Research Inventy: International Journal of Engineering And Science
Vol.4, Issue 2 (February 2014), PP -51-55
Issn (e): 2278-4721, Issn (p):2319-6483, www.researchinventy.com

Advanced Pipelined Architecture for Lifting-Based Discrete Wavelet Transform

T. Maheshwar[1], C Saritha[1], V Shiva[1]
[1,2,3] *ECE Department, Sree Dattha Institute of Engineering & Science*,

Abstract: *This paper presents a high speed 9/7 lifting 1D-DWT algorithm which is implementation on FPGA with multi-stage pipelining structure. In this brief, modifications are made to the lifting scheme, and the intermediate results are recombined and stored to reduce the number of pipelining stages. As a result, the number of registers can be reduced to 18 without extending the critical path. Compared with the architecture which without multi-stage pipeline, the proposed architecture has higher operating frequency, the design raises operating frequency around 1.5 times more fast, at the expense of about 27% more hardware area. The hardware architecture is suitable for high speed implementation*

I. Introduction

The discrete wavelet transform (DWT) is being increasingly used for image coding. It is due to the fact that DWT supports superior features like progressive image transmission by quality or by resolution. The DWT is the key component of the JPEG2000 system, [1] and it also has been adopted as the transform coder in MPEG-4 still texture coding. However, the DWT requires much more computation than the discrete cosine transform (DCT) because of filter computation. Recently, lifting scheme widely used for DWT leads a speed-up and a fewer computation compared to the classical convolution-based method. Daubechies and

Sweldens first derive the lifting-based discrete wavelet transform to reduce complex operations [2][3]. The lifting-based DWT has several. The Discrete wavelet transform (DWT) is a multiresolution analysis tool with excellent characteristics in the time and frequency domains. Through the DWT, signals can be decomposed into different subbands with both time and frequency information. The coding efficiency and the quality of image restoration with the DWT are higher than those with the traditional discrete cosine transform.

Moreover, it is easy to obtain a high compression ratio. As a result, the DWT is widely used in signal processing and image compression, such as MPEG-4, JPEG2000, and so on [1], [2].Traditional DWT architectures [3], [4] are based on convolutions. Then, the second-generation DWTs, which are based on lifting algorithms, are proposed [5], [6]. Compared with convolution-based ones, lifting-based architectures not only have lower computation complexity but also require less memory. Nevertheless, directly mapping these algorithms to hardware [7] leads to relatively long data path and low efficiency. Recently, several novel architectures based on the lifting scheme have been proposed [8]–[10]. Shi *et al.* [8] achieved an efficient folded architecture (EFA) with low hardware complexity. However, its critical path delay is$Tm + Ta$, where$Tm$and Ta are the delay of a multiplier and an adder, respectively, and the computation time of EFA is quite long. Through optimizing the lifting scheme, Wu and Lin [9] proposed a pipelined architecture to reduce the critical path to one multiplier and limit the size of the temporal buffer to $4N$, but it has one input and one output and cannot achieve high processing speed. Based on Wu and Lin's design, Lai *et al.* [10] implemented the parallel 2-D DWT. In this brief, further optimization on the lifting scheme is proposed to overcome shortages in previous works and minimize sizes of the logic units and the memory without loss of the throughput. By recombining the intermediate results of the row and column transforms, the number of pipelining stages and registers is reduced, while keeping the critical path delay as Tm. In addition, a novel architecture is developed to implement the 2-D DWT based on the above modified scheme. The parallel scanning method is employed to reduce the size of the transposing buffer. As a result, our design achieves higher efficiency. The rest of this brief is organized as follows. Section II reviews the lifting scheme and the flipping structure of the DWT, and then, we proposed our modified algorithm for DWT. Section III presents the proposed architecture for the 2-D DWT, and Section IV provides implementation results and Compare with previous architectures. Conclusion is drawn in Section V

II. Ii. Lifting-Based Discrete Wavelet Transform

A. Primitive Lifting-Based Discrete Wavelet Transform

Lifting scheme is attractive for both high throughput and low-power applications. Every finite wavelet transform can be factorized into the lifting scheme by using Euclidean algorithm. That is, the Euclidean algorithm factorizes the polyphase matrix of a DWT filter as the multiplication of alternative upper and lower triangular matrices as well as a diagonal matrix. The poly-phase matrix of wavelet transform can be represented as follows

$$P(z)=\begin{bmatrix} l_e(z) & l_o(z) \\ h_e(z) & h_o(z) \end{bmatrix} \quad (1)$$

Since *l(z)* and *h(z)* form a complementary filter pair, *P*(*z*) can be factorized into the following structure

$$P(z)=\prod_{i=1}^{m}\begin{bmatrix} 1 & 0 \\ -s_i(z^{-1}) & 1 \end{bmatrix}\begin{bmatrix} 1 & -t_i(z^{-1}) \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1/K & 0 \\ 0 & K \end{bmatrix}. \quad (2)$$

Where *K* is a constant. Therefore, the DWT is factorized into the lifting scheme. The detailed forward algorithm for 9/7 filter is described from (3) to (8). The lifting scheme for 9/7 filter consists of three steps:

We must note that the lifting scheme of the 5/3 filter can be regarded as the special case which only has a couple of the ***Predictor*** and ***Updater***. Some architectures which use the direct implementation method are proposed for 1-D lifting- based DWT of 9/7 filter [4][5]. However, the long critical path with one multiplier and two adders propagation delay is the design bottleneck for 9/7 filter with four-stage pipeline.

1) Split Step:

$$d_i^0 = x_{2i+1} \quad (3)$$
$$s_i^0 = x_{2i}$$

2) *Lifting Step*:

(the first lifting step)

$$d_i^1 = d_i^0 + \alpha \times (s_i^0 + s_{i+1}^0) \ldots (\texttt{Predictor}) \quad (4)$$
$$s_i^1 = s_i^0 + \beta \times (d_{i-1}^1 + d_i^1) \ldots (\texttt{Updater}) \quad (5)$$

(the second lifting step)

$$d_i^2 = d_i^1 + \gamma \times (s_i^1 + s_{i+1}^1) \ldots (\texttt{Predictor}) \quad (6)$$
$$s_i^2 = s_i^1 + \delta \times (d_{i-1}^2 + d_i^2) \ldots (\texttt{Updater}) \quad (7)$$

3) Scaling Step:

$$d_i = \frac{1}{K} \times d_i^2 \quad (8)$$
$$s_i = K \times s_i^2$$

III. Overall Architecture

Based on the proposed modified algorithm, a novel architecture for the 2-D DWT shown in Fig. 1 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.

B. Parallel Scan of Preprocessing and Raw Image Data

Since parallel scanning is adopted, as shown in Fig, the data of each even row and odd row of the columns are alternately read. This way, the column filter can process the column transform for the data of adjacent columns alternately. With the preprocessing module, raw image data are factorized into odd and even parts for the following filter logic unit (FLU) module.

C. One-Dimensional PE

The architecture of the proposed 1-D PE is shown in Fig. 3. This architecture can be applied in the column and row filters by selecting the RAM or Buffer properly

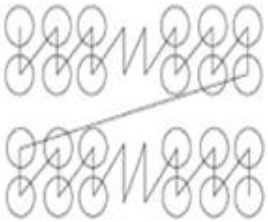

Fig. 2. Parallel scanning of the input data.

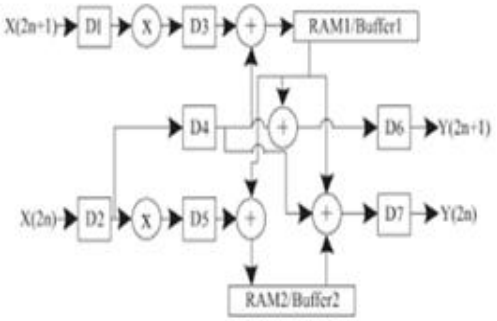

Fig. 3. Proposed 1-D PE.

TABLE I
COEFFICIENTS OF THE 9/7 FILTER FOR FORWARD AND INVERSE TRANSFORM IN BINARY FORM

Forward Transform			Inverse Transform		
Coefficients	12 bits absolute value	Value	Coefficients	12 bits absolute value	Value
β	.000011011001	-0.052978515625	K^{-1}	.110100000000	0.8125
$\beta\alpha$	.000101011000	0.083984375	γ/K	1.00010110000	1.0859375
δ/β	1000.01011111	-8.37109375	$\delta\gamma$	.011001000010	0.39111328125
$\delta\gamma$	.011001000010	0.39111328125	α/γ	1.11001011111	-1.79638671875
$1/K\delta$	10.1100011000	2.7734375	$\beta\alpha$	.000101011000	0.083984375
K	0.110100000000	0.8125	α^{-1}	.101000010110	-0.63037109375

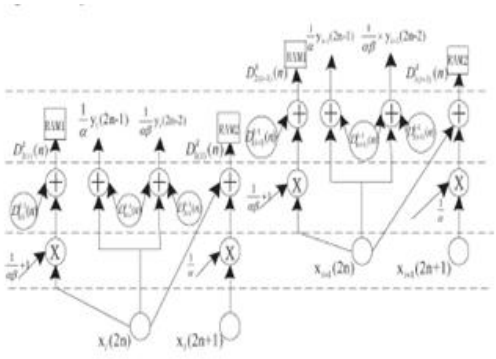

Fig. 4. Data flow of the first lifting step in the column filter.

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. When the column filter begins to work, the input data from the preprocessing module, the odd term $xi(2n + 1)$ and the even term $xi(2n)$, are sent into the column filter in each clock cycle simultaneously. In this case, i mean the column index, and n has the same meaning as k, which represents the number of scans in the column. Then, each input is multiplied by the corresponding coefficient. The result of multiplication is used in the computation of the intermediate variables $Dk\ 1(i)(n)$ and $Dk\ \ 2(i)(n)$,which will be stored in the temporal buffers RAM1 and RAM2, respectively. At the same time, if$k > 1$, $Dk-11(i)\ (n)$ and $Dk-1\ 2(i)\ (n)$ will be read for the computation of $yi(2n - 1)$ and $yi(2n - 2)$. Otherwise, the boundary extension will be processed. Because of parallel scanning, $Dk-1\ \ 1(i)\ (n)$ and $Dk-1$

2(*i*) (*n*) read from RAM1 and RAM2 in current clock cycle are calculated and saved at the corresponding places in the previous two rows. Then, they are used by the new data to carry out the current column transform
RAM1 and RAM2 are both dual-port RAM devices with the depth of 2*N*. Fig. 4 shows the data flow of the first lifting step in the column filter. Such architecture not only has low hardware complexity but also reduces the critical path between every two registers to one multiplier.

IV. Results And Discussion

The input is 16 bits each input bit width is 20 bit width. The DWT consists of registers and adders. Whenever the input is send, the data divided into even data and odd data. The even data and odd data is stored in the temporary registers. When the reset is high the temporary register value consists of zero whenever the reset is low the input data split into the even data and odd data. The input data read up to sixteen clock cycles after that the data read according to the lifting scheme. The output data consists of low pass and high pass elements. This is the 1-D discrete wavelet transform. The 2-D discrete wavelet transform is that the low pass and the high pass again divided into LL, LH and HH, HL. The output is verified in the Modelsim.

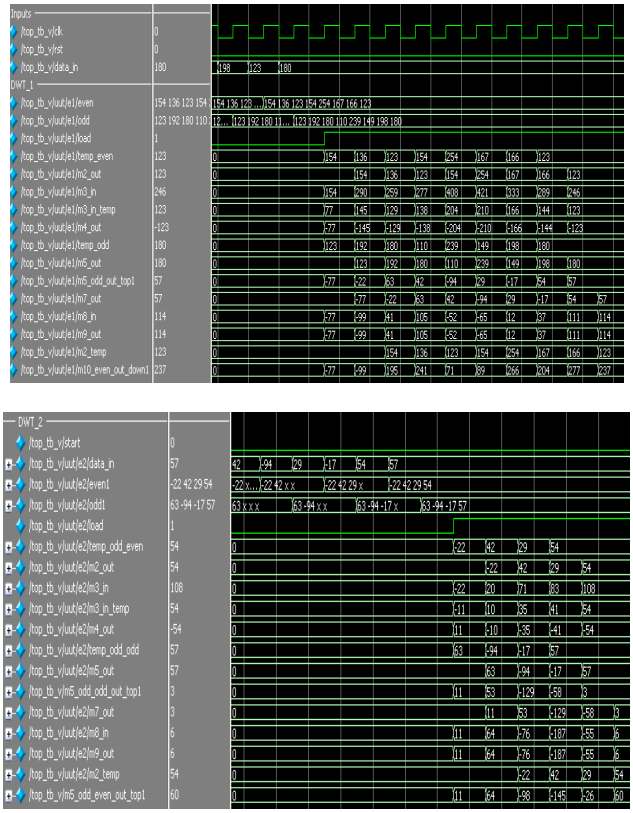

Fig 5 Simulation Result of DWT-2 Block with Both High and Low Pass Coefficients

tst1 Project Status (04/21/2010 - 17:41:36)			
Project File:	tst1.ise	**Current State:**	Placed and Routed
Module Name:	top_dwt	**• Errors:**	No Errors
Target Device:	xc3s500e-4fg320	**• Warnings:**	5 Warnings
Product Version:	ISE 10.1 - Foundation Simulator	**• Routing Results:**	All Signals Completely Routed
Design Goal:	Balanced	**• Timing Constraints:**	All Constraints Met
Design Strategy:	Xilinx Default (unlocked)	**• Final Timing Score:**	0 (Timing Report)

tst1 Partition Summary
No partition information was found.

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	386	9,312	4%	
Number of 4 input LUTs	532	9,312	5%	
Logic Distribution				
Number of occupied Slices	446	4,656	9%	
Number of Slices containing only related logic	446	446	100%	
Number of Slices containing unrelated logic	0	446	0%	
Total Number of 4 input LUTs	532	9,312	5%	
Number used as logic	292			
Number used for Dual Port RAMs	240			
Number of bonded IOBs	104	232	44%	
Number of BUFGMUXs	1	24	4%	

Fig 6: Design utilization summary

Now these low pass coefficients and the high pass coefficients were taken as the input for the further process. Hence for the DWT-2 process, low pass coefficients will be taken as the inputs and will do the process in order to calculate the low pass and high pass coefficients from the transformed coefficients of DWT-1. In DWT-2, the same process as in DWT-1 will be carried out. Hence the simulated waveform is shown in the figure 4.2.

V. Conclusion& Future Scope

Basically the medical images need more accuracy without loosing of information. The Discrete Wavelet Transform (DWT) was based on time-scale representation, which provides efficient multi-resolution. The lifting based scheme (9, 7) (The high pass filter has five taps and the low pass filter has three taps) filter give lossless mode of information. A more efficient approach to lossless whose coefficients are exactly represented by finite precision numbers allows for truly lossless encoding.

As future work, this work can be extended in order to increase the accuracy by increasing the level of transformations. This can be used as a part of the block in the full fledged application, i.e., by using these DWT, the applications can be developed such as compression, watermarking, etc.

References

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

[2]. 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, Sep. 2003.

[3]. 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.

[4]. 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.

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

[6]. I. Daubechies and W. Sweldens, "Factoring wavelet transform into lifting steps," *J. Fourier Anal. Appl.*, vol. 4, no. 3, pp. 245 267, Mar. 1998.

[7]. J. M. Jou, Y. H. Shiau, and C. C. Liu, "Efficient VLSI architectures for the biorthogonal wavelet transform by filter bank and lifting scheme," in *Proc. IEEE ISCAS*, May 2001, vol. 2, pp. 529–532.

[8]. 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.

[9]. B. F. Wu and C. F. Lin, "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, pp. 1615–1628, Dec. 2005.

[10]. 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.

[11]. 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.

[12]. P.-C. Tseng, C.-T. Huang, and L.-G. Chen, "Generic RAM-based architecture for two dimensional discrete wavelet transform with line based method," in *Proc. Asia-Pacific Conf. Circuits Syst.*, 2002, vol. 2, pp. 363–366.