A COST-ERROR OPTIMIZED ARCHITECTURE FOR 9/7 LIFTING BASED DISCRETE WAVELET TRANSFORM WITH BALANCED PIPELINE STAGES

Alireza Aminlou, Fatemeh Refan, M.R. Hashemi, Omid Fatemi, Saeed Safari

Multimedia Processing Laboratory, School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran

ABSTRACT

Discrete Wavelet Transform (DWT) is increasingly recognized in image/video compression standards, as indicated by its use in JPEG2000. The lifting scheme algorithm is an alternative DWT implementation that has a lower computational complexity. In this paper, a new high performance lifting-based architecture is presented for the 9/7 DWT engine. The proposed architecture has a balanced pipeline and improves both the computational error and hardware complexity for any given working frequency. In the proposed architecture, the constant coefficients are modified by introducing new variables to the conventional lifting structure to minimize hardware cost and computational error, imposed by quantization of coefficients. Simulation results indicate a quality improvement of up to 15 dB when compared to an architecture using the standard coefficients that has the same hardware cost and working frequency. Similarly, the hardware cost is reduced by about 20% when both architectures deliver the same PSNR when operating at the same frequency.

Index Terms— Discrete wavelet transform, Constant multiplier, Lifting-based architecture, Balanced pipeline

1. INTRODUCTION

The rapid growth of visual media in many applications has led to a variety of image and video compression standards. Wavelet transform is a domain transform that separates high and low frequency characteristics of an image to further improve the coding efficiency. The Discrete Wavelet Transform (DWT) has become a popular domain transform in signal and image processing. Convolution is the conventional method to implement DWT, while the lifting scheme, initially proposed in [1], is more efficient DWT implementation method. The lower computational complexity and reduced memory requirements of liftingbased DWT have made it the best choice for hardware implementations.

While several convolution-based architectures are introduced in [2], most DWT architectures are based on the lifting scheme [3], including one-dimensional (1-D) and two

dimensional (2-D) implementations [4]. For 1-D DWT, [5] has mapped the lifting structure directly into a pipelined architecture, but according to [6] by folding the last two pipeline stages, full hardware utilization is achieved. [7] and [8] provide optimized architectures that can be used in a wide range of different filters. On the other hand, in order to optimize the lifting-based critical path (DWT engine), a flipping architecture [9] has been introduced, in which the critical path and memory requirements are reduced by scaling the constant coefficients. [10], [11] and [12] have also focused on the efficient quantization of constant multipliers and their effects on the performance.

Although many studies have been performed on the lifting structure, only few of them have focused on either optimizing the computation engine on the basis of modifying the constant coefficients [9], or the effect of quantizing them [10], [11], and [12]. In the pipeline architectures proposed in [5] and [6] the working frequency has been improved by reducing the critical path in the slowest stage. Although, hardware cost and computational error of 9/7 DWT has been optimized in [14], the pipeline depth has not been considered to obtain a balanced architecture.

The computation engine of the lifting scheme of 9/7 DWT consists of a number of large constant multipliers, whose hardware implementation is area and power consuming. In this paper, a split structure is proposed which offers a flexible method for designing a balanced-pipeline architecture with optimized cost-error performance for the computation engine of the lifting method. In the proposed technique the value of the original transform coefficients have been modified to achieve an optimized hardware cost and transform error, while the architecture remains balanced with regards to the pipeline depth. The suitable coefficients for each working frequency can then be determined according to the hardware cost or quality requirements of each application.

The paper is organized as follows: In section 2, 1-D liftingbased 9/7 DWT structure and constant multiplier issues are introduced. In section 3, the proposed split structure and the corresponding optimization procedure are presented. The achieved performance and efficient coefficients are demonstrated in section 4, followed by conclusions.

2. BACKGROUND

In this section, first we introduce the fundamental concepts of the lifting-based wavelet. Next, Canonical Signed Digit (CSD) as a representation which offers the lowest hardware cost for constant multipliers is introduced. Finally, we explain the important issues on constant multipliers as the most important building block of the lifting-based wavelet.

2.1. Lifting Structure of 1-D 9/7 DWT

The 2-D DWT operation consists of two 1-D wavelet transforms that are being applied consecutively in the vertical and horizontal directions. Outputs of this module, Y_i s, are calculated according to equation (1) in six steps, where X_i s are inputs of the engine and P, Q, R and S are internal nodes.

$$P_{2n+1} = X_{2n+1} + \alpha (X_{2n} + X_{2n+2})$$

$$Q_{2n} = X_{2n} + \beta (P_{2n-1} + P_{2n+1})$$

$$R_{2n+1} = p_{2n+1} + \gamma (Q_{2n} + Q_{2n+2})$$

$$S_{2n} = Q_{2n} + \delta (R_{2n-1} + R_{2n+1})$$

$$Y_{2n+1} = KR_{2n+1} \qquad Y_{2n} = S_{2n} / K$$
(1)

The main path of the structure for lifting-based 1-D transform consists of five constant multipliers, which is partitioned to five pipeline stages as shown in Figure 1-(a). As each stage includes one multiplication by a constant coefficient and one addition, the working frequency of the pipeline architecture is determined by the slowest constant multiplier. Hence it is important to balance constant multipliers, in addition to reducing the overall area and computational error.

2.2. Constant Multiplier: Cost, Error and Delay

Constant multiplier is the most important and areaconsuming module of 9/7 lifting structure. Array multiplier as the primary architecture of multipliers consists of a number of row of adders. We simply define the hardware cost as the number of adders. In order to minimize the defined cost, CSD as the representation with the minimum number of '1' bits for every single constant multiplier has been proposed in [13]. As a result, the exact hardware cost is modeled by (2), where *const_mult_ones* is the number of '1' bits in the CSD representation. Another characteristic of constant multiplier is the error that exists due to the quantization of real numbers in the implementation. The mean square error of outputs is defined as (3), where z and z' are the ideal and the real outputs, respectively.

Delay is another characteristic of a constant multiplier. Delay determines the working frequency and is related to the effective bit width (EBW) of the constant coefficient, which is equal to the number of bits starting from the first leftmost '1' bit to the rightmost '1' bit. If two constant coefficients have equal effective bit width, they are balanced; E.g. two numbers a=0.011011000100, and b=110000.101000 are balanced with the effective bit width of eight.

$$CM_Cost = const_mult_ones - 1$$
 (2)

$$Error = \sum (z - z')^2 \tag{3}$$



Figure 1. The main calculation path of 1 D lifting-based structure: (a) Standards structure. (b) Proposed structure

3. THE PROPOSED METHOD

In this section, the proposed method that intends to optimize the hardware cost and error of lifting structure in 1-D 9/7 DWT is presented. In the first subsections the main idea to modify the constant multiplicands of the 1-D structure, in order to achieve a balanced architecture with lower overall hardware cost and computational error is introduced. Later the corresponding optimization procedure is explained.

3.1. Split Structure: The Main Idea

In the 1-D lifting-based structure of 9/7 DWT, the coefficient of each constant multiplier is a real number. These coefficients should be quantized for hardware implementation, hence the outputs would never be precise. On the other hand the results are calculated in a series of addition and multiplication of inputs and coefficients. Considering the above issues, we propose to change these internal coefficients in order to reach better values offering lower cost and lower computational error. In other words, each coefficient is changed with respect to other coefficients such that the final output remains unchanged. Figure 1-(b) demonstrates the proposed change. This change results in new coefficients with the definition of equation (4), where $T_n(value)$ is the truncation of value by *n* bits; The truncation value *n* is equal for all coefficients in order to have a balanced-pipeline architecture.

3.2. Split Structure: Cost and Error

The hardware cost of the Split structure is the sum of hardware costs of all constant multipliers. But computing error is more complicated. The exact value of error can be estimated using the equation in (5). For more details and definition of $E_{\rm H}$ and $E_{\rm L}$, please refer to [14].

$$E_{Total} = E_H + 2.0838E_L \tag{5}$$

3.3. Optimization Process

The optimization process, Figure 2, is used to determine the optimum coefficients in design step. We have to use the same effective bit width for constant multipliers to balance the pipeline stages. On the other hand a trade-off exists between error and cost of these constant multipliers.

In order to optimize the standard structure (Figure 1-(a)) for a given frequency, first the delay of the critical path should be determined; which is proportional to the effective bit width of each constant multiplier. Therefore, constant multipliers are chosen with the same effective bit width as they have the same delay and working frequency. Then the corresponding hardware cost and computational error is calculated. Using this process, for any given working frequency, there is only one optimum point in standard structure.

Similarly for the proposed architecture (Figure 1-(b)), the effective bit width of the constant coefficients should be chosen based on the given working frequency. But in this case the values of constants are chosen in the optimization process. Therefore the proposed architecture results in a set of solutions, offering different hardware cost and error for each working frequency. Consequently, depending on the frequency, cost, and error requirements of each application, we can choose the corresponding suitable coefficient set. The optimization process is presented in Figure 2.

```
Optimize (n: effective bit width){
For all m1 values (OBN precision)
For all m2 values (OBN precision)
For all m3 values (OBN precision)
For all m4 values (OBN precision)
calculate all ten coefficients (4):
    n1,n2,n3,n4,α',β',γ',δ',K',(k<sup>-1</sup>)'
Calc cost-error //(2), (5)
Add to list if qualified
```

Figure 2. The pseudo code of optimization process

4. SIMULATION RESULTS

Simulations were performed to evaluate the proposed method. The bit width of input data and internal nodes are not important in our model in (2). The constant coefficients of 9/7 lifting structure are considered as positive values with high precision as presented in Table 2.

Initially the performance of the standard structure for each specific frequency is determined. The effect of quantizing the original coefficients on cost and error of the lifting structure of 1-D 9/7 DWT can be found by choosing m1 to m4 variables equal to '1' in optimization procedure of Figure 2. Next, we derived a set of optimum results for the proposed structure using the modified coefficients. For this simulation, 10, 8, and 5 bits are reserved for m1 to m4 variables as optimization bit number (OBN) and are changed from 0.5 to 1.0. The simulation with OBN=10 is more accurate and finds better results, while it takes about one week for each pipeline depth on a Pentium-IV 3.00GHz with 1.0GB of RAM. On the other hand the simulation with

OBN=8 is less accurate with a faster execution time of 40 minutes on the same PC.

Both simulations are executed for various working frequencies by choosing the effective bit width of multiplier from 4 to 20. In order to calculate the improvement achieved by the proposed method, the results of these two simulations are compared in Figure 3 for pipeline depths of 11 and 12. In these figures, the horizontal axis represents the hardware cost which is the total number of '1' bits of coefficients. The vertical axis demonstrates the corresponding PSNR quality in dB, calculated from (5), STD and OBN=i represent the diagrams offered for standard and proposed architectures with effective bit width of *i*, respectively.

The diagrams show that the STD diagram offers only one point for each pipeline depth with a particular cost and error, while the proposed method results in a set of points for any pipeline depth. For example, in Figure 3, OBN=10 offers 11 choices with different cost and error. Furthermore, the proposed method improves the quality by 14 dB in average, without increasing the hardware cost (=20). Similarly, the hardware cost is reduced from 20 to 16 where the PSNR of both methods is 42 dB. A similar performance is for other values of pipeline depth. The PSNR improvements are not significant for OBN=5 in Figure 3, because only five bits is used for m_1 to m_4 . It is inferred that higher performance can be achieved using a heavier simulation with higher optimization bit number (OBN).

As an example, three points of optimization process are presented in Figure 3 for pipeline depth=11 and optimization bit number=8 for m_1 to m_4 . The middle column (STD) represents the original coefficients, which is also shown in Figure 3-(a) as STD. Left and right columns belong to the proposed structure, having the same hardware cost and quality with the standard result, respectively. These two points are shown in Figure 3-(a) on OBN=8 curve by white solid squares. Decimal values of constant multipliers and their corresponding binary representations are shown. Although coefficients are presented in binary format, their CSD representations are used for hardware cost calculation. Also, hardware costs are shown inside ().



Figure 3. Comparison of cost-quality diagram of proposed and standard structures (Pipeline depth=EBW=11)

According to this table, despite the equal cost of 20, offered in the left column, the proposed method has improved the quality by about 10 dB. When compared to the right column, it shows that in addition to 1.3 dB improvement in quality, the hardware cost is reduced by about 15%.

5. CONCLUSION

This paper addresses the trade-off between hardware cost, and computational error of a 1-D lifting-based 9/7 DWT engine in a balanced-pipeline architecture. In order to improve the performance of the lifting structure, a split architecture, in which the constant coefficients were changed to gain cost-error improvement, has been proposed. As a result, four new parameters have been added and the other six coefficients have been modified. The values of all ten coefficients were chosen carefully for different working frequencies using an optimization process. Simulation results show that the proposed method improves computational quality up to 15 dB when compared to the standard architecture, having the same hardware cost and working frequency. Similarly, the hardware cost is reduced by about 20% when both architectures deliver the same PSNR and working frequency. This method also offers a set of choices with different hardware cost and PSNR that can be selected according to the area constraint and the requirement of the application.

Table 1. Examples for comparison of the proposed and standard methods for, same cost and same PSNR (effective bit width = 11, atimization bit number = 8)

	Proposed (Same cost)	Standard (STD)	Proposed (Same PSNR)
Cost	20	20	17
PSNR	52.49	42.46	43.7
m1	0.5		0.5703125
m2	0.5		0.5
m3	0.56640625		0.5703125
m4	0.62109375		0.8125
α	11001011 (4)	11001011 (4)	111001111 (3)
β×10	11011001 (4)	11011001 (4)	1011111001 (3)
γ	1 (0)	1110001 (2)	1000000111(2)
δ	11111001 (2)	11100011 (3)	1010000111 (3)
k	10001011 (3)	10011101011 (5)	1000101 (2)
k ⁻¹	101001111 (3)	1101 (2)	1 (0)
n1	1 (0)		1001001 (2)
n2	1 (0)		1 (0)
n3	10010001 (2)		1 (0)
n4	10011111 (2)		1101 (2)

Numbers inside () are hardware costs defined in (2).

Table 2. Constant coefficients of 9/7 lifting structure

α	1.586134342059924	δ	0.443506852043971	
β	0.052980118572961	k	0.6150870524570005	
γ	0.882911075530934	k ⁻¹	0.812893066115961	

REFERENCES

- W. Sweldens, "Wavelets and the lifting scheme: A 5 minute tour," Z. Angew. Math. Mech., vol. 76, no. 2, pp. 41–44, 1996.
- [2] Chakrabarti C, Vishwanath M, Owens RM, "Architectures for wavelet transforms: A survey", Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology, vol. 14, no. 2, pp. 171-192, November 1996.
- [3] Acharya T, Chakrabarti C, "A survey on lifting-based Discrete Wavelet Transform architectures", Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology, vol. 42, no. 3, pp. 321-339, 2006.
 [4] Zervas ND, Anagnostopoulos GP, Spiliotopoulos V, et
- [4] Zervas ND, Anagnostopoulos GP, Spiliotopoulos V, et al. "Evaluation of design alternatives for the 2-Ddiscrete wavelet transform" IEEE Tran. on Circuits and Systems for Video Technology, vol. 11, no. 12, pp. 1246-1262, 2001.
- [5] C. C. Liu, Y. H. Shiau, and J. M. Jou, "Design and implementation of a progressive image coding chip based on the lifted wavelet transform," Proc. of the 11th VLSI Design/CAD Symposium, Taiwan, 2000.
- [6] C.J Lian, K. F. Chen, H. H. Chen, and L. G. Chen, "Lifting based discrete wavelet transform architecture for JPEG2000", IEEE International Symposium on Circuits and Systems, Sydney, pp. 445-448, 2001.
- [7] M. Martina, G. Masera, G. Piccinini, and M. Zamboni, "Novel JPEG2000 Compliant DWT and IWT VLSI Implementations," Journal of VLSI Signal Processing, vol. 34, pp. 137-153, 2003.
- [8] K. Andra, C. Chakrabarti, and T. Acharya, "A VLSI Architecture for Lifting-Based Forward and Inverse Wavelet Transform," IEEE Trans. of Signal Processing, Vol. 50, No. 4, pp. 966-977, 2002.
 [9] C.T. Huang, P.C. Tseng, and L.G. Chen, "Flipping
- [9] C.T. Huang, P.C. Tseng, and L.G. Chen, "Flipping structure: an efficient VLSI architecture for liftingbased discrete wavelet transform,"IEEE Transactions on Signal Processing, pp. 1080-1089, April 2004.
- [10] K. A. Kotteri, A. E. Bell and J. E. Carletta, "Improving the Performance of the Quantized Biorthogonal 9/7 Wavelet Filters," Proceedings of the 10th IEEE Digital Signal Processing Workshop, Pine Mountain, GA, October 2002.
- [11] V. Spiliotopoulos, N. D. Zervas, Y. Andreopoulos, G. Anagnostopoulos, and C. E. Goutis, "Quantization Effect on VLSI Implementations for the 9/7 DWT Filters," Proc. ICASSP, vol. 2, pp. 1197–1200, 2001.
- [12] S. Barua, K. A. Kotteri, A. É. Bell and J. E. Carletta, "Optimal, Quantized Lifting Coefficients for the Biorthogonal 9/7 Wavelet," Proceedings of ICASSP, Montreal, Canada, vol. V, pp. 193-196, May 2004.
- [13] Israel Koren, "Computer Arithmetic Algorithms", A. K. Peters, 2nd edition, 2001.
- [14] A. Aminlou, F. Refan, M. Homayouni, O. Fatemi, M.R. Hashemi, "A Split Method for Optimized Cost-Quality Hardware Implementation of Lifting-Based Discrete Wavelet Transform", Proceedings of ICASSP, Hawai'i, USA., vol. 2, pp. 89-92, 2007.