"IMPLEMENTATION OF DISCRETE WAVELET TRANSFORM USING LIFTING STRUCTURE"

A Major Project Report

Submitted in Partial Fulfillment of the Requirements for
For the Degree
Of

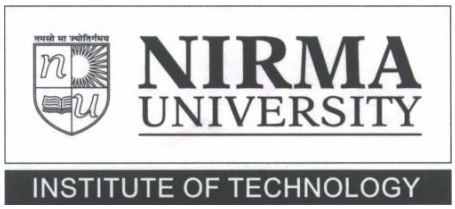
MASTER OF TECHNOLOGY
IN
ELECTRONICS & COMMUNICATION ENGINEERING

(VLSI Design)

By
G.Kiran Kumar
(04MEC008)

Project guide
Mr. N.V.Ramana
Station Engineer
Doordarshan Kendra
Ahmedabad

Facilitator at Institute
Prof N.P.Gajjar
Nirma University
Ahmedabad



Department of Electronics & Communication Engineering
INSTITUTE OF TECHNOLOGY
NIRMA UNIVERSITY OF SCIENCE & TECHNOLOGY,
AHMEDABAD 382 481

MAY 2005

CERTIFICATE

This is to certify that the Major Project Report titled" Implementation of Discrete Wavelet Transform using Lifting Structure" submitted by G.Kiran Kumar (Roll No. 04MEC008) as the partial fulfillment of the requirements for award of degree of M.Tech (EC-VLSI design) awarded by institute of technology, Nirma University, Ahmedabad embodies the work carried out by him under my supervision at Door Darshan Kendra, Ahmedabad w.e.f OCTOBER 2005-April 2006.

Date:

Place:

Signature of the Guide
Mr.N.V.Ramana
Section engineer
Doordarshan Kendra,
Ahmedabad.

CERTIFICATE

This is to certify that the Major Project Report entitled" Implementation of Discrete Wavelet Transform using Lifting Structure" submitted by G.Kiran Kumar (Roll No. 04MEC008) as the partial fulfillment of the requirements for the award of Degree of Master of Technology in Electronics and Communication (VLSI design) of Institute of Technology, Nirma University is the record of work carried out by him under my/our supervision and guidance. The work submitted has in our opinion reached a level required for being accepted for examination. The results embodied in this major project work to the best of our knowledge have not been submitted to any other University or Institution for award of any degree or diploma.

Date:
Place:

Facilitator at Institute
Prof. N.P.Gajjar
Nirma University
Ahmedabad

HOD
Dr. M. D. Desai
EC-Department
Nirma University
Ahmedabad

Director
Dr. H. V. Trivedi
Nirma University
Ahmedabad

ACKNOWLEDGEMENT

First and foremost, I express my gratitude to my guide Mr. N.V.Ramana, station engineer Doordarshan Kendra, Ahmedabad, for his guidance and support during my project work

I also thank Prof N.P.Gajjar, who always boosted me to do the things in a possible manner and giving us the path to gain sound knowledge in the field of VLSI design.

I also thank Dr.N.M.Devashrayee, M.Tech coordinator EC-VLSI design, Nirma University, Ahmedabad, for his guidance and valuable advice during my project work.

I also thank Prof Y.N.Trivedi who had helped me to select my dissertation topic and is constantly giving the valuable advises to explore the knowledge in the field of communication.

I also thank my colleagues and juniors for giving me the support and help constantly.

Thanks to my parents. Thanks to my family for all his faith. Finally, thanks to God, for helping me out at difficult times.

ABSTRACT

The project entitled “Implementation of DWT Using Lifting Scheme” is carried out as a part of M.Tech Dissertation at Doordarshan Kendra, Ahmedabad.

The project aims at providing the implementation of Discrete wavelet transform. wavelet transform is a essential tool which is used in JPEG-2000 standard .there are many implementation schemes which can be used for the implementation, like systolic array which is serial implementation and parallal implementation .In a broad way there are three implementation schemes convolutional, polyphase and the new one on which I am working is the lifting scheme .Basically lifting scheme has given the evolution of the second generation wavelets.

In JPEG-2000 they have standardized many filters, but the most useful filters are 5/3 lossless and 9/7 lossy filter .In this work I have worked on 5/3 lossless filter.
The implementation has been carried out in MATLAB.this whole structure has been implemented and have the feel of this,

Later on the reconfigurable architecture has been implemented in hardware and it has been compared with MATLAB values which I got from calculation and verified that implementation has been done at hardware also.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Motivation for thesis

A majority of today's Internet bandwidth is estimated to be used for images and video. Recent multimedia applications for handheld and portable devices place a limit on the available wireless bandwidth. The bandwidth is limited even with new connection standards. JPEG image compression that is in widespread use today took several years for it to be perfected. Wavelet based techniques such as JPEG2000 for image compression has a lot more to offer than conventional methods in terms of compression ratio. Currently wavelet implementations are still under development lifecycle and are being perfected. Flexible energy-efficient hardware implementations that can handle multimedia functions such as image processing, coding and decoding are critical, especially in hand-held portable multimedia wireless devices.

1.1.1 Aim

The project aims at implementing 5/3 wavelet filter on reconfigurable devices .This filter is crucial in standard for image compression such as JPEG-2000

Milestones

- Literature review and understanding 2D wavelet transform and implementation algorithms
- Understanding different schemes based on convolution and lifting structure
- MATLAB Implementation analysis and generation of test patterns
- Implementation of 5/3 algorithm on Xilinx based FPGAs

1.2 Background

The JPEG image compression standard is in wide spread use today. JPEG uses the Discrete Cosine Transform (DCT) as the transform, applied to 8-by-8 blocks of image data. The newer standard JPEG2000 is based on the Wavelet Transform (WT). Wavelet Transform offers multi-resolution image analysis, which appears to be well matched to the low level characteristic of human vision. The DCT is essentially unique but DWT

has many possible realizations. Wavelets provide us with a basis more suitable for representing images. This is because it can represent information at a variety of scales, with local contrast changes, as well as larger scale structures and thus is a better fit for image data.

Field programmable gate arrays (FPGAs) provide a rapid prototyping platform. FPGAs are devices that can be reprogrammed to achieve different functionalities without incurring the non-recurring engineering costs typically associated with custom IC fabrication. In this work, the target platform is the Xilinx Virtex FPGA.

The design is based on the multi-level decomposition implementation of the Discrete Wavelet Transform... Performance analysis includes the investigation of the performance enhancement due to the hardware acceleration. It is expected that the proposed design can substantially accelerate the DWT and the inherent scalability can be exploited to reach a higher performance in the future.

One drawback of the FPGA, however, is that due to the rather coarse grained reconfigurable blocks, an implementation on an FPGA is often not as efficient, in terms of space and time, as on a custom IC

1.3 Organization of Thesis

In this thesis, Chapter 2 gives a background discussion on wavelet transform based image compression. Chapter3 gives different implementations structures of DWT Chapter 4 gives the different architectures and analysis of different schemes and explains the implementation aspects of wavelet transform using lifting scheme. Chapter 5 gives the software and hardware implementations. Chapter 6 gives the software and hardware results Chapter 7 provides the conclusions and future extensions of work.

CHAPTER 2

WAVELET TRANSFORM BASED IMAGE COMPRESSION

Many evolving multimedia applications require transmission of high quality images over the network. One obvious way to accommodate this demand is to increase the bandwidth available to all users.

Of course, this "solution" is not without technological and economical difficulties. Another way is to reduce the volume of the data that must be transmitted. There has been a tremendous amount of progress in the field of image compression during the past 15 years.

In order to make further progress in image coding, many research groups have begun to use wavelet transforms. In this chapter, we will briefly discuss image compression, the nature of wavelets, and some of the salient features of image compression technologies using wavelets. Since this is a very rapidly evolving field, only the basic elements are presented.

2.1 Introduction

Image compression is different from binary data compression. When binary data compression techniques are applied to images, the results are not optimal. In lossless compression, the data (such as executables, documents, etc.) are compressed such that when decompressed, it gives an exact replica of the original data.

They need to be exactly reproduced when decompressed. For example, the popular PC utilities like WinZip or and Adobe Acrobat perform lossless compression. On the other hand, images need not be reproduced exactly.

A ‘good’ approximation of the original image is enough for most purposes, as long as the error between the original and the compressed image is tolerable. Lossy compression

techniques can be used in this application. This is because images have certain statistical properties, which can be exploited by encoders specifically designed for them. Also, some of the finer details in the image can be sacrificed for the sake of saving bandwidth or storage space. In digital images the neighboring pixels are correlated and therefore contain redundant information. Before the image is compressed, the pixels, which are correlated is to be found.

The fundamental components of compression are redundancy and irrelevancy reduction. Redundancy means duplication and irrelevancy means the parts of signal that will not be noticed by the signal receiver, which is the Human Visual System (HVS). There are three types of redundancy that can be identified:

Spatial Redundancy *is the correlation between neighboring pixel values.*

Spectral Redundancy *is the correlation between different color planes or spectral bands.*

Temporal Redundancy *is the correlation between adjacent frames in a sequence of images (in video applications).*

Image compression focuses on reducing the number of bits needed to represent an image by removing the spatial and spectral redundancies. Since our work focuses on still image compression, therefore temporal redundancy is not discussed.

2.2 Wavelet Transform as the Source Encoder

Just as in any other image compression schemes the wavelet method for image compression also follows the same procedure. The discrete wavelet transform constitutes the function of the source encoder. The theory behind wavelet transform is discussed below.

2.2.1 Measuring Frequency Content by Wavelet Transform

Wavelet transform is capable of providing the time and frequency information simultaneously. Hence it gives a time-frequency representation of the signal. When we are interested in knowing what spectral component exists at any given instant of time, we want to know the particular spectral component at that instant. In these cases it may be very beneficial to know the time intervals these particular spectral components occur. Wavelets (small waves) are functions defined over a finite interval and having an average value of zero. The basic idea of the wavelet transform is to represent any arbitrary function $f(t)$ as a superposition of a set of such wavelets or basis functions. These basis functions are obtained from a single wave, by dilations or contractions (scaling) and translations (shifts). The discrete wavelet transform of a finite length signal x(n) having N components, for example, is expressed by an N x N matrix similar to the discrete cosine transform .

2.2.2 Wavelet-based Compression

Digital image is represented as a two-dimensional array of coefficients, each coefficient representing the brightness level in that point. We can differentiate between coefficients as more important ones, and lesser important ones. Most natural images have smooth color variations, with the fine details being represented as sharp edges in between the smooth variations. Technically, the smooth variations in color can be termed as low frequency variations, and the sharp variations as high frequency variations.

The low frequency components (smooth variations) constitute the base of an image, and the high frequency components (the edges which give the details) add upon them to refine the image, thereby giving a detailed image. Hence, the smooth variations are more important than the details.

2.3 Wavelet

2.3.1 Continuous Wavelet Transform

The transform of a signal is just another form of representing the signal. It does not change the information content present in the signal. The Wavelet Transform provides a

time-frequency representation of the signal. It was developed to overcome the short coming of the Short Time Fourier Transform (STFT), which can also be used to analyze non-stationary signals.

While STFT gives a constant resolution at all frequencies, the Wavelet Transform uses multi-resolution technique by which different frequencies are analyzed with different resolutions.

A wave is an oscillating function of time or space and is periodic. In contrast, wavelets are localized waves. They have their energy concentrated in time or space and are suited to analysis of transient signals. While Fourier Transform and STFT use waves to analyze signals, the Wavelet Transform uses wavelets of finite energy.

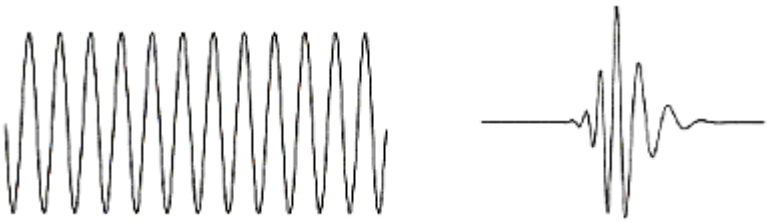


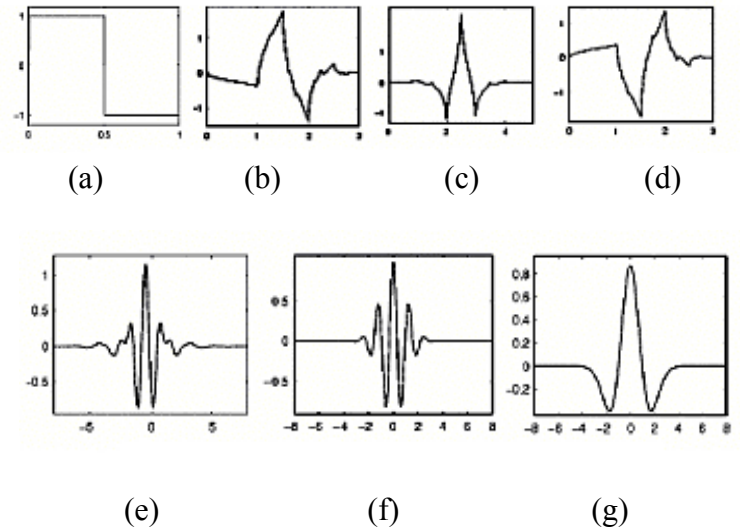
Figure 2.1 (a-left) wave; (b-right) wavelet

The wavelet analysis is done similar to the STFT analysis. The signal to be analyzed is multiplied with a wavelet function just as it is multiplied with a window function in STFT, and then the transform is computed for each segment generated. However, unlike STFT, in Wavelet Transform, the width of the wavelet function changes

With each spectral component. The Wavelet Transform, at high frequencies, gives good time resolution and poor frequency resolution, while at low frequencies. The Wavelet Transform gives good frequency resolution and poor time resolution

2.3.2 Wavelet Families

There are a number of basis functions that can be used as the mother wavelet for Wavelet Transformation. Since the mother wavelet produces all wavelet functions used in the transformation through translation and scaling, it determines the characteristics of the resulting Wavelet Transform. Therefore, the details of the particular application should be taken into account and the appropriate mother wavelet should be chosen in order to



(a) (b) (c) (d)

(e) (f) (g)

Figure 2.2 Wavelet families top (a-left) Haar; (b-next) Daubechies; (c-next) coflet; (d-right) symlet2; down (e-left) Meyer (f-next) Morlet; (g) Mexican hat

The following introduction on wavelets is based on the paper by mathematician Gilbert Strang [STRANG]. A wavelet is a localized function in time (or space in the case of images) with mean zero. A wavelet basis is derived from the wavelet (small wave) by its own dilations and translations.

CWT is provided by equation 2.1, where x(t) is the signal to be analyzed. ψ(t) is the mother wavelet or the basis function. All the wavelet functions used in the transformation are derived from the mother wavelet through translation (shifting) and scaling (dilation or compression).

$$X_{WT}(\tau,s) = \frac{1}{\sqrt{|s|}} \int x(t) \cdot \psi^{*}\left(\frac{t-\tau}{s}\right) dt \qquad \ldots\ldots\ldots\ldots\ldots\ldots \quad (2.1)$$

The mother wavelet used to generate all the basis functions is designed based on some desired characteristics associated with that function. The translation parameter τ relates to the location of the wavelet function as it is shifted through the signal.

Thus, it corresponds to the time information in the Wavelet Transform. The scale parameter *s* is defined as |1/frequency| and corresponds to frequency information. Scaling either dilates (expands) or compresses a signal. Large scales (low frequencies) dilate the signal and provide detailed information hidden in the signal, while small scales (high frequencies) compress the signal and provide global information about the signal. Notice that the Wavelet Transform merely performs the convolution operation of the signal

2.3.3 Discrete Wavelet Transform

The Discrete Wavelet Transform (DWT), which is based on sub-band coding, is found to yield a fast computation of Wavelet Transform. It is easy to implement and reduces the computation time and resources required.

$$DWT_x(k,j) = \left\langle x, \psi_{k,j} \right\rangle = 2^{\frac{-j}{2}} \sum_{n=-\infty}^{\infty} x(n)\psi\left(2^{-j}n - k\right) \quad \text{.....(2.2)}$$

As a result of the DWT a discrete set of wavelet coefficients is received. This coefficients measure the signal energy distribution in each frequency channel corresponding to the scaling parameter j at the time k. This interpretation of the DWT coefficients implies naturally the implementation using digital flters.

The foundations of DWT go back to 1976 when techniques to decompose discrete time signals were devised [5]. Similar work was done in speech signal coding which was named as sub-band coding. In 1983, a technique similar to sub-band coding was developed which was named pyramidal coding. Later many improvements were made to these coding schemes which resulted in efficient multi-resolution analysis schemes.

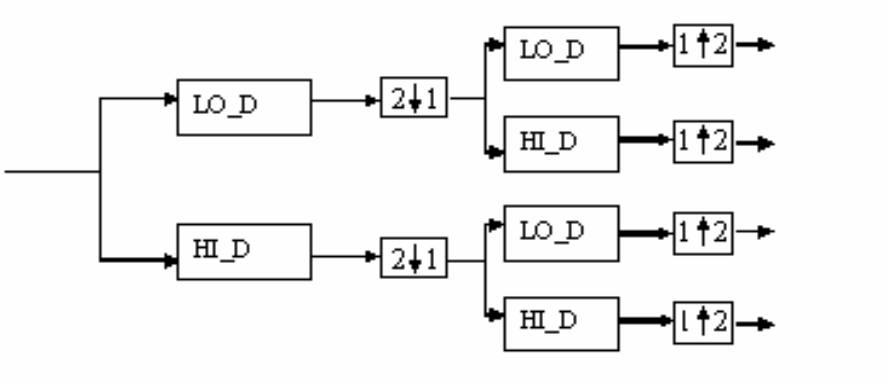


Figure 2.3: Block Diagram of FDWT

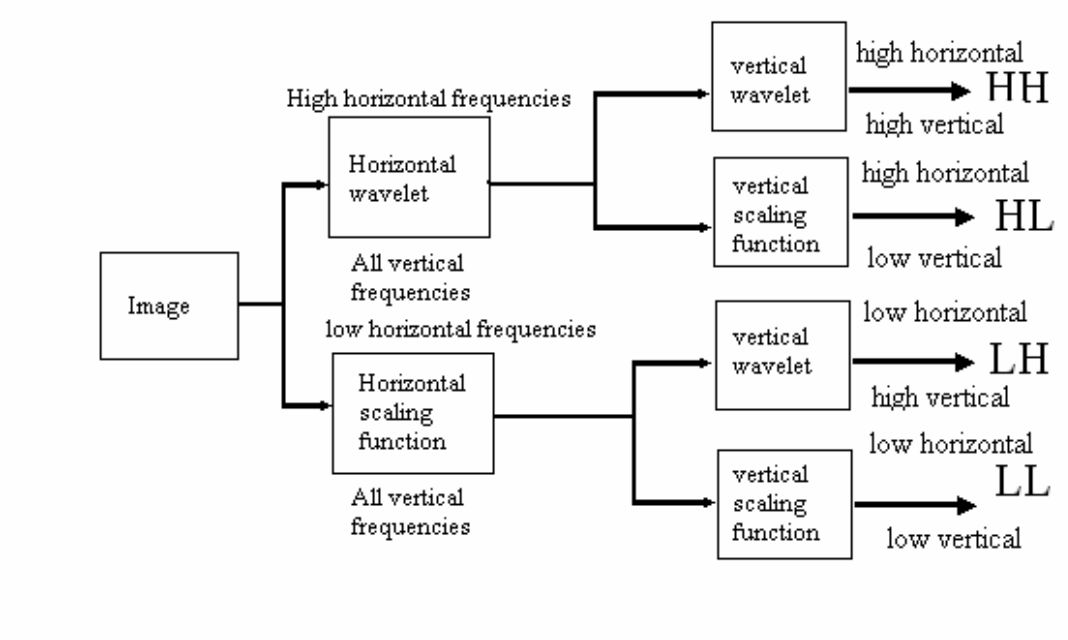


Figure 2.4 pyramidal Decomposition of image

2.3.4 Multi Resolution Analysis:

A function $f(x)$ is projected at each step j onto the subset V_j. This projection is defined by the scalar product $c_j(k)$ of $f(x)$ with the scaling function which is dilated and translated

$$C_j(k) = \left\langle f(x), 2^{-j}\phi(2^{-j}x - k)\right\rangle \text{.....(2.2)}$$

As is a scaling function which has the property:

$$\frac{1}{2}\phi\left(\frac{x}{2}\right) = \sum_n h(n)\phi(x-n) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.3)$$

At each step, the number of scalar products is divided by 2. Step by step the signal is smoothed and information is lost. The remaining information can be restored using the complementary subspace W_{j+1} of V_{J+1} in V_j. This subspace can be generated by a suitable wavelet function with translation and dilation.

$$\frac{1}{2}\psi(\frac{x}{2}) = \sum_n g(n)\phi(x-n) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.4)$$

The passage from a resolution to the next one is done by:

$$f_{j+1}(k_x, k_y) = \sum_{l_x=-\infty}^{+\infty} \sum_{l=-\infty}^{+\infty} h(l_x - 2k_x)h(l_y - 2k_y)\, f_j(l_x, l_y) \ldots\ldots (2.5)$$

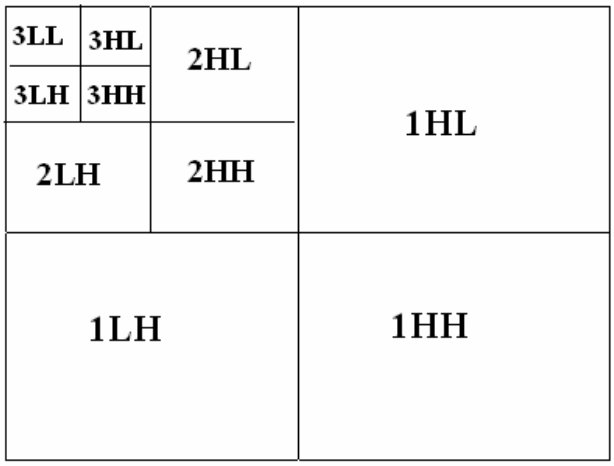


Figure 2.5 Multi Resolution Analyses

CHAPTER 3

WAVELET STRUCTURES AND ANALYSIS

3.1 JPEG-2000 Compression

The image compression is accomplished using JPEG2000 encoder, similar to most transform based coding schemes as shown in figure

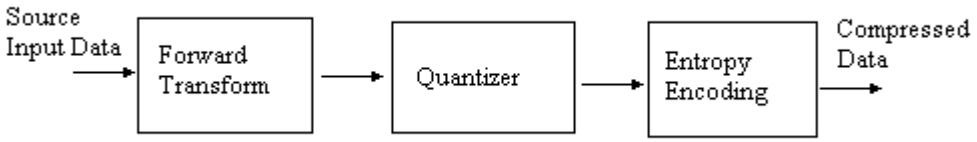


Figure 3.1 JPEG2000 Encoder Block Diagram

The Discrete Wavelet Transform (DWT) is first applied on source image data .the transform coefficients are then quantized and entropy coded before forming output .the decoder is reverse of encoder .unlike other coding schemes, JPEG2000 can be "lossy " or ' lossless " and depends on wavelet transform and quantization.

The image is decomposed using Discrete Wavelet Transform (DWT) into series of decomposition levels, each containing a number of sub-bands describe the horizontal and vertical characteristic of original image.

A 1-dimentional sub-band is decomposed into a set of low pass and high pass samples represent a smaller low resolution version of original .the high pass samples represent a smaller residual version of original and necessary for perfect reconstruction of original image from low pass samples

All of wavelet transform employing JPEG2000 compression methods are fundamentally 1D in nature. a one dimensional transform applied first in horizontal and then in vertical direction ,forms a two dimensional transform

Thus the original image decomposes into four smaller image blocks

- one with low vertical and low horizontal resolution (LL)
- one with high vertical and low horizontal resolution (HL)
- one with low vertical and high horizontal resolution (LH)
- One with high vertical and high horizontal resolution (HH).

This process of applying the one-dimensional transform in both the direction is repeated number of times on the low resolution image block. this procedure is called Dyadic decomposition

3.2 DWT Implementations

The previous section briefly discussed how the DWT of a digital image (signal) could be realized by means of filters and scaling .the image resolution (amount of detail) is determined by selecting the appropriating filtering operation while the scale can be controlled by up sampling or down sampling .

The JPEG2000 standard uses a none dimensional sub band decomposition of one dimensional set of samples into low pass and high pass samples in order to perform DWT.

The low pass samples represent down sampled residue of original dataset resolution .the high pass samples are necessary for perfect reconstruction of original image from low pass sample set .

A filter bank normally consists of low pass filter, a high pass filter decimators or expanders and delay elements. Based on these, the DWT can be implemented in several different forms.

3.2.1 Direct form Structure

The direct form structure (also known as convolution based) consist of low pass (G_0) and high pass (H_0) filters followed by decimators ,this is known as analysis stage figure2.5 (a) . the output of analysis stage is processed based on application. the original image can be recovered by synthesis stage figure 2.5(b) consist of up sampler followed by similar high pass(H_1) and low pass (G_1) filters as shown in diagram below.[2]

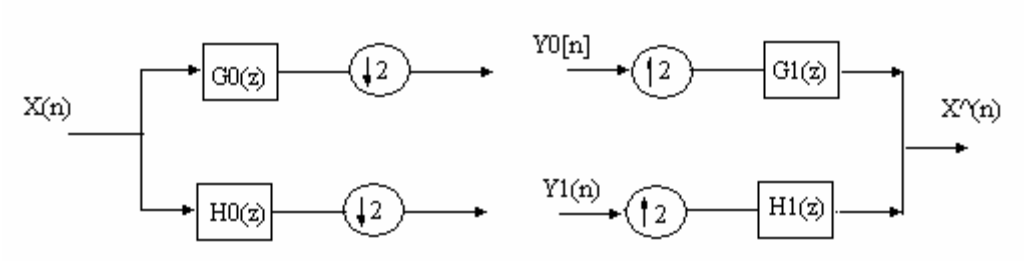


Figure 3.2: Direct Form Structure for one stage DWT

3.2.2 Polyphase structure

In the direct form structure discussed previously the image is first filtered and down sampled. Thus half of the filter samples are redundant- this is clearly inefficient .If the down sampling is done before filtering, it will reduce number of computations and thus improve efficiency (by 50%)

The input signal is split into odd and even samples (resulting an automatic decimation by a factor of 2), the filter coefficient are also split accordingly into even and odd component. The two phases are added after filtering to produce required high pass and low pass sample sets.

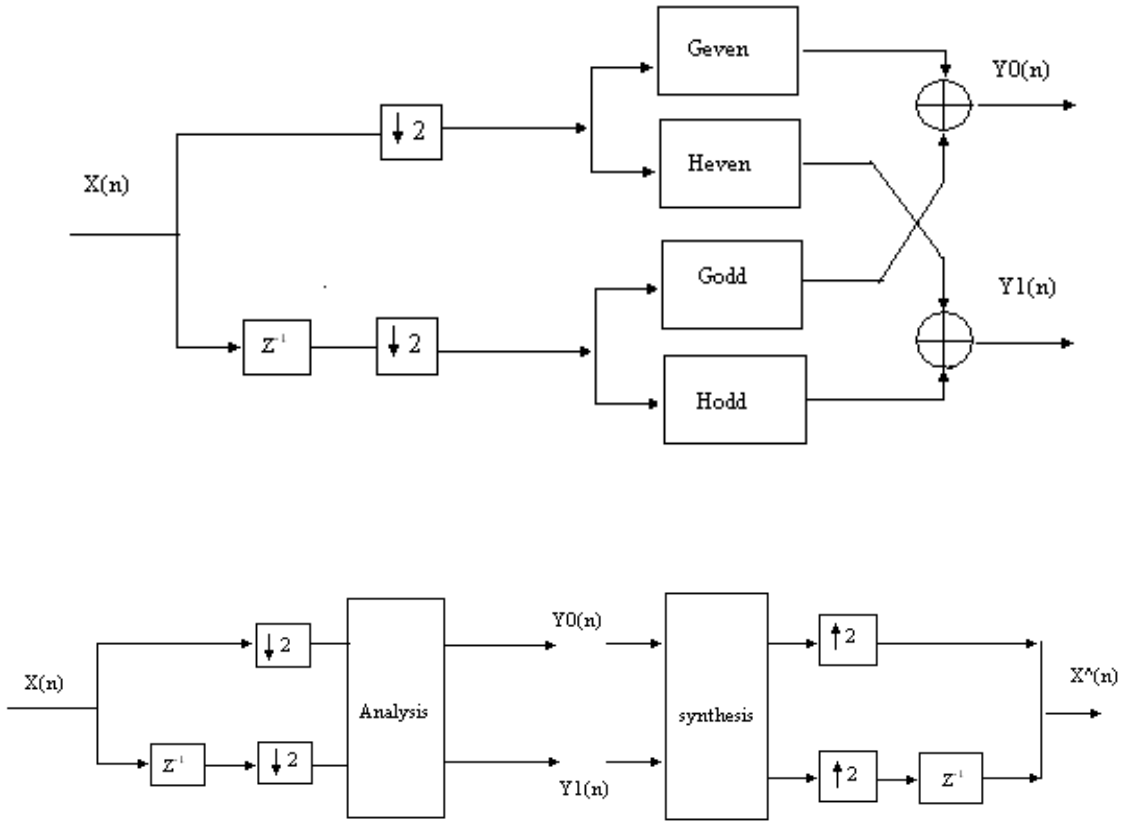


Figure 3.3 Polyphase Structure of (a) analysis filter bank (b) equivalent representation of analysis filter bank (c) synthesis filter bank

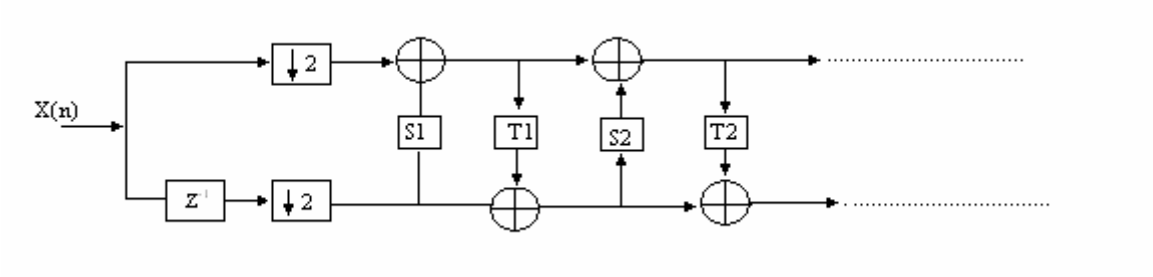
The synthesis is opposite of analysis and the filters comes first followed by up sampler again halving the number of computations in the filtering operations in addition since the odd and even terms filtered separately the filters can operate in parallel, further improving the efficiency

3.2.3 Lifting Based Structure

The lifting scheme was developed by Sweldens and allows for very efficient implementation of DWT using integer wavelet and scaling coefficients instead of floating point coefficients.

The source data is first split into even and odd samples then a predict step followed by update step performed .the predict step basically tries to predict the odd samples using even samples with a prediction operator. Assuming that consecutive statements are highly correlated (especially at the start of stream), this should be possible.

thus the high pass sub band is lifted using low pass sub band. The update step basically consists of lifting the low pass sub band using high pass sub band and is done in order to maintain the statistical properties of original input stream with the low pass sub band.



Figur3.4 lifting structure (generic)

A distinct advantage of lifting structure is that transformed data can take the same place as the input data .thus the transform can be performed without the need of additional

memory, very useful for embedded or SoC applications minimizing on expensive space .however it's limitation is that the filtering units cannot operate in parallel as each filtering unit depends on results from the previous filtering unit

3.3 Lifting Evolution

This section is giving the core of the lifting step, how it has been evolved from the polyphase structure to lifting scheme over the years to yield the benefits in terms of VLSI implementation of the DWT[1].

3.3.1 Lifting scheme

Given a polyphase matrix p (z)
Let $\tilde{h}(z)$ and $\tilde{g}(z)$ be the low pass and high pass analysis filters, and $h(z)$ and $g(z)$ be the low pass and high pass synthesis filters, corresponding polyphase matrix [3]

$$\tilde{p}(z)=\begin{bmatrix}\tilde{h}e(z) & \tilde{g}e(z)\\ \tilde{h}o(z) & \tilde{g}o(z)\end{bmatrix} \qquad p(z)=\begin{bmatrix}he(z) & ge(z)\\ ho(z) & go(z)\end{bmatrix}$$

We can always decompose this matrix into lifting steps provided
1. $|p(z)|$ =1.if not one then also some constant in the reconstruction we can manipulate the constant factor

2. ($\tilde{h}(z), \tilde{g}(z)$) complementary filter pair
Then the lifting steps are

$$\tilde{p}(z)=\begin{bmatrix}k & 0\\ 0 & 1/k\end{bmatrix}\prod_{i=1}^{m}\begin{bmatrix}1 & \ddot{s}i(z)\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ \tilde{t}i(z) & 1\end{bmatrix} \quad \text{……..(3.3)}$$

Factorization consist of three steps

1. **Prediction step :** where even samples are multiplied by time domain equivalent of $\tilde{t}i(z)$ and are added to odd samples

2. **update step :** where updated odd samples are multiplied by time domain equivalent $\ddot{s}i(z)$ and are added to even samples

3. **scaling step :** where even samples are multiplied by 1/ k and odd numbers are multiplied by k

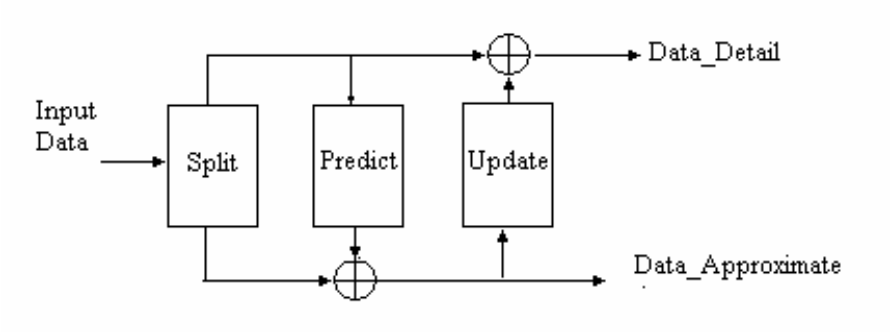


Figure3.5. lifting structure (basic)

3.3.2 Filters and Laurent polynomials

A filter h is a linear time invariant operator and is completely determined by its impulse response: $\{h_k \in R | k \in z\}$. the filter h is Finite Impulse Response (FIR) in case only finite number of filters coefficients h_k are non zero .We let then k_b (respectively k_e) be the smallest (respectively largest) integer number k for which h_k is non zero. The Z-transform of FIR filter h is Laurent polynomial $h(z)$ is given by

$$h(z) = \sum_{k=kb}^{ke} h_k z^{-k} \quad \text{.....} \quad \text{.....(3.4)}$$

The degree of Laurent polynomial is h is is defined as

$$|h| = k_e - k_b \text{.....(3.5)}$$

So length of the filter is degree of associate polynomial plus 1 .Note that the polynomial z^p seen as Laurent polynomial has degree zero, while as regular polynomial it would have a degree p [2].

The set of all Laurent polynomials with real coefficients has a commutative ring structure. The sum or difference of two Laurent polynomials is again a Laurent polynomial. The product of a Laurent polynomial of degree l and a Laurent polynomial of degree l' is a Laurent polynomial of degree $l + l'$ the ring is usually denoted as $R[z,z^{-1}]$.

Within a ring exact division is not possible in general . however for Laurent polynomial , division with remainder is possible take two polynomials a (z) and b (z) $\neq$ 0 with $|a(z)| \geq |b(z)|$ then there always exist a Lauren polynomial $q(z)$ quotient with $q(z) = |a(z)| - |b(z)|$, and a Laurent polynomial $r(z)$ (remainder) with $|r(z)| < |b(z)|$ so that

$$a(z) = b(z)q(z) + r(z) \text{.....(3.6)}$$

we can denote this as C-language notation as

$$q(z) = a(z) / b(z) \text{.....(3.7)}$$

$$r(z) = a(z)\%b(z) \text{.....(3.8)}$$

if $|b(z)| = 0$ which means $b(z)$ is monomial ,then $r(z) = 0$ and division is exact .A Laurent polynomial is invertible if and only if it is monomial

3.3.3 Euclidean Algorithm for Laurent Polynomials:

Take two Laurent polynomials $a(z)$ and $b(z) \neq 0$ with $|a(z)| \geq |b(z)|$ let $a_0(z) = a(z)$

And $b_0(z) = b(z)$ and iterate the following steps starting from i =0

$$a_{i+1}(z) = b_i(z) \text{.....(3.9)}$$

$$b_{i+1}(z) = a_i(z)\%b_i(z) \text{.....(3.10)}$$

Then $a_n(z) = \gcd(a(z), b(z))$ where n is the smallest number for which $b_n(z) = 0$

Given that $|b_{i+1}(z)| < |b_i(z)|$, there is an m so that $|b_m(z)| = 0$ the algorithm then finishes

For n=m+1 . The number of steps thus is bounded by $n \le |b(z)| + 1$ if we let

$$q_{i+1}(z) = a_i(z) \Big/ b_i(z)$$

We have that

$$\begin{bmatrix} a_n(z) \\ 0 \end{bmatrix} = \prod_{i=n}^{1} \begin{bmatrix} 0 & 1 \\ 1 & -q_i(z) \end{bmatrix} \begin{bmatrix} a(z) \\ b(z) \end{bmatrix} \text{.....(3.11)}$$

Consequently

$$\begin{bmatrix} a(z) \\ b(z) \end{bmatrix} = \prod_{i=1}^{n} \begin{bmatrix} q_i(z) & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a_n(z) \\ 0 \end{bmatrix} \text{..... ...(3.12)}$$

and thus $a_n(z)$ divides both $a(z)$ and $b(z)$,if $a_n(z)$ is monomial ,then $a(z)$ and $b(z)$ are relatively prime

The following example illustrate this

Illustration 3.1: Let us consider the (5, 3) filter, with the following

Filter coefficients:

Lowpass: $\left(-1/8, 2/8, 6/8, 2/8, -1/8\right)$

Highpass: $\left(-1/2, 1, -1/2\right)$

The polyphase matrix of above filter is

$$\tilde{p}(z) = \begin{bmatrix} \frac{-1}{8}z + \frac{6}{8} - \frac{1}{8}z^{-1} & \frac{2}{8} + \frac{2}{8}z \\ \frac{-1}{2} - \frac{1}{2}z^{-1} & 1 \end{bmatrix}_1 \quad \text{.....(3.13)}$$

Initially

$$a_0(z) = \frac{-1}{8}z^{-1} + \frac{6}{8} - \frac{1}{8}z \text{.....(3.14)}$$

$$b_0(z) = \frac{-1}{2}z^{-1} - \frac{1}{2} \text{.....(3.15)}$$

The first division gives

$$a_1(z) = \frac{-1}{2}z^{-1} - \frac{1}{2} \text{.....(3.16)}$$

$$b_i(z) = 1 \text{..... .....(3.17)}$$

$$q_{1i}(z) = \frac{1}{4} + \frac{1}{4}z \text{.....(3.18)}$$

The second division gives

$$q_2(z) = \frac{-1}{2}(z^{-1} + 1) \text{.....(3.19)}$$

The remainder is zero so no more divisions, so the final **factorization matrix** now can be decompose into **Lifting steps ,** The polyphase matrix can be decompose into upper ,lower and diagonal matrix.

This upper, lower and diagonal matrix will give the coefficients for the manipulation of DWT core structure[5].

$$p_1(z) = \begin{bmatrix} 1 & 0.25(1+z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -0.5(1+z^{-1}) & 1 \end{bmatrix} \text{.....(3.20)}$$

CHAPTER 4

ARCHITECTURE & ANALYSIS

4.1 WAVELET FILTERS

As discussed previously, the discrete wavelet transform can be implemented in several different ways, each having their own advantage s and disadvantages .the lifting based DWT algorithms however, have a advantage considering VLSI hardware implementations. it uses fewer resources and can have pipelined stages enabling higher operating speeds and lower power consumption .

The JPEG2000 standard supports both convolutions based and lifting based filters, using both lossy and lossless wavelet transforms. since lossless compression requires that no data be lost due to rounding ,a reversible wavelet transform using only rational filter coefficients is used ,In contrast ,lossy compression allows for some data to be lost during compression ,allowing non reversible wavelet transform with non rational filter coefficients to be used

4.1.1 5/3 & 9/7 FILTERS

The default irreversible (lossy) transform is implemented by means of bi orthogonal Daubechies 9-7 tap filter the default reversible filter is lee-gal 5/3 filter .the equation below describes the 5/3 and 9/7 DWT algorithms .As a matter of notation ,the input signal ,low pass sub-band (update coefficients) and high pass sub-band (predict coefficients) are denoted as x[n],s[n],and d[n], respectively .For convenience ,the even and odd input s are defined as $s_0(n)=x[2n]$ and $d_0(n)=x[2n+1]$.[3]

5/3 lifting based DWT algorithm

$$d[n] = d_o(n) - \left\lfloor \frac{1}{2}(s_o[n+1] + s_0(n) \right\rfloor \text{.....} \text{.....} \text{....} (4.1)$$

$$s[n] = s_0(n) + \left\lfloor \frac{1}{4}(d[n] + d[n-1]) + \frac{1}{2} \right\rfloor \text{.....} (4.2)$$

9/7 lifting based DWT algorithm

$$d_1[n] = d_0(n) + \alpha_0(s_0[n+1] + s_0[n]) \quad \text{.....(4.3)}$$

$$s_2[n] = s_1(n) + \alpha_3(d_2[n] + d_2[n-1]) \quad \text{.....(4.4)}$$

$$s_1[n] = s_0(n) - \alpha_1(d_1[n] + d_1[n-1]) \quad \text{.....(4.5)}$$

$$d_2[n] = d_1(n) + \alpha_2(s_1[n+1] + s_1[n]) \quad \text{.....(4.6)}$$

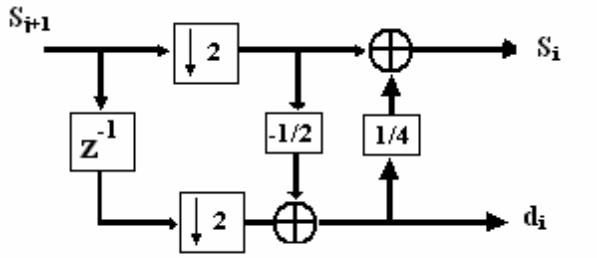


Figure 4.1 5/3 lifting based DA folded lifting based architecture

4.2 ARCITECTURE FOR 5/3 FILTER

4.2.1 folded architecture for reconfigurable devices

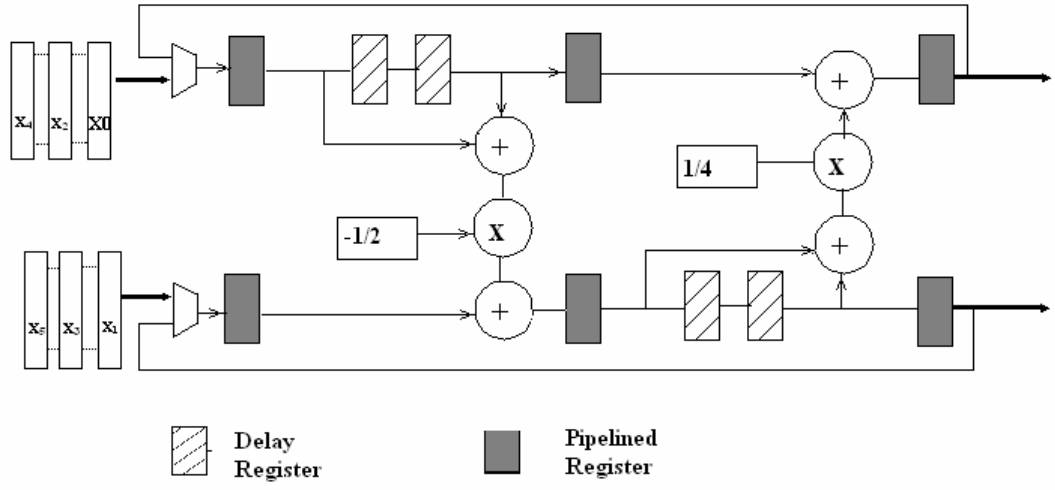


Figure 4.2 Folded architecture for reconfigurable 5/3 DWT

This architecture works on a single phase clock and takes two input samples odd and even indexed input at a time .when performing a 5/3 transform, the multiplier coefficients are ½ and ¼ for predict and update phases respectively .[5]

To accommodate for two lifting steps, multiplier coefficients are first set to α and β for the first lifting step the output coefficients are then fed back through the system with multiplication coefficients set at γ and δ for second lifting step.

Thus calculation of 9/7 output coefficients takes twice as long as 5/3 coefficients .the internal structure is fixed point DSP architecture, optimised for multiplication Coefficients .the folded architecture reduces the overall logic required to implement the algorithms

This architecture proposed takes 3 data inputs at a time as compared to other folded architectures .As can be seen by the equation itself the DWT implementations required 3 data input values at a time this architecture proposed accommodate this by introducing appropriate pipelined stages for the data inputs within the DWT core .the advantage of having such an architecture which can accommodate for 3 data inputs is fewer pipelined stages and less latency in processing steps .the implementation details is discussed further in this chapter

4.2.2 EXTENTION Algorithm for Boundaries of Image

In order to minimize errors due to filtering at boundaries ,symmetric data extensions is performed on the image .symmetric data extension adds a mirror of pixels to outside of boundaries (figure 4.3)

so that large errors are not introduced .this ensures that for filtering operations occurring at image boundaries ,one signal sample exists, corresponding to each coefficient of filter .the sizes of extention on each side depends on number of filter coefficients .[8]

Figure 4.3: symmetric data extention

The number of data terms to extend depends on whether the data set starts or ends with an even term or an odd indexed number. If the data starts or ends with an even number index, two data extensions are required. If the data starts or end s with an odd number then only one a single extension is necessary.

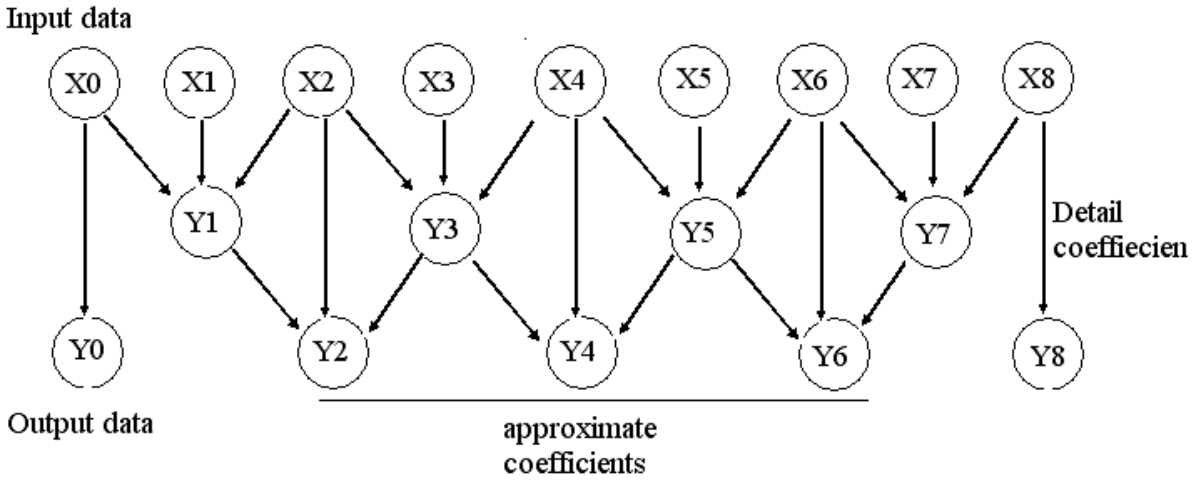


Figure 4.4 1 D DWT lifting data flow

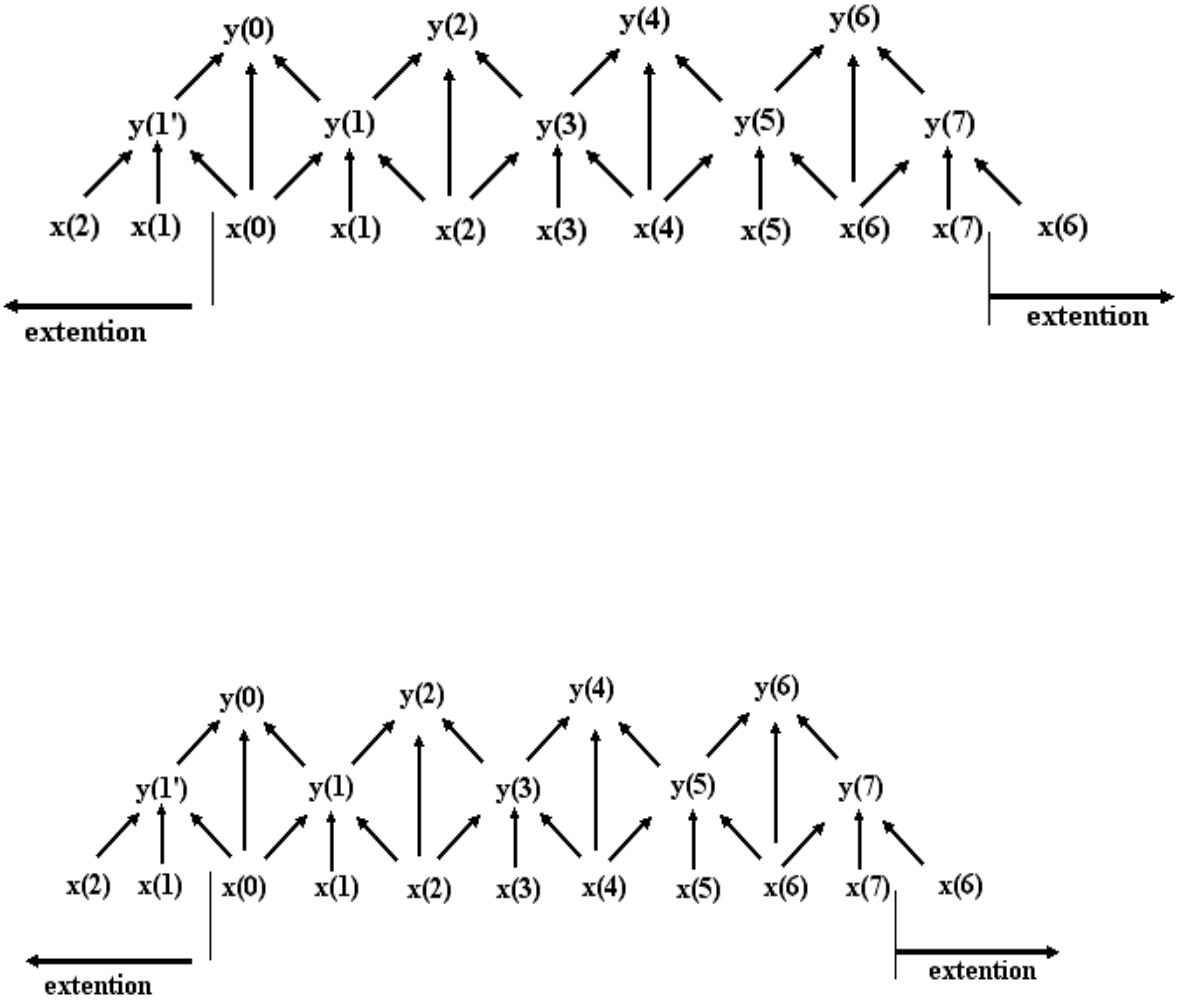


Figure 4.4 (a-top) even start term and odd end term ;(b-down) odd start term and even end term

CHAPTER 5

IMPLEMENTATION

5.1 Software Implementation

5.1.1 MATLAB

Matlab, popular mathematical modeling software, is used for the software implementation of the discrete wavelet transform. Matlab is also for the development of software modules for converting Portable Grey Map (PGM) image files to memory files, and vice versa. The Matlab programming environment provided all the necessary functions and tools needed to achieve this. The following section details the software implementation and the different software modules.

5.1 .2 *1-D Wavelet Transform (WT)*

The main intent of wavelet transform (WT) is to decompose a signal f, in terms of its basis vectors $f = \sum a_i w_i$. To have an efficient representation of signal f using only a few coefficients a_i, the basis functions should match the features of the signal we want to represent. The *(2, 2) Cohen-Daubechies-Feauveau (CDF) Wavelet* [7] is widely used for image compression because of its good compression characteristics. The original filters have 5 + 3 = 8 filter coefficients, whereas an implementation with the lifting scheme has only 2 + 2 = 4 filter coefficients. The forward and reverse filters are shown in Table 4-1. Fractional numbers are converted to integers at each stage. Though such an operation adds non-linearity to the transform, the transform is fully invertible as long as the rounding is deterministic. In Table 6.1 *x* represents the image pixel values and *s* stands for the summing or the low pass coefficients and *d* stands for the difference or the high pass coefficients.

s_i	x_{2i}
d_i	x_{2i+1}
d_i	$d_i-(s_i-s_{i+1})/2$
s	$s_i+(d_{i-1}+d)/4$

Forward transform

Table 5.1 CDF wavelet with lifting scheme

Before an actual hardware implementation it is decided to implement the WT module in software. The discrete wavelet transform (DWT) has been implemented using Matlab. The wavelet transform routine process is a modified version of the biorthogonal Cohen-Daubechies–Feauveau wavelet [9]. The one-dimensional discrete wavelet transform (DWT) can be described in terms of a filter band

The calculation of the wavelet transform requires pixels taken from one row or column at a time. In Equations (4.1) – (4.2), D_i should be calculated before processing S_i. Therefore, the odd pixel should be processed first, then the even pixel due to the data dependency. There are a total of three levels based on the 3-level decomposition wavelet transform algorithm discussed above. In each level, the rows are processed first then the columns. Each level's signal length (amount of each row/column pixels) is half of the previous level. Equations (4.1) – (4.2) are grouped into a function in the MATLAB code. This function is executed repeatedly for every row operations. After the entire set of rows is operated upon, this function is then performed on the columns. This process continues for three level hierarchies where the row/column size is decreased by a factor of 2 at each level

5.1.3 2D DWT

The model for 5/3 lossless lifting scheme based Discrete Wavelet Transform performs a 3-level decomposition on CAMERAMAN (256 x 256 gray scale) .in this it has been shown how 1Dimension DWT is used to perform 2D DWT .

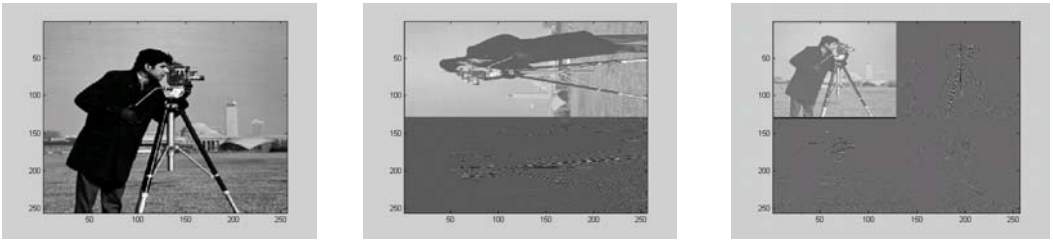


Figure 5.1 (a-left) original CAMERAMAN image; (b-centre) 1D DWT on CAMERAMAN and transposed ;(c-right) 1D DWT on (b) giving single level of decomposition

The original image(*Figure 6.1a*) is first extended at the image boundaries as required and then 1D DWT is performed on the extended image the resulting odd (predict) and even(update) coefficients are placed in a matrix, which is then transposed (*Figure 6.1b*) before the being same is being passed through the same 1D DWT core .this efficiently performs 1D DWT on the image firstly on row wise basis and then on column wise basis , in order to get one level decomposition of image (*Figure 6.1c*)

The model was also modified in order to extract the CAMERAMAN image and 1D transformed odd and even coefficient values in a suitable format .these values would be fed to HDL model and the results of HDL model Simulations are compared with MATLAB model for the verification

5.2 Hardware Implementation

The architecture prescribed here is studied and the detail hardware has been designed for VLSI implementation according to the algorithm flow to get the required output .

The 5/3 lifting based DWT architecture has been developed based DWT architecture proposed in [13] on and implemented .I will refer to this as wavelet_53 implementation

5.2.1 Overview

The 5/3 DWT is a lossless algorithm .to prevents errors due to rounding a 16-bit fixed point DSP implementation working at 7 bit precision.

The image consist of 8 bit pixel values ,which are originally shifted left by 7 places to comply with the cores architecture .the input and output of DWT core are 16 bit signed vectors. The internal data path is 20 bits to accommodate for overflow during the calculations

Since the cores are indexed for image compression – a computationally intensive task processing large number of data samples – a pipelined architecture for predict and update stages was chosen .the core works on the rising edge of clock and outputs between modules are registered (all registers are reset enable) this should be useful during the synthesis and place and route stages allowing for more slack between modules.

The following naming convention are used within the core

- The data inputs and outputs of the designs are prefixed “in_/out_’ respectively (e.g. in_even_0 [ ] and out_odd_ [ ])
- Registers introduced for delays or pipelined stages are prefixed ‘r_’. The delay stage is indicated by the relevant number being append to signal (e.g. r_start_1) implies that start signal delayed by one clock cycle
- Wires are prefixed ‘w_’, these are mainly used for signed extensions of the input and and when assigning the data outputs (e.g.w_in_even_0 [ ] and w_y_odd [ ]).

5.2.2 Predict and Update phases

The predict and update phases were coded as separable modules, in order to be reused by the different implementation. They can be combined or instantiated within the top level modules of any architecture ,since the basic block of building is to split the input stream and then it will be followed by prediction step and then updation step **.wavelet_53** makes up the DWT core here for the 1D processing .

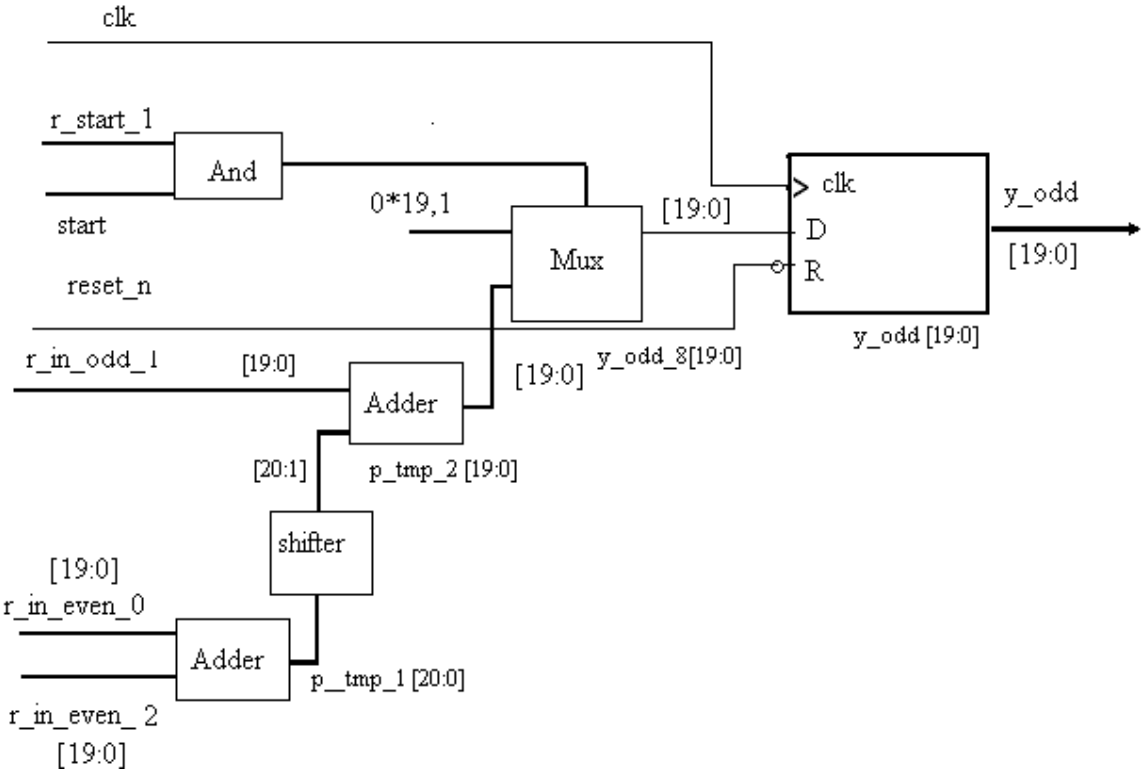


Figure 5.2 predict phase

As can be shown from the figure the two even data values have been taken in a clock and from that we are predicting the odd value (high pass) of the next level odd value that has been the high pass values for the data signal, that's how we are predicting the high pass values from the low pass signal values

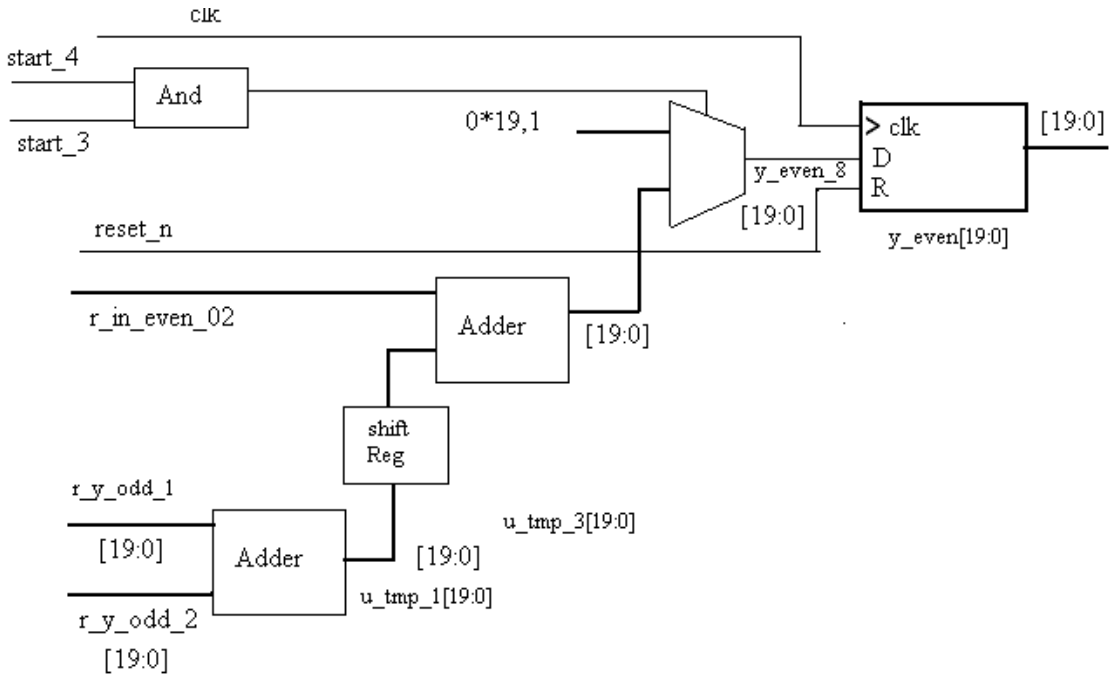


Figure 5.3 Update phase

In this architecture as we can see that the input are two odd values and we are calculating the odd one .this ,means we are updating the low pass values by using two high pass values .

Variable local to predict and update stages are prefixed p_/**u_** respectively. They are used in calculating the predict and output coefficient as described in **Equation 4.1.**

The output of the predict and update stages **y_odd** [ ] and y**_even [ ]** are registered on the rising edge of the clock

5.3 lifting _53 & wavelet_53 Implementation

The lifting _53 and wavelet_53 implementations are very similar in nature .the main difference is that wavelet_53 takes two input at a time while the lifting_53 accepts 3 data inputs at a time, thus wavelet_53 implementation requires additional pipelined stages as shown in figure

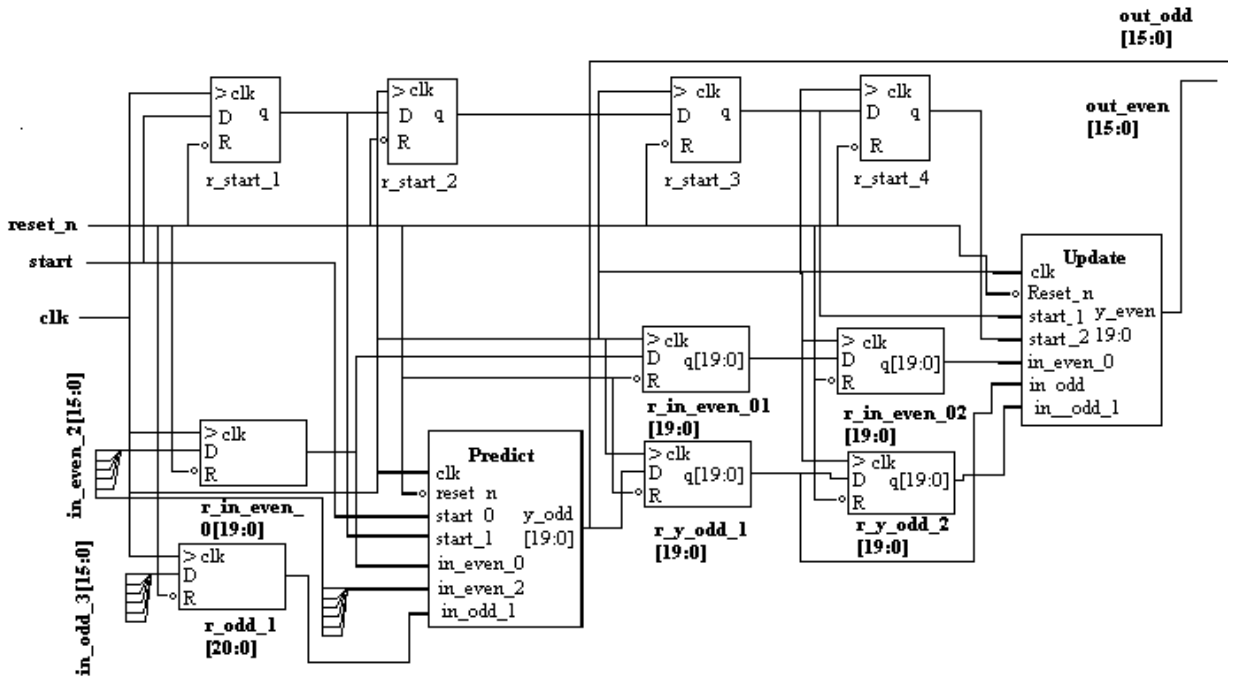


Figure 5.4 wavelet_53 architecture

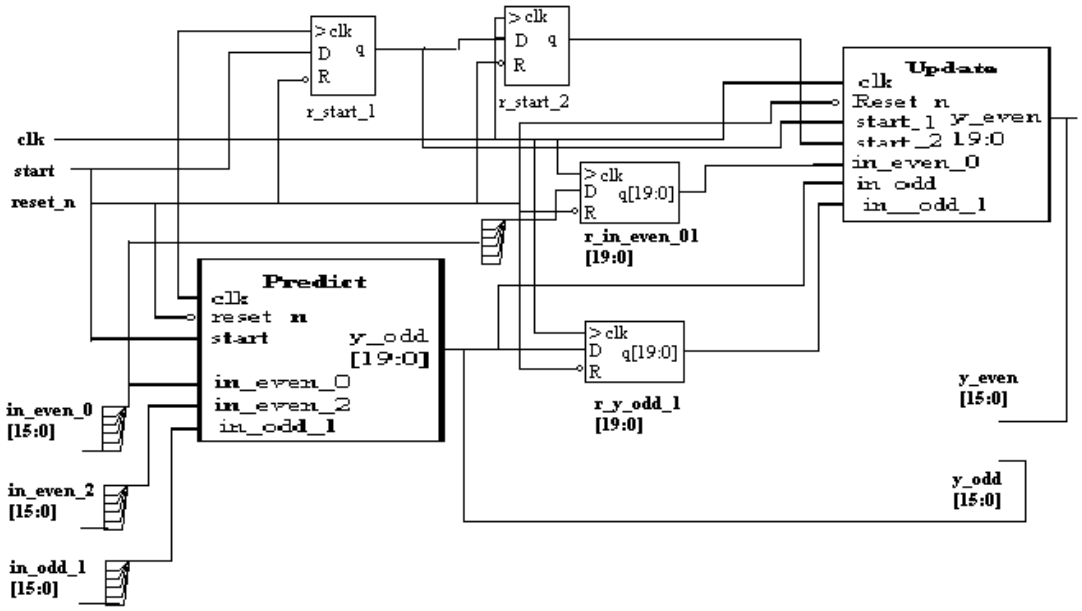


Figure 5.5 lifting_53 architecture

5.4 Test Bench

The test bench is designed to read in the provided image source file, feed the data samples into DWT core under test and store the output values into results files. It uses the dwt core to perform 1D DWT on image source data. This is considered sufficient to verify the working of DWT core and for power analysis. The simulations results are then compared with the reference results to confirm the design is functioning correctly.

The test bench needs to provide with dimensions of image source (i.e number of source and columns). It creates a memory for image according to dimensions specified and input file containing the image data is read into this memory

The test bench was initially developed to provide the data to DWT core under test based on embedded extention, with a view to converting it to an independent hardware module running alongside the DWT core ,performing the memory related functions .it cycles through the memory locations providing the necessary data extentions at the and end of the row .in the present state it is a purely behavioral model ,but the development of this as a separate hardware module can be considered as a future work.

Since the input reference source provided is pre-extended, the code within the test bench responsible for data extentions is currently commented out.

CHAPTER 6

RESULTS

6.1 MATLAB Results

Figure6.1 Input image (256 x 256)

Figure 6.2 ***Input image-extended (258 x 258)***

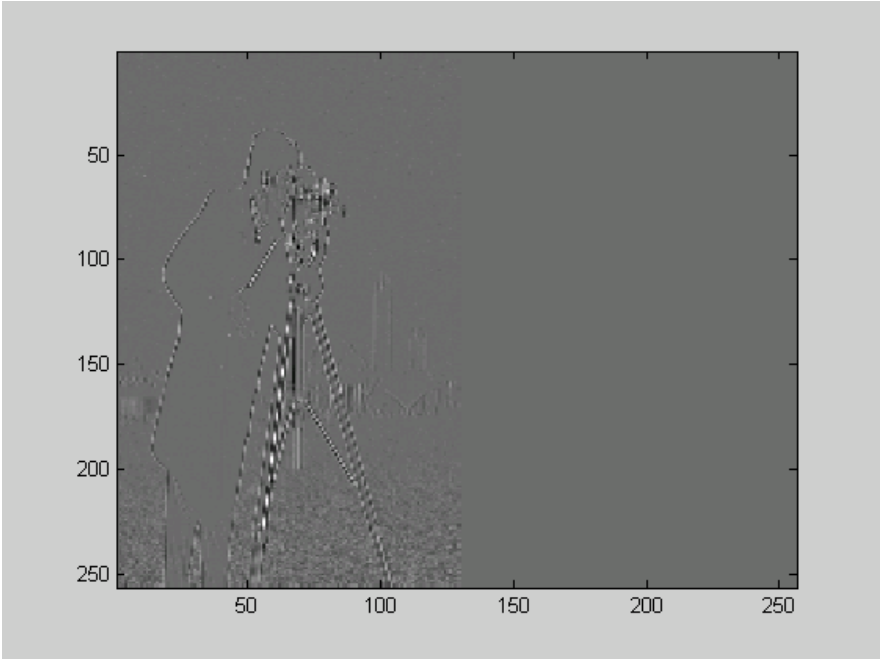


Figure 6.3 level-1 (Row processing) prediction step

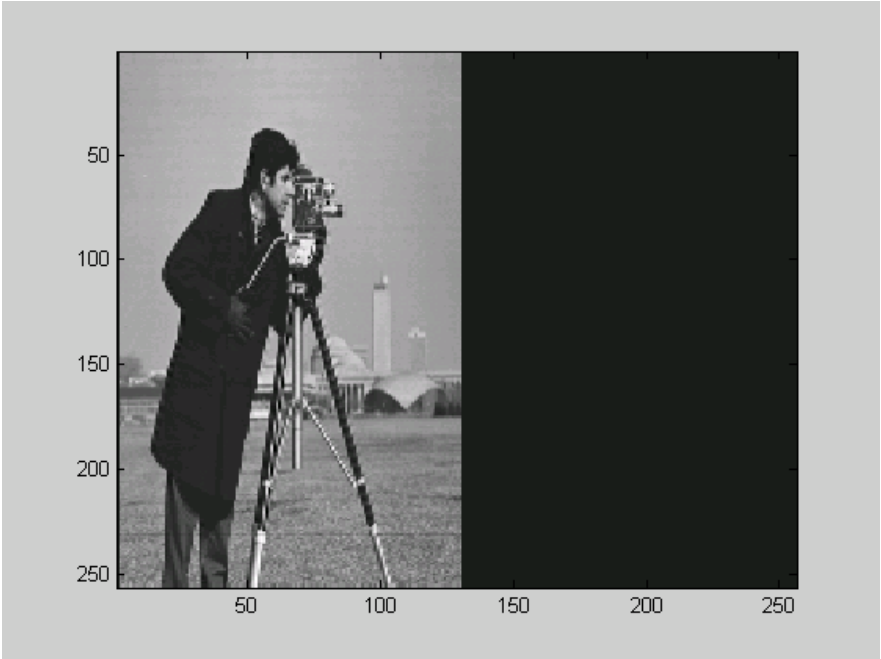


Figure 6.4 level-1 (Row processing) updation steps

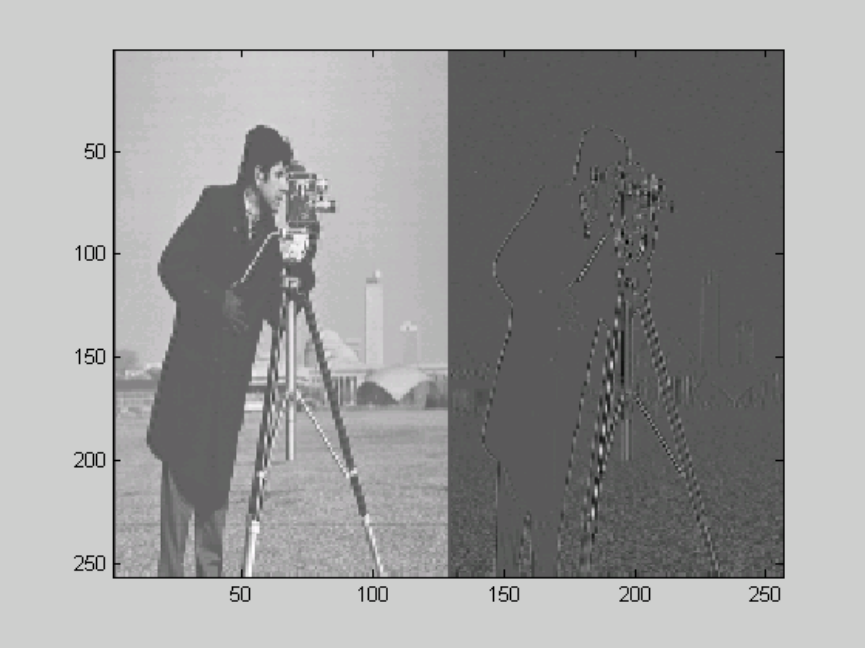


Figure 6.5 1 level 1D DWT of an image

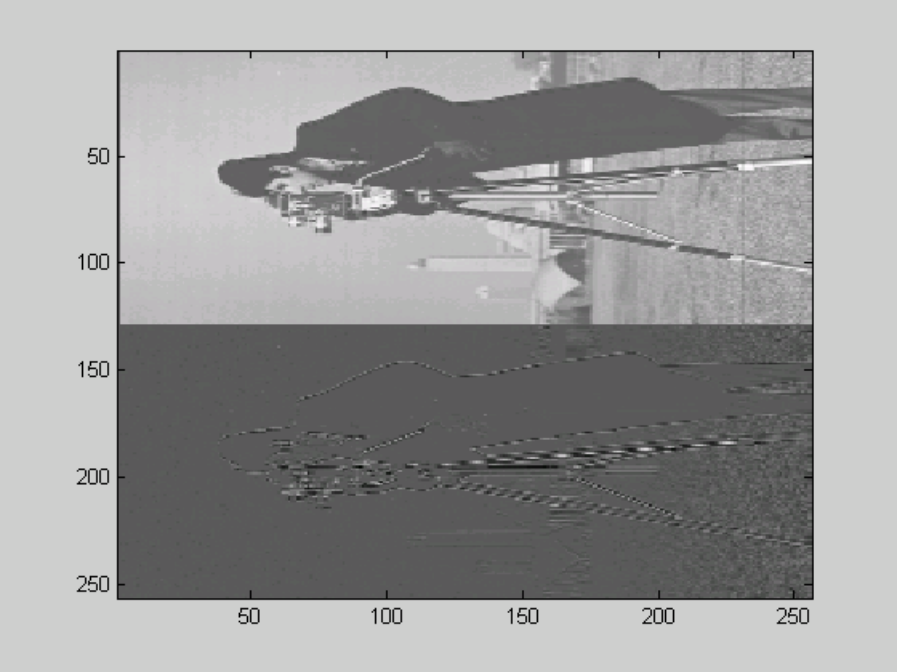


Figure 6.6 1 level 1D DWT Transpose of an image

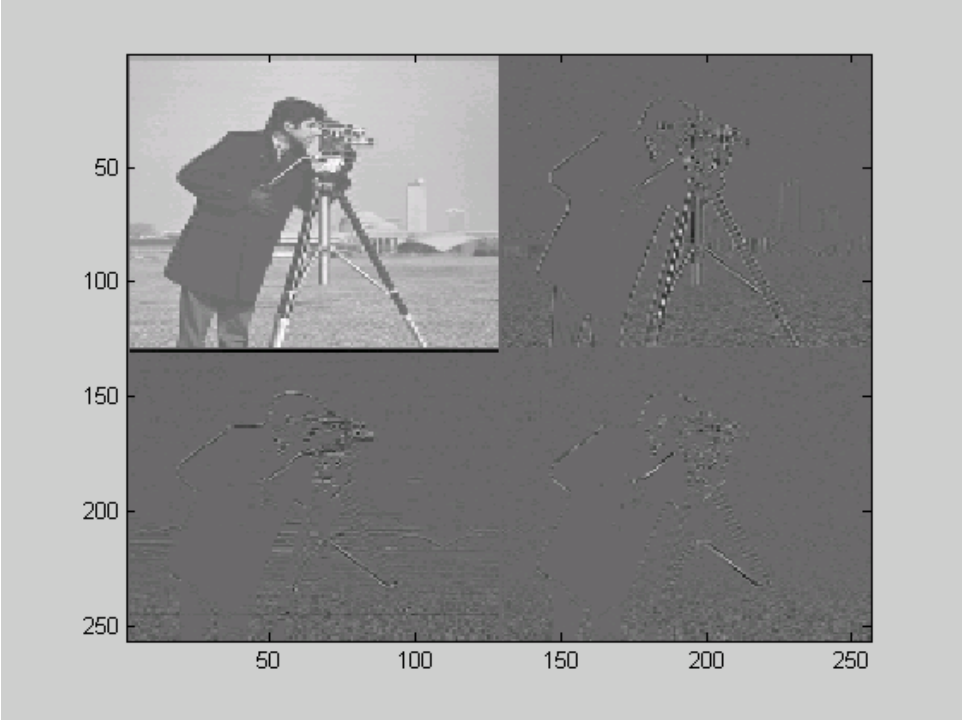


Figure 6.7 1 level 2D DWT of an image

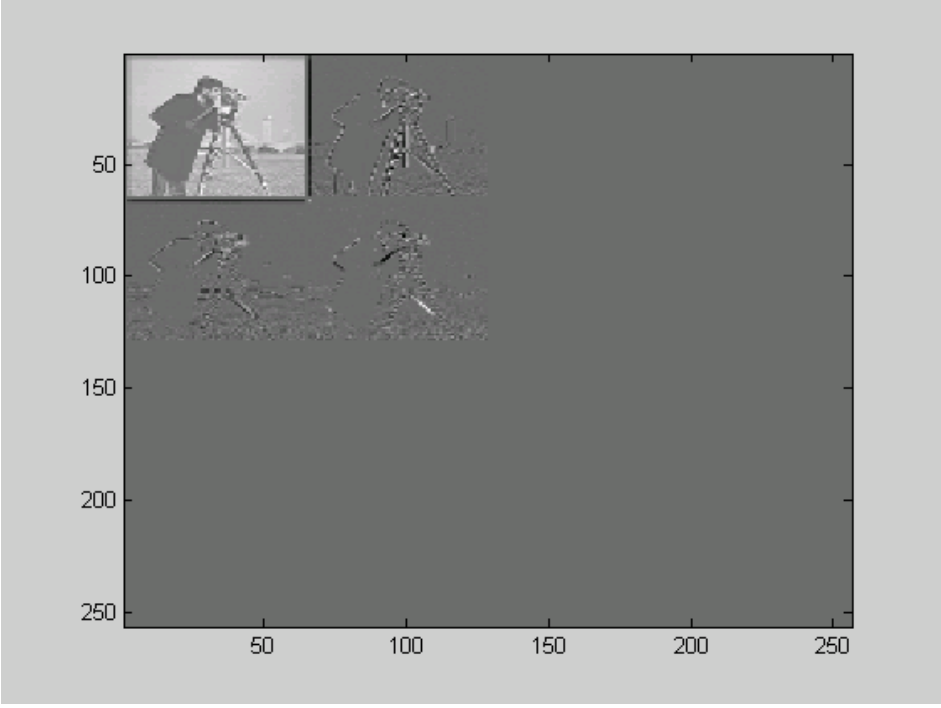


Figure 6.8 2 level 2D DWT of an image

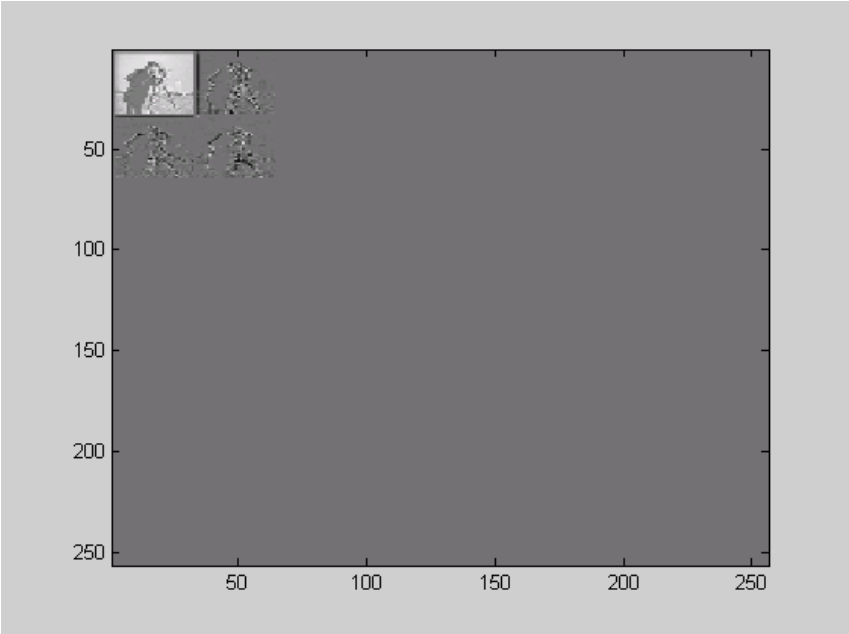


Figure 6.9 3 level DWT of an image

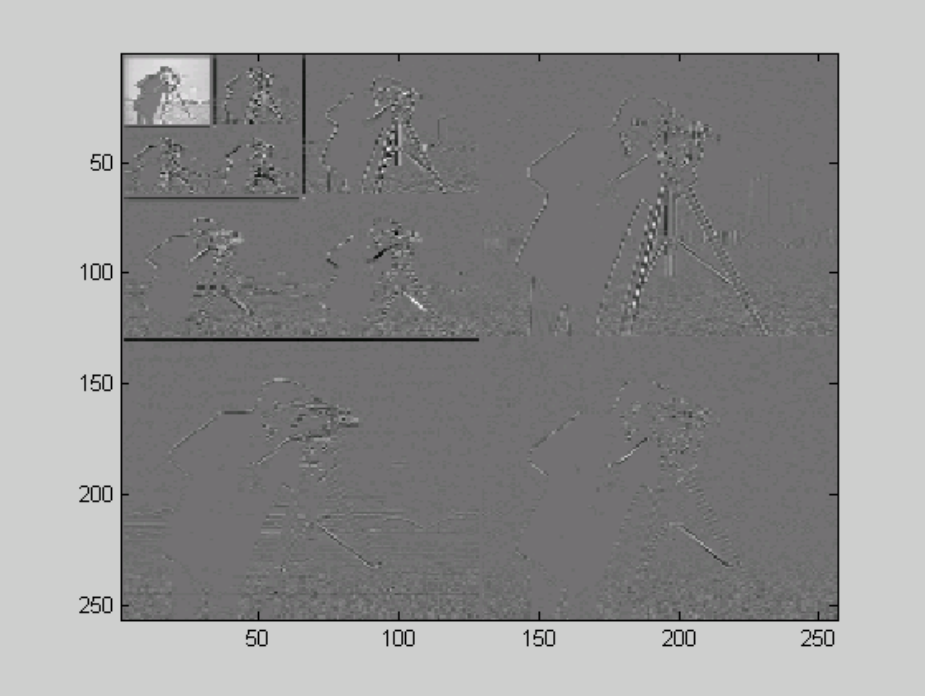


Figure 6.10 dyadic decomposition of an image (3 level)

6.2 HARDWARE RESULTS

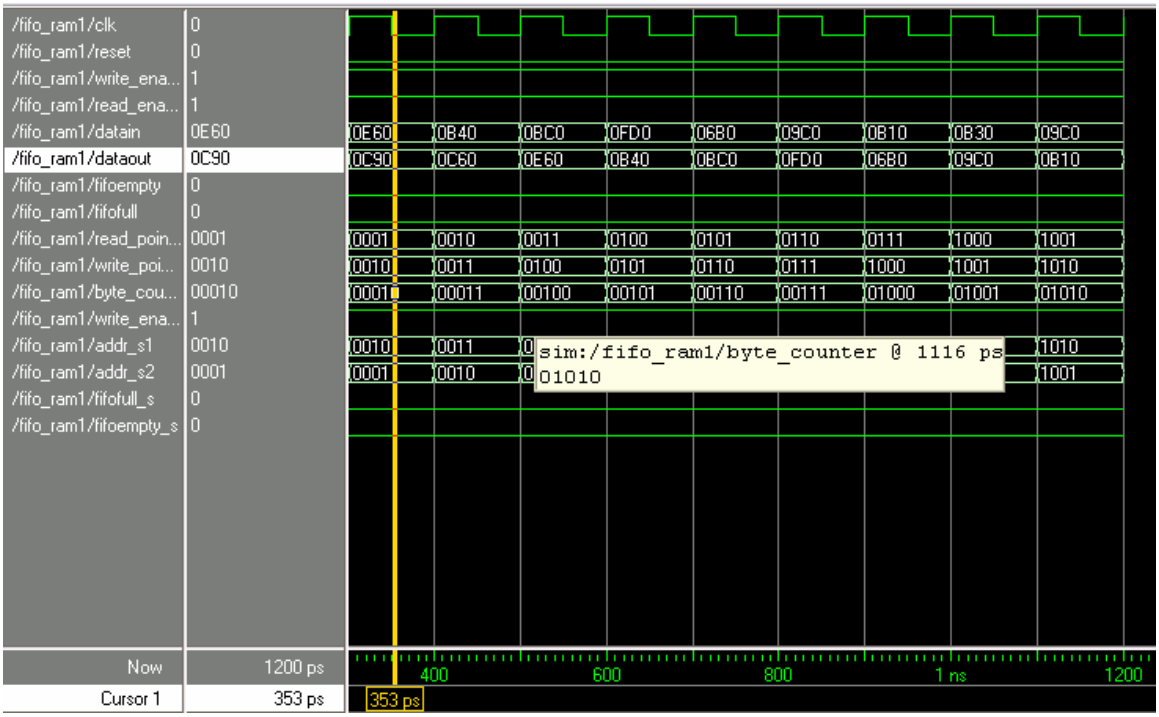


Figure 6.11 Fifo_16 bit async

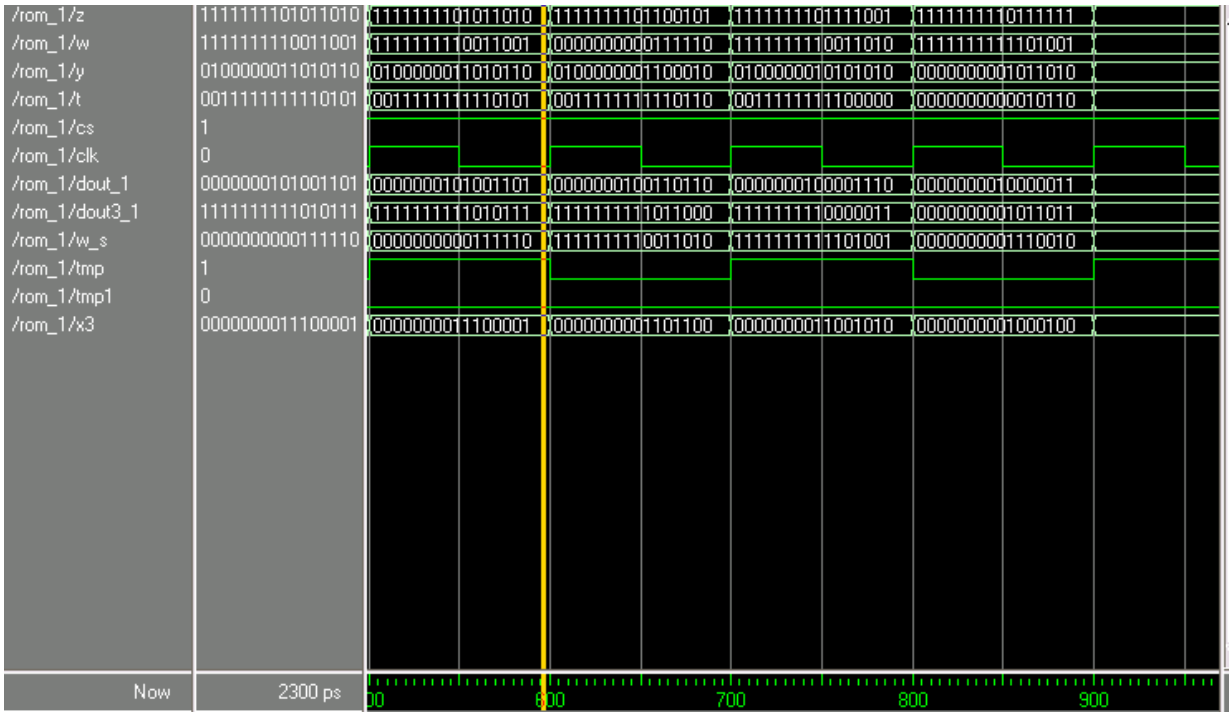


Figure 6.12 Processing of 9x9 block using the ROM (decimal)

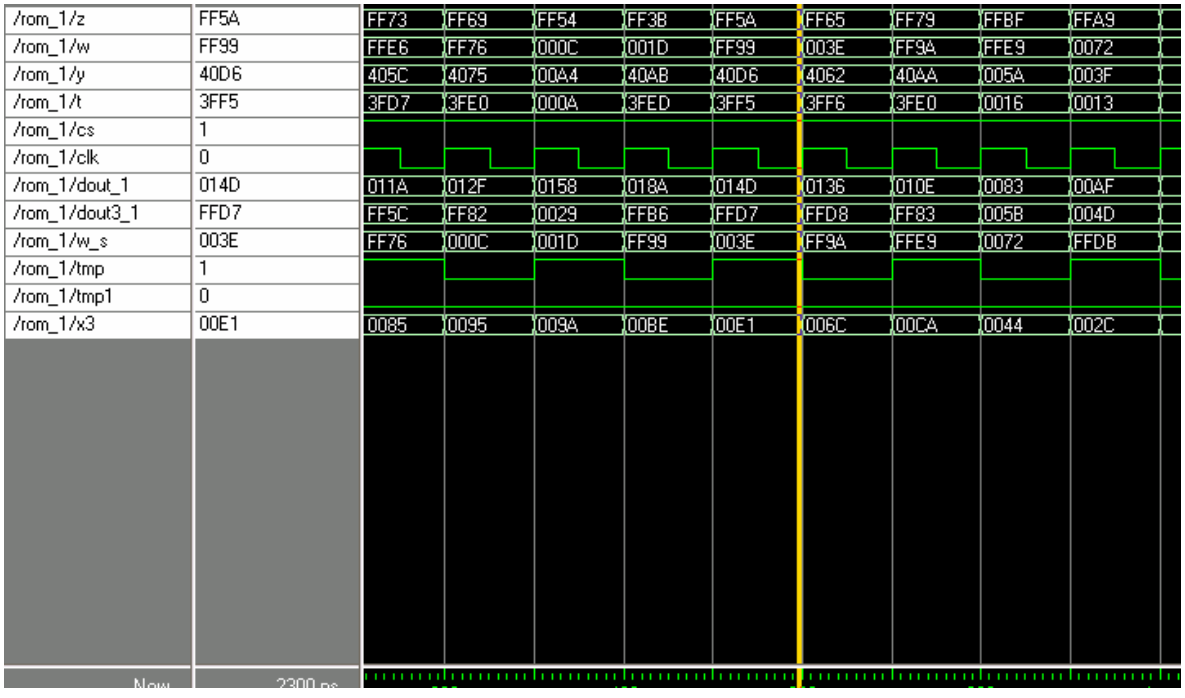


figure 6.14 Processing of 9x9 block using the ROM (hex value)

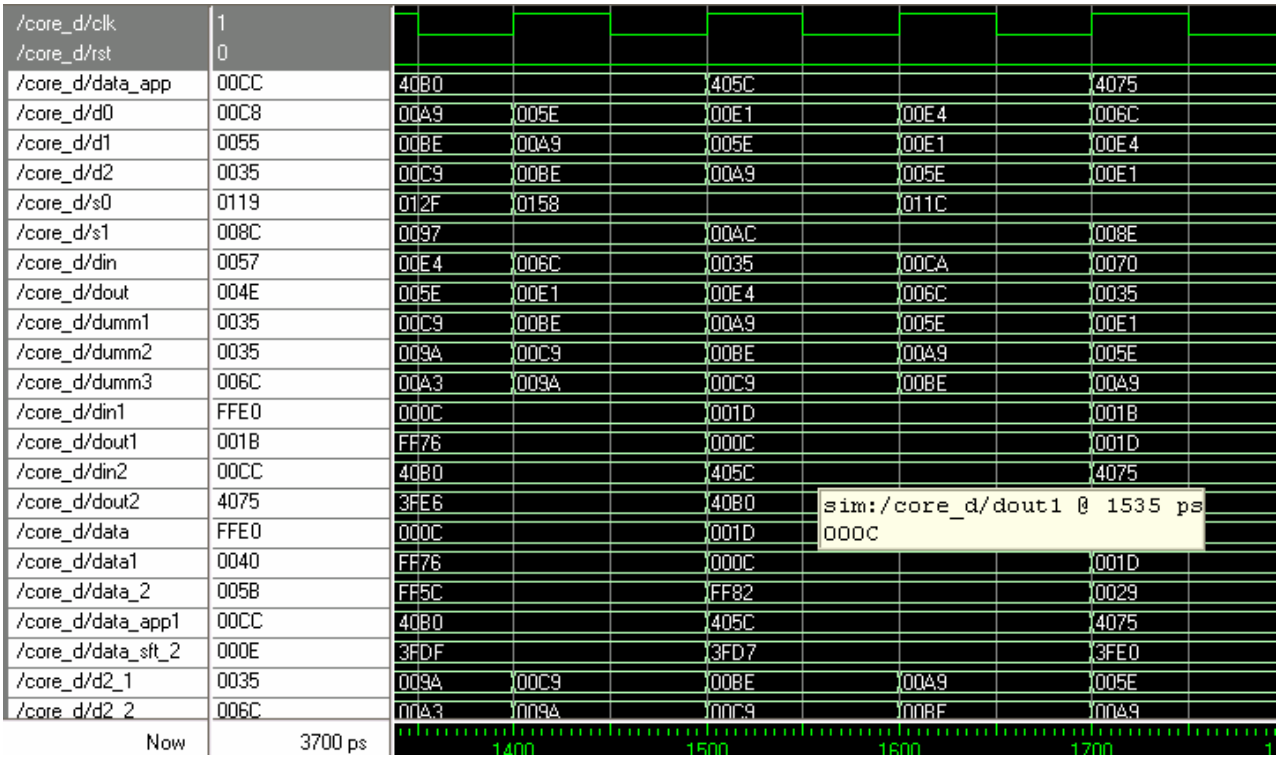


Figure 6.15 core 1D processing of (9x9) matrix (using FIFO)

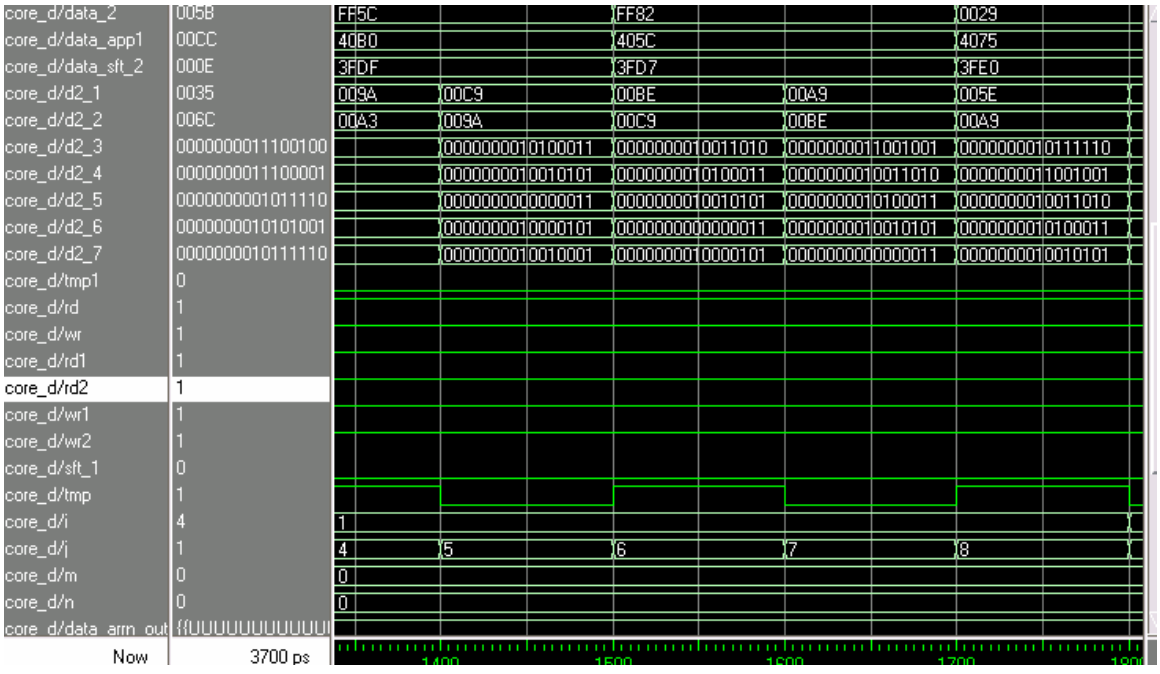


figure 6.16 core 1D processing of (9x9) matrix processing 81 elements

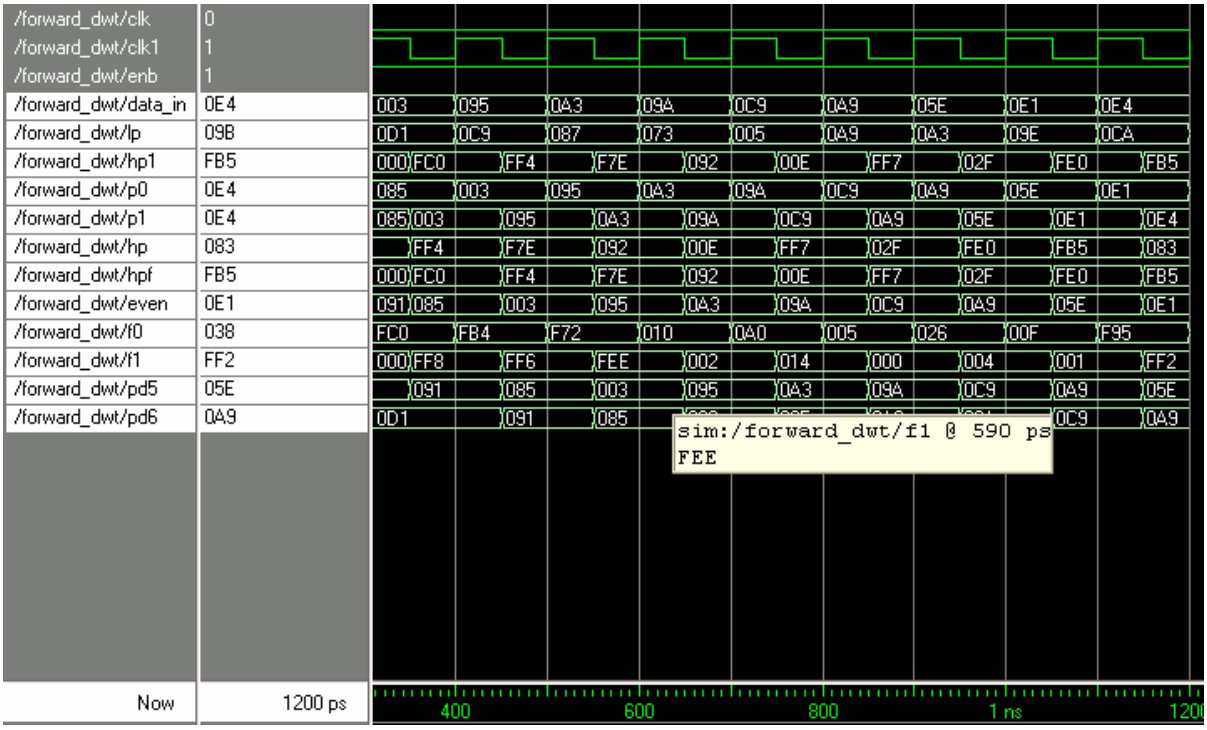


Figure 6.17 forward wavelet transform (input string)

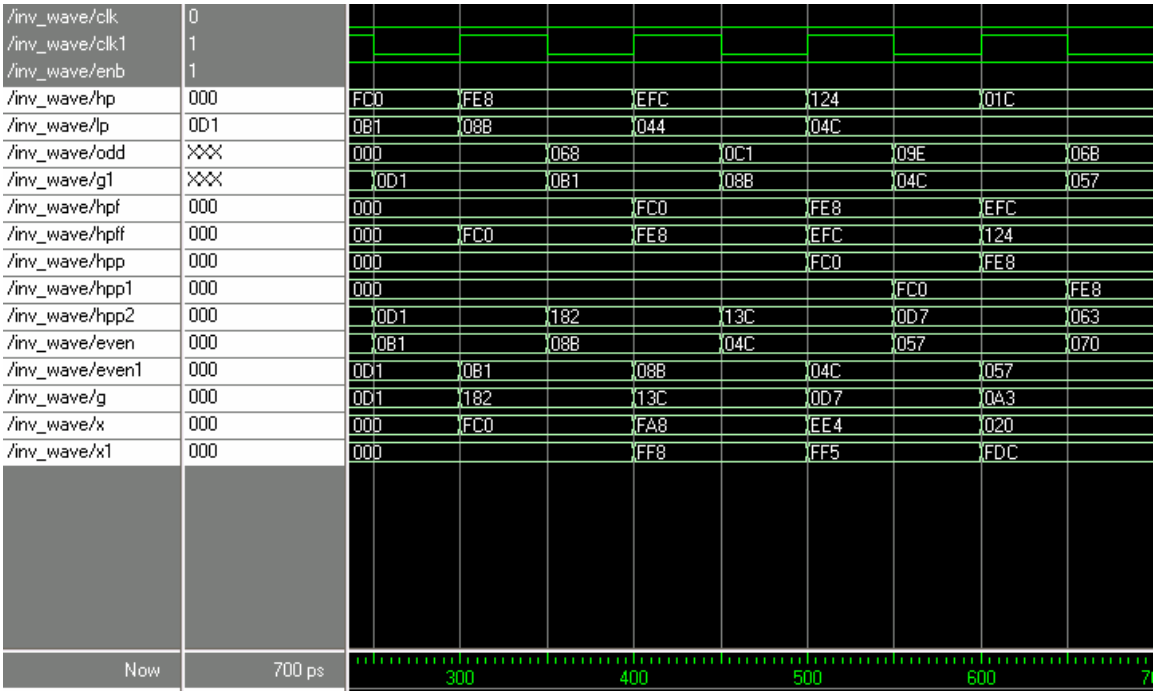


Figure 6.18 Inveverse DWT

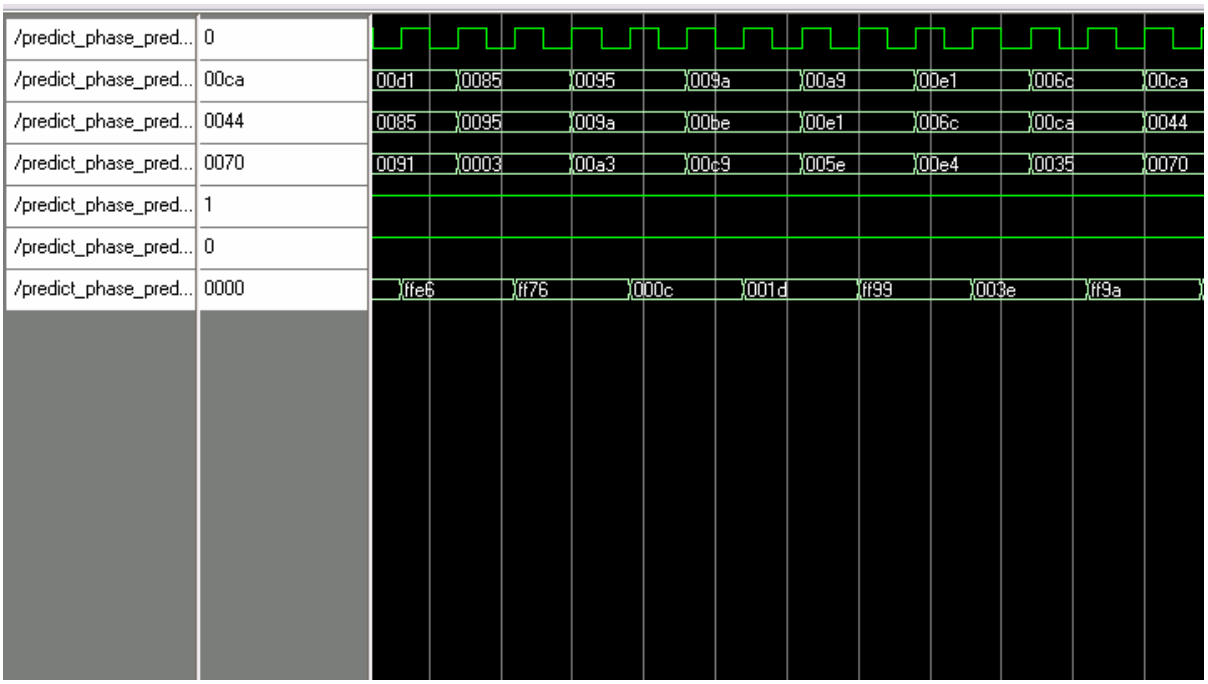


figure 6.19 predict phase

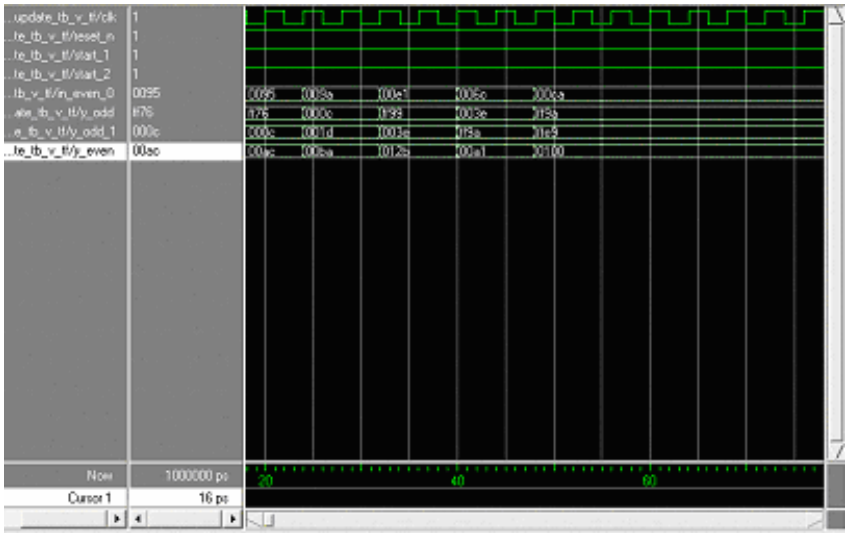


Figure 6.20 Update Phase

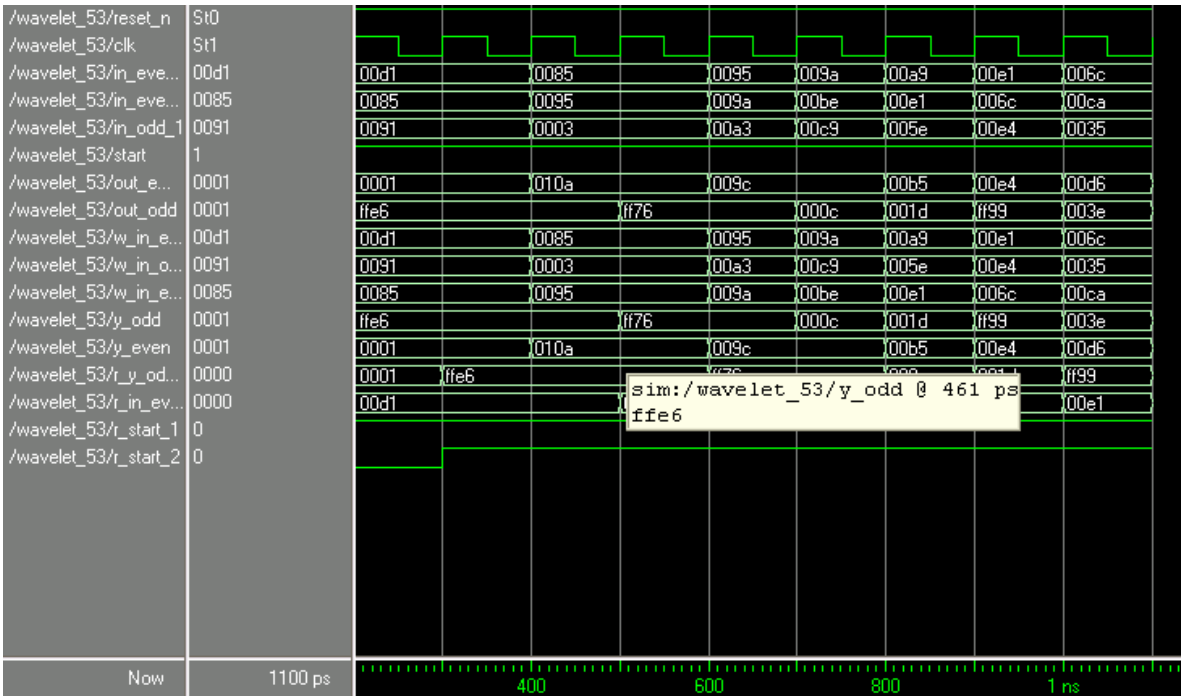


Figure 6.21 1D 5/3 filter simulation results for DWT

6.3 HARDWARE SYNTHESIS RESULTS

Unit FIFO	No of Slices	No of Flip-Flops	No of 4 input LUT	No of IOB's	No of BRAM	Gate Count	Clock (ns)	Frequency (MHz)
8 bit async	17	21	17	19	1	60712	5.693	175.654
16 bit async	17	21	17	35	1	60772	5.693	175.654

Table 6.1 synthesis results of FIFO of different widths

Unit	No of Slices	No of Flip-Flops	No of 4 input LUT	No of IOB's	No of BRAM	Gate Count	Clock (ns)	Frequency (MHz)
Forward DWT	76	113	59	38	0	59688	15.420	64.851

Table 6.2 synthesis results of forward block

Unit	No of Slices	No of Flip-Flops	No of 4 input LUT	No of IOB's	No of BRAM	Gate Count	Clock (ns)	Frequency (MHz)
Reverse DWT	78	136	53	50	1	59688	10.150	98.522

Table 6.3 synthesis results of reverse block

Unit	No of Slices	No of Flip-Flops	No of 4 input LUT	No of IOB's	No of BRAM	Gate Count	Clock (ns)	Frequency (MHz)
WAVELET_53	84	82	103	82	0	60712	12.269	81.553

Table 6.4 synthesis results o f wavelet_ 53 block

CHAPTER 7
CONCLUSION & FUTURE EXTENTION

7.1 Conclusion

This thesis introduced the basic wavelet theory used for lifting based wavelet transform. FPGA based hardware design methodology is presented. DWT algorithm using the *(2, 2) CDF* wavelet and *Debauchies-4 tap* wavelet is studied and implemented in software. Since the *(2, 2) CDF* based DWT is the simplest, it is selected for hardware implementation.

The *(2, 2) CDF* algorithm is studied and mapped for hardware implementation. The algorithm is behaviorally described using VHDL. After successful verification the design is synthesized and tested on prototype hardware. Performance analysis of the hardware design with respect to software is evaluated.

It is seen that the hardware speed up obtained is several times compared to software implementations. This framework for an FPGA-based DWT system would allow a signal/image processing application developer to generate FPGA configurations for DWT at a high level rather than spending considerable time learning and designing at a gate and routing level. Thus, the end-user will benefit from the high performances of FPGA devices while designing at a high level with familiar tools.

The results are promising when compared to software; however, further work needs to be done towards the extension of the system to handle different arithmetic representation, different wavelet analysis and synthesis schemes along with different architectures.

Reconfigurable hardware is best suited for rapid prototyping applications where the lead time for implementation can be critical. It is an ideal development environment, since bugs can be fixed and multiple design iterations can be done, with out incurring any non recurring engineering costs. Reconfigurable hardware is also suited for applications with rapidly changing requirements. In effect, the same piece of silicon can be reused.

Two different 5/3 wavelet architectures have been discussed here both have their own advantages and disadvantages, multiplication is particularly expensive in hardware so That 9/7 filter implementation have been ignored for this part of thesis. the 5/3 lifting based algorithm have proved very efficient in terms of area ,speed and power consumption

7.2 FUTURE WORK

The DWT cores can still be optimised and can be used as a separate IP module .here it has been done only for I level Row processing due to limitation of memory we have taken some pixel values to check whether the core is working properly or not ,one can go for the 2 D DWT implementation of the lifting structure .

Here 9/7 filter have not been included .So a future extention of this some one can go 9/7 implementation of lifting structure

I hope that work undertaken as a part of this project proves to be useful in achieving some of these goals

REFERENCES

[1] I. Daubechies, W and W. Sweldens, “Factoring wavelet transforms into Lifting schemes,” *J. Fourier Anal. Appl.*, vol. 4, pp. 247–269, 1998.

[2] W. Sweldens, “The lifting scheme: A new philosophy in biorthogonal Wavelet constructions,” in *Proc. SPIE*, vol. 2569, 1995, pp. 68–79.

[3] K. Andra, C. Chakrabarti, and T. Acharya, “A VLSI architecture for Lifting based wavelet transform,” in *Proc. IEEE Workshop Signal Process. Syst.*, Oct. 2000, pp. 70–79.

[4] M. Vishwanath, R. Owens, and M. J. Irwin, “VLSI architectures for the Discrete wavelet transform,” *IEEE Trans. Circuits Syst. II*, vol. 42, pp. 305–316, May 1995.

[5] W. Jiang and A. Ortega, “Lifting factorization-based discrete wavelet transform architecture design,” *IEEE Trans. Circuits Syst. Video Techno.*, vol. 11, pp. 651–657, May 2001.

[6] A. R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, “Wavelet Transforms that map integers to integers,” *Appl. Comput. Harmon. Anal.*, vol. 5, pp. 332–369, July 1998.

[7] T. Acharya, “A high speed systolic architecture for discrete wavelet Transforms,” in *Proc. IEEE Global Telecomm. Conf.*, vol. 2, 1997, pp.669–673.

[8] K.C.B Tan and T .Arslan " *Embedded Extention Algorithm for lifting based Discrete Wavelet transform* " Dept . of Electronics & Electrical Engineering, university of Edinburgh ,2002

[10] K. K. Parhi and T. Nishitani, "VLSI architectures for discrete wavelet Transforms," *IEEE Trans. VLSI Syst.*, vol. 1, pp. 191–202, June 1993.

[11] K. Andra, C. Chakrabarti, and T. Acharya, "A VLSI architecture for Lifting based wavelet transform," in *Proc. IEEE Workshop Signal Process. Syst.*, Oct. 2000, pp. 70–79.

[12] W.Sweldens. "The lifting scheme: A custom-design construction of biorthogonal wavelets", Appl. Comput. Harmon. Anal. Vol.3 (2).pp.186-200. 1996.

[13]Chung-Jr Lian, Kuan-fu Chen, hong-Hui Chen, and Liang-Gee Chen, "Lifting Based Discrete Wavelet Transform Architecture for JPEG2000",Proc ISCAS, pp.445-448, 2001

[14] W.Sweldens, "The Lifting Scheme : A construction of Second Generation Wavelets", SIAM J.Math. Anal., vol.29,no.2, pp.511-546, 1997.

[15] P.P.Vaidyanathan " Multirate Systems and Filter Banks " Pearson education ,Low Price Edition.

APPENDIX

FPGA TOOLS AND MEATODS

A.1 FPGA INTRODUCTION

As the first aim of this work is testing the feasibility and performance of an algorithm, great flexibility and versatility are required in the design description. The tools and methodology being used are able to help designers deal with architectural complexity, and are capable of system parameterization and result comparisons. Moreover, the tools should limit the time and resources needed for circuit verification.

There are two basic hardware design methodologies currently available: schematic and language-based designs. The schematic based design, despite of better optimization implementation in terms of both area and speed, does not fulfill the requirements of simplicity and speed of designing stated above. Nevertheless, the language based counterpart suffers from the synthesis tool used as well as the code style with which designs being synthesized are written, as is the case for software developing compilers.

These two factors can potentially lead to variances when comparing synthesis tool outputs. For the purpose of this work the language based design was chosen. The language based design choice relies primarily on two components: the hardware side, represented by the FPGA technology, and the programming language counterpart, represented for the language chosen, i.e. VHSIC Hardware Description Language (VHDL).

The former is discussed in the first section, where the more relevant features, from this project point of view, will be illustrated; the latter is introduced in the next section which looks at basic and fundamental concepts in VHDL.

A.1.1 SRAM-based FPGA

The SRAM FPGA gets its name from the fact that programmable connections are made using pass-transistors, transmission gates, or multiplexers that are controlled by static random access memory (SRAM) cells.

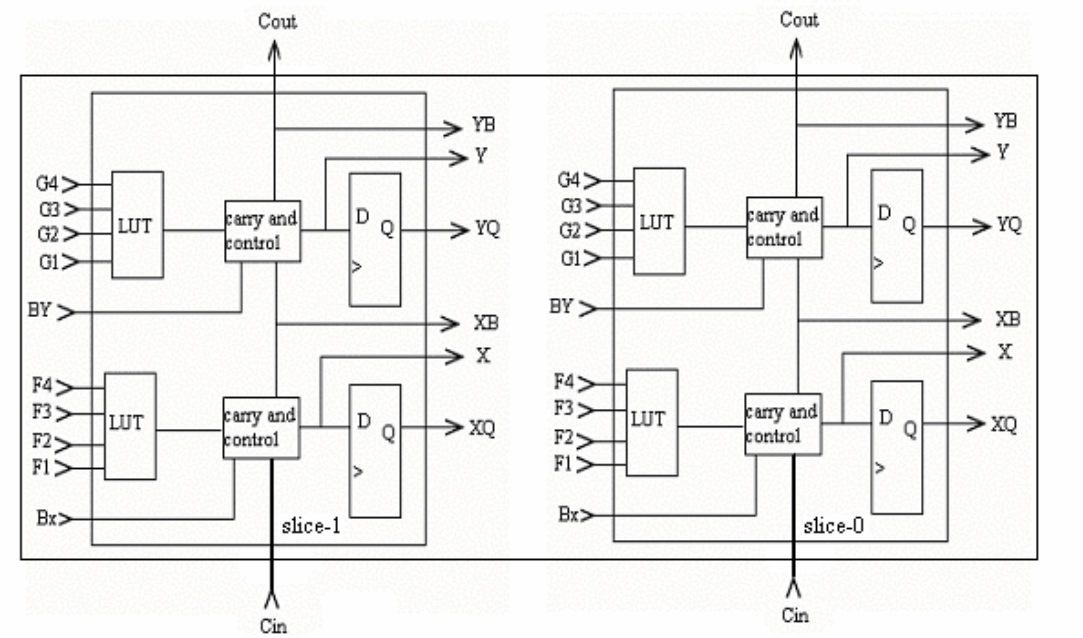


Figure A.1 slice Virtex CLB (courtesy of Xilinx inc.)

The advantage of this technology resides in the fact that it allows fast in-circuit reconfiguration. The major disadvantage, however, is the size of the chip required by the SRAM technology.

The FPGA has three major configurable elements: configurable logic blocks (CLBs), input/output blocks (IOB's), and interconnects. The CLBs provide the functional elements for constructing user's logic

As shown in figure A-1, it normally comprises of a Look Up Table (LUT) with 4 inputs and a flip-flop output. The IOBs provide the interface between the package pins and internal signal lines.
The programmable interconnect resources provide routing paths to connect the inputs and outputs of the CLBs and IOBs onto the appropriate networks. Customized configuration is established by programming internal static memory cells that determine the logic functions and internal connections implemented in the FPGA.

The XCV300, belonging to the Virtex family from Xilinx, is the FPGA chosen for this implementation. Its features, summarized in Table 3-1, are generous in gates and I/O pins allowing the designer long synthesis waits and troublesome pin assignments.

System gates	322,970
CLB Arrays	32 x 48
Logic cells	6912
Maximum IOs	316
Block RAM bits	65,536
Flip-Flop count	6144

Table A.1 Virtex FPGA XCV-300 Features

A.2 Software Tools

The following section briefly describes the software tools used to write, compile, test and synthesize the project. To develop, test, and debug the project several commercial programs played an important role. Table 4.1 provides a summary of the software tools used in this work.

Software Implementation	**MATLAB Version 6.5**
Development Environment	**Xilinx ISE Version 6.1 supported editing compilation and simulation.**
Simulations	**Xilinx Waveform Compiler and MODELSIM Version 5.7c for Xilinx ISE**
Synthesis	**Xilinx Synthesis Tool (XST) Version 6.1 and**
Place-and-Route and Back-Annotation	**Xilinx ISE Version 6.1**