OPTIMAL IMPLEMENTATION ON FPGAs BASED ON WAVELET TRANSFORM AND SELF ORGANIZING MAP FOR MEDICAL IMAGES COMPRESSION

F.Alim-Ferhat, H.Bessalah, Seddiki, O.Kerdjidj
a Advanced Technologies Development Center
Cité 20 Aout, BabaHacene, Alger, Algerie

E-mail : alim_ferhat@yahoo.fr

ABSTRACT

Medical images present specific characteristics which require to be exploited by an explicit and efficient compression algorithm. The Vector Quantization (VQ) constitutes a crucial stage in lossless Digital images compression where it allows to create a dictionary on the level "block" by a neuronal approach that of Kohonen (Self Organizing Map : SOM) which is widely used in implementation of FPGA circuits for medical image compression applications. This paper is devoted to the implementation of a new combined method based on wavelet transform and neurons network (WT-SOM) and designed for the implementation on FPGA for medical images compression applications. Simulation results show the effectiveness of our proposed method.

Keywords

wavelet transform, on line arithmetic, medical images, compression, vector quantization, Kohonen map, FPGA.

1. INTRODUCTION

For many years now, the compression of still images has been a very active area of research. The work carried out has led to the establishment of the JPEG and JPEG2000 standards. These two standards are respectively based on DCT and wavelet transform.
Many research works are focused on this area and improve their approach work a new functionality and a great adaptability for the relevant domains such as medical and satellite imaging.
Three wavelet based compression methods have served, and continue to serve, as references;[1][2][3].
A combination of the wavelet transform and many other algorithms such as neural networks have proved to be suitable for loosely medical image compression [4][5].
Lossy compression techniques do not leave an original medical image unimpaired. For natural images, coding techniques must keep to one single criterion relating to visual quality of the reconstructed image. For medical images, it is essential that the compression avoids any distortions that could modify the diagnostic interpretation of the image and the value of anatomic.
Defining the amount of distortion accepted that could preserve the reliability of the diagnosis of the reconstructed image is a complex problem and an open debate in the medical imaging field. In fact, the eligible compression rate does not only change according to the compression method applied but also largely depends on the characteristics of the image being studied; characteristics that are linked to the gathering techniques as well as to the nature of the organ being explored and to the pathology itself [6].
The main drawback of the inclusion of an artificial networks algorithm is that it is a very computationally intensive task if software implementation is performed alone; the structure is fairly easy to convert into parallel processing unit. However, it consumes much of a internal resources and results in utilizing more than a single microchip to realize the structure in pure hardware [7],[8]. Different alternatives are used to implement these kinds of algorithms in FPGAs circuits. Among them, Kohonen Self organizing map (K-SOM) [9],[10],[11].
In this paper, we propose a combined technique using the wavelet transform and the K-SOM algorithm for medical image compression with an implementation on FPGAs circuits. We propose a modilization of the S. Mallat algorithm with an on line arithmetic.
The sequential nature of S. Mallat algorithm [12] constitutes a constraint which reduces sensitively the application field of the wavelets particularly in the fields requiring a real time transmission.
To remedy to this constraint we recommend the use of the online arithmetic and pipe line architecture with a level bit in the execution of the S. Mallat's algorithm [13][14].
The wavelet coefficients in low frequency sub image are compressed and scalar quantization (SQ) using Kohonen neural network in order to preserve the maximum information of the medical image while the wavelet coefficients in high frequency sub-images are compressed and vector quantized (VQ) by using the same algorithm, in order to compress these parts with high compression rate that explore a minimum FPGA's resources [15].
Using our compressing technique, we can obtain rather satisfactory reconstructed images with large compress ratio.
This work is divided into two principal parts. In the first one, we present the chosen technique, some brief information about the wavelet theory, using S-Mallat algorithm, and a description of the on-line arithmetic and wavelet transform. We give also the choice of the most optimal parameters for medical images compression leading to an optimal choice of neurons and iterations numbers. The second part is reserved to the results and discussions. Finally, we give conclusion to our work.

2. PROPOSED COMPRESSION ALGORITHM

The compression aims to reduce the number of bits encoding an image. This is possible thanks to a redundancy of information that is generally present in an image.
This redundancy can be of statistical, spatial or spectral origin. The often used general scheme to describe the compression algorithms is shown in figure 1.

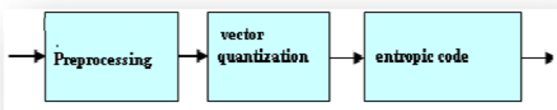

Figure 1. General scheme of compression.

The first stage is a transformation used to change the representation space of the spatial field. The obtained representation space is often called the spectral field. The quantization step achieves a reduction of represented values number. Shannon proved that it's always possible to improve the data compression by coding vectors rather than the scalar. This approach is known as vector quantization (VQ) and it deals only with irreversible algorithms. Entropic encoding is carried out without loss on the quantified values.

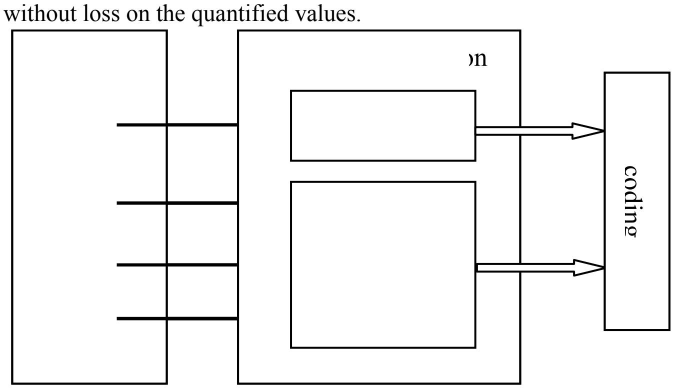

Figure 2. Block diagram of our method (WTSOM).

In our work we have developed a new micro architecture using the online arithmetic for S.Mallat algorithm implementation based on LEGALL 5/3 filter (Bloc 1of Figure 2).
In the stage of quantization (Bloc 2 of Figure 2) we seek to establish a compromise between the numbers of neurons and iterations by keeping a good quality of the original image in order to prevent medical errors caused by poor reconstruction of the image.

2.1, Bloc 1: wavelet transform

2.1.1, Review of S. Mallat Algorithm

The time frequency decomposition of a signal is simply a filtering of its low and high pass components, called L and H decompositions. When working with images, a second step of splitting is necessary to obtain the LL, LH, HH and HL decompositions [16][17].

2.1.2, Decomposition / Reconstruction

The figure 3 shows the process of decomposition in the approximation called L and details called H. The L branch is the chosen architecture to be implemented at first; this consideration is taken, due to the huge amount of information which is in the L bands
The image is decomposed into four sub- bands
(LL, LH, HL, HH) that correspond respectively to the approximation coefficients (low frequencies coefficients) and the horizontal, vertical and diagonal details coefficients (high frequencies coefficients). The size of each one represents 1/4 of the initial image size.

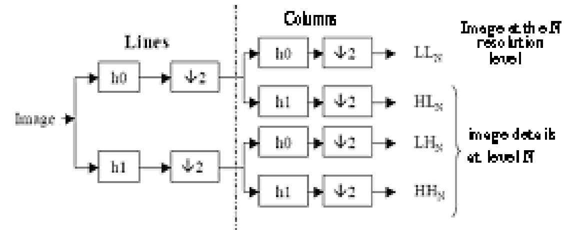

Figure 3. Principal of image decomposition by a 2D wavelet transform

2.1.3, Evaluation of the S. Mallat Algorithm by the on line Arithmetic

The sequential nature of S.Mallat algorithm constitutes a constraint which reduces sensitively the application field of the wavelets particularly in the field requiring a real time transmission.
Different applications require data to be available at high speed, like the bit transmission or bit processing when connected to different stages of a pipeline, in order to be efficient at the bit level, the implementation must have optimized data paths.
To remedy to this constraint we recommend the use of online arithmetic and pipe line architecture with a level bit in the execution of the S.Mallat's algorithm.
From this point of view, the on line calculation mode is very efficient, because of the circulation of the operands in a serial mode, most significant bit (MSB) first. The possibility of having a pipeline at the bit level makes it possible to have a massive calculation with a tolerable accuracy.

2.1.4, Review of the On Line Mode

In 1961, Avizienis [18] introduced the writing of the numbers in a redundant system of representation. Later, algorithms of calculation of elementary operations and more complex functions were elaborated. These algorithms are based on the circulation of operands in a bit by bit manner, most significant bit (MSB) first [19].

2.1.5, Mathematical Modelling of S.Mallet algorithm

The wavelet filter model is considered to be a couple of finite impulse filters FIR of order N, such that:

$$Y_n = \sum_{i=0}^{N} h_i X_{n-i} \qquad (1)$$

Y_n : Output signal ; X_n : Input signal
h_i: Coefficients of the filter
N: Is the order of the filter, it may be different for each of the used filters.

The mathematical development is similar for both high and low pass filter that is why we consider in our on-line equations a FIR filter, with no bandwidth limitation.
The on-line input output equations given at each step j are defined by:

$$X_n[j] = X_n[j-1] + x_n[j]b^{-j} \quad (2)$$

$$Y_n[j] = Y_n[j-1] + y_n[j]b^{-j} \quad (3)$$

Where b is the basis set to 2, for convenience of implementation.
The new filter expression is given by

$$Y_n[j] = \sum_{l=0}^{N} h_i X_{n-i}[j] \quad (4)$$

at the j^{th} step, the result bit $Y_n^*[j]$ is generated with a bounded error ε_j, which is defined as follow:

$$Y_n^*[j] - Y_n[j]| \le \varepsilon_j \quad (5)$$

where

$Y_n[j]$ is the correct value and $\varepsilon_j = \frac{2^{-j}}{2}$. (5.a)

The algorithm does not compute directly the real value of $Y_n[j]$, but a shifted value $Y_n^*[j]$ with p bits shift, shown as below:
The value of p is the delay and will be computed in the next section.

$$Y_n^*[j] = 2^{-p} Y_n[j] \quad (6)$$

From the equations (4), (5) and (6) we obtain,

$| 2^{-p}Y_n[j] - Y_n[j]| \le \frac{2^{-j}}{2}$, which is equivalent to :

$$|2^{-p} \left(\sum_{l=0}^{N} h_l X_{n-l}[j]\right) - Y_n[j]| \le \frac{2^{-j}}{2} \quad (7)$$

2.1.6, The partial residue [10]

We define the partial residue R[j] as:

$$R[j] = 2^j\left[2^{-p}\left(\sum_{l=0}^{N} h_l X_{n-l}[j]\right) - Y_n[j]\right] \quad (8)$$

The algorithm converges if and only if:

$$2^{-j}|R[j]| \le \frac{2^{-j}}{2}$$

$$\text{Thus : } |R[j]| \le \frac{1}{2} \quad (9)$$

This must be maintained, at a bounded value to ensure convergence at each iteration. Replacing equations, (2) and a delayed version of (3), we obtain the following result:

$$R[j] = 2^j\left[2^{-p}\left(\sum_{i=0}^{N} h_i X_{n-i}[j-1]\right) - Y_n[j-1]\right] + 2^{-p}\left(\sum_{i=0}^{n} h_i x_{n-i}[j]\right) - y_n[j]$$

The recursive expression of R[j] is simplified to:

$$R[j] = 2R[j-1] + L[j] - y_n[j] \quad (10)$$

$$\text{and : } L[j] = 2^{-p}\left(\sum_{l=0}^{n} h_i x_{n-i}[j]\right) \quad (10.a)$$

2.1.7, The complete residue

The complete residue is described by:

$$H[j] = R[j] + y_n[j] \quad (11)$$

From the equations (10) and (11):

$$H[j] = 2H[j-1] + L[j] - 2y_n[j-1] \quad (12)$$

From the relation in equation (9):

$$-1/2 \le R[j-1] \le 1/2 \quad (12.a)$$

Using (12.a) and (11)

$$-1/2 + y_n[j] \le H[j-1] \le 1/2 + y_n[j] \quad (13)$$

With $y_n[j] \in \{-1,0,+1\}$: maximum redundancy is used
Finally, the complete residue defines the value of the result bit by the inequality:

$$-3/2 \le H[j-1] \le 3/2 \quad (14)$$

From the equations (11) and (12) the selection function is given by S(H[j]) with :

$$\begin{cases} H[j] = 2H[j-1] + L[j] - 2y_n[j-1] \\ y_n[j] = S(H[j]) \end{cases} \quad (15)$$

From the relation in equation (10.a) and (12)

$$L_p[j] = 2^{-p}\sum_{i=0}^{N} h_i x_{n-i}[j] - 2y_n[j-1] \quad (16)$$

2.1.8, the delay computation

As stated before, one of the bottlenecks of the on-line arithmetic is the delay p, that is defined as how many "bits" do we inject into the equations, in order to obtain a bit result.
From equations (15), and (10.a)

$$H[j] = 2H[j-1] + L[j] - 2y_n[j-1]$$

$$H[j] = 2H[j-1] + L[j] - 2y_n[j-1] <= 3/2$$

$$\Leftrightarrow p \ge 1 + \log_2\left(|\max\left(\sum_{l=0}^{n} h_i x_{n-i}[j]\right)|\right)$$

In our study, we have used the LeGall filter (table 1); this one is used for lossless compression and reconstruction in the JPEG2000 norm.

Table 1. LeGall filter coefficients

	Decomposition	Reconstruction
h0	-1/8=-2^(-3)	-1/2=2^(-1)
h1	2/8=2^(-2)	1
h2	6/8=2^(-3)+2^(-2)	-1/2=2^(-1)
h3	2/8=2^(-2)	
h4	-1/8=2^(-3)	

2.2, Bloc 2: The quantization

Vector quantization is based on the creation of a codebook, constituted by M elements (codewords). Each codeword is supported by a vector of the same size of the input vectors size. Each input vector will be approximated by a codeword in such a way that the global distortion will be minimized.
The LL band (where the major information of the original image is preserved) will undergo a scalar quantization (SQ) in order to preserve the maximum information of the image. The three details bands LH, HL and HH, will undergo a vectorial quantization (VQ) (figure 3) in order to compress these parts with high compression rate, using kohonen network.

2.2.1, Kohonen network

The Kohonen maps or (auto organized map for self organizing SOM) also called topological maps. They allow to represent a high dimensional space, by the use of non linear projection, in a reduced dimension space [20][21]. A Kohonen network is composed of:
-An input layer: for each input vector is assigned a neuron indicating the class center.

-An output layer (or competition layer).
Indeed, the kohonen network will adapt itself to the input excitations automatically in order to activate only one neuron at the output. It doesn't perform a supervised training [16][17]. The training is carried out as follows:
1-compute the Euclidean distance between Xi and Wi,j (t):

$$Y_{i,j} = \left(\sum_{i=1}^{h} \left(X_i - W_{i,j} \right)^2 \right) \qquad (17)$$

2-Find the winner neuron

$$Y_{\min} = \min\left(y_{i,j} \right) \qquad (18)$$

3-Update the Wi weights of the network.

$$W_{i,j}(t+1) = W_{i,j}(t) + \beta_j(t)\left(X_i - W_{i,j}(t) \right) \qquad (19)$$

$\forall$ i $\in V_{ind}(t)$
X_{ii} **:** Input Vector of the training base
W_{ij}: Synoptic weight of the neuron.
t **:** Number of executed iterations (time)
β_j (t): A parameter that controls the amplitude change (training coefficient)
V_{ind} (t): A neighbouring region defined around a winner neuronal.

3. RESULTS AND DISCUSSIONS

3.1 Complexity of computation of S Mallat algorithm

In order to compare, the number of computations done by a wavelet transform, we have took an image of size 256, and LeGall5/3 filter; this could be extended to a general form.

Table 2. Order of computations

	Usual method	The proposed approach
Convolutions Number	(256x256) x 5 x 2+ (128x256) x 4 =786432	2 x (128 x 128) x 5+128 x 128 x 4 =229376
Multiplication	3932160	1146880
Addition	3145728	917504

In addition, the adjust architecture allowed the elimination of the temporary intermediate memories during the convolutions operations.

3.2, Implementation of the quantization under Matlab

In order to achieve the physical implementation on FPGA, we have carried out several tests under Matlab. The objective is to determine the most optimal parameters, such as neurons and iterations numbers.
Several types of neurological images [22],[23],[24], have been used in the learning process. These images are of BMP format, size 256*256 pixels. One of these neurological images is shown in figure 4.

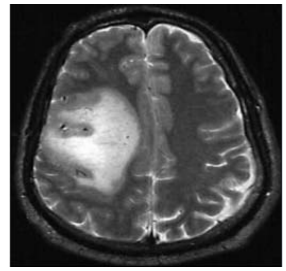

Figure 4. Example of an image used for training.

Conformably with the suggested model, these images undergo a transformation by wavelet transform dividing the input image into four blocks of 128 *128 pixels (figure 5). Blocks will be divided into two parts, and each one will attack the kohonen network. The two parts are:
- Detail part for a vector quantization.
- Approximation part for a scalar quantization

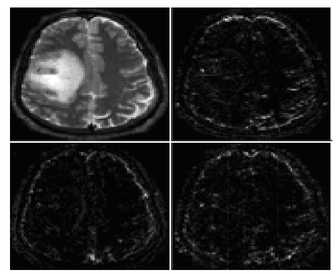

Figure 5. Block-decomposition of the image by wavelet transform .

The applied algorithm is the same for the two parts. However, it is distinguished by the nature of the input data which means:
The vector quantization is cut out in blocks of (2*2), (4*4) or (8*8).
The scalar quantization is treated pixel by pixel.
These two parts are used for the Codebook generation following rules (17), (18) and (19).
In order to determine the different parameters of Kohonen network (neurons number, iterations number and the size of input vectors for the vector quantization), we have performed several attempts on various images.
The reconstituted images quality is evaluated by the following indicators:
PSNR (Peak Signal to Noise Ratio), MQE(Middle quadratic error), the compression rate (CR) and the number of Bits per Pixel (NBP) which are defined as follows:

$$\text{MQE} = \frac{1}{T} \sum_{i=1}^{T} (X_i - x_i)^2 \qquad (20)$$

With T: size of the image
x_i : i^{th} pixel of the original image
X_i : i^{ith} pixel of the reconstituted image

$$\text{PSNR} = 10 * \log_{10} \left(\frac{255^2}{MQE} \right) \text{ (in dB)} \qquad (21)$$

$$\text{CR} = 100 * (1 - (\text{NBP}/8) \qquad (22)$$

$$\text{NBP} = \frac{\log_2 (\text{number of neurons in the map})}{\text{size of the input vector}} \qquad (23)$$

Tables 3 and 4 resume the tests results performed on the reconstructed images decomposed on 2×2, 4×4 and 8×8 blocks.

3.2.1, Test 1 for 16 neurons and 50 iterations

Figure 6 shows the reconstructed images for each decomposed block with 16 neurons and 50 iterations.

Table 3. Test for 16 neurons and 50 iterations

block	PSNR	MQE	CR	NPB
2*2	30.9863	51.8138	63.7199	1
4*4	31.0119	51.5103	69.1619	0.25
8*8	30.9999	51.6524	70.5224	0.0625

Bloc 2*2 Bloc 4*4 Bloc8*8

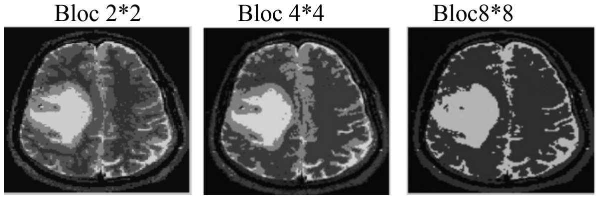

Figure 6. Reconstructed images for 16 neurons and 50 Iterations.

3.2.2, Test 2 for 25 neurons and 50 iterations

Figure 7 shows the reconstructed images for each decomposed block with 25 neurons and 50 iterations

Table 4. Test for 25 neurones and 50 iterations

block	PSNR	MQE	CR	NPB
2*2	31.0249	51.33564	68.75	1
4*4	31.0462	51.1048	73.4375	0.25
8*8	31.0513	51.0415	74.6094	0.0625

Bloc 2*2 Bloc 4*4 Bloc8*8

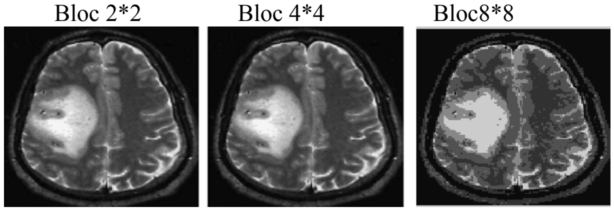

Figure 7. Reconstructed images for 25 neurons and 50 iterations.

Table 5. The different PSNR's and CR's

	Image1	Image2	Image3	Image4	CR(%)
PSNR	31.0105	30.5448	30.4287	30.2137	73,3375
	31.0130	30.5541	30.4476	30.2226	71,2997
	31.047	30.5653	30.4443	30.2382	69,1619
	31.0438	30.5698	30.4338	30.2419	62,5213

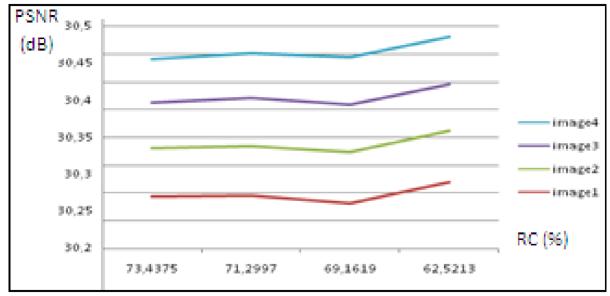

Figure 8. PSNR according to the compression rate.

In order to validate our algorithm, several tests have been performed on different images to guarantee the convergence of our system (table 3). We have traced the PSNR depending on the compression rate, we obtained the same curves. We have obtained a good PSNR which varies between 30.2 and 33.45 dB. For a perfect reconstruction this one must be incorporated between 30 and 40db [23].

3.3, Implementation of kohonen algorithm on FPGA for the VQ

The execution of Kohonen algorithm is very slow under Matlab. This is primarily due to the sequential calculation of equations (17), (18) and (19).

The performances of this algorithm can be improved by using a parallel architecture implemented on a VirtexII FPGA circuit.

This work is constituted of three principal steps:

- Compute the Euclidean distance between Xi and Wi,j (t)
- Find a winner neuron by the calculation of the minimum value of all neurons.
- Update the Wi weights of the network which depends on the training coefficient $\beta_j(t)$.

These two modules allow to determine the new synoptic weights $W_{i,j}(i+1)$,

Figure 9 shows the suggested architecture.

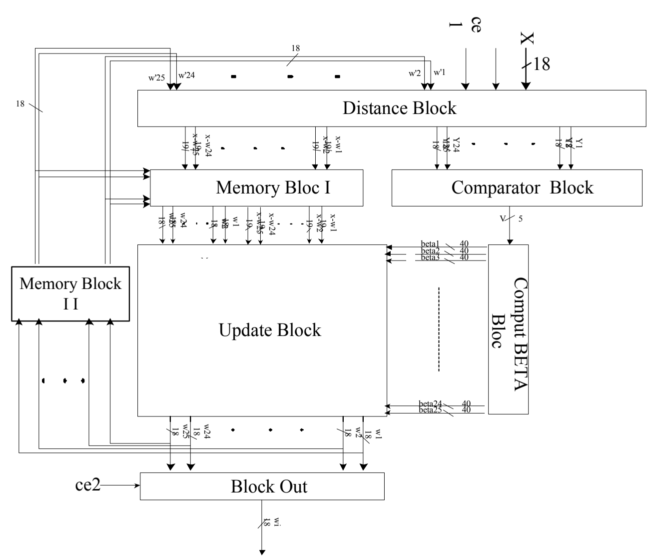

Figure 9. Global architecture of vector quantization.

3.3.1, Module 1: Euclidean distance computation

This module, illustrated by figure 10, allows us to compute the Euclidean distance.

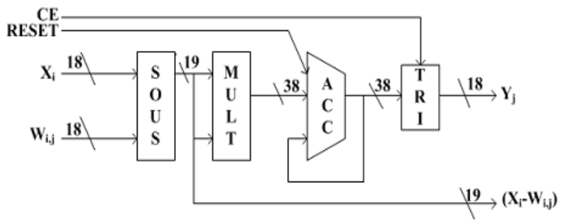

Figure 10. Module for the Euclidean distance computation

3.3.2, Module 2: Winner neuron computation

The module which allows us to compute equation (18) for the winner neuron is illustrated by figure 11.

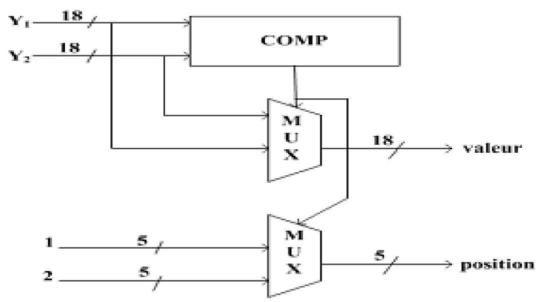

Figure 11. Module of Y_{min} .

3.3.3, Module 3: β_j (t) computation

This module allows us to compute the various β coefficient values necessary for the update. It has an input for the index (ind) which comes from the previous stage and generates 25 values of beta (βj) at the output. It is composed of 4 cells: table P and Q, β1 and Gv module.

The diagram block is illustrated by figure 12. The corresponding equations for this module are:

P = fix ((ind-1)/5) + 1 (24)
Q = ind-5* fix ((ind-1)/5) (25)
Mj = fix ((j-1)/5) + 1 (26)
Lj= j-5* fix ((j-1)/5) (27)

P and Q are values depending on the found index (ind represents the whole value and Fix is a function that rounds the elements to the nearest integers). These values are computed by Matlab and stocked in LUT tables in order to accelerate the calculation process.

Lj, Mj are constants which depend on the neuron index and the value of β . We obtain for each index a value of P, Q and 25 values of Lj, Mj.

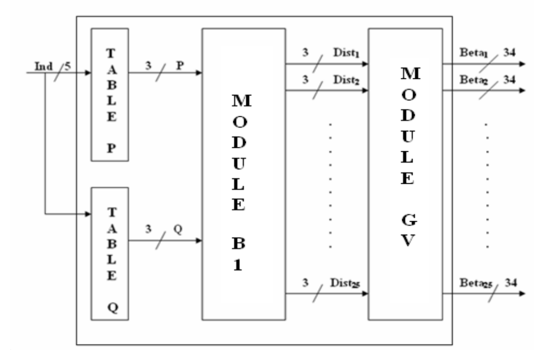

Figure 12. Block calculation of (β_j).

3.3.4, Module 4: Update

The update block allows the generation of new synoptic points $W_{(i+1),j}$ according to equation (19). It contains 75 inputs and 25 outputs:

- 25 input for the initial synoptic weights, (W_j), coded on 18 bits.
- 25 input for the (X_i-$W_{i,j}$) coming from the arithmetic unit of distance, coded on 19 bits.
- 25 input for the β's coming from the preceding block, coded on 34 bits.
- 25 output for the new synoptic weights $W_{(i+1),j}$, coded on 18 bits.

This module is constituted of 25 identical cells; the internal circuit of these cells is illustrated on figure 13.

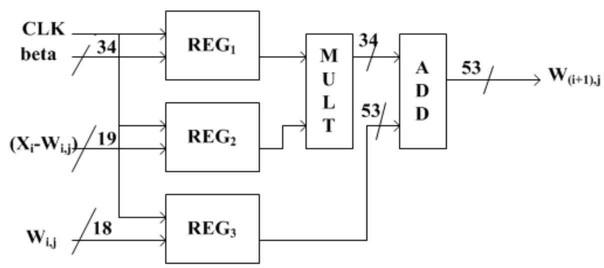

Figure 13. Update block of Wi synoptic Weights.

3.3.5, Results of FPGA implementation

In this work, the developed architecture was designed on Xilinx environment Foundation Series 7.li The various blocks of the architecture were described with the VHDL language .To guarantee correct behavior, they were simulated, synthesized and implemented, respectively, by the tools: Modelsim PE 6.0and XST.

The proposed architecture implementation is performed on the Virtex II circuit (xc2v6000-6ff1152). The performances in occupancy rate and execution deadlines are illustrated by table 6.

Table 6. Internal resources of FPGA

Number of Slices:	14806 out of 46592	31%
Number of Slice Flip Flops:	2410 out of 93184	2%
Number of 4 input LUTs:	27259 out of 93184	29%
Number of bonded IOBs:	474 out of 1108	42%
Number of BRAMs:	75 out of 168	44%
Number of GCLKs:	2 out of 16	12%

Operating Frequency	12.15 Mhz
Deadlines for an iteration	82ns
Deadlines for 50 iterations	14 s

The above coding scheme is applied to 256-256 and 8 bits /pixel original image (figure 14).

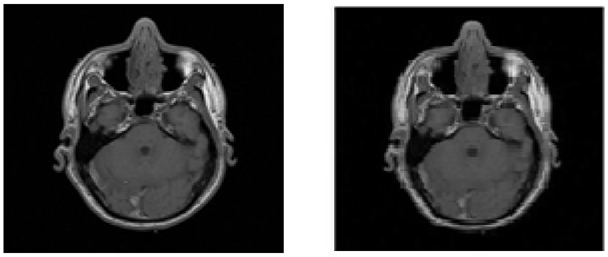

Figure 14. Original brain image. (left). Reconstructed image (right**);** PSNR=33.45 db, CR=69.16.

4.INTERFACE

This interface allow to the user to do the compression of medical images with lossy or lossless ,it give also the possibility to choose compression with the region of interest.

5. CONCLUSION

In this paper, we have proposed and implemented a compression technique with optimal parameters for medical images compression as a combined technique using neuronal network and wavelet transform.
These transforms are usually used by the scientific community for images compression in separated or combined forms.
In our work we have developed a new architecture using the online arithmetic for S.Mallat algorithm implementation based on LEGALL 5/3 filter used in the JPEG 2000 standard.
The acceleration constraint of coding related to the sequential character of S.Mallat algorithm has been surmounted due to the use of online arithmetic mode on one hand and the realization of a pipe line on the level bit on the other hand. In addition, the adjust architecture allowed the elimination of the temporary intermediate memories during the convolutions operations.
In the stage of quantization, we have seek to establish a compromise between the numbers of neurons and the quality of the original image in order to prevent the medical errors caused by a poor reconstruction of the image and it's ability to be implemented on FPGA circuit.

Acknowledgments: we would like to express my gratitude to Pr: Bouyoucef.K, Boualga.K Khoudri.A from (C.A.C of Blida, Algaria). whose expertise our images .

6. REFERENCES

[1] Shapiro, J.M, "Embedded image coding using zero trees of wavelet coefficients", IEEE Transction .on Signal Processing, Vol.41, No.12, 1993, p.3445-3462.

[2] A. Said, W. Pearlman, "A new fast and efficient image coder based on set partitioning on Hiearchical Trees", IEEE Transaction Circuit and Systems for Video Technology, Vol.6, No.3,1996, p.243-250.

[3] D. Taubman, "High performance scalable image compression with EBCOT" IEEE Transactions .on Signal Processing, Vol.9, No.7, 2000, p.1158-70.

[4] Hong, W. Ling, L. Da-shun, Q. XunL,"image compression based on wavelet transform and vector quantization" Proc. IEEE Conf.on machine learning and cybernetics .Beijing,0-7803-7508,novembre 2002.

[5] Tracy, D. Keshab, K.P. Vladimir, "combining neural networks and the wavelet transform for image compression", IEEE . 0-7803-0946-4, 1993.

[6] B.J. Erickson, "Irreversible compression of medical images", Journal of Digital Imaging, Vol.15, No.1, mars 2002, p.5-14.

[7] W. Kurdthongmee, "A novel Kohonen SOM-based image compression architecture suitable for moderate density FPGAs, Elsevier image and vision computing",2008.

[8] M. Porrmann., H. Kalte, U. Witkowski, J.C. Niemann, U. R¨uckert, "A dynamically reconfigurable hardware accelerator for self-organizing feature maps", Proc. 5th World Multi-Conference *vo*lume 3, Orlando, Florida, USA, July 2001. p 242–247.

[9] J. Grando, M. A. Vega, R. Pérez, J. M. Sànchez, J. A. Gomez, "Using FPGAs to Implement Artificial Neural Networks", IEEE 1-4244-0395-2/06/2006.

[10] A. Muthuramalingam, S. Himavathis, E. Srinivasan, "neural network implementation using FPGA: issues and application", International journal of information technology, Vol. 4, No.2, ISSN 1305-2403, 2007.

[11] T. Heskes, "Self-Organizing Maps, Vector Quantization, and Mixture Modelling", IEEE transaction on neuronal networks, VOL. 12, NO. 6, november 2001.

[12] S.G Mallat, "A theory for multiresolution signal decomposition: The wavelet representation", IEEE transactions on pattern Analysis and Machine Intelligence, Vol 2 , No .7, July 1989.

[13] H. Bessalah, F. Alim, M. A. Bencherif, S. Seddiki**, "**Toward an Embedded Image Wavelet Transform Implementation Approach", SHAKERVERLAG ; Transactions on Systems, Signals & Devices Vol. 2, No. 3,. TSSD 1861-5252/ c 2006.

[14] H. Bessalah, F.Alim Ferhat, S.Seddiki, O.Kerdjidj " On Line Wavelets Transform on a Xilinx, FPGA Circuit to Medical Images Compression" International Conference on Digital Image Processing (ICDIP),Thailand march 2009.

[15] F. Alim Ferhat, H. Bessalah, Sofiane. Seddiki, "WT-SOM network implementation on FPGA for the medical images compression" Proceedings of the 5th international conference on Soft computing as transdisciplinary science and technology CSTST '08 : ISBN:978-1-60558-046-3 2008 Available at [http://portal.acm.org/results.cfm?coll= guide&dl=guide&cfid=20375883&cftoken=71581472.

[16] A.Lewis, G.Knowlles "Image Compression Using the 2-D Wavelet", IEEE Transactions on Image Processing, Vol. 1 *No. 2 1992. p.244-250.*

[17] S.McQuillan, J.V.McCanny, "A systematic methodology for the design of high performance recursive digital filters", ieee transactions on computers, vol 44, No. 8, august 1995.

[18] A. Avizienis, "Signed-Digit number representations for fast parallel arithmetic", IRE Transactions on Electronic Computers, September 1961, p389–400

[19] M. D. Ercegovac, “On-line Arithmetic for Recurrence Problems”, SPIE vol. 1556 Adv. Sig. Proc. Alg. Arch. and Imp.II (1991).

[20] T. Kohonen, “The self-organizing map”, Proc. of the IEEE, 78(9) , September 1990, 1464-1480.

[21] C. Foucher, “Algorithmes neuronaux et non neuronaux de construction de dictionnaire pour la quantification vectorielle en traitement d'images”, thèse en traitement du signal, Université de Rennes, Décembre 2002.

[22] MEDEISA Medical Datatbase for the Evaluation of Image and Signal Processing, 2006.

[23] Ali. Nait, A. Christine, Cavaro Ménard, “compression des images et des signaux médicaux”, ISBN 978-2-7462-1493-4, lavoisier, 2007

[24] http://www.medeisa.net/

[25] http://www.Xilinx.com