

# Vlsi Architecture For Pipelined Lifting Based 2-D Dwt With Booth Multiplier

G.Lavanya<sup>1</sup>, Saswata Maitra<sup>2</sup>, N.V.Maheswara Rao<sup>3</sup>

<sup>1</sup>M.Tech. student, Dept. of ECE, GVP College of Engg for Women, Visakhapatnam, A. P., India

<sup>2</sup>Scientist- 'D' in Directorate of Navigation & Embedded Computers, RCI, DRDO, at Hyderabad

<sup>3</sup>Assistant Professor, Dept. of ECE, GVP College of Engg for Women, Visakhapatnam, A. P., India

## Abstract

*In this paper, a scheme for the design of area efficient and high speed pipelined VLSI architecture for the computation of pipelined 2-d discrete wavelet transform using lifting scheme with booth multiplier is proposed. The main focus of the scheme is to reduce the number and period of clock cycles and efficient area with little or no overhead on hardware resources. The pipelining architecture speeds up the clock rate of DWT and the booth multiplier reduces the number of partial products as compared to normal multiplier. The 2-dimensinal discrete wavelet transform lifting scheme algorithm has been implemented using MATLAB program and the architecture has been coded in verilog HDL on Xilinx platform. The proposed scheme requires the least computing time for pipelined 2-D DWT and achieves the less area for implementation, compared with other architectures. So this architecture is realizable for real time processing of DWT computation applications.*

## 1. Introduction

The advantages of the wavelet transform over conventional transforms, such as the Fourier transform, are now well recognized. Because of its excellent locality in time-frequency domain, wavelet transform is remarkable and extensively used for signal analysis, compressing and denoising. Since the DWT has been increasingly used in many different areas of science and engineering mainly because of the multi resolution decomposition property of the transformed signals.

In this paper, a scheme for the design of pipeline architecture for a fast computation of the DWT is developed. The goal of fast computation is achieved by minimizing the number and period of clock cycles. The proposed architecture achieves the one-multiplier delay constraint but uses less internal memory compared to the related architectures. Moreover, the proposed architecture implements the 9/7 filters by cascading the three

main components.

Due to recent advances in the technology, implementation of the DWT on Field Programmable Gate Array (FPGA) and Digital Signal Processing (DSP) chips has been widely developed. Based on [4], the main challenges in the hardware architectures for 1-D DWT are the processing speed and the number of multipliers. The number of multipliers in each pipeline stage determines the clock speed of the structure.

## 2. Literature Review

Robi polikar [1] "The Wavelet Tutorial" the author has explained wavelet theory in four parts. Part 1 contains overview: why wavelet transforms, part 2 fundamentals: the Fourier transform and the short term Fourier transform, resolution problems, part 3 introduce multi resolution analysis: the continuous wavelet transform and the part 4 is on multi resolution analysis: the discrete wavelet transform and he mentioned why we use it in signal processing and image processing.

S.G.Mallat [2] proposed "A Theory for Multi resolution Signal Decomposition: The Wavelet Representation". In this paper we shows that the wavelet theory recently developed by the mathematician Y.Meyer enables us to understand and model the concepts of resolution and scale. The author proved that this difference of information can be computed by decomposing the signal on a wavelet orthonormal basis and that it can be efficiently calculated with pyramid transform.

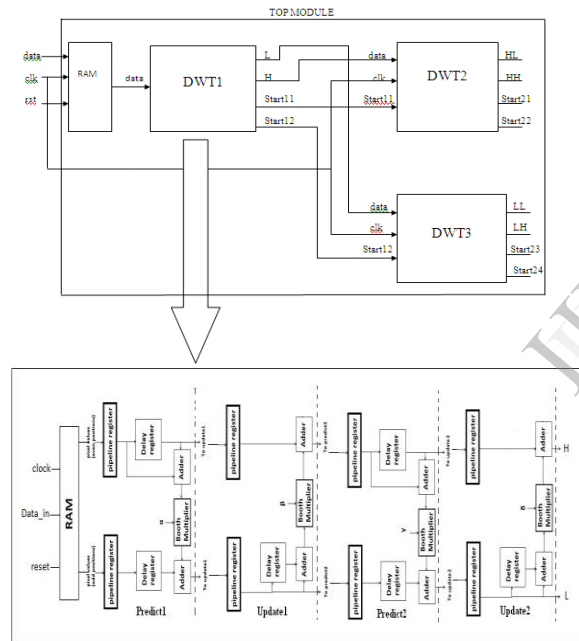
I. Daubechies and W. Sweldens [3] "Factoring Wavelet Transforms into Lifting Steps," They have discussed how any discrete wavelet transform or two band sub band filtering with finite can be decomposed into a finite sequence of simple filtering steps, called lifting steps. This decomposition corresponds to a factorization of the polyphase matrix of the wavelet or sub band filters

into elementary matrices.

C.T. Huang, P.C. Tseng, L.G. Chen [6], “Flipping Structure: An Efficient VLSI Architecture for Lifting Based Discrete Wavelet Transform,” It can provide a variety of hardware implementations to improve and possibly minimize the critical path as well as the memory requirement of the lifting based discrete wavelet transform by flipping conventional lifting structures. The precision issues are also analyzed.

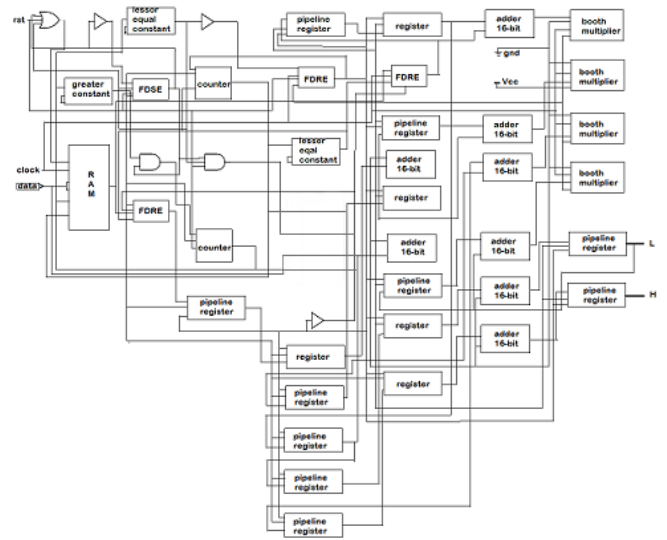
### 3. Proposed Architecture

The structure processes all input samples that arrive in pairs at consecutive clock pulses and the results for each pair are ready after five cycles.



**Fig.1. Top Module of Proposed 2D DWT Architecture**

However, due to the pipelined structure, the clock frequency is higher than that of parallel architectures. There is a trade-off between the clock speed and the number of pipeline stages. Fig.1. shows the Top module of proposed architecture for the pipelined DWT computation.



**Fig.2. Architecture of DWT**

The internal diagram of DWT is shown in fig.1. Where the DWT consists of Predict 1, Update 1, Predict 2 & Update 2 the internal blocks are shown in fig.1. Where the coefficients are given as the one of the input to the booth multiplier. The design of this architecture mainly focused on the booth multiplier for reduce the number of partial products as compared to normal multiplier.

#### Booth Multiplier

Conventional array multipliers, like the Braun multiplier and Baugh Woolley multiplier achieve comparatively good performance but they require large area of silicon, unlike the add-shift algorithms, which require less hardware and exhibit poorer performance. The Booth multiplier makes use of Booth encoding algorithm in order to reduce the number of partial products by considering two bits of the multiplier at a time, thereby achieving a speed advantage over other multiplier architectures.

**Table.1 Booth Algorithm**

$X_i$	$X_{i-1}$	Operations	Comments	$Y_i$
0	0	Shift only	String of zeros	0
1	0	Sub and shift	Beg of string of ones	1
1	1	Shift only	String of zeros	0
0	1	Add and shift	End of string of ones	1

This algorithm is valid for both signed and unsigned numbers. It accepts the numbers in 2's complement form based on radix-2 computation. In this five stage pipeline structure for 2D DWT, all stages need to share the computation. Hence, all the stages need to be synchronized with one the design of this architecture mainly focused on the booth multiplier for reduce the number of partial products as compared to normal multiplier and pipeline structure to speed up the system.

#### 4. 2D-DWT Computation

The architecture can be applied to implement the lifting-based 2-D DWT. As the DWT intrinsically constitutes a pair of filtering operations, a unified representation of the polyphase matrix is introduced as follows:

$$P(z) = \prod_{i=0}^m \begin{pmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{pmatrix}$$

Where  $h(z)$  and  $g(z)$  stand for the transfer functions for the low pass and high pass filter banks, respectively, and all suffixes e and o in the literature correspond to even and odd terms, respectively. Thus, the transform is symbolized with the equation

$$\begin{pmatrix} \lambda(z) & \gamma(z) \end{pmatrix} = \begin{pmatrix} x_e(z) & z^{-1}x_o(z) \end{pmatrix} P(z)$$

With  $\lambda(z)$  and  $\gamma(z)$  signifying the filtered low pass and high pass parts of the input  $x(z)$ .

The lifting scheme factorizes the polyphase representation into a cascade of upper and lower triangular matrices and a scaling matrix which subsequently return a set of linear algebraic equations in the time domain bringing forth the possibility of a pipelined processor.

For instance, the common Daubechies (9, 7) filter bank can be factorized as

$$P(z) = \begin{pmatrix} 1 & \alpha(1+z^{-1}) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \beta(1+z) & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1/\tau \end{pmatrix} \begin{pmatrix} 1 & \gamma(1+z^{-1}) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \delta(1+z) & 1 \end{pmatrix}$$

The related algebraic equations are

$$s_i^0 = x_{2i} \quad \text{(Splitting)}$$

$$d_i^0 = x_{2i+1} \quad \text{(Predict P1)}$$

$$d_i^1 = d_i^0 + \alpha \times (s_i^0 + s_{i+1}^0) \quad \text{(Update U1)}$$

$$s_i^1 = s_i^0 + \beta \times (d_{i-1}^1 + d_i^1) \quad \text{(Predict P2)}$$

$$d_i^2 = d_i^1 + \gamma \times (s_i^1 + s_{i+1}^1) \quad \text{(Update U2)}$$

$$s_i^2 = s_i^1 + \delta \times (d_{i-1}^2 + d_i^2) \quad \text{(Scaling S1)}$$

$$s_i = \zeta \times s_i^2 \quad \text{(Scaling S2)}$$

$$d_i = \left(\frac{1}{\zeta}\right) \times d_i^2 \quad \text{(Scaling S2)}$$

Where  $\alpha = -1.586134342$ ,  $\beta = -0.05298011854$ ,  $\gamma = 0.8829110762$ ,  $\delta = 0.4435068522$ , and  $\zeta = 1.149604398$  [16], and also  $0 \leq i \leq L-1$ ,  $L$  is the data length.

Following are the modification of above equations, the final equations are

$$s_i^0 = x_{2i} \quad \text{(Splitting)}$$

$$d_i^0 = x_{2i+1} \quad \text{(Predict P1)}$$

$$d_i^1 = A \times d_i^0 + (s_i^0 + s_{i+1}^0) \quad \text{(Update U1)}$$

$$s_i^1 = B \times s_i^0 + \frac{(d_{i-1}^1 + d_i^1)}{16} \quad \text{(Predict P2)}$$

$$d_i^2 = C \times d_i^1 + \frac{(s_i^1 + s_{i+1}^1)}{2} \quad \text{(Update U2)}$$

$$s_i^2 = D \times s_i^1 + \frac{(d_{i-1}^2 + d_i^2)}{2} \quad \text{(Scaling S1)}$$

$$s_i = K0 \times s_i^2 \quad \text{(Scaling S2)}$$

$$d_i = K1 \times d_i^2 \quad \text{(Scaling S2)}$$

where  $A = (1/\alpha) = -0.630463$ ,  $B = (1/16\alpha\beta) = 0.743750$ ,  $C = (1/32\beta\gamma) = -0.668067$ ,  $D = (1/4\gamma\delta) = 0.638443$ ,  $K0 = (64\alpha\beta\gamma\delta) = 2.590697$  and  $K1 = (32\alpha\beta\gamma/\delta) = 1.929981$  (up to six fractional digits) and also  $0 \leq i \leq L-1$ ,  $L$  is the data length.

The critical path delay for the above normal algebraic lifting equations is  $5T_m + 8T_a$ , where  $T_m$  and  $T_a$  denote the multiplier and adder delay, respectively. The primary reason behind this large delay is stacking of multipliers from the inputs to outputs. This scales the delay down to  $3T_m + 4T_a$ . Again this multiplier is replaced with Booth multiplier to reduce the number of partial products there by the area and delay will automatically reduced.

The five pipeline stages are used to improve the processing time, but the critical path is still restricted by the computation of predictor or updater (i.e., two adders and one multiplier propagation delay).

### 5. Simulation Results

The original image data word length is 8-bits per pixel, this amount of bits is not enough for wavelet transform coefficients. It is observed that the retrieved image in inverse discrete wavelet transform has distortion, because of overflow condition. The overflow occurs when the addition operation result may be larger than can be held in the word length being used. In DWT the word length of wavelet coefficients will grow gradually in FDWT in each DWT level, and as the DWT levels are more. To select the appropriate word length, the MATLAB program is used for writing appropriate code to simulate the (9/7) wavelet filter in (FDWT). With (FDWT) MATLAB program the Lena picture of size (512×512×8) is used as original input image, it is analyzed for three levels of decomposition by using different word lengths, as shown in the Fig.3.

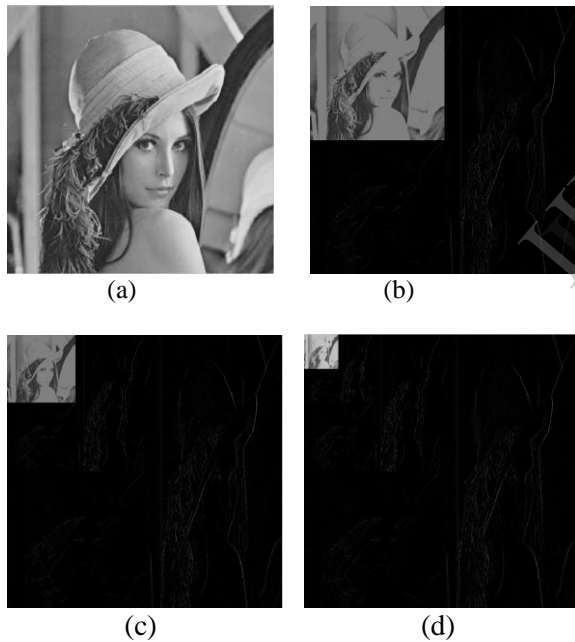


Fig.3.(a) original image, (b) one decomposition level, (c) two decomposition levels (d) three decomposition levels.

#### Simulation results for proposed architecture using ModelSim XE III 6.4b and Xilinx ISE 10.1

In order to evaluate the performance of the architecture resulting from the proposed scheme, we need to make use of certain metrics that

characterize the architecture in terms of the hardware resources used and the computation time. A pipelined architecture is implemented and the simulation result of that is shown in below figures.

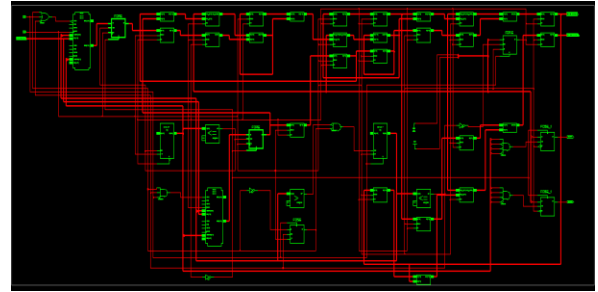


Fig.4. RTL Schematic of Proposed Design

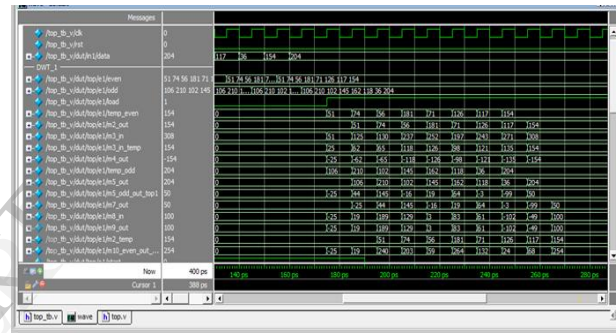


Fig.5.Simulation Results for First Decomposition Level of DWT

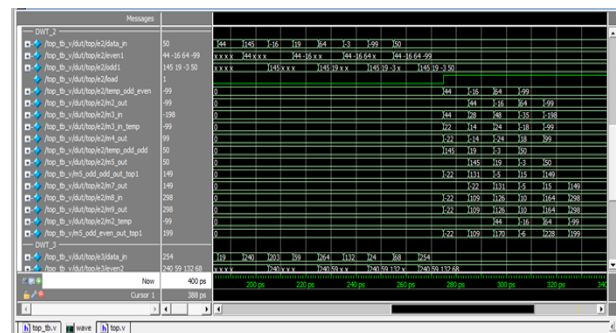


Fig.6. Simulation Results for Second Decomposition Level of DWT

The hardware resources used for the filtering operation are measured by the number of multipliers and the number of adders, and that used for the memory space and pipeline latches is measured by the number of registers. The hardware resources utilization is shown in table 2.

**Table 2 Hardware Resources of Proposed 2D DWT Pipelined Architecture**

Resource	Used	Available in Total	Percentage used
CLB Slices	246	4656	5%
Flip Flop Slices	424	9312	4%
4-input LUTS	216	9312	2%
Bonded IOBs	42	190	22%

```

Timing Summary:
-----
Speed Grade: -4

Minimum period: 4.014ns (Maximum Frequency: 249.128MHz)
Minimum input arrival time before clock: 3.858ns
Maximum output required time after clock: 4.283ns
Maximum combinational path delay: No path found

Timing Detail:
-----
All values displayed in nanoseconds (ns)

-----
Timing constraint: Default period analysis for Clock 'clk'
Clock period: 4.014ns (frequency: 249.128MHz)
Total number of paths / destination ports: 643 / 301

-----
Delay:          4.014ns (Levels of Logic = 0)
Source:         e1/m14/Mshreg_P_8 (FF)
Destination:    e1/m14/P_8 (FF)
Source Clock:   clk rising
Destination Clock: clk rising

Data Path: e1/m14/Mshreg_P_8 to e1/m14/P_8

```

**Fig.8. Timing Constraints for Proposed 2D DWT Computation****Table 3 Comparison of Various 2 D DWT Architectures**

Architecture	T <sub>c</sub> (ns)
Parallel	17.8
Systolic	11.8
Pipelined	11.8
DRU	10.2
IP core	11.8
Pipeline with parallelism	8.7
fixed point 1D DWT	8.280
proposed	4.014

## 6. Conclusion

In order to assess the effectiveness of the proposed scheme, pipeline architecture has been designed and simulated. The simulation results have shown that the architecture designed based on the proposed scheme requires the smallest number of clock cycles to compute output samples and a reduction of at least 60% in the period of the clock cycle in comparison to those required by the other architectures with a comparable hardware

requirement. Finally the principle of pipelining architecture using lifting scheme with the help of booth multiplier was presented in this paper for the design of architecture for the 2-D DWT computation is extendable to that for the 3-D DWT computation using modified Booth Multiplier.

## Acknowledgement

We are thankful to the Research Center Imarat (RCI), DRDO, Hyderabad and the Head of the Department, Electronics and Communications Engineering, GVP College of Engg for women, Visakhapatnam and all others who are involved in completing this project.

## References

- [1] Robi Polikar, "The Wavelet Tutorial". <http://engineering.rowan.edu/~polikar/WAVELETS/Wttutorial.html>
- [2] S.G.Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, pp. 674–693, July 1989.
- [3] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," J. Fourier Anal. Appl., vol. 4, no. 3, pp.245–267, 1998.
- [4] Lan, N. Zheng, Y.Liu, "Low power and high-speed VLSI architecture for lifting-based forward and inverse wavelet transform." IEEE Trans. Consumer Electron. 51 (2), 379–385 (2005).
- [5] T. C. Denk and K. K. Parhi, "Systolic VLSI architectures for 1-D discrete wavelet transforms," in Signals, Systems & Computers, 1998. Conference Record of the Thirty-Second Asilomar Conference on, vol. 2, (Pacific Grove, CA), pp. 1220–1224, Nov. 1998.
- [6] C.T. Huang, P.C.Tseng, L.G. Chen, "Flipping structure: an efficient VLSI architecture for lifting based discrete wavelet transform", IEEE Trans. Signal Process. 52 (2004), pp.1080–1089.
- [7] M. Vishwanath, "The recursive pyramid algorithm for the discrete wavelet transform," Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on], vol. 42, pp. 673–676, Mar. 1994.
- [8] Durga Sowjanya, K N H Srinivas and P Venkata Ganapathi "FPGA Implementation of Efficient VLSI Architecture for fixed point 1D DWT using Lifting scheme", International Journal of VLSI design & Communication Systems (VLSICS) Vol.3, No.4, August 2012.