

**DESIGN AND IMPLEMENTATION OF LIFTING BASED DAUBECHIES
WAVELET TRANSFORMS USING ALGEBRAIC INTEGERS**

A Thesis Submitted to the College of

Graduate Studies and Research

In Partial Fulfillment of the Requirements

For the Degree of Master of Science

In the Department of Electrical and Computer Engineering

University of Saskatchewan

Saskatoon

By

Pranav Balakrishnan

© Copyright Pranav Balakrishnan, April, 2013. All rights reserved

PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Electrical and Computer Engineering
57 Campus Drive
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5A9

ABSTRACT

Over the past few decades, the demand for digital information has increased drastically. This enormous demand poses serious difficulties on the storage and transmission bandwidth of the current technologies. One possible solution to overcome this approach is to compress the amount of information by discarding all the redundancies. In multimedia technology, various lossy compression techniques are used to compress the raw image data to facilitate storage and to fit the transmission bandwidth.

In this thesis, we propose a new approach using algebraic integers to reduce the complexity of the Daubechies-4 (D4) and Daubechies-6 (D6) Lifting based Discrete Wavelet Transforms. The resulting architecture is completely integer based, which is free from the round-off error that is caused in floating point calculations. The filter coefficients of the two transforms of Daubechies family are individually converted to integers by multiplying it with value of 2^x , where, x is a random value selected at a point where the quantity of losses is negligible. The wavelet coefficients are then quantized using the proposed iterative individual-subband coding algorithm. The proposed coding algorithm is adopted from the well-known Embedded Zerotree Wavelet (EZW) coding. The results obtained from simulation shows that the proposed coding algorithm proves to be much faster than its predecessor, and at the same time, produces good Peak Signal to Noise Ratio (PSNR) at very low bit rates.

Finally, the two proposed transform architectures are implemented on Virtex-E Field Programmable Gate Array (FPGA) to test the hardware cost (in terms of multipliers, adders and registers) and throughput rate. From the synthesis results, we see that the proposed algorithm has low hardware cost and a high throughput rate.

ACKNOWLEDGEMENTS

First, I would like to express my gratitude to my supervisor, Dr. Khan A. Wahid for introducing me to this research field and for his continuous support, inspiration, and guidance throughout the project. His vast knowledge and expertise in this field added considerably to my graduate experience.

Also, I would like to express my profound gratitude to the professors of our department for their knowledge and support in the successful completion of my course work.

Then, I would like to thank my thesis committee for their guidance and insightful comments.

Finally, I would like to thank my family and friends for their love, encouragement, and support.

TABLE OF CONTENTS

PERMISSION TO USE.....	i
ABSTRACT	ii
ACKNOWLEDGEMENT.....	iii
TABLE OF CONTENTS.....	iv
LIST OF FIGURES.....	vii
LIST OF TABLES.....	vi
LIST OF ABBREVIATIONS.....	viii
CHAPTER 1 INTRODUCTION	
1.1 Image Compression.....	1
1.2 Thesis Motivation.....	3
1.3 Thesis Objective.....	4
1.4 Thesis Organization.....	5
CHAPTER 2 DISCRETE WAVELET TRANSFORM	
2.1 Introduction.....	7
2.2 What are Wavelets?	8
2.3 Multi-Resolution Analysis and Subband Coding.....	10
2.4 Conditions for Perfect Reconstruction.....	13
CHAPTER 3 LIFTING BASED DISCRETE WAVELET TRANSFORM	
3.1 Introduction.....	15
3.2 Constructing Second Generation Wavelets.....	16
3.3 Applying Lifting Based Transform to Images.....	19
3.4 Daubechies-4 (D4) Lifting Wavelet Transform.....	22

3.5	Daubechies-6 (D6) Lifting Wavelet Transform.....	24
3.6	Previous Works.....	26
CHAPTER 4 PROPOSED LIFTING-BASED ALGORITHM		
4.1	Introduction.....	29
4.2	Algebraic Integer Quantization (AIQ).....	30
4.3	AIQ Based Daubechies-4 Lifting Scheme.....	32
4.4	AIQ Based Daubechies-6 Lifting Scheme.....	35
CHAPTER 5 PROPOSED QUANTIZATION SCHEME		
5.1	Introduction.....	39
5.2	Quantization in Wavelet Transforms.....	39
5.3	Embedded Coding.....	40
5.4	Proposed Quantization by Adaptive-Iterative Coding.....	44
5.5	Run-Length Encoding (RLE).....	46
CHAPTER 6 EXPERIMENTAL RESULTS AND DISCUSSION		
6.1	Introduction.....	47
6.2	Analysis of Image Using Wavelet Transform.....	47
6.3	Simulation Results.....	50
6.4	FPGA Synthesis Results.....	59
CHAPTER 7 CONCLUSION AND FUTURE WORK		
7.1	Summary and Accomplishments.....	64
7.2	Recommendations for Future Work.....	65
REFERENCES.....		67

LIST OF FIGURES

Figure 2.1: Wavelet functions of some of the popular wavelets.	9
Figure 2.2: Three Level Wavelet Decomposition Tree.....	11
Figure 2.2: Three Level Wavelet Reconstruction Tree.....	12
Figure 2.3: Three Level 2-D DWT Decomposition of an Image.....	13
Figure 3.1: a) and b) are the Analysis and Synthesis Representation of the Polyphase Matrix.	18
Figure 3.2: Basic Steps in Lifting Scheme.....	19
Figure 3.3: Inverse Lifting Transform.....	20
Figure 3.4: Lifting Based Daubechies-4 Forward Transform Block.....	23
Figure 3.5: Lifting Based Daubechies-6 Forward Transform Block.....	26
Figure 4.1: AIQ Model of the Lifting Based D4 DWT.....	33
Figure 4.2: Average PSNRs Plotted Against Values of τ and μ for the Image ‘Lena’.....	34
Figure 4.3: Error added using the proposed integer based D4 architecture.....	35
Figure 4.4: AIQ Model of the Lifting Based D6 DWT.....	36
Figure 4.5: Error added using the proposed integer based D6 architecture.....	38
Figure 5.1: General diagram of an image processing algorithm	39
Figure 5.2: Wavelet Coefficients Relationship and the Scanning Order.....	41
Figure 6.1: Analysis of Decomposed Image at Different Levels.....	49
Figure 6.2: Standard Images.....	51
Figure 6.3: CT Images.....	51
Figure 6.4: Endoscopic Images.....	52
Figure 6.5: Comparison of image ‘Lena’ at different bitrates and their corresponding PSNRs....	55
Figure 6.6: Analysis of the Image Using Replicate Function	58
Figure 6.7: Output Waveform.....	60

LIST OF TABLES

Table 5.1: System Configuration.....	45
Table 5.2: Performance Comparison of The Proposed Coding Algorithm.....	46
Table 6.1: Energy Compaction of the Sub Bands at Various Decomposition Levels.....	48
Table 6.2: PSNR Results for Standard and CT Images.....	54
Table 6.3: Resulting PSNR for Endoscopic Images.....	56
Table 6.4: Performance of the Proposed Quantization Calculated for the Image ‘Lena’	57
Table 6.5: Hardware Comparison of the Proposed AIQ Architecture with Other Techniques.....	61
Table 6.6: Comparison of Logic Utilization of the Proposed Algorithm on Hardware.....	62
Table 6.7: Comparison between Two Proposed Architectures.....	63

LIST OF ABBREVIATIONS

AIQ	Algebraic Integer Quantization
bpp	bits per pixel
CDF	Cohen-Daubechies-Feauveau
D4	Daubechies-4 Wavelet Transform
D6	Daubechies-6 Wavelet Transform
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
ECG	Electrocardiography
EZW	Embedded Zerotree Wavelet
FPGA	Field Programmable Gate Array
HDL	Hardware Description Language
HVS	Human Visual System
IWT	Integer Wavelet Transform
IZ	Isolated Zero
JPEG	Joint Photographic Experts Group
LWT	Lifting based Discrete Wavelet Transform
PSNR	Peak Signal to Noise Ratio
RLE	Run-Length Encoding
SPIHT	Set Partitioning in Image Hierarchical Trees
UWB	Ultra-Wide Band
VLSI	Very Large Scale Integration
ZRT	Zero Tree Root

Chapter 1

Introduction

1.1 Image Compression

With the advanced development in Internet and multimedia technologies, the amount of information that is handled by computers has grown exponentially over the past decades. An image is such information represented as a positive function on a plane. The value of this function at each point specifies the luminance or brightness of the picture at that point. These values are known as pixels. The value of the luminance at each pixel is represented to a pre-defined precision. Eight bits of precision for luminance is common in imaging applications. The eight-bit precision is motivated by both the existing computer memory structures (1 byte = 8 bits) as well as the dynamic range of the human eye. So if we consider a grayscale image of size 512×512 , the pixels vary from 0 to 255 levels of luminance. The canonical representation requires $512 \times 512 \times 8 = 2,097,152$ bits. This shows that, in order to store one grayscale image of size 512×512 , we require 256MB of memory. Larger number of such image requires huge amount of storage space and transmission bandwidth that the current technology is unable to handle technically and economically. One of the possible solutions to this problem is to compress the information so that the storage space and transmission time can be reduced. This is the main function of image compression.

A typical image compressor is comprised of transform, quantization and coding blocks. The transform is used to represent the image pixels into fewer coefficients without any loss of information. The decomposed coefficients are then compressed using quantization. The

quantization uses set of predefined steps to remove any redundant information from the image. At this stage the compression becomes irreversible. The coding stage converts the compressed coefficients into binary values and adds it to the bitstream for the ease of transmission. This method of compression is called lossy compression. The decompressor is used to perform the inverse operations of the encoder to get the original image.

One such transform is the wavelet transform [1, 2]. The Discrete Wavelet Transform (DWT) [3, 4], with its multiresolution capability, is widely used in many applications like, image and video coding, biomedical signal processing for low-power pacemakers, ultra-wideband (UWB) wireless communication pattern recognition, etc. There are many transforms that are included in the Discrete Wavelet transform. Out of the many transforms, a widely used variation is the Daubechies wavelets (D2-D20) [23]. This thesis is limited to only two of the Daubechies wavelets (D4 and D6). This is because the D4 and D6 transforms offer good decomposition when compared to simple D2, and also the complexity of the transform architecture increases from D8. Due to the few problems with the conventional Discrete Wavelet transform (such as; it uses large number of auxiliary memory and since it is Fourier based, it cannot be applied for many complex real life applications), we use the lifting based Daubechies-4 (D4) and Daubechies-6 (D6) DWT. The lifting based architecture is primarily designed for the purpose of implementation. The lifting scheme is much faster than the traditional approach, less complex requirements for hardware, and is more suitable for real-time processing. Due to these beneficial features and advantages, the lifting scheme (CDF 9/7 transform; with four simple filter coefficients offer flexibility and practically very low computational error) is included as the core transform in the new JPEG2000 standard [9].

1.2 Thesis Motivation

With the development in nanotechnology, hardware size has decreased significantly and is packed with many functions. Due to the small size of the devices, the storage capacity and resources are limited. It is necessary to compress all the information before storing or transmitting the data. It would be ironic if the encoder takes most of the resources on the hardware. Hence, there is always a demand for efficient low complexity algorithms for image processing.

The use of conventional floating point lifting based algorithm introduces error which degrades the quality of the image when reconstructed. This is due to the inability to exactly represent the filter coefficient values during implementation. In order to eliminate the round-off error for DWT, Wahid et al. [26, 47] proposed a new integer based mapping technique, known as Algebraic Integer Quantization (AIQ), to compute the Daubechies-4 tap and Daubechies-6 tap filter coefficients. The AIQ based approach converts all the filter coefficients into integers and provides error free calculation of the wavelet coefficients. However, the former AIQ based technique is based on the conventional DWT which has a low throughput rate and high hardware complexity.

In our work, we extend the AIQ algorithm for the lifting based Daubechies-4 and Daubechies-6 DWT. We reduce the number of filter coefficients and implement complete integer based architecture. The algorithm uses very little hardware resources and can be used for real-time processing on smaller devices. In addition to this, we propose a new adaptive subband quantization approach to code the wavelet coefficients. Most available quantization algorithms do not provide efficient compression of the original image, whereas, algorithms that offer good

quality compression are targeted towards specific types of images. On the other hand, algorithms that perform both the previous functions tend to be highly complex. Embedded Zerotree Wavelet (EZW) [29] is one of the popular wavelet image coding algorithms that offers good reconstructed image quality at a very low bit rate, and handles all types of images. Despite the advantages, the EZW algorithm is very complex and has very slow processing time. Our proposed coding algorithm performs on individual subbands using iterative coding structure. This method of individual subband coding is much faster with good quality image compression. The output obtained in our case is binary values which can be directly transmitted without any extra conversion steps.

1.3 Thesis Objective

This thesis is directed towards computing efficiently the D4 and D6 forward wavelet transform filters using AIQ based lifting scheme. The objectives of this research can be categorized as follows,

- 1) The integer based architecture of the two lifting based transforms should be less complex and the quality of reconstructed image should be similar to the floating point transform.

The quality will be measured in terms of Peak Signal to Noise Ratio (PSNR) as given by,

$$PSNR = 20 \times \log_{10} \frac{255}{\sqrt{\frac{1}{M \times N} \sum_{n=1}^N \sum_{m=1}^M (x_{m,n} - x'_{m,n})^2}} \quad (1.1)$$

- 2) The proposed iterative coding algorithm should be much faster than EZW while maintaining good quality reconstructed image at high compression rates.

- 3) The two proposed transform algorithms should use very little resources when implemented on Field Programmable Gate Array (FPGA).
- 4) To discuss the performance of our algorithm with various other techniques.

All the details of the implementation of the two algorithms will be discussed in this thesis. All the results from simulation and hardware synthesis will be discussed in terms of image reconstruction and hardware complexity along with the architectures from many other existing techniques.

1.4 Thesis Organization

This thesis consists of six chapters. In Chapter 2 we discuss briefly about the Discrete Wavelet Transform. Chapter 2 also shows the most important features of using Discrete Wavelet Transform for image decomposition. In Chapter 3 we illustrate the disadvantages of using the Discrete Wavelet Transform and the use of Lifting based Discrete Wavelet Transform. We also discuss the advantages of using the Lifting scheme over the traditional wavelets. In Chapter 4 we discuss in detail about the proposed Algebraic Integer Quantization (AIQ) approach, used to improve the lifting scheme. We also discuss the reduced complexity of the Daubechies-6 lifting scheme. In Chapter 5, we propose a new quantization technique to code the wavelet coefficients. We also compare the processing time of the proposed algorithm with other famous coding techniques. In the first half of Chapter 6, we discuss performance of the two proposed algorithm using the simulation results. The PSNR and bit rate are determined for both encoders using several types of images. In the second half of Chapter 6, we discuss the complexity of the two

proposed transforms on hardware. Finally, in Chapter 7 we summarize the entire thesis work and provide recommendations for possible improvements of future work.

Chapter 2

Discrete Wavelet Transform

2.1 Introduction

Discrete Wavelet Transform (DWT) [1][2][3][4] is a technique used in image processing for compressing data. Image compression is essential to provide ease of transmission and storage. The discrete wavelet transform has been extensively used by many researchers in recent years [5][6] due to its impressive structure and time-frequency characteristics. The DWT transforms discrete signals from time domain to time frequency domain. Due to many interesting features, the discrete wavelet transform is being used in many practical applications like speech compression, which provides faster transmission in mobile communication. DWT is used in medical applications for its real-time processing capabilities. Also, in denoising, feature extraction, edge detection, echo cancellation, etc.

On the other hand, many researchers have used the Discrete Cosine Transform (DCT) [7] as the primary transform scheme for many years and it is included in the well known JPEG [8] standard. But at high compression rates the DCT produces “blocking artifacts”. The input image is split into blocks (8 x 8 or 16 x 16 etc) and DCT is performed on each of the blocks. At high compression rates, a mismatch occurs between the adjacent blocks near the boundaries. This mismatch causes blocks to appear in the reconstructed image. However, the DWT performs on an entire image in row by row fashion, thus it eliminates the possibility of producing blocking artifacts. Moreover, DCT requires huge number of memory buffers for storing the input blocks and for its transform matrix. Efficiency of the DCT relies on choosing the size of the blocks and

its transform matrix. DWT on the other hand works with minimum number of registers, since the decomposition is carried out in row by row fashion. This makes DWT less complex and more suitable for hardware implementation. Due to these compelling factors, the Discrete Wavelet Transform is integrated in the most well known standards like JPEG2000 [9], MPEG-4, etc.

2.2 What are Wavelets?

Wavelets [10] are sets of basis functions used in the analysis of signals and images. For many decades, scientists desired to use a function, more appropriate than the sine and cosine signal, to represent choppy signals. Wavelets are functions that are created as a superposition of some set of functions mainly used for approximating data. Wavelets are well suited for approximating data with sharp discontinuities. A wavelet is generally a portion of a complete waveform, and it decays quickly. The wavelet analysis uses a wavelet prototype function called the “analyzing wavelet” or “mother wavelet”. Temporal analysis is resolved using the high frequency version of the prototype wavelet, while the frequency analysis is better resolved using the low frequency version of the prototype wavelet. Hence the concept of wavelets is to look at the input data at various scales and analyze them in different resolutions.

The concept of wavelet transform was first introduced by Jean Morlet in 1982 [11]. Morlet provided mathematical tool and considered that family of functions are constructed from a single function known as “mother wavelet” $\psi(t)$. These functions are given by [12],

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right), \quad a, b \in R, a \neq 0 \quad (2.1)$$

Figure 2.1 shows the wavelet functions of some of the popular wavelets used in image processing. All the figures are generated using MATLAB image processing tool.

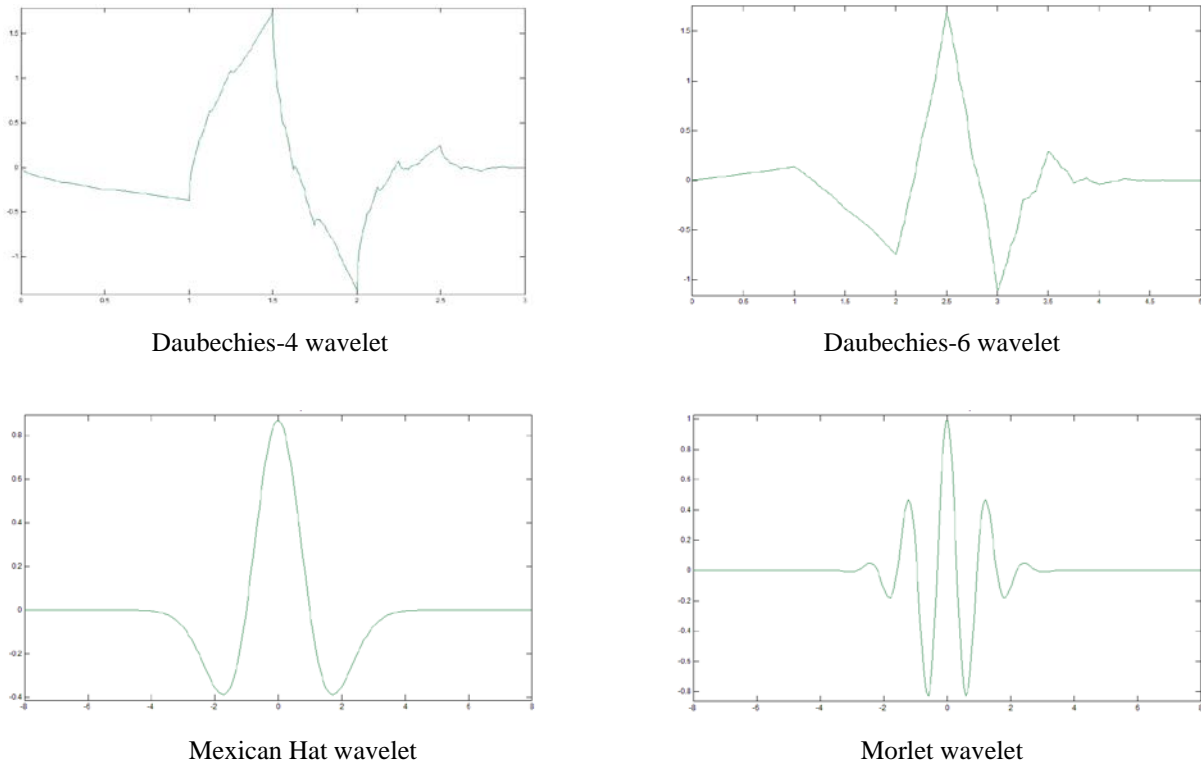


Figure 2.1: Wavelet functions of some of the popular wavelets.

The parameter a represents dilation (scaling) and it corresponds to the degree of compression. The parameter b represents translation which determines the time location of the wavelets. If $|a| > 1$, then the output is a compressed version of the mother wavelet and it represents the high frequency components. On the contrary, if $|a| < 1$ then $\psi_{a,b}(t)$ has larger time-widths than $\psi(t)$ and it corresponds to the low frequency components. Thus, wavelets have their time-widths related to their frequencies. Also, the wavelet transform at high frequencies gives good time resolution but poor frequency resolution, while, at low frequencies, it gives good frequency resolution but poor time resolution. In other words, high frequency components do not appear for

a long duration, whereas, the low frequencies last for the entire duration of the signal. This time-frequency characteristic makes the wavelets an excellent tool in the field of image processing.

2.3 Multi-Resolution Analysis and Subband Coding

In the Discrete Wavelet Transform, image data can be analyzed using an analysis filter bank followed by a decimation operation. In image compression, it is most common to use an analysis filter bank with a set of low pass and high pass filters at each stage. These filters are designed to adapt certain characteristics of data well suited for some applications. The decimation operation is most commonly known as subsampling [13]. Subsampling refers to removing some samples of the signal. For example, subsampling by two refers to dropping every other sample of the signal. Subsampling does not change the resolution of the signal. Resolution is a measure of detail information in the signal and is affected by the filtering operations.

DWT analyses the input signal at different frequencies with different resolutions by decomposing the signal into approximation and detail information. The DWT uses two set of function called wavelet function and scaling function. The decomposition of the signal into different frequencies is obtained through successive low pass and high pass filtering of the time domain signal. The original input signal $x[n]$ is passed through a low pass filter $h[n]$ and a high pass filter $g[n]$. The signal now has a frequency of $\pi/2$ radians instead of π . Hence, according to the Nyquist's rule, half of the samples can be discarded without any loss in the information. The signal can be therefore subsampled by 2, removing half of the redundant samples. This in turn doubles the scale. This constitutes a one level decomposition and can be mathematically expressed as,

$$y_{high}(k) = \sum_n^{\infty} x[n].g[2k - n] \quad (2.2)$$

$$y_{low}(k) = \sum_n^{\infty} x[n].h[2k - n] \quad (2.3)$$

where n is an integer, $y_{high}(k)$ and $y_{low}(k)$ are the outputs of high pass and low pass filters respectively after subsampling by 2.

This decomposition reduces the time resolution by half, while the frequency resolution is doubled, since the signal now contains only half the frequencies constituting the entire signal, which reduces the uncertainty of the frequency by half. This procedure is commonly referred to as subband coding [14]. Figure 2.2 shows a one dimensional three level wavelet decomposition of the signal $x[n]$ using the low pass filter $h_0[n]$ and high pass filter $g_0[n]$. This is known as Mallet tree decomposition [15]. At each level, the high pass filter produces detail information $d[n]$ and the low pass filter produces coarse approximation $s[n]$.

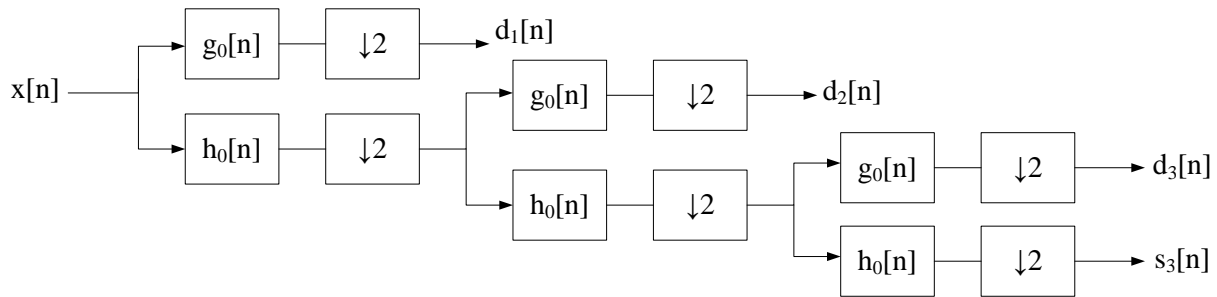


Figure 2.2: Three level wavelet decomposition tree.

After each decomposition level, the next stage approximation and detail components are extracted from the low frequency information. This further reduces the frequencies by half by discarding 50% of the redundant samples thus improving the resolution. This multi resolution capability is one of the prominent features of the wavelet transform. The filtering and decimation process are repeated till the desired level is reached. The number of levels depends on the length of the input signal.

The reconstruction of the original signal is achieved by performing the reverse process of decomposition. The approximation and detail components are upsampled by 2 and then passed through the low pass and high pass filters respectively. The low pass and high pass outputs are finally merged together. The process is repeated for the appropriate number of levels till the original signal is obtained. Figure 2.3 shows the inverse wavelet transform with the low pass filter $h_1[n]$ and the high pass filter $g_1[n]$.

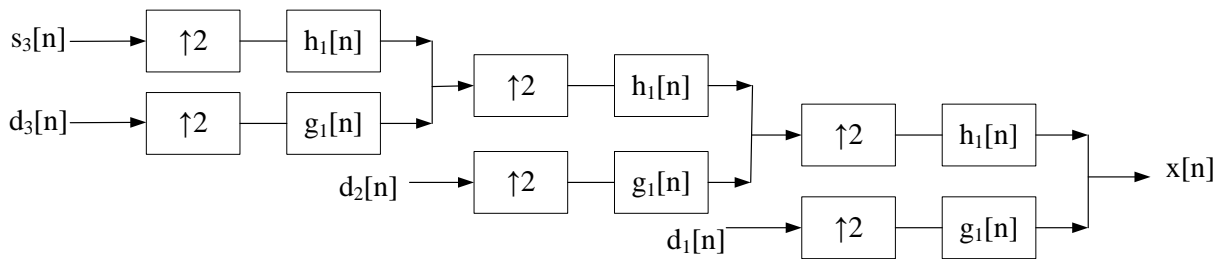


Figure 2.3: Three level wavelet reconstruction tree.

For an $N \times N$ image, the DWT is computed by processing the image using 1-D DWT in both horizontal and vertical direction to yield a 2-D DWT. The image is decomposed into four subbands, one coarse approximation LL and three detail subbands LH, HL and HH, each of size $N/4 \times N/4$. The next level decomposition is continued on the LL subband. The process is

repeated for the desired number of levels, forming a pyramidal structure. Figure 2.3 shows the three level 2-D DWT decomposition of an image.

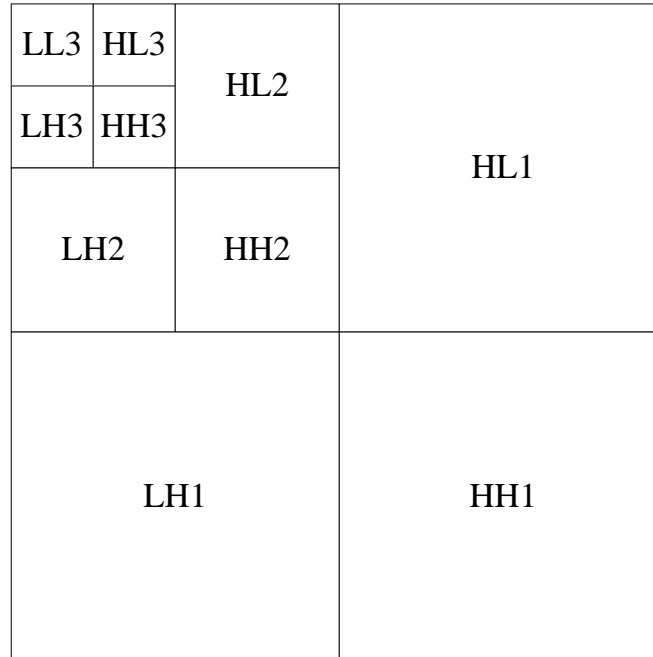


Figure 2.4: Three level 2-D DWT decomposition of an image.

2.4 Conditions for Perfect Reconstruction

In a Wavelet transform, it is necessary that the original signal be reconstructed perfectly from the wavelet coefficients. This can only be achieved by having a good combination of analysis and synthesis filters. The analysis and synthesis filters can be checked for perfect reconstruction by satisfying certain conditions. Let $h_0[n]$ and $g_0[n]$ be the analysis low pass and high pass filters respectively, while $h_1[n]$ and $g_1[n]$ are the synthesis low pass and high pass filters respectively. Then the conditions for perfect reconstruction are given by [16],

$$h_0[-n].h_1[n]+g_0[-n].g_1[n]=0 \quad (2.4)$$

$$h_0[n].h_1[n]+g_0[n].g_1[n]=2 \quad (2.5)$$

The first condition states that the reconstruction is free of aliasing and the second condition states that the amplitude distortion has an amplitude of one. From the equation, it can be noted that the perfect reconstruction condition does not change if the analysis and synthesis filters are switched.

It is difficult for most wavelet filter banks to produce good output coefficient values while satisfying the condition of perfect reconstruction. The performance of the wavelet transform filters can be determined by calculating the Peak Signal to noise Ratio (PSNR) between the reconstructed image and the original image. If the quality of the reconstructed image is closer to that of the original image, then the PSNR value measured between the output and the input will be high.

Chapter 3

Lifting Based Discrete Wavelet Transform

3.1 Introduction

The lifting scheme [17][18] or the lifting based discrete wavelet transform (LWT) is an efficient approach to construct the so called ‘second generation wavelets’. These wavelets in general are not necessarily translations and dilations of a function. The wavelets explained in the Chapter 2 are called first generation wavelets or classic wavelets. The lifting scheme has some additional advantages in comparison to the classic wavelets. The Lifting scheme allows faster implementation of the wavelet transform. In the classic transform, the signal is split into high pass and low pass signals and then subsampled, whereas the lifting scheme makes optimal use of the similarities between the low pass and high pass filters to speed up the calculation, sometimes increasing the speed by a factor of 2. The lifting scheme allows a fully in-place computation of the wavelet transform, so no auxiliary memory is needed.

Another interesting feature of the lifting scheme is that all the constructions are performed in the spatial domain. This is in contrast to the classic approach which relies heavily on the frequency domain. There are two main advantages of working in the spatial domain. First, it does not require a Fourier transform as the prerequisite for the construction of wavelets. Secondly, most practical applications are not Fourier based, hence lifting can be freely applied to complex situations. In the traditional wavelet transform, it is not immediately clear that the inverse transform is an actual inverse of the forward transform. The perfect reconstruction can only be checked using the Fourier analysis. On the other hand, the inverse transform of the lifting

scheme can be found immediately by undoing the operations of the forward transform. In other words, just replacing a “+” with a “-” and vice versa represents the inverse transform. One of the major advantages of the Lifting scheme is that, it can be applied to many real life situations that require functions and transforms to adapt to irregularly sampled data. Also, to analyse data that reside on curves or surfaces and to solve equations on curves and surfaces, the lifting scheme plays a vital role. In short, the second generation wavelets are more advantageous to use when compared to the traditional wavelet transform. For many years now, the lifting scheme has been widely used in research [19][20][21] with many improvements. The popular JPEG2000 standard [9] features lifting scheme as its core transform.

3.2 Constructing second generation wavelets

The basic idea behind the lifting scheme is to start with an initial wavelet called the “Lazy Wavelet”. There is no function associated with these lazy wavelets, except that it has the properties of a wavelet. The lifting scheme then tries to construct new wavelets by adding new basis functions. The lifting scheme then improves the properties of the constructed wavelet by finding a close correlation between the low and high frequency components. This is the inspiration behind the name “lifting scheme”.

Daubechies and Sweldon [17][18] showed that a new structure of wavelet transform can be constructed from any orthogonal and biorthogonal filters by employing factorization of a polyphase matrix. So the lifting scheme begins with a well known set of filters, say (h, g) , and the filters are split into even and odd. The polyphase matrix is given by,

$$P(z) = \begin{bmatrix} he & ge \\ ho & go \end{bmatrix} \quad (3.1)$$

The polyphase matrix is factorized using successive division approach (Euclidean algorithm) by choosing the appropriate Laurent polynomials from the filters h and g . Each step involves selecting the Laurent polynomials and finding the exact quotient and remainder. It is not possible for a set of filters to be exactly divisible by each other. But the coefficient at each division can be replaced by a suitable remainder that is more likely to be divisible at the next factorization. The suitable remainder can be obtained only by choosing the right quotient. Let $a(z)$ and $b(z) \neq 0$ with $|a(z)| \geq |b(z)|$ be any two Laurent polynomials. There always exists a quotient $q(z)$ and remainder $r(z)$, so that

$$a(z) = b(z) \cdot q(z) + r(z) \quad (3.2)$$

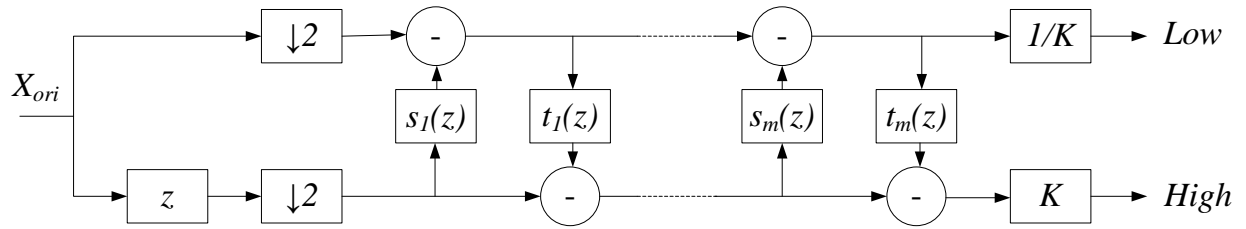
When $r(z) = 0$, the division by the polynomial is exact. The division of two Laurent polynomials is also a Laurent polynomial. The factorization of the polyphase matrix is not unique. There exists many possibilities for choosing the quotient and the division can proceed to achieve an entirely different set of lifting coefficients. The aim of the factorization is to represent the polyphase matrix as a set of upper and lower triangular matrices. This can be written as,

$$P(z) = \prod_{i=1}^m \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix} \begin{bmatrix} K & 1 \\ 1 & 1/K \end{bmatrix} \quad (3.3)$$

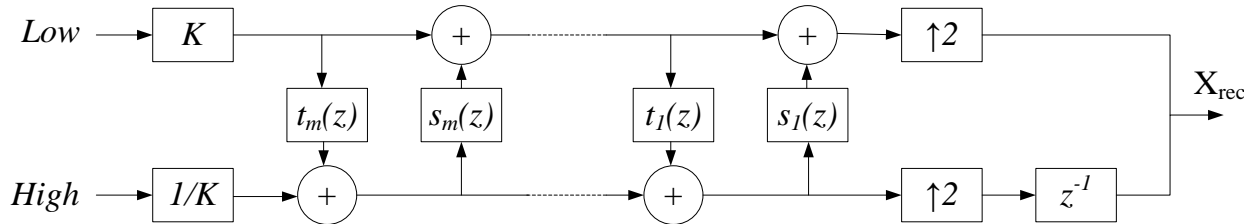
where, K is a non-zero constant and the Laurent polynomials $s_i(z)$ and $t_i(z)$ make up the primal and dual lifting stages respectively. The polyphase matrix corresponding to the forward transform is now given by,

$$\tilde{P}(z) = \prod_{i=1}^m \begin{bmatrix} 1 & 0 \\ -s_i(z^{-1}) & 1 \end{bmatrix} \begin{bmatrix} 1 & -t_i(z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1/K & 1 \\ 1 & K \end{bmatrix} \quad (3.4)$$

In the case of orthogonal filters, $P(z) = \tilde{P}(z)$. Depending on the factorization, there may exist many primal and dual lifting steps. Figure 3.1 shows the lifting based forward and inverse transform with m lifting steps.



a) Forward transform with m primal and dual lifting steps



b) Inverse transform with m primal and dual lifting steps

Figure 3.1.a) and b) are the analysis and synthesis representation of the polyphase matrix.

3.3 Applying lifting based transform to images

The process of implementing lifting steps on images is very similar to using it on discrete-time signals, except that signals are one dimensional while images are two dimensional. First, let us consider a signal X with 2^i samples. The signal when decomposed gives a coarse signal s_{i-1} and a detail signal d_{i-1} . The lifting transform generally consists of three steps: split, predict and update, as shown in Figure 3.2.

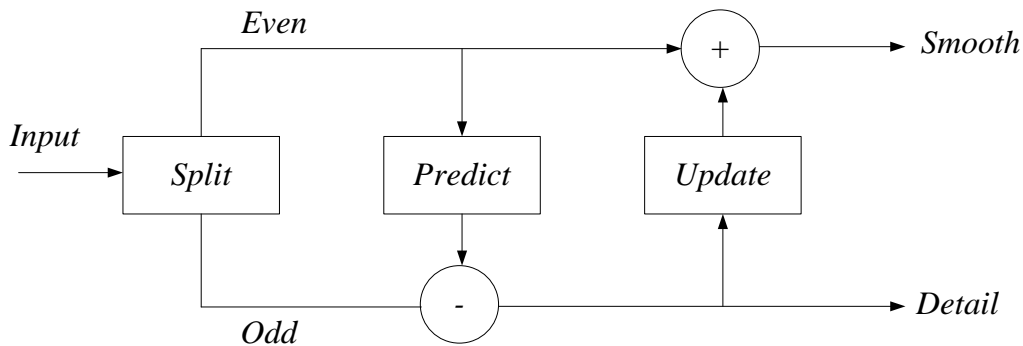


Figure 3.2: Basic steps in lifting scheme

The input is first split into two sets of samples. One set containing the even samples X_{2l} and the other contains all the odd samples X_{2l+1} . Each set now holds exactly half the number of samples compared to the original. The process of splitting the input into even and odd is called the lazy wavelet transform. If a signal has a local correlation structure, then the even and odd subsets will be highly correlated. It is possible to predict one set from the other with a reasonable accuracy. In this case the even sample is predicted from the odd one. The difference is then propagated. The difference is therefore the detail component d_{i-1} . Depending on the prediction operator, it is possible to represent the detail more efficiently. The even signal is then approximated by

selecting the suitable update operator to replace the even signal with an average. The computations are carried out in place: the even locations are replaced by averages and the odd ones with detail. This can be shown as,

$$\begin{aligned}
 (X_{2l+1}, X_{2l}) &= \textit{Split}(X) \\
 \textit{Detail}(d_l) &= X_{2l+1} - \textit{Predict}(X_{2l}) \\
 \textit{Smooth}(s_l) &= X_{2l} + \textit{Update}(d_l)
 \end{aligned}
 \tag{3.5}$$

The inverse transform is fairly simple. By undoing the predict and update step, it is possible to obtain the original signal. By just changing the signs of the predict and update steps, the original even and odd sample is reconstructed. Finally, the even and odd samples are merged together to form the original signal, this is shown in Figure 3.3.

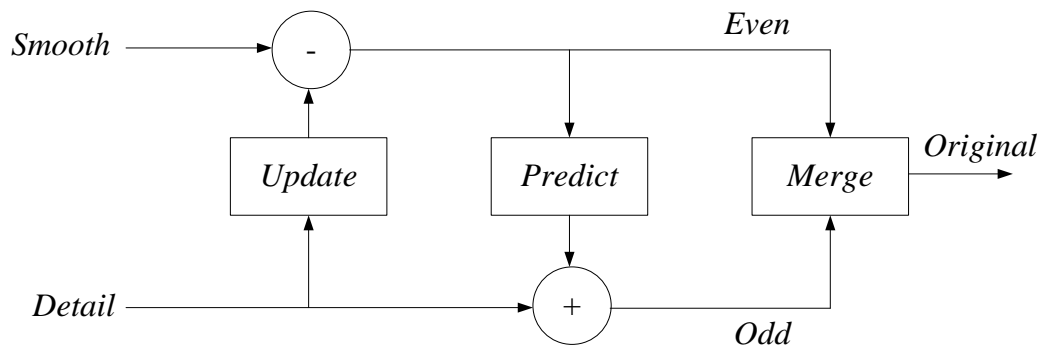


Figure 3.3: Inverse lifting transform

The lifting block shown in Figure 3.2 is a 1-D wavelet transform. In order to decompose an image, the input image is subjected to vertical and horizontal scanning using a 1-D transform. In the case of an image, the 2-D transform gives one coarse subband (LL) and three detail subbands (HL, LH and HH). Further decomposition can be achieved by applying the transform to the LL subband. The procedure can be extended to the desired number of decomposition levels.

The key dilemma is to decide upon which transform architecture to use. There are many lifting based transforms developed in recent years, but only a few have been found to be efficient in various applications. One such popular lifting scheme is the CDF 9/7 transform, which is used in the JPEG2000 standard. In our research, we focus on the popular Daubechies-4 tap (D4) and 6 tap (D6) lifting based discrete wavelet transform [17]. The reason for using these two variations is that, when optimized, the D4 and D6 prove to be faster and less complex for real-time applications, which is discussed in detail in the latter part of this thesis.

Daubechies family of orthogonal wavelets runs from D2 to D20 [23]. The complexity of the Daubechies wavelets increases with the order. D4 and D6 are the two most widely used wavelets of the entire series. One of the key features of the Daubechies wavelets is the vanishing moments. The vanishing moments are instances when the wavelet function becomes zero. So based on this, Daubechies [48] designed a type of wavelet for the given vanishing moments and obtained the minimum size discrete filter. The conclusion is that, for p vanishing moments, the minimum filter size is $2p$. D2 has one vanishing moment, D4 has two, D6 has 3 and so on. The vanishing moments are a necessary condition for the smoothness of the wavelet function. They also support the regularity and symmetry of the wavelet function. A high number of vanishing moments help to compress the regular parts of the signal better. However, it increases the size of support of the wavelets which can cause problems when the signal is discontinuous. This is one of the reasons why D4 and D6 are more widely used. They are more suitable for this research since they can be readily converted to an integer based transform with faster processing capability and reduced complexity.

3.4 Daubechies-4 (D4) Lifting Wavelet Transform

Daubechies-4 tap orthogonal filter is the simplest of the wavelet member in the Daubechies family with two vanishing moments, which is best to compress perfectly linear signals. The 4 tap corresponds to the number of analysis filter coefficients. The filter pairs h and g are then given by [22],

$$\begin{aligned} h(z) &= C_0 + C_1 z^{-1} + C_2 z^{-2} + C_3 z^{-3} \\ g(z) &= -C_3 z^2 + C_2 z^1 - C_1 + C_0 z^{-1} \end{aligned} \quad (3.6)$$

where, the four coefficients C_0, C_1, C_2 and C_3 are given by,

$$C_0 = \frac{1+\sqrt{3}}{4\sqrt{2}}, \quad C_1 = \frac{3+\sqrt{3}}{4\sqrt{2}}, \quad C_2 = \frac{3-\sqrt{3}}{4\sqrt{2}} \quad \text{and} \quad C_3 = \frac{1-\sqrt{3}}{4\sqrt{2}} \quad (3.7)$$

Now, the polyphase matrix is,

$$P(z) = \tilde{P}(z) = \begin{bmatrix} he(z) & ge(z) \\ ho(z) & go(z) \end{bmatrix} = \begin{bmatrix} C_0 + C_2 z^{-1} & -C_3 z^1 - C_1 \\ C_1 + C_3 z^{-1} & C_2 z^1 + C_0 \end{bmatrix}, \quad (3.8)$$

and the factorization is given by [17],

$$P(z) = \tilde{P}(z) = \begin{bmatrix} 1 & -\sqrt{3} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{\sqrt{3}}{4} + \frac{\sqrt{3}-2}{4} z^{-1} & 1 \end{bmatrix} \begin{bmatrix} 1 & z \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{3}+1}{\sqrt{2}} & 0 \\ 0 & \frac{\sqrt{3}-1}{\sqrt{2}} \end{bmatrix} \quad (3.9)$$

Hence, on the analysis side the polyphase matrix is factored as,

$$\tilde{P}(1/z)^t = \begin{bmatrix} \frac{\sqrt{3}+1}{\sqrt{2}} & 0 \\ 0 & \frac{\sqrt{3}-1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ z^{-1} & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{\sqrt{3}}{4} + \frac{\sqrt{3}-2}{4}z \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\sqrt{3} & 1 \end{bmatrix} \quad (3.10)$$

This corresponds to the forward transform implementation given by the following expressions,

$$\begin{aligned} d_l^{(1)} &= X_{2l+1} - \sqrt{3}X_{2l} \\ s_l^{(1)} &= X_{2l} + \frac{\sqrt{3}}{4}d_l^{(1)} + \frac{\sqrt{3}-2}{4}d_{l+1}^{(1)} \\ d_l^{(2)} &= d_l^{(1)} - s_{l-1}^{(1)} \\ d_l &= \frac{\sqrt{3}-1}{\sqrt{2}}d_l^{(2)} \\ s_l &= \frac{\sqrt{3}+1}{\sqrt{2}}s_l^{(2)} \end{aligned} \quad (3.11)$$

The inverse transform is performed by reversing the above process and flipping the signs. The polyphase matrix factorization shown is not unique. This research is based on the above polyphase matrix, as it has simpler lifting steps and less complex coefficients. Consider $C = \sqrt{3}$ and $R = \sqrt{2}$, Figure 3.4 shows the forward transform block of the Daubechies-4 lifting wavelet transform.

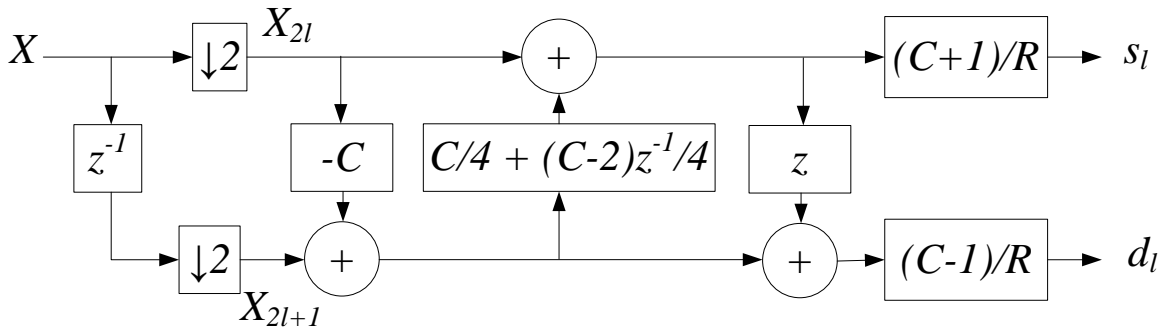


Figure 3.4: Lifting based Daubechies-4 forward transform block.

3.5 Daubechies-6 (D6) Lifting Wavelet Transform

The next member in the class of Daubechies wavelets, with three vanishing moments, is the Daubechies-6 transform. The three vanishing moments are best to compress quadratic signals. The D6 lifting has six filter coefficients. The complexity of the transform is considered to be higher than D4. From [22] we have the six filter coefficients as,

$$\begin{aligned}
 C_{-2} &= \sqrt{2} \left(1 + \sqrt{10} + \sqrt{5 + 2\sqrt{10}} \right) / 32 & C_{-1} &= \sqrt{2} \left(5 + \sqrt{10} + 3\sqrt{5 + 2\sqrt{10}} \right) / 32 \\
 C_0 &= \sqrt{2} \left(10 - 2\sqrt{10} + 2\sqrt{5 + 2\sqrt{10}} \right) / 32 & C_1 &= \sqrt{2} \left(10 - 2\sqrt{10} - 2\sqrt{5 + 2\sqrt{10}} \right) / 32 \\
 C_2 &= \sqrt{2} \left(5 + \sqrt{10} - 3\sqrt{5 + 2\sqrt{10}} \right) / 32 & C_3 &= \sqrt{2} \left(1 + \sqrt{10} - \sqrt{5 + 2\sqrt{10}} \right) / 32
 \end{aligned} \tag{3.12}$$

The polyphase matrix is now given by [17],

$$P(z) = \tilde{P}(z) = \begin{bmatrix} he(z) & ge(z) \\ ho(z) & go(z) \end{bmatrix} = \begin{bmatrix} C_{-2}z + C_0 + C_2z^{-1} & -C_{-3}z - C_1 - C_{-1}z^{-1} \\ C_{-1}z + C_1 + C_3z^{-1} & C_{-2}z + C_0 + C_2z^{-1} \end{bmatrix} \tag{3.13}$$

The factorisation of the polyphase matrix gives six lifting coefficients as shown below,

$$P(z) = \tilde{P}(z) = \begin{bmatrix} 1 & 0 \\ \alpha & 1 \end{bmatrix} \begin{bmatrix} 1 & \beta' + \beta z^{-1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \gamma' z + \gamma & 1 \end{bmatrix} \begin{bmatrix} 1 & \delta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \zeta & 0 \\ 0 & 1/\zeta \end{bmatrix} \tag{3.14}$$

where,

$$\begin{aligned}
\alpha &= -0.4122865950, \\
\beta &= -1.5651362796, \\
\beta' &= 0.3523876576, \\
\gamma &= 0.0284590896, \\
\gamma' &= 0.4921518449, \\
\delta &= -0.3896203900, \\
\zeta &= 1.9182029462
\end{aligned} \tag{3.15}$$

This leads to the forward transform factorization of the polyphase matrix, by reversing the steps and transposing the matrix, given by,

$$\tilde{P}(1/z)^t = \begin{bmatrix} \zeta & 0 \\ 0 & 1/\zeta \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \delta & 1 \end{bmatrix} \begin{bmatrix} 1 & \gamma'z^{-1} + \gamma \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \beta' + \beta z & 1 \end{bmatrix} \begin{bmatrix} 1 & \alpha \\ 0 & 1 \end{bmatrix} \tag{3.16}$$

The implementation of the forward transform matrix can be expressed as,

$$\begin{aligned}
s_l^{(1)} &= X_{2l} + \alpha X_{2l+1} \\
d_l^{(1)} &= X_{2l+1} + \beta' s_l^{(1)} + \beta s_{l+1}^{(1)} \\
s_l^{(2)} &= s_l^{(1)} + \gamma' d_l^{(1)} + \gamma' d_{l-1}^{(1)} \\
d_l^{(2)} &= d_l^{(1)} + \delta s_l^{(2)} \\
s_l &= \zeta s_l^{(2)} \\
d_l &= \frac{d_l^{(2)}}{\zeta}
\end{aligned} \tag{3.17}$$

The above expression can be represented in the form of lifting step as shown in Figure 3.5 below,

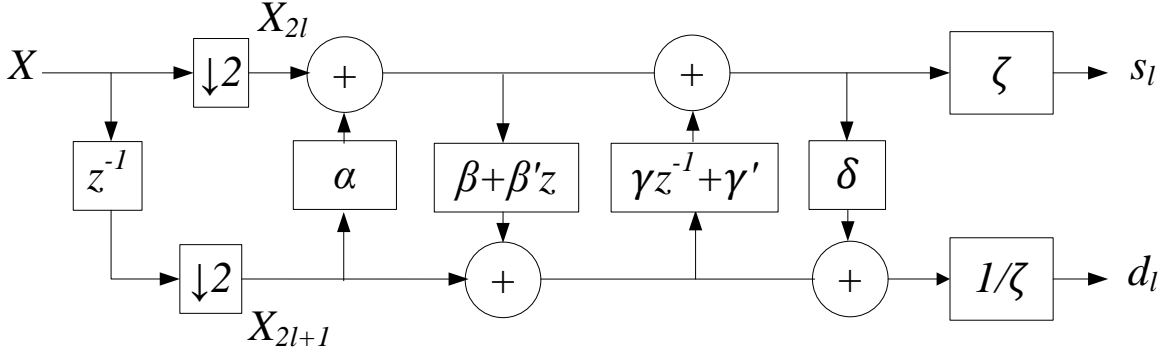


Figure 3.5: Lifting based Daubechies-6 forward transform block.

Now in this work, we apply integer mapping, known as Algebraic Integer Quantization (AIQ), to error-freely map the filter coefficients with integers. The mapping technique is described in the following chapter.

3.6 Previous Works

The architecture proposed by Q.Dai et al [34] is a multidimensional m-D DWT with multistage pipelining, hardware is tested using Daubechies-4 tap filter. The main drawback of the design is the huge RAM module which increases the hardware cost. The design adds to more number of multipliers and adders, and does not provide efficient implementation of the wavelet filter coefficients. On the other hand, the wavelet processing core proposed by S.W. Lee et al [35] is based on the fast-lifting based 9/7 DWT. The design uses a complex control circuitry and large kernel size to perform multiplication. The folded 9/7 DWT architecture proposed by M. Martina et al [36] gives a multiplierless implementation of the 9/7 DWT, but has low throughput rate. Also, the design is based on the floating point implementation of the filter coefficients, that introduces round-off error at the output. Similarly, the folded lifting based 9/7 architecture proposed by G. Shi [37] possesses low critical path latency, but at the cost of low throughput

rate, more intermediate registers and high control complexity, compared to the direct lifting approach. The architecture of Y.K. Lai [38] is based on pipelined lifting based 9/7 transform which uses register array to perform the scanning operation. This increases the number of memory buffers used in the design. The other drawback of the design is the use of 12 bit absolute values to represent the filter coefficients. This is also true in the architecture proposed by C.T. Huang et al [39] and B.F. Wu [40]. Both the design follows a multistage pipelining of lifting based 9/7 filter based on flipping structure. The drawback of the design includes implementation of the filters using 12 bit precision. The design [40] has low throughput rate and high number of memory buffers. The architecture proposed by Y.H. Seo et al [41] is a combination of lifting based 5/3 and lifting based 9/7 transform. The input bit length is 8 bits, however, the output coefficients bit length is 17. Out of the 17 bits, 10 bits are allocated for whole number and 7 bits are allocated for partial number. The architecture uses a large number of logic cells and has high control complexity. S.K. Paek et al [42] proposed an architecture based on semi-recursive D4 DWT. The main noticeable disadvantage of the design is the huge hardware cost. This is because the design uses twelve 16 bit adders, four 8 x 16 multipliers and twelve 16 x 16 multipliers. Also, DWT filter coefficients and DWT input data should be 16 bit. K.K. Parhi et al [43] introduced two architectures for 1D DWT: folded and digit serial. The architecture has several drawbacks such as complex routing, large silicon area and high power consumption.

The architectures proposed by M. Ali et al [45] and K. Benkrid et al [46] are based on the D8 DWT filter. Despite the performance of D8, the design has large area (hardware cost) and complex calculations. Recently, K. Wahid [47] proposed the integer based implementation of D4 and D6 DWT which eliminates the round-off error that is caused in the floating point implementation. Despite the many advantages of the AIQ design, the transform used is the

conventional DWT, which uses large number of memory buffers and uses large area on hardware. So in this thesis, we concentrate on using the lifting based D4 and D6 and propose an integer based lifting architecture.

Chapter 4

Proposed Lifting-Based Algorithm

4.1 Introduction

A more efficient approach to lossless compression is the use of integer transforms. The transform coefficients exhibit the feature of being exactly represented by finite precision numbers, and this allows for truly lossless coding. The lifting scheme has been introduced for the efficient computation of DWT. Its main advantage with respect to filter bank structure lies in its better computational efficiency and in fact it enables a new method for filter design. Moreover, the integer wavelet transform (IWT) can be computed from any real valued wavelet filter by simply modifying the lifting scheme. Therefore the lifting scheme represents a distinguished choice for the implementation of encoders with lossy or lossless compression capabilities, providing a common core for computing both DWT and IWT [24].

Moreover, integer based transforms are much faster than the floating point arithmetic in almost all general purpose computers because the floating point wavelet transform demands for longer data length than the integer wavelet transform. Another benefit of using integer wavelet is the reversibility, that is, the image can be reconstructed losslessly because all the coefficients are integers and can be stored without rounding off errors [17]. This research is focused on deriving a completely lossless integer based lifting scheme for the Daubechies-4 tap and Daubechies-6 tap transforms in the form of algebraic integers.

4.2 Algebraic Integer Quantization (AIQ)

The idea of using algebraic integers in signal processing applications was first explored by Cozzens and Finkelstein [25]. Later, Wahid et al. proposed a more generalized version of it, named as Algebraic Integer Quantization (AIQ), and showed its application to Discrete Wavelet Transforms [26][47]. The AIQ is a lossless mapping technique which is used to reduce the round off error when implementing a transform. This error is due to the lack of exact representation of the irrational numbers which forms the basis of the filter coefficients. The error propagates through the decomposition stage and degrades the quality of the image during reconstruction. Using AIQ technique, it is possible to minimize the approximation error and efficiently compute the Daubechies-4 and Daubechies-6 discrete wavelet transform [26]. The AIQ technique was never applied to lifting based wavelets; as a result, in this thesis work, we propose AIQ scheme to compute the lifting based Daubechies-4 and Daubechies-6. The lifting based D4 and D6 are faster and more efficient than the conventional ones. AIQ can be readily implemented for both the transforms without increasing the complexity of the lifting transform, while maintaining the precise representation of the irrational coefficients by an equivalent AIQ based transformed coefficients.

A complete integer based lifting wavelet transform is created by performing scaling of the lifting filter coefficients [27]. The scaling parameter is chosen as the factor of 2. The scaling is restricted to the factor of 2 for the ease of implementation. This is because multiplication by 2 is just shifting the bits. Since multiplications are costly operations on hardware, measures are taken to reduce the complexity. Scaling constants other than 2 could add to the increase in hardware complexity. Let the scaling factor of the filter coefficients be 2^x . The value of x can be calculated

where the coefficients are completely lossless. The value of x can vary with the type of the transform. The filter coefficients are scaled and rounded to integer values. The inverse scaling 2^{-x} is also performed in each of the lifting step. This is to ensure that the bit lengths of coefficients are kept unaltered when performing a 2-D multiresolution transform. But multiplying and dividing at the same step can produce a floating point output at each stage. To ascertain that the intermediate values be kept as integers, the input image is also scaled by a factor of 2. This keeps the resolution of the coefficients as integer. Let the input scaling be denoted as 2^y . Also, the scaling coefficients at the end of the transform can be removed and implemented together with the inverse scaling step 2^{-y} at the quantization stage [28].

The proposed integer based transform does not change the structure of the lifting scheme, instead the values are changed to integers using scaling factors and finally reverted back to original values at the end of the transform. This is same as multiplying and dividing by the same number. All the lifting filter coefficients share the same scaling parameter. The coefficients obtained after decomposition are same as the one obtained using conventional lifting transform. The best value of x and y are found by calculating the average PSNRs of an image at different bit rates such as 4, 2, 1, 0.5 and 0.25bpp. The PSNRs are plotted for various values of x and y . The minimum value of x and y where the PSNR is at maximum is chosen for scaling, this is shown in Figure 4.2. The AIQ technique also explores the possibility of reducing the number of filter coefficients. By selecting the right scaling factor, one or more filter coefficients can be converted to a factor of 2. This reduces the number of multiplications in a transform which in turn reduces the area and power on hardware.

4.3 AIQ based Daubechies-4 lifting scheme

The Daubechies-4 (D4) lifting scheme contains three coefficients which constitutes three multiplication operations. All the three coefficients are irrational, that is, the value $\sqrt{3}$ has infinite number of decimal (as well as binary) bits. Truncating the decimal bits leads to error which accumulates at each stage of decomposition. To avoid this, both the input and the irrational coefficients are scaled and replaced by integers with greater value. Let the D4 lifting scheme scaling parameter for filter coefficients be 2^μ , while the input scaling parameter be 2^τ . The new polyphase of the AIQ based D4 lifting scheme can be factored as,

$$P^1(z) = \begin{bmatrix} 1 & A/2^\mu \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ (B+Cz^{-1})/2^\mu & 1 \end{bmatrix} \begin{bmatrix} 1 & z \\ 0 & 1 \end{bmatrix} \quad (4.1)$$

where,

$$A = \left[2^\mu * -\sqrt{3} \right], B = \left[2^\mu * \frac{\sqrt{3}}{4} \right] \text{ and } C = \left[2^\mu * \frac{\sqrt{3}-2}{4} \right] \quad (4.2)$$

The above polyphase matrix corresponds to the inverse lifting operation. The lifting step on the encoding side is shown in Figure 4.1. The output low pass and high pass shown are the new decomposed coefficients obtained prior to the inverse scaling stage.

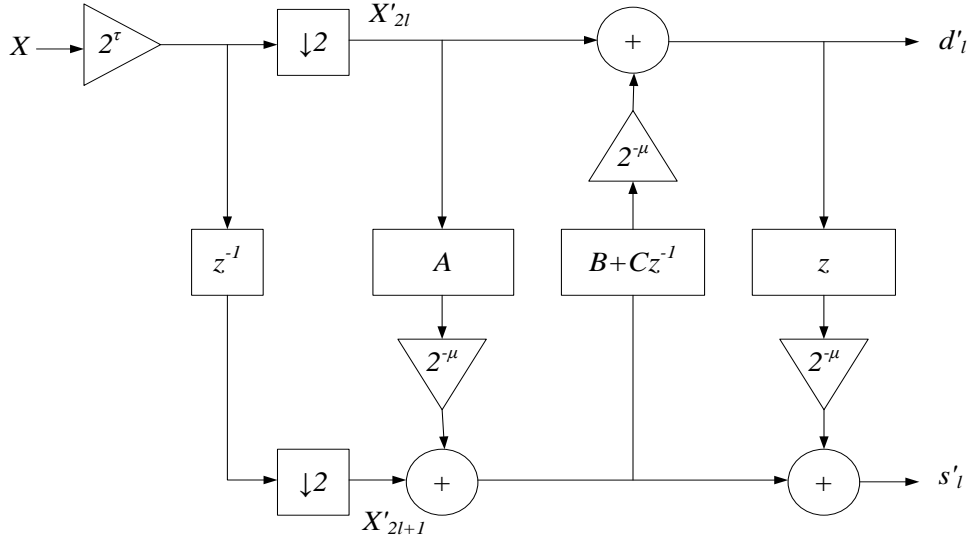


Figure 4.1: AIQ model of the Lifting based D4 DWT

here,

$$\begin{aligned}
 X'_{2l} &= 2^\tau * X_{2l} \\
 X'_{2l+1} &= 2^\tau * X_{2l+1}
 \end{aligned}
 \tag{4.3}$$

The selected range for μ and τ is $\mu \in [4, 25]$ and $\tau \in [1, 15]$ respectively. To evaluate the performance, we have used benchmark images used in image processing, such as, 'Lena'. The image is a grayscale image of size 512 x 512. The image is decomposed and reconstructed for various bit rates such as 4, 2, 1, 0.5 and 0.25bpp. To achieve the required bit rate we used the proposed quantization approach (will be discussed further in the next chapter). Figure 4.2 shows the MATLAB mesh plot of average PSNRs against values of μ and τ .

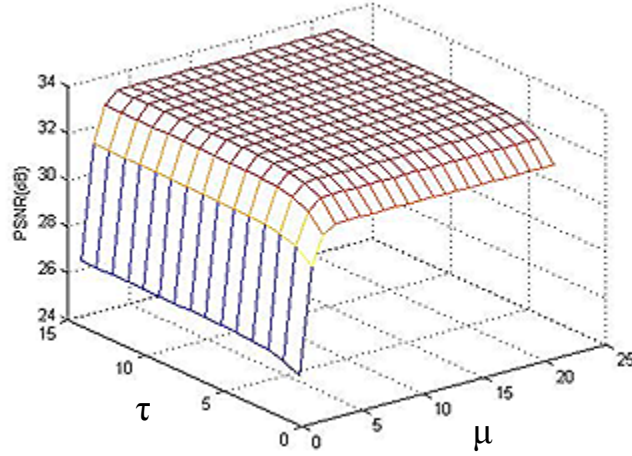


Figure 4.2: Average PSNRs plotted against values of τ and μ for the image ‘Lena’.

The maximum value occurs at 33.31dB after which the value of PSNRs at constant. The constant PSNR value shows that the new integer coefficients are lossless. The minimum value of μ and τ at which the maximum PSNR occurs is 8 and 11 respectively. We use these values for scaling. The new integer based lifting D4 filter coefficients are formulated as,

$$\begin{aligned}
 A &= \left\lfloor 2^8 * -\sqrt{3} \right\rfloor \approx 443 \\
 B &= \left\lfloor 2^8 * \frac{\sqrt{3}}{4} \right\rfloor \approx 110 \\
 C &= \left\lfloor 2^8 * \frac{\sqrt{3}-2}{4} \right\rfloor \approx -17
 \end{aligned} \tag{4.4}$$

The forward transform implementation is now given by,

$$\begin{aligned}
 d_l^{(1)} &= X'_{2l+1} - \left\lfloor (443 * X'_{2l}) / 2^8 \right\rfloor \\
 s'_l &= X'_{2l} + \left\lfloor (110 * d_l^{(1)} - 17 * d_{l+1}^{(1)}) / 2^8 \right\rfloor \\
 d'_l &= d_l^{(1)} - s'_{l-1}
 \end{aligned} \tag{4.5}$$

The above integer model of the D4 lifting scheme contains three multiplications and four additions.

Finally, the error introduced in the process is determined by calculating the difference between the two decomposed images (of the image 'Lena') obtained by using the conventional lifting based D4 and the proposed integer based D4 algorithm. Figure 4.3 shows the histogram levels of the error image.

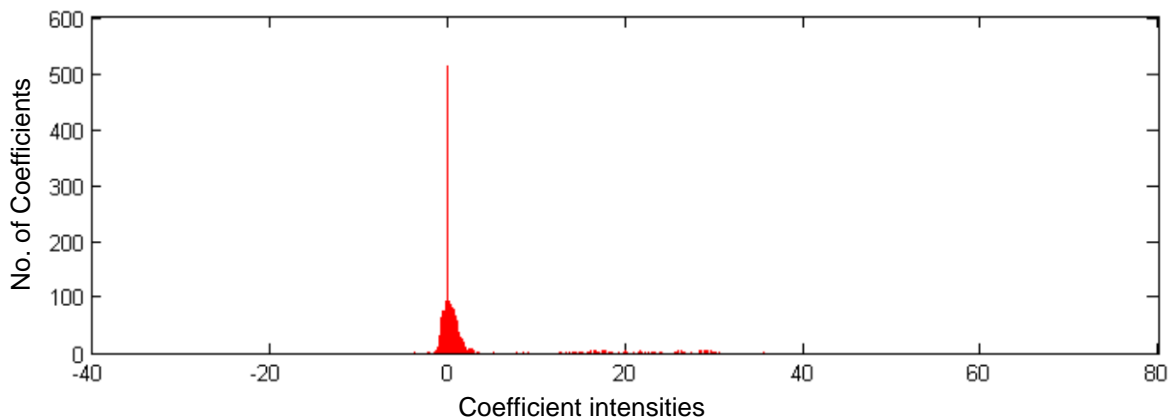


Figure 4.3: Error added to the output coefficient values using the proposed integer based D4 architecture.

4.4 AIQ based Daubechies-6 lifting scheme

The lifting based Daubechies-6 (D6) forward transform contains six filter coefficients excluding the final scaling coefficient. The complexity of D6 is higher than D4. With more number of lifting steps there is a high risk of error accumulation. Moreover, all the D6 lifting coefficients are irrational numbers. A slightest decimal truncation can result in big difference in the output coefficient values, which is crucial at high compression rates. D6 gives very little choice for modifications. This is because, adding many improvements to the D6 transform may end up increasing its complexity. Based on similar AIQ technique to the one used in lifting based D4,

the proposed D6 lifting scheme has both lossless and reduced complexity. Consider the filter scaling parameter as 2^η and the input scaling parameter as 2^ρ . The new polyphase matrix with new transformed coefficients are given by,

$$P'(z) = \begin{bmatrix} 1 & 0 \\ a/2^\eta & 1 \end{bmatrix} \begin{bmatrix} 1 & (bz^{-1} + b')/2^\eta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ (c + c'z)/2^\eta & 1 \end{bmatrix} \begin{bmatrix} 1 & d/2^\eta \\ 0 & 1 \end{bmatrix} \quad (4.6)$$

where,

$$a = \lfloor 2^\eta * \alpha \rfloor, b = \lfloor 2^\eta * \beta \rfloor, b' = \lfloor 2^\eta * \beta' \rfloor, c = \lfloor 2^\eta * \gamma \rfloor, c' = \lfloor 2^\eta * \gamma' \rfloor \text{ and } d = \lfloor 2^\eta * \delta \rfloor \quad (4.7)$$

Figure 4.4 corresponds to the block diagram of the polyphase matrix on the analysis side. The scaling coefficients at the end of the transform are removed and are implemented together with the inverse scaling at the quantization step.

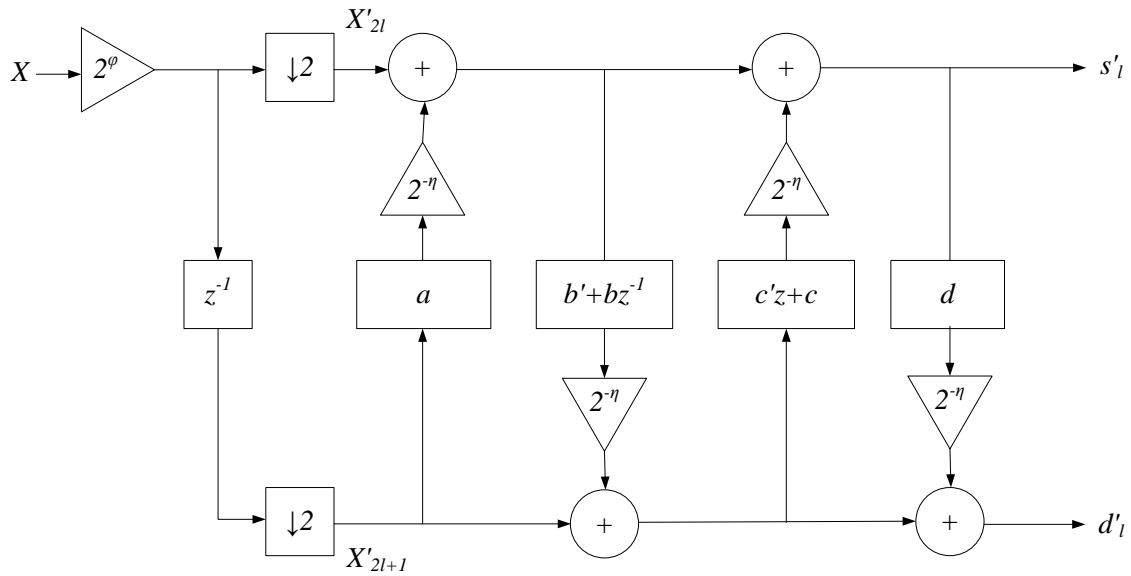


Figure 4.4: AIQ model of the Lifting based D6 DWT

here,

$$\begin{aligned} X'_{2l} &= 2^\varphi * X_{2l} \\ X'_{2l+1} &= 2^\varphi * X_{2l+1} \end{aligned} \quad (4.8)$$

The values of η and φ are chosen in the range of $\eta \in [4, 25]$ and $\varphi \in [0, 15]$. The average PSNRs are plotted for bit rates 4, 2, 1, 0.5 and 0.25bpp. The standard grayscale image ‘Lena’ of size 512 x 512 is used to plot the average PSNRs against the range of values of η and φ . The values of η and φ are found to be 7 and 1 respectively. The filter coefficients are multiplied by the new scaling factor. The new integer based lifting D6 filter coefficients are formulated as,

$$\begin{aligned} a &= \lfloor 2^7 * \alpha \rfloor = \lfloor 2^7 * -0.4122865950 \rfloor \simeq -52 \\ b &= \lfloor 2^7 * \beta \rfloor = \lfloor 2^7 * -1.5651362796 \rfloor \simeq -200 \\ b' &= \lfloor 2^7 * \beta' \rfloor = \lfloor 2^7 * 0.3523876576 \rfloor \simeq 45 \\ c &= \lfloor 2^7 * \gamma \rfloor = \lfloor 2^7 * 0.0284590896 \rfloor \simeq 4 \\ c' &= \lfloor 2^7 * \gamma' \rfloor = \lfloor 2^7 * 0.4921518449 \rfloor \simeq 64 \\ d &= \lfloor 2^7 * \delta \rfloor = \lfloor 2^7 * -0.3896203900 \rfloor \simeq -49 \end{aligned} \quad (4.9)$$

The above integer coefficient give a lossless implementation of the D6 lifting scheme. The above equation shows six filter coefficients technically representing six multiplication operations. But, the coefficients 4 and 64 are multiples of 2. This is merely shifting the bits, and does not cost as much as multiplication operation on hardware. According to the AIQ technique, we reduce 2 coefficients, but the lifting step stays true. The modified forward implementation of the integer based D6 LWT is,

$$\begin{aligned}
s_l^{(1)} &= X'_{2l} - \left\lfloor (52 * X'_{2l+1}) / 2^7 \right\rfloor \\
d_l^{(1)} &= X'_{2l+1} + \left\lfloor 45 * s_l^{(1)} - 200 * s_{l+1}^{(1)} / 2^7 \right\rfloor \\
s'_l &= s_l^{(1)} + \left\lfloor 1 * d_l^{(1)} + 16 * d_{l-1}^{(1)} / 2^5 \right\rfloor \\
d'_l &= d_l^{(1)} - \left\lfloor 49 * s'_l / 2^7 \right\rfloor
\end{aligned} \tag{4.10}$$

The final forward transform expression shown above has 4 multiplications and 6 additions. This brings the complexity of the transform closer to the well known CDF 9/7 lifting scheme. The bit length of the input increase by 1, since φ is 1. The most common bit length of a grayscale image is 8 bits per pixel for an input image. The above proposed Daubechies-6 lifting is less complex and faster compared to the conventional lifting based D6. The perfect reconstruction function stays true for both AIQ based D4 and D6. This is because the steps of scaling are repeated in the inverse transform, which is evident from the polyphase matrix.

Now, the error added to the output coefficient values are calculated for the image 'Lena' by calculating the difference between the decomposed images of the conventional and the proposed technique. Figure 4.5 shows the histogram levels of the error image.

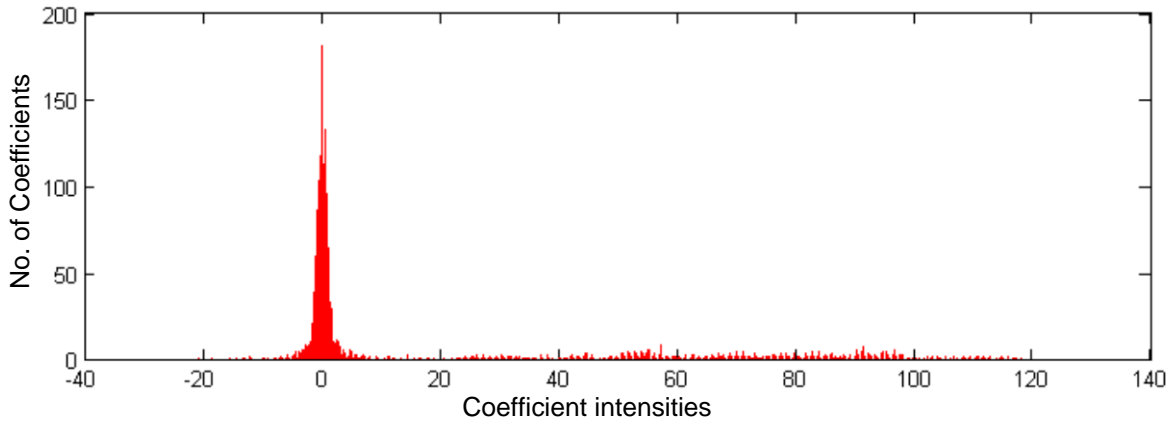


Figure 4.5: Error added to the output coefficient values using the proposed integer based D6 architecture.

Chapter 5

Proposed Quantization Scheme

5.1 Introduction

Quantization refers to the process of approximating a continuous set of values in the image with finite set of values. The input to the quantizer is always the original data and the output is always one amongst those finite set of values. This is similar to throwing away the samples by sampling operation. A good quantizer represents the original data with minimum loss and distortion. Hence quantization process is lossy and irreversible. It uses statistical analysis or mathematical expressions to remove most of the redundant information. This is crucial in applications where the storage area is limited. The amount of compression achieved through quantization is measured in terms of bit rate or compression ratio. Figure 5.1 shows the block diagram of an image compressor.

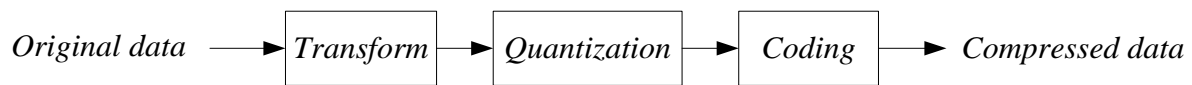


Figure 5.1: General diagram of an image processing algorithm.

5.2 Quantization in Wavelet Transforms

There are two types of quantization; Scalar quantization and vector quantization. In scalar quantization, each coefficient is treated separately in producing the output. But in vector quantization, the coefficients are group together as vectors and are processed together to produce the output. The use of scalar quantization is simple and fast, whereas vector quantization is

advantageous in case of optimality, but at the cost of increase in computational complexity. So, the most effective technique is the use of scalar quantization, which quantizes the wavelet coefficients individually. Scalar quantization uses step size to better approximate the wavelet coefficients. This is as simple as using a threshold value to remove all the data below that value, which is not so effective at high compression rates. The low pass coefficients carry the actual information, hence neglecting some values can result in loss of quality. Whereas, the high pass coefficients carry the details, and so it can be heavily quantized to achieve high compression rates. The problem with the wavelet transform is that the wavelet coefficient levels vary depending on type of image. The idea of using a fixed quantization matrix or step size proves to be ineffective in wavelets when the human visual system (HVS) is considered. This is because the significant coefficients appear at random in the high pass subbands depending on the amount of details in an image. Removing the significant coefficients from the high pass subband causes the intensities of the pixels to change after reconstruction. In order to preserve the visual quality of the image and to achieve the required bit rate, an adaptive quantization technique is needed.

5.3 Embedded Coding

When coding the wavelet coefficients into a bit stream, it is important not only to code the magnitude of the significant data but also its spacial location. For many years embedded coding is used to compress the wavelet coefficients. Embedded coding compresses the coefficients and codes its magnitude as a binary value, along with the information about its location. One of the most widely known embedded coding techniques is the Embedded Zerotree Wavelet (EZW). The EZW encoder was first developed by Shapiro [29] to use with the wavelet transforms. Unlike the Discrete cosine transform (DCT) which uses fixed quantization table, the EZW

encoder is based on progressive encoding which compresses the data into bitstream with increasing accuracy. Lossless compression can also be achieved with the EZW technique.

To begin with, EZW is a significance mapping algorithm that uses series of iterations to code the transform coefficients in the order of importance. EZW works on the principle that “An insignificant coefficient very likely has its descendants in the next finer scale as insignificant” [29]. The coefficients are encoded depending on the parent-child relationship. A low coefficient has four descendants in the next higher scale. Each of those four individual coefficients can be related to four descendants in the next higher subband forming a quad tree structure. Thus each parent has four children. So a zerotree is a quad tree of which values of all the nodes are equal to or smaller than the root. Figure 5.2 shows how the coefficients are linked to its descendants in the next scale and its scanning order.

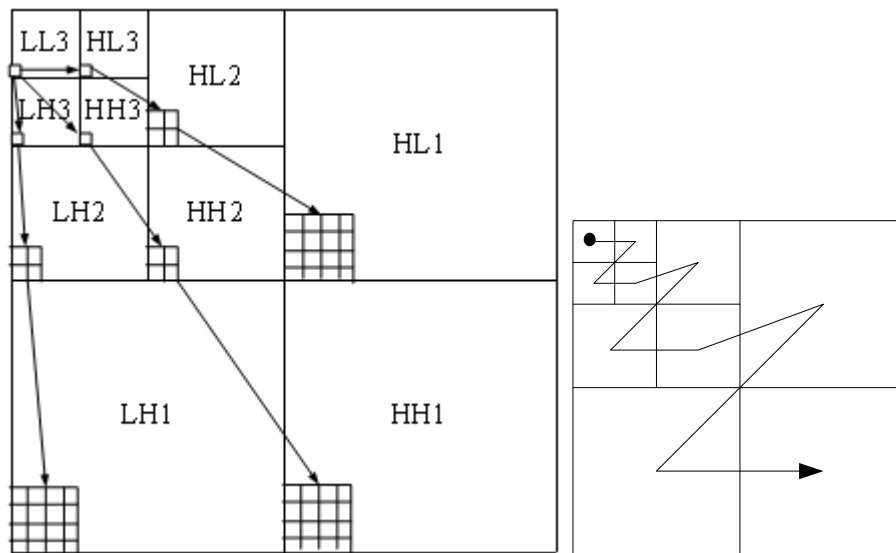


Figure 5.2: Wavelet coefficients relationship in the different subbands (left) and the scanning order (right)

The scanning order is generally called Morton scanning order. It is selected based on the importance of the subband, which are LL, HL, LH and finally HH. The coefficients are coded into the bitstream in this order. Let the wavelet coefficients be $C(x,y)$. The initial threshold is found using the following expression,

$$t_0 = 2^{\lfloor \log_2 (\text{MAX}(|C(x,y)|)) \rfloor} \quad (5.1)$$

The EZW encoder uses two passes to code the coefficients. In the first pass (dominant pass), the wavelet coefficients are scanned progressively using the initial threshold and a symbol is outputted for every coefficient. If the coefficient is larger than the threshold, then **P** (positive) is coded, if the coefficient is smaller than minus the threshold an **N** (negative) is coded. If the coefficient is the root of a zerotree then a **ZRT** (zerotree root) is coded, if the coefficient is smaller than the threshold but it is not the root of a zerotree, then an **IZ** (isolated zero) is coded and finally if the coefficient has no children then **Z** (zero) is coded. The process is repeated for all the subbands in the order. Finally, all the coefficients that are found significant are placed in the subordinate list and their positions in the image are made zero to prevent them from being coded again. At the end of each dominant pass the threshold is halved and the above process is repeated again.

The second pass (subordinate pass) codes all the values in the subordinate list into binary data. The subordinate pass is also called as refinement. After every dominant pass the subordinate list is scanned. For the first subordinate pass, the two intervals are selected between initial threshold t_0 and $2t_0$. The first interval is t_0 to $t_0 + t_0/2$ and the second interval is $(t_0 + t_0/2) + 1$ to $2t_0$. If the coefficient in the subordinate list is inside the first interval, then 0 is coded. If the coefficient

belongs to the upper interval, then 1 is coded. At the each subordinate pass, the intervals are halved and the sub upper and lower intervals are assigned 1 and 0, respectively. The dominant pass and subordinate pass are repeated till the threshold value reaches the minimum threshold. The minimum threshold value is selected depending on the desired bit rate. If the minimum threshold value is 0, then EZW becomes completely lossless.

Finally, the symbols coded in the dominant pass are converted to binary using separate coding techniques like arithmetic coding or Huffman coding. The final bitstream is the compressed data in binary format. The header of the bitstream contains information needed for reconstruction such as initial threshold, image size and the number of decomposition levels. The coding structure of EZW shows that it strictly neglects all the floating point information and considers all the coefficients as integers. The decoder follows the exact opposite steps of the encoder. In the inverse quantization, the decoder traces the every step back to produce the actual coefficient values. The EZW decoder uses successive approximation technique to reconstruct the exact coefficient values.

The EZW technique is considered to be computationally complex. In addition to the use of many symbols, the EZW relies on additional coding techniques to code its final results. The EZW algorithm was later replaced by a much faster and computationally superior coding technique known as Set Partitioning in Hierarchical trees (SPIHT) [30]. The SPIHT has amazing features which uses transform characteristics to find groups with similar MSBs. It uses the concept of bit-plane ordering, to code the magnitudes. Binary representations of the magnitude-ordered coefficients are transmitted first. This is found to be extremely advantageous over EZW because those significant coefficients that have descendants as significant in the next spatial orientations

are coded together. No extra coding step is necessary since the final output is a stream containing binary data of the coefficients. The SPIHT technique produces reconstructed images with exceptionally visual quality than EZW.

5.4 Proposed Quantization by Adaptive-Iterative Coding

Despite all the advantages, EZW and SPIHT are computationally very complex. This is because the coding scheme stores the entire image to perform the iterations. In addition, for each coefficient that is checked for significance, its corresponding descendants are also scanned. This is done for all the nodes till the last descendants. This is really time consuming and occupies a lot of the storage area. Though SPIHT performs this scan faster than EZW, it stores three huge lists to code the data directly in binary values.

To eliminate the disadvantages of both EZW and SPIHT, we propose an adaptive-iterative coding scheme based on the EZW technique. The idea is to code the subbands separately. By this way, the subband once it is coded can be made readily available for transmission. But the coding order of the subband is same as the importance of the subband. Instead of using various symbols to denote the coefficients, the proposed method uses only three symbols of binary values. If the coefficient is above the particular threshold, then '0' is coded. If the coefficient is below negative threshold, then '10' is coded. If the coefficients are not significant, then '1' is coded. There is more number of insignificant coefficients than significant coefficients. Hence the symbols are selected particular way to reduce the bit rate. Similar to EZW, the proposed coding scheme uses two passes to code the coefficients. The coefficients that are coded in the two passes are all in binary format. To reduce the bit rate further, the proposed method uses Run-Length Encoding (RLE) (section 5.5) followed by binary bit allocation for all the run-lengths. The header of the

bitstream contains information about the maximum allocated bit size of the run-lengths, initial threshold, image size and number of decomposition levels.

The proposed technique is very effective and fast when compared to EZW and SPIHT. The former coding scheme does not require entire image to start the coding operation. All the values are in binary format which makes it easier for transmission. In addition, separate minimum threshold values are selected for each subbands, which makes it more flexible for different types of images. The decoder uses reverse steps of the encoder to reconstruct the actual coefficients. The successive approximation function ensures that all the coefficients are reconstructed precisely. Table 5.2 shows the processing time and PSNR performance of the proposed technique with the other two techniques. The values are found using MATLAB image processing software. The system configuration used for this experiment is given in Table 6.2. The image used is a 512 x 512 grayscale image processing standard image ‘Lena’. The PSNR value between the original and reconstructed image is calculated at the bit rate of 1.0bpp (at 87.5% compression). From Table 5.2, it is evident that the proposed quantization algorithm is much faster than EZW and SPIHT, but with a small loss in PSNR.

Table 5.1: System Configuration

Processor	Intel Core i3 @ 2.27GHz
Memory	4GB RAM
GPU	1GB ATI Radeon Graphics
Operating System	Windows 7
Simulation Software	Matlab Professional (version 7.14)

Table 5.2: Performance of the proposed coding algorithm compared with other algorithms.

Integer D6 LWT	Proposed	EZW	SPIHT
Processing time (secs)	4.16	641.21	16.08
PSNR (dB)	31.35	31.41	31.86

5.5 Run-Length Encoding (RLE)

Run length encoding is based on the idea of encoding a consecutive occurrence of the same symbol. This is achieved by replacing a series of repeated symbols with a count and the symbol. In the case of the proposed quantization, there will be a lot of zeros and ones that can be exploited with RLE. The following example illustrates the concept of using RLE to encode input data with frequent runs of zeros or ones.

Illustration

Original (Random) data

1 1 0 0 0 1 1 1 0 0 0 0 0 0 1 1 1 1 1 0 1 1 1 1

Encoded data

2 1 3 0 3 1 6 0 5 1 2 1 0 4 1

In general, the run length encoding can be written as: (run, value). RLE is a simple technique that is used to represent the redundant symbols. If the data contains large number of zeros or ones consecutively occurring in the bitstream, then it can be greatly reduced using the RLE. All the run values are then directly converted to their corresponding binary values and transmitted across the channel.

Chapter 6

Experimental Results and Discussion

6.1 Introduction

In this chapter, we discuss the results of the AIQ based Daubechies-4 and Daubechies-6 lifting wavelet transform along with the proposed quantization and compare its performance with other existing methods. Then, we will proceed with the hardware implementation of the proposed transform algorithm. Moreover, the result obtained from FPGA synthesis is compared with other transform schemes to measure the complexity of the proposed AIQ architecture.

6.2 Analysis of Image Using Wavelet Transform

The multi-resolution decomposition technique allows wavelets to decorrelate an image and concentrate the energy in few coefficients. The energy distribution changes with the number of decomposition levels. As the decomposition level increases, the number of approximation coefficients decreases, but energy of the coefficients in the low pass subband also decreases. Therefore, energy compaction in an image is the amount of information that is self-contained in the transform coefficients.

To select the right decomposition levels to be used in this research, we use the most commonly used standard image processing grayscale image 'Lena'. The grayscale image is of size 512 x 512 and constitutes 8 bit per pixels. The transform used for calculating energy is the Daubechies-4 wavelet transform. The wavelet transform analyzes the image in Figure 6.1(a) by decorrelating the high pass in the image from the low pass. In doing so, the wavelet transform is able to

concentrate most of the information about the image to a few coefficients. Figure 6.1(b) shows the output of the image after first-level decomposition. This image is divided into 4 subbands, mainly, LL (top-left), HL (top-right), LH (bottom-left) and HH (bottom-right). The approximated version of the image is found in the LL subband while the details of the image are found in the rest of the subbands. Table 6.1 shows that the LL region contains the most information about the image having the energy of 99% while the remaining 1% is distributed in the LH, HL and HH subbands for the first level decomposition. This energy distribution suggests that the values of the coefficients in the LL region are significant while the values of the coefficients in the other detail subbands are insignificant.

In order to contain so much information in a small region (LL), the magnitudes of the wavelet coefficients increase. This can be shown by the increase in the range of the wavelet coefficients in Figure 6.1(f). Besides that, a large number of insignificant coefficients in the detailed subbands are found in the zero or near the zero region.

Table 6.1: Energy compaction of the subbands at various decomposition levels

Daubechies-4 DWT	LL Subband (energy in %)	HL Subband (energy in %)	LH Subband (energy in %)	HH Subband (energy in %)
1-Level	99.8545	0.0386	0.0905	0.0164
2-Level	99.5258	0.1244	0.2911	0.0588
3-Level	98.8852	0.2611	0.7026	0.1510
4-Level	97.8704	0.4658	1.3842	0.2795
5-Level	96.2421	0.7333	2.5170	0.5077

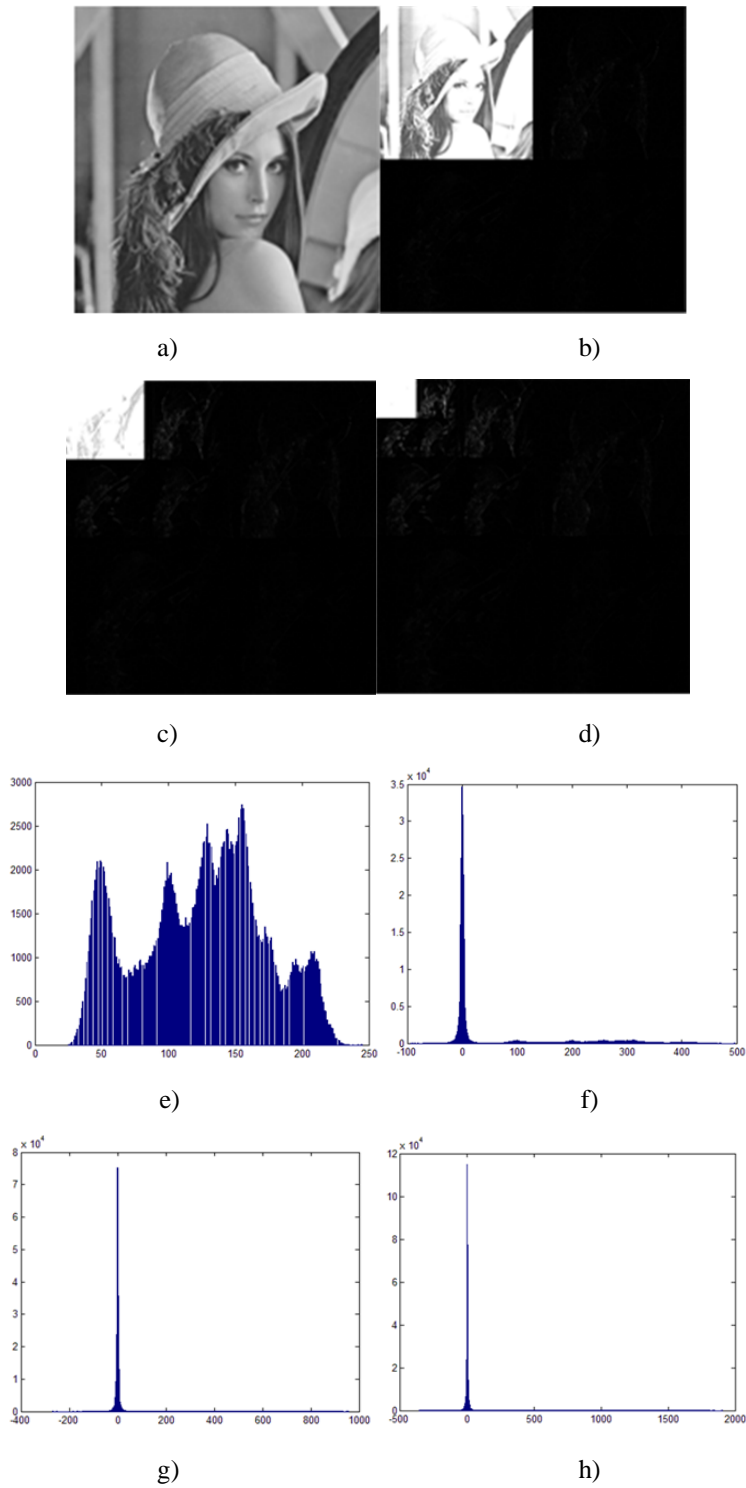


Figure 6.1: Analysis of decomposed image at different levels: a) original image, b) 1-level decomposed image, c) 2-level decomposed image, d) 3-level decomposed image, e) histogram of original image f) histogram of 1-level decomposed image, g) histogram of 2-level decomposed image and h) histogram of 3-level decomposed image.

As the decomposition level increases, the range of the wavelet coefficients increases and a larger number of insignificant coefficients are formed near the zero regions, as shown in Figure 6.1(h). However, there is a drop in the energy level for the LL sub-images as shown in Table 6.1. But in the case of HL, LH and HH subbands, the energy levels increase steadily. The total energy for each decomposition decreases with increase in decomposition level. This is significant especially for level 4 onwards.

However, the reason for the drop in the total energy is due to the Heisenberg Inequality, which states that, it is impossible to localize a fixed amount of energy to an arbitrary small time interval [31]. This explains the leaks in the energy as the decomposition level increases. Thus, we choose 3-level decomposition for our proposed AIQ architecture. This is to ensure that the image has sufficient decorrelation and at the same time to preserve as much energy as possible.

6.3 Simulation Results

The performance of the proposed AIQ algorithm is analyzed by simulating the algorithm in MATLAB 7.14. The algorithm is tested for compression efficiency and quality of reconstructed image. The images used for this analysis are shown in Figure 6.2 (5 Standard images (size 512 x 512)), Figure 6.3 (2 CT images (size 256 x 256)) and Figure 6.4 (10 Endoscopic images (size 256 x 256)).



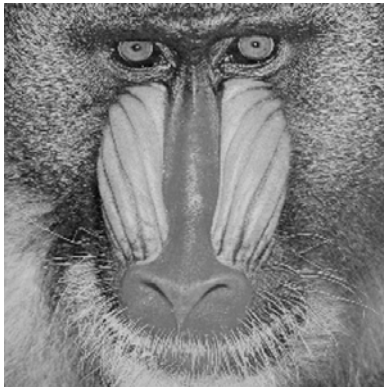
Lena



Barbara



Camerman



Mandrill

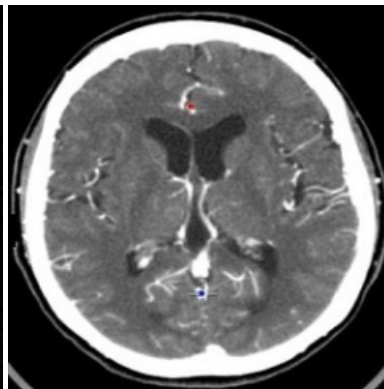


Peppers

Figure 6.2: Standard Images.



CT image 1



CT image 2

Figure 6.3: CT Images.



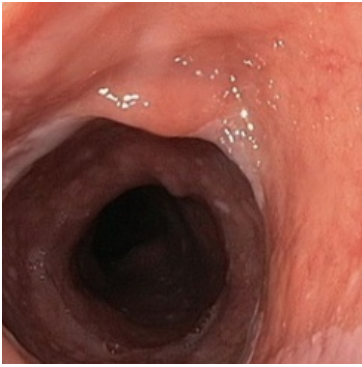
1



2



3



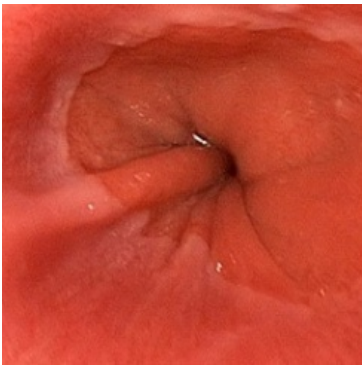
4



5



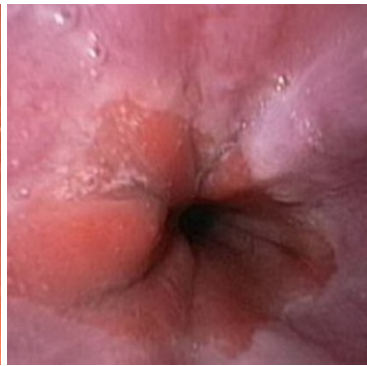
6



7



8



9



10

Figure 6.4: Endoscopic Images

The images are decomposed to three levels using AIQ based Daubechies-4 and AIQ based Daubechies-6 transform. The image is compressed using the proposed quantization technique and the bit rate is calculated to see the compression performance. The percentage of compression can be calculated in terms of Compression Ratio as given by,

$$\text{Compression Ratio} = \left(1 - \frac{\text{acquired bit rate}}{\text{original bit rate (8 bpp)}} \right) \times 100\% \quad (6.1)$$

Table 6.2 and Table 6.3 shows the PSNR comparison for the two transforms. Also, Figure 6.5 shows the comparison of image ‘Lena’ for the proposed two architectures at various bit rates. The proposed integer based algorithm is no different from the conventional floating point algorithm in terms of visual quality. The objective is to analyze the performance of the proposed algorithm with the conventional lifting based transform. From Table 6.2, it is clear that the performance of the integer based architecture is almost similar to the conventional lifting scheme. In some case, the proposed algorithm out performs the conventional lifting transform. Overall, the proposed AIQ based Daubechies-6 LWT performs better than proposed AIQ based Daubechies-4 LWT.

From Table 6.3 we can note that, the algorithm works really well for endoscopic images. For the bit rate 1.0bpp (Compression ratio of 87.5%) the average PSNR of all the images is >35dB which is more than the satisfactory value needed for endoscopic images [32]. The proposed algorithm is well suited to handle wide range of images.

Table 6.2: PSNR (dB) results for standard and CT images

Bit rate (bpp)	4	2	1	0.5	0.25
Barbara					
Proposed AIQ D4	35.28	30.45	26.96	24.71	23.07
Proposed AIQ D6	35.94	31.19	27.48	25.1	23.13
Conventional D6	36.09	31.21	27.55	25.08	23.19
Lena					
Proposed AIQ D4	39.03	35.83	32.55	29.36	26.29
Proposed AIQ D6	39.02	36.43	33.19	29.96	26.84
Conventional D6	39.1	36.46	33.16	29.92	26.7
Mandrill					
Proposed AIQ D4	32.37	28.38	25.46	23.45	22.04
Proposed AIQ D6	33.22	29.01	25.94	23.71	22.08
Conventional D6	33.27	28.98	25.93	23.66	22.06
Peppers					
Proposed AIQ D4	37.02	34.21	32.1	29.59	26.64
Proposed AIQ D6	37.29	34.33	32.48	29.95	27.11
Conventional D6	37.23	34.34	32.39	29.97	27.08
Camera Man					
Proposed AIQ D4	41.51	38.26	33.79	30.06	26.7
Proposed AIQ D6	40.91	38.5	34.29	30.58	27.22
Conventional D6	42.03	39.15	34.42	30.67	27.26
CT image 1					
Proposed AIQ D4	35.63	30.77	26.91	23.66	20.73
Proposed AIQ D6	35.31	31.02	27.3	24.05	21.44
Conventional D6	35.99	31.31	27.44	24.19	21.33
CT image 2					
Proposed AIQ D4	34.98	30.22	26.32	23.26	20.58
Proposed AIQ D6	35.24	30.75	27.05	23.8	21.02
Conventional D6	35.28	30.73	27.11	23.84	20.97



Original Image



4.0bpp, 39.03dB 2.0bpp, 35.83dB 1.0bpp, 32.55dB 0.5bpp, 29.36dB 0.25bpp, 26.29dB



4.0bpp, 39.02 dB 2.0bpp, 36.43 dB 1.0bpp, 33.19 dB 0.5bpp, 29.96 dB 0.25bpp, 26.84dB



4.0bpp, 39.1 dB 2.0bpp, 36.46 dB 1.0bpp, 33.16 dB 0.5bpp, 29.92 dB 0.25bpp, 26.7dB

Figure 6.5: Comparison of image 'Lena' at different bitrates and their corresponding PSNRs. The images are arranged according to results as seen in Table 6.2.

Table 6.3: Resulting PSNR (dB) for Endoscopic images

Bit rate (bpp)	4	2	1	0.5	0.25
Image 1					
Proposed AIQ D4	42.28	40.11	37.36	33.93	30.1
Proposed AIQ D6	43.05	41.13	38.23	34.2	30.18
Image 2					
Proposed AIQ D4	40.89	40.03	37.76	35.05	31.42
Proposed AIQ D6	41.91	40.17	37.97	35.36	31.81
Image 3					
Proposed AIQ D4	38.3	37.25	35.44	33.07	30.5
Proposed AIQ D6	38.83	37.32	35.49	33.4	30.44
Image 4					
Proposed AIQ D4	37.74	36.83	34.51	31.7	29.53
Proposed AIQ D6	39.06	36.91	34.6	32.14	29.6
Image 5					
Proposed AIQ D4	38.19	37.18	35.45	33.55	31.65
Proposed AIQ D6	39.12	37.42	35.52	33.61	31.71
Image 6					
Proposed AIQ D4	39.79	39.07	36.51	33.68	30.56
Proposed AIQ D6	40.83	39.44	36.69	34.22	30.55
Image 7					
Proposed AIQ D4	38.28	36.94	35.46	34.02	32.45
Proposed AIQ D6	38.51	37.02	35.47	34.18	32.47
Image 8					
Proposed AIQ D4	41.28	40.18	37.63	34.76	32.41
Proposed AIQ D6	42.67	40.44	38.01	35.02	32.62
Image 9					
Proposed AIQ D4	40.79	40.05	37.39	34.91	32.19
Proposed AIQ D6	42.04	40.18	37.73	35.13	32.48
Image 10					
Proposed AIQ D4	40.84	39.93	37.45	35.18	32.24
Proposed AIQ D6	42.91	40.2	38	35.54	32.96

From Table 6.4, it is evident that the proposed algorithm is faster than EZW and SPIHT with almost similar PSNRs. Though our quantization method is adopted from EZW, the processing time is greatly reduced with minimal loss of quality. In addition, we use a multistage replicate function which plays a major role in the improvement of PSNR and visual quality.

Table 6.4: Performance of the proposed quantization calculated for the image ‘Lena’.

Integer D6 LWT	Proposed	EZW	SPIHT
Processing time (secs)	4.16	641.21	16.08
PSNR (dB)	31.35	31.41	31.86

The use of the replicate function is to improve the quality of the image edges. The addition of the redundant bits does not cause huge difference in the calculated bit rate because the run length coding codes all the redundant bits into fewer bits. Let us compare the PSNR values for image ‘Lena’ from Table 6.2 (calculated with replicate function) with Table 6.4 (calculated without replicate function). So, the PSNR for the proposed AIQ based D6 LWT with replicate is 33.40dB while the PSNR for the one without using replicate is 31.35dB. The loss of quality can be seen from Figure 6.6. The images are reconstructed at 1.0 bpp using AIQ based D6 LWT. Figure 6.6(b) shows the image reconstructed using replicate function and Figure 6.6(c) shows the image reconstructed without using replicate function. It is clear that, the image edges are not reconstructed properly, which explains the decrease in PSNR.

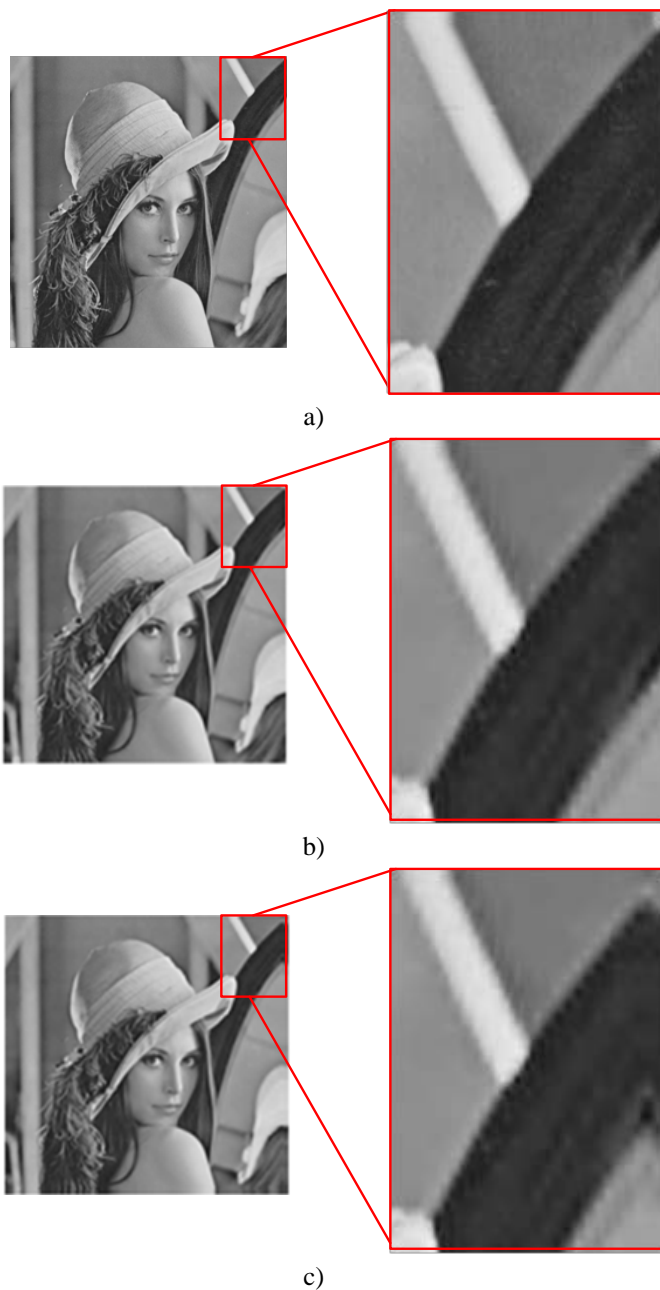


Figure 6.6: a) Original Image 'Lena', b) Reconstructed Image using AIQ D6 LWT with replicate, and c) Reconstructed Image using AIQ D6 LWT without replicate

Now that we have a perfect integer based lifting algorithm, the next step is to analyze the complexity of the proposed algorithm on hardware. The implementation of the proposed algorithm provides an estimate of the area and resource utilization on hardware. The next section

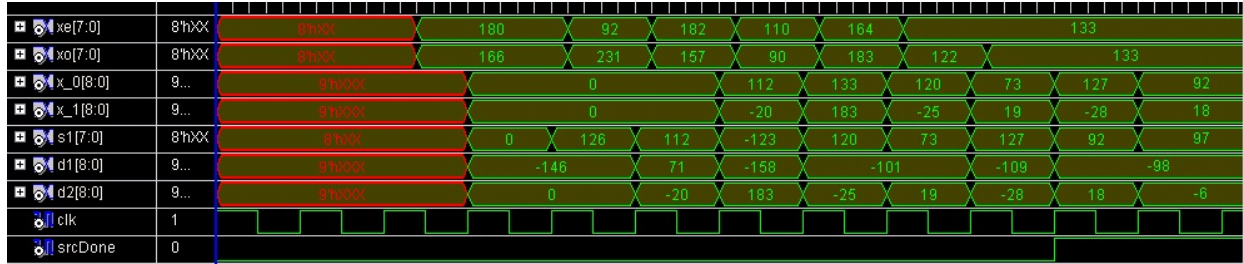
explains in detail, the implementation and performance of the proposed transform algorithm compared to other well known architectures.

6.4 FPGA Synthesis Results

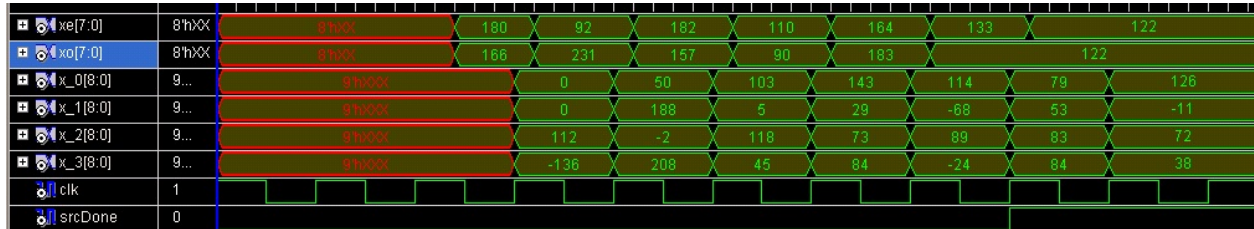
Field Programmable Gate Arrays [33] popularly known as FPGAs is an alternative for implementation of digital logic in systems. They are prefabricated silicon chips that can be programmed electrically to implement any digital design.

The proposed implementation is done on Xilinx ISE using Verilog as the HDL. The reason for selecting Xilinx is to provide the ease of comparison with other research methods. The design is synthesized on a high performance FPGA: Chip family: Virtex-E, Device name: XCV300E, Model: PQ240. Initially, the Verilog HDL code generated for the proposed transform algorithm is simulated using the Xilinx ISE's simulation tool. The output decomposed coefficient values are compared to the output from the MATLAB simulation. This ensures that the HDL code created produces correct results.

Figure 6.7 shows the output waveform obtained by extracting the HDL code using a test bench. x_e and x_o are the two inputs to the transform, while x_0 and x_1 are the corresponding low pass and high pass output.



a)



b)

Figure 6.7: Output waveform: a) the proposed AIQ D4 LWT, b) the proposed AIQ D6 LWT.

Table 6.5 shows the comparison of our results with other architectures in terms of hardware cost (multipliers, adders, registers) and throughput. Compared to the other methods, the proposed two transform algorithms have the lowest hardware cost. When compared to the conventional D6 lifting scheme, having 6 multipliers, the proposed AIQ D6 has only 4 multipliers. Also, the proposed algorithm has high throughput rate. This is mainly due to the nature of the discrete wavelet transform. Registers are used to store the input values and then processed together. But in case of the lifting scheme the calculations are in-place, i.e., it takes two inputs and outputs two coefficients (approximation and detail). So, from Table 6.5 we can note that the proposed lifting scheme only uses 4 registers. The inputs are not delayed as in other methods, but propagated immediately to the predict and update steps. This is real time processing capability of the lifting scheme.

In addition, most architectures use kernel sizes (8 x 8, 16 x 16 and so on) to multiply the input with the filter coefficients. In the case of [35], multiplication of large kernel size is required

which makes the control circuitry complicated. The architecture in [42] uses 16 bit data for DWT coefficients and DWT input. The hardware cost comprises of twelve 16 bit adders, four 8 x 16 multipliers and twelve 16 x 16 multipliers. The hardware cost of our lifting based D4 and D6 is better than the more famous conventional 9/7 lifting based transform [44]. Despite the advantageous of the folded architecture of [37] and [38], both the transforms are based on floating point implementation which is well known to cause errors that are propagated to the output. The control complexity of the architecture in [37] is more than the traditional lifting scheme. The algorithm in [38] uses a 12 bit floating point value to represent the filter coefficients. Compared to these techniques, our AIQ based lifting scheme is purely integer based.

Table 6.5: Hardware comparison of the proposed AIQ architecture with other techniques.

Method	Scheme	Transform	Multipliers	Adders	Registers	Throughput Input/Output
[34]	DWT	D4, D6	8/12	6/10	--	--
[35]	Lifting	CDF-9/7	4	12	--	--
[36]	DWT	CDF-9/7	0	19	9	1/1
[37]	Lifting	CDF-9/7	2	4	10	1/1
[38]	Lifting	CDF-9/7	4	8	22	2/1
[39]	Lifting	CDF-9/7	4	8	4	2/1
[40]	Lifting	CDF-9/7	2	4	20	1/1
[41]	Lifting	CDF-9/7	4	8	28	2/1
[42]	DWT	D4	14	12	72	2/1
[43]	DWT	D4	16	12	164	2/1
[44]	Lifting	CDF-9/7	4	8	6	2/1
Proposed AIQ D4	Lifting	D4	3	4	4	2/1
Proposed AIQ D6	Lifting	D6	4	6	4	2/1

Table 6.6 shows the logic cell utilization of the proposed lifting based architecture on Virtex-E FPGA. The architecture in [41] is implemented on Stratix chip of Altera for FPGA. The architecture is a combination of lifting based 5/3 and 9/7. The output coefficient length is made long for the purpose of precision. Out of the 17 bits, 10 bits are allocated for whole number and 7 bits are allocated for partial number. The architecture in [45] and [46] are based on different transform. It is implemented on the Virtex-E family with the device name XCV300E. This gives a rough estimate of where our architecture stands in terms of resource utilization. The AIQ based architecture in [47] is based on the discrete wavelet transform of Daubechies-4 and Daubechies-6 transform. Compared to the previous works, our lifting based AIQ offers good performance and has very low hardware complexity. The Virtex-E XCV300E FPGA is comprised of a total of 6,912 logic cells, of which only 212 is occupied, constituting only about 3% of the total.

Table 6.6: Comparison of logic utilization of the proposed algorithm on hardware.

Method	Scheme	Transform	Logic Cells	Input Bit Length	Output Coefficient Length
[41]	Lifting Processor for JPEG2000	CDF-9/7 LWT	5820	8	17
[45]	Conventional	D8 DWT	1120	9	9
	Distributed Arithmetic	D8 DWT	748	9	9
[46]	Hard Router	D8 DWT	900	9	8
	Benkrid Architecture	D8 DWT	632	9	8
[47]	D4-FP	D4 DWT	603	8	9
	D4-AIQ	D4 DWT	279	8	8
	D6-FP	D6 DWT	819	8	8
	D6-AIQ-1	D6 DWT	857	8	8
	D6-AIQ-2	D6 DWT	765	8	8
Proposed	AIQ D4 LWT	D4 LWT	282	8	8
	AIQ D6 LWT	D6 LWT	212	8	8

Finally, from Table 6.6, we see that the proposed AIQ D4 lifting scheme, with a much simpler structure, occupies more number of logic cells than the D6. This is mainly due to bit lengths of the filter coefficients and intermediate values in the proposed D4 lifting. Also, the AIQ based D4 uses more 4-input LUTs than D6. Table 6.7 shows the detailed synthesis report comparison of the proposed two transforms. It is evident that the D4 uses slightly more resources than D6.

Table 6.7: Comparison between two proposed architectures based on advanced HDL synthesis report.

Resources	Proposed AIQ D4 LWT	Proposed AIQ D6 LWT
Multipliers	8x9-bit(1), 9x8-bit(1), 9x9-bit(1)	8x6-bit(2), 8x7-bit(1), 8x9-bit(1)
Adders	17-bit(1), 19-bit(2), 9-bit(1)	13-bit(2), 15-bit(1), 16-bit(3)
Comparators	2-bit(1)	--
4-input LUTs	234	174
Maximum Frequency	163.292MHz	32.368MHz

Furthermore, the conventional lifting based D6 has a greater hardware cost, but the performance is better in terms of PSNR and visual quality. However, the AIQ based D6 uses lesser resources on hardware yet performs better in terms of PSNR. When compared to the proposed D4 transform, the proposed D6 lifting scheme has a superior architecture. The AIQ based D6 lifting scheme can reconstruct images with finer details. So, it can be used for applications that requires high accuracy (e.g., fingerprint recognition, ECG signal denoising) and high visual quality (e.g., Medical Endoscopic Imaging).

Chapter 7

Conclusion and Future Work

7.1 Summary and Accomplishments

In this thesis, we have presented algebraic integer based computation of Daubechies-4 (D4) and Daubechies-6 (D6) lifting scheme. The AIQ based D6 has lesser number of filter coefficients than the conventional approach. Later, we have also proposed an adaptive quantization technique to compress the wavelet coefficients. The proposed quantizer has an iterative coding structure and has a high compression performance at faster processing time, due to individual subband coding. The quantization followed by the Run Length Encoding (RLE) ensures that the desired low bit rate is achieved while maintaining good image reconstruction. The image compressor (transform, quantization and coding) codes the input image into series of binary values and inserts the data into the bitstream. The image compressor was simulated in Matlab image processing tool. Three-level wavelet decomposition was used on various images to determine the performance of the two algorithms. Overall, the proposed D6 handled all the images effectively and reconstructed images with good PSNR and visual quality at high compression rates (at 87.5% compression rate, the PSNR of the endoscopic images are $>35\text{dB}$, while the PSNR of the benchmark image 'Lena' is 33.40dB).

Next, the two proposed algorithms were tested for the complexity on hardware. The HDL code of the two transform architectures were implemented on Virtex-E XCV300E PQ240 FPGA using the Xilinx ISE tool. The synthesized results were compared with the other existing methods in terms of hardware cost and resource utilization. Overall, the two proposed techniques proved to

be better than the other techniques. Moreover, the proposed AIQ based D6 LWT used less hardware resources (logic cells: 212 and LUTs: 174) than the proposed D4 transform (logic cells: 282 and LUTs: 234).

To summarize, the proposed AIQ based Daubechies-6 lifting wavelet transform is less complex on hardware and also efficient in handling detailed images. Even at high compression rates, the quality of the reconstructed images is excellent when compared to other integer based transform algorithms. Compared to the conventional lifting based 9/7 transform (the one used in JPEG2000 standard), the proposed AIQ based D6 uses very little resources on the hardware.

7.2 Recommendations for future work

The future work needs to be targeted towards reducing the critical path delay and VLSI fabrication of the proposed AIQ based Daubechies-6 Lifting wavelet transform. Some recommendations for improvements include,

- If the latency due to multiplication is T_m and the latency due to addition is T_a , then the critical path latency of the proposed AIQ based D4 LWT is $3T_m+4T_a$, while that of the proposed AIQ based D6 is $4T_m+6T_a$.
- The critical path delay can be reduced by implementing a folded lifting structure and using pipelining to improve the flow.
- The folded architecture also reduces the number of multipliers by switching between two multipliers.

- Also, it would be very useful to investigate different factorizations of the polyphase matrix to implement the transforms.

References

1. M. Vetterli and C. Herley, "Wavelets and filter banks: theory and design", IEEE Transactions on Signal Processing, vol. 40, no. 9, pp. 2207-2232, 1992.
2. M.L. Hilton ,Bjorn D. Jawerth, Ayan Sengupta, "Compressing Still and Moving Images with Wavelets", Multimedia Systems, vol. 2, no. 5, 1994.
3. S. Mallat, A wavelet tour of signal processing, New York: Academic, 1998.
4. I. Daubechies, "The Wavelet Transform Time-Frequency Localization and Signal Analysis", IEEE Transactions on Information Theory, vol. 36, no. 5, pp: 961-1005, 1990.
5. A .Shahbahrami, "Improving the performance of 2D Discrete Wavelet Transform using data-level parallelism", International conference on High Performance Computing and Simulation (HPCS), pp: 362 – 368, 2011.
6. Chih-Hsien Hsia, Jing-Ming Guo, Jen-Shiun Chiang "Improved Low-Complexity Algorithm for 2-D Integer Lifting-Based Discrete Wavelet Transform Using Symmetric Mask-Based Scheme Chiang", IEEE Transactions on Circuits and Systems for Video Technology, vol. 19, no.8, pp: 1202 – 1208, 2009.
7. N. Ahmed, T. Natarajan and K. Rao, "Discrete Cosine Transform", IEEE Transactions on Computers, vol. C-23, no.4, pp: 90-93, 1974.
8. G. K. Wallace, "The Jpeg still Image Compression Standard", IEEE Transactions on Consumer Electronics, vol. 38, no.4, pp: xviii - xxxiv,1992.
9. T. Acharya and Ping-Sing Tsai, JPEG2000 Standard for image Compression, 2004.
10. Amara Graps, "An Introduction to Wavelets", IEEE Computational Science and Engineering, vol.2, no.2, pp: 51-60, 1995.
11. J. Morlet, G. Arens, E. Fourceau and D. Giard " Wave propagation and sampling theory, Part1: Complex signal land scattering in multilayer media", vol.47, no.2, pp: 203-221, 1982.
12. A. Grossmann and J.Morlet , " Decomposition of Hardy functions into square integrable wavelets of constant shape", SIAM Journal of Mathematical Analysis, vol.15, no.4, pp: 723-736, 1984.
13. <http://engineering.rowan.edu/~polikar/WAVELETS/WTtutorial.html>; the Wavelet Tutorial by Robi Polikar.
14. C. Rafael C. Gonzalez, and Richard E. Woods, Digital Image Processing, New Jersey: Pearson Prentice Hall, Third Edition, 2008.

15. S. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol.1, no.7, pp: 674-693, 1989.
16. Uwe Meyer-Baese, *Digital Signal Processing with Field Programmable Gate Arrays*, Springer-Verlag, 2001.
17. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps", *Journal of Fourier Analysis and Applications*", vol. 4, no. 3, pp: 245-267, 1998.
18. W. Sweldens, "The Lifting Scheme: A construction of second-generation wavelets", *SIAM Journal of Mathematical Analysis*, vol. 29, no. 2, pp: 511-546, 1997.
19. L. Cheng, D.L. Liang and Z.H. Zhang "Popular biorthogonal wavelet filters via a lifting scheme and its application in image compression", *IEEE Proceedings Vision, Image and Signal Processing*, vol. 150, no. 4, pp: 227–232, 2003.
20. K.A. Kotteri, S. Barua, A.E. Bell, and J.E Carletta, "A comparison of hardware implementations of the biorthogonal 9/7 DWT: convolution versus lifting", *IEEE Transactions on Circuits and Systems II*, vol. 52, no. 5, pp: 256–260, 2005.
21. Hongyu Liao "Novel Architectures for Lifting Based Discrete Wavelet Transform", *Proceedings of IEEE CCECE on Electrical and Computer Engineering*, vol. 2, pp:1020-1025, 2002.
22. I. Daubechies. "Orthonormal bases of compactly supported wavelets", *Communication on Pure and Applied Mathematics*, vol. 41, no. 7, pp: 909–996, 1988.
23. I. Daubechies, *Ten Lectures on Wavelets*, SIAM, Philadelphia, 1992.
24. Geert Uytterhoeven, Dirk Roose and Adhemar Bultheel, "Integer Wavelet Transforms using the Lifting Scheme", *CSCC Proceedings*, pp: 6251-6253, 1999.
25. J. H. Cozzens and L. A. Finkelstein, "Computing the Discrete Fourier Transform using Residue Number Systems in a Ring of Algebraic Integers", *IEEE Transactions on Information Theory*, vol. 31, no. 5, pp:580-588, 1985.
26. K. A. Wahid, V. S. Dimitrov, and G. A. Jullien, "Error-Free Arithmetic for Discrete Wavelet Transforms using Algebraic Integers", *Proceedings of the IEEE Symposium on Computer Arithmetic*, pp: 238-244, 2003.
27. J. Eric Balster, B.T. Fortener and W.F. Turri "Integer Computation of Lossy JPEG2000 Compression", *IEEE Transaction on Image Processing*, vol. 20, no. 8, pp: 2386–2391, 2011.

28. C.T. Huang P.C. Tseng, and L.G. Chen, "Flipping structure: An efficient VLSI architecture for lifting-based discrete wavelet transform", IEEE Transactions on Signal Processing, vol. 52, no. 4, pp: 1910–1916, 2004.
29. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients", IEEE Transactions on Signal Processing, vol. 41, no.12, pp: 3445-3462, 1993.
30. A. Said and W. Pearlman, "A new, fast and efficient image codec based on set partitioning", IEEE Transactions on Circuits and Systems for Video Technology, vol.6, no.3, pp: 243-250, 1996.
31. J.S. Walker, A Primer on Wavelets and their Scientific Applications, Chapman & Hall/CRC, 1999.
32. R. Istepanian, N. Philip, M. Martini, N. Amso and P. Shorvon, "Subjective and objective quality assessment in wireless teleultrasonography imaging", IEEE Engineering in Medicine and Biology Society, pp: 5346 – 5349, 2008.
33. <http://coep.vlab.co.in/?sub=29&brch=88&sim=228&cnt=1>; Sakshat Virtual Labs.
34. Q. Dai, X. Chen and C. Lin, "A novel VLSI architecture for multidimensional discrete wavelet transform", IEEE Transactions on Circuits and Systems for Video Technology, vol. 14, no. 8, pp: 1105-1110, 2004.
35. S. Lee and S. Lim, "VLSI design of a wavelet processing core", IEEE Transactions of Circuits and Systems for Video Technology, vol. 16, no. 11, pp: 1350-1360, 2006.
36. M. Martina and G. Masera, "Multiplierless, folded 9/7-5/3 wavelet VLSI architecture", IEEE Transactions on Circuits and Systems II, vol .54, no. 9, pp: 770-774, 2007.
37. G. Shi, W. Liu, L. Zhang, F. Li , "An efficient folded architecture for lifting based discrete wavelet transform ", IEEE Transactions on Circuits and Systems II, vol. 56, no. 4, pp: 290-294, 2009.
38. Y. Lai, L. Chen, Y. Shih, "A high-performance and memory-efficient VLSI architecture with parallel scanning method for 2-D lifting-based discrete wavelet transform", IEEE Transactions on Consumer Electronics, vol. 55, no. 2, pp: 400 – 407, 2009.
39. C. Huang, P. Tseng and L. Chen, "Flipping structure: an efficient VLSI architecture for lifting based discrete wavelet transform", IEEE Transactions on Signal Processing, vol.52, no.4, pp: 1080–1089, 2004.
40. B. Wu and C. Lin, "A high-performance and memory-efficient pipeline architecture for the 5/3 and 9/7 discrete wavelet transform of JPEG2000 codec", IEEE Transactions on Circuits and Systems for Video Technology, vol. 15, no. 12, pp: 1615-1628, 2005.

41. Y. Seo and D. Kim , "VLSI architecture of line-based lifting wavelet transform for motion JPEG2000", IEEE Journal of Solid-State Circuits, vol. 42, no. 2, pp: 431-440, 2007.
42. S. Paek and L. Kim, "2D DWT VLSI architecture for wavelet image processing", IEEE Electron Letters, vol. 34, no. 6 pp: 537–538, 1998.
43. K. Parhi and T. Nishitani, "VLSI architectures for discrete wavelet transforms", IEEE Transactions on Very Large Scale Integration Systems, vol. 1, no. 2, pp:191–202, 1993.
44. J. M. Jou, Y. H. Shiau and C. C. Liu, "Efficient VLSI architectures for the biorthogonal wavelet transform by filter bank and lifting scheme", IEEE International Symposium on Circuits and Systems, vol. 2, pp: 529–532, 2001.
45. M. Ali and Al-Haj, "Fast Discrete Wavelet Transformation Using FPGAs and Distributed Arithmetic", International Journal of Applied Science and Engineering, vol. 1, pp: 160-171, 2003.
46. K. Benkrid, A. Benkrid and D. Crookes, "A novel FIR filter architecture for efficient signal boundary handling on Xilinx VIRTEX FPGAs", Proceedings 11th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, pp: 273-275, 2003.
47. K.Wahid, V. Dimitrov and G. Jullien, "VLSI architectures of Daubechies wavelet transforms using algebraic integers ", J. Circuits Systems and Computer, vol. 13, no. 6, pp:1251–1270, 2004.
48. I. Daubechies, "Orthonormal bases of compactly supported wavelets," Communications on Pure and Applied Mathematics, vol. 41, no. 7, pp. 909-996, 1988.