

Design and FPGA Implementations of Four Orthogonal DWT Filter Banks Using Lattice Structures

Dr. Jassim M. Abdul-Jabbar

Zainab R. S. Al-Omary

University of Mosul. College of Engineering, Computer Engineering Department,

Abstract

In this paper, lattice structures for DWT are introduced through the design and FPGA implementations of the orthogonal Daubechies filter banks of orders 2, 4, 6 and 8. Multipliers and shift-add methods are both used to perform multiplication operations for these types of filter banks. Two implementation techniques are introduced, namely; the pipelining technique that is efficient from the throughput point of view, and the area efficient bit-serial implementation technique. The obtained results show the ability to achieve high throughput using pipelining (with 2 output samples / clock) on behalf of the area allocation. While bit-serial technique minimizes the allocated area on behalf of the throughput which may decrease with increasing filter order. As compared with other recent implementations, the results of implementing the designed filter banks using the SPARTAN-3E FPGA kit are efficient in minimizing implementation complexity to 0.584 - 0.712 of its corresponding values for different structures in recent hardware implementations. It is also obtained that the resulting structures can operate at high frequencies (up to 47.09 MHz).

Keywords: Lattice structures, DWT, Filter banks, FPGA, Bit-serial, Pipelining.

التصميم والبناء باستخدام FPGA

لأربعة أجراف مرشحات متعامدة للتحويل المويجي المقطع باستعمال الهياكل المتشابكة

زينب رامي صالح العمري

د. جاسم محمد عبد الجبار

جامعة الموصل- كلية الهندسة- قسم هندسة الحاسوب

الخلاصة

في هذا البحث تم تقديم الهيكل الشبكي للتحويل المويجي المقطع من خلال تصميم أربعة من أجراف المرشحات (Daubechies) المتعامدة ذات المرتبة 2 و 4 و 6 و 8 وتنفيذها باستخدام (FPGA). وقد أجريت عملية الضرب في أجراف المرشحات عملياً باستخدام كل من الضرب المباشر وطريقة الترحيف والجمع، وللتقنيتين المقترحتين، حيث تم التصميم باستخدام تقنية خطوط الأنابيب (Pipelining) ذات الإنتاجية العالية وتقنية (Bit-serial) ذات الكفاءة العالية من ناحية المساحة المحجوزة للتصميم. وقد أظهرت النتائج الموضحة في البحث إمكانية الحصول على إنتاجية باستخدام خطوط الأنابيب تصل إلى إخراجين في النبضة الواحدة على حساب زيادة المساحة المحجوزة مقارنة بتقنية (Bit-serial) والتي تؤدي إلى حيز أقل مساحة ممكنة لتنفيذ التصميم على حساب إنتاجية النظام والتي تنخفض بزيادة مرتبة المرشح. أظهرت نتائج بناء أجراف المرشحات المصممة باستخدام جهاز (SPARTAN-3E) كفاءة في تقليل تعقيد المعماريات التي تم بناؤها وامتلاكها لنسبة استغلالية للكائنات المادية المتاحة لها تصل إلى 0.584 - 0.712 من نسبتها في معماريات سابقة لطرائق بناء مشابهة كانت تناولت هياكل مختلفة استخدمت لتنفيذ ذات المرشحات. هذا بالإضافة إلى الحصول على هياكل ناتجة تشتغل في ترددات عالية تصل إلى 47.09 MHz.

Received: 14 - 1 - 2010

Accepted: 11 - 1 - 2011

1. Introduction

The wavelet transform is the representation of a function by wavelets, providing a time-frequency representation of the signal. The transform was initially in a continuous form and is called the Continuous Wavelet Transform (CWT), which gives the wavelet coefficients in a detailed manner. The redundancy accompanies this type of transformation brought on the need to a sampled version of the Wavelet Transform [1]. That is the Discrete Wavelet Transform (DWT), DWT is a powerful tool for signal processing and analysis [2]. The success of DWT stems from its ease of computation and its inherent decomposition of an image into non-overlapping subbands that enables the design of efficient algorithms and allows for incorporation of the human visual system [3].

The DWT of a signal $x[n]$ is computed by passing it through a series of filters. First the samples are passed through a lowpass filter with impulse response $H_0[n]$ giving approximation coefficients. The signal is decomposed simultaneously using a highpass filter $G_0[n]$, giving the detailed coefficients. Since half the frequencies of the signal are removed, the filter outputs are down sampled by 2 [4].

The recent development of DWT, which can be implemented efficiently with a filter bank, has provided a way to efficiently utilize wavelets for signal processing [2]. DWT has four different structures to be implemented with, they are the direct form, polyphase, lifting and the lattice structures. Each of these structures has its advantages and drawbacks which have to be considered according to the specific application being implemented [5].

The availability of high-density, low-cost FPGA devices has given digital designers lots of flexibility to design custom digital architectures [6], where the FPGA is the hardware platform that can be used to implement just about any hardware design by writing a description for the digital circuit using a Hardware Description Language (HDL) [7].

The rest of this paper is organized as follows: section 2 introduces the lattice structures for DWT, Section 3 explains the design of the lattice DWT for a group of the Daubechies filter banks. Matlab7.6 programming is used in section 4 to verify the results and to estimate a suitable coefficient's wordlength to work with. The implementation results are given in section 5, when implementing using a Xilinx FPGA device. A comparative study of the implemented filters with other implementations is given in section 6. And finally, section 7 concludes this paper.

2. Lattice Structures

The wavelet filter banks have highly efficient lattice structures which are easy to implement [5], they preserve both modularity and regularity with a high degree of parallelism that allows for high throughput [8].

The lattice structure has many advantages, such as better coefficient quantization response as well as a reduction by a factor of two of the stages needed for a given filter order [9]. That reduces the number of coefficients and thus the number of multipliers.

An efficient hardware implementation of the lattice structure of wavelet transform can be accomplished using a number of Processing Elements (PE), that are a chain of similar parts that work together to solve a single problem. They are arranged in a regular form, where the computational demand on each individual processing element is quite low. In the lattice structure, a processing element in both of the analysis and synthesis sides consists of two adders and two multipliers. A single delay element appears before each PE at the analysis side and after it at the synthesis side with a scaling gain appears at the end of each side. The i^{th} processing element at the analysis side of the lattice structure is shown in Fig. 1.

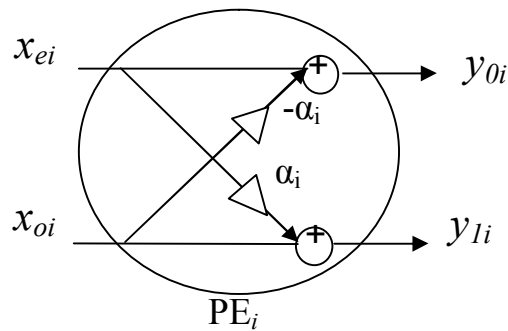


Fig.1: The i^{th} stage lattice DWT processing element, x_{ei} and x_{oi} are the even and odd inputs of the i^{th} stage, respectively, α_i is the value of the lattice coefficient at the i^{th} stage, and y_{oi} & y_{li} refer to the forward even and odd results of the i^{th} stage, respectively

The lattice DWT offers a novel tool for the design of multiplexer-demultiplexer (Mux-Demux) information channels such as data transmission in optical fiber networks and telemetric multi-channel equipments, where parallel data channels should be multiplexed into one signal, transmitted and then reconstructed [10]. Figure 2 shows an example of the dual-channel mux-demux arrangement using the lattice DWT synthesis and analysis parts.

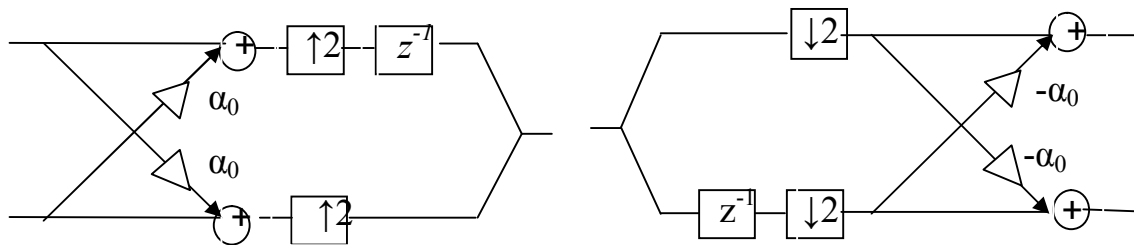


Fig.2 Lattice DWT- based construction of the dual-channel Mux-Demux unit.

The design is accomplished in the next section for the Daubechies filter banks of orders 2, 4, 6 and 8. These filter banks are orthogonal and have the characteristics of satisfying the perfect reconstruction property [11], exploiting large number of vanishing moments which makes them effective feature extraction tools [12]. They also use overlapping windows, so the high frequency coefficient spectrum reflects all high frequency changes. Therefore, Daubechies wavelets are useful in compression and noise removal of audio signal processing [11].

3. Filter Design

The first design step is to find the polyphase matrix $H_p(z)$ of the specified filter bank and a similar matrix of the lattice structure. Where the filters' polyphase representation is expressed as [13]

$$H_0(z) = H_{0\text{even}}(z^2) + z^{-1}H_{0\text{odd}}(z^2) \quad \dots (1)$$

$$G_0(z) = G_{0\text{even}}(z^2) + z^{-1}G_{0\text{odd}}(z^2) \quad \dots (2)$$

The transfer function in matrix form for the polyphase structure will be (depending on the form given in ref.[13]):

$$\begin{bmatrix} y_0 \\ y_1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} x_e \\ z^{-1}x_o \end{bmatrix} \quad \dots (3)$$

Where x_e and x_o are used to denote the even and odd terms of input x , respectively, and the polyphase matrix $H_p(z)$ has the following form:

$$H_p(z) = \begin{bmatrix} H_{oeven}(z) & H_{oodd}(z) \\ G_{oeven}(z) & G_{oodd}(z) \end{bmatrix} \quad \dots (4)$$

Afterwards, a comparison between these two matrices will be achieved in order to obtain the lattice coefficients.

3.1 Daubechies-2 filter bank lattice design

For this type of filter banks, $H_0(z)$ and $G_0(z)$ are as follows [14]:

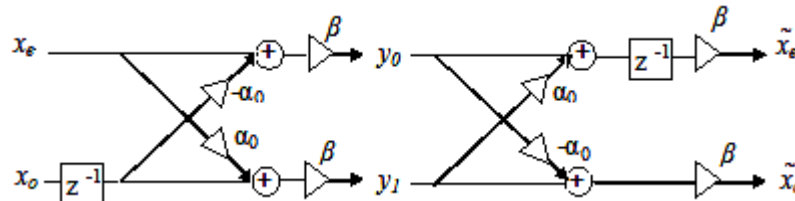
$$H_0(z) = \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}z^{-1} \quad \dots (5)$$

$$G_0(z) = \frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}z^{-1} \quad \dots (6)$$

From (3) and (4), the polyphase matrix will be as follows:

$$H_p(z) = \begin{bmatrix} H_{oeven}(z) & H_{oodd}(z) \\ G_{oeven}(z) & G_{oodd}(z) \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \quad \dots (7)$$

while the lattice transfer function of the analysis side, depending on the structure given in Fig. 3, is given in the matrix form as



$$\begin{bmatrix} y_1 \\ y_0 \end{bmatrix} = \begin{bmatrix} \alpha_0\beta & \beta \\ \beta & -\alpha_0\beta \end{bmatrix} \begin{bmatrix} x_e \\ z^{-1}x_o \end{bmatrix} = H_l(z) \begin{bmatrix} x_e \\ z^{-1}x_o \end{bmatrix} \quad \dots (8)$$

Fig.3 The analysis and synthesis sides of the Daubechies-2 lattice filter bank.

From (7) and (8), it can be easily obtained that the lattice coefficients are

$$\beta = \frac{1}{\sqrt{2}} \text{ and } \alpha_0 = 1$$

3.2 Daubechies-4 filter bank lattice design

For the Daubechies-4 filter bank, the two equations of $H_0(z)$ and $G_0(z)$ are [14]

$$H_0(z) = h_0 + h_1z^{-1} + h_2z^{-2} + h_3z^{-3} \quad \dots (9)$$

$$G_0(z) = -h_3z^2 + h_2z - h_1 + h_0z^{-1} \quad \dots (10)$$

where

$$h_0 = \frac{1+\sqrt{3}}{4\sqrt{2}}, h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}}, h_2 = \frac{3-\sqrt{3}}{4\sqrt{2}}, h_3 = \frac{1-\sqrt{3}}{4\sqrt{2}}$$

After separating $H_0(z)$ and $G_0(z)$ into even and odd parts, the polyphase matrix will be as follows:

$$H_p(z) = \begin{bmatrix} h_0 + h_2z^{-1} & h_1 + h_3z^{-1} \\ -h_3z - h_1 & h_2z + h_0 \end{bmatrix} \quad \dots (11)$$

while the corresponding lattice matrix depending on the design given in Fig. 4 will be as

$$H_l(z) = \begin{bmatrix} \beta(1 - \alpha_0\alpha_1z^{-1}) & \beta(\alpha_0 + \alpha_1z^{-1}) \\ \beta(-\alpha_1z - \alpha_0) & \beta(-\alpha_0\alpha_1z + 1) \end{bmatrix} \quad \dots (12)$$

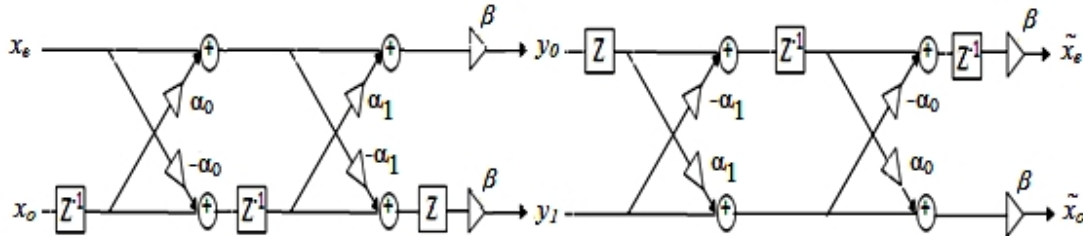


Fig.4 The analysis and synthesis sides of the Daubechies-4 lattice filter bank.

Thus the lattice coefficient values will be obtained by comparing (11) and (12) as $\alpha_0 = 1.732050808$, $\alpha_1 = -0.267949192$, and $\beta = 0.482962913$.

It can be noted that adding an advance element at the end of the analysis side make it be able to compare the two matrices of the polyphase and lattice structures.

3.3 Daubechies-6 filter bank lattice design

For this type of filter banks, the LPF and HPF equations are [14]

$$H_0(z) = h_{-2}z^2 + h_{-1}z + h_0 + h_1z^{-1} + h_2z^{-2} + h_3z^{-3} \quad \dots (13)$$

$$G_0(z) = -h_3z^2 + h_2z - h_1 + h_0z^{-1} - h_{-1}z^{-2} + h_{-2}z^{-3} \quad \dots (14)$$

where

$$h_{-2} = \frac{\sqrt{2}(1+\sqrt{10}+\sqrt{5+2\sqrt{10}})}{32}, \quad h_{-1} = \frac{\sqrt{2}(5+\sqrt{10}+3\sqrt{5+2\sqrt{10}})}{32}$$

$$h_0 = \frac{\sqrt{2}(10-2\sqrt{10}+2\sqrt{5+2\sqrt{10}})}{32}, \quad h_1 = \frac{\sqrt{2}(10-2\sqrt{10}-2\sqrt{5+2\sqrt{10}})}{32}$$

$$h_2 = \frac{\sqrt{2}(5+\sqrt{10}-3\sqrt{5+2\sqrt{10}})}{32}, \quad h_3 = \frac{\sqrt{2}(1+\sqrt{10}-\sqrt{5+2\sqrt{10}})}{32}$$

After separating $H_0(z)$ and $G_0(z)$ into even and odd parts, the polyphase matrix will be as follows:

$$H_p(z) = \begin{bmatrix} h_{-2}z + h_0 + h_2z^{-1} & h_{-1}z + h_1 + h_3z^{-1} \\ -h_3z - h_1 - h_{-1}z^{-1} & h_2z + h_0 + h_{-2}z^{-1} \end{bmatrix} \quad \dots (15)$$

while the lattice corresponding matrix will be as follows when utilizing Fig. 5:

$$H_l(z) = \begin{bmatrix} k_1\beta & k_2\beta \\ l_1\beta & -l_2\beta \end{bmatrix}$$

where

$$k_1 = z - \alpha_0\alpha_1 - \alpha_1\alpha_2 - \alpha_0\alpha_2z^{-1} \quad \dots (16)$$

$$k_2 = -\alpha_0z - \alpha_1 + \alpha_0\alpha_1\alpha_2 - \alpha_2z^{-1} \quad \dots (17)$$

$$l_1 = \alpha_1 + \alpha_0z^{-1} + \alpha_2z - \alpha_0\alpha_1\alpha_2 \quad \dots (18)$$

$$l_2 = -\alpha_0\alpha_1 + z^{-1} - \alpha_0\alpha_2z - \alpha_1\alpha_2 \quad \dots (19)$$

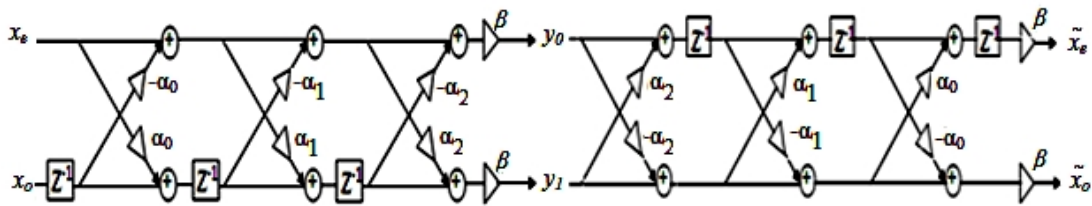


Fig.5 The analysis and synthesis sides of the Daubechies-6 lattice filter bank.

It can be noted that two advance elements are considered to be at the end of both the upper and lower branches of the lattice structure of this filter bank for the comparison purpose. These advance elements can be neglected during the implementation as they have similar effect on both branches. Thus the lattice coefficients are $\alpha_0 = -2.425494$, $\alpha_1 = 0.546095158$, $\alpha_2 = -0.105889275$, and $\beta = 0.332671$.

3.4 Daubechies-8 filter bank lattice design

For the Daubechies-8 filter bank, the LPF and HPF equations are [14]

$$H_0(z) = h_0 + h_1z^{-1} + h_2z^{-2} + h_3z^{-3} + h_4z^{-4} + h_5z^{-5} + h_6z^{-6} + h_7z^{-7} \dots (20)$$

$$G_0(z) = -h_7 + h_6z^{-1} - h_5z^{-2} + h_4z^{-3} - h_3z^{-4} + h_2z^{-5} - h_1z^{-6} + h_0z^{-7} \dots (21)$$

where

- $h_0 \approx 0.230377813309$, $h_1 \approx 0.714846570553$,
- $h_2 \approx 0.630880767940$, $h_3 \approx -0.027983769417$,
- $h_4 \approx -0.187034811719$, $h_5 \approx 0.030841381836$,
- $h_6 \approx 0.032883011667$, $h_7 \approx -0.010597401785$.

After separating $H_0(z)$ and $G_0(z)$ into even and odd parts, the polyphase matrix will be as follows:

$$H_p(z) = \begin{bmatrix} h_0 + h_2z^{-1} + h_4z^{-2} + h_6z^{-3} & h_1 + h_3z^{-1} + h_5z^{-2} + h_7z^{-3} \\ -h_7 - h_5z^{-1} - h_3z^{-2} - h_1z^{-3} & h_6 + h_4z^{-1} + h_2z^{-2} + h_0z^{-3} \end{bmatrix} \dots (22)$$

Using Fig. 6, the corresponding matrix of the lattice is given by

$$H_l(z) = \begin{bmatrix} k_1\beta & k_2\beta \\ l_1\beta & -l_2\beta \end{bmatrix} \dots (23)$$

where

$$k_1 = 1 - \alpha_0\alpha_1z^{-1} - \alpha_1\alpha_2z^{-1} - \alpha_0\alpha_2z^{-2} - \alpha_1\alpha_3z^{-2} - \alpha_0\alpha_3z^{-3} - \alpha_2\alpha_3z^{-1} + \alpha_0\alpha_1\alpha_2\alpha_3z^{-2} \dots (24)$$

$$k_2 = -\alpha_0 - \alpha_1z^{-1} + \alpha_0\alpha_1\alpha_2z^{-1} - \alpha_2z^{-2} + \alpha_0\alpha_1\alpha_3z^{-2} - \alpha_3z^{-3} + \alpha_0\alpha_2\alpha_3z^{-1} + \alpha_1\alpha_2\alpha_3z^{-2} \dots (25)$$

$$l_1 = \alpha_3 - \alpha_0\alpha_1\alpha_3z^{-1} - \alpha_1\alpha_2\alpha_3z^{-1} - \alpha_0\alpha_2\alpha_3z^{-2} + \alpha_1z^{-2} + \alpha_0z^{-3} + \alpha_2z^{-1} - \alpha_0\alpha_1\alpha_2z^{-2} \dots (26)$$

$$l_2 = -\alpha_0\alpha_3 - \alpha_1\alpha_3z^{-1} + \alpha_0\alpha_1\alpha_2\alpha_3z^{-1} - \alpha_2\alpha_3z^{-2} - \alpha_0\alpha_1z^{-2} + z^{-3} - \alpha_0\alpha_2z^{-1} - \alpha_1\alpha_2z^{-2} \dots (27)$$

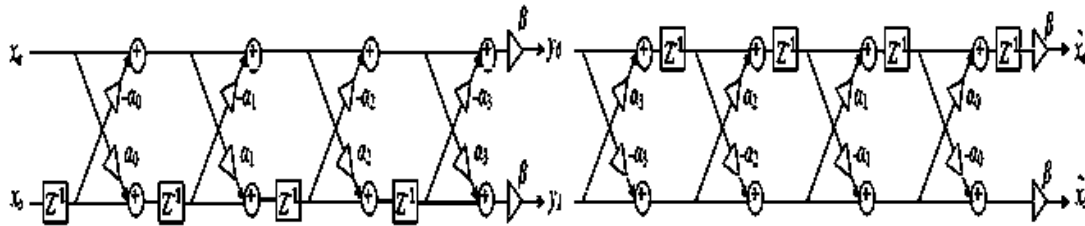


Fig.6 The analysis and synthesis sides of the Daubechies-8 lattice filter bank.

And the values of the lattice coefficients will be as follows: $\alpha_0 = -3.102931488$, $\alpha_1 = 0.810257495$, $\alpha_2 = -0.261939878$, $\alpha_3 = 0.046$, and $\beta = 0.230377813$.

It can be noted that, the signs of the lattice multiplier coefficients alternate between stages, and that, the values of the multiplier coefficients α_i decrease with the increase of i .

When comparing between the analysis and synthesis sides, it can also be noted that the delay element is removed from the beginning of the lower branch to the end of the upper branch of each stage, the coefficient α_i remains in its place on the two sub-branches but with reversed signs and the first analysis stage becomes the last synthesis one. This is due to the time reversal of the analysis side needed to be performed in the synthesis side in order to satisfy the perfect reconstruction condition.

4. Matlab Representation

The filter bank designs are described using Matlab programming for verification. The results show that these structures can be achieved with perfect reconstruction when performing the analysis-synthesis operations using the lattice structures. That is the difference between the original and reconstructed signals is zero. Matlab programs are used to find the minimum wordlength of the lattice coefficients for acceptable PSNR values which is considered to be about 30dB. This wordlength is found to be 6 bits as indicated by the PSNR readings taken for the designed filter banks using a group of gray scale test images, as shown in Fig.7.

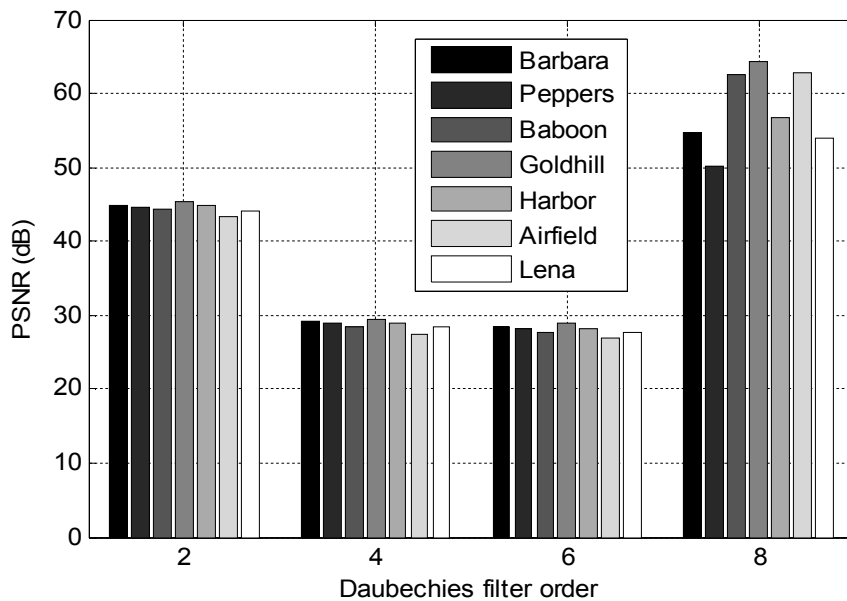


Fig.7 PSNR values versus the filter order for a group of gray scale images using the word length of 6 bits.

The processing of two-Dimensional (2-D) signals is accomplished by a one-Dimensional (1-D) interdealing. Each image is first translated from a 2-D to 1-D signal by scanning it row by row and putting the result in a vector form. Such vector is applied at the input of the proposed wavelet structure for analysis and reconstruction. The result of the reconstruction side is then re-transformed from a 1-D to 2-D signal that represents the result. The original, reconstructed, approximation and details of some of the standard gray scale test images for the Daubechies-6 and 8 filter banks are shown in Figs. 8 - 11 and Figs. 12 - 15, respectively.



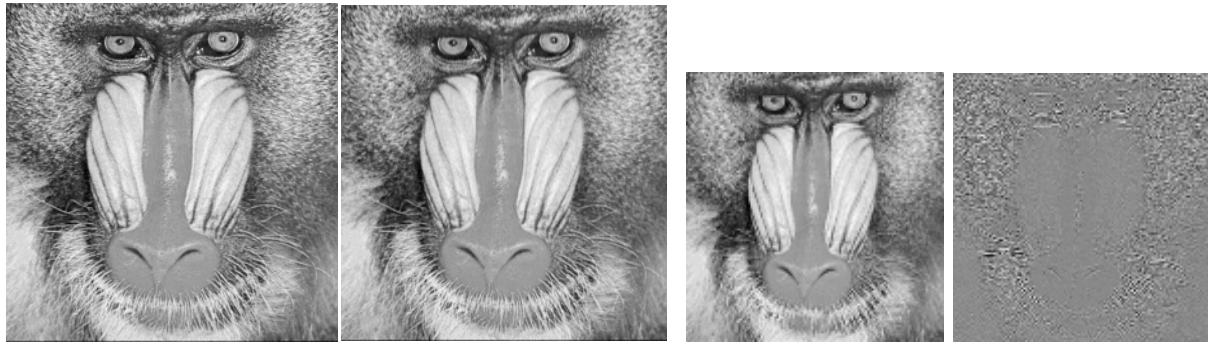
Fig. 8 The analysis and synthesis of Lena image using Daubechies-6 filter bank.



Fig. 9 The analysis and synthesis of Goldhill image using Daubechies-6 filter bank.



Fig. 10 The analysis and synthesis of Harbor image using Daubechies-6 filter bank.



Original image *Reconstructed image* *Approximation* *Details*
Fig. 11 The analysis and synthesis of Baboon image using Daubechies-6 filter bank.



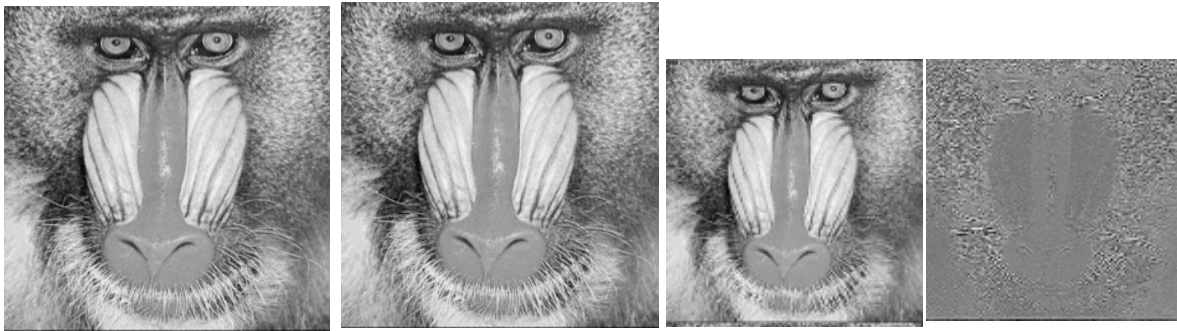
Original image *Reconstructed image* *Approximation* *Details*
Fig. 12 The analysis and synthesis of Lena image using Daubechies-8 filter bank.



Original image *Reconstructed image* *Approximation* *Details*
Fig. 13 The analysis and synthesis of Goldhill image using Daubechies-8 filter bank.



Original image *Reconstructed image* *Approximation* *Details*
Fig. 14 The analysis and synthesis of Harbor image using Daubechies-8 filter bank.



Original image *Reconstructed image* *Approximation* *Details*
 Fig. 15 The analysis and synthesis of Baboon image using Daubechies-8 filter bank.

5. FPGA Implementations

Two techniques are used to perform the DWT implementation using the designed lattice structures. The first is the pipelining, causing the architecture to have a throughput of two output samples every clock cycle, while the second is the bit-serial implementation technique that minimizes the area allocation by using a single processing element to perform the whole task, bearing in mind the increase of the processing time as a disadvantage of the latter. The filter bank architectures are implemented using one of the Xilinx FPGA devices, the SPARTAN-3E kit. This device has a capacity of (4656) logic slices where the total number of CLBs is (1164) (each CLB contains four slices). It has 20 dedicated multipliers, 20 BRAMs and can operate at a maximum clock speed of 50MHz.

5.1 Pipelining implementation results

Pipelining is a technique of decomposing a sequential process into suboperations, with each subprocess being executed in a special dedicated segment that operates concurrently with all other segments. The technique is efficient for those applications that need to repeat the same task many times with different sets of data [15]. Therefore, the pipeline technique is used to implement the lattice structure that consists of a sequence of similar PEs, in order to increase the amount of accomplished processing during a given time interval.

The pipelining implementation requires two inputs every clock cycle, gives a throughput of two samples per clock cycle (after having the first input processed) and, as the filter order increases, the cost of implementing also increases. This implementation of the lattice structure that have two output branches provides us with the ability of giving two output samples every clock cycle, which is a very important property for real time applications.

The results of implementation using the SPARTAN-3E device are summarized in Table 1 for the two cases of using the dedicated multipliers and the shift-add operations to perform multiplication between the lattice coefficients and the data.

From Table 1, it can be noted that the use of the shift-add operations for multiplication increases the number of exploited slices with making no use of the dedicated multipliers available in the device.

Table 1 Pipelining area results.

Filter Bank type	Resource	Using Multipliers			Using shift-add operations		
		exploited	total	utilization ratio	exploited	total	utilization ratio
Daub.2	slices	62	4656	1%	121	4656	2%
	slice flip flops	31	9312	0%	54	9312	0%
	4-input LUT	96	9312	1%	226	9312	2%
	multipliers	4	20	20%	0	20	0%
Daub.4	slices	115	4656	2%	358	4656	7%
	slice flip flops	106	9312	1%	149	9312	1%
	4-input LUT	196	9312	2%	690	9312	7%
	multipliers	12	20	60%	0	20	0%
Daub.6	slices	164	4656	3%	383	4656	8%
	slice flip flops	166	9312	1%	196	9312	2%
	4-input LUT	266	9312	2%	720	9312	7%
	multipliers	16	20	80%	0	20	0%
Daub.8	slices	231	4656	4%	454	4656	9%
	slice flip flops	238	9312	2%	255	9312	2%
	4-input LUT	266	9312	4%	814	9312	8%
	multipliers	16	20	80%	0	20	0%

5.2 Bit-serial implementation results

When implementing in a bit-serial manner, a single processing element is exploited to perform the whole task, each two input samples are folded through this processing element to be processed for number of times equal to the number of filter structure stages. The cost is lower as compared with the pipelining implementation, but as the filter order increases, the time needed to process a pair of input samples also increases.

The results of implementing using the SPARTAN-3E device are shown in Table 2 in the two cases of using the dedicated multipliers and the shift-add operations.

Table 2 bit-serial area results.

Filter Bank type	Resource	Using Multipliers			Using shift-add operations		
		exploited	total	utilization ratio	exploited	total	utilization ratio
Daub.4	slices	150	4656	3%	400	4656	8%
	slice flip flops	184	9312	1%	365	9312	3%
	4-input LUT	237	9312	2%	742	9312	7%
	multipliers	8	20	40%	0	20	0%
Daub.6	slices	160	4656	3%	447	4656	9%
	slice flip flops	188	9312	2%	365	9312	3%
	4-input LUT	264	9312	2%	827	9312	8%
	multipliers	8	20	40%	0	20	0%
Daub.8	slices	176	4656	3%	540	4656	11%
	slice flip flops	213	9312	2%	422	9312	4%
	4-input LUT	277	9312	2%	984	9312	10%
	multipliers	8	20	40%	0	20	0%

6. A Comparative Study

A comparison is made with other implementations for two of the four Daubechies filter banks using other wavelet structures in previous researches given in Refs. [5], [9] and [16] for the Daubechies-4 filter bank and given in Refs. [9] for the Daubechies-8 filter bank. The comparison indicates that smaller area can be achieved using the implemented lattice structures, and in addition they have higher circuit frequency as compared with these researches. Table 3 shows the comparison with these references using the same devices, while Table 4 shows the comparison of the implementation of Daubechies-4 filter bank using SPARTAN3E device as compared with other implementations using ALTERA devices.

Table 3 Comparison with other implementations of the Daubechies-4 and 8 filter banks using the same devices.

Filter bank type	Device	Structure	Research results			The proposed		
			No. of slices	No. of Mul.	Freq. (MHz)	No. of slices	No. of Mul.	Freq. (MHz)
Daub.4	VIRTEX V100	in [9]	488	0	18.00	427	0	42.15
Daub.4	SPARTAN2 XC2S200	in [16]	584	0	28.56	416	0	36.79
Daub.8	VIRTEX V100	in [9]	837	0	14.80	489	0	44.86

Table 4 Comparison with other implementations of the Daubechies-4 filter bank using ALTREA devices.

Structure	No. of slices	No. of multipliers	frequency (MHz)	exploited device
in [5]	404	0	40.16	ALTERA EPF10K30
in [16]	404	0	30.60	ALTERA Cyclone II board
The proposed	358	0	47.09	Xilinx spartan3E XC3S500E

7. Conclusions

The lattice structures for DWT have been designed and tested through the implementation of the Daubechies filter banks. The results have shown that they are very appropriate DWT structures from area allocation point of view. The filter banks have been implemented and the results have been tested for different FPGA devices, setting the optimization goal of the synthesizer to be the area. The results of implementing the designed filter banks using the SPARTAN-3E FPGA kit highlight an efficient implementation with minimum hardware utilization, having the area allocation

being minimized to 0.584 - 0.712 of the corresponding recent implementations using the same filter banks.

From the comparative tables in the previous section, it can be noticed that, the pipelining implementation of the lattice structure is very efficient and is suitable for high speed applications (minimum delay = 0.02 μ secs.). But, when area is the constraint, as the filter order increases, the most suitable lattice implementation scheme becomes the bit-serial implementation, bearing in mind the increase of the processing time for a specified task is a disadvantage of this implementation scheme.

References

- [1] S. Tsai, "Power Transformer Partial Discharge (PD) Acoustic Signal Detection using Fiber Sensors and Wavelet Analysis, Modeling, and Simulation", Master's Thesis, Electrical and Computer Engineering, The Virginia Polytechnic Institute and State University, December 2002.
- [2] M. Estes, "The Discrete Wavelet Transform". ECE 402–Digital Signal Processing, Spring 2001.
- [3] K. Kotteri, "Optimal, Multiplier less Implementations of the Discrete Wavelet Transform for Image Compression Applications", Master's Thesis, Electrical Engineering Department, Virginia Polytechnic Institute and State University, April 2004.
- [4] M. Joshi, R. Manthalkar and Y. Joshi, "Image Compression Using Curvelet, Ridgelet and Wavelet Transform, A Comparative Study", The International Congress for Global Science and Technology (ICGST), International Journal on Graphics, Vision and Image Processing (GVIP), ISSN 1687-398X, Vol. 8, Issue 3, October 2008.
- [5] D. Sripathi, "Efficient Implementations of Discrete Wavelet Transforms Using FPGAs", Master's Thesis, Electrical and Computer Engineering Department, Florida State University, November 2003.
- [6] R. Dubey, "Introduction to Embedded System Design Using Field Programmable Gate Arrays", Springer-Verlag London Limited, ISBN 978-1-84882-015-9, 2009.
- [7] J. Deschamps, G. Bioul and G. Sutter, "Synthesis of Arithmetic Circuits: FPGA, ASIC, and Embedded Systems", John Wiley & Sons, Inc., Hoboken, New Jersey, ISBN 13-978-0471-68783-2, 2006.
- [8] M. Vesterbacku, K. Palmkvist and L. Wanhammar, "Maximally Fast, Bit-Serial Lattice Wave digital Filters", Digital Signal Processing Workshop Proceedings, IEEE, pp. 207-210, September 1996.
- [9] V. Herrero, J. Cerda, R. Gadea, M. Peiro and A. Sebastia, "Implementation of 1-D Daubechies Wavelet Transform on FPGA", Group of Design of Digital Systems, Electronics Engineering Department, Valencia University, 2000. From internet on: www.upv.es/dsd/publica/articulos/paper192.pdf
- [10] J. Olkkonen and H. Olkkonen, "Discrete Lattice Wavelet Transform", IEEE Transactions on Circuits and Systems-II: Express Briefs, Vol. 54, No. 1, January 2007.

- [11] M. Mahmoud, M. Dessouky, S. Deyab and F. Elfouly, "Comparison between Haar and Daubechies Wavelet Transformations on FPGA Technology", Proceedings of World Academy of Science, Engineering and Technology, Vol. 20, ISSN 1307-6884, April 2007.
- [12] T. Chiang, "Design and Performance Evaluation of a Discrete Wavelet Transform-absed Multi-Signal Receiver", School of Graduate Studies, Wright State University, July 2006. From internet on: www.etsd.ohiolink.edu/send-pdf.cgi/Chiang,%20Tony.pdf
- [13] T. Acharya and P. Tsai, "JPEG2000 Standard for Image Compression", John Wiley & Sons, Inc., Hoboken, New Jersey, ISBN 0-07-252261-5, 2005.
- [14] P. Getreuer, "Filter Coefficients to Popular Wavelets", May 2006. From internet on: www.mathworks.com/matlabcentral/files/5502/popwav.pdf
- [15] M. Mano, "Computer System Architecture", Prentice-Hall, Inc., A Simon & Schuster Company, Englewood Cliffs, New Jersey, 07632, ISBN 0-13-175738-5, 1993.
- [16] U. Meyer-Baese, A. Vera, A. Meyer-Baese, M. Pattichis and R. Perry, "Discrete Wavelet Transform FPGA Design using MatLab/Simulink", University of New Mexico, ECE Department, Albuquerque, NM 87131, 2006. From internet on: www.ivpcl.org/2006_DiscreteWavelet.pdf

The work was carried out at the college of Engineering. University of Mosul