A SPLIT METHOD FOR OPTIMIZED COST-QUALITY HARDWARE IMPLEMENTATION OF LIFTING-BASED DISCRETE WAVELET TRANSFORM

Alireza Aminlou, Member, IEEE, Fatemeh Refan, Maryam Homayouni, Student Member, Omid Fatemi, Member, IEEE, Mahmoud Reza Hashemi, Member, IEEE

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 with optimized error vs. hardware complexity is presented for DWT. The proposed architecture modifies the constant coefficients by introducing new variables to the conventional lifting structure to minimize hardware cost and quantization error. In order to achieve the most efficient coefficients, an optimization process has been implemented. Simulation results indicate an average quality improvement of 7.5 dB with the same hardware complexity/cost. Similarly, for achieving the same quality as the conventional hardware implementations the proposed architecture is 20% less complex. The appropriate coefficients can be determined according to the cost and error requirements of each application.

Index Terms— Discrete wavelet transform, constant multiplier, lifting-based architecture

1 INTRODCUTION

The rapid growth of visual media in many applications has led to a variety of image and video compression standards. Domain transform is one of the primary parts of any image encoder. 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. Frequency localization and multi-resolution structure are some of the main features of DWT that has made it suitable for image compression. Convolution is the conventional method to implement DWT, while the lifting scheme, initially proposed in [1], is an efficient DWT implementation method. The lower computational complexity and reduced memory requirements of lifting-based 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. In addition, an implementation using MACs (Multiply and Accumulate) and registers is proposed in [7]. The dual scan architecture in [8] achieves the utilization of 100% in the hardware data path, while [9] and [10] 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), flipping architecture [11] is introduced, in which the critical path and memory requirements are reduced by scaling the constant coefficients. [12], [13] and [14] have also focused on the efficient quantization and its effect 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 [11], or the effect of quantizing them [12], [13], and [14]. The computation engine of the lifting scheme consists of a number of 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 optimized cost-error architecture for the computation engine of the lifting method. In the proposed technique the original transform coefficients have been modified, in order to achieve an optimized hardware cost and transform error. The suitable coefficients can then be determined according to the cost or quality requirements of each application.

The paper is organized as follows: In section 2, related concepts including 1-D lifting-based DWT structure, constant multiplier issues and Canonical Signed Digit (CSD) representation as the optimum method for constant multipliers 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, 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. Therefore, 1-D lifting-based transform can be considered as the core of any DWT module. Outputs of this module, Y_is, are calculated according to equation (1) in six steps, where X_is are inputs of the engine and P, Q, R and S are internal nodes.

$$\begin{aligned} P_{2n+1} &= X_{2n+1} + \alpha(X_{2n} + X_{2n+2}) \\ Q_{2n} &= X_{2n} + \beta(P_{2n-1} + P_{2n+1}) \\ R_{2n+1} &= p_{2n+1} + \gamma(Q_{2n} + Q_{2n+2}) \\ S_{2n} &= Q_{2n} + \delta(R_{2n-1} + R_{2n+1}) \\ Y_{2n+1} &= KR_{2n+1} \qquad Y_{2n} = S_{2n} / K \end{aligned} \tag{1}$$

The equivalent architecture of lifting-based 1-D transform is illustrated in Figure 1 where each arrow represents a multiplication by a constant factor, specified by its label number. According to this structure the main path from inputs to the high and low outputs consists of nine constant multipliers. These constant multipliers are area and power-consuming which highlights the importance of optimizing this structure.

2-2 Constant Multiplier: Cost and Error

Constant multiplier is the most important and area-consuming module of 9/7 lifting structure. Array multiplier as the primary architecture of multipliers consists of a number of rows adders. The area occupied by an array multiplier is equal to the total number of 1-bit full adders multiplied by the area of a 1-bit adder. To put it more simply, the needed area is proportional to the number of 1-bit adders. In addition, in a constant array multiplier the number of 1-bit full adders in each row is equal to the bit width of the variable value. As a result, we simply define the hardware cost as the number of 1-bit full adders, which depends on the representation of the constant value. 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 [15]. It takes advantage of the two common methods, normal and booth. As a result, the exact hardware cost can be modeled by (2), where *const_ones* is the number of '1' bits in the CSD representation.

$$CM\_Cost = const\_ones - 1 \tag{2}$$

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

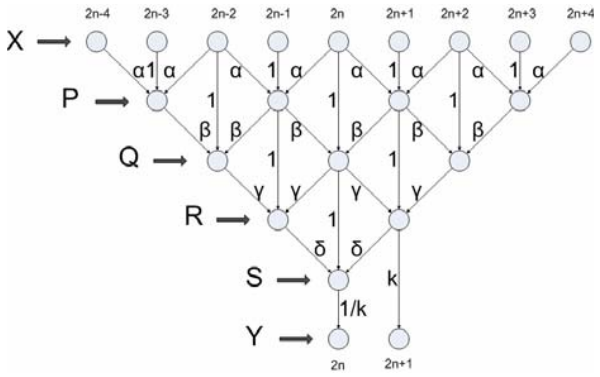


Figure 1. The lifting-based structure of 9/7 DWT filter

The other important factor in constant multiplier is error that exists due to the quantization of real numbers in the implementation. The error in an arithmetic operation for a single calculation stage is defined as the difference of the real output and ideal output. The ideal output is the result of the arithmetic operation with actual inputs, and the real output is achieved from the quantized multiplier. The mean square error of outputs is defined as (3), where z and z' are the ideal and the real outputs, respectively.

3 THE PROPOSED METHOD

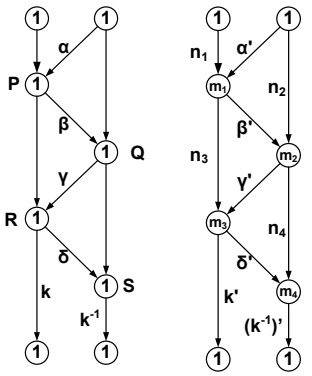
In this section we propose a method that intends to optimize the hardware cost and error of lifting structure in 1-D 9/7. First, we propose the main idea to modify the constant multiplicands of the 1-D structure, in order to achieve a lower overall hardware cost and error. Second, the corresponding optimization procedure is explained.

3-1 Split Structure

In the 1-D lifting-based structure of 9/7 DWT shown in Figure 1, the coefficient of each constant multiplier is a real number. These coefficients should be quantized for hardware implementation, so 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 to find better values offering lower cost and higher quality performance. In other words, each coefficient is changed depending on the others, in a way that the final output remains unchanged. Figure 2 demonstrates the proposed change. In this figure the labels on the internal nodes represent their new weight instead of '1'. 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 main path from inputs to outputs in the proposed structure, Figure 2-b, includes four additions (*P*, *Q*, *R* and *S* nodes), and ten multiplications. First, the two inputs are multiplied by m_1 and $m_1\alpha$ and added through the P node. Thus the output of this node is m_1 times more than the node P of Figure 2-a. P node is then multiplied by β' ($\beta m_2/m_1$) and added through the node *Q* with the result of the multiplication of second input by m_2. In this way the effect

of the multiplication by m_1 is eliminated and the new coefficient m_2 is the multiplied value in Q node, and so on. In this structure, in every step the effect of the previous coefficient is omitted and a new one is introduced, consequently the outputs will remain unchanged.



(a) (b)

$$
\begin{aligned}
n_1 &= T_n(m_1) \\
n_2 &= T_n(m_2) \\
n_3 &= T_n(m_3/m_1) \\
n_4 &= T_n(m_4/m_2) \\
\alpha' &= T_n(\alpha \times m_1) \\
\beta' &= T_n(\beta\, m_2/m_1) \\
\gamma' &= T_n(\gamma\, m_3/m_2) \\
\delta' &= T_n(\delta\, m_4/m_3) \\
(k^{-1})' &= T_n(k^{-1}/m_4) \\
k' &= T_n(k/m_3)
\end{aligned}
\qquad (4)
$$

Figure 2. The main calculation path of 1 D lifting-based structure: (a) Standards structure. (b) Proposed structure

3-2 Optimization Process

In order to optimize the lifting architecture, we have to minimize error and hardware cost. But there is a trade-off between quality and cost. In order to minimize error, we have to use coefficients with higher precision and this requires higher hardware cost and vice versa. This trade-off leads us towards a set of solutions instead of a single result, where each solution offers a specific quality and cost. Depending on the cost or error requirements of each application, we can choose the corresponding suitable coefficient set.

The optimization procedure, shown in Figure 3, has 14 loops. It has four new variables, m_1 to m_4, which should be chosen carefully using the first four loops. Next, the new coefficients are calculated, according to the equations of Figure 2. Then the bit widths of all ten modified coefficients are changed in individual loops. Finally, for every choice, cost and error are estimated. This choice is then added to the selected coefficient sets if it offers a better cost or error comparing to existing ones. Using the optimization process the best coefficients and their corresponding bit widths will be obtained for any cost or error constraint.

As explained in section 2-2, the hardware cost of multipliers is defined as the total number of '1' bits of constant multipliers when the CSD representation is used for constant values. But computing error is more complicated. The exact value of error can be estimated using the equation in (5). According to this equation, total error is equal to the weighted sum of errors of low and high outputs. Weight of low error is 2.0838 due to L2-Norm factor. Equation (9), the exact formula to calculate low output error is derived from equations (6) to (8), where z_{Li} and z'_{Li} are ideal and real low outputs, CL_i and CL'_{Li} represent ten ideal and truncated convolution coefficients based on the lifting coefficients, d_{Li} denotes the difference of z_{Li} and z'_{Li}, and N is the number of all possible cases of input value (e.g. 255). After expansion of (9), there are x_ix_i, x_ix_{i+1} …, and x_ix_{i+8} terms which are the result of autocorrelation between the values of input signals such as an image. We note that all autocorrelation factors (x_ix_{i+k}) are proportional to x_ix_i, thus all of them can be estimated by x_i^2, which leads to the simplified equation (9). The high output error (E_H) can be calculated in the same manner using (11).

```
for all m1 values
 ...
  for all m4 values
   calculate all ten coefficients (4):
   n1,n2,n3,n4,α',β',γ',δ',K',(k^-1)'
   for all bit_width of n1
    for all bit_width of n1
     ...
      for all bit_width of (k^-1)'
       Calc cost-error //(2), (5)
       Add to list if qualified
```

Figure 3. The pseudo code of the optimization process

$$E_{Total} = E_H + 2.0838E_L \qquad (5)$$

$$E_L = \sum (z_{Li} - z'_{Li})^2 \qquad (6)$$

$$z_{Li} = \sum_{k=-4..+4} CL_k \times x_{i+k} \qquad z'_{Li} = \sum_{k=-4..+4} CL'_k \times x_{i+k} \qquad (7)$$

$$CL = CL_i(\alpha, \beta, \gamma, \delta, k, k^{-1}) \qquad (8)\text{a}$$

$$CL'_k = CL'_{ki}(\alpha', \beta', \gamma', \delta', k', k^{-1}{}', n_1, n_2, n_3, n_4) \qquad (8)\text{b}$$

$$E_L = \sum_N \Big(\sum_{k=-4..+4} dL_k x_{i+k} \Big)^2 \qquad dL_k = CL_k - CL'_k \qquad (9)$$

$$E_L \approx \sum_N x_i^{\,2} \times \left[\sum_{k=-4..+4} dL_k \right]^2 = K \times \left[\sum_{k=-4..+4} dL_k \right]^2 \qquad (10)$$

$$E_H \approx \sum_N x_i^{\,2} \times \left[\sum_{k=-3..+3} dH_k \right]^2 = K \times \left[\sum_{k=-3..+3} dH_k \right]^2 \qquad (11)$$

4 SIMULATION RESULTS

Our simulation includes two parts. First we find the effect of quantizing the original coefficients on cost and error of the lifting structure of 1-D 9/7 DWT. This result can be achieved by choosing m_1 to m_4 variables equal to '1' in optimization procedure of Figure 3. Next, we derived a set of optimum results using the modified coefficients. For this simulation, 5 bits are reserved for m_1 to m_4 variables and are changed from 0.5 to 1.0. In both simulations, the bit widths of all ten coefficients changes from 5 to 10. In order to calculate the improvement achieved by the proposed method, the results of these two simulations are compared in Figure 4. In this figure, the horizontal axis is 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).

The diagram of Figure 4 shows that the proposed method improves the quality by 7.5 dB in average, without increasing the hardware cost, and reduces hardware cost by 80% of the original structure for the same PSNR quality. As an example, three points of Figure 4 are highlighted in Table 1. The first point represents the original coefficients with optimized quantization, while the two others belong to the proposed structure derived from the optimization process. According to this table, despite the equal cost offered by the first and second points, the proposed method has improved the quality by about 11.29 dB (40%). On the other hand, comparing the first and the third points shows that in addition to 2.58 dB improvement in quality, the cost is reduced by about 19%.

5 CONCLUSION

This paper addressed the trade-off between hardware cost, and quality of a 1-D lifting-based DWT engine. In order to improve the performance of this structure we proposed a split architecture, in which the constant coefficients were changed to gain cost-error improvement. As a result, four new parameters have been added and the other six coefficients have been modified. The values and bit width of all ten coefficients were chosen carefully using an optimization process, consisting of 14 loops. Simulation results show an average quality improvement of 7.5 dB with the same hardware complexity. Similarly, for achieving the same quality as the conventional hardware implementations the proposed architecture is 20% less complex.

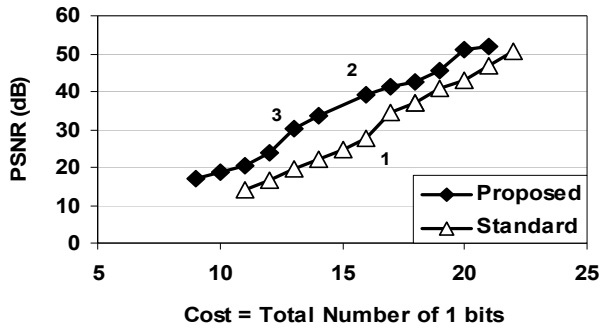


Figure 4. Comparison of cost-quality diagram of proposed and standard structures.

Table 1. Output examples from the optimization process

		1: Standard		2: Proposed		3: Proposed	
Cost, Q		16, 27.76		16, 39.05		13, 30.34	
α		1.578125		1.189453125		1.08984375	
β		0.052734375		0.0352783203125		0.0625	
γ		0.8828125		0.9921875		0.74609375	
Δ		0.443359375		0.640380859375		0.5234375	
K		0.61328125		1.09326171875		0.890625	
k^{-1}		0.8125		1.0		1.0	
n_1	n_3	--	--	0.75	0.75	0.6875	1.0
n_2	n_4	--	--	0.5	1.625	0.8125	1.0

REFERENCES

[1] 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 al. "Evaluation of design alternatives for the 2-D-discrete 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] W. H. Chang, Y. S. Lee, W. S. Peng, and C. Y. Lee, "A line-based, memory efficient and programmable architecture for 2D DWT using lifting scheme," IEEE Int. Symposium on Circuits and Systems, Australia, pp. 330-333, May 2001.

[8] H. Liao, M. K. Mandal, and B. F. Cockburn, "Efficient Architectures for 1-D and 2-D Lifting-Based Wavelet Transform", IEEE Trans. on Signal Processing, Vol. 52, No 5, pp. 1315-1326, 2004.

[9] 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.

[10] 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.

[11] C.T. Huang, P.C. Tseng, and L.G. Chen, "Flipping structure: an efficient VLSI architecture for lifting-based discrete wavelet transform,"IEEE Transactions on Signal Processing, pp. 1080-1089, April 2004.

[12] 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.

[13] 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.

[14] S. Barua, K. A. Kotteri, A. E. 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.

[15] Israel Koren, "Computer Arithmetic Algorithms", A. K. Peters, 2nd edition, 2001.