B-SPLINE FACTORIZATION-BASED ARCHITECTURE FOR INVERSE DISCRETE WAVELET TRANSFORM

Chao-Tsung Huang, Po-Chih Tseng, and Liang-Gee Chen

DSP/IC Design Lab, Graduate Institute of Electronics Engineering and
Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan
Email:{cthuang, pctseng, lgchen}@video.ee.ntu.edu.tw

ABSTRACT

In this paper, the VLSI architecture for the Inverse Discrete Wavelet Transform (IDWT) is proposed on the basis of B-spline factorization that is the intrinsic property of DWT and comprises two parts: B-spline part and distributed part. After the polyphase decomposition, the former can be constructed by Pascal or direct implementation. And the latter one can be implemented by serial or parallel filter architecture. The B-spline-based architectures can reduce multipliers but would introduce additional adders compared with convolution-based architectures. Because the hardware complexity of adders is much less than that of multipliers, B-spline-based architectures could provide smaller hardware complexity for DWT and IDWT. The case study of the (10,18) filter will also be given to demonstrate the efficiency.

1. INTRODUCTION

DWT has been developed as an efficient DSP tool for signal analysis, image compression, and video compression [1]. In the last decade, many VLSI architectures have been proposed for DWT, which are mainly convolution-based [2] and lifting-based [3]. The former is to implement two-channel filter banks directly. And the latter is to exploit the relationship of lowpass and highpass filters to saving multipliers and adders [4, 5].

We have proposed a new category of forward DWT architectures based on the B-spline factorization [6]. According to [7], any DWT filter bank can be factorized into the B-spline part and distributed part. The B-spline part contributes to all important wavelet properties. And the distributed part is used to design FIR DWT filters. The B-spline-based architectures could provide fewer multipliers while the lifting scheme fails to reduce the complexity.

In this paper, we extend the idea of B-spline architectures for forward DWT to IDWT. The direct implementation of the B-spline part is also introduced because the Pascal implementation may be too complex to reduce hardware complexity. After the Type-I or Type-II polyphase decomposition [8], the distributed part can be implemented by serial or parallel filters. The organization of this paper is as follows. Section 2 reviews previous arts of DWT architectures. The B-spine factorization theory is described and extended to IDWT in section 3. The proposed VLSI architecture for IDWT is presented in section 4. And one case study of the (10,18) filter is given to demonstrate the efficiency in section 5. This paper is summarized in section 6.

This work was supported in part by MOE Program for Promoting Academic Excellence of Universities under the grant number 89E-FA06-2-4-8, in part by National Science Council, Republic of China, under the grant number 91-2215-E-002-035, and in part by the MediaTek Fellowship.

2. PREVIOUS DWT ARCHITECTURES

As illustrated in [6], there are three categories of architectures for DWT. And it is similar for IDWT. As shown in Fig. 1, the IDWT is equivalent to a synthesis filter bank which includes a lowpass filter $\widetilde{H}(z)$ and a highpass filter $\widetilde{G}(z)$. The convolution-based architecture can be constructed by use of polyphase decomposition as shown in Fig. 2. Then the four filters can be implemented by serial or parallel filters.

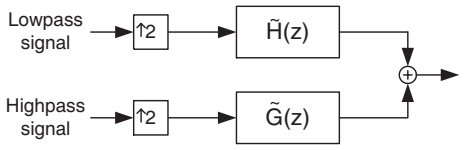


Fig. 1. Synthesis filter bank

On the other hand, lifting scheme [4] has been widely used to reduce the multipliers and adders by exploring the relation between lowpass and highpass filters. And any two-channel filter bank of perfect reconstruction property could be factorized into the corresponding lifting structures [5]. The IDWT architecture can be easily derived from the lifting-based architecture for DWT. However, the complexity re-

duction may fail for linear even-tap filter banks, such as the (6,10) filter [6]. This is because the lifting steps of these filter banks are not linear such that the number of required multipliers will be nearly the same as that of the convolution-based architectures that adopt the linear property.

We have proposed the third category of architectures for DWT based on B-spline factorization. It can still reduce the hardware complexity when the lifting scheme fails. The following sections will introduce the B-spline-based architectures for IDWT.

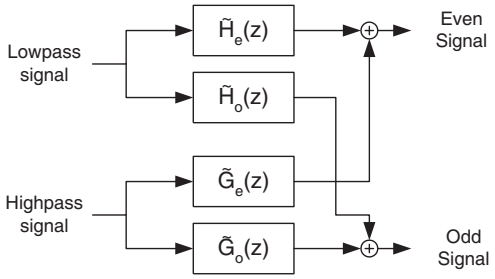


Fig. 2. Convolution-based architecture

3. B-SPLINE FACTORIZATION

According to [7], the lowpass filter $H(z)$ and the highpass filter $G(z)$ of any DWT can be factorized into the B-spline part and distributed part. And the perfect reconstruction implies that the synthesis lowpass filter $\widetilde{H}(z) = G(-z)$ and the synthesis highpass filter $\widetilde{G}(z) = -H(-z)$, which results in the similar equations as the DWT:

$$\begin{aligned} \widetilde{H}(z) &= (1+z^{-1})^{\gamma_{\widetilde{H}}} \cdot Q(z) \\ \widetilde{G}(z) &= (1-z^{-1})^{\gamma_{\widetilde{G}}} \cdot R(z) \end{aligned} \tag{1}$$

where the first and second terms of the right-hand side are the B-spline part and distributed part, respectively.

4. PROPOSED ARCHITECTURE FOR IDWT

In this section, the proposed architecture based on B-spline factorization is presented. The general architecture is introduced first, and then the direct implementation of the B-spline part is presented.

4.1. General architecture

After substituting the equation (1) into Fig. 1, the general architecture can be derived by use of polyphase decomposition [8] as shown in Fig. 3, where $Q(z) = Q_e(z^2) + z^{-1}Q_o(z^2)$ and $R(z) = R_e(z^2) + z^{-1}R_o(z^2)$. The four filters ($Q_e(z)$, $Q_o(z)$, $R_e(z)$, and $R_o(z)$) can be implemented by using many DSP VLSI techniques, such as serial filter, parallel filter, pipelining, and retiming.

There are two two-input-two-output B-spline parts. One is for $(1+z^{-1})^{\gamma_{\widetilde{H}}}$, and the other is for $(1-z^{-1})^{\gamma_{\widetilde{G}}}$.

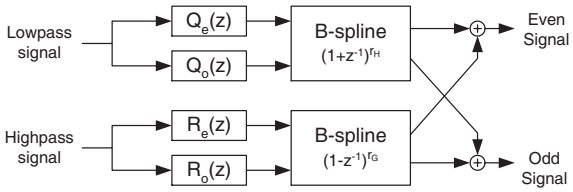
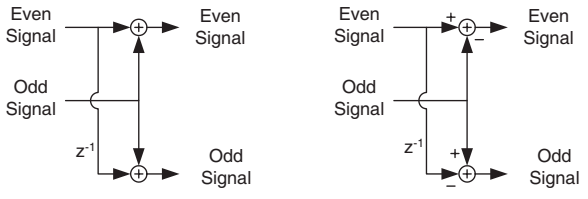


Fig. 3. General B-spline-based architecture

4.2. Direct implementation for B-spline part

In [6], the Pascal implementation of the B-spline part is discussed, which exploits the similarity of the two B-spline parts to reduce adders. However, the Pascal implementation of long-tap filters will be too complex to be derived, and the complexity reduction is not guaranteed. Then the direction implementation of the B-spline parts can be used instead.

The concept is to implement $(1+z^{-1})$ and $(1-z^{-1})$ first, and then the B-spline parts can be constructed by serially connecting $(1+z^{-1})$ and $(1-z^{-1})$. But two-input-two-output structures of $(1+z^{-1})$ and $(1-z^{-1})$ can not be derived from polyphase decomposition. We propose to implement them by considering the physical connection of signals as shown in Fig. 4. We assume the even signals are prior to odd signals. When connecting the B-spline part to the distributed part, the priority of signals need to be handled carefully.



(a) For $(1+z^{-1})$ (b) For $(1-z^{-1})$

Fig. 4. Direct implementation for the B-spline part

5. CASE STUDY

In this section, the case of the (10,18) filter is studied. As for the cases of the JPEG 2000 default (9,7) filter and the (6,10) filter, the results will be very similar with the cases of

the forward DWT [6] because the hardware complexities of forward and inverse DWT are very similar.

5.1. (10,18) filter

The coefficients of (10,18) analysis filter bank are given in [9]. The analysis lowpass filter is a symmetric 10-tap filter, and the highpass filter is an anti-symmetric 18-tap filter. The coding efficiency can be better than the well-known (9,7) filter [9, 10]. The synthesis filter bank can be derived from $\widetilde{H}(z) = G(-z)$ and $\widetilde{G}(z) = -H(-z)$.

After polyphase decomposition, the convolution-based architecture of (10,18) can be constructed as Fig. 5. $\widetilde{H}_e(z)$ and $\widetilde{H}_o(z)$ are mirror-images of each other, and so are $\widetilde{G}_e(z)$ and $\widetilde{H}_o(z)$. Thus, the number of multipliers can be reduced by using mirror-images, but it would require more registers. Here we discuss two architectures. The architecture I is to implement the four filters as parallel filters directly such as to minimize the number of registers. The required numbers of multipliers, adders, and registers are 26, 26, and 14, respectively. (h_0 and g_0 can be shared because they are parallel filters.) The architecture II is to minimize the number of multipliers by use of mirror-images. The required numbers of multipliers, adders, and registers are 14, 26, and 24, respectively. The critical path of these two architectures are both $T_m + 4T_a$, where T_m and T_a are the execution time taken for a multiplier and an adder, respectively.

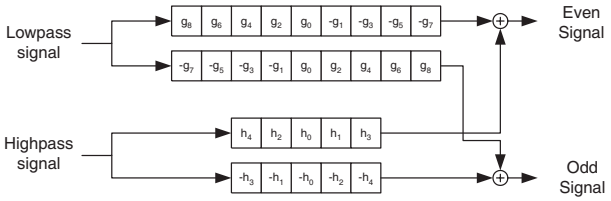


Fig. 5. Convolution-based architecture for (10,18) filter

5.2. Proposed architecture

The B-spline factorization of the synthesis filter bank is as follows:

$$\widetilde{H}(z) = \frac{(1+z^{-1})^9}{8}(v_1 z^{-8} + v_2 z^{-7} + v_3 z^{-6} + v_4 z^{-5} + v_5 z^{-4} + v_4 z^{-3} + v_3 z^{-2} + v_2 z^{-1} + v_1)$$
$$\widetilde{G}(z) = \frac{(1-z^{-1})^5}{4}(v_6 z^{-8} + v_7 z^{-7} + v_8 z^{-6} + v_7 z^{-5} + v_6 z^{-4}) \quad (2)$$

where $v_1 = 0.0076535, v_2 = -0.0687398, v_3 = 0.2681664$, $v_4 = -0.6004576$, $v_5 = 0.808888$, $v_6 = -0.1154104$, $v_7 = -0.57672$, and $v_8 = -1.0994$.

The proposed architecture of the synthesis (10,18) filter bank is shown in Fig. 6. The four filters of the distributed part are all symmetric so that the number of multipliers can be reduced into a half. The denominators 8 and 4 of the equation (2) are introduced only for the precision issues. These two denominators are implemented by shifting right the signals between the $(1 + z^{-1})$ and $(1 - z^{-1})$ stages, which are not shown in Fig. 6 for simplicity.

There are two possible implementations for these filters: serial or parallel filters. If the four filters are implemented as serial filters, the critical path will be T_m+11T_a. Because the registers can not be shared among these filters, the required number of registers is 24. On the other hand, the critical path will be $T_m + 13T_a$ and the registers can be shared and reduced to 20 if parallel filters are adopted. In Fig. 6, retiming can be performed to $R_e(z)$ and $R_o(z)$ to decrease the number of registers. And pipelining can also be used to shorten the critical path. In concept, the pipeline can be cut at the half of the critical path with four additional pipelining registers.

5.3. Comparison

The proposed B-spline factorized architectures as well as the aforementioned convolution-based ones have been verified by use of Verilog-XL and synthesized into gate-level netlists by Synopsys Design Compiler with standard cells from Artisan 0.25-μm cell library. The comparison and synthesis results are shown in Table 1, where the internal bit-widths are all 16-bit, the multipliers are all 16-by-16 multiplications, and the adders are also 16-bit for comparison. The gate counts are given in combinational and non-combinational gate counts separately. The former contributes to the multipliers and adders while the latter is responsible to the registers. For circuit synthesis, the timing constraints are set as tight as possible. The lifting-based architecture will not reduce the hardware complexity as the case of (6,10) filter because the lifting steps can not be all linear. Thus, the lifting-based architecture is not included in this comparison.

For this cell library, we choose the pipelining points for the proposed architectures as shown in Fig. 6 to minimize the critical path. According to Table 1, the proposed architectures could require fewer gate counts under the same timing constraints. Although the proposed architectures need more adders, most of the adders are not on the critical path such that they can be implemented with smaller area and lower speed adders.

6. SUMMARY

In this paper, we present efficient IDWT architectures based on the B-spline factorization. The direct implementation of the B-spline part is also proposed. Because the distributed part is usually designed as small as possible, the proposed

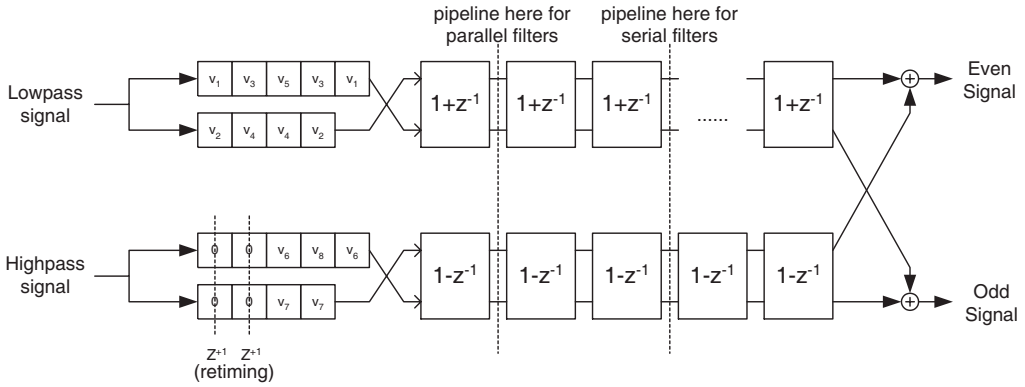


Fig. 6. Proposed architecture for (10,18) filter

Table 1. Comparisons for DWT architectures of the (10,18) filter

Architecture	Multiplier	Adder	Critical Path	Register	Timing (ns)	Comb. Gate Count	Non-comb. Gate Count
Convolution Arc. I	26	26	Tm+4Ta	14	12.5	50986.1	2182.0
Convolution Arc. II	14	26	Tm+4Ta	24	12.2	32776.6	2892.0
Bspline-serial + retiming	8	40	Tm+11Ta	24	19.5	19489.1	2630.7
Bspline-serial + retiming + pipe.	8	40	~(Tm+11Ta)/2	28	12.3	21961.0	3112.0
Bspline-parallel + retiming	8	40	Tm+13Ta	20	22.5	21121.6	2409.0
Bspline-parallel + retiming + pipe.	8	40	~(Tm+13Ta)/2	24	13.35	21952.7	2851.7

B-spline factorized architectures would require fewer multipliers than the convolution-based and lifting-based ones. Although more adders are required, many adders can be implemented with small area and low speed adders because most of them are not on the critical path. From the case study of the (10,18) filter, the efficiency of the proposed architecture is demonstrated.

7. REFERENCES

[1] D. Taubman, "Successive refinement of video: fundamental issues, past efforts and new directions," in *International Symposium on Visual Communications and Image Processing*, 2003.

[2] C. Chakrabarti, M. Vishwanath, and R. M. Owens, "Architectures for wavelet transforms: A survey," *Journal of VLSI Signal Processing*, vol. 14, pp. 171–192, 1996.

[3] K. Andra, C. Chakrabarti, and T. Acharya, "A VLSI architecture for lifting-based forward and inverse wavelet transform," *IEEE Transactions on Signal Processing*, vol. 50, no. 4, pp. 966–977, Apr. 2002.

[4] W. Sweldens, "The lifting scheme: a custom-design construction of biorthogonal wavelets," *Applied and Computaional Harmonic Analysis*, vol. 3, no. 15, pp. 186–200, 1996.

[5] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *The Journal of Fourier Analysis and Applications*, vol. 4, pp. 247–269, 1998.

[6] C.-T. Huang, P.-C. Tseng, and L.-G. Chen, "VLSI architecture for discrete wavelet transform based on B-spline factorization," in *IEEE Workshop on Signal Processing Systems*, 2003, pp. 346–350.

[7] M. Unser and T. Blu, "Wavelet theory demystified," *IEEE Transactions on Signal Processing*, vol. 51, no. 2, pp. 470–483, Feb. 2003.

[8] P. P. Vaidyanathan, *Multirate systems and filter banks*, Prentice Hall, 1993.

[9] M. J. Tsai, J. D. Villasenor, and F. Chen, "Stack-run image coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 5, pp. 519–521, Oct. 1996.

[10] N. Polyak and W. A. Pearlman, "A new flexible biorthogonal filter design for multiresolution filterbanks with application to image compression," *IEEE Transactions on Signal Processing*, vol. 48, no. 8, pp. 2279–2288, Aug. 2000.