

## Article

# SoC-Based Edge Computing Gateway in the Context of the Internet of Multimedia Things: Experimental Platform

Maher Jridi \*, Thibault Chapel, Victor Dorez, Guénolé Le Bougeant and Antoine Le Botlan

ISEN Yncrea Ouest, VISION Team, 20 rue Cuirassée Bretagne, 29200 Brest, France;  
Thibault.Chapel@isen-bretagne.fr (T.C.); Victor.Dorez@isen-bretagne.fr (V.D.);  
Guenole.LeBougeant@isen-bretagne.fr (G.L.B.); Antoine.LeBotlan@isen-bretagne.fr (A.L.B.)

\* Correspondence: maher.jridi@isen-bretagne.fr; Tel.: +33-669-750-693

Received: 4 October 2017; Accepted: 10 January 2018; Published: 12 January 2018

**Abstract:** This paper presents an algorithm/architecture and Hardware/Software co-designs for implementing a digital edge computing layer on a Zynq platform in the context of the Internet of Multimedia Things (IoMT). Traditional cloud computing is no longer suitable for applications that require image processing due to cloud latency and privacy concerns. With edge computing, data are processed, analyzed, and encrypted very close to the device, which enable the ability to secure data and act rapidly on connected things. The proposed edge computing system is composed of a reconfigurable module to simultaneously compress and encrypt multiple images, along with wireless image transmission and display functionalities. A lightweight implementation of the proposed design is obtained by approximate computing of the discrete cosine transform (DCT) and by using a simple chaotic generator which greatly enhances the encryption efficiency. The deployed solution includes four configurations based on HW/SW partitioning in order to handle the compromise between execution time, area, and energy consumption. It was found with the experimental setup that by moving more components to hardware execution, a timing speedup of more than nine times could be achieved with a negligible amount of energy consumption. The power efficiency was then enhanced by a ratio of 7.7 times.

**Keywords:** Internet of Multimedia Things; power efficiency; adaptive signal processing; approximate computing; FPGA design

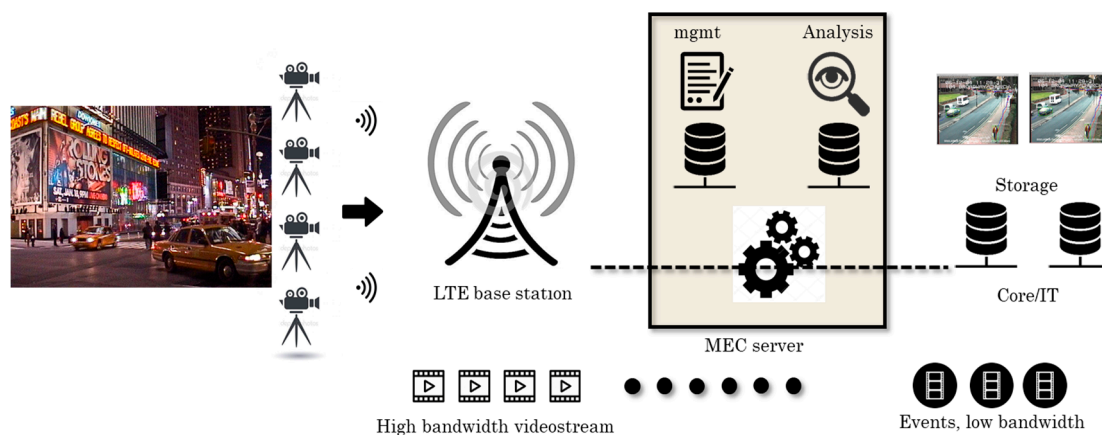
## 1. Introduction

### 1.1. Context

Today, the Internet of Things (IoT) is no longer a strategic initiative. It is a reality for major industry, government, and consumer sectors. With an estimated 15 billion connected objects and an annual growth of 17%, the IoT are rapidly becoming part of our everyday lives. The most basic architecture of the IoT is composed of three layers: perception, network, and application. With this architecture, captured data are transmitted to the cloud to be analyzed in order to create added value. It is in this way that more than 90% of big data has been generated since 2011, with more than 2/3 of big data being multimedia [1]. However, the cloud cannot support and analyze the continuously increasing quantity of data. Cloud computing is no longer suitable for applications that require video analytics due to the long data transmission latency and privacy concerns. A new paradigm often referred as edge computing (fog computing, near sensor computing) has emerged over the past few years to overcome these problems. Shi, et al. define edge computing as enabling technologies “allowing computation to be performed at the edge of the network, on downstream data on behalf of cloud

services and upstream data on behalf of IoT services” [2]. The goal of edge computing is to process data very close to the device, and send meaningful information only to higher layers. According to the International Data Corporation (IDC) [3], it is expected that by 2019 at least 40% of IoT-created data will be stored, processed, analyzed, and acted upon close to, or at the edge of, the network.

The Internet of Multimedia Things (IoMT) can be considered as a class of the IoT. The IoMT is composed of smart things equipped with the capability to observe and/or interact with the physical environment [4]. Today, for smart city applications, it is difficult to imagine an IoT architecture without multimedia things. Contrary to classical IoT devices, where sensors acquire and transmit information with low data rates, the use of multimedia things implies new challenges, such as high data rates, noise resilience and power efficiency. Hence, edge computing becomes more interesting due to the huge quantity of data generated by the video sensors. By extending the cloud where the things are (pre-processing data very close to the camera), edge computing enables us to avoid loading the network with some useless data, act rapidly on connected things (without cloud latency), and secure things [5]. The European Telecommunications Standards Institute (ETSI) has established many use cases for mobile edge computing. One interesting use case concerns video analysis, where the cameras can send high bandwidth video streams to a particular node (gateway) positioned at the edge of the network. The last perform video analysis and send low bandwidth streams to the network core [6], as depicted in Figure 1.



**Figure 1.** The European Telecommunications Standards Institute (ETSI) use case of the mobile edge computing applied to multimedia things.

### 1.2. Related Work

Mobile edge computing has several benefits compared to the traditional cloud-based computing paradigm [7]. It has been found that for running face recognition applications, the response time is reduced from 900 to 169 ms by moving computation from the cloud to the edge [8]. Moreover, for some other applications, such as wearable cognitive assistance, the response time is reduced and the energy consumption could also be reduced by 30% to 40% [9]. Chun, et al. in [10] have addressed cloud offloading in terms of energy/performance trade-off. Their prototype could reduce by 20× the running time and energy for tested applications. In terms of application use cases, behavioral analytics based on a multimedia network is one of the most common applications, where the behavioral analysis (a vehicle traveling in the wrong direction) is carried out in the camera node [7]. Also, in telemedicine, new IoMT frameworks have been developed as in [11] to empower patients via health promotion, assessing reductions in health deficits and improvements in quality of life.

However, design constraints related to the perception layer become stronger and should be taken into account when designing IoT applications in the context of edge computing. Two main issues should be considered when designing IoMT applications. The first one concerns the algorithmic level. Indeed, image processing algorithms are often sophisticated and could be simplified. In this

level, a selection of algorithms with lightweight implementation is mandatory. Then, to further reduce the algorithmic computational complexity, the use of approximate computing as in [12,13] can be beneficial. The second issue is regarding the architectural design. For high performance edge computing systems, a field programmable gate array (FPGA) can be used as a key component that can implement customized hardware logic and perform image processing tasks in real-time. Latest generation FPGAs provide small, low-cost and low-power devices that are ideal for edge computing applications.

Most of the state-of-the-art methods developed recently are dedicated exclusively to the software or hardware implementation of a single node, often called a smart camera. Regarding the software implementation, in [14], an autonomous smart INCA+ camera has been designed on a software processor for face detection. Another software implementation of smart cameras is described in [15], where a real-time object tracking system is developed to transmit only high-level information. In [16,17], a GPU implementation of a face recognition scheme using spectral correlation has been developed and a speedup of  $4\times$  was obtained compared to implementation based on a generic processor. On the other hand, the hardware implementation is often based on FPGA. Today, FPGAs are fabricated as system-on-chip (SoC) platforms containing hardware and software sections on the same chip. The hardware part is used when high speed and parallel computation are required, while the software is employed to add more flexibility and connectivity to the designed system. The use of platforms based on SoC architecture have gained particular attention [18–21]. However, the existing platforms in [18,19] do not use FPGA as the hardware accelerator of computationally complex algorithms and those of [18–21] do not consider the HW/SW co-design to evaluate the energy/quality of the designed systems. In [22], a smart camera node was designed in order to avoid falls by elderly persons in daily life. The designed prototype uses the HW/SW implementation for fall detection in a clock speed which is comparable to real-time systems. However, this implementation is dedicated to a single node and no communication functions were planned.

### 1.3. Objective of the Work

The objective of this paper is to build a proof-of-concept platform of edge computing for IoMT based on HW/SW co-design. The platform is composed of a camera sensor, a flexible SoC for image processing, and a communication function between nodes. Each IoMT node can be considered as a receiver or emitter by using a dedicated protocol. The goal of this work is to show how the use of the algorithm/architecture co-design can be useful for IoMT applications. For algorithmic issues, we started from the theory (approximate computing, chaos generation, fusion and diffusion schemes and so on) and finally obtained an experimental platform that can be used as a gateway in the context of the IoMT. For the architectural level, we used the Zynq design flow to speed up the conception time and we used FPGA as the hardware accelerator. We designed a digital gateway that can be used on the edge of the network in order to compress and encrypt multiple images. We take the simultaneous compression and encryption of multiple images as the use case. Compression is necessary for all image applications that require data transmission, as in the IoMT, while image encryption has become more and more important in the last few years. The proposed platform could be considered the cornerstone of IoMT applications, including video surveillance, face recognition, pedestrian detection, etc. Hence, this paper presents some details of an open and experimental platform that can be shared by the community in order to design image and video processing applications. The proposed platform contains the following processing: image acquisition (here images are stored in memories), compression, encryption and transmission. By algorithmic and architectural optimizations, we decreased the constraints related to time and energy consumption while maintaining at the same time an acceptable quality of experience. On the algorithmic level, we considered the approximate computing of the DCT used in the compression, and we employed a simple chaotic generators for image encryption. While, the architectural optimization is provided by hardware and software partitioning of the whole system. Four configurations of the system partitioning were considered so that the trade-off time/energy can

correspond to very broad range of applications. Finally, we should note that the presented design is flexible and can include other treatments such as face recognition, motion detection, and identification.

The rest of the paper is organized as follows: in Section 2, we detail the algorithmic considerations for IoMT applications. Section 3 is devoted to the hardware considerations. The HW/SW partitioning is analyzed in Section 4, where four configurations are illustrated. The experimental setup and the performance analysis are given in Section 5, before the conclusion.

## 2. Algorithmic Considerations for IoMT Applications

### 2.1. Need for Image Compression and Encryption

Today, multimedia content is rich and varied with the introduction of new formats, such as 4 K, 8 K, high dynamic range (HDR), wide color gamut (WCG), high frame rate (HFR)), and with the emergence of new applications such as virtual reality, video 360, etc. Coding, decoding, and down/uploading multimedia content are the most commonly used functionalities for any IoMT application. The low-power realization of these functions in mobile and portable things is therefore important for a reasonably good battery life. The total power consumption to run these applications comprises two major parts: (1) the power consumption in compression (or decompression), which is related to the video codec complexity, and (2) the power consumed in transmitting or downloading images, which is a function of the bit rate and depends on the compression efficiency of the codec. When considering the IoMT, the power consumed in compression and decompression is becoming increasingly dominant, since the transmission range is short. Therefore, there is a need to compress multimedia content with a relatively simple algorithm. Moreover, there is a need to protect multimedia content. A viable solution to simultaneously compress and encrypt multiple images has been presented in [23]. The method is based on the discrete cosine transform (DCT) and a specific spectral filtering. A set of  $n$  images was considered, as illustrated in Figure 2. Each image is transformed in the spectral domain by DCT, then filtered in order to keep a few low-frequency components. The latter are fused in a random manner to constitute a compressed and ciphered image. An improved version of the original work has been proposed in [24], where the size of the spectral block was adapted to the target image. The size adaptation is performed by applying the root-mean-square (RMS) time-frequency criterion. The straightforward application of this method in the context of edge computing suffers from two major drawbacks. The first concerns the high computational time of the compression process and the secret key calculations. The second is related to the low resistance against statistical and differential attacks, since the encryption scheme consists of shuffling spectral components of the compressed images without any substitution.

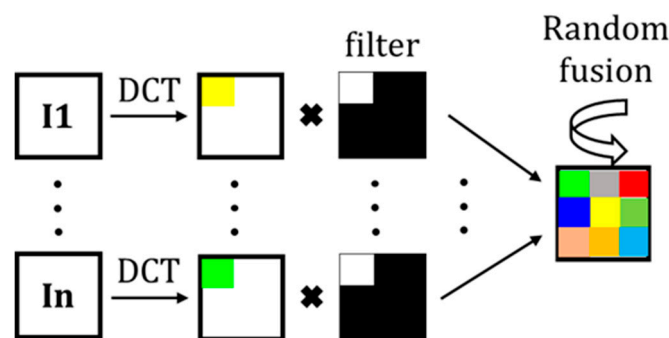


Figure 2. Scheme of the simultaneous compression, fusion and encryption.

### 2.2. Approximate Computing for Compression Improvement

The original method achieves compression by using DCT as many times as necessary to compress multiple images. Since DCT is computationally intensive and its complexity grows non-linearly with

the image size, in this paper we propose to use an approximation of DCT as in [12,13]. The main objective of the DCT approximation is to get rid of the multiplications that consume most of the power and computation time, and also to obtain a meaningful estimation of DCT. Indeed, the approximate DCT consists of replacing the twiddle factors of the basis vectors by  $\{-1, 0, 1\}$  elements. Hence, the multiplication operations used for the exact DCT computation are replaced by additions and subtractions, which have a lower latency and consume less energy and silicon area compared to the multiplications. It has been demonstrated that approximate DCT hugely decreases the computation requirements of the DCT and slightly decrease the quality of the reconstructed images. More specifically, 16-point 1-D approximate DCT consumes 64 adders and eight shift operators, while the exact DCT calculation requires 86 multipliers, 100 adders and eight shifters. In [13], it was found that the gain of energy per output coefficient is estimated to be equal to 16 with the approximate DCT compared to the exact one, and the gain in terms of area delay product is around 13. Moreover, the quality of experience is slightly degraded with the DCT approximation, since the PSNR of the reconstructed images with approximation is decreased by less than 0.8 dB when the compression ratio is lower than 1:2. Finally, to be compliant with the proposed method in [23], each  $N \times N$  image is compressed and then low-pass filtered to get  $N' \times N'$  compressed image. In this work, we considered four input images and  $N' = N/4$ , which implies a compression ratio of 16:1. It is worth noting that DCT approximation simplifies the compression process in terms of computational requirements, but has no effect on the size of the compressed bitstream.

### 2.3. Encryption Improvement

Traditional DES-, AES- and RSA-based encryption methods are not applicable to image encryption due to the computation speed of modern computers, and to the intrinsic properties of image like pixel correlation and voluminous file size. The chaotic generators have benefited from the growing need to secure data and at the same time they have demonstrated a great potential for image encryption. The robustness of chaotic-based cryptosystems is due to the intrinsic properties of chaotic generators like sensitivity to initial conditions and control parameters, non-periodicity, deterministic behavior and a large key space [25]. With these features, chaotic-based cryptosystems have excellent properties of diffusion and confusion, which are mandatory in cryptography. More particularly, one-dimensional non-linear iterative functions are capable of showing chaotic behavior under some conditions related to the tuning parameter and initial conditions. Moreover, their lower computational complexity has facilitated their adoption in many image encryption schemes based on several chaotic maps, such as Matthews [25], logistic [26], Henon [27], Skew Tent [28], Arnold [29], and Baker [30].

The flow chart of the encryption scheme used in this work was inspired by that of [26], where an improvement of the existing method in [24] is provided by using a software-based implementation. Extensive numerical simulations have been illustrated in [31] to prove the robustness of the proposed scheme against brute force, statistical, differential, and chosen plaintext attacks. Figure 3 depicts the encryption scheme, where  $a$  and  $s_0$  are the initial conditions of chaotic generators, which are developed in the next section. We should finally note that the approximate DCTs are from [12,13]. The method of multiple image compression and encryption is from [23] and the chaotic generators are used in the literature. There is no novelty in that issues are taken separately. In this work, we have used a set of algorithmic improvements simultaneously in order to alleviate the hardware constraints of the overall scheme. It was found that this choice is beneficial in terms of encryption efficiency and that the influence on the quality of reconstructed images is negligible.



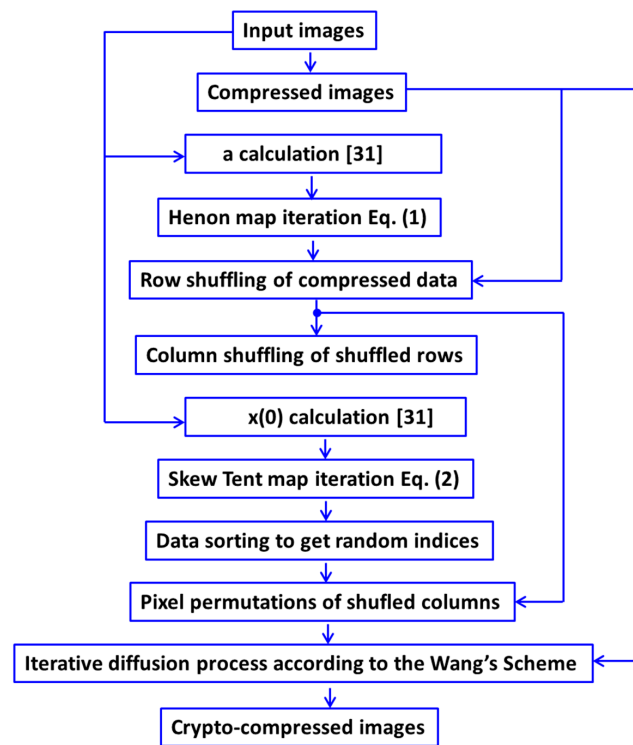


Figure 3. Flow chart of the encryption scheme.

### 3. Architectural Considerations for IoMT Applications

#### 3.1. Design Flow

The approximation of discrete cosine transform (DCT) is useful for reducing its computational complexity without significant impacts on its coding performance. Most of the existing algorithms for the approximation of the DCT target only the DCT of small transform lengths, and some of them are non-orthogonal. In this work, we use an approximation of 16-point DCT. For more details about the approximation, please consult the details in [13]. Hence, the double matrix multiplication used to perform the DCT is computed without multiplications. Only adders, subtracts, and shifters are required.

The design flow used with Xilinx tools includes HLS (high-level synthesis) steps performed by Vivado HLS. The goal is the fast prototyping of process elements to develop IP (intellectual property). With Vivado HLS, we transformed a C code into an HDL language. For our work, we used VHDL as the target language. The DCT needs a two-dimensional matrix multiplication applied twice. Before starting the computation of the second matrix multiplication, we need to use the results of the first matrix multiplication. Some directives are applied to the C code in order to have high-level control over the implementation. For our work, we only used three directives: `INLINE off`, `PIPELINE` and `INTERFACE`. The first directive (`INLINE`) increases potential parallelism at the cost of extra hardware. With `INLINE`, instead of treating the function as a single hardware unit, this directive makes HLS inline the function every time it is called. The second (`PIPELINE`) directive is useful for DCT calculation that uses several loops and for the matrix multiplication of two nested loops. We can apply the `PIPELINE` directive to loops. This causes inputs to be passed to the loop more frequently. A pipelined loop can process new inputs every M clock cycles, where M is the initiation interval (in our case, we left this interval at one). This directive increases the speed of the calculation, but also increases the number of logic elements. Finally, with the `INTERFACE` directive, we indicated to the HLS how to pass parameters between functions and between the PS and the PL. We applied this directive with `s_axi_lite` option on the top level function and on the parameters coming from the PS.

### 3.2. Hardware Description of the Chaotic Generators

In this paper, we proposed to use simple chaotic generators like Henon and Skew Tent maps in order to alleviate the hardware constraints on the edge computing system. The Henon map is a well-known discrete time dynamical system that exhibits chaotic behavior, and is defined by:

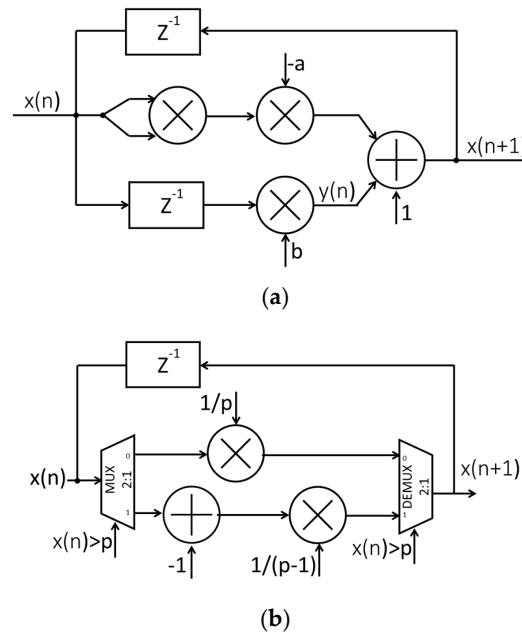
$$\begin{aligned} x_{n+1} &= 1 + y_n - a \cdot x_n^2 \\ y_{n+1} &= b \cdot x_n \end{aligned} \quad (1)$$

where  $a$  and  $b$  are the bifurcation parameters of the maps, and  $x_0$  and  $y_0$  are the initial conditions and the traditional practice is to use  $x_0 = y_0 = 0$ . In order to achieve chaotic behavior,  $a$  and  $b$  are often taken to be equal to 1.4 and 0.3, respectively. As explained before, in this work we used  $b = 0.3$  and  $a$  is a constant that depends on the content of input images.

The Skew Tent map, also called 1-D asymmetric tent map, is used in many cryptographic applications due its simplicity, large key space, and high key sensitivity. Its map is defined by:

$$s_{n+1} = \begin{cases} s_n/p & \text{if } s_n \in [0, p] \\ (1 - s_n)/(1 - p) & \text{if } s_n \in (p, 1] \end{cases} \quad (2)$$

where  $s_n \in [0, 1]$  is the state of the system and  $p \in (0, 1)$  is the control parameter. The initial condition  $s_0$  and  $p$  can be used as secret keys. In this work,  $p$  is fixed to 0.54321. Since the chaotic sequences will be generated on FPGA, the hardware description of both generators should be provided. The proposed signal flow graph (SFG) of the Henon and Skew Tent map are presented in Figure 4.



**Figure 4.** Proposed Signal Flow Graph (SFG) for: (a) Henon map, (b) Skew Tent map.

As shown in Figure 4, the proposed architecture of the chaotic generators can be obtained by simple logic and arithmetic operators, which are suitable for hardware implementation. Both generators are described in the VHDL language, but could be also described with Vivado HLS.

### 3.3. Hardware Setup

Our global hardware architecture is composed of two ZedBoard Xilinx evaluation boards, two Zigbee transceiver modules, and two High-Definition Multimedia Interface (HDMI) displays used

for debugging and prototype validation. The ZedBoard is based on Zynq XC7Z020. This device has a large memory (DDR3 and QPSI) and several communication interfaces (USB, Ethernet, Peripheral Module PMODs, HDMI, etc.). The ZedBoard SoC has an embedded ARM Cortex A9 processor commonly named PS (processing system) and a large FPGA fabric called PL (programmable logic). The communication between PS and PL is performed via AXI (advanced extensible interface) buses. An Analog Devices ADV7511 HDMI Transmitter provides a digital video interface to the ZedBoard. The HDMI display is connected to the SoC via HDMI port. The use of an HDMI display is inappropriate for the context of IoMT, but we have made this choice in order to add a debug functionality to the designed system. The HDMI display is split into two parts in order to show input images and de/(en)crypted images. We should notice that input images are stored in a RAM before their processing.

Image transmission is carried out using Xbee S6B wireless modules which provide simple serial to IEEE 802.11 connectivity. The Xbee wireless module is connected to the ZedBoard through the Pmod E. Sending and receiving data between the Zynq SoC and the Xbee module are done through the Universal Asynchronous Receiver/Transmitter UART transmission. The Zynq was configured such that the pins RX (reception) and TX (transmission) of the UART are connected to the Pmod E. The Pmod E is used because it is connected directly to the PS part of Zynq.

The UART frame consists of a start bit, a data byte, and a stop bit. The UART of the ZedBoard can send and receive data with a symbol rate of 921,600 bit/s. The symbol rate of the UART of Zynq is smaller than the XBee wireless module; it is the UART transmission which limits the maximum throughput. The creation of the UART connection is achieved by configuring the MIO 10 and 11 located on the PS transmission UART (named UART 0). The RX signal (input signal) of the UART is located in the MIO 10. The TX signal (output signal) of the UART is located in the MIO 11. After placement, routing and implementation of hardware design on Vivado, one can notice that MIO 10 and 11 are connected to the G7 and B4 ports, respectively.

## 4. Edge Computing HW/SW Co-Design

### 4.1. Hardware Description

The proposed crypto compression is implemented under Xilinx Vivado tool. The device is the ZedBoard manufactured by Digilent and the hardware description language is the VHDL. The vivado block design is illustrated in Figure 5. As indicated, nine blocks are used.

(1) The Zynq processing system is the software interface around the Zynq-7000 Processing System. (2) The HDMI IP for ZedBoard from Avnet: Reference Design implements the HDMI digital video interface in Xilinx FPGAs. (3) The reconfigurable compression and encryption IP is the proposed IP including DCT and chaos according to algorithms of Section 2. The compression DCT IP was developed under vivado HLS, while the chaos generators were described in VHDL. (4) The reconfigurable decompression and decryption IP is the proposed IP including Inverse Discrete Cosine Transform (IDCT) and chaos according to the algorithms in Section 2. The decompression DCT IP was developed under vivado HLS, while the chaos generators were described in VHDL. (5) The Axi GPIO (led, switch, push button) for the application: The switches select an index for the secret key, leds display the index, and push buttons select the emitter/receiver or software/hardware solutions. (6) The iic output of the HDMI IP to be used for the initialization. (7) The AXI peripheral block provides connections between the PL and PS. (8) The reset processing system to asynchronously restart the system. (9) The clock generation of 148.5 MHz for the HD video display.

The proposed compression/decompression modules have reconfigurable architectures. They allow the computation of 32-, 16- and 8-point DCT/IDCT by using the same computation elements. This is possible thanks to our proposed algorithm in [12] which was validated in [13].



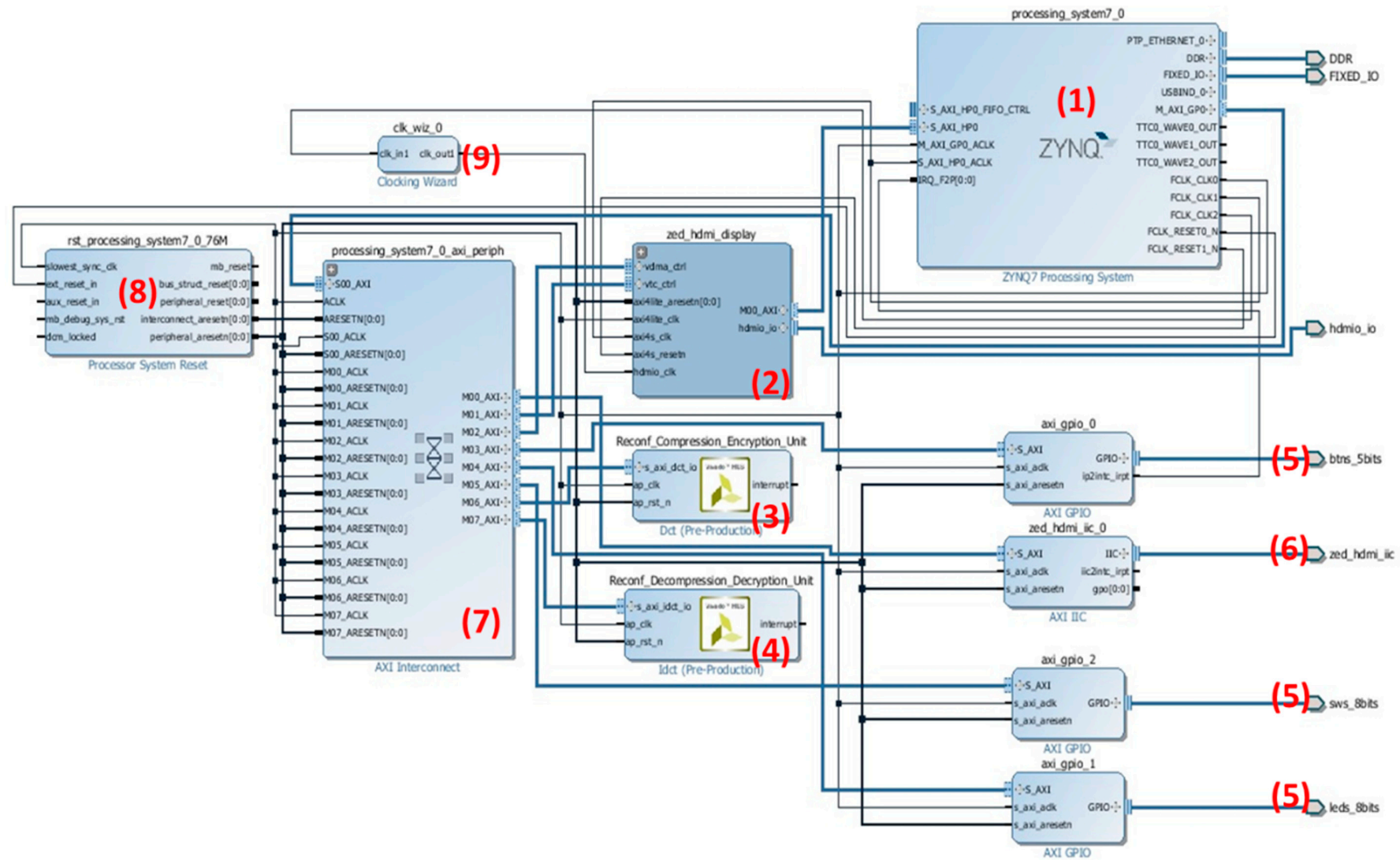


Figure 5. Block diagram of the proposed system implemented on ZedBoard.

#### 4.2. Hard/Soft Co-Design

The processing chains for transmitting and receiving crypto-compressed images are illustrated at the top of Figure 6. The transceiver protocol of the Xbee modules is implemented on the PS side, while the HDMI display is carried out on the PL by using the Xilinx IP for HDMI. The rest of the processing blocks can be implemented on software or on hardware. Therefore, we explored the HW/SW co-design of these blocks by creating four design variants that each implement a different configuration. The reason for creating several configurations is to move processing blocks from PS to PL and to quantify the system performances obtained with each configuration. All of these considered configurations are shown in Figure 7a,b for the transmission and reception chain, respectively. For the transmission chain, we considered a software-based implementation for Config1. We then included the chaos-based encryption and fusion on the PL side for Config2. Config3 moves the filtering on the PL. While the approximate computation of the DCT is added on the hardware for Config4. Hence, the block diagram of Figure 5 corresponds to the last configuration. Image decomposition to  $16 \times 16$  pixel blocks was done on the PS since no computation is required.

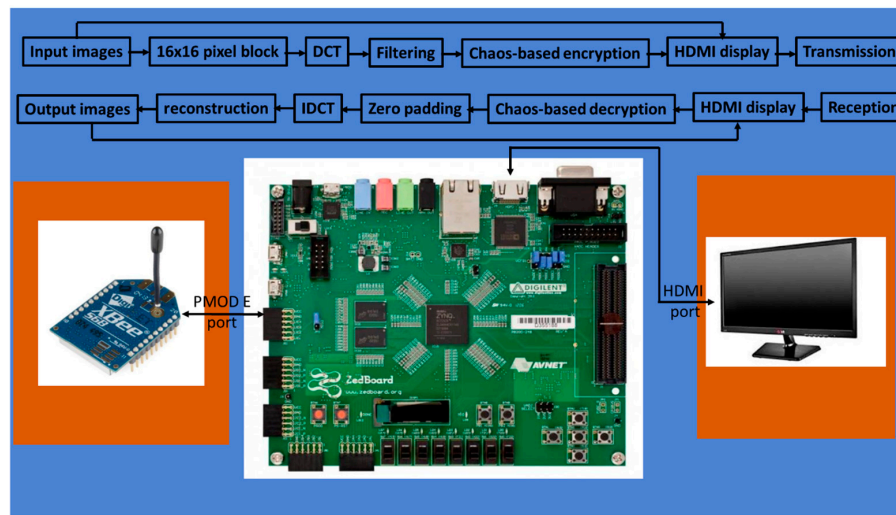
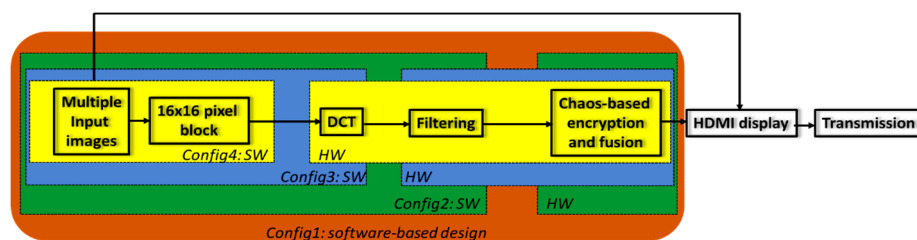


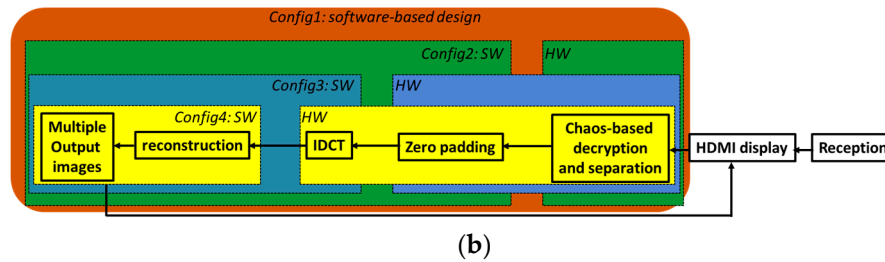
Figure 6. Hardware setup of the node architecture.

A similar approach was taken for the reception chain, in which four variants were considered. The first configuration is a software-based solution for decrypting and decompressing multiple images. Config2 decrypts and defuses received images by hardware implementation of the chaos-based decryption and separation. Config3 adds a zero padding block on the PL. The IDCT approximate computation is included on the PL side in Config4. Reconstruction of the four received images was done exclusively in software. The four configurations are depicted in Figure 7.



(a)

Figure 7. Cont.



**Figure 7.** Block diagram of the four HW/SW configurations. (a) Transmission, (b) reception chain.

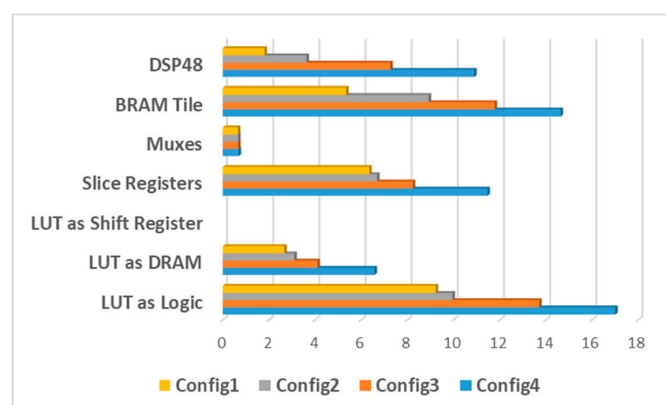
#### 4.3. Resource Utilization

The four previously illustrated variants were implemented on ZedBoard. The number of hardware resources and their ratio compared to available resources are depicted in Table 1.

**Table 1.** Resource utilization for several configurations.

	Config4		Config3		Config2		Config1	
	Number	Ratio	Number	Ratio	Number	Ratio	Number	Ratio
LUT as Logic	9053	17.01	7300	13.71	5300	9.96	4912	9.23
LUT as DRAM	195	6.58	195	4.11	195	3.11	195	2.67
LUT as Shift Register	950	6.58	520	4.11	346	3.11	270	2.67
Slice Registers	12,207	11.47	8766	8.24	7120	6.69	6766	6.35
Muxes	181	0.69	176	0.67	176	0.67	173	0.66
BRAM Tile	20.5	14.64	16.5	11.87	12.5	8.93	7.5	5.35
DSP48	24	10.9	16	7.27	8	3.63	4	1.81

We considered for resource utilization the number of LUTs (look up tables) synthesized as logic elements, the number of DRAM, registers, multiplexers, Block Ram BRAM memory tiles, and the DSP48 blocks. It can be seen that Config1 has the lowest resources utilization ratio, since it includes only the HDMI IP, and AXI buses to interconnect the PS and PL on hardware. In contrary, Config4 has the highest hardware utilization ratio. In all system variants, a maximum of 17% of ZedBoard resources is used, as illustrated in Figure 8. This means that the remaining resources can be used for other image processing blocks (e.g., detection, recognition, etc.).



**Figure 8.** Resource utilization ratio.

The on-chip power consumption profiling is provided in Table 2. The power numbers quoted are an estimation from a power calculator, not measured. These values are used instead of the measured power because they are easily obtained. When measuring the consumed power of a given algorithm

on FPGA, we have to take into consideration the static and the dynamic powers consumed during the run-time. Moreover, it is established that estimated power consumption can be efficient when comparing designs to each other. It was found that the estimated dynamic power consumption was slightly higher in Config4 compared with that of Config1. The use of DCT, IDCT, encryption and decryption with chaos generators, filtering, and zero paddings contribute to an increase in power consumption of about 100 mW.

Finally, we should mention that the data path delay obtained with the four configurations is equal to 2.25 ns. This means that when we incrementally included many processing blocks on the PL, the delay data path did not increase, since all the processing was pipelined. However, with Config1, since the major part of the processing was done in the PS, the execution time was much higher than that obtained with Config2, Config3, and Config4. More particularly, the computation time was measured for the compression and encryption of four gray scale images of  $512 \times 512$  pixels. When we used Config1 and Config4, the computation time was equal to 8.685 s and 897.28 ms, respectively. Therefore, the use of Config4 allowed a speedup of about 9.68.

According to the ETSI metrics for mobile edge computing in [32], the power efficiency  $\mu$  can be calculated as a relationship between the consumed power and the traffic.

$$\mu = \frac{\text{Power}}{\text{Traffic}} \quad (3)$$

For four gray scale images of  $512 \times 512$  pixels, the compressed bitstream contains 221 bits. The power efficiency obtained by including more processing blocks in the PL is about 213 nJ/bit, while the power efficiency obtained by including processing blocks in the PS is about  $1.64 \times 10^3$  nJ/bit. Thus, Config4 allowed a power efficiency about 7.7 times better than what we could obtain with Config1.

**Table 2.** Profiling of power consumption (W).

	Config4	Config3	Config2	Config1
Clock	0.056	0.05	0.044	0.041
LUT as Logic	0.03	0.02	0.02	0.01
Signal	0.042	0.028	0.021	0.012
Block RAM	0.028	0.021	0.013	0.007
MMCM	0.117	0.117	0.117	0.117
DSP48	0.016	0.011	0.008	0.005
I/O	0.038	0.038	0.038	0.038
Total dynamic power	0.00	0.00	0.00	0.00
Static power	0.171	0.168	0.167	0.166
Power efficiency	$2.13 \times 10^{-7}$			$1.64 \times 10^{-6}$

We finally should note that hardware performance may be enhanced when using predictive computation techniques as in [33] to predict zero quantized DCT coefficients, or also by a fine hardware optimization of arithmetic operations used in the DCT, as in [34].

## 5. Experimental Results

### 5.1. Experimental Setup

The proposed experimental testbed setup is presented in Figure 9. Two IoMT nodes were considered, where each node is composed of an Xbee transceiver, connected to a SoC-based ZedBoard by means of a cap card and a PMOD port. The HDMI display could not be used in case of real applications, but we used it in order to see encrypted and decrypted images. Moreover, we used a host computer connected to both nodes with two UART links in order to control the status of the node

at run time. This is useful since it can indicate the number of received bits. If the number of received bits is lower, the system will ask for retransmission.

Each node can be used as transmitter or receiver according to the user preference. Once the Xbee modules are configured in the same network and the ZedBoards are powered on and programmed, the user can press on the top push button of the first ZedBoard card and then press on the down push button of the other ZedBoard card. The first card is selected for transmission, while the second is the receiver side. Then the user can select to execute on the emitter node Config1 or Config4 by using the right or left push button and the eight switches for the encryption key. The input images as well as the crypto-compressed image are displayed on the HDMI display. The user can now press the middle button of the second card to open the reception, and then on the middle button of the first card to send crypto-compressed images. On the receiver side, the images can be decoded by Config1 or Config4 as in the encoding step. At this stage, the user has to indicate by switch selection the correct key in order to decrypt images. The test flow is shown in Figure 10.

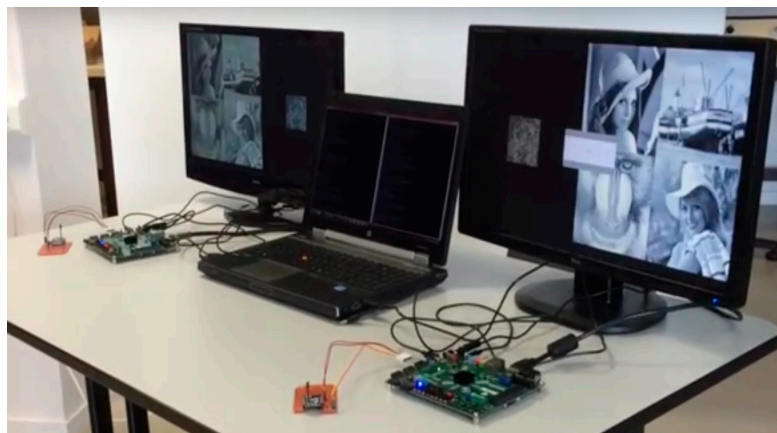


Figure 9. Experimental setup.

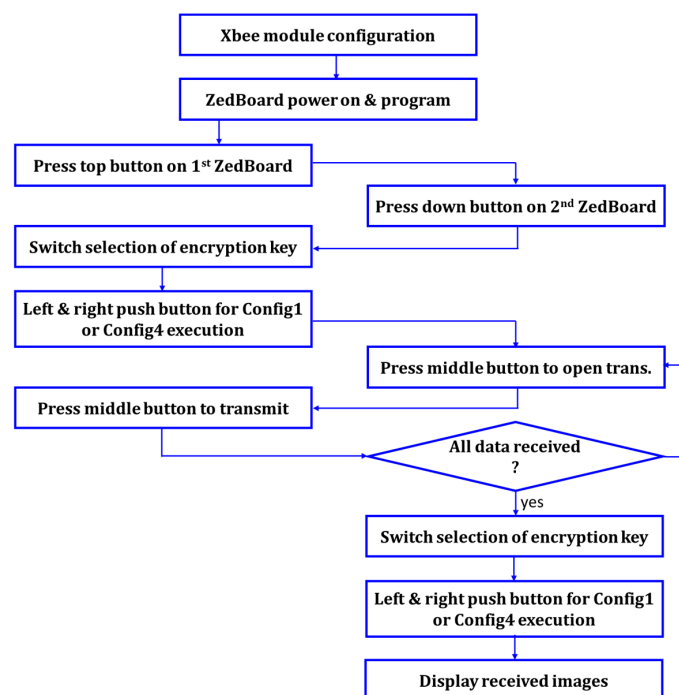
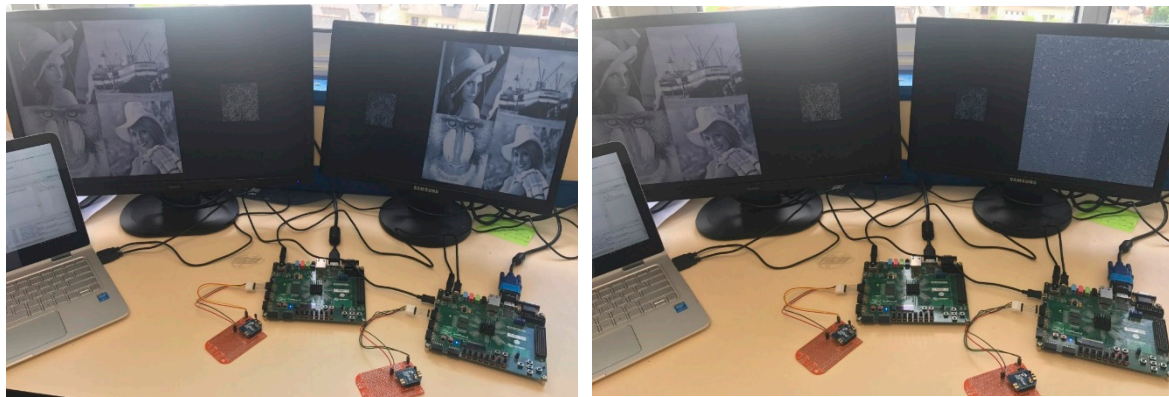


Figure 10. Test flow applied to the experimental setup.



## 5.2. Results

The FPGA-based implementation results for the IoMT application are presented in Figure 11. The original images and the decrypted images with the correct key are shown on the left, while the original images and the decrypted images with the wrong key are presented on the right side of Figure 11. A demo of this experimental setup can be found in [35].



**Figure 11.** FPGA implementation results. **Left** figure shows the case decryption with the correct key. **Right** figure shows the case of decryption with the wrong key.

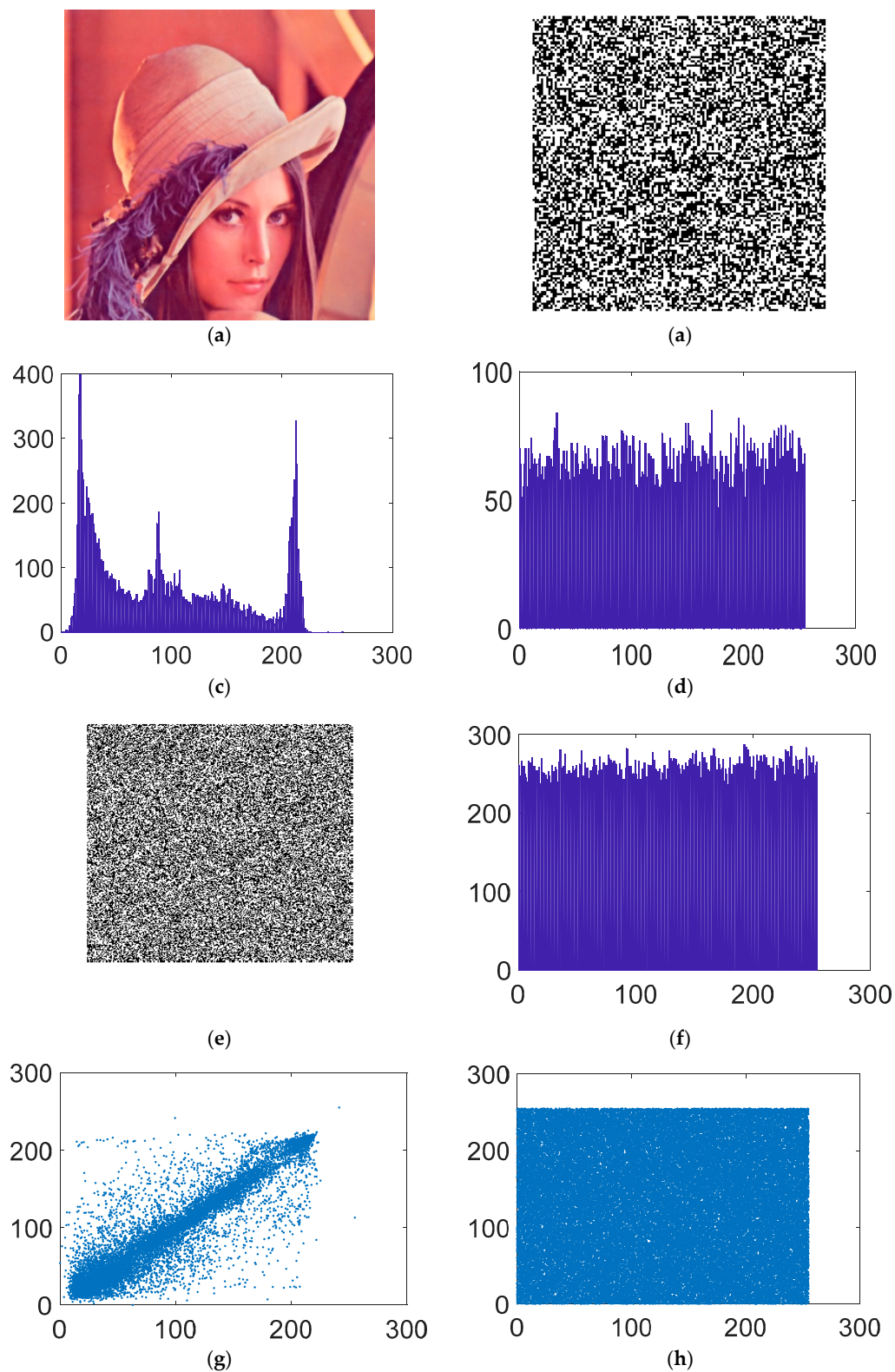
## 5.3. Performance Analysis

The use of approximate computing decreases the perceptual quality of the reconstructed images. We calculated the average peak signal-to-noise ratio (PSNR) of reconstructed eight-bit greyscale  $512 \times 512$  images. The forward and inverse approximate transforms are applied to obtain the reconstructed images and to compare PSNR with exact DCT-based compression. It was shown that the proposed approximate DCT results for the PSNR degradation were about 0.8 dB and less than 0.6 dB when a compression ratio of 50% and 12.5% were applied, respectively. Hence, we can conclude that there is a decrease in accuracy, but that the loss is still negligible.

Encryption performance evaluation was studied in [36], where some Matlab codes were developed for simulation purposes. In this work, plaintext and ciphered images were acquired with the processor and encryption analysis was addressed, as illustrated in Figure 12. One of the four input images is the Lena image shown in Figure 12a. Its corresponding ciphered image is shown in Figure 12b. The histogram of both images is depicted in Figure 12c (for plaintext) and Figure 12d (for the ciphered image). It was found that the ciphered image has no particular template that can indicate the content of the plaintext image. The four ciphered images and their corresponding histogram are illustrated in Figure 12e,f, respectively. It was found that the histogram is uniformly distributed, which can protect the image from statistical attacks. Finally, an adjacent pixel correlation analysis was carried out. The horizontal adjacent pixel correlation analysis is depicted in Figure 12g (for four plaintext images) and Figure 12h (for four ciphered images). It was found in the left image of Figure 12h that there is no correlation between adjacent pixels, which signifies a high security level. As in [36], we calculated the coefficient  $\rho$  of the adjacent pixel correlation and we found that its value was about  $5 \times 10^{-3}$ , which means that there is no correlation between adjacent pixels of the ciphered image.

In the case of the use of the correct key for the decryption, the image quality is subjectively evaluated. It was found that up to 10 m and 30 m of distance between the emitter and receiver, the quality of experience was about 90% and 80%, respectively. However, when the emitter and receiver were not in “line of sight”, the distance should not exceed 3 m for an acceptable quality of experience. Beyond 50 m of distance, the transmission is not guaranteed. Table 3 summarizes the obtained results.





**Figure 12.** Encryption performance of the proposed system. (a) plaintext image, (b) ciphered image, (c) histogram of (a), (d) histogram of (b), (e) ciphered and fused 4 images, (f) histogram of (e), (g) adjacent pixel correlation of (a), (h) adjacent pixel correlation of (e).

**Table 3.** Evaluation of the transmission quality.

Distance (m)	Line of Sight	Received Data (byte)	Duration (s)	Quality of Experience (%)
0.5	Yes	2,621,144	3	100
0.5	No	2,621,144	3	80
3	Yes	2,621,144	3	90
3	No	2,621,144	3	80
10	Yes	2,621,144	3	90
10	No	2,621,144	3.1	50
30	Yes	2,621,144	3.1	90
30	No	2,621,144	3.1	50
50	Yes	0		–

## 6. Conclusions

In this paper we proposed an experimental platform for mobile edge computing to be used in the context of the IoMT. The proposed platform could be considered as the cornerstone of IoMT applications, including video surveillance, face recognition, pedestrian detection, etc. We have demonstrated that by algorithmic and architectural optimizations, we can decrease the constraints related to time and energy consumption. Moreover, we have shown that by using a HW/SW co-design, we can decrease the power efficiency of the IoMT node. Future work will focus on including more image and video processing applications in the HW part of the node.

**Acknowledgments:** This work is financed by ISEN Yncrea Ouest.

**Author Contributions:** Thibault Chapel, Victor Dorez, Guenole Le Bougeant, and Antoine LeBotlan are students at ISEN Yncrea Ouest. They have developed a part of this work for their tutored project. Maher JRIDI has proposed, developed, and coordinated this project. The redaction of the paper is also provided by the corresponding author.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Cisco Global Cloud Index: Forecast and Methodology, 2015–2020. Available online: [https://www.cisco.com/c/dam/m/en\\_us/service-provider/ciscoknowledgenetwork/files/622\\_11\\_15-16-Cisco\\_GCI\\_CKN\\_2015-2020\\_AMER\\_EMEAR\\_NOV2016.pdf](https://www.cisco.com/c/dam/m/en_us/service-provider/ciscoknowledgenetwork/files/622_11_15-16-Cisco_GCI_CKN_2015-2020_AMER_EMEAR_NOV2016.pdf) (accessed on 11 January 2018).
2. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge Computing: Vision and Challenges. *IEEE Internet Things J.* **2016**, *3*, 637–646. [CrossRef]
3. IDC FutureScape: Worldwide Internet of Things 2017 Predictions. Available online: <https://www.idc.com/url.do?url=/getfile.dyn?containerId=US41910716&attachmentId=47261052&elementId=54464737&term=&position=5&page=9&perPage=25&id=2493c5c2-af0a-4031-bc18-0d42e52fd356> (accessed on 11 January 2018).
4. Alvi, S.A.; Afzal, B.; Shah, G.A.; Atzori, L.; Mahmood, W. Internet of multimedia things: Vision and challenges. *Ad Hoc Netw.* **2015**, *33*, 87–111. [CrossRef]
5. Cisco. *Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are*; White Paper; Cisco: San Jose, CA, USA, 2015.
6. Mobile-Edge Computing (MEC) Service Scenarios, ETSI GS MEC-IEG 004 V1.1.1 (2015-11). Available online: [http://www.etsi.org/deliver/etsi\\_gs/MEC-IEG/001\\_099/004/01.01.01\\_60/gs\\_MEC-IEG004v010101p.pdf](http://www.etsi.org/deliver/etsi_gs/MEC-IEG/001_099/004/01.01.01_60/gs_MEC-IEG004v010101p.pdf) (accessed on 11 January 2018).
7. Li, N.; Yan, B.; Chen, G.; Govindaswamy, P.; Wang, J. Design and implementation of a sensor-based wireless camera system for continuous monitoring in assistive environments. *Pers. Ubiquitous Comput.* **2010**, *14*, 499–510. [CrossRef]
8. Yi, S.; Hao, Z.; Qin, Z.; Li, Q. Fog computing: Platform and applications. In Proceedings of the 3rd IEEE Workshop Hot Topics Web System Technologies (HotWeb), Washington, DC, USA, 12–13 November 2015; pp. 73–78.

9. Ha, K.; Chen, Z.; Hu, W.; Richter, W.; Pillai, P.; Satyanarayanan, M. Towards wearable cognitive assistance. In Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services, Bretton Woods, NH, USA, 16–19 June 2014; pp. 68–81.
10. Chun, B.G.; Ihm, S.; Maniatis, P.; Naik, M.; Patti, A. CloneCloud: Elastic execution between mobile device and cloud. In Proceedings of the 6th Conference on Computer Systems, Salzburg, Austria, 10–13 April 2011; Review of the Crypto-Compression System and Improvements; pp. 301–314.
11. Orsini, M.; Pacchioni, M.; Malagoli, A.; Guaraldi, G. My smart age with HIV: An innovative mobile and IoMT framework for patient's empowerment. In Proceedings of the 2017 IEEE 3rd International Forum on Research and Technologies for Society and Industry (RTSI), Modena, Italy, 11–13 September 2017; pp. 1–6.
12. Jridi, M.; Ayman, A.; Meher, P.K. A Generalized Algorithm and Reconfigurable Architecture for Efficient and Scalable Orthogonal Approximation of DCT. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2015**, *62*, 449–457. [[CrossRef](#)]
13. Jridi, M.; Meher, P.K. A Scalable Approximate DCT Architectures for Efficient HEVC Compliant Video Coding. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *27*, 1815–1825. [[CrossRef](#)]
14. Kleihorst, R.; Abbo, A.; Schueler, B.; Danilin, A. Camera mote with a high-performance parallel processor for real-time frame-based video processing. In Proceedings of the 2007 IEEE Conference on Advanced Video and Signal Based Surveillance, London, UK, 25–28 September 2007; pp. 69–74.
15. Fleck, S.; Lanwer, S.; Straßer, W. A smart camera approach to real-time tracking. In Proceedings of the 2005 13th European Signal Processing Conference, Antalya, Turkey, 4–8 September 2005; pp. 1–4.
16. Ouerhani, Y.; Jridi, M.; AlFalou, A. Fast face recognition approach using a graphical processing unit “GPU”. In Proceedings of the 2010 IEEE International Conference on Imaging Systems and Techniques, Thessaloniki, Greece, 1–2 July 2010; pp. 80–84.
17. Ouerhani, Y.; Jridi, M.; AlFalou, A.; Brosseau, C. Graphics Processor Unit Implementation of Correlation Technique using a Segmented Phase Only Composite Filter. *Opt. Commun.* **2013**, *289*, 33–44. [[CrossRef](#)]
18. Rosello, V.; Portilla, J.; Riesgo, T. Ultra low power FPGA-based architecture for wake-up radio in wireless sensor networks. In Proceedings of the 37th Annual Conference on IEEE Industrial Electronics Society, Melbourne, Australia, 7–10 November 2011; pp. 3826–3831.
19. Berder, O.; Sentieys, O. PowWow: Power optimized hardware/software framework for wireless motes. In Proceedings of the 2010 23rd International Conference on Architecture of Computing Systems (ARCS), Hannover, Germany, 22–23 February 2010; pp. 1–5.
20. Engel, A.; Koch, A. Heterogeneous wireless sensor nodes that target the Internet of Things. *IEEE Micro* **2016**, *36*, 8–15. [[CrossRef](#)]
21. Gomes, T.; Pinto, S.; Salgado, F.; Tavares, A.; Cabral, J. Building IEEE 802.15.4 Accelerators for Heterogeneous Wireless Sensor Nodes. *IEEE Sens. Lett.* **2017**, *1*, 1–4. [[CrossRef](#)]
22. Senouci, B.; Charfi, I.; Heyrman, B.; Dubois, J.; Miteran, J. Fast prototyping of a SoC-based smart-camera: A real-time fall detection case study. *J. Real-Time Image Process.* **2016**, *12*, 649–662. [[CrossRef](#)]
23. AlFalou, A.; Brosseau, C.; Abdallah, N.; Jridi, M. Simultaneous fusion, compression and encryption of multiple images. *OSA Opt. Express* **2011**, *19*, 24023–24029. [[CrossRef](#)] [[PubMed](#)]
24. AlFalou, A.; Brosseau, C.; Abdallah, N.; Jridi, M. Assessing the performance of a method of simultaneous compression and encryption of multiple images and its resistance against various attacks. *Opt. Express* **2013**, *21*, 8025–8043. [[CrossRef](#)] [[PubMed](#)]
25. Matthews, R. On the derivation of a “Chaotic” encryption algorithm. *Cryptologia* **1984**, *8*, 29–41. [[CrossRef](#)]
26. Rostami, M.J.; Shahba, A.; Saryazdi, S.; Nezamabadi-pour, H. A novel parallel image encryption with chaotic windows based on logistic map. *Comput. Electr. Eng.* **2017**, *62*, 384–400. [[CrossRef](#)]
27. Su, Y.; Tang, C.; Chen, X.; Li, B.; Xu, W.; Lei, Z. Cascaded Fresnel holographic image encryption scheme based on a constrained optimization algorithm and Henon map. *Opt. Lasers Eng.* **2017**, *88*, 20–27. [[CrossRef](#)]
28. Talarposhti, K.M.; Jamei, M.K. A secure image encryption method based on dynamic harmony search (DHS) combined with chaotic map. *Opt. Lasers Eng.* **2016**, *81*, 21–34. [[CrossRef](#)]
29. Zhu, H.; Zhao, C.; Zhang, X.; Yang, L. An image encryption scheme using generalized Arnold map and affine cipher. *Opt. Int. J. Light Electron Opt.* **2014**, *125*, 6672–6677. [[CrossRef](#)]
30. Liu, Z.; Li, S.; Liu, W.; Liu, S. Opto-digital image encryption by using Baker mapping and 1-D fractional Fourier transform. *Opt. Lasers Eng.* **2013**, *51*, 224–229. [[CrossRef](#)]

31. Jridi, M.; AlFalou, A. Real-Time and Encryption Efficiency Improvements of Simultaneous Fusion, Compression and Encryption Method Based on Chaotic Generators. *Opt. Lasers Eng.* **2017**, *102*, 59–69. [CrossRef]
32. ETSI GS MEC-IEG 006: Mobile Edge Computing Market Acceleration MEC Metrics Best Practice and Guidelines V1.1.1. 2017. Available online: [http://www.etsi.org/deliver/etsi\\_gs/MEC-IEG/001\\_099/006/01.01.01\\_60/gs\\_MEC-IEG006v010101p.pdf](http://www.etsi.org/deliver/etsi_gs/MEC-IEG/001_099/006/01.01.01_60/gs_MEC-IEG006v010101p.pdf) (accessed on 15 September 2017).
33. Jridi, M.; Meher, P.K.; Alfalou, A. Zero-quantised discrete cosine transform coefficients prediction technique for intra-frame video encoding. *IET Image Process.* **2013**, *7*, 165–173. [CrossRef]
34. Jridi, M.; Alfalou, A.; Meher, P.K. Optimized architecture using a novel subexpression elimination on loeffler algorithm for DCT-based image compression. *VLSI Des.* **2012**, *2012*, 209208. [CrossRef]
35. Video Content. Demo. Available online: <https://www.youtube.com/watch?v=Wb3nQ336Kt4> (accessed on 15 September 2017).
36. Jridi, M.; Alfalou, A. Real-time and encryption efficiency improvements of simultaneous fusion, compression and encryption method based on chaotic generators. *Opt. Lasers Eng.* **2018**, *102*, 59–69. [CrossRef]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).