A VLSI Implementation of a New Simultaneous Images Compression and Encryption Method

Maher Jridi, Ayman Alfalou

A VLSI Implementation of a New Simultaneous Images Compression and Encryption Method

Maher Jridi, *Member, IEEE*, and Ayman AlFalou, *Senior, IEEE*
Département d'optoélectronique, Laboratoire L@bISEN
20 rue cuirassé Bretagne CS 42807, 29228 Brest Cedex 2, France
e-mail: (maher.jridi@isen.fr and ayman.alfalou@isen.fr)

***Abstract*—In this manuscript, we describe a fully pipelined single chip architecture for implementing a new simultaneous image compression and encryption method suitable for real-time applications. The proposed method exploits the DCT properties to achieve the compression and the encryption simultaneously. First, to realize the compression, 8-point DCT applied to several images are done. Second, contrary to traditional compression algorithms, only some special points of DCT outputs are multiplexed. For the encryption process, a random number is generated and added to some specific DCT coefficients. On the other hand, to enhance the material implementation of the proposed method, a special attention is given to the DCT algorithm. In fact, a new way to realize the compression based on DCT algorithm and to reduce, at the same time, the material requirements of the compression process is presented. Simulation results show a compression ratio higher than 65% and a PSNR about 28 dB. The proposed architecture can be implemented in FPGA to yield a throughput of 206 MS/s which allows the processing of more than 30 frames per second for 1024x1024 images.**

I. Introduction

Reconfigurable hardware in the form of Field Programmable Gate Arrays (FPGAs) have been proposed to obtain high performance and economical price to implement image processing applications like face recognition, detector or airport security [1]. For these applications, we need to use communication systems with a good security level (encryption) and an acceptable transmission rate (compression rate). In the literature, several encryption and compression techniques can be found. However, for some applications such as detectors, the encryption and the compression techniques cannot be deployed independently and in a cascade manner without considering the impact of one technique over another [2]. To solve this problem, we developed a new technique to simultaneously compress and encrypt multiple images [3].

The main idea of our approach consists, firstly, in multiplexing the spectra of different transformed images (to be compressed and encrypted) by a Discrete Cosine Transform (DCT) and secondly in implementing the proposed system in FPGA. Consequently, special attention is given to the DCT algorithm implementation in the context of image compression. In fact, the DCT is the heart of the proposed compression and encryption method. It has been widely used in speech and image compression due to its good energy compaction [4]. However its computational requirement is a heavy burden in the real time simultaneous compression and encryption application. Different DCT architectures have been proposed to exploit signal proprieties to improve the tradeoff between computation time and hardware requirement. Among these, the DCT algorithm proposed by Loeffler [5], has opened a new area in digital signal processing by reducing the number of required multiplications to the theoretical limit. In this paper we use the DCT architecture for image compression and we demonstrate that the number of arithmetic operators can be reduced without dramatically decreasing the compressed image quality. In fact, by exploiting the spacial correlation of input images, we can reduce the number of arithmetic operators from 11 multipliers and 29 adders to 4 multipliers and 14 adders. Simultaneously, in order to perform the security level, a second stage a using random number generator is applied to some specific DCT outputs.

This paper is organized as follows: the description of the proposed simultaneous compression and encryption method is presented in section II. Section III is dedicated to the optimization of the DCT architecture. Implementation results using FPGA are illustrated in the last section before conclusion.

II. Method principle

We proposed a new technique, based on our methods presented in [3] and [6], which can carry out compression and simultaneous encryption using random number generator and Discrete Cosine Transform (DCT). The main idea of our approach consists in multiplexing the spectra of different transformed images separately by a DCT.

The choice of the DCT is justified by the use of the DCT in many standards such as JPEG [7], MPEG [8] and ITU-T H261 [9]. Moreover, we need fewer DCT coefficients than DFT coefficients to get a good approximation to a typical signal [10]. In fact, by applying the DCT, the most of the signal information tends to be concentrated in a few low-frequency components. Consequently, the higher frequency coefficients are small in magnitude and can be ignored in the compression and encryption process.

Fig. 1 presents the synoptic diagram of the proposed compression and encryption system. In the left side, 4 input gray level images are presented ($P1$, $P2$, $P3$, $P4$). To apply to each of these images a full parallel DCT algorithm, we need to parallelize each image by blocks of 8 pixels. This operation can be done by a serial to parallel block composed by 8 flip-flops.

Then, 4 DCT blocks are used to transform the 4 input

978-1-4244-6493-7/10/$26.00 ©2010 IEEE

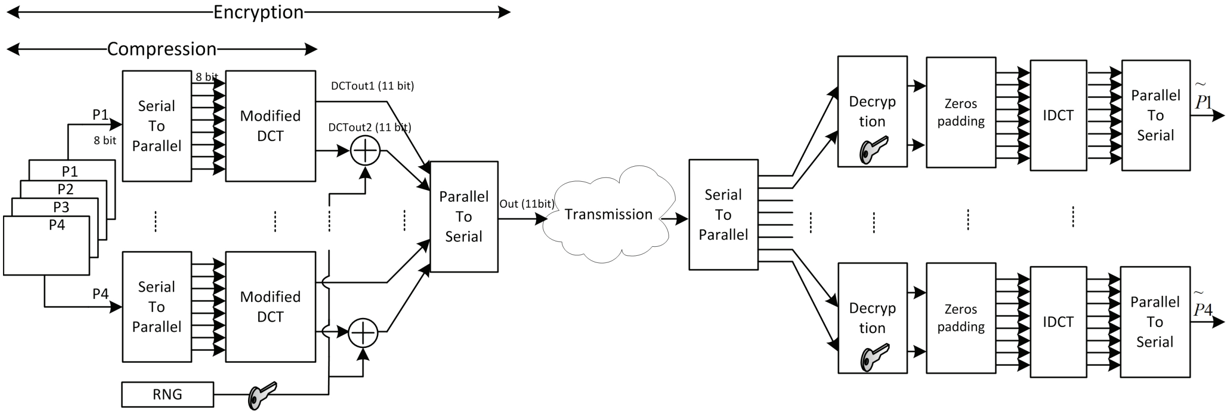

Fig. 1. Synoptic diagram of the proposed compression and encryption system

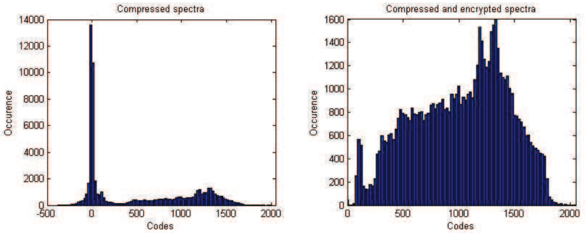

Fig. 2. Histogram of DCT outputs

images. These DCTs are employed to regroup lower frequency components of the DCT. In fact, by taking into account only the first and the second DCT outputs among 8, we get a good approximation of input pixels. Hereafter, we use the following notations: $DCTout1$ for the first DCT output and $DCTout2$ for the second one.

Concerning the encryption process, the algorithm is based on the next observation: the multiplexed spectrum plane presents alternatively one high value $DCTout1$ followed by one low value $DCTout2$. In terms of security, we can imagine that behind the histogram on the left of Fig. 2 we can find DCT coefficients. In fact, as it will be explained in the next section, the low value of the $DCTout2$ is due to the spatial correlation between 8 successive pixels presented in input images.

In order to ensure a good encryption level against any hacking attempt, we propose to add to $DCTout2$ a positive random value to have a data values close to $DCTout1$. As mentioned in Fig. 2, the addition of a random number can drastically modify the characteristic spectral distribution of the DCT

The security key will be sent separately as a private encryption key. Once secure and compressed information safely reach the authorized receiver, the image extraction can be easily done by reversing the various steps used in the whole process:

- Subtract the received image by the security key;
- Add 6 zeros to each block (zeros padding);
- Run an Inverse DCT (IDCT).

III. DCT ARCHITECTURE

The DCT is the heart of the proposed compression and encryption method. Therefore, an optimization of the whole proposed method requires a DCT optimization. In this section, we present the modified DCT architecture and the data encoding of DCT outputs in order to allow an acceptable compression ratio and a relatively high image quality.

A. Related Work

The N-point DCT of N input samples $x(0)$,...,$x(N-1)$ is defined as:

$$X(n) = \sqrt{\frac{2}{N}} C(n) \sum_{k=0}^{N-1} x(k) \cos \frac{(2k+1)n\pi}{2N} \qquad (1)$$

where $C(0) = 1/\sqrt{2}$ and $C(n) = 1$ if $n \neq 0$.

In literature, many fast DCT algorithms are reported. In [11], the authors show that the theoretical lower limit of 8-point DCT algorithm is 11 multiplications. Since the number of multiplications of Loeffler's algorithm [5] reaches the theoretical limit, we use this algorithm as the reference to this work. A modified signal flow graph of 8 inputs 2 outputs DCT is presented in Fig. 3. We will explain the reasons of modifications in the next section.

In [12] one realization based on Loeffler algorithm is shown. A low power design is obtained with this algorithm.

In [13] use the recursive DCT algorithm and their design requires less area than conventional algorithms. The authors of [13] use Distributed Arithmetic (DA) multipliers and show that N-point DCT can be obtained by computing N N/2-point inner products instead of computing N N-point inner products. In [14], a new DA architecture called NEDA is proposed, aimed at reducing the cost metrics of power and area while maintaining high speed and accuracy in digital signal processing (DSP) applications. Mathematical analysis proves that DA can implement inner product of vectors in the form of two's complement numbers using only additions, followed by a small number of shifts at the final stage. Comparative studies show that NEDA outperforms widely used approaches such as

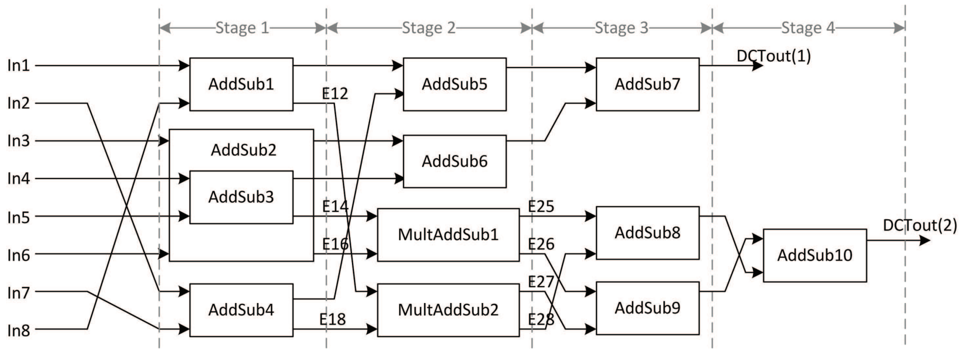

Fig. 3. Signal flow graph of 8 inputs 2 outputs DCT

multiply/accumulate (MAC) and DA in many aspects.
In this paper, we will not optimize the arithmetic operators but we present a new algorithm based on Loeffler one and makes dependency between the compression ratio and the material complexity. Consequently, optimizations in [12], [13] or [14] can be used with the presented new algorithm.

B. Proposed DCT architecture

The circuit of the proposed algorithm of the DCT is inspired from the Loeffler one. Therefore, some similarities exist between these two circuits. For example, we propose to compute DCT outputs on four stages as shown in Fig. 3. Each stage contains some arithmetic operations. The first stage is performed by 4 AddSub (adder and subtracter) blocks while the second one is composed of 2 AddSub and 2 MultAddSub (Multiplier, Adder and Subtracter) blocks. The details of these blocks are shown in Fig. 4.
Moreover, some important differences should be mentioned. Since the proposed DCT circuit accepts 8 pixels per clock cycle and delivers 2 outputs against 8 outputs in the original Loeffler algorithm, we decide to change the DCT architecture to compute only necessary DCT coefficients $DCTout1$ and $DCTout2$.
It should be outlined that traditionally, one possible manner for compression based on DCT algorithm consists in computing all DCT outputs (8 outputs for 8 pixels) and after that some special points are selected. The whole computation time and latency are therefore very high.
The changes in DCT architecture are as follows:
First, only necessary paths to compute $DCTout1$ and $DCTout2$ are kept as shown in the Fig. 3. Thus, we can economize 5 multipliers, 2 adders and 2 subtracter compared to the Loeffler architecture.
Then, we can notice that in Fig. 3 only the first outputs of AddSub5 to AddSub10 are used. Therefore, the AddSub5 to AddSub10 blocks are reduced to 1 adder per block. Consequently, 6 additional subtracters can be saved.
Finally, the $DCTout2$ can be written as follows:

$$\begin{aligned} DCTout2 &= (E25+E28)+(E26+E27) \\ &= (E18*cos\,(\pi/16)+E12*sin\,(\pi/16)) \\ &+ (-E16*sin\,(3\pi/16)+E14*cos\,(3\pi/16)) \\ &+ (-E18*sin\,(\pi/16)+E16*cos\,(\pi/16)) \\ &+ (E16*cos\,(3\pi/16)+E14*sin\,(3\pi/16))(2) \end{aligned}$$

After factorizations, $DCTout2$ can be written as follows:

$$\begin{aligned} DCTout2 &= E18*\overbrace{(cos\,(\pi/16)-sin\,(\pi/16))}^{c_1} \\ &+ E16*\overbrace{(cos\,(3\pi/16)-sin\,(3\pi/16))}^{c_2} \\ &+ E14*\overbrace{(cos\,(3\pi/16)+sin\,(3\pi/16))}^{c_3} \\ &+ E12*\overbrace{(cos\,(\pi/16)+sin\,(\pi/16))}^{c_4} \qquad (3) \end{aligned}$$

According to these equations, the MultAddSub blocks of Fig. 3 can be replaced by more simple blocks. In fact, the original block requires 1 adder, 1 subtracter and 4 multipliers to compute the outputs. Loeffler reduces the number of arithmetic operators to 3 multipliers and 3 adders per block. In this work, as presented in Fig. 4 the MultAddSub block can be replaced by only two multipliers. Like this, we economize 6 adders and 2 multipliers.
Using these three optimization levels, the proposed DCT architecture requires 4 multipliers and 14 adders to compute relevant and representative data outputs for image compression against 11 multipliers and 29 adders proposed by Loeffler.

C. Data encoding

The minimization of data length implies less computation, and consequently, lower power consumption and higher speed. On the other hand, truncating introduces errors at the outputs and degrades the PSNR (Peak Signal to Noise Ratio). Thus a trade-off between power and PSNR is made. In the input side of the proposed method, the pixels of input images are encoded using unsigned 8-bit values. In the output side, $DCTout1$ contains the major part of the information, so this value must be encoded by the maximum number of bits. $DCTout1$ results in 3 successive additions of input pixels. Consequently, and considering the carry of each addition, the $DCTout1$ is encoded by using 11 bits.
For the constant c_i, $i \in [1,4]$ of (3) we can employ the

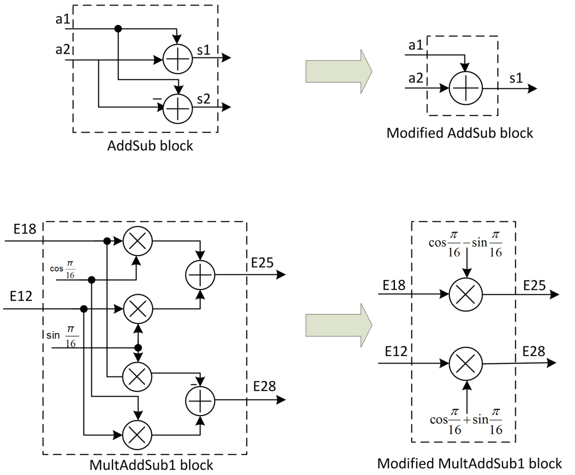

Fig. 4. Arithmetic operator blocks

coefficients encoding used in [15] and detailed by the next equation:

$$\widetilde{c_i} = round\left(c_i * 2^{8-1} - 1/c_{i,max}\right) \tag{4}$$

For $DCTout2$ encodage, we can take into account the spatial correlation of images. In fact, we can suppose that for image size of 256*256 pixels or higher, the block of each 8 adjacent pixels of the same line are very correlated and have a very close value. Consequently, signals E12, E14, E16 and E18 from Fig. 3 which are the subtraction of input image pixels from $In1$ to $In8$ have a very low value. In the same way, the signals E25 to E28 also have a lower value compared to input pixel images. Consequently, we can limit the $DCTout2$ by $-FS \leq DCTout2 \leq FS$ where $FS = 2^8$ is the full scale of the input images.

On the other hand, for encryption process, the encrypted $DCTout2$ have to be close to $DCTout1$ which is in $\left[0, 2^3 * FS\right]$. Consequently, the $DCTout2$ is added to $\tilde{r}$, a generated random number expressed by the next equation:

$$\tilde{r} = round\left(FS + 6 * FS * r\right) \tag{5}$$

where r is an uniformly distributed pseudorandom number, $0 \leq r \leq 1$. Finally, $DCTout1$, $DCTout2$ are encoded using 11 bits.

The data encoding allows a high compression ratio. In fact, since the 4 spectra will be regrouped in a single plane, the consequent compression ratio for input image sizes of S is:

$$R = \left(1 - \frac{S * 11bits}{4 * S * 8bits}\right) \times 100\% = 65.62\% \tag{6}$$

Moreover, for higher compression rate, we can use the correlation between the neighboring pixels to encode the second DCT coefficient using only 7 bits. The obtained compression ratio can achieve a value of about 72%

Fig. 5. Input images

IV. Validation

A. Methodology

A fixed point Matlab Simulink model has been established to validate the proposed method. This step is very important to validate the the algorithm structure before the material implementation. Concerning the description language, we decide to use VHDL rather than DIME-C and Mitrion-C which produce less efficient hardware design. In fact, DIME-C and Mitrion-C are much easier to program than VHDL, but visibility to hardware details allowing optimizations is lost due to abstraction [16]. In addition, the VHDL standard language gives the choice of implementing target devices (FPGA family, CPLD, ASIC) at the end of the implementation flow. It means that the models reported here are synthesized and may be implemented on arbitrary technologies [17].

Simulation results of the VHDL model are reported in Fig. 5 and Fig. 6 and show that original images are rebuilt correctly with a PSNR average between four images about 28 dB.

B. FPGA implementation

The original DCT Loeffler architecture and the proposed one in this article have been implemented in the same kind of FPGA boards, that is, Virtex 5 of xc5vlx330t. In order to illustrate the differences in hardware consumption, the FPGA implementation results are presented in Table 1. From this comparison we can notice that the proposed DCT architecture reduces the area consumption (slices and Look Up Tables, LUTs) at a rate higher than 50 %.

Furthermore, the throughput, expressed in Millions of Samples per second (MS/s), presents a light increase compared to the Loeffler architecture. The throughput of 206 MS/s allows the processing of more than 30 frames per second. Finally, it should be pointed out that the modified DCT and the proposed compression and encryption method have the same throughput: the proposed method is for sure fully pipelined.

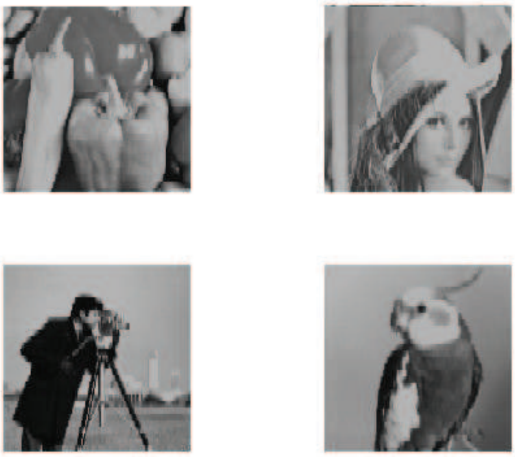

Fig. 6. Output images

TABLE I
SYNTHESIS RESULTS

Characteristics	Loeffler	Modified DCT	Compression method
Slice registers	507	247	1536
Slice LUTs	1293	492	2058
Fully used LUT	316	162	955
Throughput (MS/s)	191.867	206.423	206.423

V. CONCLUSION

In this manuscript, a new method of simultaneous compression and encryption based on a DCT transformation is presented. An optimized DCT algorithm is proposed to reduce real time application requirements. This algorithm needs only 4 multiplications to compute relevant DCT output data. The FPGA implementation of the whole method shows improvements in terms of throughput, area and power consumption. To prove the good performances, the proposed algorithm is compared favorably with several existing methods.

ACKNOWLEDGMENT

The author thanks Dominique Maratray for her help and advice.

REFERENCES

[1] M.W. James, *An Evaluation of the Suitability of FPGAs for Embedded Vision Systems*, CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, pp. 131-137, 2005.

[2] A. Alfalou and C. Brosseau, *Image Optical Compression and Encryption Methods*, OSA: Advances in Optics and Photonics, vol 1, pp. 589-636, 2009.

[3] A. Alfalou, M. Elbouz, M. Jridi and A. Loussert, *A new simultaneous compression and encryption method for images suitable to optical correlation*, Optics and Photonics for Counterterrorism and Crime Fighting V, edited by Colin Lewis, Proc. of SPIE Vol. 7486, 74860J-1-8, 2009.

[4] K. Rao and P. Yip, *Discrete cosine transform algorithms advantages applications*, Academic Press, New York, 1990.

[5] C. Loeffler and A. Lightenberg and G.S. Moschytz , *Practical fast 1-D DCT algorithm with 11 multiplication*, IEEE, ICAPSS, pp. 988-991, May 1989.

[6] A. Loussert, A. Alfalou, R. El Sawda, and A. Alkholidi, *Enhanced System for image's compression and encryption by addition of biometric characteristics*, International Journal on Software Engineering and Applications, pp. 111-118, 2008.

[7] ISO/IEC JTC1/SC2/WG8, JPEG-8-R8, *JPEG technical specification*, 1990.

[8] ISO/IEC JTC1/SC2/WG11, MPEG 90/176, *Coding of moving picture and associated audio*, 1990.

[9] ISO/IEC DIS 10 918-1, *Digital compression and coding of continuous-tone still image*, 1992.

[10] K. F Blinn, *What's the deal with the DCT?*, IEEE Computer Graphics and Applications, pp. 78-83, July 1993.

[11] P. Duhamel and H. H'mida, *New 2^n DCT algorithm suitable for VLSI implementation*, IEEE, ICAPSS, pp. 1805-1808, November 1987.

[12] C.Y Pai, W.E. Lynch and A.J. Al-Khalili, *Low-Power data-dependant 8x8 DCT/IDCT for video compression*, IEE, Proceedings. Vision, Image and Signal Processing, Vol. 150, pp. 245-254, August 2003.

[13] S. Yu and E.E. Swartzlander Jr, *DCT implementation with distributed arithmetic*, IEEE Transactions on Computers, Vol. 50, No.9, pp, 985-991, September 2001.

[14] A. Shams , A. Chidanandan, W. Pan and M.A Bayoumi, *NEDA : A low-power high-performance DCT architecture*, IEEE transactions on signal processing, Vol. 54, No.3, pp, 955-964, 2006.

[15] E. Darakis and J.J. Soraghan, *Reconstruction domain compression of phase-shifting digital holograms*, Journal of Applied Optics, Vol. 46, pp. 351-356, January 2007.

[16] S. H. Park, D. R. Shires and B. J. Henz, *Coprocessor computing with FPGA and GPU*, 3rd ed. DoD HPCMP Users Group Conference. Seattle, WA, pp. 366-370, July 2008.

[17] M. Jridi and A. AlFalou , *Direct digital frequency synthesizer with CORDIC algorithm and Taylor series approximation for digital receivers*, European Journal of Scientific Research, Vol. 30, No. 4, pp. 542-553, August 2009.