

# An Efficient VLSI Architecture for Lifting-Based 2-D Discrete Wavelet Transform

*Shruthi. U, M.Tech, VLSI & ES Dept. Of Electronics and Communication, Reva ITM, Bangalore-560064*  
*Dr. Bharathi. S. H, Professor, Dept. of Electronics and Communication, Reva ITM, Bangalore-560064*

**Abstract**—A high speed and reduced-area 2-D discrete wavelet transform (2-D DWT) architecture is proposed and the design is simulated. Initially modifications are made to the lifting scheme, and the intermediate results are recombined and stored to reduce the number of pipelining stages. The architecture uses three basic 1D-DWT image processing elements. The proposed architecture only requires less registers between the row and column filters as the transposing buffer, and a higher efficiency can be achieved with reduced delay.

**Index Terms**—Discrete wavelet transform (DWT), flipping structure, lifting scheme, pipeline, VLSI architecture.

## I. INTRODUCTION

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. 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. The use of software implementation of DWT image compression provides flexibility for manipulation but it may not meet timing constraints in certain applications. Hardware implementation of DWT has practical obstacles. First, the high cost of hardware implementation of multipliers. Filter bank implementation of DWT contains two FIR filters. It has traditionally been implemented by convolution or the finite impulse response (FIR) filter bank structures. Such implementations require both large number of arithmetic computations and storage, which are not desirable for either

high speed or low power image/video processing applications.

Therefore a new approach called the lifting scheme based wavelet transform was first proposed by Sweldens based on a spatial construction of the second generation wavelet and a very versatile scheme for its factorization has been suggested in Sweldens[1]. The lifting scheme has many advantages over the previous approaches. In particular, all the interesting properties of wavelets, such as bi-orthogonality and regularity, are defined by linear relationships between the filter bank coefficients. As a consequence, it is easier to design wavelet filters. Unlike convolution wavelets, lifting scheme does not depend on Fourier transform of the wavelets. As a consequence, wavelets can be designed on arbitrary lattices in spatial domain. Since the lifting scheme makes optimal use of similarities between the high and low pass filters to speed up the calculation of wavelet transform, it has been adopted in the image processing techniques.

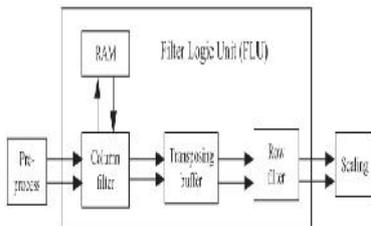
By recombining the intermediate results of the row and column transforms, the number of pipelining stages and registers is reduced. A novel architecture is developed to implement the 2-D DWT based on the modified scheme evaluated using the previous algorithm equations[4]. The parallel scanning method is employed to reduce the size of the transposing buffer.

## II. PROPOSED ARCHITECTURE FOR THE 2-D DWT

### A. Overall architecture

Based on the proposed modified algorithm, a novel architecture for the 2-D DWT shown in Fig. 8 is proposed. First, the serial-parallel conversion for the original data in the

preprocessing module is carried out. After that, data are sent into the column filter for the column transform. Next, the output data of the column filter are sent into the transposing buffer, where the data transposition is operated to meet the order of the data flow required by the row filter. Then, the row filter begins to read the data from the transposing buffer for the row transform. Finally, the scaling module is used to finish the scaling computation.



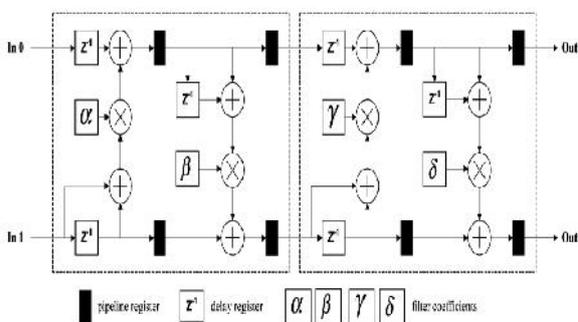
**Fig. 8. Proposed Architecture Of 2D-DWT.**

Column filter uses one processing element and Row filter uses two 1D-lifting processing elements. In overall proposed architecture three such 1D lifting processing elements are used to perform 2D-DWT.

**B. 1d-processing element**

The architecture of the proposed 1-D PE is shown in Fig. 6. This architecture can be applied in the column and row filters by selecting the RAM or Buffer properly. In order to reduce the size of the transposing buffer between the column and row filters and to improve the processing speed, our design adopts the architecture of two-input/two-output.

The  $\alpha$ ,  $\beta$ ,  $\gamma$ , and  $\delta$  values are extracted from the intermediate values obtained for the input image pixel values.



**Fig6: 1D Processing Element.**

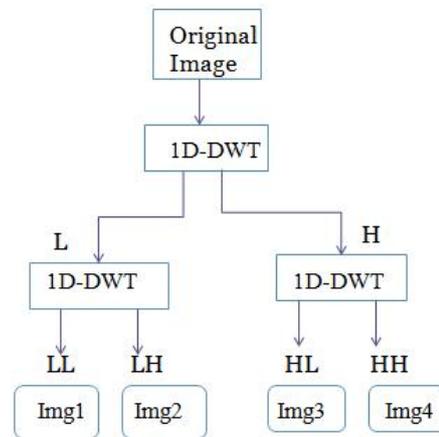
Column processor uses one such 1D-DWT processing element and Row processor uses two such

1D-DWT processing elements. These two processors are used together to perform 2D-DWT.

**III. DESIGN SIMULATION**

The hardware architecture was discussed previously, now let us see how this is designed to simulate in the provided software tools. The figure 7.1 below shows the implementation flow of simulation. The proposed architecture is simulated in Modelsim Altera 6.3g version simulator and Matlab is used to convert image to text and text to image.

Initially the image is read using the tool MATLAB, here the original image file will be converted into text file for further processing, the conversion involves some basic MATLAB functions which read the image pixel values, image intensity in RGB format and converts it into GRAY format with proper resizing. Here MATLAB reads the image pixel by pixel in each row and each column of the image data, and converts the pixel value from decimal to hexadecimal value, to perform further processing in digital form.



**Fig.7.1: Flow of Data Processing in Simulation**

The image pixel values read by the MATLAB are extracted to the MODELSIM tool for further simulation. Initially the 1D-DWT is performed on the converted pixel values to obtain the lower and higher frequency components of the original image, these are called as L (low) and H (high) components of the original image. This is the first level of decomposition of original image. Only one 1D-DWT element is used to perform the first level of decomposition.

The lower and higher frequency image components (L and H) obtained in first level of decomposition is used to perform the second level of decomposition to obtain 2D-Discrete Wavelet Transformed image, which is the aim of the project. The second level of decomposition uses two 1D-DWT performing blocks as shown in the figure 7.1 above, for each lower and higher frequencies of first level decomposition. Performing 1D-DWT is nothing but extracting lower and higher frequency components of the original image, and 2D-DWT is again extracting lower and higher frequency components from the first level decomposition components.

The lower frequency and higher components are fed to two separate 1D-DWT performing blocks. When we perform 1D-DWT on the L (lower) frequency image components we obtain the two second level decomposed image components, called as LL (Horizontal Low Frequency – Vertical Low Frequency) frequency component and LH (Horizontal Low Frequency – Vertical High Frequency) frequency component.

Similarly when we perform 1D-DWT on the H (higher) frequency image component of first level decomposition, we obtain two more image components of second level decomposition. These image components are called as HL (Horizontal High Frequency – Vertical Low Frequency) and HH (Horizontal High Frequency – Vertical High Frequency) frequency components. These four i.e. LL, LH, HL and HH frequency components are the final 2D-DWT frequency components.

These four frequency components are read in MATLAB to convert it to image. Here the text file of frequency components is converted to image file. MATLAB uses some predefined functions to convert text into image. Finally the four images are concatenated into single image for better differentiation.

## IV.SIMULATION RESULTS

The proposed architecture is simulated in Modelsim Altera 6.3g version simulator and Matlab is used to convert image to text and text to image. The simulation waveforms, the input and output images of the proposed design are as explained and shown below.

### A. Simulation Waveforms

The output waveforms obtained after simulating the design in MODELSIM ALTERA 6.3g are as shown in figure below for 256x256 pixel values.



### B. Input Image

Input image taken to perform 2D DWT:



**Input Image (original image)**

### C. Output images after performing 2D DWT:



**Horizontal low & Vertical Low (LL) [image1]**



**Horizontal low & Vertical High (LH) [image2]**



**Horizontal High & Vertical Low (HL) [image3]**



**Horizontal High & Vertical High (HH) [image4]**

**D. Design Summary**

The following table gives the estimated values for device utilization; the device used is 6slx45csg324-3.

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	117	54576	0%
Number of Slice LUTs	1509	27288	5%
Number of fully used LUT-FF pairs	78	1548	5%
Number of bonded IOBs	105	218	48%
Number of BUFG/BUFGCTRLs	1	16	6%

Total memory usage is 247264 kilobytes

Timing Summary

Speed Grade: -3

Minimum period: 24.358ns (Maximum Frequency:

41.055MHz)

Minimum input arrival time before clock: 5.790ns

Maximum output required time after clock: 35.716ns

Maximum combinational path delay: No path found

Delay: 24.358ns

**V.CONCLUSION**

The project work presented a novel architecture for the 2-D DWTs. The modified one lifting step circuit can work within three pipelining stages with fewer registers, and the low critical path delay. The proposed architecture is designed and simulated using two tools, **MODELSIM ALTERA 6.3g** version simulator and **MATLAB 7.10.0 (R210a)**. Matlab tool is used to read the original image and for conversion of image from image to text and text to image. Entire simulation is done in tool Modelsim Altera. According to the results, the total memory usage is 247264 kilobytes and the total delay is 24.358ns with a speed grade of -3. Therefore the proposed architecture can achieve high speed with lower hardware complexity and smaller storage size compared to previous architectures.

**REFERENCES**

[1] W. Sweldens, "The new philosophy in biorthogonal wavelet constructions," in *Proc. SPIE.*, 1995, vol. 2569, pp. 68–79.

[2] I. Daubechies, W. Sweldens,(1998), " Factoring wavelet transform into lifting steps", *J. Fourier Anal. Appl.* 4, 247–269.

[3] S.Mallat ,(1989), "A theory for multiresolution signal decomposition: the wavelet representation", *IEEE Trans. Pattern Anal. Mach. Intell.* 11, 674–693.

[4] C.-T. Huang, P.-C. Tseng, and L.-G. Chen, "Flipping structure: An efficient VLSI architecture for lifting-based discrete wavelet transform," *IEEE Trans. Signal Process.*, vol. 52, no. 4, pp. 1080–1089, Apr. 2004.