Analysis of VLSI Architecture For AI Based 1D/2D wavelet filter using Daub-6 Wavelet Filter

M.Mercy Parimala[1]
Assistant professor of ECE[1]
Mekapati Rajamohan Reddy Institute of Technology and Sciences,Udayagiri.

Abstract:
A multiplier-less architecture based on algebraic integer representation for computing the Daubechies-6-tap wavelet transform for 1-D/2-D signal processing is proposed. This architecture improves on previous designs in a sense that it minimizes the number of parallel 2-input adder circuits. The algorithm was achieved using brute-force numerical optimization of the algebraic integer representation. The proposed architecture furnishes exact computation up to the final reconstruction step, which is the operation that maps the exactly computed filtered results from algebraic integer representation to fixed-point. Compared to our recent work, this architecture shows a reduction of 27.n-16 adder circuits, where is the number of wavelet decomposition levels. The design is physically implemented for a 4-level 1-D/2-D decomposition using a Xilinx Virtex-6 vcx240t-1ff1156 field programmable gate array (FPGA) device operating at up to a maximum clock frequency of 344/ 168MHz. The FPGA implementation of 1-D/2-D are tested using hardware co-simulation using an ML605 board with clock of 100 MHz A 45 nm CMOS synthesis of 2-D designs show improved clock frequency of better than 306 MHz for a supply voltage of 1.1 V.

Keywords: 1-DWT,2-DWT,Wavelets, Daub-4,Daub-6 filters,Verilog.

I.Introduction

The discrete wavelet transform (DWT) has been widely used in many fields, such as image compression, speech analysis and pattern recognition, because of its capability of decomposing a signal at multiple resolution levels. Due to the intensive computations involved with this transform, the design of efficient VLSI architectures for a fast computation of the transforms have become essential, especially for real-time applications and those requiring processing of high-speed data. The objective of this thesis is to develop a scheme for the design of hardware resource-efficient high-speed pipeline architectures for the computation of the DWT. The goal of high speed is achieved by maximizing the operating frequency and minimizing the number of clock cycles required for the DWT computation with little or no overhead on the hardware resources. In this thesis, an attempt is made to reach this goal by enhancing the inter-stag and intra-stage Parallelisms through a systematic exploitation of the characteristics inherent in discrete wavelet transforms.

Discrete wavelet transform The Discrete Wavelet Transform (DWT) gets its name from being discrete-time and discrete-scale. In other words, the DWT coefficients may have real (floating-point) values, but the time and scale values used to index these coefficients are integers. In the DWT, a sub-band filter transforms a discrete-time signal into an average signal and a detail signal.

Fourier transfrom The Fourier Transform is good for sinusoidal signals, like a voice pattern. But the Fourier Transform has difficulty with discontinuous signals, like an image with edges. Both transforms break down a signal into simpler signals, called sinusoids for the Fourier Transform, and wavelets for the wavelet transform. These simpler signals can be added together to recreate the original.

In order to enhance the inter-stage parallelism, a study is undertaken for determining the number of pipeline stages required for the DWT computation so as to synchronize their operations and utilize their hardware resources efficiently. This is achieved by optimally distributing the computational load associated with the various resolution levels to an optimum number of stages of the pipeline. This study has determined that employment of two pipeline stages with the first one performing the task of the first resolution level and the second one that of all the other resolution levels of the 1-D DWT computation, and employment of three pipeline stages with the first and second ones performing the tasks of the first and second resolution levels and the third one performing that of the remaining resolution levels of the 2-D DWT computation, are the optimum choices for the development of 1-D and 2-D pipeline architectures, respectively. The enhancement of the intra-stage parallelism is based on two main ideas. The first idea, which stems from the fact that in each consecutive resolution level the input data are decimated by a factor of two along each dimension, is to decompose the filtering operation into subtasks that can be performed in parallel by operating on even- and odd numbered samples along each dimension of the data .It is shown that each sub task, which is essentially a set of multiply-accumulate operations ,Can be Performed by employing a MAC-cell network consisting of a two-dimensional array of bit-wise adders. These cond idea in enhancing the intra-stage parallelisms to maximally extend the bit-wise addition

operations of this network horizontally through a suitable arrangement of bit-wise adders so as to minimize the delay of its critical path.

In recent years, the discrete wavelet transform (DWT) has been widely and increasingly used in many fields such as image compression, speech analysis and pattern recognition because of its capability of decomposing a signal at multiple resolution levels. The DWT decomposes a signal into components in different octaves or frequency bands by choosing appropriate scaling and shifting factors where the small scaling factor corresponds to fine details of the signal and the large scaling factor to coarse details, and the shifting factor corresponds to the time or space localization of the signal . In contrast to other transforms, such as Fourier or cosine transforms where the signals are represented in frequency domain only, the DWT decomposes a signal so that it is represented more efficiently and localized in both time (space) and frequency domains. In other words, in the DWT, the time (space) information is not lost in the transformed signal, which is very attractive for the analysis of signals, especially for signals with non-stationary or transitory characteristics.

In accordance with the multiple-level decomposition of a signal, the computation of the DWT can be performed by repeating a process in which a fully scalable window is shifted along the dimensions of the signal with the window size becoming shorter in each repetition. The computing processes of the DWT can be carried out by executing recursively a set of instructions developed in software programs such as SimuWave in Simulink, Wavelet toolbox in MATLAB and WavBox in Toolsmiths. The software implementation for the computation of the DWT is flexible in setting different values of the parameters of the transform and changing the codes for the algorithms.

Regardless of the effort devoted to the design of software algorithms and optimized codes for their implementations, no general-purpose or DSP processor used for their implementation can provide a performance in terms of the computing speed and resource optimization that can possibly be achieved by a hardware implementation. Hardware implementations, in which the computation of the DWT is performed by a custom hardware circuit, it is possible to address the requirements of specific applications such as the speed, power or size of the circuit. In the literature, there exist a number of design efforts on the development of architectures for the DWT computation that focus on such requirements of applications. However, many applications of the DWT computation involve large-volume data such as image or video. The fact that the DWT is multiple resolution level operation adds even more to the vastness of the data to be processed, which adversely affects the requirements of speed, power and the circuit area of the architectures for such applications. Thus, it remains a challenging task to design high-speed, low-power and area-efficient VLSI architectures to implement the DWT.

II.Literature review

2.1.THE BASIC 1-D DWT

The DWT multiplies the input by the shifts (translation in time) and scales (dilations or contractions) of the wavelet. Below are variables commonly used in wavelet architecture literature.

J is the total number of octaves
j is the current octave (used as an index. 1 <= j <= J)
N is the total number of inputs
n is the current input (used as an index. 1 <= n <= N)
L is the width of the filter (number of taps)
k is the current wavelet coefficient

W(n,j) represents the DWT, except:

W(n,0) which is the input signal

The low-pass output is:

$$W(n,j) = \sum_{m=0}^{2n} W(m,j-1)*lowfilter(2n-m)$$

and the high-pass output is:

$$W_h(n,j) = \sum_{m=0}^{2n} W(m,j-1)*highfilter(2n-m)$$

T

2.2.THE 2-D DISCRETE WAVELET TRANSFORM

Why use a 2-D DWT? Why not just treat it as a 1-D signal? The 2-D DWT gives better compression. Also, 1-D representation of 2-D data introduces edge effects, or artificial discontinuities. An application with 2-D data need a true 2-D DWT. 1-D is used in speech and music, while 2-D is used in image compression. The 2-D differs from the 1-D by going in both the X and Y directions, shown in Figure 1.3 for 1 octave. Even 3-D data, such as video, can use a 2-D DWT followed by difference encoding. However, the 3-D DWT outperforms the 2-D DWT.

Two-Dimensional Wavelet Transforms have proven to be highly effective tools for image analysis. In this paper, we present a VLSI implementation of four- and six-coefficient Daubechies Wavelet Transforms using an algebraic integer encoding representation for the coefficients. The Daubechies filters (DAUB4 and DAUB6) provide excellent spatial and spectral locality, properties which make it useful in image compression. In our algorithm, the algebraic integer representation of the wavelet coefficients provides error-free calculations until

the final reconstruction step. This also makes the VLSI architecture simple, multiplication-free and inherently parallel. Compared to other DWT algorithms found in the literature, such as embedded zero-tree, recursive or semi-recursive, linear systolic arrays and conventional fixed-point binary architectures, it has reduced hardware cost, lower power dissipation and optimized data-bus utilization. The architecture is also cascadable for computation of one- or multi-dimensional Daubenchies Discrete Wavelet Transforms.

III.Implemenatation method:

3.1.1 An analysis of Daubechies discrete wavelet transform based on algebraic integer encoding scheme

A new and novel encoding scheme of Daubechies wavelet coefficients for implementing discrete wavelet transform based on algebraic integer is proposed. This encoding technique eliminates the requirements to approximate the transformation matrix elements. Instead of approximating the matrix coefficients, we are able to obtain the exact representations for them. As a result, we achieve error-free calculations up to the final reconstruction step where we can choose an approximate substitution precision based on hardware/accuracy trade-off. A comparison between Daubechies 4 and 6 coefficients is also performed. The last part demonstrates that the new encoding technique offers better performance compared to the classical binary (fixed-point binary) design and it is also very well suited for high-speed VLSI implementation.

3.1.2New techniques for rationalizing orthogonal and biorthogonal wavelet filter coefficients

Most wavelet filters reported in the literature have irrational coefficients. Hardware implementation can be simpler if the filter coefficients are rational valued. In this paper, we present novel methods to rationalize both orthogonal and bi-orthogonal filter coefficients with perfect reconstruction and vanishing moments preservation. Rational orthogonal filter coefficients are obtained using lattice structures. Rational bi-orthogonal filter coefficients are obtained using the complementary filter technique in which one set of filter coefficients is expressed in terms of the other. The rationalized filters have characteristics that are very close to the original irrational filters. The techniques are simple yet general enough to be used for almost any filter bank unlike the techniques in previously reported works.

3.1.3Area and power-efficient architecture for high-throughput implementation of lifting 2-D DWT

We have suggested a new data-access scheme for the computation of lifting two-dimensional (2-D) discrete wavelet transform (DWT) without using data transposition. We have derived a linear systolic array directly from the dependence graph (DG) and a 2-D systolic array from a suitably segmented DG for parallel and pipeline implementation of 1-D DWT. These two systolic arrays are used as building blocks to derive the proposed transposition-free structure for lifting 2-D DWT. The proposed structure requires only a small on-chip memory of (4N + 8P) words and processes a block of P samples in every cycle, where N is the image width. Moreover, it has small output latency of nine cycles and does not require control signals which are commonly used in most of the existing DWT structures. Compared with the best of the existing high-throughput structures, the proposed structure requires the same arithmetic resources but involves 1.5N less on-chip memory and offers the same throughput rate. ASIC synthesis result shows that the proposed structure for block size 8 and image size 512 512 involves 28% less area, 35% less area-delay product, and 27% less energy per image than the best of the corresponding existing structures. Apart from that, the proposed structure is regular and modular; and it can be easily configured for different block sizes.

3.1.4.Memory-efficient high-speed convolution-based generic structure for multilevel 2-D DWT

In this paper, we have proposed a design strategy for the derivation of memory-efficient architecture for multilevel 2-D DWT. Using the proposed design scheme, we have derived a convolution-based generic architecture for the computation of three-level 2-D DWT based on Daubechies (Daub) as well as bi-orthogonal filters. The proposed structure does not involve frame-buffer. It involves line-buffers of size 3(K-2)M/4 which is independent of throughput-rate, where K is the order of Daubechies/bi-orthogonal wavelet filter and M is the image height. This is a major advantage when the structure is implemented for higher throughput. The structure has regular data-flow, small cycle period TM and 100% hardware utilization efficiency. As per theoretical estimate, for image size 512 × 512, the proposed structure for Daub-4 filter requires 152 more multipliers and 114 more adders, but involves 82 412 less memory words and takes 10.5 times less time to compute three-level 2-D DWT than the best of the existing convolution-based folded structures. Similarly, compared with the best of the existing lifting-based folded structures, proposed structure for 9/7-filter involves 93 more multipliers and 166 more adders, but uses 85 317 less memory words and requires 2.625 times less computation time for the same image size. It involves 90 (nearly 47.6%) more multipliers and 118 (nearly 40.1%) more adders, but requires 2723 less memory words than the recently proposed parallel structure and performs the computation in nearly half the time of the other. Inspite of having more arithmetic components than the lifting-based structures, the proposed structure offers significant saving of area and power over the other due to substantial reduction in memory size and smaller clock-period. ASIC synthesis result shows that, the proposed structure for Daub-4 involves 1.7 times less area-delay-product (ADP) and consumes 1.21 times less energy per image- (EPI) than the corresponding best available convolution-based structure. It involves 2.6 times less ADP and consumes 1.48 times less EPI than the parallel lifting-based structure.

3.1.5 Optimization of wavelet decomposition for image compression and feature preservation

A neural-network-based framework has been

developed to search for an optimal wavelet kernel that can be used for a specific image processing task. In this paper, a linear convolution neural network was employed to seek a wavelet that minimizes errors and maximizes compression efficiency for an image or a defined image pattern such as micro calcifications in mammograms and bone in computed tomography (CT) head images. We have used this method to evaluate the performance of tap-4 wavelets on mammograms, CTs, magnetic resonance images, and Lena images. We found that the Daubechies wavelet or those wavelets with similar filtering characteristics can produce the highest compression efficiency with the smallest mean-square-error for many image patterns including general image textures as well as micro calcifications in digital mammograms. However, the Haar wavelet produces the best results on sharp edges and low-noise smooth areas. We also found that a special wavelet (whose low-pass filter coefficients are 0.32252136, 0.85258927, 0.38458542, and -0.14548269) produces the best preservation outcomes in all tested micro calcification features including the peak signal-to-noise ratio, the contrast and the figure of merit in the wavelet lossy compression scheme. Having analyzed the spectrum of the wavelet filters, we can find the compression outcomes and feature preservation characteristics as a function of wavelets. This newly developed optimization approach can be generalized to other image analysis applications where a wavelet decomposition is employed.

3.1.6 VLSI architecture for Daubenchies 4-tap and 6-tap wavelet filters using algebraic integers

This paper proposes a novel algebraic integer (AI) based multi-encoding of Daubechies-4 and -6 2-D wavelet filters having error-free integer-based computation. Digital VLSI architectures employing parallel channels are proposed, physically realized and tested. The multi-encoded AI framework allows a multiplication-free and computationally accurate architecture. It also guarantees a noise-free computation throughput the multi-level multi-rate 2-D filtering operation. A single final reconstruction step (FRS) furnishes filtered and down-sampled image outputs in fixed-point, resulting in low levels of quantization noise. Comparisons are provided between Daubechies-4 and -6 designs in terms of SNR, PSNR, hardware structure, and power consumptions, for different word lengths. SNR and PSNR improvements of approximately 30% were observed in favor of AI-based systems, when compared to 8-bit fixed-point schemes (six fractional bits). Further, FRS designs based on canonical signed digit representation and on expansion factors are proposed. The Daubechies-4 and -6 4-level VLSI architectures are prototyped on a Xilinx Virtex-6 vcx240t-1ff1156 FPGA device at 282 MHz and 146 MHz, respectively, with dynamic power consumption of 164 mW and 339 mW, respectively, and verified on FPGA chip using an ML605 platform.

3.2 COMPUTATIONS OF DISCRETE WAVELET TRANSFORMS

3.2.1 Computation of the 1-D DWT

According to the method for computing the 1-D DWT can be viewed as a sequence of operations along a binary tree consisting of a set of two-channel filter banks. Fig. 2.3 shows an example of a binary tree for a 3-level DWT computation of1-D signal s(n). It is seen from this figure that the decomposition at any level of the DWT is computed by using a two-channel filter bank consisting of one high pass filter GH(z)and one low pass filter GL(z), followed by a decimation operation by a factor of two in each channel. For a given resolution level j (j=1, 2, 3), the output samples of the two channels consist of a low pass component and a highpass component , ofwhich only the component is used as input for the decomposition at the next levelj+1. The computation for the resolution level j has a complexity of $O(N_0L/2^{j-1})$, where $N_0$and L are, respectively, the number of samples of the input signal s(n) and the length of each of the two filters. It should be noted that as the resolution level j increases, the dilation parameter of the wavelets associated with the resolution level j becomes smaller and smaller, which results in representing the signal by functions having a coarser scale.

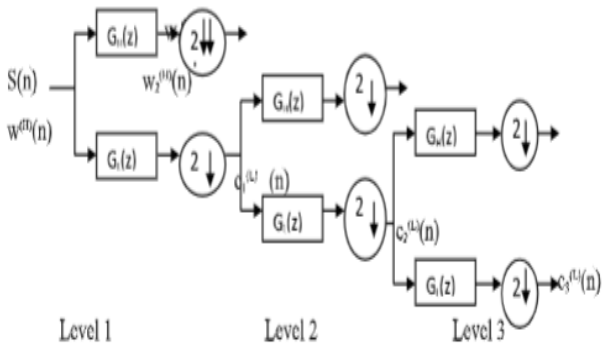

Figure 3.2.1: Binary tree representation of a 3-level 1-D DWT decomposition.

3.2.2 Computation of the 2-D DWT

The computation of the 2-D DWT is more involved than that of the 1-D DWT, both in terms of the amount of processing as well as the complexity of the algorithm used for the computation.(i) Separable Approach for the 2-D DWT Computation A straightforward way to perform the computation of the 2-D DWT is to use a separable approach. In the separable approach, the impulse response G(z1, z2) of each 2-Dfilter used for the DWT computation is product separable, i.e., G(z1, z2) = G1(z1)G2(z2),The filter G1(z1) is used to process the 2-D data of successive rows (columns). Then, the resulting 2-D data is processed successively along the columns (rows) using the filter G2(z2). A binary tree representation for a 2-level DWT computation of 2-D signal s(n1, n2)based on the separable approach is shown in Fig. 2.4. It is seen from this figure that the computation for the decomposition of a given level j consists of two decomposition steps :row-wise

decomposition of the 2-D input data using and column-wise decomposition of the 2-D data resulting from the row-wise decomposition. In the row-wise decomposition, each row of the 2-D input data is filtered using the two-channel horizontal filter bank(G_{1H}(z1) or G_{1L}(z1)) and then down-sampled by a factor of two, to produce horizontal high pass and low pass components, each component having one-half of the numbers of samples in the rows of the 2-D input data. In the column-wise decomposition, each column of the two resulting components is filtered by using the two-channel vertical filter bank (G_{2H}(z2) or G2L(z2)) and down-sampled by a factor of two so that in total four components, specified as the HH component , LH component ,HL component and LL component , are obtained as outputs ofthe given level j. Among the four outputs, only the LL component is used forthe computation of the next resolution level, which is an iteration of the above two steps.

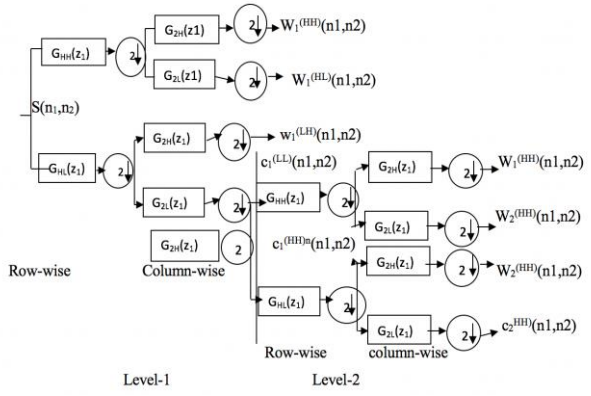

Figure 3.2.2: Binary tree representation of the computation of a 2-level 2-D DWT based on separable approach.

It should be noted that each of the four resulting components has one-quarter of the number of samples of the 2-D input data to the jth level. The computation of the resolution level j has a complexity of $O(N_0M_0L/4^{j-1})$, where N0 and M0 are the numbers of the rows and columns of the 2-D input data.(ii) Non-separable Approach for the 2-D DWT Computation Obviously, separable approach is a simple way to compute the 2-D DWT. However, separable filters being a special class of 2-D filters are not capable to approximate well all arbitrary frequency responses. In this regard, a non-separable approach of the 2-Dcomputation provides more flexibility. In the non-separable approach depicted in Fig.2.5, the DWT of a 2-D signal s, is computed by carrying out four separate 2-Dfiltering operations using four 2-D filters: a high pass-high pass (HH) filter G_{HH}(z1, z2), a high pass-low pass (HL) filter G_{HL}(z1, z2), a low pass-high pass (LH) filter G_{LH}(z1, z2), and a low pass-low pass (LL) filter G_{LL}(z1, z2). The output signals of these four filters are then decimated by a factor of two in the horizontal and vertical directions producing, respectively, the HH, HL, LH and LL components.

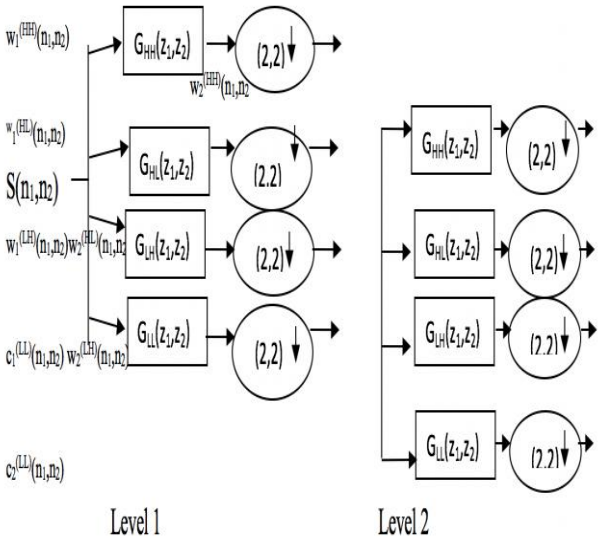

Figure 3.2.3: Representation of the computation of a 2-level 2-D DWT based on non-separable approach.

The computation of the resolution level j using the non-separable approach has a complexity of $O(N_0M_0L/4^{j-1})$,, where $N_0$and M0 are, respectively, the numbers of rows and columns of the 2-D input data, and L2is the number of coefficients in each of the L×L 2-D filters.2.2 Review of the Architectures.

3.2.4 Categorization of the Architectures

In recent years, many architectures have been proposed for the DWT computation. These architectures aim at providing high performances, in terms of their speed, area, throughput, latency and power consumption. The filtering operation involved in the DWT computation is usually the convolution operation, that is, FIR filtering. The structure of the filter could be a direct realization, or it could be a systolic, lattice, bitwise or lifting based realization depending on the way that the basic convolution operation is manipulated or formulated. For example, the lifting scheme proposed by Sweldens exploits the relationship that exists between the low pass filter GL and the high pass filter GH for the computation of the DWT.

The filtering operation is carried out by using a processor that employs a certain type of filter structure. An architecture may use one or multiple such processors to perform the DWT computation. For the purpose of reviewing these existing architectures, we categorize them as single-processor architectures, parallel-processor architectures and pipeline architectures, depending on their configuration and the number of processors used by them. In a single-processor architecture, only one processor carries out the filtering operation by computing the samples of the DWT in a recursive manner. In a parallel-processor architecture, multiple processors are used to carry out the filtering operations so that more than one sample is computed at a time. In this type of architecture ,the filtering operations to decompose the input signal into various

components are carried out in parallel, whereas the computations of various resolution levels are still performed recursively by the parallel processors. In a pipeline architecture, a certain number of stages, each consisting of one or more processors, are pipelined so that the computation of each decomposition level as well as that of the multiple resolution levels are performed in parallel. Fig. 2.6 depicts the block diagrams for the three categories of the architectures. In each of these three broad categories, architectures may differ considerably because of the internal structures of processors employed for the filtering operation. In the following, examples of 1-D and 2-D architectures are given for each of the categories, and their salient features discussed.

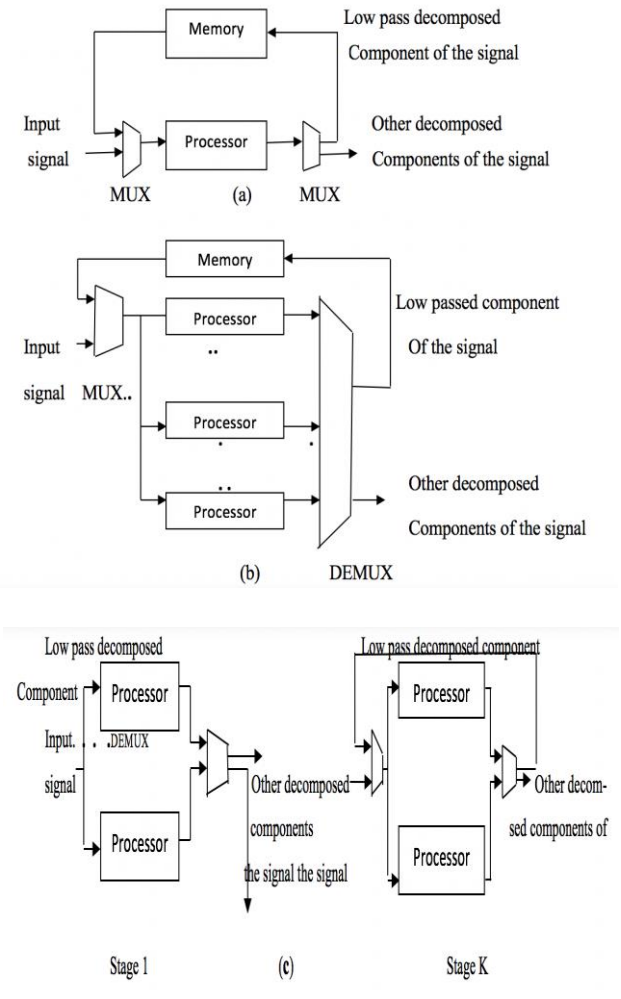

Figure 3.2.4: Block diagrams of three types of architectures. (a) Single-processor architecture, (b) parallel-processor architecture, and (c) pipeline architecture.

3.3.Introduction to verilog

Verilog HDL is one of the two most common Hardware Description Languages (HDL) used by integrated circuit(IC) designers. The other one is VHDL.

HDL's allows the design to be simulated earlier in the design cycle in order to correct errors or experiment with different architectures. Designs described in HDL are technology-independent, easy to design and debug, and are usually more readable than schematics, particularly for large circuits. Verilog can be used to describe designs at four levels of abstraction:

(i) Algorithmic level (much like c code with if, case and loop statements).

(ii) Register transfer level (RTL uses registers connected by Boolean equations).

(iii) Gate level (interconnected AND, NOR etc.).

(iv) Switch level (the switches are MOS transistors inside gates).

The language also defines constructs that can be used to control the input and output of simulation. More recently Verilog is used as an input for synthesis programs which will generate a gate-level description (anetlist) for the circuit. Some Verilog constructs are not synthesizable. Also the way the code is written will greatly effect the size and speed of the synthesized circuit. Most readers will want to synthesize their circuits, so non-synthesizable constructs should be used only for test benches. These are program modules used to generate I/O needed to simulate the rest of the design. The words "not synthesizable" will be used for examples and constructs as needed that do not synthesize. There are two types of code in most HDLs: Structural, which is a verbal wiring diagram without storage.

```
assign a=b & c | d; /* "|" is a OR */
assign d = e & (~c);
```

Here the order of the statements does not matter. Changing e will change a. Procedural which is used for circuits with storage, or as a convenient way to write conditional logic. always @(posedge clk) // Execute the next statement on every rising clock edge. count <= count+1;

Procedural code is written like c code and assumes every assignment is stored in memory until over written. For synthesis, with flip-flop storage, this type of thinking generates too much storage. However people prefer procedural code because it is usually much easier to write, for example, if and case statements are only allowed in procedural code. As a result, the synthesizers have been constructed which can recognize certain styles of procedural code as actually combinational. They generate a flip-flop only for left-hand variables which truly need to be stored. However if you stray from this style, beware. Your synthesis will start to fill with superfluous latches. This manual introduces the basic and most common Verilog behavioral and gate-level modeling constructs, as well as Verilog compiler directives and system functions. Full description of the language can be found in Cadence Verilog-XL Reference Manual and Synopsys HDL Compiler for Verilog Reference Manual. The latter emphasizes only those Verilog constructs that are supported for synthesis by the Synopsys Design Compiler synthesis tool. In all examples, Verilog keyword are shown in boldface. Comments are shown in italics.

IV.SIMULATION RESULTS

4.1.Simulation Results:

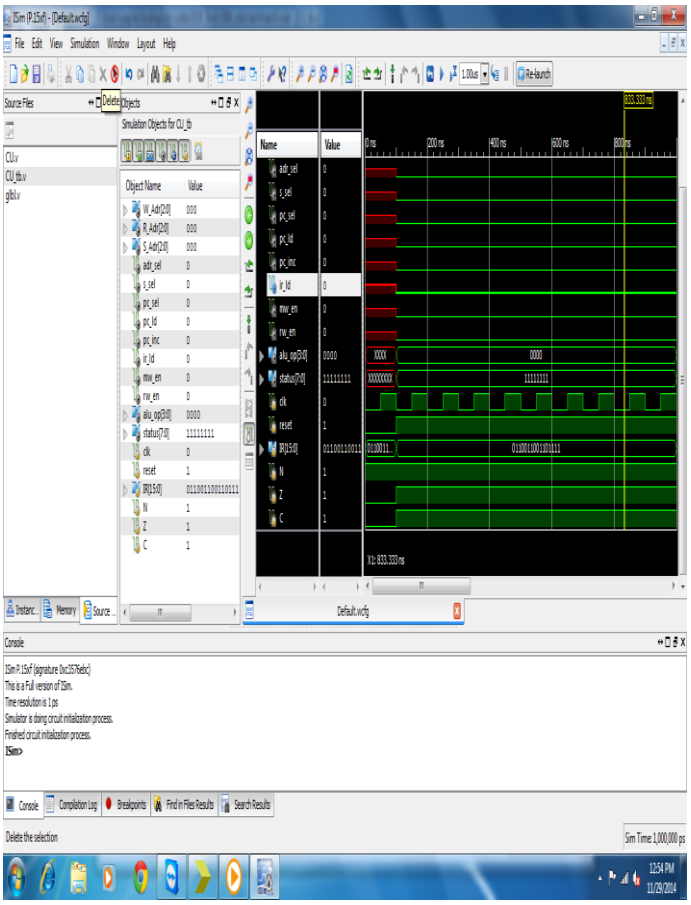

Fig:4.1.1.Simulation Results

The 16-bit input is given in the pixels. Output changes according to the input value.
The clock signals are also changed according to the input so that the changes in the power value can be estimated easily.

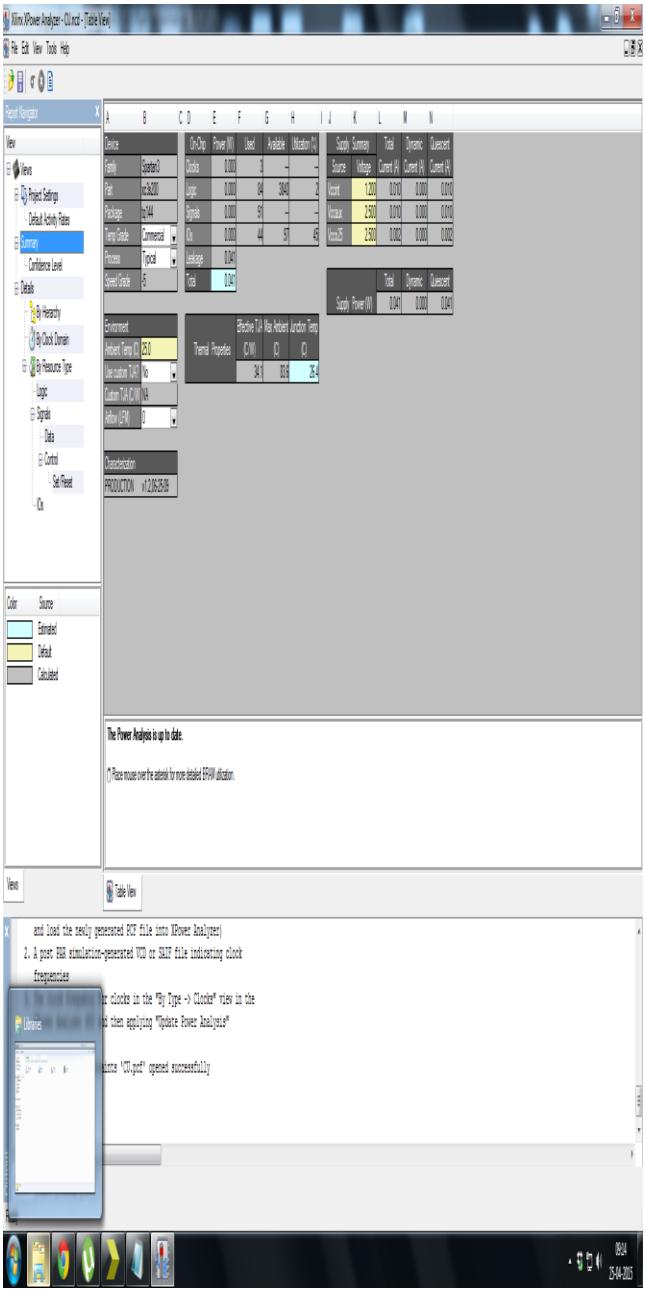

Fig:4.1.2.Simulation Results

The Power values are reduced when compared to the previous methods. The clocknet, dynamic & Quiescent power values are listed in the above fig.

4.2.Comparison of Power Values:

Power(watt)	Word-Length(bits)

	8	10	12	14	16
Clock net	0.091	0.104	0.112	0.128	0.412
Quiescent	4.271	4.272	4.272	4.273	4.273
Dynamic	0.304	0.319	0.337	0.354	0.371
Total	4.575	4.59	4.609	4.626	4.664

Fig:4.2.1. Comparison of Power(Watt)

The power values changes according to the 16-bit input value. The power values are reduced when compared to the previous methods. The values also changes based on the no. of components used in CSD method.

	Total	Dynamic	Quiescent
Supply Power(watt)	0.041	0.001	0.041

Fig:6.2.2. Comparison of Supply Power(Watt)

The supply power decreased when compared to the power in previous method. By using CSD approximation method no. of adders are also reduced .

V.CONCLUSION& FUTURE SCOPE

CONCLUSION:

We proposed a recursive architecture and a dual-scan architecture for computing the DWT based on the lifting scheme. In the previous chapters, we described the procedures for implementing 1-D and 2-D versions of the RA and DSA for calculating any lifting-based wavelet transforms. We also illustrated the details of the hardware design by describing implementations of the 1-D 9/7 RA, the 1-D Daub-4 DSA, the 2-D Daub-4 RA, and the 2-D Daub-4 DSA as examples.

Compared to previous implementations of the lifting-based DWT, the proposed architectures have higher, and hence more efficient, hardware utilization and shorter computation time. In addition, since the recursive architectures can continuously compute the DWT coefficients as soon as the input data become available, the memory size required for storing the intermediate results is minimized. Hence, the sizes and power consumptions of both the 1-D and 2-D recursive architectures are reduced compared to other implementations. In addition, since the designs are modular, they can be easily extended to implement the separable multi-dimensional DWT by cascading multiple basic 1-D DWT processors.

However, there are limitations in our architecture designs: each specific datapath can only calculate one type of lifting-based wavelet. Although it should be relatively straightforward to design new architectures following our procedure, and many image applications only use a few types of DWT, our architectures are not yet capable of providing the convenience and the generality of any lifting schemes. Theoretically, constructing general synthesis procedures for the RA and DSA architectures should be feasible, but that would be beyond the scope of our thesis research. The proposal of such general architectures is one possible direction for future work.

Another direction for future work could be to extend our architectures to implement the lifting scheme for multi-wavelet applications. Multi-wavelet analysis has been found promising in applications, such as image compression and de-noising . Recent research has revealed that multi-wavelets can also be constructed using the lifting scheme and any compactly supported multi-wavelet can be factored into lifting steps. Hence, it seems likely that our architectures could be modified to implement multi-wavelets.

FUTURE SCOPE:

In the digital signal processing we have discrete wavelet transform which g
number of LUTS in FPGA devices which further reduces
area and power consumption which are mainly considered
in VLSI.

VI.Referances:

[1] K. Wahid, V. Dimitrov, and G. Jullien, " Analysis of VLSI architectures for AI based 1-D/2-D Daub-6 wavelet Filter banks with low Adder count," *J. Circuits, Syst. Comp*.vol. 13, no. 6, pp. 1251–1270, 2004.

[2] K. A. Wahid, V. S. Dimitrov, G. A. Jullien, and W. Badawy, "An algebraic integer based encoding scheme for implementing Daubechies discrete wavelet transforms," in *Proc. Asilomar Conf. Signals, Syst.Comp.*, 2002, vol. 1, pp. 967–971.

[3] K. A.Wahid, V. S. Dimitrov, G. A. Jullien, andW. Badawy, "An analysis of Daubechies discrete wavelet transform based on algebraic integer encoding scheme," in *Proc. Third Int. Workshop DCV*, 2002, pp.27–34.

[4] G. Xing, J.Li, S. Li, andY.-Q.Zhang, "Arbitrarily shaped video-object coding by wavelet," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 10, pp. 1135–1139, 2001.

[5] S. Murugesan and D. B. H. Tay, "New techniques for rationalizing orthogonal and biorthogonal wavelet filter coefficients," *IEEE Trans. Circuits Syst.*, vol. 59, no. 3, pp. 628–637, Mar. 2011.

[6] J. Walker, *A Primer on Wavelets and their Scientific Applications*.Boca Raton, FL: Chapman & Hall/ CRC Press, 1999.[7] B. K. Mohanty, A. Mahajan, and P. K.Meher, "Area and power-efficient architecture for high-throughput implementation of lifting 2-D
DWT," *IEEE Trans. Circuits Syst.—II: Express Briefs*, vol. 59, pp. 434–438, 2012.

[8] B. K. Mohanty and P. K. Meher, "Memory-efficient high-speed convolution-based generic structure for multilevel 2-D DWT," *IEEE Trans.Circuits Syst. Video Technol.*, vol. 23, pp. 353–363, 2013.

[9] M. A. Islam and K. A. Wahid, "Area- and power-efficient design of Daubechies wavelet transforms using folded AIQ mapping," *IEEE Trans. Circuits Syst. II*, vol. 57, no. 9, pp. 716–720, Sep. 2010.

[10] K. A. Wahid, M. A. Islam, and S.-B. Ko, “Lossless implementation of Daubechies 8-tap wavelet transform,” in *Proc. IEEE Int. Symp. Circ. Syst.*, Rio de Janeiro, Brazil, May 2011, pp. 2157–2160.
[11] Y.Wu, R. J. Veillette, D. H. Mugler, and T. T. Hartley, “Stability analysis of wavelet-based controller design,” in *Proc. American Control Conf.*, 2001, vol. 6, pp. 4826–4827.
[12] K. A. Wahid, V. S. Dimitrov, and G. A. Jullien, “Error-free arithmetic for discrete wavelet transforms using algebraic integers,” in *Proc. 16th IEEE Symp. Comput. Arithmetic*, 2003, pp. 238–244.
[13] S.-C. B. Lo, H. Li, and M. T. Freedman, “Optimization of wavelet decomposition for image compression and feature preservation,” *IEEE Trans. Med. Imag.*, vol. 22, no. 9, pp. 1141–1151, 2003.
[14] M. Martone, “Multiresolution sequence detection in rapidly fading channels based on focused wavelet decompositions,” *IEEE Trans. Commun.*, vol. 49, no. 8, pp. 1388–1401, 2001.
[15] B. K. Mohanty and P. K. Meher, “Merged-cascaded systolic array
for VLSI implementation of discrete wavelet transform,” in *Proc. APCCAS*, 2006, pp. 462–465.

Ms.Mercy Parimala:- She completed her Master of Technology in Electronics and communication Engineering from Sri Krishna Devaraya University,Anantapur in the year 2015 with specialization in EMVL. She has given guidance to many students in their thesis work of M.Tech She has 2 years teaching Experience and presently working as Asst. Professor in Mekapati Rajamohan reddy Institute of Technology, Udayagiri,SPSR Nellore. She has done Bachelor's of Technology from JNTUA University in the year 2013 in Electronics and Communication Engineering.