



AN EFFICIENT LIFTING SCHEME ARCHITECTURE FOR 2D DISCRETE WAVELET TRANSFORM

V. Vaishnavi¹ and M. Thamarai²

¹Department of Electronics and Communication, Asian College of Engineering and Technology, Coimbatore, India

²Department of Electronics and Communication, Karpagam College of Engineering, Coimbatore, India

ABSTRACT

A high-speed and reduced-area lifting architecture for 2D Discrete Wavelet Transform computation and the 2-D DWT Image Decomposition is proposed in this work. Lift scheme is one of the wavelet computation techniques. Prior DWT architectures are mostly constructed on the basic lifting scheme or the flipping structure. In order to attain a critical path with only one multiplier, at least four pipelining stages are mandatory for one lifting step, or a large temporal buffer is required. In this work, modifications are made in the lifting scheme as the Radix-8 booth multiplier is used and the intermediate values are recombined and stored to reduce the number of pipelining stages and the registers. The two-input/two-output parallel scanning architecture is adopted in the design. The detailed analysis is performed to compare the proposed architecture with the modified architecture in terms of hardware complexity computation time and Power consumption. In the proposed architecture, the number of LUTs reduced to 50%, power consumption is reduced to 89mw, and computation time delay is reduced to 36.6% when compared to the conventional Lifting Scheme.

Keywords: discrete wavelet transforms (DWT), flipping structure, lifting scheme, pipeline, VLSI architecture.

1. INTRODUCTION

The Discrete Wavelet Transform (DWT) has become a very versatile signal processing tool over the last decade. It has been effectively used in signal and image processing applications. The advantage of DWT over other traditional transformations is that it performs multiresolution analysis of signals with localization both in time and frequency. The DWT is being increasingly used for image compression today since it supports features like progressive image transmission, image manipulation, region of interest coding, etc. The coding efficiency and the quality of image restoration with the DWT are higher than those with the traditional discrete cosine transform. Furthermore, it is easy to attain a high compression ratio. So the DWT is widely used in signal processing and image compression, such as MPEG-4, JPEG 2000, and so on [1], [2]. Traditional DWT architectures [3], [4] are based on convolutions. Then, the second-generation DWTs, are based on lifting algorithms are proposed [5], [6]. Compared with convolution-based, lifting-based architectures require lesser computation complexity and also require less memory. Directly mapping these algorithms to hardware [7] leads to relatively long data path and low efficiency.

Several different architectures based on the lifting scheme have been proposed. An efficient folded architecture (EFA) with low hardware complexity is discussed by G. Shi; W. Liu *et al* [8]. However, computation time of EFA is quite long. A pipelined architecture is discussed by B. F. Wu and C. F. Lin [9], to reduce the critical path to one multiplier and limit the size of the temporal buffer to $4N$, high processing speed cannot be achieved because it has one input and one output. The parallel 2-D DWT is discussed by Y. K. Lai, L. F. Chen, and Y. C. Shih [10], the design is a pipelined two-input/two output architecture, and a 2×2 transposing module with four registers, the critical path delay is one T_m . But it needs eight pipelining stages to complete the 1-

D DWT and it requires 22 registers for computation. The flipping structure is discussed by C.-T. Huang, P.-C. Tseng and L.-G. Chen [11]. But, the flipping structure has a large temporal buffer, and lead to longer critical path delay due to fewer pipelining stages, various efficient lifting architectures are discussed in [12], [13], [14] and [15]. High speed VLSI implementation of 2D DWT is discussed in [16]. Different pipelined architectures are discussed in [17], [18] and [19]. An efficient multiplier less design is discussed in [20] and Lifting structure with Booth multiplier is discussed in [21].

Further optimization on the lifting scheme is proposed to overcome drawbacks in former works and reduce sizes of the logic units and the memory without loss of the throughput. The number of pipelining stages and registers is reduced, by recombining the intermediate values of the row and column transforms and keeping the critical path delay as T_m . In addition, a novel architecture is established to implement the 2-D DWT based on the above modified scheme. To reduce the size of the transposing buffer the parallel scanning method is employed. As a result, the design achieves higher efficiency.

2. PROPOSED ALGORITHM

The existing architectures for implementing the DWT are mainly classified into two categories: convolution based and lifting based approach. The lifting-based architectures have advantages over the convolution-based in computational complexity and memory requirement. The lifting scheme was first proposed by Daubechies and Sweldens in 1996 [5], [6]. It illustrates that every finite-impulse response wavelet or filter bank can be factored into a cascade of lifting steps. The polyphase matrices for the wavelet filters can be decomposed into a sequence of alternating upper and lower triangular matrices multiplied by a diagonal



normalization matrix. The entire lifting scheme of the 9/7 filter has two lifting steps and one scaling step.

To optimize the critical path of the lifting-based hardware implementation, by changing the coefficients in lifting formulas [11] a modified algorithm is employed, and is described as follows:

The flipping based lifting scheme implementation is described as follows.

$$\frac{1}{\alpha} y(2n+1) = \frac{1}{\alpha} x(2n+1) + x(2n) + x(2n+1) \quad (1)$$

$$\frac{1}{\beta} y(2n) = \frac{1}{\beta} x(2n) + y(2n-1) + y(2n+1) \quad (2)$$

$$\frac{1}{\gamma} H(2n+1) = \frac{1}{\gamma} y(2n+1) + y(2n) + y(2n+2) \quad (3)$$

$$\frac{1}{\delta} L(2n) = \frac{1}{\delta} y(2n) + H(2n-1) + H(2n+1) \quad (4)$$

Here, the lifting coefficients α , β , γ , and δ , and Scaling constant K are $\alpha \approx -(3/2)$, $\beta \approx -16$, $\gamma \approx (5/4)$ and $\delta \approx (32/15)$, respectively. With the flipping structure, we can achieve one multiplier delay by pipelining. But it needs the temporal buffer with the size of $11N$ to cache the intermediate data.

Substituting equation (1) into (2) with the associative law and reordering the expression

$$\begin{aligned} \frac{1}{\alpha\beta} y(2n) &= \frac{1}{\alpha\beta} x(2n) + \frac{1}{\alpha} y(2n-1) + \frac{1}{\alpha} y(2n+1) \\ &= [(\frac{1}{\alpha\beta} + 1) x(2n) + \frac{1}{\alpha} x(2n-1) + x(2n-2)] + [\frac{1}{\alpha} x(2n+1) + x(2n) + x(2n-2)] \end{aligned} \quad (5)$$

Four intermediate variables, $Dk 1(n)$, $Dk 2(n)$, $Dk 3(n)$, and $Dk 4(n)$, are defined, where k stands for different values in the row and column transforms. k represents the number of rows in progress in the row transform, whereas it represents the number of scans in the column transform. One parallel scan takes two adjacent rows for computation in column transform. Hence,

$$Dk 1(n) = \frac{1}{\alpha} x(2n+1) + x(2n) \quad (6)$$

$$Dk 2(n) = (\frac{1}{\alpha\beta} + 1) x(2n) + \frac{1}{\alpha} x(2n-1) + x(2n-2) \quad (7)$$

$$Dk 3(n) = \frac{1}{\gamma} y(2n+1) + y(2n) \quad (8)$$

$$Dk 4(n) = (\frac{1}{\delta\gamma} + 1) y(2n) + \frac{1}{\gamma} y(2n-1) + y(2n-2) \quad (9)$$

By rearranging the equation from (1) to (4)

$$\frac{1}{\alpha} y(2n+1) = Dk 1(n) + x(2n+2) \quad (10)$$

$$\frac{1}{\alpha\beta} y(2n) = Dk 2(n) + Dk 1(n) + x(2n+2) \quad (11)$$

$$\frac{1}{\gamma} H(2n+1) = Dk 3(n) + y(2n+2) \quad (12)$$

$$\frac{1}{\delta\gamma} L(2n) = Dk 4(n) + Dk 3(n) + y(2n+2) \quad (13)$$

Compared with the lifting based flipping scheme, the modified algorithm suggests a data combination with different coefficients in even data and simplifies the computation process and reduces the number of registers.

The predictor is combined with the updater in this proposed algorithm. In the two-input/ two-output architecture the high-pass and low-pass signals are calculated in parallel. The high-pass signal and the low-pass signal can be obtained from the third pipelining stage. Hence, the pipelining stages of the 1-D processing element (PE) are narrow to three, and the number of registers is further reduced. The Computation of the Lifting based DWT consists of Adders, Multipliers and D- flipflops. In the proposed work, the Radix-8 booth Multiplier is used instead of Wallace tree Multiplier which is used in the existing works.

3. 2D-DWT OVERALL ARCHITECTURE

Based on the proposed algorithm, first the binary pixel values of the input image is given as an input to pre-processing module of Column filter. After that, column transform is performed. The output data of the column filter are sent into the transposing buffer. The data transposition is operated to meet the order of the data flow required by the row filter. Next, the row filter begins to read the data from the transposing buffer for the row transform. At last, the scaling module is used to finish the scaling computation shown in Figure-1.

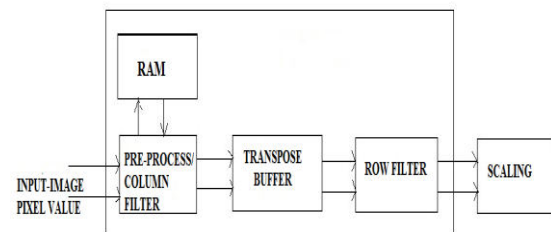


Figure-1. Overall 2D DWT architecture.

A) Pre-processing of raw image / column filter

For the proposed algorithm, the 8×8 sub block of binary input image pixel values are considered as the data for input to the proposed architecture. The Lifting transform is performed for 8-bit value of each 8 input data. The parallel scanning is performed, as shown in Fig. 2, the data of each even row and odd row of the columns are alternately read. The column filter, which is designed for 64 input datas performs the column transform.



Figure-2. Parallel scanning of input data.

B) Transpose buffer

In order to reduce the size of the transposing buffer between the column and row filters and to improve the processing speed, the two-input/two-output architecture is adopted. As shown in Figure-3 the output



data of the column filter are sent into the transposing buffer. The data transposition is performed using three registers and two multiplexers and thus output data is made to meet the order of the data flow required by the row filter.

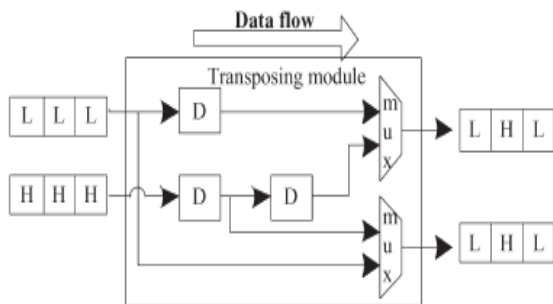


Figure-3. Architecture of transposing module and the order of input and output.

C) Row filter

The output of the transpose buffer is given as the input to the Row filter. The row filter is designed for 64 input and output data's and performs row transform. The row transform is performed by taking Lifting transform 8 times.

D) Scaling

Finally the Scaling computations are performed. Here two constants are used for scaling k_1 and k_2 respectively. The constant $k_1 = 1/2$ for even columns and $k_2 = 3/2$ for odd columns.

4. RESULTS AND DISCUSSION

The Original image of 'Lena' with 512 x 512 size is taken as input image is shown below in Figure-4. The input image is resized to 256 x 256 size and its pixel values are shown in Figure-5.

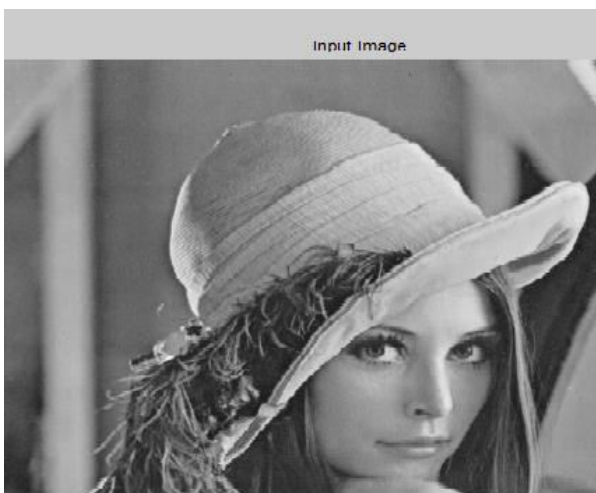


Figure-4. Input image.

1	163	162	160	162	164	160	159	158
2	162	162	160	162	164	160	159	158
3	163	160	160	160	162	159	157	156
4	160	158	159	157	160	160	154	153
5	155	157	157	157	159	157	157	155
6	156	158	156	153	159	158	156	157
7	157	157	156	156	158	157	155	159
8	158	157	157	157	156	157	156	158

Figure-5. Resized image pixel value.

The Verilog coding for the Lifting transform is performed with 8 bit value. The decimal values obtained from the input image is taken separately as 8 x 8 sub blocks and it is converted to its corresponding binary values. These binary values are then provided as the input to perform the Lifting transform.

Lifting Transform is performed by column filtering, transposing row filtering and scaling operations in the architecture. The 2-D DWT Architecture is designed for 64 input and output data, the binary pixel values of the resized input data are given as the input to the 2-D DWT Architecture.

The parallel scanning is adopted to read the data of each even row and odd row of the columns alternately. The column filter which is designed for 64 input data can process the column transform. Following that, the transposing and row transforms are performed.

Finally the Scaling computations are performed. Here two constants are used for scaling k_1 and k_2 respectively. Thus the Simulated output Result of the 2-D DWT Architecture for 9/7 filter is shown in Figure-6.

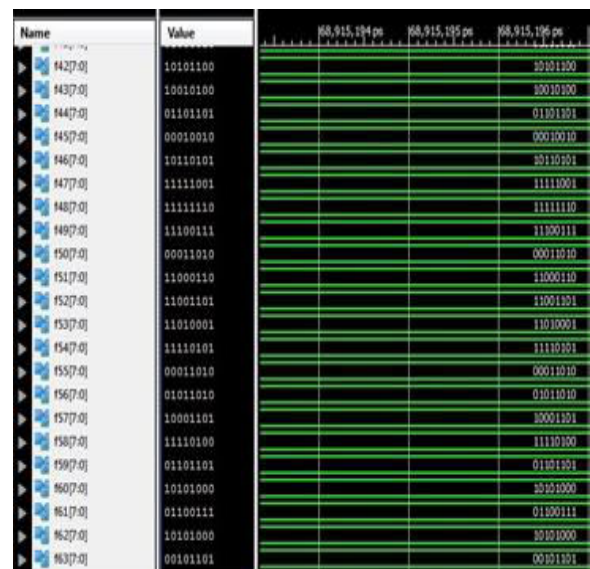


Figure-6. Simulated output result of the 2-D DWT architecture for 9/7 filter.



5. PERFORMANCE COMPARISON AND ANALYSIS

Table-1 shows the synthesis result of Comparison of Area between Flipping Structure Algorithm [11] and Proposed Algorithm. Compared with the flipping structure the modified algorithm uses less number of flip flops and LUTs.

Table-1. Performance comparison of flipping structure algorithm and proposed algorithm.

Logic utilizations	Flipping structure algorithm	Proposed algorithm
No. of slices	271	198
No. of 4 input LUTs	490	360
No. of bonded IOBs	130	130
No. of slices MULT 18x18s	4	2
No. of GCLKS	1	1

Table-2. Comparison of power dissipation between flipping structure algorithm and proposed algorithm.

power dissipation	Flipping structure algorithm	Proposed algorithm
Total thermal power dissipation	84.16mw	80.03mw
Core dynamic thermal power dissipation	3.63mw	1.67mw
Static thermal power dissipation	51.78mw	51.77mw
I/O Thermal power dissipation	28.76mw	26.58mw

The Power Dissipation is less in the Proposed algorithm compared with the flipping structure algorithm as shown in Table-2.

Table-3. Comparison of timing constraints between flipping structure algorithm and modified lifting algorithm.

Timing constraints	Flipping structure algorithm	Proposed algorithm
Tsu	11.942ns	11.327ns
Tco	8.465ns	7.92ns
Th	-0.556ns	-0.149ns

The Timing constraints set up time (Tsu), clock to out time (Tco), Hold time (Th) is better compared to the flipping structure algorithm as shown in Table 3.

The Table-4 shows the Area of the Wei Zhang Architecture [22] is compared with the proposed Architecture in which the Radix-8 Booth Multiplier is used so that the Area is efficiently reduced in the proposed Architecture.

Table-4. Comparison of area between the Wei Zhang architecture [22] and proposed architecture.

Logic utilization	Wei Zhang architecture[22]	Proposed architecture
Number of slice flip flops	904	825
No. of 4 input LUTs	2,995	1,870
No. of occupied slices	1,853	1,213
No. of slices containing only related logic	1,853	1,213
No. of slices containing only unrelated logic	0	0
Total no. of 4 input LUTs	3,059	1,934
Number used as logic	2,995	1,870
Number used as 16x1 RAMs	64	64
No. of bonded IOBs	33	33
No. of GCLKs	1	1
No. of GCLK IOBs	1	1
Total equivalent gate count for design	33,613	26,097
Additional JTAG gate count for IOBs	1,632	1,632

The Timing delay of the Wei Zhang Architecture [22] and Proposed Architecture with its Gate delay and the net delay is determined in the Synthesis Report. The total timing delay obtained for the Proposed Architecture is 16.102ns which is lesser than the Wei Zhang Architecture [22] in which the total timing delay is 36.980ns.



Table-5. Comparison of power consumption between the Wei Zhang architecture [22] and proposed architecture.

Power summary	Wei Zhang architecture[22]	Proposed architecture
Total estimated power consumption	92mw	89mw
Vccint 1.80V:	85mw	82mw
Vcco33 3.30V:	7mw	7mw
Clocks:	58mw	55mw
Inputs:	1mw	1mw
QuiescentVccint 1.80V:	27mw	27mw
QuiescentVcco33 3.30V:	7mw	7mw

The proposed architecture consumes less power compared to the Wei Zhang architecture[22]. The clock power consumption is 55mw which is lesser than the power consumption of the Wei Zhang. Thus the power consumption is efficient in the proposed architecture.

CONCLUSIONS

Thus a novel architecture for the 1D and 2D DWTs is proposed. The modified one lifting step circuit can work within three pipelining stages with fewer registers, and the critical path delay is T_m . The detailed analysis is performed to compare the Wei Zhang architecture [22] with the proposed architecture in which Radix-8 Booth Multiplier is used. In terms of hardware complexity the number of LUTs used is reduced to 50%, Power consumption is reduced to 89mw, and computation time delay is reduced to 36.6%. Hence the proposed architecture achieves high speed with lower hardware complexity and smaller storage size. In future, the size of the transpose buffer can be reduced and the performance of the work can also be compared with the other architectures.

REFERENCES

- [1] Xing G, Li J and Zhang Y Q, "Arbitrarily shaped video-object coding by wavelet," IEEE Trans. Circuits Syst. Video Technol., vol. 11, no. 10, (2001)1135–1139.
- [2] Lo S C B, Li, H & Freedman MT, "Optimization of wavelet decomposition for image compression and feature preservation," IEEE Trans. Med. Imag., vol. 22, no. 9 (2003)1141–1151.
- [3] Parhi K K and Nishitani T, "VLSI architecture for discrete wavelet transforms," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 1, no. 2, (1993). 191-202.
- [4] Wu P and Chen L, "An efficient architecture for two-dimensional discrete wavelet transform," IEEE Trans. Circuits Syst. Video Technol., vol. 11, no. 4, (2001) 536-545.
- [5] Sweldens W, "The new philosophy in biorthogonal wavelet constructions," in Proc. SPIE., 1995, vol. 2569, 68-79.
- [6] Daubechies I and Sweldens W, "Factoring wavelet transform into lifting steps," J. Fourier Anal. Appl., vol. 4, no. 3, (1998) pp. 245-267.
- [7] Jou J M, Shiao Y H., and Liu CC, "Efficient VLSI architectures for the biorthogonal wavelet transform by filter bank and lifting scheme," in Proc. IEEE ISCAS, vol. 2, (2001) 529–532.
- [8] Shi G, Liu W and Zhang L, "An efficient folded architecture for lifting-based discrete wavelet transform," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 56, no. 4, (2009) 290–294.
- [9] Wu B F and Lin C F, "A high-performance and memory-efficient pipeline architecture for the 5/3 and 9/7 discrete wavelet transform of JPEG2000 codec," IEEE Trans. Circuits Syst. Video Technol., vol. 15, no. 12, (2005)1615–1628.
- [10]Lai YK. Chen LF, and. Shih YC, "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, (2009). 400-407.
- [11]Huang CT., Tseng, P-C and Chen LG., "Flipping structure: An efficient VLSI architecture for lifting-based discrete wavelet transform," IEEE Trans. Signal Process. vol. 52, no. 4, (2004) 1080–1089.
- [12]Tseng PC., Huang CT, and Chen L G., "Generic RAM-based architecture for two dimensional discrete wavelet transform with line-based method," in Proc. Asia-Pacific Conf. Circuits Syst., vol. 2, (2002) 363–366.
- [13]Xiong C., Tian J, and Liu J, "Efficient architectures for two-dimensional discrete wavelet transform using lifting scheme," IEEE Trans. Image Process., vol. 16, no. 3, (2007) 607–614.
- [14]Liao H., Mandal M. K., and Cockburn B F, "Efficient architectures for 1-D and 2-D lifting-based wavelet



transforms,” IEEE Trans. Signal Process., vol. 52, no. 5, (2004) 1315-1326.

- [15] Xiong, C Y, Tian J.W. and Liu J, “A note on ‘flipping structure: An efficient VLSI architecture for lifting-based discrete wavelet transform’,” IEEE Trans. Signal Process. vol. 54, no. 5(2006) 1910-1916.
- [16] Cheng C and Parhi K K., “High-speed VLSI implement of 2-D discrete wavelet transform,” IEEE Trans. Signal Process. vol. 56, no. 1, (2008) 393- 403.
- [17] Mohanty B K. and Meher P K., “Throughput-scalable hybrid-pipeline architecture for multilevel lifting 2-D DWT of JPEG 2000 coder,” in Proc. IEEE Int. Conf. Appl.-Specific Syst., Archit. Processors, (2008) 305-309.
- [18] Song J and Park IC., “Novel pipelined DWT architecture for dual-line scan,” in Proc. IEEE Int. Symp. Circuits Syst., (2009). 373-376.
- [19] Wu Z G and Wang W, “Pipelined architecture for FPGA implementation of lifting-based DWT,” in Proc. Int. Conf. Elect. Inform. Control Eng., (2011), 1535–1538.
- [20] Mohanty B K and Meher P K., “Efficient multiplierless designs for 1-D DWT using 9/7 filters based on distributed arithmetic,” in Proc. Int. Symp. Integr. Circuits, (2009) 364–367.
- [21] Nishat Bano (2012) “VLSI Design of Low Power Booth Multiplier”. International journal of Scientific Engineering Research, Vol.3, No. 2(2012).
- [22] Wei Zhang, Zhe Jiang, Zhiyu Gao, and Yanyan and Liu “An Efficient VLSI Architecture for Lifting-Based Discrete Wavelet Transform”, IEEE Transactions on Circuits, vol. 59(2012)158-162.