International Journal of Soft Computing 2 (2): 292-301, 2007

A Blind Watermarking Algorithm Implementation for Digital Images and Video

Sourour Karmani, Ridha Djemal and Rached Tourki
Laboratoire d'Electronique et de Microélectronique, Faculté des Sciences de Monastir,
Avenue de l'Environnement, 5019 Monastir-Tunisie

Abstract: With the growth of new image technologies, copyright protection becomes an important issue to preserve digital images and video properties. In this paper, we propose an improved implementation of a blind object watermarking scheme for images and video streams. To make the watermark robust and perceptual invisible, we have used a two dimensional wavelet transform as a transformation domain for both image and watermark. We have also used an optimized quantization and replacement strategies within the insertion process. A hardware architecture is designed and tested in order to evaluate the performance of our proposed watermarking algorithm. This architecture has been implemented on the Altera Stratix II prototyping board.

Key words: Watermarking, 2D-DWT, insertion process, RTL implementation, FPGA

INTRODUCTION

Thanks to the development of electronic technologies and telecommunications and also to the amelioration of image and signal processing techniques, the digital information could be accessible and easily treated. So, users can download, store and retransmit digital documents such as images, video and audio without respecting copyright of ownerships. This has motivated many researchers to develop many digital watermarking techniques. The watermarking consists on the insertion in a digital document an invisible and indelible watermark. Once it is inserted in the image, the watermark must be too difficult to be removed. Il should also be robust to the following operations:

- Signal processing
- Geometric distortion such as compression, adding noise and scaling.
- Conversion techniques D-A/A-D,

The insertion of the watermark could be done in the spatial domain (Bender *et al*., 1996; Osborn *et al*., 1993; Van *et al*., 1994) directly on the images pixels or in the transformed domain (Chan *et al*., 2001; Mohanty, 2000). (DCT: Discrete Cousin Transform, DFT: Discrete Fourrier Transform and DWT: Discrete Wavelet Transform). The multi-resolution technique based on the wavelet transform (Xie and Arce, 1998;Hus and Wu, 2000; Dielt *et al*., 2003) has become an interesting issue in last years because it gives a spacio-frequentiel vision, which optimise the insertion and brings a high security against attacks. The rest of the study is organised as follow.

Related works: Many watermark algorithms have been proposed to address the ownership protection for image and video stream. These techniques can uses an additive embedding of a spread-spectrum sequence or a quantize-and-replace strategy. Either the image approximation or the details sub-bands of the wavelet domain can be modified.

Spread-spectrum watermarking: Based on the work of Cox (1997) in the DCT domain, Kim and Moon (1999) utilizes DWT coefficients of all sub-bands including the approximation image to equally embed a random Gaussien distributed watermark sequence in the whole image. Perceptually significant coefficients are selected by an adaptatif level to achieve high robustness. However, the location of the watermark information is not protected and open for malicious attacks. Following the design of his multi-threshold wavelet coding scheme, (Wang and Jay, 1998) proposes a watermarking algorithm that refines Kim's thresholding scheme and selects significant coefficients on a per sub-bands basis. Here random skipping of significant coefficient is discussed as a means to achieve non-invertibility and to improve watermark security although this will also limit the robustness and capacity of the scheme.

Quantization-based watermarking: Instead of adding a signal to the selected coefficients, sets of coefficients are quantized to represent the binary watermark information (Kunder, 1999; Xie and Arce, 1998). Xie's approach selects non-overlapping sets of coefficient of size 3x1 from the wavelet domain's approximation image. The median coefficient of each set is then quantized to embed one bit of watermark information. The algorithm proposed

Corresponding Author: Sourour Karmani,Laboratoire d'Electronique et de Microélectronique, Faculté des Sciences de Monastir, Avenue de l'Environnement, 5019 Monastir-Tunisie

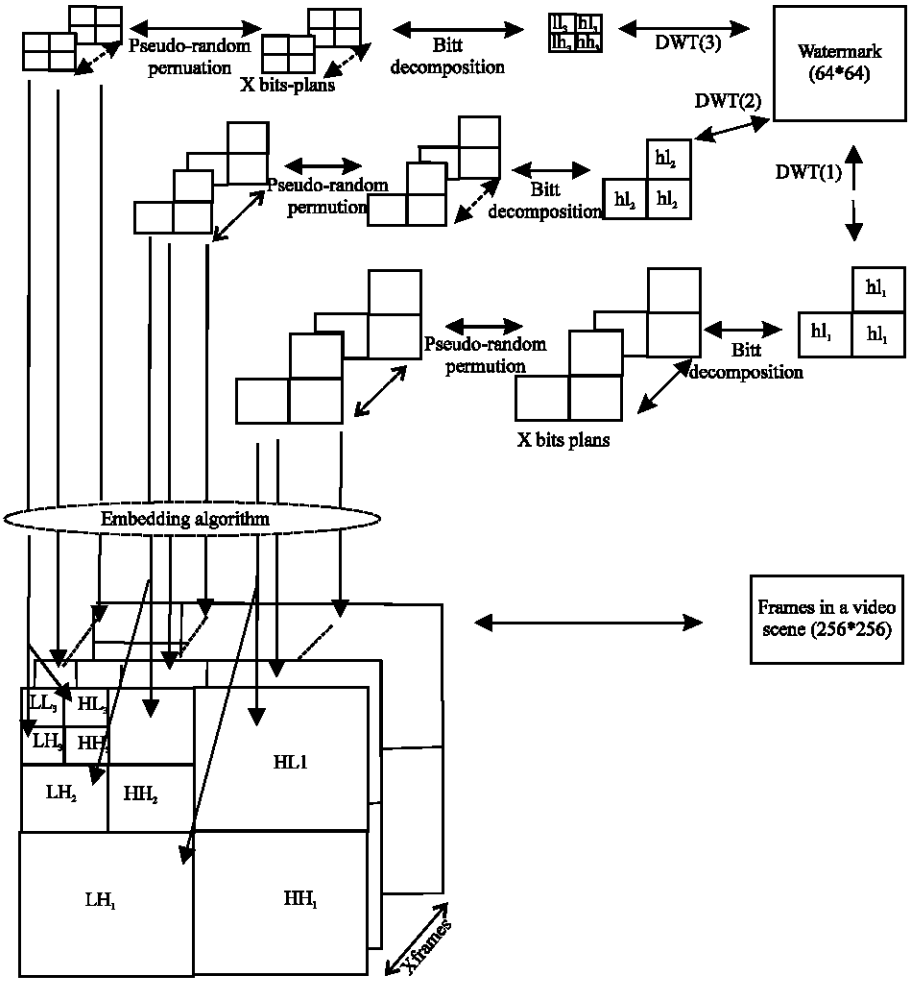

Fig. 1: Embedding watermarking process

by Kunder is similar but selects three coefficients from the details sub-bands: One from the LH, HL and HH sub-bands at the same spatial location and decomposition level. Unlike the technique of spread-spectrum, this technique doesn't need the presence of the original image during the step of detection and it so called a blind technique.

These two techniques are based on the use of pseudo-random sequences while generating watermark or during the selection of the emplacement of insertion. This will allow to a better security of the watermarking system. In addition, other approaches that aim to improve security by employing a key-dependent wavelet transform domain include the work of Wang and Winner. For example, the work of Wang (Hsa and Wu, 2000) proposes randomly generated orthonormal filter banks depending on the image as a major part of the private key. Then, (Werner and Meerwald, 2003) proposes the use of parametrized wavelet filters as a method to protect wavelet-based watermarks against unauthorized detection and to increase the resilience against malicious removal attacks. Based on two or more parameters we can create a whole family of wavelet filters that we can use as key space.

In this respect, we propose an optimized approach to blind watermark of images and video based on the 2D-DWT followed by a specific insertion strategy. Unlike most watermarking scheme, our proposed method is optimized to achieve the best tradeoff between computing complexity and hardware constraints.

Our proposed watermarking algorithm

General watermarking scheme: The general model of insertion of the watermark could be decomposed on the following steps:

- Transform the original image in the selected insertion domain.
- Select the components of the original image which have to be computed when component designate all pixels of the image or the result of the frequential (TCD, TFD) or the multi-resolution transformation.

- Fix the watermark which can be predefined or generated with the help of a random generator. This watermark can lead to many treatments such as: spread-spectrum, transformation in the image domain, cutting and decomposition into bit-plan.
- Add the watermark on the selected elements of the image with the appropriate method of insertion (addition, substitution, replacement).

Proposed algorithm: The proposed watermark embedding process as depicted in the Fig. 1 satisfy to the following choices:

- Perform a two-dimensional wavelet transform for both video stream and watermark.
- Use a predefined watermark.
- Use the quantization strategy during the embedding of the watermark information.

The watermarking process consists on performing a 2D-DWT on both images and watermark before applying the insertion process. In fact, the video sequence (composed with images at the 256×256 gris level for our application), is transformed with a 2D-DWT to obtain a multi-resolution decomposition at the level Y (HH_y, HL_y, LH_y, LL_y, HH_{y-1}, HL_{y-1}, LH_{y-1}, ..., HH_1, HL_1, LH_1). The watermark to be embedded in the video consists of an image 64×64 (gris level), and will follow the following transformations:

- Transform the watermark with a DWT to obtain a multi-resolution decomposition at the level Y (hhy, hl_y, lh_y, ll_y, hh_{y-1}, hl_{y-1}, lh_{y-1}, ..., hh_1, hl_1, lh_1).
- Perform a decomposition on X bits-plans (X will be determined by the length of data) of the sub-bands for every level.
- Execute a pseudo-random permutation of X bit-plans in order to spread the watermark.

After all these transformations are followed by embedding every bit-plan of the watermark into the corresponding level of decomposition of the frame (every X frames). This insertion will be performed in the medium frequencies HL_i and LH_i (i: 1 à Y). We will apply a relation of order between the frames' coefficients by using appropriate algorithm. Finally, an inverse DWT will be achieved on the frames after applying the insertion to obtain a watermarked video.

Watermarking process features: Our proposed watermarking process has the following features:

Blind watermarking algorithm: The detection of the watermark does not necessitate the presence of the original video sequence.

Invisibility: Thanks to the fact that almost the energy of the whole natural images are concentrated under low sub-bands, low frequency wavelet coefficients (LL) of sub-bands of the frame are not watermarked. As a result the watermarking obtained is invisible.

Resistance to the loss by compression: To remedy the loss of compression induced because of the elimination of details (components of high frequency: HH_i), the wavelet coefficients of the watermark are embedded in the medium frequency of the video.

Resistance to the loss by rejecting frames: The watermark must be present every where in the video with a periodic way (every X frames). This makes the attacks by the rejection of frames inefficient.

Security: By employing pseudo-random permutation between bits-plans of the watermark, we increase the security of the watermarking system.

Architectural mapping: The proposed watermarking technique presented before as hardware architecture needs a judicious cutting out which respect to temporal constraints of the video flow. So, we keep a hybrid architecture model in which the optimum touches two parts: the supervising part and the treatment part. Our architecture uses the following blocs:

- The multi-resolution decomposition (2D-DWT).
- The bit-plans images and video decomposition.
- The pseudo random permutation between bit-plans.
- The bit-plans embedding in the sub-bands of frame.

The multi-resolutiondecomposition step

Background: Many architectures have been developed to implement a 2-D wavelet transform. Three basic architectures, surveyed in (Zeruas, 2001).

A typical level-by-level architecture: Uses a single processing module that first processes the rows, and then the columns. Intermediate values between row and column processing are stored in a memory; since this memory must be large enough for the entire image, external memory are usually used. Access to the memory is sometimes done in row-wise order, and sometimes in column-wise order, so high-bandwidth access modes can be used; as a result, external memory access can become a performance bottleneck.

The block-based architecture: Is like the level-by-level except that it breaks the image into blocks small enough to fit in an embedded memory that are processed separately. The processed image can suffer from visual artifacts unless extra computations are done at the boundaries between blocks.

A typical line-based architecture (Crysafis and Orlega, 2000): includes separate processing modules from each level of transformation to be done, and initiates column processing as soon as a sufficient number of rows has been processed.

Another architecture proposed by Barua *et al.* (2005), looks as a *hybrid* of level-by-level and line-based architectures. This solution is proposed to achieve a small yet high-performance system. So, it keeps the hardware size small by using a single computing module iteratively for the higher level computations, and going back to external memory between levels. Howaver, unlike a typical level-by-level architecture, they do not use the external memory between row and column processing of a single level. They use line-based processing within a level, with embedded memory for the buffer space between row and column processing.

Our architecture was a mixture between the Typical line-based and Baruamodels. Our choice takes the nature of the whole application into account. In fact, the wavelet transform step was followed by another process which imposes many constraints that have an influence on the choice of the architecture. Our model will be clearly developed in the next sections.

The 2D-DWT architecture organisation: We apply a wavelet transform which decomposes the image in a whole of sub-bands with different resolutions (LL, LH, HL et HH). This decomposition technique serves to obtain under images by a recurrent application. First, on the source image and then on the approximation obtained with every level of image. The decomposation consists on filter banks pass-low, pass-high applied successively according to the rows and columns of the frame that will be transformed. The proposed architecture for the implementation of the 2D-DWT uses.

- Two spread modules of treatment: the TL module for rows treatment and the TC module for columns treatment.
- An internal memory: used to store intermediate wavelet coefficients writing by TL and reading by TC. It also store sub-bands generated by TC, which will be used by the insertion process (HL_i, LH_i) and by the next level of the transform (Ll_i) show in Fig. 2.

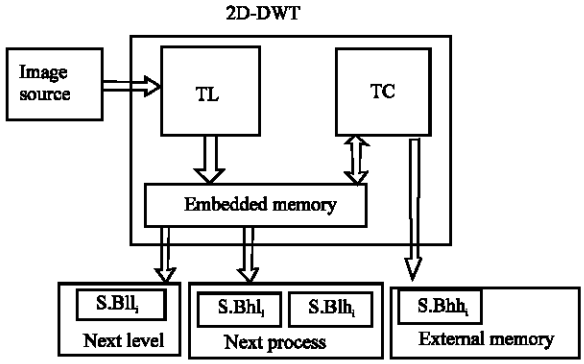

Fig. 2: Block diagrams of the 2D-DWT

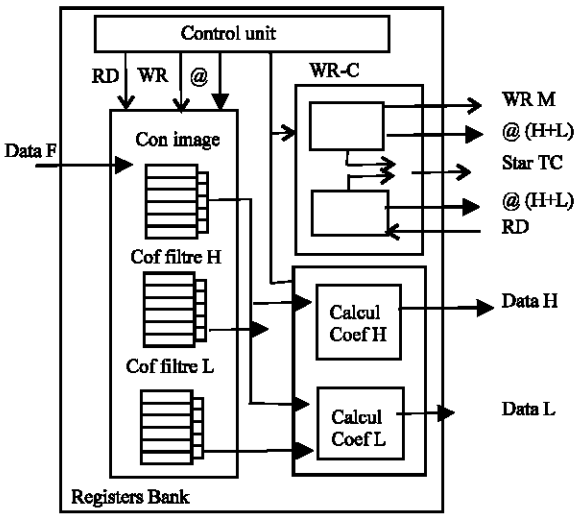

Fig. 3: Internal architecture of the TL block

This architecture operates on a source image and generates four sub-bands hh, hl, lh and ll. The TL bloc starts computing when there is the N first value of the first row on the file source (N is the length of used filter) and gives the wavelet approximation coefficients (l) and details (h). These coefficients are buffered on internal memory so that we can obtain a succession of columns which will be exploited directly by the TC bloc, the last one maakes a simultaneous reading of the coefficients h et l and then a parallel treatment needed to supply the four sub-bands.

Internal architecture of TL bloc: The Fig. 3 presents the architectural details related to the TL block which constitutes a basic DWT block. This scheme integrates the following basic units:

Register bank: We have used three register banks organisation. The two first allow the storage of filter

```
Repeat
read m(j)
read f[i: i+P]
IF m(j) = 1 then
f(i) = max [ f(i), f(i+1), f(i+2)
ELSE
f(i) = min [ f(i), f(i+), f(i+2)
END IF
write f[ ]
i = i+P
j = j+1
unitl (j>k ou i>N)
```

Fig. 4: Insertion scheme

coefficients pass-low (FL_i) and pass-high (FH_i), while the third one contains the reading values from the source image.

Calculation unit: It is duplicated to make the calculation of high and low coefficients. This calculation is realized by the use of operation multiplication with accumulation operation.

Controlunit: It initiates the activities of the three register banks and also controls the computing unit. In fact, this component generates an address from which it reads three elements from the register banks found in the same address to be used by the computing unit.

Addressing unit: It is used to carry wavelet coefficients until the TC block through the internal memory. This technique of directing is a little bit particular because it performs the transformation of coefficient orientations (row and columns) in order to prepare the future treatment. The computing unit is duplicated in order to operate in a parallel way where the addressing component and the synchronisation signals are the same for the two memories.

Internal architecture of TC bloc: The architecture of this block is composed of two TC blocks which operate simultaneously. The first block operates on the high coefficient (HH and HL) where the second one uses the low coefficient, given by the TL component, and delivers two sub-bands LH and LL.

The bit-plans decompositionstep: Unlike the majority of the previous works (Chan and Lyu; Xiamou and Zheming, 2000), the decomposition step is achieved in the transformed domain and not in the pixelique one. This choice is justified by all the potentialities given by VHDL language for this kind of computation to allow a high optimized hardware architecture which can suit with the real time contsraints.

The pseudo-random permutation step: The pseudo random permutation function consists in changing the order of X bit-plans, resulting in a simple permutation of the signs affectation order between computation unities. This solution make the implementtaion easier that the other solution which consists on making a pseudo random permutation of coefficients into every bit-plans. The second solution requires additionnal memory ressources compared to our proposed solution.

Architecture of the bit-plans insertion step: The implemented insertion technique is resumed in the following scheme: Show in Fig. 4 Where f(i) is the $i^{ème}$ wavelet coefficients of the frame and m(j) the $j^{ème}$ binary coefficient of the watermark. In this study, j increases by one, while i increases according to size rapport between bits-plans of the watermark and the frame sub-bands.

The blocs of insertion receive as an input

- One (lh_i) or two (hh_i and hl_i or lh_3 and ll_3) sub-bands containing the binary coefficients of bit-plans of the watermark.
- One sub-bands of medium frequency of frames (HL_i ou LH_i) containing wavelet coefficients.

This bloc generates one watermarked sub-band: HL^T_i or LH^T_i .

As it is shown by the Fig. 5, our architecture is constituted by a Finite State Machine (FSM) which communicates with an internal register and with inputs/outputs. The handshaking signals perform the synchronization between different tasks. This block can be configured, by a generic input, according to the insertion step number which represents the reading vector length related to HL_i or LH_i sub-bands. The transformed vector is then transferred to the output of this block. The Fig. 1 shows an example of insertion in the three decomposition levels. According to the 64×64 gry-level watermark and 256×256gry-level frame sizes we can notes the following remarks:

- In the first level insertion, hh_1 and hl_1 sub-bands are embedded successively in HL_1 with a 8 vector-length-coefficient where lh_1 was embedded in LH_1 with a 16 vector-length-coefficient.
- In the second level, hh_2 and hl_2 sub-bands are embedded successively in HL_2 with 8 vector-length-coefficients where lh_2 was embedded in LH_2 with 16 vector-length-coefficients.
- In the third level, hh_3 and hl_3 sub-bands (resp. ll_3 and lh_3) are embedded successively in HL_3 (resp. LH_3) with 8 vector-length-coefficient.

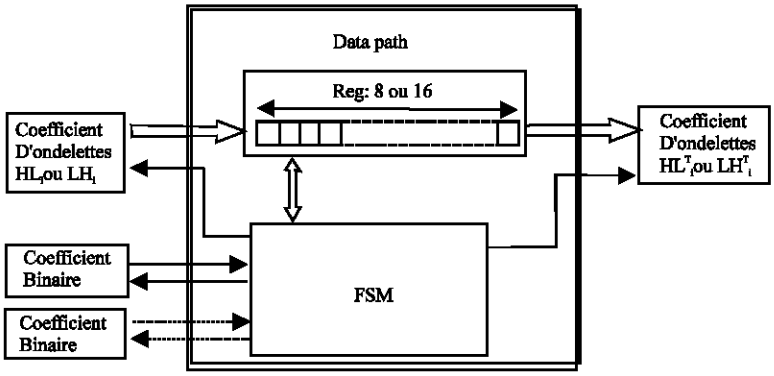

Fig. 5: Insertion block architecture

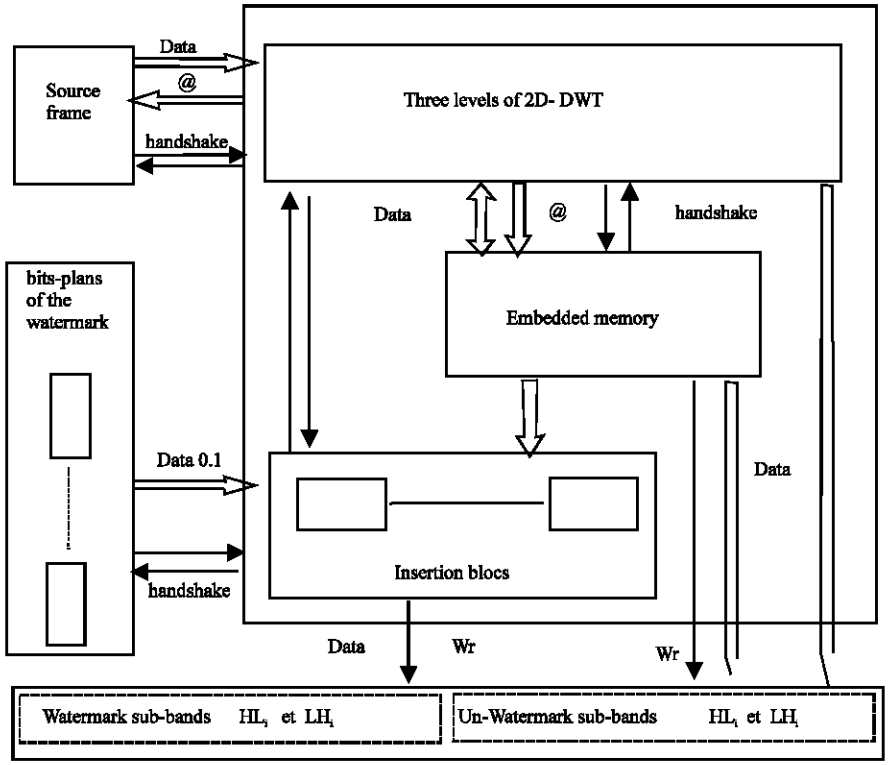

Fig. 6: General block diagram architecture

Global architecture: The general block diagrams of our proposed blind watermarking algorithm are shown in the Fig. 6. This architecture includes the following components:

- The 2D-DWT based computing component: It consists on three levels decomposition related to digital images and video. These levels exchange synchronization signals where data transfer is achieve only with the internal memories.
- The second component is constituted by internal DPRAM memories which operate in parallel way.
- The insertion component: This block inserts one bit plan of the watermark in a sub-band of medium frequency and gives a watermarked sub-bands which added to the un-watermarked sub-bands (low and high frequency) constitutes the whole watermarked frame.

This architecture is first used to perform the required watermark computing using only the two first blocs. The results are then stored into the external memory which will be used after as an input for frames computing. We remark that the waist of the image,

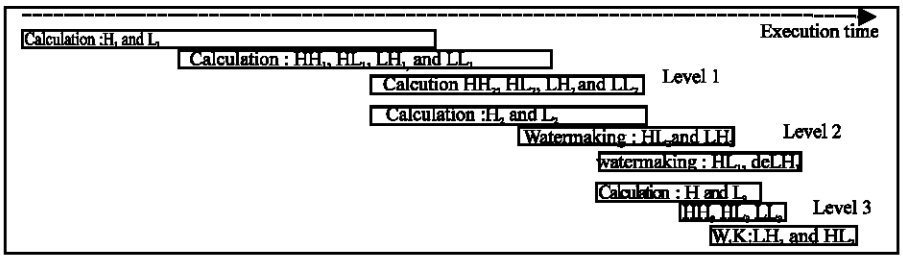

Fig. 7: Processes temporal distribution

the number of the level of decomposition and the length of filter used are generics.

Parallelism between processes: The proposed architecture was described at the RTL level in VHDL language. We have then used the Altera Modelsim to validation our architecture. According to the same level, many processes are performed in a parallel way in order to satisfy the temporal constraints, such as:

- Calculation of H_i and L_i sub-bands made by TL blocs.
- Calculation of HH_i, HL_i, LH_i and LL_i sub-bands made by TC_i blocs.
- Watermarking of HL_i and LH_i sub-bands.

For two successive decomposition levels , different processes were overlapped as shown in Fig. 7.

The parallel computing of detail and approximation coefficients H_i and L_i, as well as HH_i, HL_i, LH_i and LL_i is performed by the duplication of calculation units inside TC_i and TL_i blocs for the same level. The same approach is applied for computing the watermarking of the middle-frequency sub-bands HL_i and LH_i.

Reusing in the architecture: As shown by Fig. 7, the treatment made by TL_1(resp. TC_1) was finished before the activation of TL_3 (resp. TC_3). Consequently, we can use the same blocs $TC_{0,3}$ and $TL_{0,3}$ for the first and third level of decomposition, so we can re-use the first processing module.

The architecture of those units ($TC_{0,3}$ and $TL_{0,3}$) must allow a second reset and a re-initialisation of counters. In addition, multiplexers and demultiplexers are necessary for data and handshaking signals in order to specify the present level.

Implementation results: After the development of the generic architecture of the watermarking module, a step of evaluation and simulation is achieved. For the first evaluation, we have used fixed images. The same treatment can be applied on video sequences by taking into consideration the synchronization aspect.

The RTL VHDL description is synthesized, placed and routed on an Altera STRATIX II EP2S606C57ES FPGA using the Altera Quartus version 5.0 software toolset. The description takes sixteen bit input data. The implementation requires that the filter set coefficients first be quantized into fixed-point. Fixed-point precisions in the RTL VHDL for our architecture are chosen so as to ensure that no overflow occurs at the most significant end of signals, and no truncation or rounding occurs at the least significant end.

Device characteristic: The embedded memory used in our architecture, must follow models of used device. Stratix II devices feature the TriMatrix™ memory structure, consisting of three sizes of embedded RAM blocks. The TriMatrix™ architecture provides complex memory functions for different applications in FPGA designs. For examples:

- M512(512 bits) blocks are used for first-in-first-out (FIFO) functions and clocks domain buffering where memory bandwidth is critical.
- M4K (4 Kbits) blocks are ideal for applications requiring medium-sized memory, such as Asynchronous Transfer Mode (ATM) cell processing
- M-RAM (32Kbits) blocks are suitable for large buffering applications, such as Internet Protocol (IP) packet buffering and system cache.

The TriMatrix™ memory blocks support various memory configurations, including single-port, simple dual- port, true dual-port, shift register and ROM modes. For our application we adapt the simple dual-port mode. Table 1 shows the capacity and distribution of the TriMatrix memory blocks in Stratix II

RESULTS

Using a 256*256 input frame and a 64*64 watermark, our application requires 90% of internal memory of the EP2S606C57ES device. However, these memories are not homogeneous (2 M-RAM, 255 M4K and 329 M512) and so we can't embed all the sub-bands in the memory mapping space. To resolve this problem, we have reduced by the half the size of the frame as soon as the size of the watermark to conserve the size vector-length coefficients during the insertion process. Table 2 illustrates the synthesis results. For example, the required memory for our architecture is a bout 22% of the total memory space with the FPGA resulting of the size reduction for the frame and the watermark.

Table 3 shows the distribution of different sub-bands on the memory models of the device. We don't use M-RAM because its size exceeds our needs.

Architecture optimisations: Different architectures are developed and optimised until we lead to our adopted architecture.

- First, we have used separate processing modules for each level of transform and then different embedded memory.
- Next, we have re-used the embedded memory for the first level to store coefficients coming from third levels. Then we have used multiplexer and demultiplexer in input and output of the memory.
- We have used one processing module and one embedded memory for the first and the third level. We have also brought some modifications in the module (re-initialisation, new reset) and in the rest of the architecture (multiplexers and demultiplexers).

Table 1: TriMatrix memory capacity and distribution in Stratix II device

Device	M512 columns/blocks	M4K columns/blocks	M-RAM Blocks	Total RAM bits
Ep2s60	7/329	5/255	2	2.544, 192

Table 2: Flow summary

Total Aluts	8.122/48.352(16%)
Total register	1424
Total pins	291/493(56%)
Total virtual pins	0
Total memory bits	573.440(22%)
DSP block 9-bit element	20/288(6%)
Total dlls	0/6(0%)

Table 3: Memory distribution

Niveau	H	L	LH	HL	LL
1 et 3	32 x M4K	32 x M4K	16 x M4K	16 x M4K	16 x M4K
2	8 x	8 x	4 x M4K or 32 x M512 4 x	4 x M4K or 32 x M512 4 x	4 x M4K or 32 x M512 4 x

Table 4: Characteristics of different architectures

Type	Memory bits	ALUT	Register	Pin	DSP block
1	688.128 (26%)	10.410 (21%)	1580	258 (51%)	20 (6%)
2	655.360 (25%)	9.910 (20%)	1576	255 (51%)	20 (6%)
3	655.360 (25%)	8.130 (16%)	1423	249 (50%)	20 (6%)
4	573.440 (22%)	8.122 (16%)	1424	281 (56%)	20 (6%)

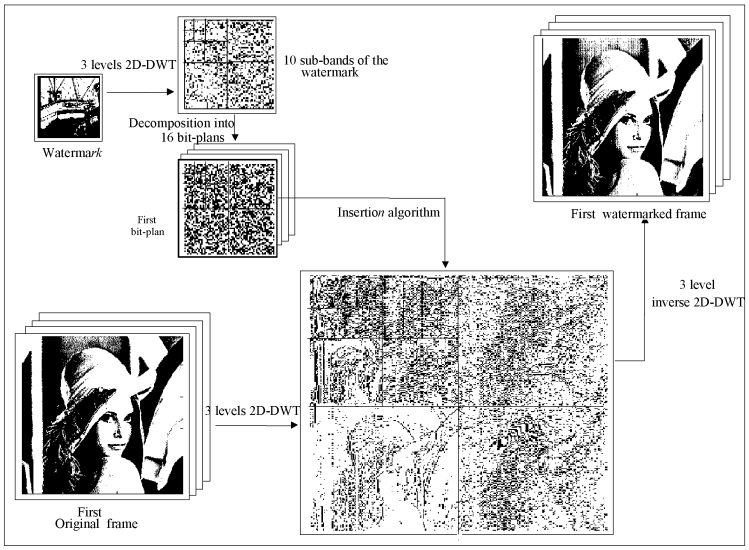

Fig. 7: Simulation results for frame and watermarks

- At the end, we have reduced the internal memory by store wavelet coefficients that are not used by another process in an external memory.

Table 4 summarizes the characteristic variation of the architecture during these optimisations of point of view: size of the embedded memory and number of ALUTs, registers, pins and DSP blocs.

Simulation results: The watermarked image was built from sub-band images from a simulation of architecture which is described into VHDL at RTL level (Register Transfer Level). These sub-bands (frame or watermark) obtained from a 2D-DWT are constituted of binary vectors (sign+ciel and floor parts) presented wavelet coefficients. Many C programs are developed to transform these vectors into useful floats by MATLAB in order to be illustrated. The Fig. 8 illustrates the result of computing performed on an image«boat» used as a watermark. A multi-resolution decomposition with three levels is made following a step of bit-plan decomposition to watermark the first frame.

The Lena image is detained as a test frame in which we have applied a multi-resolution decomposition with three levels followed by an insertion in medium frequencies sub-bands. After the VHDL-MATLAB adaptation, an inverse 2D-DWT is achieved on the watermarked sub-bands. Figure 8 shows the frame before and after watermarking operation.

To quantize the visual quality, we have used the metric of calculation of PSNR used on the original image and on the watermarked one. The result of evaluation shows that the watermarking application has not affected the visual quality of the image because we have obtained a PSNR of 30,34 db.

CONCLUSION

In this study we have proposed an improved hardware watermarking scheme for images and video frames. This scheme consists on a blind technique based on a 2D-DWT multi-resolution analysis. The proposed architecture is generic in term of the image size and the filters length used for the 2D-DWT transform implementation. Experimental results show that this technique gives an interesting properties of security and robustness. Our RTL architecture exploits parallellism and pipeline techniques to sweet well with the real time caracter of the video frames. In addition after the optimisation of the last version, only about 25% of the memory and FPGA ressources realted to the STRATIX II proptotyping board are used. The proposed architecture can be consisdered as a generic architecture to validate other watermarking technique like the lifting scheme based one.

REFERENCES

Barua, S., J.E. Carletta, K.A. Kotteri, A.E .Bell, 2005. An efficient architecture for liftimg-based two-dimensional discrete wavelet transforms, INTEGRATION, the VLSI J., pp: 341-352.

Bender, W., D. Gruhl, N. Morimoto and A. Lu, 1996. techniques for data hiding, IBM Sys. J., 35: 313-336.

Chan, P.W., M.R. Lyu, A new scheme for robust blind digital vidéo watermarking: Approch, Evaluation an Experimentation.

Chan, R.H., F.R. Lin and K.M. Yeung, 2001. A frequency domain based watermarking scheme with spatial repetition coding. Proceedings of the 5th World Multi-Conference on Syctemics, Cybernetics and Informatics, (Eds.): N. Callaos, D. Rosario, B. Sanchez, pp: 35-40.

Chrysafis, C. and A. Ortega, 2000. Line-based, reduced memory, wavelet image compression, IEEE Trans. Circuits Sys. Video Tech., pp: 378-389.

Cox, I.J., J. Kilian, T. Leighton and T.G. Shamoon, 1997. Secure spread spectrum watermarking for multimidia, in: Proceeding of the IEEE International Conference on Image Processing, ICIP, Santa Barbara, CA, USA, 6: 1673-1687.

Dielt, W., P. Meerwald and A. Uhl, 2003. Protection of wavelet-based watermarking systems using filter parametrization, Signal Processing, pp: 2095-2116.

Hsu, C.T. and J.L. Wu, 2000. Image Watermarking by Wavelet decomposition, Acad. Inform. Manag. Sci. J., pp: 70-86.

Kim, J.R. and Y.S. Moon, 1999. A robust wavelet-based digital watermark using level-adaptatif thresholding, in: Proceeding of the 6th IEEE International Conference on Image Processing, ICIP, Kobe, Japon, pp: 202.

Kunder, D., 1999. Improved digital watermarking through diversity and attack characterization, in Proceeding of the ACM Workshop on Multimidia Security, Orlando, FL, USA, pp: 53-58.

Mohanty, S.P., A.l., 2000. A DCT Domain Visible Watermarking Technique for Images, In: Proceeding of the IEEE International Conference on Multimidia and Expro, Hilton New York and Towers, New York City, NY, USA.

Osborn, C.F., A.Z. Tirkel, G.A. Rankin, R. Van Schyndel and W.J. Ho, 1993. Electronic watermark. In Digital Image Computing, Technologie and Application, Sydney Australia, pp: 666-672.

Van R., Schyndel, A.Z. Tirkel and C.F. Osborn, 1994. A Digital Watermark, In 1st IEEE International Conference on Image Precessing, Austin Texas USA, 2: 86-90.

Wang, H.J. and C.C. Jay Kuo, 1998. Watermark design for embedded wavelet image codec. In: Proceeding of the SPIE's 43rd Annual Meeting, Applications of Digital Image Processing, San Diago, CA, USA, 3460: 338-398.

Xiamou Niu, Zheming Lu and Shenghe Sun, 2000. Digital image watermarking based on multi-resolution decomposition, IEEE. Elec. Lett., pp: 1108-1110.

Xie, L., G.R. Arce, 1998. Joint wavelet compression and authentification watermarking, In: Proceeding of the IEEE international conference on image proceeding, ICIP, Chigago, IL,USA

Zervas, N.D., G.P. Anagnostopoulos, V. Spiliotopoulos, Y. Andreopoulos and C.E. Goutis, 2001. Evaluation of design alternatives for the 2-D discrete wavelet transform, IEEE Trans. Circuits Sys. Video Tech., pp: 1246-1262.