

# A Novel, Efficient Architecture for the 1D, Lifting-Based DWT with Folded and Pipelined Schemes

Chin-Fa Hsieh<sup>1</sup> Tsung-Han Tsai<sup>2</sup> Neng-Jye Hsu<sup>1</sup> Chih-Hung Lai<sup>1</sup>

{c0935, njhsu, s9143078}@cc.chit.edu.tw<sup>1</sup>, han@ee.ncu.edu.tw<sup>2</sup>

<sup>1</sup>Department. of Electronics Engineering, China Institute of Technology, Taipei, Taiwan

<sup>2</sup>Department. of Electrical Engineering, National Central University, Chung-Li, Taiwan

## Abstract

In this paper, we propose a novel, efficient VLSI architecture for the implementation of one-dimension, lifting-based discrete wavelet transform (DWT). Both of the folded and the pipelined schemes are applied in the proposed architecture; the former scheme supports higher hardware utilization and the latter scheme speed up the clock rate of the DWT. The architecture has been coded in Verilog HDL, then verified successfully by the platform of Quartus-II of version 5.0. Finally, it was realized with the FPGA device of Cyclone family from Altera Corp.

**Keywords:** lifting, discrete wavelet transform

## 1. Introduction

Discrete wavelet transform (DWT) has been widely used in many different fields of audio and video signal processing. Recently, DWT is being increasingly used as effective solutions to the problem of image compression. One well-known example is that DWT has been adopted by the JPEG2000[6], one of the several popular image compression standards defined by the Joint Picture Expert Group (JPEG), due to the efficient decomposition of a signal into several components (sub-bands) with DWT. In general, DWT can be implemented by direct convolution and several DWT architectures implemented by filter convolution have been proposed. However, such an implementation suffers the need of a large number of computations and a large storage resource[3].

A new scheme, termed the lifting scheme,[2] which reduces the number of computation has been proposed for the DWT. Lifting scheme has several advantages, such as in-place computation of the wavelet coefficients, integer-to-integer wavelet transform etc. Up to now, several VLSI implementations have been proposed based on the lifting scheme.[3][4][5] Among them, Chen[3] proposed a multilevel lifting-based wavelet transform

architecture. Chen's architecture takes only half the time for the computation of DWT when compared with other designs, but it still works under a large clock period.

In this article, we propose an architecture using folded and pipelined schemes to implement a 3-level, 1D lifting-based DWT. The proposed architecture effectively shortens the critical path and therefore enhances the clock period, without adding the number of adders and multipliers/shifters when compared with the architecture in [3].

This article is organized as follows. In Section 2, we review the background of the lifting scheme briefly. The proposed architecture for the 3-level, 1D lifting-based DWT is presented in Section 3. The simulation results of the proposed architecture and its FPGA implementation is described in Section 4. Finally in Section 5, we give a conclusion of this article.

## 2. Background

### 2.1. Discrete Wavelet Transform

DWT can decompose the input samples in multi-resolution.[1] As shown in Figure 1, the implementation of the discrete wavelet transform is based on the filter banks, where G and H denote a high-pass filter and a low-pass-filter, respectively. After each filtering, the number of the output samples is decimated by a factor of 2. The samples generated by the high pass filters are completely decomposed; meanwhile, the other samples generated by the low pass filters are applied to the next-level computation for further decomposition.

### 2.2. Lifting Scheme Wavelet Transform

The lifting scheme is a new algorithm proposed for the implementation of the wavelet transform [2]. It can reduce the computational complexity of DWT

involved with the convolution implementation. Furthermore, the extra memory required to store the results of the convolution can also be reduced by in-place computation of the wavelet coefficient with the lifting scheme.

The lifting scheme consists of the following three steps to decompose the samples, namely, splitting, predicting, and updating. Figure 2 illustrates the three steps associated with the lifting scheme based DWT for the one-dimensional signal: (1) Split step: The input samples  $l$  are split into even samples and odd samples; (2) Predict step (P): The even samples are multiplied by the predict factor and then the results are added to the odd samples to generate the detailed coefficients; (3) Update step (U): The detailed coefficients computed by the predict step are multiplied by the update factors and then the results are added to the even samples to get the coarse coefficients.

The equations of the lifting scheme for the (5,3) discrete wavelet transform is shown as follows,

$$h_{j,i} = l_{j-1,2i+1} + \alpha l_{j-1,2i} + \alpha l_{j-1,2i+2}; \quad (1)$$

$$l_{j,i} = l_{j-1,2i-2} + \beta h_{j,i} + \beta h_{j,i-1}; \quad (2)$$

where (1)  $h$  and  $l$  are the detailed and the coarse coefficients, respectively; (2)  $\alpha$  and  $\beta$  are the predict factor and the update factor of the (5,3) filter, respectively; and (3)  $i$  and  $j$  represent the input sample index and the decomposition level, respectively.

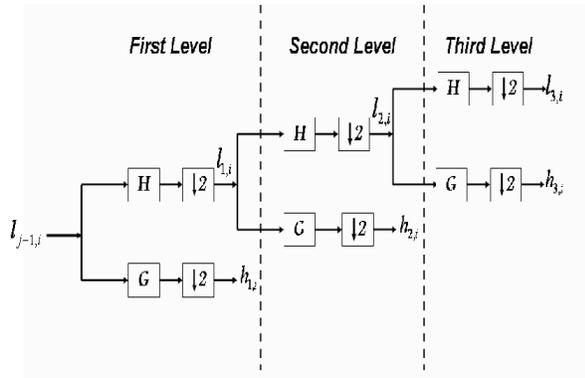


Fig. 1: Three-level analysis DWT

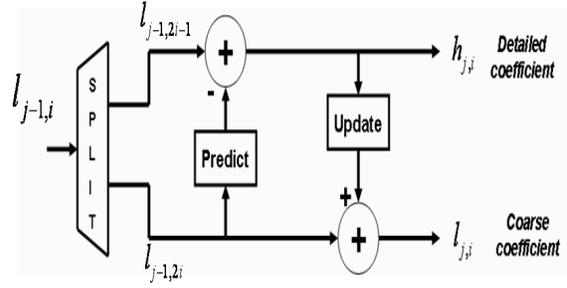


Fig. 2: Lifting DWT Module

### 3. The Proposed Architecture

The proposed folded-pipelined architecture when used for the 1D, 3-Level lifting-based (5,3) DWT is composed of the predict module, the update module, a set of registers and five multiplexing switches. The function of each part is described as follows:

#### 3.1. Predict Module

The predict module is shown in Figure 3. As mentioned in Section 2, the predict module is used to compute the detailed coefficients. Initially, the even sequence comes from MUX\_A, then it is multiplied by the predict filter coefficient  $\alpha$ , and the result is stored in Register D1. When the corresponding odd sequence comes from MUX\_B, it will be added to the data from register D1 and the result will be stored in the register D2 temporarily for two clock unit. Finally, the data stored in D2 will pass through MUX\_C and is added to the data from register D1 to generate the detailed coefficient.

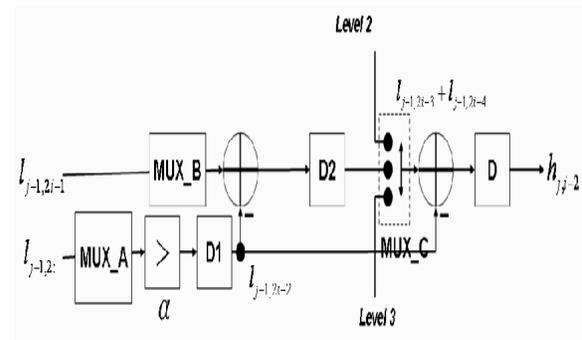


Fig. 3: Predict module of (5,3) filter

#### 3.2. Update Module

The update module is shown in Figure 4. As mentioned in Section 2, the update module is used to compute the coarse coefficients. The detailed coefficient coming from the predict module is firstly

multiplied by the update filter coefficient  $\beta$ . The result of the previous step is then added to the corresponding even sequence and the sum is temporarily stored in the register D3 for two clock unit. Then the data stored in D3 will pass through the MUX\_E and is added to the delayed detailed coefficient multiplied by  $\beta$  to get the coarse coefficient.

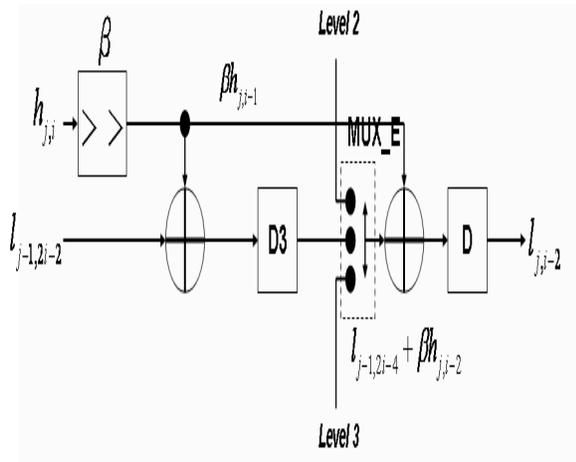


Fig. 4: Update module of (5,3) filter

### 3.3. Register Module

The register module contains four sets of units denoted as register set A, register set B, register set C, and register set D. The main function of these register is to temporarily keep the data to meet the timing plan.

Register set A is used to keep the input samples for the computation of the coarse coefficients. Register set B and Register set C are used to temporarily store the data for computing the final detailed coefficients and coarse coefficients, respectively. Register set D is used to store the coarse coefficients generated by the predict module for the computation of the detailed coefficients of the next decomposition level.

### 3.4. Multiplexing Module

Multiplexers can ensure the re-use of the hardware resource and that the correct sample joins the associated computation according to the timing plan. In the architecture proposed in this article, there are 5 multiplexes, denoted as MUX\_A, MUX\_B, MUX\_C, MUX\_D, and MUX\_E. Under the timing control, the multiplexer will guide the samples to the correct path to join the proper computation to get the high-passed or low-passed coefficients.

## 4. FPGA implementation and performance analysis

### 4.1. FPGA implementation

The architecture proposed in this article is presented in Fig 5. In order to verify if the proposed architecture can work correctly, we encode the architecture with Verilog HDL and then implement it on the platform of Quartus-II of version 5.0. The image for testing is the famous Lena with block size of  $64 \times 64$ . With the width of the data bus being 8 bits, the function and logic simulation are carried out on the platform. Then the Verilog HDL code is synthesized and transferred to Cyclone FPGA Chip. The result is compared with the associated output generated by MATLAB's program.

### 4.2. Performance analysis

In Table 1, the results with respect to the hardware conditions reported by the proposed paper are shown and compared with those reported by papers [3]—[4]. When compared with the architectures in [3] and [4], the proposed architecture requires the same number of adders and multipliers (shifters). However, unlike the paper [3] requiring the delay up to  $2 \cdot T_a + T_{co} + T_{su}$ , the architecture proposed in this article can shorten the delay of the critical path to a  $T_a$  due to the usage of the pipelined scheme, where  $T_a$  is the delay of an adder,  $T_{co}$  is the delay of a register, and  $T_{su}$  is the setup time of a register. Besides, the proposed architecture doesn't need extra memories which are instead required by [4]. The proposed architecture was successfully synthesized using Cyclone device family from Altera Corp. Table 2 shows the simulation results of the synthesis report generated by Quartus-II of version 5.0.

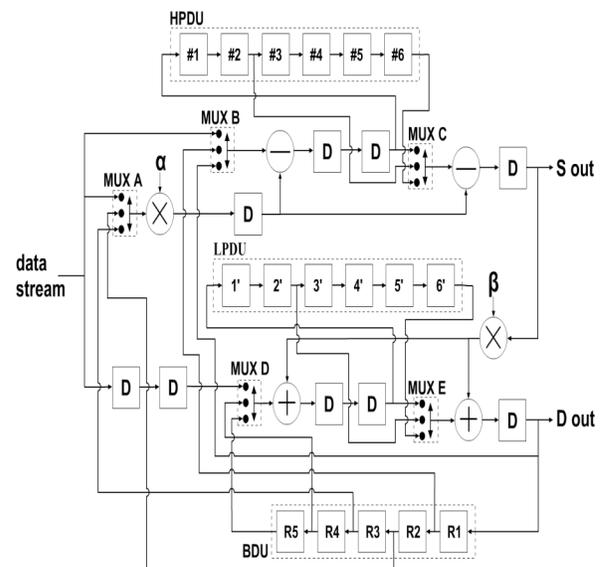


Fig. 5: Architecture of (5,3) wavelet transform

## 5. Conclusion

The lifting scheme has the advantages of less operation of in-place computation. In this paper, an efficient DWT architecture utilizing folded and pipelined method has been proposed. The architecture has been verified successfully and realized with the FPGA device of Cyclone family from Altera Corp.

## 6. References

- [1] S. Mallat, "Multifrequency Channel Decompositions of Images and Wavelet Models," IEEE Trans. Acoustics, Speech, and Signal Processing, vol. 37, pp.2091-2110, 1989.
- [2] G. Sweldens, "The Lifting Scheme: A Custom-Design Construction of Biorthogonal Wavelet," Applied and Computational Harmonic Analysis, vol. 3, pp. 186-200, 1996.
- [3] P. Y Chen, "VLSI implementation for one-dimensional multilevel lifting-based wavelet transform," IEEE Trans. on Computers, vol. 53, no. 4, pp. 386-398, April 2004.
- [4] K. Andra, C. Chakrabati and T. Acharya, "A VLSI Architecture for Lifting-based Forward and Inverse Wavelet Transform," IEEE Trans. on Signal Processing, vol. 50, no. 4, pp966-977, April 2002.
- [5] C.-T. Huang, P.-C. Tseng, and L.-G. Chen, "Efficient VLSI Architectures of Lifting-Based Discrete Wavelet Transform by Systematic Design Method," Proc. IEEE Int'l Symp. Circuits and System, vol. 5, pp.565-568, May 2002.
- [6] ISO/IEC,JPEG2000 Committee Drafts

Table2: Synthesis report

Synthesis Report	
FPGA	Altera CycloneII EP2C35F672C8
total logic element	249/33,216(<1%)
total registers	221
total memory bits	0/483,840(<0%)
Clock Setup	209.38MHz(period=4.776ns)
total pins	26/475(5%)

Table1: Comparison table

	[3]	[4]	proposed
Adders	4	4	4
Multipliers/ Shifters	2	2	2
Switches	4	memory controller	5
clock period	$2T_a+T_{co}+T_{su}$	$T_a+T_{co}+T_{su}$	$T_a+T_{co}+T_{su}$
Storage Element	Register	Memory+ Register	Register