

REVERSIBLE INTEGER-TO-INTEGERS WAVELET TRANSFORMS FOR IMAGE COMPRESSION: IMPLEMENTATION AND EVALUATION

M. A. Ben Ayed, M. Loulou, N. Masmoudi, L. Kamoun

Laboratoire d'Electronique et des Technologies de l'Information

Ecole Nationale d'Ingénieurs de Sfax

B.P: W 3038 Sfax, Tunisie

Nouri.Masmoudi@enis.rnu.tn

(Received 29 January 2001 Accepted 12 November 2002)

ABSTRACT

There has been a growing interest in the reversible integer to integer wavelet transforms for image coding. In this paper, several of those particular transforms are integrated using a VHDL description on a FPGA circuit. Two different architectures are proposed, one uses the lifting framework as an architectural support and the other takes advantage of the two finite impulse response filter (FIR) structures representing the wavelet transform function. Evaluation is performed on the basis of their computational complexity, latency, hardware occupancy, and finally maximum operating clock frequency of the circuit obtained. Of the transforms considered, some of them seem to perform particularly well depending on the architecture used.

Keywords: image coding, codec, compression, VHDL, FPGA circuit, CLB, sub band transform, reversible integer to integer

I. INTRODUCTION

In the past few years, there has been a great amount of interest in wavelet image compression. In fact, JPEG 2000 suggest wavelet transforms instead of the discrete cosine transform (DCT) to compress still images. Also, in MPEG-4 (ISO/IEC 14496-2, 1998), wavelet transform is one candidate method for the video inter-frame coding. These extreme recommendations for the wavelet transform are due to the fact that for the case of wavelet based video codec, the temporal redundancy is reduced by block matching (Musmann *et al.*, 1985); the spatial redundancy is removed by the discrete wavelet transform (DWT) (Mallat, 1989 a, b). For still image compression, DWT outperforms the DCT in almost all criteria such as compression rate, minimum hardware occupancy, blocking artifacts and computational complexity as we will show in the later sections.

Different wavelet transform functions have been proposed for still image and video codec (Adams and Kossentini, 2000). From those functions there has been a growing interest in the reversible integer to integer wavelet transform. This comes from the fact that it maps integer to integer. Indeed, such transforms are invertible in finite-precision arithmetic (*i.e.*, reversible), and approximate the linear wavelet transforms from which they are derived.

Efficient system architecture design for the DWT has received much attention recently (Rioul and Duhamel, 1992; Ortega *et al.*, 1999; Jiang *et al.*, 1999). These authors have focused on two major fields (Jiang *et al.*, 2001): the memory necessary for the DWT computation, which is mostly in sequential and communication overhead required by parallel DWT algorithm.

In this paper several reversible integer-to-integer wavelet transforms are implemented using a VHDL description and synthesized onto an FPGA board. Two different architectures are proposed: one uses the lifting framework and the other takes advantage of the two FIR filter structures representing the wavelet transform function. Then, a comparison on the basis of their computational complexity, latency, hardware occupancy, and finally maximum operating clock frequency of the circuit obtained is conducted. By using the results of such an evaluation, not only one is able to build systems with improved compression efficiency, but can also quantify the tradeoffs between computational complexity and the throughput of the circuit.

The remainder of this paper is structured as follows: Section II begins by introducing the discrete wavelet transform and the multiresolution analysis with an emphasis on the two proposed frameworks of architectures for implementation. This is followed in Section III, by a brief comparison of the proposed integer to integer transforms in terms of computational complexity and techniques of implementation. Then, in Section IV, various experimental results are presented and analyzed in detail. Finally, it is concluded with a brief summary of the results and some closing remarks in Section V.

II. DWT AND MULTIREOLUTION ANALYSIS

Meyer and Mallat (Mallat, 1989 a, b) reported in 1986 that orthonormal wavelet decomposition and reconstruction can be implemented in the multiresolution signal analysis framework. The multiresolution analysis is now a standard way to construct orthonormal wavelet bases. In fact, it permits us to analyze a signal in multiple frequency bands. There are two existing approaches of multiresolution analysis: the Laplacian pyramid and sub band coding, which were developed independently in the late 1970s (Burt and Adelson, 1983). However, sub band coding is the mostly used technique for the construction of the wavelet transforms. In fact, its objective is to divide the signal spectrum into independent sub bands in order to treat the signal sub bands individually for different purposes. Sub band coding is an efficient tool for multiresolution spectral analysis and has been successfully applied in speech coding and analysis (Akunsu and Haddad, 1992).

Given an original data sequence $X(n)$, $n \in Z$. The lower resolution approximation of the signal $\lambda(n)$ is derived by low pass filtering with a FIR filter having its impulse response $h'(n)$:

$$\lambda(n) = \sum_k h'(k - 2n) X(k) \tag{1}$$

That is the correlation between $X(n)$ and $h'(n)$ down sampled by a factor of two. In order to compute the detail information $\gamma(n)$ lost by low pass filtering with $h'(n)$, a FIR band pass filter with the impulse response $g'(n)$ is applied to the initial sequence $X(n)$ as:

$$\gamma(n) = \sum_k g'(k - 2n) X(k) \tag{2}$$

That is the correlation between $X(n)$ and $g'(n)$ down sampled by a factor of two. Note that IIR filters can also be used, they have the disadvantage that their infinite response leads to infinite data expansion (Morlet *et al.*, 1982). For any practical use of IIR filter bank the output data stream has to be cut which leads to a loss of data. This looks only at FIR filters.

The above two equations are illustrated by Figure 1.

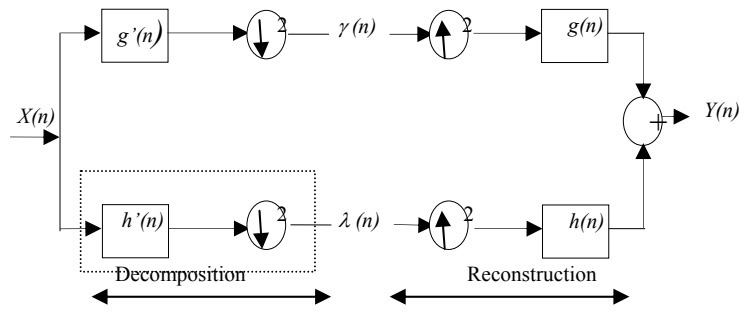


Figure 1. A one-stage filter bank for signal analysis and reconstruction.

Figure 1 shows how a one-stage wavelet transform uses two analysis filters, each one of them followed by a sub sampling for the forward transform. Therefore, it seems only logical to construct the inverse transform by first performing an up sampling step and then to use two synthesis filters h (low pass) and g (band pass) to reconstruct the signal. The conditions for perfect reconstruction are now given by (Daubechies and Sweldens, 1998) as:

$$\begin{aligned} h(n)h'(n^{-1}) + g(n)g'(n^{-1}) &= 2 \\ h(n)h'(-n^{-1}) + g(n)g'(-n^{-1}) &= 0 \end{aligned} \tag{3}$$

III. POLYPHASE REPRESENTATION

The polyphase representation is now derived. In fact, in the filter stage shown in the left part of Figure 1 the signal is first filtered and then sub sampled. In other words, one throws away half of the filtered samples and keep only the even numbered samples. Clearly,

this is not efficient and therefore one would like to do the sub sampling before the filtering in order to cut the computational effort by a factor of two at least (Jer Min Jou *et al.*, 2001).

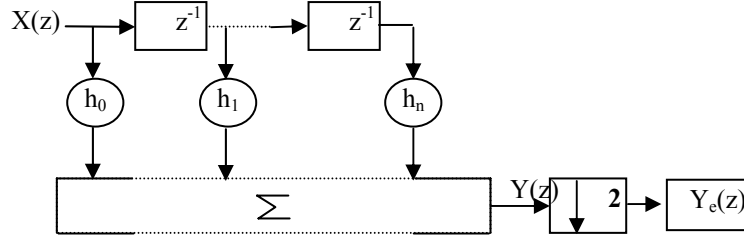


Figure 2. A standard FIR filter with subsampling of the output.

In Figure 2, a standard FIR filter in a general form having an impulse response $h(n)$ is drawn followed by a sub sampling. If one writes down its output signal $y(n)$, one obtains the following equation:

$$y(n) = \sum_{i=0}^{\infty} h(i).x(n-i) \quad (4)$$

If we apply the z-transform just before the sub sampling, for several consecutive samples we obtain:

$$\begin{aligned} & \vdots \\ y_0 &= h_0 x_0 + h_1 x_{-1} Z^{-1} + h_2 x_{-2} Z^{-2} + \dots \\ y_1 &= h_0 x_1 + h_1 x_0 Z^{-1} + h_2 x_{-1} Z^{-2} + \dots \\ y_2 &= h_0 x_2 + h_1 x_1 Z^{-1} + h_2 x_0 Z^{-2} + \dots \\ & \vdots \end{aligned} \quad (5)$$

Sub sampling of $y(z)$ will now remove the middle line in (5) and all the other odd lines, the sub sampled output signal $y_e(z)$ could then be rewritten as:

$$y_{even}(z) = h_{even}(z).x_{even}(z) + Z^{-1}.h_{odd}(z).x_{odd}(z) \quad (6)$$

The above equation can be rewritten in time domain as follow:

$$y_{even}(n) = \sum_{i=0}^{\infty} h_{even}(i).x_{even}(n-i) + h_{odd}(i-1).x_{odd}(n-i-1) \quad (7)$$

with

$$h_{even}(n) = h(2n), \quad h_{odd}(n) = h(2n+1), \quad x_{odd}(n) = x(2n+1), \quad \text{and} \quad x_{even}(n) = x(2n)$$

One can now come up with a more efficient implementation of a FIR filter with sub sampling of the output (Jer Min Jou *et al.*, 2001). Figure 3 illustrates clearly the internal architecture of the dashed box represented in Figure 1.

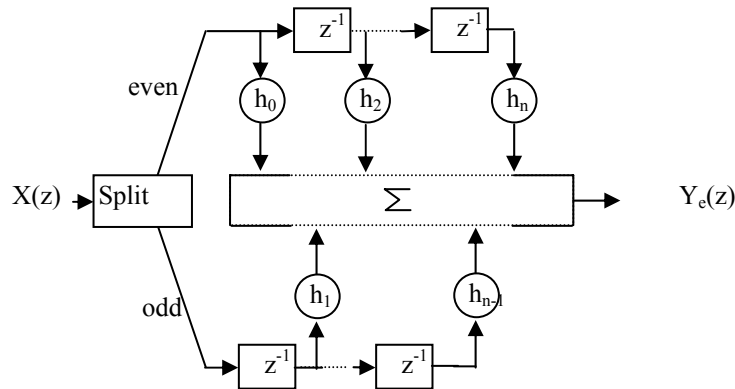


Figure 3. More efficient implementation of FIR filter with sub sampling.

If one applies this remodeled FIR filter to the wavelet transform filter stage in Figure 1, one ends up with two equations like (6), one for each filter, so that if switched to vector notation, one can write:

$$\begin{pmatrix} \lambda(z) \\ \gamma(z) \end{pmatrix} = P'(z) \begin{pmatrix} x_e(z) \\ z^{-1}x_o(z) \end{pmatrix} \tag{8}$$

where $P'(z)$ is the polyphase matrix:

$$P'(z) = \begin{pmatrix} h'_e(z) & h'_o(z) \\ g'_e(z) & g'_o(z) \end{pmatrix} \tag{9}$$

For the inverse wavelet transform illustrated by the right part of Figure 1, looking again at equation 2 but now imagining it as being the result of filtering an up-sampled sequence of samples. If one assumes that the inserted zeroes are the odd samples, then all terms with an odd numbered $x(z)$ vanish and one can divide the output samples $y(z)$ in odd and even sequences:

$$\begin{aligned} y_e(z) &= h_e(z)x_e(z) \\ z y_o(z) &= h_o(z)x_e(z) \end{aligned} \tag{10}$$

An equation similar to (8) in the vector notation form for the reconstruction filter stage can now be written:

$$\begin{pmatrix} y_e(z) \\ z y_o(z) \end{pmatrix} = P(z) \begin{pmatrix} \lambda_e(z) \\ \gamma_e(z) \end{pmatrix} \quad (11)$$

where $P(z)$ is a second polyphase matrix, the dual of the first:

$$P(z) = \begin{pmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{pmatrix} \quad (12)$$

In the following Figure 4, the filter stage of Figure 1 is redrawn, this time using polyphase matrices.

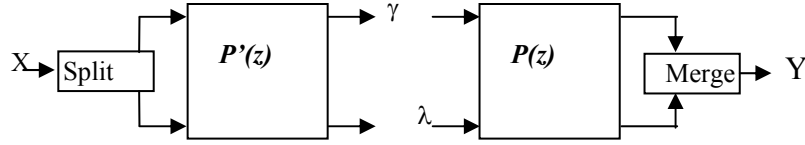


Figure 4. Polyphase representation for one-stage filter bank.

The polyphase representation will be the base of the first architecture. In fact, the original signal will be split in two parts, each one will be filtered by the corresponding coefficients of the low-band and pass-band filters depending on equations (6) and (10).

IV. LIFTING REPRESENTATION

It is clear that the condition for perfect reconstruction can be written as:

$$P'(z^{-1}) P(z) = I \quad (13)$$

Assume that $P(z)$ is invertible and using Cramer's rule to calculate its inverse, one finds:

$$P(z)^{-1} = P'(z^{-1}) = \frac{1}{h_e(z)g_o(z) - h_o(z)g_e(z)} \begin{pmatrix} g_o(z) & -g_e(z) \\ -h_o(z) & h_e(z) \end{pmatrix} \quad (14)$$

From this, it follows that if one demands that the determinant of $P(z)=1$, i.e. $h_e(z)g_o(z) - h_o(z)g_e(z) = 1$, then not only will $P(z)$ be invertible, but also

$$\begin{aligned} h'_e(z) &= g_o(z^{-1}) \\ h'_o(z) &= -g_e(z^{-1}) \\ g'_e(z) &= -h_o(z^{-1}) \\ g'_o(z) &= h_e(z^{-1}) \end{aligned} \quad (15)$$

Note that if the determinant of $P(z)=1$ the filters $h(z)$ and $g(z)$, $h'(z)$ and $g'(z)$ will be relatively prime. The polyphase matrix is a matrix of Laurent polynomials and since it is demanded that its determinant has to be equal to 1, it is known that the filter pair (h,g) is complementary, the lifting theorem (Daubechies and Sweldens, 1998) now states that any such pair of filters can be factored into lifting steps. The polyphase matrix becomes of the form:

$$P'(z) = \begin{pmatrix} K_1 & 0 \\ 0 & K_2 \end{pmatrix} \prod_{i=1}^m \begin{Bmatrix} \begin{pmatrix} 1 & s_i(z) \\ 0 & 1 \end{pmatrix} \\ \begin{pmatrix} 1 & 0 \\ t_i(z) & 1 \end{pmatrix} \end{Bmatrix} \quad (16)$$

where K_1 and K_2 are scaling constants unequal to zero and $s_i(z)$ and $t_i(z)$ are two polynomials. The filter stage of Figure 1 is redrawn in Figure 5, this time using a lifting scheme.

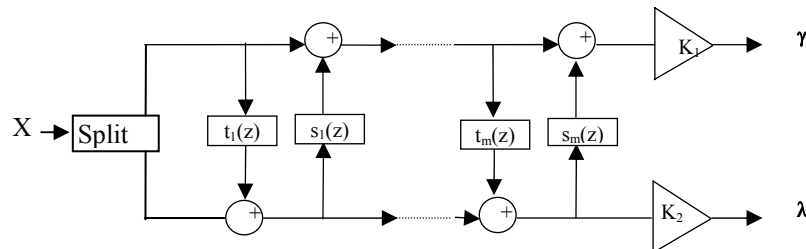


Figure 5. One stage of a filter bank using lifting schema.

The lifting schema has three properties that are not found in many other transforms. In fact, the inverse transform is immediately clear: change the sign of all scaling factors, replace split by merge and go from right to left, *i.e.* reverse the data flow. This easy invertibility is always true for all lifting schemes. The next property is that lifting is not causal (Sweldens, 1997). Usually this is not really a problem, we can always delay the signal enough to make it causal, which will increase latency. The last important property is that in (Daubechies and Sweldens, 1998) it is proven that for long filters, the lifting scheme reduces computational complexity in half, compared to the standard iterated filter bank algorithm. In this paper, we will see if this schema can perform better than the polyphase representation of the wavelet transform.

The two different representations proposed in this paper have the ability to over feet any DWT. There are probably as many wavelet transforms as there are wavelets. However, the most used in the still image and video codecs are the wavelets that map integer to integer due to their simplicity of implementation.

V. THE PROPOSED INTEGER TO INTEGER TRANSFORM AND THEIR COMPLEXITY

As mentioned earlier, reversible integer to integer version of transforms have the potential for simpler implementation than their conventional counterpart. In the conventional

case, all operands are real-valued. In the reversible integer to integer case, however, many operands are integer. For that reason, all the reversible transforms considered in this study are calculated using only fixed-point arithmetic. In particular, only integer addition, subtraction, multiplication, and division operations are required.

The forward transformation equations for each of the transforms are given in Table 1 (Adams and Kossentini, 2000). The N/M wavelet notation represents the analysis and synthesis filter's degree respectively. The input signal, low-pass subband signal, and band-pass subband signal are denoted as $x[n]$, $s[n]$, and $d[n]$, respectively. For convenience, we also define the quantities $s_0[n] \Leftrightarrow x[2n]$ and $d_0[n] \Leftrightarrow x[2n+1]$. The inverse transformation equation can be trivially deduced from the forward transformation equations, and thus are not given.

TABLE 1

Forward Transformations

$$\begin{aligned}
 &5/3 \left\{ \begin{aligned} d[n] &= d_0[n] - \lfloor \frac{1}{2}(s_0[n+1] + s_0[n]) \rfloor \\ s[n] &= s_0[n] + \lfloor \frac{1}{4}(d[n] + d[n-1]) + \frac{1}{2} \rfloor \end{aligned} \right. \\
 &2/6 \left\{ \begin{aligned} d_1[n] &= d_0[n] - s_0[n] \\ s[n] &= s_0[n] + \lfloor \frac{1}{2}d_1[n] \rfloor \\ d[n] &= d_1[n] + \lfloor \frac{1}{4}(-s[n+1] + s[n-1]) + \frac{1}{2} \rfloor \end{aligned} \right. \\
 &5/11-C \left\{ \begin{aligned} d_1[n] &= d_0[n] - \lfloor \frac{1}{2}(s_0[n+1] + s_0[n]) \rfloor \\ s[n] &= s_0[n] + \lfloor \frac{1}{4}(d_1[n] + d_1[n-1]) + \frac{1}{2} \rfloor \\ d[n] &= d_1[n] + \lfloor \frac{1}{16}(s_1[n+2] - s_1[n+1] - s_1[n] + s_1[n-1]) + \frac{1}{2} \rfloor \end{aligned} \right. \\
 &9/7-F \left\{ \begin{aligned} d_1[n] &= d_0[n] + \lfloor \frac{1}{128}(203(-s_0[n+1] - s_0[n])) + \frac{1}{2} \rfloor \\ s_1[n] &= s_0[n] + \lfloor \frac{1}{4096}(217(-d_1[n] - d_1[n-1])) + \frac{1}{2} \rfloor \\ d[n] &= d_1[n] + \lfloor \frac{1}{128}(113(s_1[n+1] + s_1[n])) + \frac{1}{2} \rfloor \\ s[n] &= s_1[n] + \lfloor \frac{1}{4096}(1817(d_1[n] + d_1[n-1])) + \frac{1}{2} \rfloor \end{aligned} \right.
 \end{aligned}$$

Reversible integer-to-integer wavelet transforms approximate their parent linear transforms. For that reason, the characteristics of the parent transforms are of a great interest. Consequently, the number of vanishing moments of analysis and synthesis wavelet functions and the regularity of the analysis and synthesis scaling functions are given for each transforms in Table 2 in order to analyze in an objective manner the obtained experimental results.

VI. TARGET IMPLEMENTATION

In the early 1980s, in the quest for more flexibility and rapid prototyping at low cost, re-configurable hardware in the form of Field Programmable Gate Arrays (FPGAs) was

introduced into the IC market. FPGAs are an efficient hardware target when only small scale production is needed because it reduces the time to market. Consequently, FPGAs offers attractive combination of low cost, high performance, and flexibility. The internal architecture of FPGAs is characterized by a set of Configurable Logic Blocks (CLBs) that have exactly the same structure. A CLB has 13 inputs and 4 outputs that can be connected to other CLB inputs and outputs or to input-output blocks (IOB). Inside each CLB there are two four input logic function generators and one three input logic function generator. The inputs to the first function generator, F1, F2, F3, and F4, come from outside the block. The inputs to the second function generator, G1, G2, G3, and G4, also come from outside the block. The outputs of these two function generators, F and G, are two of the three inputs to the third function generator, which creates output H. The third input, H1, comes from outside the block. Any two of these three outputs, F, G, and H, may leave the CLB for connection to another CLB. The remaining four inputs and two outputs are associated with the sequential part of the CLB. Each configurable logic block has two D-type flip-flops. Both flip-flop outputs leave the CLB.

TABLE 2

Wavelet Transforms: (a) Parameter Values, and (b) Parameter Definitions

(a)

Transform	N'	N	R'	R
5/3	2	2	0.000	1.000
2/6	3	1	0.000	1.000
5/11-C	4	2	0.000	2.142
9/7-F	4	4	1.034	1.701

(b)

Parameter	Definition
N'	Number of vanishing moment of analysis wavelet
N	Number of vanishing moment of synthesis wavelet
R'	Regularity of analysis scaling function
R	Regularity of synthesis scaling function

In order to implement the proposed wavelets, we have generated an adequate and efficient structural VHDL code for each architectural framework of the proposed wavelets. The structural VHDL definitions have been synthesized using Foundation Express version 4.1 from Xilinx. For the place and route and bit generation, we rely on Xilinx low-level tools. In the following Figures 6 and 7, we will illustrate the structure of a VHDL code for the two different architectures for the direct transformation in other words the analysis part. The structure of the inverse transformation can be trivially deduced.

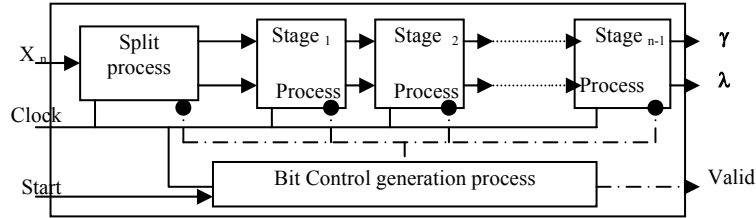


Figure 6. Structural VHDL representation of one stage of a wavelet decomposition using lifting schema.

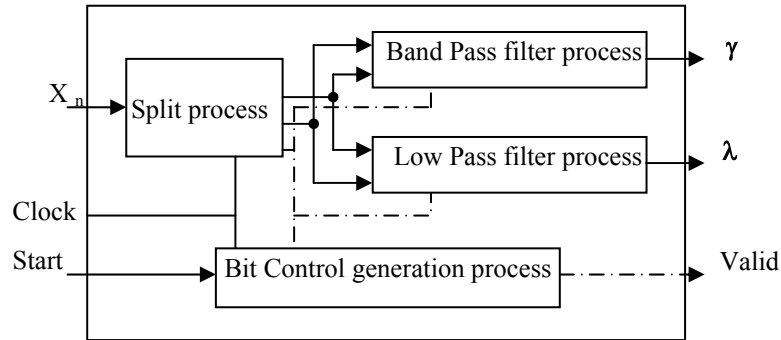


Figure 7. Structural VHDL representation of one stage of a wavelet decomposition using polyphase schema.

VII. ANALYSIS OF THE EXPERIMENTAL RESULTS

All the VHDL codes have been implemented on the Xilinx XC 4010 EPQ 208-1 FPGA (Speed grade -1). Tables 3 and 4 show the experimental results.

From an objective point of view and looking to the above Tables, no one can predict that there is a wavelet that outperforms all the others. In fact, some perform better in one architectural schema and fail in the other. At the same time, the two different architectural frameworks, studied in our case, gave different results for different wavelets and there is no major over performing noticed in our experimental results. Probably, having an idea about the characteristics of the parent transforms as described in the previous section, one might understand and analyze objectively the obtained result. First of all, the hardware occupancy in terms of number of used CLBs seems to have a major proportionality with respect to the number of the vanishing moment of the corresponding analysis and synthesis wavelet. In fact, as the number of vanishing moments increases the number of occupied CLBs increases too and vice versa. Secondly, we notice a major link between the regularity of the mother wavelet and the maximum delay in the corresponding transforms. In fact, the 5/11-c transform with

the most considerable regularity in its synthesis wavelet has got the maximum pin to pin delay. Thirdly, the maximum clock frequency depends heavily on the maximum delay. Fourthly, lifting architecture requires much more latency in terms of clock cycles. This is obvious since one needs more stages in the lifting schema and we add delay to make the lifting schema causal. Finally, the most interesting remark is that lifting architecture performs better in terms of hardware occupancy and operating frequency for the transforms with long filters. In fact, there is a major reduction in the computational complexity. This final and exciting statement has been proved by Daubechies (Daubechies and Sweldens, 1998), and confirmed experimentally in this study.

TABLE 3
Lifting Architecture: (a) Direct Transformation, (b) Inverse Transformation

(a):

Transforms	N° CLBs	Max Delay (Ns)	Latency (period)	Frequency (MHZ)
5/3	94	10.106	10	40
2/6	50	6.844	8	15
5/11-C	110	8.606	16	50
9/7-F	207	24	13	10

(b)

Transforms	N° CLBs	Max Delay (Ns)	Latency (period)	Frequency (MHZ)
5/3	76	11.9	7	40
2/6	45	12	7	40
5/11-C	74	54	7	10
9/7-F	204	15.97	9	20

TABLE 4
Polyphase Architecture: (a) Direct Transformation, and (b) Inverse Transformation

(a)

Transforms	Number of CLBs	Max Delay (Ns)	Latency (period)	Frequency (MHz)
5/3	65	10.818	6	50
2/6	59	8.238	7	40
5/11-C	163	10.011	11	50
9/7-F	388	24	11	15

(b)

Transforms	Number of CLBs	Max Delay (Ns)	Latency (period)	Frequency (MHz)
5/3	70	12.025	6	40
2/6	53	12	6	40
5/11-C	167	36.5	6	15
9/7-F	222	13.658	7	33

CONCLUSION

In this paper several reversible integer-to-integer wavelet transform were implemented using a VHDL description on a FPGA circuit technology. Two different architectures were proposed, one used the lifting framework as an architectural support and the other made advantage of the two FIR filter structures representing the wavelet transform function. Then, a comparison on the basis of their hardware occupancy, latency, and finally maximum operating clock frequency of the circuit obtained is conducted. We have shown that a number of characteristics of the linear parent transforms potentially influences the effectiveness of the implementation, including the number of vanishing moments of analysis and synthesis wavelet functions, the regularity of the analysis and synthesis scaling functions. Finally we concluded by the statement that lifting schema could outperform the polyphase form for the transforms that have long filters.

REFERENCES

- Adams, M.D. and Kossentini, F. 2000. Reversible integer to integer Wavelet Transforms for image compression: Performance Evaluation and Analysis. *IEEE Trans. On Image Processing*, 9(6): 1010-1024.
- Akunsu, A. N., and Haddad, R. A. 1992. *Multiresolution signal decomposition*. Academic Press, Boston.
- Burt, P.J. and Adelson. 1983. The Laplacien Pyramid as a compact image code. *IEEE Trans. On Comm.*, 31(4): 532-540.
- Daubechies, I. and Sweldens, W. 1998. Factoring Wavelet Transforms into lifting steps. *J. Fourier Anal. Appl.*, 4(3).
- ISO/IEC JTC1/SC29/WG11 N1902, FDIS of ISO/IEC 14496-2, Nov. 1998. MPEG-4 Video Group, Generic coding of audio-visual object Part 2-Visual, Atlantic city,.
- Jer Min Jou, Yeu-Horng, S. and Chin-Chi, L., 2001. Efficient VLSI architecture for the biorthogonal wavelet transform by filter bank and lifting scheme. *IEEE International Symposium on Circuits and Systems*, 2: 529-532.
- Jiang, W. and Ortega, A. 1999. Efficient DWT system architecture design using filter bank factorization. in *Proc. ICIP*, 2: 749-753.
- Jiang, W. and Ortega, A. 2001. Lifting Factorization-based DWT architecture design. *IEEE Trans. On circ. For video Tech.*, 11(5): 651-657.
- Mallat, S.G. 1989a. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Pattern Anal. Machine Intell.*, 11: 674-693.
- Mallat, S.G. 1989b. Multifrequency channel decompositions of images and wavelet models. *IEEE Trans. On Acous. Speech and Sig. Proc. ASSP-37*, 12: 2091-2110.
- Morlet, J. and Arens, G., Fourgeau, I., Giard, D. 1982. Wave, propagation and sampling theory. *Geophysics*, 47: pp. 203-236.
- Musmann, H. G., Pirsch, P. and Grallert, H.J. 1985. Advances in picture coding. *Proc. IEEE* 73: 523-548.
- Ortega, A., Jiang, W., Fernandez, P. and Chrysafis, C.G. 1999. Implementations of the discrete wavelet transform: complexity, memory, and parallilezation issues, in *Proc. SPIE: Wavelet application in signal processing VII*. 3813: 386-400.
- Rioul, O., Duhamel, P. 1992. Fast algorithms for discrete and continous wavelet transforms. *IEEE Trans. Info. Theo.*, 38: 569-586.
- Sweldens, W. 1997. The lifting scheme: a construction of second generation wavelets. *Siam J. Math. Ana.*, 29(2).