has used the built-in Nios processor in addition to specialized logic to perform the steganography steps. The proposed design has simplified the access to the memory since it assigns one secret byte to one pixel of the cover image. Accessing and processing data at the bytes boundary simplifies hardware design and reduces design area and power.

In [67], a steganography system to protect the secret message using two levels of security is suggested. At the first level, the data is encrypted by an encryption method called Character Bit Shuffler (CBS). Then, the encrypted data is concealed inside an image using the LSB technique. The suggested method uses two channels of RGB image to hide the secret data, which simplifies the embedding process and gives better image imperceptibility. The proposed system is implemented on FPGA board and is programmed using the Verilog Hardware Description Language (HDL).

Amirtharajan et al. [77] have used 2-D image processing to design and analyze prototype hardware to perform secret sharing. Hardware modeling for this process can provide portability and speed improvement. The proposed work is implemented on FPGA with LSB embedding and recovery modules. The chip is designed to embed the given data in an image automatically. In addition, the recovery module is designed to retrieve the embedded data from the cover image. Timing analysis shows that the hardware can execute the secret sharing about 740 times faster than the software architecture.

In [78], the steganography and cryptography are combined to improve the security of confidential data and imperceptibility of the stego object. The secret message is encrypted with two keys using the RC6 algorithm. After that, the cover object is broken into miniature constructs of similar size. The encrypted message is embedded in each block using eight scan patterns. For the final embedding, one random scan pattern path is selected that achieve high imperceptibility. The proposed algorithm is implemented on a Cyclone II FPGA. The FPGA implementation has reduced the computational complexity of the algorithm resulting in higher throughput. It produces a throughput of 441.38 Mbps for data embedding, and 914.29 Mbps for data extraction.

3.2 LSB Matching

In LSB Matching (LSBM) method, the matching operation is performed between the bits of the secret message and the cover image pixels based on that, 1 is added or subtracted to the values of cover image pixels randomly. Clearly, LSBM prevents asymmetry, which presents in LSB replacement; therefore, it is more difficult to detect [79, 80].

Mahmood et al. [79] have presented an implementation of color image steganographic system on FPGA based on the merge between the random pixel manipulation and the LSB matching. After converting the RGB color space to the YCrCb color space, the secret message is embedded into Cr and Cb, which are less sensitive regions according to the human visual system. Different methods of LFSR have been used to generate random embed addresses. The software implementation runs on average in 2.67 s per image, whereas the hardware architecture achieves an average of 4.18 ms per image.

Nandhini et al. [80] have proposed an architecture for LSB Matching Revisited (LSBMR) steganography algorithm with and without pipelining technique. The LSBMR reduces the Expected Number of Modifications Per Pixel (ENMPP) for the cover image to 0.37. The proposed architecture decreases the computational delay of the LSBMR method. The result is analyzed for both the software and hardware level. Verilog code is implemented for the proposed model using Xilinx Spartan 6 system. In the proposed method, the secret message cannot be fully retrieved. However, the PSNR of the extracted secret image reaches to 48.1 dB.

Pathak and Bansal [62] have presented an implementation for a one-third probability embedding algorithm on FPGA based hardware. The one-third probability algorithm reduces the probability of change per pixel to 1/3. Thus, it shows a better imperceptibility and better resistance to numerous steganalysis attacks. The work presented optimizes many of the operations and elements from the original one-third probability algorithm respect to hardware implementation. The proposed work is implemented on Xilinx Spartan 6 FPGA.

3.3 Multiple Bit-Planes

The main principle of the Bit-Plane Complexity Segmentation (BPCS) technique is that the cover image is divided into informative regions and noise-like regions. The secret data is embedded into the noise-like regions in the bit planes of the cover image without deterioration. Kait and Chauhan [81] have presented a hardware implementation of the BPCS steganography technique in the Xilinx Spartan 3E FPGA family. The embedding process in software platform requires intensive computation, so this technique is applied in FPGA to improve the processing speed. With this technique, 50% to 60% of secrete data can be concealed in the cover image.

In work [41], the hiding information using Sobel edge detection is presented. The fact that the alteration at edges is not well differentiated is used. So, edges can hide more data without losing image quality. The algorithms are implemented using MATLAB for software and VHDL for hardware design. The algorithms are tested and verified on hardware using a Xilinx Spartan 3E. The algorithms are implemented at 8-bit grayscale image data of size 256×256. The results obtained by implementing the algorithms using VHDL were better than those obtained by MATLAB. A comparison between the two methods shows that the two images have a high value of PSNR, which ranges between (72.1 to 82.8).

3.4 Difference Expansion/ Texture

Context technique takes advantage of noisy regions, and those with abrupt gray levels change in an image. Thus, it is difficult to detect hidden information. The process of locating that region is highly repetitive and costly in terms of computation. Gómez-Hernández et al. [82] have introduced a hardware architecture for context steganographic technique in FPGA of the Altera. The technique is implemented on FPGA to increase the processing speed. The hardware architecture is 252x faster than software implementation. The implementation results show a throughput of 61.5 Mbps.

In [46], a simple algorithm for hiding data with a low-complexity steganography system in images as a host is proposed. A designed threshold value is introduced to adjust the PSNR of the stego image. When ALF is set to 1, the value in the segment is consistent, allowing data hiding on the cover image with approximately no loss. When there is a certain demand for hiding capacity, ALF can be adjusted to have a better PSNR of the stego image. When the cover image is flat, a larger ALF and bigger segmentation length can obtain greater hiding capacity. The system has been implemented in Xilinx›s FPGA, which can save resources and achieve real-time data processing. The results show that the algorithm takes up fewer resources, has low power consumption, and is easy to implement.

Timarchi et al. [83] have introduced a hardware implementation of a Modified ConText (MCT) algorithm. The MCT algorithm uses a threshold value for deciding the embedding process. The threshold level is used to compare pixels differences in each sub-block. The using of the threshold level leads to faster and power-efficient implementation. The proposed architecture is implemented on a Spartan-3 FPGA device.

3.5 Exploiting Modification Direction (EMD)

EMD method converts the secret data from binary into secret digits in (2n+1)-ary notational system. Then, each secret digit is carried by ‘n’ cover pixels. It increments or decrements at most one of the n cover image pixels by 1. EMD algorithm has high embedding efficiency, low distortion, and good security. Elshazly et al. [84] have proposed a developed Generalized EMD (GEMD) image steganography algorithm to hide high payload capacity while maintaining the stego-image quality. An improved Pixel Segmentation Strategy with an Indicator Bit (PSS-IB) is used to overcome the defects of previous EMD algorithms. Suggested algorithms are simulated using MATLAB and implemented by FPGA with Xilinx System Generator (XSG) on Spartan 3E Kit. The proposed algorithm has a capacity of embedding (up to 2,359,296 bits in a cover image of size 512x512) and maintains the stego-image quality (up to 50.15 dB).

Shet et al. [52] have presented a high-speed architectures design and FPGA implementation of EMD algorithms, namely, Basic EMD (BEMD), Modulus EMD (MEMD), and Fully EMD (FEMD) for image steganography. The pipelining and parallel processing are exploited to improve the operation speed for each module. The design is implemented on the Xilinx FPGA device. The proposed framework reaches to 549 fps, which can be considered suitable for real-time operation. The results show that the PSNR of the BEMD algorithm is the highest compared with FEMD and MEMD algorithms.

3.6 Interpolation-Error

Premalatha and Amsaveni [85] have presented an implementation of a low-complexity steganography system with a digital image as a host signal. In this method, the error values of the interpolation are calculated. Then, additive expansion is applied to interpolation errors during the embedding process. The embedding and extracting processes are implemented using Simulink blocks. Thus, the Simulink blocks are converted to VHDL and then implemented on the FPGA.

Laces and Garcia-Hernandez [65] have proposed an FPGA hardware architecture of low complexity steganography system using additive interpolation-error expansion. The proposed system is implemented on three different integration technologies of Xilinx FPGAs. The FPGA families, which are Virtex 4 (90 nm), Virtex 5 (65 nm), and Virtex 6 (40 nm). A comparison between these families is presented. It is found that using Virtex 6 (40nm) technology utilizes fewer hardware resources and high clock frequency operation compared with larger integration technologies. However, the power consumption is higher. The proposed hardware system is about 64x faster than software implementation. In addition, it is capable of processing about 2.8 channels of 4K video in real-time. Figure 5 explains the embedding and extraction processes.

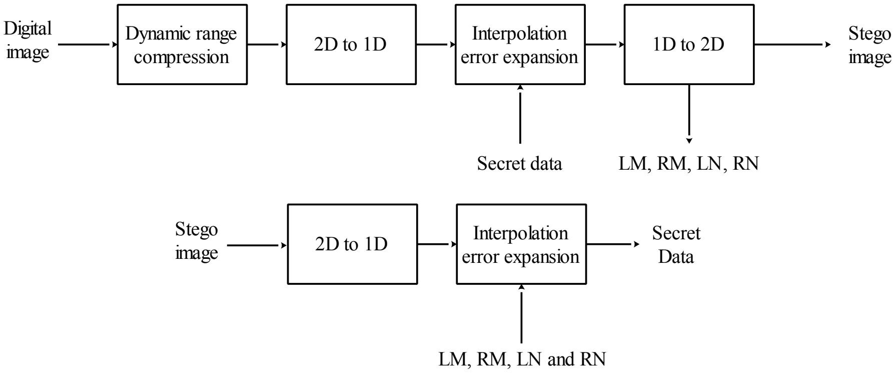

Figure 5: Embedding and extraction processes for [65]

Sakthivel and Sankar [86] have proposed an FPGA implementation of data hiding and extraction algorithm in real-time on the Altera Cyclone II device family. The proposed algorithm has used a neighbor mean image interpolation technique. In this method, the values of the neighbor pixel are used to calculate the mean values. Then, the calculated mean values are inserted at the position of pixels whose values are not yet allocated. The stego-image has good visual quality with high resolution at the cost of high complexity. The concealed data can be retrieved by inverting the process of interpolation and embedding.

4. Steganography Implementation on FPGA based on Transform Domain

The steganography techniques based on the transform domain give better robustness and security compared with the spatial domain techniques [22, 66]. On the other hand, they lack a high embedding capacity and require complex repeated operations that are computationally expensive [62, 66].

4.1 Discrete Wavelet Transform (DWT)

Shete et al. [22] have implemented a steganography system based on DWT using Matlab and FPGA device. The cover image is transformed into the

https://doi.org/10.59222/ustjet.2.1.3

frequency domain. Then, the secret image is embedded in the transform coefficients of the cover image. Finally, the inverse DWT is performed to get stego-image. The proposed system uses the Xilinx Spartan 3A for hardware implementation. It provides a faster response in terms of processing time compared to Matlab. The performance is evaluated based on PSNR, MSE, BER, and processing time. The DWT method offers high PSNR and low MSE and BER. The proposed FPGA design requires only 5.4 ns to obtain stego-image while the Matlab implementation requires 0.7 sec. Figure 6 illustrates the block diagram for the steganography system.

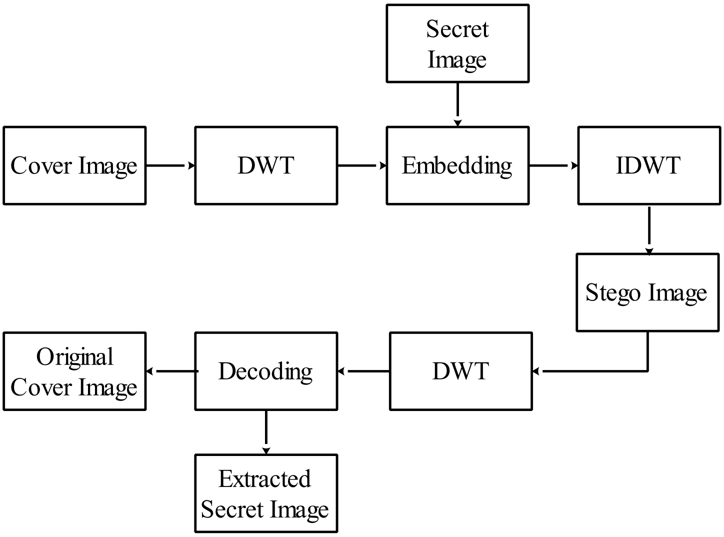

Figure 6: The block diagram system for [22]

In [87], an FPGA implementation of DWT based approach for image steganography is presented. The proposed design uses only the LL band of both cover and secret images for encoding and decoding purposes. The proposed technique takes 4-bit MSBs of a binary secret image pixel and merges with 4-bit MSBs of a cover image pixel. Consequently, the design requirement of memory is less for hardware implementation. Therefore, this results to increase the operating frequency of the architecture. The hidden process is carried out with a pre-shared key, which is used to generate the pixel position for encoding, which provides better security. The proposed design is implemented on the FPGA board of Spartan 6 and is coded using

https://doi.org/10.59222/ustjet.2.1.3

VHDL language. The proposed approach provides good PSNR of stego-image. However, the secret message cannot be fully retrieved. The PSNR of extracted secret message in the range of 65.4-71.8 dB.

In [50], a steganography model is designed using lifting-based DWT, secret image encryption, and a modified LSB approach. The proposed model is designed and implemented on Artix-7 FPGA. It operates with high speed at an average maximum frequency of 126.5 MHz to meet real-time requirements. The performance of the steganography model is evaluated using PSNR and MSE. The proposed model is robust since the results obtained by introducing noise show a small difference in PSNR of around 0.550.

Mohd et al. [88] have presented an algorithm for embedding and extracting a secret message in a transform domain based on Haar wavelet without any loss of secret information. This is accomplished by special clipping mechanism, as well as by modifying the positioning of the secret bit in the transform coefficients. The embedding algorithm begins with performing 2D DWT on the cover image. Next, a single secret bit is embedded in each LSB of the cover image coefficient. The final step is to generate the stego-image by perform IDWT on the stego-image DWT coefficients. The algorithm is implemented on FPGA. The embedding time is 14.446 ms and the extraction time is 5.291 ms.

Hardware implementation of the steganographic algorithm is proposed in [89], which is based on the DWT method and AES encryption. The proposed scheme uses double security, which makes it difficult to identify and decrypt. The parallel implementation has also improved the process of large amounts of data at a frequency of 0.162 Mbps. The proposed implementation improves the efficiency of hardware resources, allowing practical to implement it in real systems.

4.2 Integer Wavelet Transform (IWT)

IWT produces integer transform coefficients. It has lower computational complexity than DWT. Ramalingam et al. [90] have proposed a reconfigurable hardware architecture for implementing an adaptive random image steganography algorithm based on IWT. Haar-IWT has been used to separate each 8×8 pixel blocks of the cover image into sub-bands namely, LL, LH, HL, and HH. Then, the secret data is randomly concealed in LH, HL, and HH blocks via Moore and Hilbert Space-Filling Curve (SFC) scan patterns. Thus, one scan pattern is chosen for the final embedding based

https://doi.org/10.59222/ustjet.2.1.3

on the best results of MSE and PSNR. Parallelism is applied to overcome the computational overhead of IWT. The proposed system uses the Cyclone II FPGA for hardware implementation. It takes 1.6 μs to embed data into blocks coefficients. Figure 7 shows an adaptive IWT block diagram.

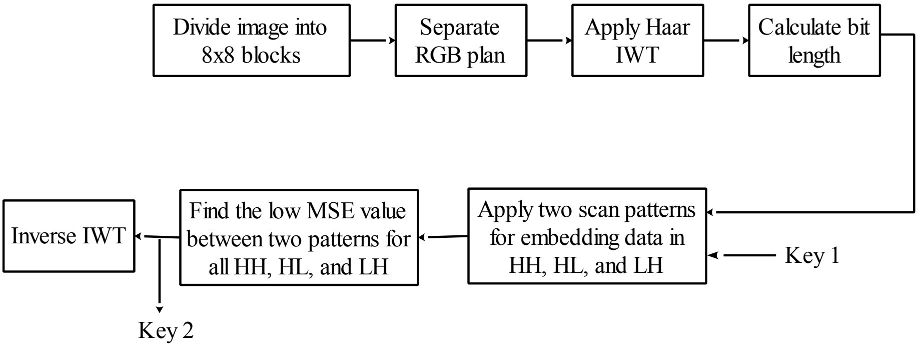
Divide image into 8x8 blocks
Separate RGB plan
Apply Haar IWT
Calculate bit length
Apply two scan patterns for embedding data in HH, HL, and LH
Key 1
Find the low MSE value between two patterns for all HH, HL, and LH
Inverse IWT
Key 2

Figure 7: Adaptive IWT block diagram for [90]

4.3 Discrete Cosine Transform (DCT)

The DCT transforms an image into corresponding elementary frequency components. The image pixels are expressed as a sum of sinusoids of different amplitudes and frequencies. DCT is commonly used in JPEG image compression. In [91], a new architecture is presented to perform simultaneous compression and encryption. The image is first divided into blocks of 8x8 pixels. Then, the modified DCT is computed separately for each block. The modified DCT is an optimized model, which uses just 4 multipliers and 5 adders, and has a throughput of 8 pixels per clock. The proposed design is implemented on Spartan-3 FPGA and is coded in Verilog HDL.

The study [47] has proposed FPGA-based image quality access control hardware in the DCT compressed domain. Serial and parallel hardware implementation are made and comparison details are provided. The proposed architecture is synthesized for FPGA using Xilinx's ISE and tested on a large number of benchmark images. It can be seen that for an image of size (512×512), the parallel implementation for the access control encoder and decoder achieves a high throughput of 11.37 and 11.41 Mb/s, respectively, at the maximum operating frequency of 111.03 MHz.

5. Results Comparison and Discussion

As seen from the previous sections, there are various hardware implementations of image steganography methods based on the FPGA platform. The algorithms have been categorized according to the cover image domain, i.e. spatial domain and transform domain. The algorithms are analyzed in terms of their characteristics, FPGA used, resource utilization, maximum operating frequency, speedup, throughput, efficiency, PSNR and Bpp. It can be noted from Table 1 that the spatial domain techniques are easy and simple for data embedding in software and hardware. However, they are limited in security and less robust against various attacks such as compression, noise cropping and rotating. On the other hand, the transform domain techniques are more robust and secure. In addition, they offer multiple planes of cover image and this improves the choice of the effective embedding locations. However, the processes of embedding and extraction the secret data are more complex and time-consuming compared to that for the spatial domain. Though, the cost of the computation complexity is not a serious problem anymore, since the hardware implementations of the transform-based steganography techniques using FPGA significantly improve the processing time efficiently.

Table 1: Comparison of characteristics of spatial domain and transform domain [9, 48]

Characteristics	Spatial Domain	Transform Domain
- System type	Simple	Complex
- Pixel Manipulation	Direct (In cover image pixels directly)	Indirect (In transformed coefficients)
- Computational Complexity	Less computation time	High computational time
- Embedding Capacity (Payload)	High	Limited
- Visual Quality (Imperceptibility)	High	Medium
- Robust (Compression, Noise Cropping, Rotating, etc.)	Weak (Highly prone)	(Strong) Less prone
- Security/ Undetectability (Steganalysis Attacks)	Low (Vulnerable to steganalysis attacks)	High (Resistant to steganalysis attacks)
- Hardware Implementation	Easy	Complex

https://doi.org/10.59222/ustjet.2.1.3

Table 1: continued

Characteristics	Spatial Domain	Transform Domain
- Hardware Resource Utilization	Low resource utilization	High resource utilization
- Hardware Speed	Fast	Medium

The comparison between FPGAs family, usage of Logic Elements, and a maximum frequency of steganography algorithms implemented in FPGA are demonstrated in Table 2. It can be noted that most FPGA implementations have been implemented with the help of Altera or Xilinx FPGAs. In addition, Table 2 shows that the steganography methods based on transform domain as in [90], [88], [89], and [87] utilize more Logic Elements (LEs) than the steganography methods based on spatial domain due to the computational complexity.

Table 2: Comparisons of FPGA family, no. of slices, no. of slice flip flops, no. of 4 input LUTs, no. of bonded IOBs, LEs, and the maximum frequency for various designs

References	FPGA Family	No. of Slices	No. of Slice Flip Flops	No. of 4 input LUTs	No. of bonded IOBs	LEs	Max. Freq. (MHz)
[22]	Xilinx (Spartan 3A-XC3S50A)	42	64	64	5	-	183.76
[42]	Xilinx (Virtex II Pro XC2V500FG256-6)	-	-	-	-	-	144.3
[73]	Altera (Cyclone II EP2C20F484C7)	-	-	-	-	340	100
[74]	Xilinx (Spartan 3E)	270	239	340	4	-	-
[67]	-	-	-	-	-	451	-
[79]	Xilinx (Virtex-II XC2VP30)	2,411	-	-	-	-	107.75
[80] (With Pipelining: Embedding, Extraction)	Xilinx (SPARTAN-6)	-	-	-	-	-	340.38, 271.30
[62]	Xilinx (Spartan 6 series XC6SLX45T)	160	214	-	171	-	104.97
[83]	Xilinx (Spartan 3)	64	45	50	-	-	167
[84]	Xilinx (Spartan 3E)	108	96	148	103	-	-

https://doi.org/10.59222/ustjet.2.1.3

Table 2: continued

References	FPGA Family	No. of Slices	No. of Slice Flip Flops	No. of 4 input LUTs	No. of bonded IOBs	LEs	Max. Freq. (MHz)
[52]	Xilinx (Artix-7 XC7A75TCSG324–1)	-	-	-	-	-	144
[65] (Integration technology: 90 nm, 65 nm, 40 nm)	Xilinx (Virtex 4, Virtex 5, Virtex 6)	-	-	-	-	-	84.2, 96.4, 100.4
[87]	Xilinx (Spartan 6 XC6SLX45-3csg324)	514	297	2108	-	-	153.31
[90]	Altera (Cyclone II EP2C35F672C6)	-	-	-	-	11222	-
[71]	Xilinx (Spartan2 x2s100)	1195	-	-	43	-	20.49
[72]	Xilinx (Spartan2 x2s100)	337	205	393	57	-	23.88
[82]	Altera (Cyclone II EP2C35F672C6)	-	-	-	-	204	106.75
[86]	Altera (Cyclone II EP2C5AF256A7)	-	-	-	-	108	58.48
[88]	Altera (Cyclone II)	-	-	-	-	1572	72.2
[78] (Embedding, Extraction)	Altera (Cyclone II)	-	-	-	-	10513, 1975	50
[68]	Xilinx Virtex 7 VC707	-	5,023	2,552	-	-	350
[46]	Xilinx xc6slx100-3fgg676	660	1004	627	4	-	-
[50]	ARTIX-7	352	179	1272	-	-	127.915
[89]	Spartan6 XA6SLX100	867	-	1853	-	-	40.54

Hardware architectures proposed in [22], [74], [79], [82], [77], and [86] for steganography implementation on FPGA claim to be several times faster than the software architecture. It can be noted that the speedup varies from steganography system to another. It starts from 8 and reaches to several thousand. This indicates to the efficiency of the steganography system implementation. Table 3 shows the comparison between the processing time for the software and FPGA implementations.

https://doi.org/10.59222/ustjet.2.1.3

Table 3: Comparison of processing time for the software and FPGA implementations

References	Timing required		Ratio (Speedup)
	Software (sec)	Hardware	
[22] (2-LSB, DWT)	8.6, 0.78	5.4 ns	-
[73] (Embedding)	-	1.96 ms	-
[74] (4-LSBs, 3-LSBs, 2-LSBs, 1-LSB)	0.199567, 0.143214, 0.120556, 0.116042	0.013110 sec	15.23, 10.92, 9.20, 8.85
[79]	2.6727	0.00418 sec	639
[84]	-	2.62 ms	-
[65] (Embedding, Extraction) (Integration technology: 90 nm, 65 nm, 40 nm)	-	1.07, 0.936, 0.868 ms 1.02, 0.897, 0.832 ms	64
[90] (Embedding)	-	1.6 μs	-
[82]	8.089	0.0325 sec	252
[77] (4-LSBs, 3-LSBs, 2-LSBs, 1-LSB)	8.486, 8.522, 8.465, 8.337	0.0115 sec	740
[86]	23.010351	896 μs	25681.195
[88] (Embedding, Extraction)	-	14.446, 5.291 ms	-
[78] (Embedding, Extraction)	-	3.48, 1.68 μs	-
[56] (Encryption, Decryption)	0.174, 0.059	250 ns, 250 ns	696000, 236000
[70] (Encryption, Decryption)	-	2.48ns, 3.019ns	-

Throughput is also an important comparison parameter to measure the performance of steganography implementation on FPGA. The high throughput is better, which indicates the quality of the algorithm design. In addition, the throughput varies according to the integration technology that is used in hardware designing. Table 4 compares the steganography systems based on throughput and efficiency.

Table 4: Comparison of Throughput and Efficiency

References	Throughput achieved	Efficiency
[79]	3 Mbps	-
[65] (Integration technology: 90 nm, 65 nm, 40 nm)	641.26, 734.17, 764.63 Mpps	42.069 x 10^3, 75.57 x 10^3, 89.284 x 10^3
[91]	8 Mbps	
[72]	95.532 Mbps	
[82]	61.54 Mbps	-
[78] (Embedding, Extraction)	441.38, 914.29 Mbps	0.569
[68]	4.09 Mbps	-
[89]	162.9 Kbps	-

PSNR reflects the visual quality of the stego-image compared with the cover image. The high value of the PSNR indicates better quality of the stego-image. The embedding capacity is the number of secret data that is embedded (in bits) into the cover image. It is expressed in Bits Per Pixel (Bpp). Maintaining high embedding capacity while preserving image visual quality is considered a major challenge in steganography. Table 5 compares the value of the PSNR and the Bpp of some algorithms implemented in hardware.

Table 5: Comparison of PSNR and embedding capacity in Bpp

References	PSNR (dB)	Bpp
[35] (Mixed 3221 LSB, 2-bit LSB, 4-bit LSB, Mixed 332 LSB)	40.95, 44.11, 31.76, 39.13	2, 2, 4, 2.67
[22] (Cameraman: 2-LSB, DWT)	45.40, 54.60	2
[42]	77.84 to 84.46	1
[73]	51	1
[74] (Lena: 4-LSBs, 3-LSBs, 2-LSBs, 1-LSB)	34.81, 40.73, 46.37, 51.14	4, 3, 2, 1
[76] (Lena: 4-LSBs, 3-LSBs, 2-LSBs, 1-LSB)	37.43, 43.53, 49.57, 55.49	4, 3, 2, 1
[66] (Lena)	40.1	8 (color)
[67] (Lena)	54.14	2 (color)

https://doi.org/10.59222/ustjet.2.1.3

Table 5: continued

References	PSNR (dB)	Bpp
[79] (Babbon)	51.3	2 (color)
[80] (Lena)	48.13	0.63
[62] (Lena)	52.94	1
[84] (Lena)	50.15, 47.57, 43.87, 41.37	4.5, 6, 7.5, 9 (color)
[52] (Woman: MEMD, FEMD, BEMD)	45.11, 49.89, 52.11	-
[85] (Lena: Interpolation, LSB)	44.03, 54.52	1
[87] (Lena)	75.25	-
[90] (Lena)	62.02, 58.16, 53.43	1, 2, 3 (color)
[91]	55.41	-
[71]	-	1
[77] (Lena: 4-LSBs, 3-LSBs, 2-LSBs, 1-LSB)	32.07, 35.72, 44.02, 51.13	4, 3, 2, 1
[86]	51.33	0.63
[88]	43.6	2.97 (color)
[78] (Lena: 4-LSBs, 3-LSBs, 2-LSBs, 1-LSB)	35.35, 41.12, 47.71, 53.48	4, 3, 2, 1
[70] (Lena)	35.6168	4
[68] (Lena)	39.0872	2
[69] (Cameraman: 4-LSBs, 3-LSBs, 2-LSBs, 1-LSB)	35.75, 42.17, 47.67, 54.4	4, 3, 2, 1
[46] (AVG)	42.99	2.12

6. Conclusion

This paper attempts to present a comprehensive review of various current methods of image steganography that are implemented on FPGAs, including general operation, requirements, and performance evaluations. Spatial domain methods are simple and easy to embed a high secret data into digital images. Thus, most of the published steganography techniques are carried out on FPGA in the spatial domain. Implementation of FPGA offers lower cost, reconfigurability, and shorter time to market. This literature might assist

https://doi.org/10.59222/ustjet.2.1.3

future researchers who wish to explore this field to analyze the feasibility of hardware implementation of steganography for real-time image processing on FPGA.

References

[1] N. S. Lingamallu and V. Veeramani, "Secure and covert communication using steganography by Wavelet Transform," *Optik*, vol. 242, p. 167167, 2021.

[2] P. C. Mandal, I. Mukherjee, G. Paul, and B. Chatterji, "Digital Image Steganography: A Literature Survey," *Information Sciences*, pp. 1451–1488, 2022.

[3] J. Fridrich, *Steganography in Digital Media: Principles, Algorithms, and Applications*: Cambridge University Press, 2010.

[4] M. Sathya and S. Chitra, "FPGA Implementation of LSB Replacement Steganography Using DWT," *International Journal of Current Engineering and Scientific Research (IJCESR)*, vol. 4, pp. 50-54, 2017.

[5] A. S. Mahajan and S. G. Khadke, "Hardware Implementation of LSB Steganography Using MATLAB and FPGA," *International Journal of Computer Science Trends and Technology (IJCST)*, vol. 3, pp. 41-44, 2015.

[6] J. Chaudhari and K. Bhatt, "FPGA Implementation of Image Steganography: A Retrospective," *International Journal of Engineering Development and Research*, vol. 2, pp. 2117-2121, 2014.

[7] R. Atta and M. Ghanbari, "A high payload data hiding scheme based on dual tree complex wavelet transform," *Optik*, vol. 226, p. 165786, 2021.

[8] J. MSharafi, Y. Khedmati, and M. M. Shabani, "Image Steganography Based on a New Hybrid Chaos Map and Discrete Transforms," *Optik*, vol. 226, p. 165492, 2021.

[9] M. Hussain, A. W. A. Wahab, Y. I. B. Idris, A. T. S. Ho, and K.-H. Jung, "Image Steganography in Spatial Domain: A Survey," *Signal Processing: Image Communication*, vol. 65, pp. 46-66, 2018.

[10] I. J. Kadhim, P. Premaratne, P. J. Vial, and B. Halloran, "Comprehensive Survey of Image Steganography: Techniques, Evaluations, and Trends in Future Research," *Neurocomputing*, vol. 335, pp. 299-326, 2019.

[11] A. M. Ahmadian and M. Amirmazlaghani, "A Novel Secret Image Sharing with Steganography Scheme Utilizing Optimal Asymmetric Encryption Padding and Information Dispersal Algorithms," *Signal Processing: Image Communication*, vol. 74, pp. 78-88, 2019.

[12] E. Cole, Hiding in Plain Sight: *Steganography and the Art of Covert Communication*: Wiley Publishing, Inc., 2003.

[13] S. Karakus and E. Avci, “Application of Similarity-Based Image Steganography Method to Computerized Tomography Images,” in *2019 7th International Symposium on Digital Forensics and Security (ISDFS)*, 2019, pp. 1-4.

[14] I. Makhdoom, M. Abolhasan, and J. Lipman, “A Comprehensive Survey of Covert Communication Techniques, Limitations and Future Challenges,” *Computers & Security*, p. 102784, 2022.

[15] I. j. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*, 2nd ed.: Morgan Kaufmann, 2008.

[16] T. Filler and J. Fridrich, “Design of Adaptive Steganographic Schemes for Digital Images,” in *Media Watermarking, Security, and Forensics III*, 2011, p. 78800F.

[17] J. R. Kullayappa, K. S. M. M. Mohinuddin, and C. M. Aslam, “FPGA Implementation of the 2/3 LSB Steganography using DWT,” *International Journal of VLSI System Design and Communication Systems (IJVDCS)*, vol. 2, pp. 1169-1174, 2014.

[18] A. Ganorkar and S. Agrawal, “Implementation of Steganography on FPGA,” *International Journal of Recent Advances in Engineering & Technology (IJRAET)*, vol. 2, pp. 21-25, 2014.

[19] N. Rashmi and K. Jyothi, “An Improved Method for Reversible Data Hiding Steganography Combined with Cryptography,” in *2nd International Conference on Inventive Systems and Control (ICISC)*, 2018, pp. 81-84.

[20] M. S. Lahase and S. A. Dhole, “Hybrid Encryption and Decryption Method Using LSB and RSA in Steganography,” *International Journal of Electrical, Electronics and Data Communication (IJEEDC)*, pp. 68-70, 2015.

[21] N. A. Roslan, N. I. Udzir, R. Mahmod, and A. Gutub, “Systematic literature review and analysis for Arabic text steganography method practically,” *Egyptian Informatics Journal*, 2022.

[22] K. S. Shete, M. Patil, and J. S. Chitode, “Least Significant Bit and Discrete Wavelet Transform Algorithm Realization for Image Steganography Employing FPGA,” *International Journal of Image, Graphics and Signal Processing*, vol. 8, pp. 48-56, 2016.

[23] B. V. Lakshmi and B. V. Raju, “FPGA Implementation of Lifting DWT based LSB Steganography using Micro Blaze Processor,” *International Journal of Computer Trends and Technology (IJCTT)*, vol. 6, pp. 6-14, 2013.

[24] S. Dhawan and R. Gupta, “Analysis of various data security techniques of steganography: A survey,” *Information Security Journal: A Global Perspective*, vol. 30, pp. 63-87, 2021.

[25] G. R. Kumar, M. M. P. Reddy, and T. L. Kumar, “An Implementation of LSB Steganography Using DWT Technique,” *International Journal of Engineering Research and General Science*, vol. 2, pp. 398-403, 2014.

[26] P. C. Mandal, I. Mukherjee, and B. Chatterji, "High capacity steganography based on IWT using eight-way CVD and n-LSB ensuring secure communication," *Optik*, vol. 247, p. 167804, 2021.

[27] H. Arora, C. Bansal, and S. Dagar, "Comparative Study of Image Steganography Techniques," in *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, 2018, pp. 982-985.

[28] J. Wang, M. Cheng, P. Wu, and B. Chen, "A Survey on Digital Image Steganography," *Journal of Information Hiding and Privacy Protection*, vol. 1, pp. 87-93, 2019.

[29] K. Shete, M. Patil, and J. S. Morbale, "FPGA Implementation of Image Steganography Using LSB and DWT," *International Journal of Computer Science and Network (IJCSN)*, vol. 4, pp. 847-853, 2015.

[30] A. Almawgani, A. R. Alhawari, A. T. Hindi, W. H. Al-Arashi, and A. Al-Ashwal, "Hybrid image steganography method using Lempel Ziv Welch and genetic algorithms for hiding confidential data," *Multidimensional Systems and Signal Processing*, vol. 33, pp. 561-578, 2022.

[31] P. Joseph and S. Vishnukumar, "A Study on Steganographic Techniques," in *2015 Global Conference on Communication Technologies (GCCT)*, 2015, pp. 206-210.

[32] G. R. J and R. Ganesh, "Review of Recent Strategies in Cryptography-Steganography Based Security Techniques," in *2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET)*, 2018, pp. 1-5.

[33] H. Ge, M. Huang, and Q. Wang, "Steganography and Steganalysis Based on Digital Image," in *4th International Congress on Image and Signal Processing*, 2011, pp. 252-255.

[34] P. Johri, A. Mishra, S. Das, and A. Kumar, "Survey on Steganography Methods (Text, Image, Audio, Video, Protocol and Network Steganography)," in *3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2016, pp. 2906-2909.

[35] B. J. Mohd, T. Hayajneh, Z. A. Khalaf, and A. V. Vasilakos, "A Comparative Study of Steganography Designs based on Multiple FPGA Platforms," *International Journal of Electronic Security and Digital Forensics*, vol. 8, pp. 164-190, 2016.

[36] Y. Yao, W. Zhang, N. Yu, and X. Zhao, "Defining Embedding Distortion for Motion Vector-Based Video Steganography," *Multimedia tools and Applications*, vol. 74, pp. 11163-11186, 2014.

[37] V. Holub and J. Fridrich, "Digital Image Steganography Using Universal Distortion," in *Proceedings of the first ACM Workshop on Information Hiding and Multimedia Security*, 2013, pp. 59-68.

[38] T. Pevný, T. Filler, and P. Bas, "Using High-Dimensional Image Models to Perform Highly Undetectable Steganography," in *12th International Workshop on Information Hiding*, 2010, pp. 161-177.

[39] B. J. Mohd, A. Sa'ed, B. Al-Naami, and S. Alouneh, "Image Steganography Optimization Technique," in *International Joint Conference on Advances in Signal Processing and Information Technology*, 2012, pp. 205-209.

[40] B. J. Mohd, A. Sa'ed, B. Na'ami, and T. Hayajneh, "Hierarchical Steganography Using Novel Optimum Quantization Technique," *Signal, Image and Video Processing*, vol. 7, pp. 1029-1040, 2013.

[41] D. Ayyed, "Image Steganography Based Sobel Edge Detection Using FPGA," *Technium*, vol. 2, pp. 23-43, 2020.

[42] K. S. Shet, A. R. Aswath, M. C. Hanumantharaju, and X.-Z. Gao, "Design and Development of New Reconfigurable Architectures for LSB/Multi-Bit Image Steganography System," *Multimedia Tools and Applications*, vol. 76, pp. 13197-13219, 2017.

[43] A. Odeh, K. Elleithy, and M. Faezipour, "Fast Real-Time Hardware Engine for Multipoint Text Steganography," in *IEEE Long Island Systems, Applications and Technology (LISAT) Conference 2014*, 2014, pp. 1-5.

[44] L. Desai and S. Mali, "Crypto-Stego-Real-Time (CSRT) System for Secure Reversible Data Hiding," *VLSI Design*, 2018.

[45] J. J. Garcia-Hernandez, C. Feregrino-Uribe, R. Cumplido, and C. Reta, "On the Implementation of a Hardware Architecture for an Audio Data Hiding System," *Journal of Signal Processing Systems*, vol. 64, pp. 457-468, 2011.

[46] J. Wei, Z. Quan, Y. Hu, J. Liu, H. Zhang, and M. Liu, "Implementing a Low-Complexity Steganography System on FPGA," in *2021 9th International Conference on Intelligent Computing and Wireless Optical Communications (ICWOC)*, 2021, pp. 64-68.

[47] H. Mandal, A. Phadikar, G. K. Maity, and T.-L. Chiu, "FPGA based low power hardware for quality access control of compressed gray scale image," *Microsystem Technologies*, pp. 1-14, 2018.

[48] S. Debnath, M. Kalita, and S. Majumder, "A Review on Hardware Implementation of Steganography," in *2017 Devices for Integrated Circuit (DevIC)*, Kalyani, India, 2017, pp. 149-152.

[49] J. G. Gurav, J. Singh, and M. Tiwari, "Implementation Of LSB Steganography Technique On FPGA For Highly Secured Image," *World Journal of Advanced Engineering and Technology* vol. 1, pp. 18-22, 2016.

[50] A. Mahesh and K. Raja, “Design of an Efficient Steganography Model using Lifting based DWT and Modified-LSB Method on FPGA,” *International Journal of Advanced Computer Science and Applications*, vol. 10, pp. 226-231, 2019.

[51] S. K. Moon and R. D. Raut, “Hardware-based application of data security system using general modified secured diamond encoding embedding approach for enhancing imperceptibility and authentication,” *Multimedia Tools and Applications*, vol. 78, pp. 22045-22076, 2019.

[52] K. S. Shet, A. R. Aswath, M. C. Hanumantharaju, and X.-Z. Gao, “Novel High-Speed Reconfigurable FPGA Architectures for EMD-Based Image Steganography,” *Multimedia Tools and Applications*, vol. 78, pp. 18309-18338, 2019.

[53] I. Kuon, R. Tessier, and J. Rose, *FPGA Architecture: Survey and Challenges*: Now Publishers Inc, 2008.

[54] S. Rajagopalan, R. Amirtharajan, H. N. Upadhyay, and J. B. B. Rayappan, “Survey and Analysis of Hardware Cryptographic and Steganographic Systems on FPGA,” *Journal of Applied Sciences*, vol. 12, pp. 201-210, 2012.

[55] R. Cumplido, C. Feregrino-Uribe, and J. J. Garcia-Hernandez, “Implementing Digital Data Hiding Algorithms in Reconfigurable Hardware - Experiences on Teaching and Research,” in *6th International Workshop on Reconfigurable Communication-Centric Systems-on-Chip (ReCoSoC)*, Montpellier, France, 2011, pp. 1-6.

[56] B. K. Yakti, S. Madenda, S. A. Sudiro, and P. Musa, “Processing Speed Comparison of the Least Significant Bit (LSB) Steganography Algorithm on FPGA and Matlab,” in *2021 Sixth International Conference on Informatics and Computing (ICIC)*, 2021, pp. 1-7.

[57] B. J. Mohd, T. Hayajneh, S. e. Abed, and A. Itradat, “Analysis and Modeling of FPGA Implementations of Spatial Steganography Methods,” *Journal of Circuits, Systems, and Computers*, vol. 23, p. 1450018, 2014.

[58] S. Baddap, K. Khomane, P. Deshmukh, and P. Shilpa, “Hardware Implementation of LSB Steganography for Data Security,” *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, vol. 2, pp. 59-63, 2015.

[59] K. Pansare and A. Kureshi, “A Review–FPGA Implementation of Different Steganographic Technique,” *International Journal of innovation Research in Science, Engineering and Technology*, vol. 3, pp. 377-381, 2014.

[60] D. N. Aini, S. N. Putro, E. H. Rachmawanto, and C. A. Sari, "Survey of Methods in the Spatial Domain Image Steganography based Imperceptibility and Payload Capacity," in *2019 International Seminar on Application for Technology of Information and Communication (iSemantic)*, 2019, pp. 434-439.

[61] A. A. Abd EL-Latif, B. Abd-El-Atty, and S. E. Venegas-Andraca, "A Novel Image Steganography Technique Based on Quantum Substitution Boxes," *Optics and Laser Technology*, vol. 116, pp. 92-102, 2019.

[62] K. Pathak and M. Bansal, "A FPGA based Steganographic System Implementing a Modern Steganalysis Resistant LSB Algorithm," *Defence Science Journal*, vol. 67, pp. 551-558, 2017.

[63] G. Swain, "Very High Capacity Image Steganography Technique Using Quotient Value Differencing and LSB Substitution," *Arabian Journal for Science and Engineering*, vol. 44, pp. 2995-3004, 2019.

[64] P. Xue, H. Liu, J. Hu, and R. Hu, "A Multi-Layer Steganographic Method Based on Audio Time Domain Segmented and Network Steganography," in *AIP Conference Proceedings*, 2018, p. 020046.

[65] W. A. P. Laces and J. J. Garcia-Hernandez, "FPGA Implementation of a Low Complexity Steganographic System for Digital Images," in *14th International Conference on Computer and Information Science (ICIS)*, 2015, pp. 319-324.

[66] B. J. Mohd, S. Abed, T. Al-Hayajneh, and S. Alouneh, "FPGA Hardware of the LSB Steganography Method," in 2012 International Conference on Computer, *Information and Telecommunication Systems (CITS)*, 2012, pp. 1-4.

[67] A. AlWatyan, W. Mater, O. Almutairi, M. Almutairi, A. Al-Noori, and S. Abed, "Security Approach for LSB Steganography Based FPGA Implementation," in *2017 7th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO)*, 2017, pp. 1-5.

[68] S. M. Ismail, A. M. Ghidan, and P. W. Zaki, "Novel chaotic random memory indexing steganography on FPGA," *AEU-International Journal of Electronics and Communications*, vol. 125, p. 153367, 2020.

[69] V. Prasad and P. K. Shah, "A Data Security module based on Crypto and Steganography techniques," in *2021 2nd International Conference for Emerging Technology (INCET)*, 2021, pp. 1-5.

[70] K. Nandhini and B. Gomathi, "Implementation of LSB Based Steganography Algorithms in FPGA," *International Journal of Scientific Research in Network Security and Communication*, vol. 6, pp. 32-37, 2018.

[71] H. Farouk and M. Saeb, "Design and Implementation of a Secret Key Steganographic Micro-Architecture Employing FPGA," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition Designers' Forum*, 2004, pp. 212-217.

[72] H. A. Farouk and M. Saeb, "An Improved FPGA Implementation of the Modified Hybrid Hiding Encryption Algorithm (MHHEA) for Data Communication Security," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, 2005, pp. 76-81.

[73] R. Sundararaman and H. N. Upadhyay, "Stego System on Chip with LFSR based Information Hiding Approach," *International Journal of Computer Applications*, vol. 18, pp. 24-31, 2011.

[74] C.-W. Huang, C. Chou, Y.-C. Chiu, and C.-Y. Chang, "Embedded FPGA Design for Optimal Pixel Adjustment Process of Image Steganography," *Mathematical Problems in Engineering*, vol. 2018, 2018.

[75] Q. Do Vinh and I. Koo, "FPGA Implementation of LSB-based Steganography," *Journal of Information and Communication Convergence Engineering*, vol. 15, pp. 151-159, 2017.

[76] E. A. Elshazly, S. A. S. Abdelwahab, R. M. Fikry, S. M. Elaraby, O. Zahran, and M. El-Kordy, "FPGA Implementation of Robust Image Steganography Technique based on Least Significant Bit (LSB) in Spatial Domain," *International Journal of Computer Applications*, vol. 145, pp. 43-52, 2016.

[77] R. Amirtharajan, R. Balaguru, and V. Ganesan, "Design and Analysis of Prototype Hardware for Secret Sharing Using 2-D Image Processing," *International Journal of Computer Applications*, vol. 4, pp. 0975-8887, 2010.

[78] B. Ramalingam, R. Amirtharajan, and J. B. B. Rayappan, "Multiplexed Stego Path on Reconfigurable Hardware: A Novel Random Approach," *Computers and Electrical Engineering*, vol. 55, pp. 153-163, 2016.

[79] A. F. K. Mahmood, Nada Abdul and S. S. Mohmmad, "An FPGA Implementation of Secured Steganography Communication System," *Tikrit Journal of Engineering Sciences*, vol. 19, pp. 14-23, 2012.

[80] K. Nandhini and S. Arivazagan, "FPGA Implementation of LSB-MR based Steganography Algorithms," *ICTACT Journal on Microelectronics*, vol. 4, pp. 560-565, 2018.

[81] V. S. Kait and B. Chauhan, "BPCS Steganography for Data Security using FPGA Implementation," in *2015 International Conference on Communications and Signal Processing (ICCSP)*, 2015, pp. 1887-1891.

https://doi.org/10.59222/ustjet.2.1.3

[82] E. Gómez-Hernández, C. Feregrino-Uribe, and R. Cumplido, “FPGA Hardware Architecture of the Steganographic Context Technique,” in *18th International Conference on Electronics, Communications and Computers (CONIELECOMP 2008)*, 2008, pp. 123-128.

[83] S. Timarchi, M. A. Alaei, and H. Koushkbaghi, “Novel Algorithm and Architectures for High-Speed Low-Power ConText-Based Steganography,” in *19th International Symposium on Computer Architecture and Digital Systems (CADS)*, 2017, pp. 1-6.

[84] E. A. Elshazly, S. A. S. Abdelwahab, R. M. Fikry, O. Zahran, S. M. Elaraby, and M. El-Kordy, “FPGA Implementation of Image Steganography Algorithms using Generalized Exploiting Modification Direction and Pixel Segmentation Strategy,” in *2018 35th National Radio Science Conference (NRSC), Misr International University (MIU)*, Cairo, Egypt, 2018, pp. 258-265.

[85] P. Premalatha and A. Amsaveni, “Data Hiding in a Digital Image with FPGA Implementation,” in *Online International Conference on Green Engineering and Technologies (IC-GET)*, 2016, pp. 1-5.

[86] S. M. Sakthivel and A. R. Sankar, “FPGA Implementation of Data Hididng in Grayscale Imagesusing Neighbour Mean Interpolation,” in *2nd International Conference on Electronics and Communication Systems (ICECS)*, 2015, pp. 1124-1127.

[87] N. S. H.N., P. M. Prakash, S. S. Kashyap, and S. Sarkar, “FPGA Implementation of Image Steganography using Haar DWT and Modified LSB Techniques,” in *International Conference on Advances in Computer Applications (ICACA)*, 2016, pp. 26-31.

[88] B. J. Mohd, T. Hayajneh, and A. N. Quttoum, “Wavelet-Transform Steganography: Algorithm and Hardware Implementation,” I*nternational Journal of Electronic Security and Digital Forensics*, vol. 5, pp. 241-256, 2013.

[89] I. Algredo-Badillo, F. R. Castillo-Soria, K. A. Ramirez-Gutierrez, L. Morales-Rosales, A. Medina-Santiago, and C. Feregrino-Uribe, “Lightweight security hardware architecture using DWT and AES algorithms,” *IEICE TRANSACTIONS on Information and Systems*, vol. 101, pp. 2754-2761, 2018.

[90] B. Ramalingam, R. Amirtharajan, and J. B. B. Rayappan, “Stego on FPGA: An IWT approach,” *The Scientific World Journal*, pp. 1-9, 2014.

[91] U. R. Rajeesh and R. R. Ahamed, “High Speed DCT Based Image Steganography Implementation on FPGA,” *International Journal of Engineering Research & Technology (IJERT)*, vol. 3, pp. 762-765, 2014.