# A Lightweight FPGA-Based Steganography Accelerator Core for Edge Platforms

Cuong Pham-Quoc[1,2(✉)]

[1] Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam
cuongpham@hcmut.edu.vn

[2] Vietnam National University - Ho Chi Minh City (VNU-HCM), Thu Duc, Ho Chi Minh City, Vietnam

**Abstract.** In the era of edge computing, where IoT devices are increasingly connected to the Internet, data security and privacy have become critical concerns. Steganography, a technique for embedding hidden information within digital media, offers a lightweight security measure that complements traditional cryptographic methods. However, implementing steganographic techniques efficiently in edge devices poses challenges related to computational overhead, power consumption, and real-time performance. This paper introduces a lightweight FPGA-based steganography accelerator core designed for edge computing platforms. Our design optimizes resource efficiency, low latency, and adaptability while ensuring robust data hiding and retrieval capabilities. The proposed system is evaluated with respect to hardware utilization, power efficiency, and embedding performance. We evaluated our system on the Xilinx Kria KV260 edge computing platform, analyzing hardware utilization, execution time, power consumption, and throughput across different image sizes. Experimental results demonstrate that our FPGA-based accelerator outperforms a traditional ARM processor-based software implementation, achieving up to 2.42× speed-up with dual-core configurations. Furthermore, power efficiency analysis shows that the system effectively minimizes energy consumption, making it suitable for power-sensitive edge applications. The proposed solution is highly adaptable and can be integrated with other hardware accelerators, such as video encoding or network security modules, to enhance processing capabilities in IoT environments.

**Keywords:** FPGA · Hardware accelerator · DCT · Steganography · Edge Computing · IoTs

## 1 Introduction

In the era of edge computing, while more and more IoT devices are connected to the Internet [16], data security and privacy have become critical concerns

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2025 Workshops, LNCS 15891, pp. 381–392, 2026.
https://doi.org/10.1007/978-3-031-97617-9_25

due to the increasing reliance on distributed devices and resources restricted [7]. Steganography, a technique for concealing information within digital media, has gained attention as a lightweight security measure that complements traditional cryptographic methods. However, implementing steganographic techniques efficiently in edge devices presents challenges related to computational overhead, power consumption, and real-time performance.

Field-Programmable Gate Arrays (FPGAs) offer an attractive solution to address these challenges by providing hardware acceleration with low power consumption and high processing efficiency. In this paper, we propose a lightweight FPGA-based steganography core optimized for edge computing platforms. Our design prioritizes resource efficiency, low latency, and adaptability to various edge applications while ensuring robust data hiding and retrieval capabilities.

The proposed steganography core is evaluated in terms of hardware utilization, power efficiency, and embedding performance. The results demonstrate its suitability for security-sensitive applications in edge environments, such as IoT networks, industrial automation, and smart surveillance systems. Using FPGA-based acceleration, this work contributes to the development of secure and efficient data hiding techniques tailored for edge computing platforms.

The main contributions of our paper can be summarized in three folds.

1. We introduced our lightweight FPGA-based Steganography accelerator core targeting IoT platforms for embedded hidden messages into containers such as images or videos.
2. We illustrate our proposed system with the prototype on the Xilinx Kria KV260 edge computing platform to evaluate performance and throughput.
3. We analyze our experimental results with the prototype platform and different image sizes for comparison of future studies.

The remainder of the paper is organized as follows. Section 2 shows the background on the work related to steganography and FPGA-based systems. We introduce our proposed FPGA-based steganography accelerator core in Sect. 3. Section 4 shows our prototype with Xilinx Kria KV260 edge computing. Experiments with various image sizes and the prototype platform are analyzed in Sect. 5. Finally, Sect. 6 concludes our paper.

## 2 Preliminaries and Related Work

In this section, we first present the background of steganography in data security. We then analyzed related work in the literature that targets the implementation of FPGA-based steganography.

### 2.1 Steganography

Steganography is the practice of concealing information within a carrier medium in a way that prevents detection by unauthorized parties [5,8,9]. Unlike cryptography, which focuses on encrypting data to make it unreadable without a

decryption key, steganography aims to hide the existence of the communication itself. This covert approach makes steganography an effective tool for secure data transmission, digital watermarking, and anti-forensic applications.

Traditional steganographic techniques embed information into various digital media such as images, audio files, videos, and text. Common methods include Least Significant Bit (LSB) substitution, transform domain techniques, and adaptive steganography, each offering different trade-offs between security, capacity, and imperceptibility. With the rise of modern communication systems and digital forensics, researchers continue to develop advanced steganographic approaches that enhance robustness against detection while maintaining efficiency.

The growing adoption of edge computing and resource-constrained devices has introduced new challenges and opportunities for steganography. Efficient and lightweight steganographic solutions are increasingly required to ensure secure data exchange in environments with limited processing power and storage. As a result, hardware-based steganography, particularly FPGA and ASIC implementations, has gained attention for its ability to accelerate steganographic processes while minimizing power consumption and latency.

### 2.2 FPGA-Based Steganography Related Work

FPGA-based steganography cores are specialized hardware implementations designed to enhance the performance and efficiency of steganography techniques that involve hiding secret data within digital media. These cores leverage the reconfigurability and parallel processing capabilities of Field Programmable Gate Arrays (FPGAs) to achieve high-speed data embedding and extraction, making them suitable for real-time applications.

FPGA-based implementations of steganography algorithms, such as the Exploiting Modification Direction (EMD) and Least Significant Bit (LSB) methods, offer significant advantages in terms of speed and flexibility. These architectures are designed to handle variable image resolutions and pixel group sizes, allowing real-time processing at high frame rates. For example, EMD-based systems can embed secret messages in images with resolutions up to $512 \times 512$ pixels at 549 frames per second [15]. Similarly, LSB-based systems achieve high throughput by incorporating pipelining and parallel operations, processing data at rates of 183.48 frames per second [13].

The use of FPGAs in steganography provides several benefits, including reduced computational delay, increased throughput, and lower power consumption. These systems are capable of real-time data processing, which is crucial for applications requiring immediate data embedding and extraction [2]. Furthermore, FPGA implementations are cost-effective and offer flexibility to adapt to different steganography algorithms and cover media types, such as images, audio, and video [6].

Despite the advantages, FPGA-based steganography systems face challenges such as the complexity of algorithm implementation and the need for efficient resource utilization. Future research may focus on optimizing these systems for

even higher performance and exploring new algorithms that can be effectively implemented on FPGAs. There is also potential for integrating cryptographic techniques to improve the security of the steganographic process [1,14].

## 3 Proposed FPGA-Based Steganography Core Architecture

In this section, we introduce our FPGA-based steganography accelerator core. We design the core according to the hardware accelerator paradigm [11] so that the core can be integrated with different hardwired or soft general-purpose processors to develop applications.

### 3.1 Overview Architecture

Figure 1 illustrates the overview architecture of our steganography core. The core communicates with the host processor through a communication infrastructure (bus, shared memory, etc.) for transferring input data (parameters, messages, images/videos, etc.) and output results. Exchange registers and two buffers (one for Discrete Cosine Transform - DCT, and one for Steganography) are used for these purposes.

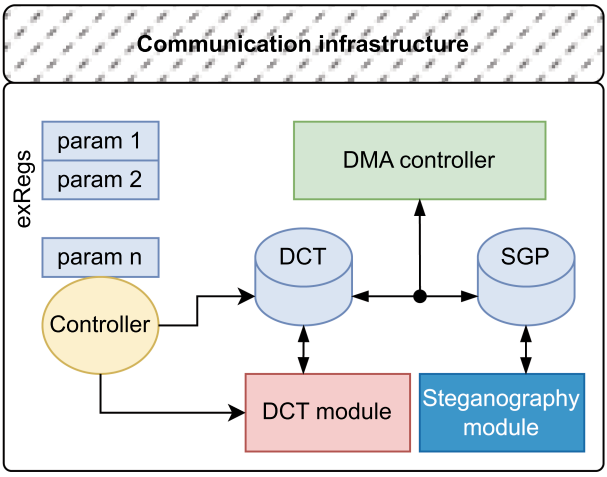


**Fig. 1.** The overview of the lightweight FPGA-based steganography accelerator core for IoT platforms

The core components of the accelerated steganography engine consist of two main modules: the discrete cosine transform (DCT) computation module and the steganography module. The DCT module performs discrete cosine transform operations on input data matrices extracted from background image frames. The

DCT-transformed data are then transferred to the buffer of the steganography module, where the hidden messages are embedded using steganographic techniques. The final output is transmitted to the processor via DMA. The details of the two modules are discussed in the following subsections.

### 3.2 DCT Module

The 2D discrete cosine transform (DCT) is formed by two consecutive steps of the 1D discrete cosine transform, connected by a transposition step. Initially, the 1D DCT reads data from the input matrix row by row, then a series of computations is performed, and the results are stored in an intermediate matrix column by column. Once the input matrix has been fully processed and the intermediate matrix is completely filled, the second 1D DCT begins, repeating the same process with the intermediate matrix as input, and reading the transposed results from the first 1D DCT. The final results are then stored in the output matrix. This process is illustrated in Fig. 2. Since this work focuses on the $8 \times 8$ DCT for images, two 1D DCTs with 8-point computations are used to perform the 2D DCT.

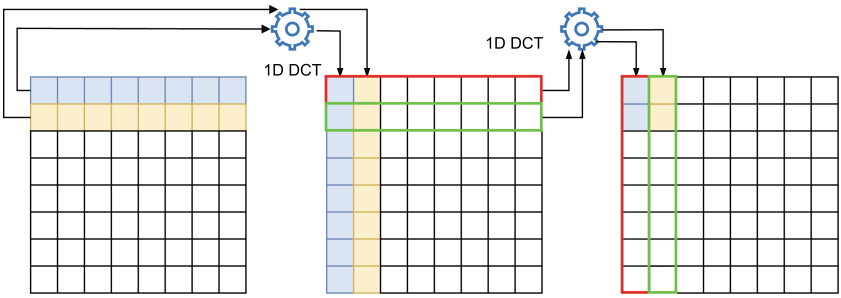


**Fig. 2.** 2D-DCT computation based on 2× 1D-DCT

Based on the DCT principle mentioned above, we designed the architecture of the DCT calculation module using a pipelined processing model to improve performance, as shown in Fig. 3. The processing of a DCT matrix consists of four stages, where the last two stages repeat the first two. These stages include butterfly computations, multiply-accumulate (MAC) operations, and transpose map.

The task of the Butterfly unit is to compute the sum and difference of each pair of numbers from the input matrix data. The input data is an $8 \times 8$ matrix, where each row is processed sequentially and fed into the Butterfly unit for computation. The output of this unit for each row produces eight values, with four values being the result of addition and the other four being the result of subtraction for each pair of numbers in the corresponding row of the input matrix.

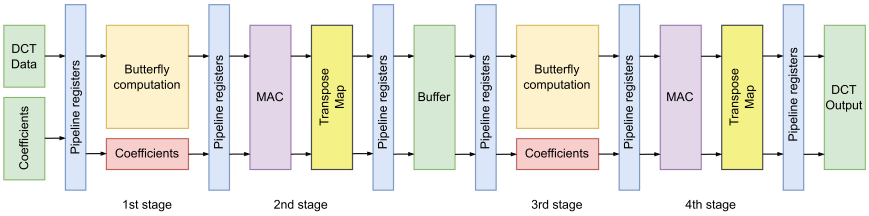


**Fig. 3.** The pipeline architecture for 2D DCT computation

The `MAC` unit performs multiplication and accumulation on the output values of the `Butterfly` unit to compute the DCT coefficient values. The summed values from the `Butterfly` unit is multiplied by the coefficients of the matrix, which are retrieved from the memory block (buffer), and the accumulated results produce the DCT coefficients. The same process is applied to the difference values from the Butterfly unit. The computed results from the MAC unit are stored in RAM to be used in subsequent computation stages.

The Transpose map block plays a crucial role in the DCT acceleration module. Due to the nature of the algorithm, the output data from the MAC unit does not follow a specific order. The Transpose Map block is responsible for transposing the matrix and mapping the output results from the MAC unit to their correct memory addresses in storage.

### 3.3 Steganography Module

Based on the steganography algorithms developed, the steganography system utilizes the Fully Exploiting Modification Direction (EMD-3) algorithm. According to the EMD-3 algorithm, we choose $n = 3$, which means that for every three values of the cover data, we can embed hidden information with a maximum value of $3^3 = 27$. For each byte of hidden information, the system uses six values of the cover data for embedding. Figure 4 illustrates the architecture of the Steganography module with three functional groups: "steganography algorithm implementation," "data read/write," and "control."

To implement the steganography algorithm, the `Extract`, `Decision`, and `Embedding` blocks perform tasks based on the algorithm. The `Extract` block functions in two modes: embedding and extracting hidden information from input data. In embedding mode, it handles data preprocessing and computes the extract function. The output of the Extract block consists of cover data (or carrier data) and the result of the extract function. The `Decision` block is responsible for computing the decision function based on the output of the extract function and the hidden information value. Using these inputs, the Decision block determines whether modifications to the cover data values are necessary. The `Embedding` block is the most critical component, responsible for comparing and embedding hidden information into the cover data. It uses the output of the extract function of the `Extract` block and the decision function of

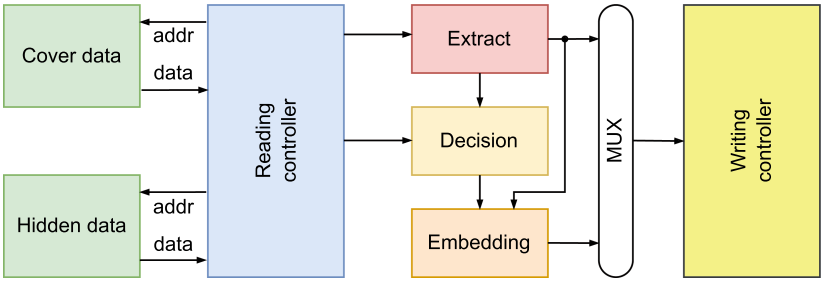


**Fig. 4.** The architecture for the steganography module

the `Decision` block. Based on the comparison result of the `Decision` block, the `Embedding` block calculates values and decides whether to increase or decrease each cover data value by 1.

To read/write data, we build the `Reading` and `Writing` controllers. The `Reading Controller` block performs the task of reading data values from external memory and splitting these data into individual data units. Then it loads them into the Input Register block for data packaging. The `Writing Controller` block operates similarly to the Reading Controller block, verifying output data and segmenting the result data into individual units suitable for storage in external memory.

Finally, for handling operations of data read/write and steganography, the `Control unit` checks the parameters configured by the user and the system's operating mode to manage and regulate the output data appropriately. It also controls the stages in the pipeline of the Steganography accelerator system. Additionally, the Control block monitors the completion of the data hiding or extraction process to configure registers and notify the software.

## 4 FPGA Edge Computing Prototype

This section presents our prototype FPGA-based edge computing platform with our integrated steganography accelerator core[1]. In this work, we use the Xilinx Kria KV260 edge computing system [3] to deploy our prototype. The platform contains a Xilinx Zynq UltraScale+ MPSoC ZU5CG with 256K logic cells, 144 Block RAM (36 Kbits/block), and other components. The core modules are described by Verilog HDL and synthesized with Xilinx Vivado [4]. Along with the reconfigurable logic for building the Steganography accelerator core, the Zynq FPGA chip also contains a hardwired dual-core ARM Cortex-A53 processor for executing software. One of the use-cases is to embed a signature into images/videos for certifying that the images/videos are generated from trust devices.

[1] The author thanks Mr. Le Phu Thuan for describing the core with Verilog-HDL.

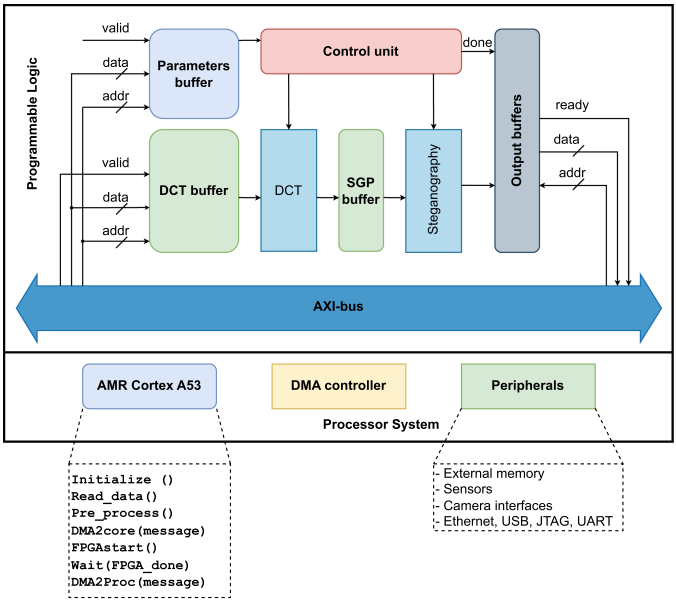


**Fig. 5.** The prototype system with Xilinx Kria KV260 board

Figure 5 presents an overview of our prototype system. In this prototype, the hardwired ARM processor is a host processor to control and collect data from peripherals such as sensors and cameras and to communicate with the cloud through the Ethernet interface. To certify the images, videos, the host processor starts transferring data to our Steganography core by Direct Memory Access (DMA) through the AXI bus interface for the signing process. During this interval, the host processor can wait or perform other tasks. Images/videos after embedded message or signature are transferred back from the accelerator core for sending to the cloud.

## 5 Experiments

This section presents experiments to evaluate the effectiveness of the Steganography accelerator core and the FPGA-based prototype systems. The experiment results are then examined, comparing the system's performance with the accelerator against using only software on the ARM processor. In addition, power consumption will be evaluated, analyzed, and compared.

### 5.1 Synthesis Results

The Steganography accelerator core, developed by Verilog-HDL, and the entire system are synthesized and configured on the Kria board by Xilinx Vivado 2023.2. For testing purposes, we synthesize the system that consists of one

DCT & one Steganography modules and two modules of each (two DCT and two Steganography), respectively. The results of the synthesis reveal that with two `DCT & Steganography` modules, the system can operate at 150 MHz and requires 2.973 W of total power consumption. Table 1 presents the usage of hardware resources for the system with one and two `DCT & Steganography` modules, respectively.

**Table 1.** Hardware resources usage for the proposed system with one or two `DCT & Steganography` modules

| Resources | 2 cores | | 1 core | | Available |
|---|---|---|---|---|---|
| | Amount | Percentage | Amount | Percentage | |
| Look-up tables (LUTs) | 50,109 | 42% | 24,670 | 21% | 117,120 |
| Flip-flop (FFs) | 28,693 | 24% | 14,597 | 12% | 234,240 |
| DSP | 58 | 4% | 29 | 2% | 1,248 |
| Block RAMs (BRAMs) | 176 | 50% | 176 | 50% | 352 |

As presented in the table, our system uses up to 42% of the hardware resources of the FPGA, indicating the potential to integrate additional modules `DCT` and `Steganography` to further enhance performance. Using the hardware accelerator computing model, the system also enables the incorporation of specialized cores for specific tasks, such as video encoding/decoding [10] or attack detection [12], thereby improving processing efficiency and overall performance in IoT edge computing environments.

### 5.2 Simulation Results

Figure 6 illustrates how the `DCT module` operates as a pipeline. When the module begins the computation, each output value corresponds to a mapped address. For the `DCT module`, in each cycle, it computes two output values. Thus, it only takes 32 clock cycles to write the entire $8 \times 8$ DCT matrix (containing 64 DCT coefficients).

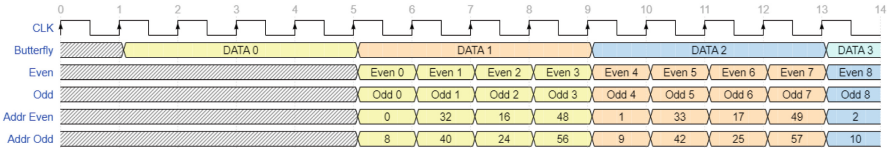


**Fig. 6.** The waveform for simulation the DCT module

For the `Steganography` module, the data hiding process requires four cycles from the start to package the first data block to embed the file. Similarly, at

the system's output, the data packet must be split into smaller subpackets and written into BRAM. Figure 7 illustrates the process of segmenting the read data from BRAM into subfiles and packaging the data every three subdata units. Data packets are read every three cycles to ensure continuous processing of the entire dataset.

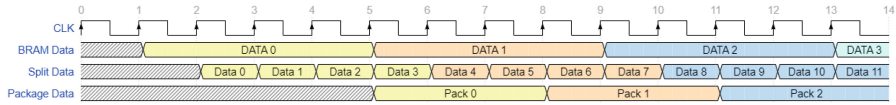


**Fig. 7.** The waveform for simulation the Steganography module

### 5.3 Performance Evaluation and Comparison

We evaluate the processor performance using one and two `DCT` & `Steganography` modules with different image sizes: $512 \times 512$, $256 \times 256$, and $128 \times 128$ pixels. The data hiding process is performed using: (1) A hardwired ARM core operating at 1.5 GHz, call software; and (2) Our accelerated system, which utilizes a dedicated Steganography accelerator core.

Table 2 shows the execution time of our accelerator system (with one and two `DCT` & `Steganography` modules) and of ARM processor only. As shown in the table, when one `DCT` & `Steganography` modules are deployed, we manage to achieve speed-ups by up to 1.54× compared to the ARM processor functioning at 1.5 GHz. When two of the modules are integrated, our system outperforms the ARM processor by up to 2.42×.

**Table 2.** Execution time (s) with different image sizes

| Image | 2 cores | | 1 core | | Software |
|---|---|---|---|---|---|
| | Exec. time | Speed-up | Exec. Time | Speed-up | |
| 512 × 512 | 3.12 | 2.17× | 4.58 | 1.48× | 6.80 |
| 256 × 256 | 0.91 | 1.97× | 1.61 | 1.53× | 1.78 |
| 128 × 128 | 0.19 | 2.42× | 0.30 | 1.54× | 0.46 |

In addition to comparing execution time, we also evaluated processing bandwidth using the parameters pixels per second (pixel/s) and hidden data size per second (bit/s). Furthermore, we assess the system's energy consumption using the micro-joule metrics per Byte (mJ/byte) and micro-joules per Pixel (mJ/pixel) to measure energy usage based on the amount of hidden information or image size. Table 3 compares throughput and energy efficiency between different hidden data and image sizes. As shown in the table, the system functions

well with small-size images in terms of throughput bit/s and energy consumption per hidden byte. This characteristic is suitable for edge computing systems where energy is a critical issue.

**Table 3.** Throughput and energy consumption

| Image | Data (byte) | pixel/s | bit/s | mJ/byte | mJ/pixel |
|---|---|---|---|---|---|
| 128 × 128 | 124 | 54,631 | 413 | 16.99 | 0.127 |
| 256 × 256 | 375 | 56,415 | 322 | 22.39 | 0.128 |
| 512 × 512 | 975 | 57,169 | 212 | 33.98 | 0.126 |

## 6 Conclusion

This paper presented a lightweight FPGA-based steganography accelerator core optimized for edge computing environments. The proposed system efficiently embeds hidden messages within digital media while maintaining low power consumption and high processing performance. Through synthesis and experimental evaluation on the Xilinx Kria KV260 platform, we demonstrated that our design effectively accelerates steganographic operations compared to software-based implementations. The results highlight significant improvements in execution time, throughput, and energy efficiency, making the accelerator a viable solution for IoT and edge AI applications.

Future work will explore further optimizations, including the integration of advanced compression techniques and the extension of support for additional media types such as audio and video. Additionally, we aim to enhance the security robustness of the system against steganalysis attacks, ensuring its applicability in diverse real-world scenarios requiring covert communication and data protection.

**Acknowledgement.** We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study.

## References

1. FPGA based data hiding through steganography. Int. J. Recent Technol. Eng. (2020). https://doi.org/10.35940/ijrte.a2751.059120
2. Al-Ashwal, A.Y., Al-Arashi, W.H., Saad, A., Al-Shadadi, M.M.: Comprehensive survey of image steganography systems based on FPGA implementation. Univ. Sci. Technol. J. Eng. Technol. **2**(1), 35–70 (2024)
3. AMD Xilinx: Kria KV260 vision AI starter kit user guide (UG1089) (2022). https://docs.amd.com/r/en-US/ug1089-kv260-starter-kit. Accessed 31 Jan 2024

4. AMD Xilinx: Xilinx vivado design suite 2023.2 (2023). https://www.xilinx.com/products/design-tools/vivado.html. Accessed 31 Jan 2024
5. Cox, I., Miller, M., Bloom, J., Fridrich, J., Kalker, T.: Digital Watermarking and Steganography. Morgan Kaufmann (2007)
6. He, J., Sun, W., Jiang, H.: Implementation and verification of steganography IP core based on HLS and openCV. J. Phys. Conf. Ser. **2245**, 012010 (2022)
7. Hossain, M.M., Fotouhi, M., Hasan, R.: Towards an analysis of security issues, challenges, and open problems in the internet of things. In: 2015 IEEE World Congress on Services, pp. 21–28. IEEE (2015)
8. Hussain, M., Wahab, A., Idris, Y., Ho, A.T., Jung, K.H.: Image steganography in spatial domain: a survey. Signal Process. Image Commun. **65**, 46–66 (2018)
9. Kadhim, I.J., Premaratne, P., Vial, P.J., Halloran, B.: Comprehensive survey of image steganography: techniques, evaluations, and trends in future research. Neurocomputing **335**, 299–326 (2019)
10. Pham-Quoc, C.: FPGA-based hardware/software codesign for video encoder on IoT edge platforms. In: International Conference on Computational Science and Its Applications, pp. 82–96. Springer (2023)
11. Pham-Quoc, C., Heisswolf, J., Werner, S., Al-Ars, Z., Becker, J., Bertels, K.: Hybrid interconnect design for heterogeneous hardware accelerators. In: 2013 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 843–846. IEEE (2013)
12. Pham-Quoc, C., Thinh, T.N.: FPGA-enabled efficient framework for high-performance intrusion prevention systems. In: International Conference on Computational Science and Its Applications, pp. 83–98. Springer (2023)
13. Sathish Shet, K., Aswath, A., Hanumantharaju, M., Gao, X.Z.: Design and development of new reconfigurable architectures for LSB/multi-bit image steganography system. Multimedia Tools Appl. **76**, 13197–13219 (2017)
14. Savani, V., Mecwan, A., Gajjar, V.: Design and implementation of audio steganography on FPGA. STM J. **1**, 1–6 (2021). https://doi.org/10.37591/JOEDT.V1I1-3.4951
15. Shet, K.S., Aswath, A., Hanumantharaju, M., Gao, X.Z.: Novel high-speed reconfigurable FPGA architectures for EMD-based image steganography. Multimedia Tools Appl. **78**(13), 18309–18338 (2019)
16. Statista Research Department: Internet of things - number of connected devices worldwide 2015-2025 (2016). https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/. Accessed 01 Apr 2023