

VLSI ARCHITECTURE FOR LIFTING BASED DISCRETE WAVELET TRANSFORM

DHANABAL R¹, BHARATHI V², HYMA SOMAN³, SAIRA SALIM⁴

^{#1}Assistant Professor (Senior Grade) ,VLSI Division,SENSE, VIT University,

^{*2}Assistant Professor,GGR College of Engineering ,Vellore,

^{#3,4}MTECH Students ,Sense department,VIT University, Vellore- 632014,Tamil Nadu, India

E-mail: rdhanabal@vit.ac.in, bharathiveerappan@yahoo.co.in, hymasoman@gmail.com, salimsaira@gmail.com

ABSTRACT

A lifting based 2D DWT with efficient folded architecture and parallel scanning is being proposed. The architecture results in lesser hardware complexity and memory requirement due to multiplexing of 2 stages of lifting architecture. The 2D DWT architecture is realized by cascading two 2D processing elements. The coefficients for the lifting stage were chosen according with 9/7 filter. The 1D processing element has a column filter, transposing buffer and a row filter in it. The use of parallel scanning reduces the size of transposing buffer. Combining the intermediate results of row and column, the number of pipelining stages and registers are also reduced. The throughput obtained are 2 input and 2 output per cycle. The critical path for proposed architecture is one T_m .

Keywords: DWT, VLSI, Wavelet Transform, Architecture

1. INTRODUCTION

Discrete Wavelet Transform enables the decomposition of images into different sub bands both in frequency and time domain. Hence it enables multiresolution solution and thereby giving high compression ratios. DWT has an upper hand over traditional Discrete Cosine Transform and thus its being widely used in signal processing and image processing areas.

The first generation DWT used convolution based techniques for image compression which had less quality [1][2]. The second generation DWT uses lifting schemes with different techniques for the same [3][4]. The lifting scheme is preferred over convolution, because of its lesser memory requirement and lesser complexity. Several architectures were proposed with respect to lifting based scheme like efficient folded architecture, flipping architecture, pipelined architecture etc. The throughput of each architecture varies and critical path also varies.

The efficient folded architecture EFA [5] is having lesser hardware complexity. But the critical path delay calculated is found to be $T_m + T_a$. Delay of multiplier is taken as T_m and

delay of adder is taken as T_a . Thus the critical path computation time is very high in EFA. Similarly the flipping structure also had longer computation time and with large temporal buffer. [6] Also the pipelined architecture gives high throughput of 2 input and 2 output enabling parallel processing. But the number of registers are high.[7]

2. LIFTING BASED DISCRETE WAVELET TRANSFORM

The lifting DWT consists of generally 3 stages- split, predict and update. During split stage the given data is divided to smaller levels O, E. We decompose the given data to 2 sets such that the O component has some useful information compared to E. In the predictor stage we use L to predict H component based on correlation between the two. The polyphase matrix of DWT has upper and lower triangular matrices with diagonal matrix. The polyphase matrix is shown as fig 1

$$P(z) = \begin{bmatrix} L_c(z) & L_o(z) \\ H_e(z) & H_o(z) \end{bmatrix}$$

The $L(z)$ and $H(z)$ forms complementary filter pairs namely the low frequency and high

frequency components respectively. P(z) can be factorized to

$$P(z) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & t(z)^{-1} \\ 0 & K \end{pmatrix} \begin{pmatrix} 1/K \\ - \end{pmatrix}$$

The lifting steps in detail are

- a) Split step: The data is given into odd and even samples.

$$o_i^0 = x(2n+1)$$

(1)

$$e_i^0 = x(2n)$$

(2)

- b) Predict and Update

I lifting stage:

$$o_i^1 = o_i^0 + \alpha (e_i^0 + e_{i+1}^0) \text{ predict}$$

(3)

$$e_i^0 = e_i^1 + \beta (o_i^0 + o_{i+1}^0) \text{ update}$$

(4)

II lifting stage:

$$o_i^1 = o_i^0 + \gamma (e_i^0 + e_{i+1}^0) \text{ predict}$$

(5)

$$e_i^0 = e_i^1 + \delta (o_i^0 + o_{i+1}^0) \text{ update}$$

(6)

- c) Scaling:

$$o_i = 1/K * o_i^2$$

(7)

$$e_i = K * e_i^2$$

(8)

and row processing element. The concatenation of predict and update stages reduces the hardware requirement. The equations (4-6) and adjusted suitably to get the format

$$1/\alpha o(i) = x(2i+1) + x(2i) + x(2i-1) \quad (9)$$

$$1/\beta e(i) = x(2i) + o(i-2) + o(i) \quad (10)$$

$$1/\gamma \text{High}(i) = 1/\gamma o(i) + e(i) + e(i+2) \quad (11)$$

$$1/\delta \text{Low}(i) = 1/\delta e(i) + \text{High}(i) + \text{High}(i-2) \quad (12)$$

Now considering equations 9 and 10 and doing associative rule

$$\begin{aligned} 1/\alpha\beta e(i) &= 1/\alpha\beta x(2i) + 1/\alpha o(i) + 1/\alpha o(i-2) \\ &= [(1/\alpha\beta + 1)x(2i) + 1/\alpha x(2i-1) + x(2i-2)] \\ &\quad + [1/\alpha x(2i+1) + x(2i) + x(2i+2)]. \end{aligned}$$

Thus the coefficients are obtained for multiplying in two stages of DWT. Here we use a single hardware to implement the 2 D DWT as the coefficients are being selected using multiplexer to switch between Ist lifting stage and II lifting stage. The coefficients multiplied during I stage are (1/α and 1/αβ + 1) and during II stage (1/γ and 1/γδ + 1). Thus we get efficient compression. A schematic representation of the proposed architecture I shown in fig b. The input image is fed to a column processing element and a transposing buffer is used to switch data between the row processing element. The column and row processing element are 2 dimensional and directly 2 lifting stages are performed. The inclusion of RAM is used to store image temporarily and retrieve when required. Here we obtain output in parallel. We feed 2 input at a time and we also obtain 2 output thus it performs 2 input output parallel processing. Also pipeline stages are included for reducing the critical path to one multiplier.

3. PROPOSED LIFTING SCHEME.

In the proposed lifting architecture, the predict and update stages are merged in the 9/7 filter. 2D DWT is implemented using a column

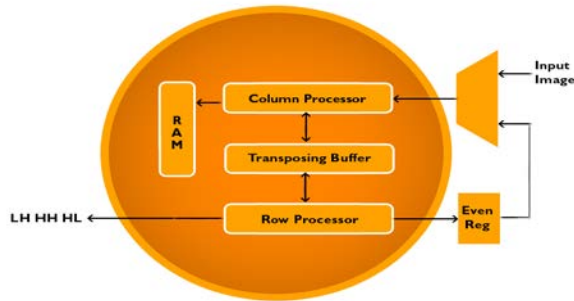


Fig b. Block diagram of 2 D Lifting DWT

Here after one lifting stage, only low pass filter components are fed back and 2nd level compression is done. By increasing the number of feedback of low pass components we obtain multi resolution of image.

First we have a raw image which is divided into two parts using correlation technique into odd and even components. It is

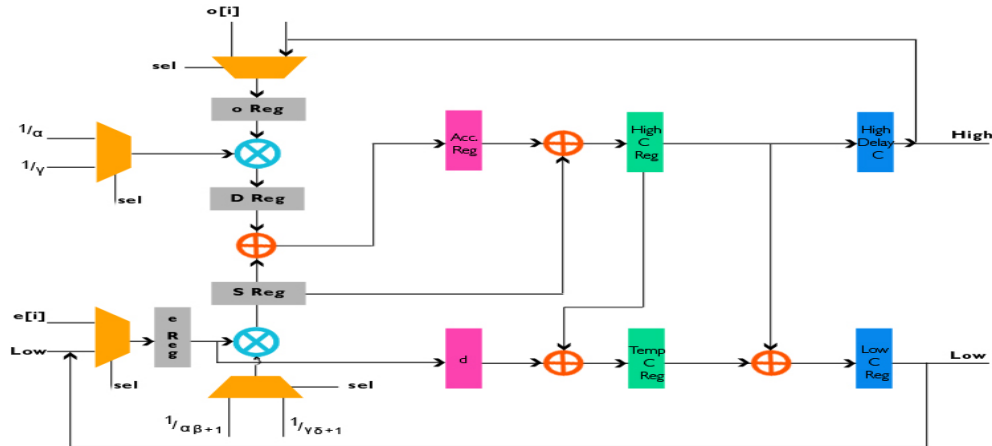


Fig c Block diagram of Column Processing Element

The coefficients and inputs are selected using multiplexer by a select signal sel. The odd and even signals are stored in O reg and E reg respectively. To these value earlier calculated coefficients are multiplied and stored to D reg and S reg. after this 3 levels of pipelining is done. After computation we obtain high pass and low pass components of column processing element. Its denoted as high C and Temp C and during

then fed to column processor first and after first lifting stage it is passed to transposing buffer and this performs shifting of image with requirement as of row processing element. After processing of 1st stag in row processor its fed to coloumn processor again to perform 2nd lifting stage followed by row filter. Here parallel processing was aimed at reducing the size of the transposing buffer helping in performing computation of 2 coefficients concurrently along row direction.

3.1 Column Processing Element

The inputs to column processing element during the 1st stage are the odd and even signals and during II stage are High and low signals. These are being selected between a multiplexer enabling hardware reduction and effective implementation of 2 D coloumn processing element. The block diagram representation of 2 D coloumn processing element is shown in fig c

final pipelined stage its given to High delay C and Low C respectively. These are fed back to the input again to perform the 2nd lifting stage. Only low pass components are compressed to obtain a new image.

3.2 Row Processing Element

The row processing element also has same architecture similar as of coloumn processing

element. It also process odd and even signals simultaneously and provide output. Thus we obtain 2 output at a time giving parallel processing.

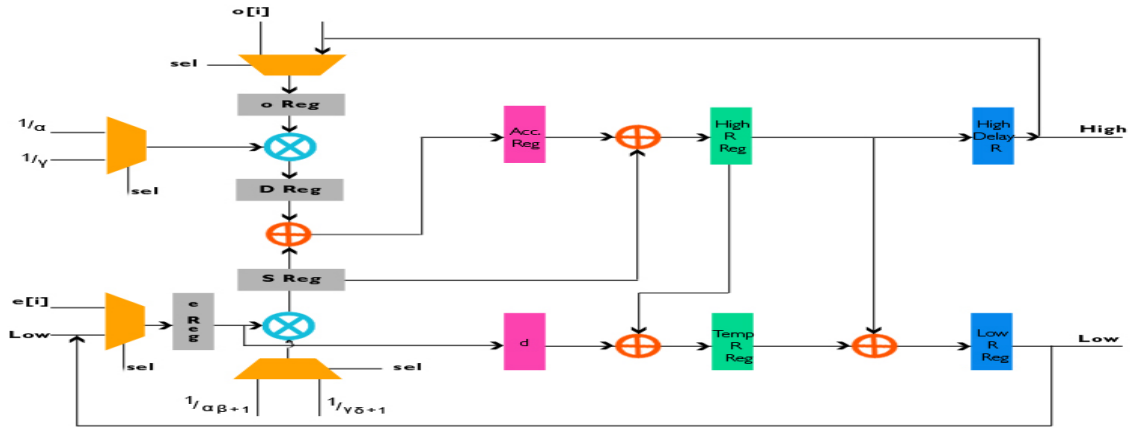


Fig- d Block dig of Row processing Element

3.3 Transposing Buffer

Since parallel processing is done, the inputs have to switch between the modes needed as per requirement of processing elements. Thus in the proposed architecture, it uses a 2*2 transposing buffer which prevents the changing the size of it as per frame size as per earlier design.[8]

4.PERFORMANCE ANALYSIS

Table 1 and Table 2 shows the comparison of proposed DWT of 9/7 filter with various architectures based on the resource utilization. The proposed architecture has a throughput of 2 input and 2 output with a critical path of T_m ie one multiplier, with reduction of resources cause of hardware minimization of 2nd lifting stage.

Table 1: 1D DWT core of 9/7 filter

Architecture	Multiplier	Adder	Register	Critical path	Throughput rate per cycle
Cell Based [10]	12	16	56	T_m	2 input 2 output
Parallel Scanning [8]	10	16	44	T_m	2 input 2 output
Modified lifting [11]	6	8	42	T_m	1 input 1 output
Proposed	6	8	22	T_m	2input 2 output

Table 2: 2D DWT core of 9/7 filter

Architecture	Multiplier	Adder	Register	Critical path	Throughput rate per cycle
Shi [9]	2	4	10	$T_m + T_a$	2 input 2 output
Cell Based [10]	4	8	28	T_m	2 input 2 output
Parallel Scanning [8]	4	8	22	T_m	2 input 2 output
Modified lifting [11]	2	4	20	T	1 input 1 output
Proposed	2	4	20	T_m	2 input 2 output

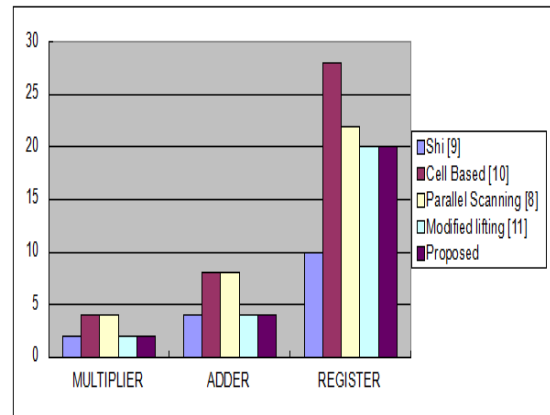
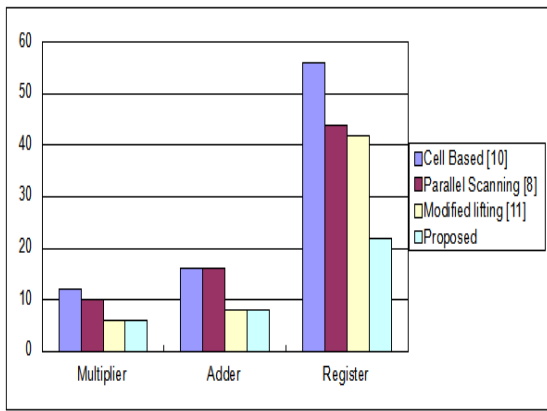


Figure e: Comparison of various architectures in 1D DWT Core with respect to adders, multipliers and registers

Figure f: Comparison of various architectures in 2D DWT Core with respect to adders, multipliers and registers

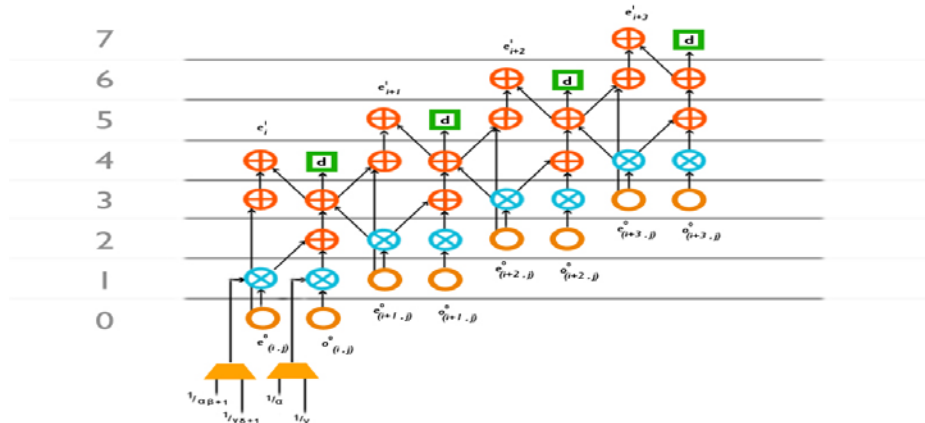


Fig - g DFG of Processing Element

5. CONCLUSION

In this paper we have developed and memory efficient and lesser hardware 2 D DWT with a critical path of T_m . The core of DWT has 2D row and column processing elements and with 2*2 transposing buffer and temporal memory RAM. The transposing buffer size was reduced by using the parallel scanning method. The architecture was implemented using 45 nm Technology and coded using Verilog HDL and simulated using Cadence NC launch and synthesized in RTL compiler.

REFERENCES

- [1] K. K. Parhi and T. Nishitani, "VLSI architecture for discrete wavelet transforms," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 1, no. 2, pp. 191–202, Jun. 1993.
- [2] P. Wu and L. Chen, "An efficient architecture for two-dimensional discrete wavelet transform," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 4, pp. 536–545, Apr. 2001.
- [3] W. Sweldens, "The new philosophy in biorthogonal wavelet constructions," in *Proc. SPIE.*, 1995, vol. 2569, pp. 68–79.
- [4] I. Daubechies and W. Sweldens, "Factoring wavelet transform into lifting steps," *J. Fourier Anal. Appl.*, vol. 4, no. 3, pp. 245–267, Mar. 1998.
- [5] G. Shi, W. Liu, and L. Zhang, "An efficient folded architecture for liftingbased discrete wavelet transform," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 56, no. 4, pp. 290–294, Apr. 2009.
- [6] 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.
- [7] Y. K. Lai, L. F. Chen, and Y. C. Shih, "A high-performance and memory-efficient VLSI architecture with parallel scanning method for 2-D lifting-based discrete wavelet transform," *IEEE Trans. Consum. Electron.*, vol. 55, no. 2, pp. 400–407, May 2009.
- [8] Yeong-Kang Lai, Member, IEEE, Lien-Fei Chen, Student Member, IEEE, and Yui-Chih Shih, "A High-Performance and Memory-Efficient VLSI Architecture with Parallel Scanning Method for 2-D Lifting-Based Discrete Wavelet Transform" *IEEE Transactions on Consumer Electronics*, Vol. 55, No. 2, MAY 2009.
- [9] G. Shi, W. Liu, and L. Zhang, "An efficient folded architecture for lifting based discrete wavelet transform," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 56, no. 4, pp. 290–294, Apr. 2009.
- [10] Y.-H. Seo, and D.-W. Kim, "VLSI architecture of line-based lifting wavelet transform for motion JPEG2000," *IEEE J. Solid-State Circuits*, vol. 42, no. 2, pp. 431–440, Feb. 2007.
- [11] B.-F. Wu and C.-F. Lin, "A high-performance and memory-efficient pipeline architecture for the 53 and 97 discrete wavelet transform of JPEG2000 codec" *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 12, Dec. 2005.