Design and Analysis of Efficient Reconfigurable Wavelet Filters

Amit Pande and Joseph Zambreno
Department of Electrical and Computer Engineering
Iowa State University
Ames, IA 50011
Email: {*amit, zambreno*}*@iastate.edu*

***Abstract*—Real-time image and multimedia processing applications such as video surveillance and telemedicine can have dynamic requirements of system latency, throughput, and power consumption. In this paper we discuss the design of reconfigurable wavelet filters for image processing applications that can meet such dynamic requirements. We generate several efficient hardware designs based on a derived family of bi-orthogonal 9/7 filters. An efficient folded and multiplier-free implementation of a 9/7 filter is obtained with the help of nine adders, which is a significant improvement over other existing approaches. We also propose an architecture that allows for on-the-fly switching between 9/7 and 5/3 filter structures. A performance comparison of these filters and their hardware requirements with other existing approaches demonstrates the suitability of our choice.**

I. Introduction

The Discrete Wavelet Transform (DWT) has emerged as a powerful tool for compression and is being used in many multimedia and signal processing applications. The large computational complexity and memory requirements involved in real-time image processing algorithms have been a bottleneck for such systems. Hardware implementations such as those using ASICs or FPGAs are capable of accelerating these computations by exploiting the inherent algorithmic parallelism. DWT has been widely used for image and video coding, and many hardware implementations have been proposed in the research literature [1], [2], [3], [4], [5], [6], [7]. These implementations aim at reducing hardware complexity in order to improve system throughput.

The still image codec JPEG2000 [8], video coding techniques including motion JPEG2000, CEZW [9], CSPIHT [10], and EBCOT [11] use the DWT for its interesting properties such as multi-resolution analysis and time-frequency domain representation. DWT is attractive for multimedia compression because of a) its perfect reconstruction property of the analysis and synthesis wavelets, b) the absence of perceptual degradation at block boundaries, and c) high compression ratios.

The DWT of an image consists of multiple levels of decomposition into low and high pass bands. In Fig. 1, the low pass and high pass filters are labeled $h_0(n)$ and $h_1(n)$ respectively. A set of these filters is used twice (across all rows and columns) to obtain four frequency sub-bands. Thus, applying a 2-D DWT transforms a $M \times N$ image into four $M/2 \times N/2$ images - three detailed images along the horizontal (LH), vertical (HL), and diagonal (HH), and one coarse approximation (LL) of the original image. LL represents the low frequency component while LH, HL, and HH represent the high frequency components of the decomposed signal. Two common approaches for achieving wavelet decomposition are spectral factorization in the frequency domain and the lifting scheme. As will be explained in Section III, we obtain an efficient hardware implementation for a family of parameterized filters by using a spectral factorization-based approach and adding one degree of freedom in the LHBF equation.

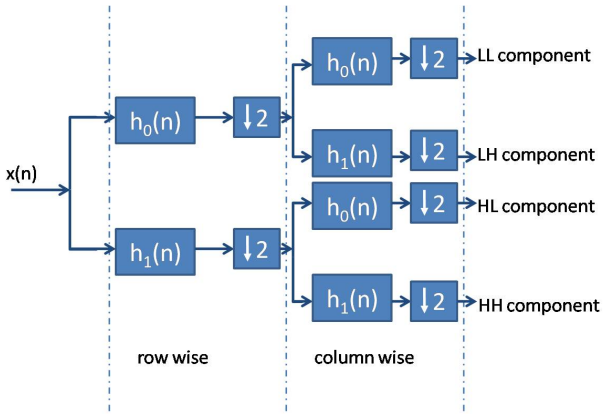


Fig. 1. Basic stages of a one level 2-D wavelet transform operation

Figure 2 explains the working of a reconfigurable wavelet filter for multimedia applications. Many typical applications using DWT have dynamic resource requirements. For example, a video surveillance system [12] may require a low-performance, low-power mode during ordinary operation, while during some important event a higher-performance mode would be needed. During intervals of time such a system would require transmission of a higher quality signal over a potentially sparse resource network. Thus, scalability is a desired feature of an efficient implementation. Hardware reconfiguration of the filters can enable a scalable and power-efficient implementation, by intelligently distributing the hardware resources available.

The use of reconfigurable hardware also allows for on-the-fly switching between different filter architectures, promising an adaptive solution. Variable instances of DWT kernels and various types of filters can be used in the implementation of

this module. The DWT kernel can be implemented using varying lengths, leading to varying image compression properties of the DWT block. As shown in Fig. 2, other requirements for multimedia processing systems include high throughputs for real-time processing and an efficient scheme to ensure high quality reconstruction of the compressed data. Our proposed approach can provide a set of solutions for the dynamic requirements of system performance and power consumption.

The contributions of this paper can be summarized as follows:

- We discuss the development of a family of parameterized biorthogonal 9/7 filters and the deriviation of binary coefficient filters.
- We obtain an efficient multiplier-free implementation of these filters.
- We describe the implementation of a power-efficient switching scheme between 5/3 and 9/7 wavelet filters.
- We obtain a folded 9/7 filter architecture suitable for efficient implementation on reconfigurable platforms.

The remainder of this paper is organized as follows. Section II discusses the motivation behind this work and existing work in this direction. Section III discusses the mathematical background to arrrive at binary filter coefficients. A parameterized design for 9/7 filters is also developed in this section. In Section IV, we arrive at simple hardware structures and present the simulation results. In Section V, we conclude the paper with a look towards planned future work.

II. Motivation and Insight

Biorthogonal Wavelet Filter Banks (BWFBs) are commonly used for image processing. Due to irrational values in their filter coefficients, any associated hardware implementation requires a high precision architecture. Many research works have faced the problem of reducing the DWT complexity [1], [4], [6]. The most common image processing wavelet filter is CDF-9/7, which is accepted as the standard for lossless compression in JPEG2000 [8]. The implementation of CDF-9/7 over fixed point arithmetic leads to quantization error and requires large amounts of hardware resources for perfect image reconstruction. Recently, multiplier-free implementations have been proposed to reduce the computational complexity [6], [13]. Most implementations approximate the irrational coefficients into nearby rational values, leading to image reconstruction quality tradeoffs.

Binary coefficient values for low and high pass filters can lead to a more accurate implementation using finite-precision arithmetic. In this paper, we develop the theory of biorthogonal 9/7 filters in order to obtain perfect reconstruction multiplier-free implementations of the DWT. Previous work has shown that this can help in achieving a simpler and faster implementation, by requiring only nineteen adders [13]. In this paper we derive a new expression for wavelet filter coefficients to obtain all binary rational coefficients. This reduces the number of adders required by our implementation significantly and achieves a significantly better image reconstruction results as compared to the original 9/7 filter. The condition of perfect reconstruction is also satisfied by our derived filters.

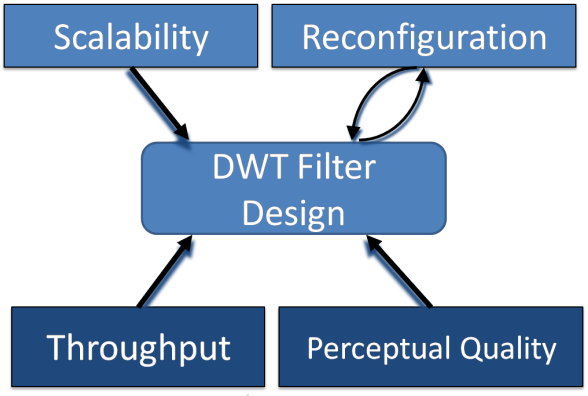


Fig. 2. Conceptual overview of the DWT filter design constraints and desired features

In [14], the authors present a technique to rationalize the coefficients of wavelet filters that will preserve biorthogonality and perfect reconstruction. They develop a parameterized model to achieve the desired biorthogonal 9/7 filters. This approach also preserves regularity of the structure by preserving most of the zeros at $z = -1$. This approach has been exploited to obtain the various binary coefficient multiplier-free implementations of the 9/7 filters in this paper.

III. Parameterized 9/7-tap Biorthogonal Filter

A. The CDF-9/7 Filter

We first discuss the construction of famous Cohen-Daubechies-Feauveau (CDF) 9/7 filter [15]. CDF-9/7 filters are constructed directly based on the spectral factorization method. These wavelets have symmetric scaling and wavelet functions, i.e., both the low pass and high pass filters are symmetric.

There are four filters that comprise the two-channel biorthogonal wavelet system. Two filters are required for transform (analysis), and two for inverse transform (synthesis). The analysis and synthesis low pass filters are denoted by H_0 and G_0, respectively. The analysis and synthesis high pass filters are denoted by H_1 and G_1, respectively, and are obtained by quadrature mirroring the low pass filters:

$$H_1(z) = z^{-1}G_0(-z),\ G_1(z) = zH_0(-z). \tag{1}$$

If we define $D(z) = G_0(z)H_0(z)$ the Perfect Reconstruction (PR) condition simplifies to the following:

$$D(z) + D(-z) = 2. \tag{2}$$

This equation is solved using Lagrange Half Band Filters (LHBF), $L_K(z)$ where:

$$D(z) = L_K(z) = z^K \left(\frac{1+z^{-1}}{2}\right)^{2K} \alpha(k) \tag{3}$$

and

TABLE I
ANALYSIS LOW PASS FILTER (H_0) COEFFICIENTS FOR THE BIORTHOGONAL 9/7 TAP FILTER

$i\backslash\alpha$	1.6848	-1.667	$--1.8$	2
±4	0.026748757411	$1/32$	$1/32$	$1/64$
±3	-0.016864118443	$-1/32$	0	0
±2	-0.078223266529	$-1/16$	$-3/32$	$-1/8$
±1	0.266864118443	$9/32$	$1/4$	$1/4$
0	0.602949018236	$19/32$	$5/8$	$23/32$

TABLE II
ANALYSIS HIGH PASS FILTER COEFFICIENTS (H_1) FOR THE BIORTHOGONAL 9/7 TAP FILTER

$i\backslash\alpha$	1.6848	-1.667	-1.8	-2
±3	0.091271763114	$1/16$	$1/16$	$1/16$
±2	-0.057543526229	$-1/16$	$-1/16$	0
±1	-0.591271763114	$-9/16$	$-9/16$	$-9/16$
0	1.11508705	$9/8$	$9/8$	1

$$\alpha(k) = \sum_{n=0}^{K-1} \left({}^{K+n-1}C_n\right) \left(\frac{2-(z+z^{-1})}{4}\right)^n \tag{4}$$

is simplified for $K = 4$ to get the filter coefficients.

B. Parameterized Filter Design

Table I and II (with label $\alpha = -1.6848$) provide the filter coefficients of the CDF-9/7 filter. A look at these tables explain the inherent difficulties in the hardware implementation of the original CDF-9/7 filter. While this filter has high compression performance, it will lead to lossy compression due to truncation involved in the irrational filter coefficients. Moreover, hardware multipliers would be needed to implement this in any design with reasonable precision. This problem is alleviated by a parameterized design. Tay et al. [14] pose the condition on $D(z)$ to derive the binary rational coefficients to achieve new sets of 9/7 filters by adding more degrees of freedom to the original LHBF equation (by introducing a free parameter α):

$$\begin{aligned} H_0(Z) &= K_h(Z+1)(Z^3+AZ^2+VZ+C) && (5)\\ G_0(Z) &= K_g(Z+1)^2(Z+\alpha) && (6)\\ D(z) &= K_hK_g(Z+1)^3(Z+\alpha) \\ &\quad \times(Z^3+AZ^2+BZ+C). && (7) \end{aligned}$$

The PR condition on $D(Z)$ gives simultaneous constraint equations which simplify to give solutions for A, B, and C (and simultaneously for the filter coefficients) in terms of α:

$$A = -(3+\alpha) \tag{8}$$

$$B = \frac{9\alpha^3+35\alpha^2+48\alpha+24}{3\alpha^2+9\alpha+8} \tag{9}$$

$$C = \frac{8(1+\alpha)^3}{3\alpha^2+9\alpha+8}. \tag{10}$$

Here, we have $Z = (z+z^{-1})/2$. Setting α to -1.6848 gives back the original 9/7 filter pair.

C. Numerical Study

The parameter α can be varied to achieve a family of biorthogonal filter pairs for DWT implementation. Setting $\alpha = -1.6848$ gives us the CDF-9/7 filter which have been proven to have good compression performance. Next, we perform a numerical study to explore a set of binary coefficients filter which is in close proximity to the CDF-9/7 filter. A MATLAB simulation was performed to obtain the quantization error for the filter coefficients with α varying from -1.5 to -2 (in vicinity of the $\alpha = -1.6848$ value). The results are presented in Fig. 3.

It can be observed that the minimization of this error occurs at $\alpha = -2$, where quantization error drops down to 0. We can also observe local minima of curves around two regions in the vicinity of $\alpha = -1.6848$ (at $\alpha = -1.66$ and $\alpha = -1.8$ approximately). We also derive approximate filter coefficients from these minimas to obtain binary coefficients 9/7 filter. These filter coefficients are reported in Tables I and II.

D. Le Gall's 5/3 Filter

Le Gall et al. [16] solved the PR condition by substituting $D(z) = a_0 + a_2z^{-2} + a_3z^{-3} + a_2z^{-4} + a_0z^{-6}$ with the condition $a_0 \in [-\frac{1}{8}, 0]$. For $a = \frac{1}{16}$ the simplification leads to the famous Le Gall's 5/3 filter pair. The coefficients for this filter are given in Table III. This filter is another common filter used for lossless compression and is accepted in JPEG2000 standards [8]. This filter has lower latency than the ones studied earlier but provides lesser image compression capabilities.

IV. HARDWARE ARCHITECTURES

The following optimization steps were performed before attempting a hardware implementation:

- Observe in Tables I, II, and III that the coefficients of $x(i \pm k)$ are the same. Thus they can be grouped together to reduce the hardware complexity.

$$w_0 = x(0), \tag{11}$$

$$w_1 = x(i-1) + x(i+1), \tag{12}$$

$$w_2 = x(i-2) + x(i+2, \tag{13}$$

$$w_3 = x(i-3) + x(i+3). \tag{14}$$

- Division by binary coefficients (e.g. $1/64, 1/16, 1/4$) was performed using arithmetic shift operations.
- The input stream was pipelined. Thus, as shown in Fig. 4 our architecture takes one pixel (or channel input) as the input and outputs the low and high pass signal coefficients with a finite latency.

As can be seen in Fig. 4, our Le Gall's 5/3 filter implementation requires only eight adder/subtractor units. Our 9/7

TABLE III
COEFFICIENTS FOR LE GALL'S 5/3 FILTER

i	$h_0(i)$	$h_1(i)$
± 2	$-1/8$	0
± 1	2/8	$-1/2$
0	6/8	1

filter implementations for $\alpha = -1.67$ required 19 adders and 14 shift registers. For $\alpha = -1.8$, our design requires 17 adder/subtractor units and 13 shift registers. But we observe that the design for $\alpha = -2$ requires only 12 adder/subtractors units and only 10 shift registers.

As described in Fig. 2, the reconfigurable implementation must allow dynamic switching between wavelet filters. Our implementation allows for easy enabling and disabling of the extra hardware to obtain the choice between a more power-efficient binary 5/3 filter versus a more compression-efficient 9/7 filter. In the remainder of this section we describe an architecture to allow for this dynamic switching.

Let us consider an input signal $x(i)$. The low and high pass outputs of this filter ($low(i)$ and $high(i)$ respectively) are obtained by convolution of $x(i)$ with $h_0(i)$ and $h_1(i)$ respectively:

$$low(i) = \sum_{k=-4}^{k=4} h_0(k) \cdot x(i-k), \tag{15}$$

$$high(i) = \sum_{k=-3}^{k=3} h_1(k) \cdot x(i-k). \tag{16}$$

Substituting the values of filter coefficients from Table I, II, and III, we can factorize our 9/7 filter coefficients in terms of 5/3 filter outputs. The subscripts A, B, and C are used to denote the filters obtained with $\alpha = -1.67$, -1.8, and -2, respectively:

$$\begin{aligned} low_A(i) =& 1/2 \times low_{53}(i) - (1/4 + 1/16) \times high_{53}(i) \\ &+ (1/2 + 1/32) \times w_0 + 1/32 \times (w_4 - w_3) \end{aligned} \tag{17}$$

$$\begin{aligned} high_A(i) =& 1/2 \times low_{53}(i) + (1/2 + 1/4) \times high_{53}(i) \\ &- 1/4 + 1/16) \times w_1 + 1/16 \times (w_3) \end{aligned} \tag{18}$$

$$\begin{aligned} low_B(i) =& 1/2 \times low_{53}(i) + 1/4 \times high_{53}(i) \\ &+ 1/32 \times (w_4 - w_3) \end{aligned} \tag{19}$$

$$\begin{aligned} high_B(i) =& 1/2 \times low_{53}(i) + (1/2 + 1/4) \times high_{53}(i) \\ &- (1/4 + 1/16) \times w_1 + 1/16 \times (w_3) \end{aligned} \tag{20}$$

$$low_C(i) = low_{53}(i) - 1/32 \times w_0 + 1/64 \times w_4 \tag{21}$$

$$\begin{aligned} high_C(i) =& 1/2 \times high_{53}(i) - 1/32 \times w_1 \\ &+ 1/32 \times w_3. \end{aligned} \tag{22}$$

Figure 4 (a-c) provides the implementation details of these architectures. The dark (yellow) region is the hardware required for the implementation of Le Gall's 5/3 filter. The filters parameters need not be updated. Only the unshaded parts of the architecture would to be switched off to implement a low power 5/3 filter. $Low_A(i)$ and $high_A(i)$, or $low_{53}(i)$ and $high_{53}(i)$ can be selected as the output depending upon the mode of operation. When operating in 5/3 filter mode only the yellow shaded region of the architecture would be used thus reducing considerably the power consumption of the system. This figure shows the conceptual design and architecture and doesn't include the pipeline stages of these structures. A folded architecture can be developed for the $\alpha = -2$ case where the low and high pass output coefficients are dependent only on low and high pass values respectively of 5/3 filter. This is presented in Fig. 4(d).

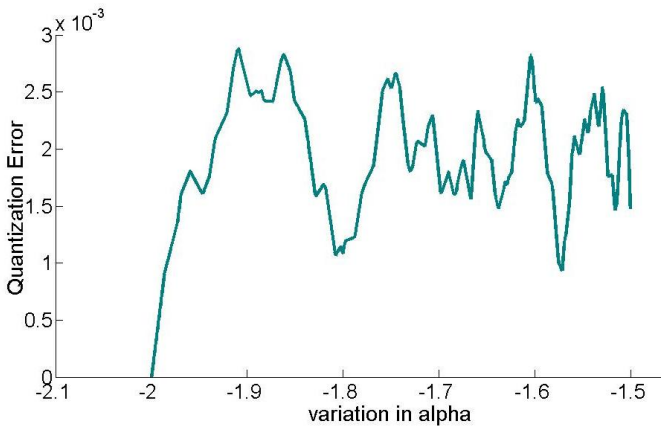


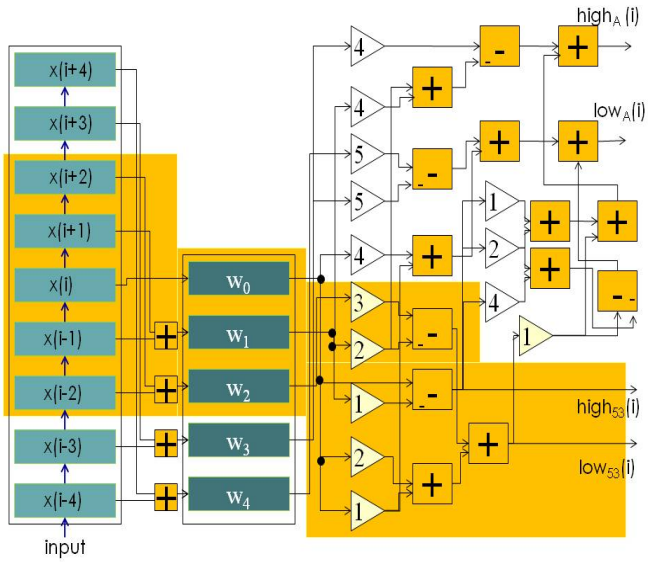
Fig. 3. Numerical Analysis of Quantization error for seven bit finite representation of filter coefficients

V. EXPERIMENTAL RESULTS

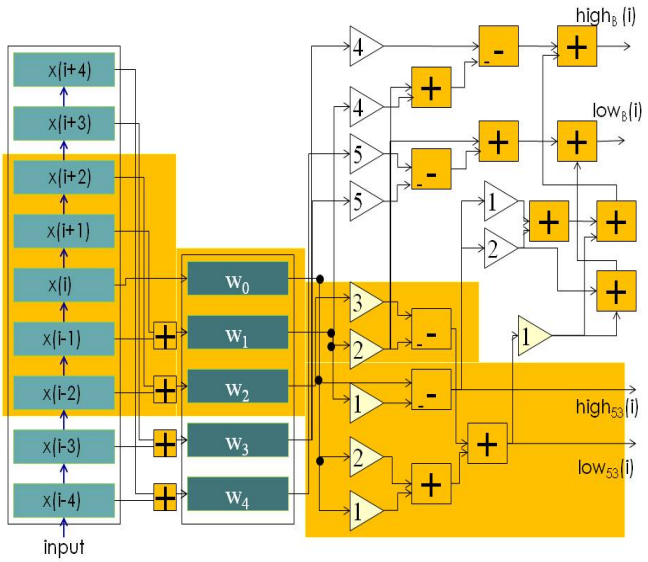
We targeted a Xilinx XC5VLX30 FPGA for our experiments, using ModelSim 6.0c for simulation purposes and Xilinx ISE 9.1i for synthesis. In this section we detail the area and performance results.

Direct implementation of the CDF-9/7 filter gave a clock frequency of 107 MHz, while requiring 16 multiplier units. Martina et al. [13] report a clock frequency of about 200 MHz through a multiplier-free implementation, targeting 0.13 μm VLSI technology. Table IV summarizes the performance of our Xilinx Virtex-V implementation, and compares our results with other recent works. All the parameterized binary implementations outperform the existing implementations in terms of number of required adders and clock frequency.

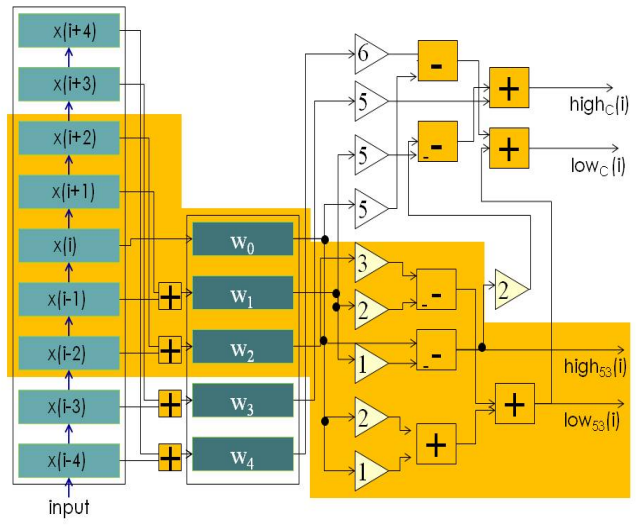
Our initial non-pipelined design obtained a clock frequency of about 108 MHz, due to its long critical path. The critical path of the circuit lies from the w_i registers to the final output $low_C(i)$ or $high_C(i)$, passing through signals $low_C(i)$ or $high_C(i)$. We then pipelined this computation into several stages and obtained a faster implementation. The $\alpha = -2$ architecture showed a clock frequency of about 390 MHz. This design requires less FPGA resources (registers and LUTs) than the $\alpha = -1.67$ and $\alpha = -1.8$ architectures. The folded architecture variant for $\alpha = -2$ was also implemented, resulting in a faster clock frequency and less adders (leading



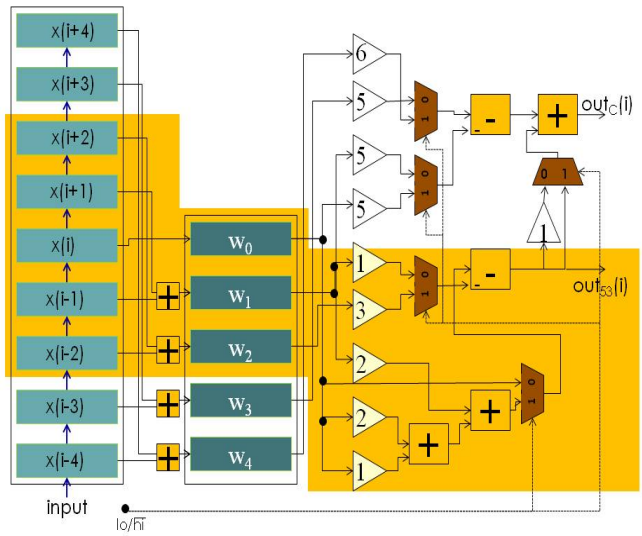
(a) $\alpha = -1.67$



(b) $\alpha = -1.8$



(c) $\alpha = -2$



(d) $\alpha = -2$ (folded)

Fig. 4. Hardware architectures for biorthogonal 9/7 filter

to fewer logic slices). The design of binary coefficients filter also helped us to achieve perfect reconstruction of image signals. In general, pipelining of the datapath resulted in an increased number of slice utilization in our designs. The $\alpha = -2$ architecture, however, reaches almost the same bit slice utilization in our FPGA implementation as the initial CDF 9/7 filter. The PSNR performance of the reconstructed images is satisfactory for $\alpha = -2$ as reported in the table.

The usage of simple adders by our designs free the MACs and multipliers present in most modern FPGAs for other image processing modules in an FPGA-based multimedia system. The folded design for $\alpha = -2$ requires the least number of adders, slice registers, and LUTs.

This proposed architecture can run (over line-based DWT architectures) at 390 MHz, enabling it to process High Definition Video frames (1440×1080) in an estimated 5 ms time. As previously mentioned, the shaded (yellow) regions in Fig. 4 show the baseline 5/3 filter implementation. Thus the architecture can be optimized to switch on-the-fly to 5/3 mode in order to save power.

TABLE IV
COMPARISON OF BINARY FILTER FEATURES AND HARDWARE RESOURCES REQUIREMENTS

Features	**CDF-9/7**	$\alpha = -2$ folded	$\alpha = -2$	$\alpha = -1.80$	$\alpha = -1.67$	**[17]**	**[5]**	**[18]**	**[13]**	**[6]**
Adders	15	9	12	17	19	19	15	8	19	21
Multipliers	16	0	0	0	0	0	0	4	0	0
PSNR grade	A	B	B	A	A	C	C	A	B	B
Reconfigurable?	N	Y	Y	Y	Y	N	N	N	Y	N
Slices (Registers)	144	208	213	253	294	-	-	-	-	-
Slices (LUTs)	80	175	194	217	289	-	-	-	-	-
No. of bit Slices	210	245	259	311	375	-	-	-	-	-
Clock Frequency (MHz)	106.98	389.03	317.31	310.56	310.22	-	-	-	200	-

VI. CONCLUSIONS AND FUTURE WORK

This paper introduces a family of parameterized bi-orthogonal wavelet filters for efficient implementation over fixed point hardware. Results show that the folded $\alpha = -2$ filter gives a simple yet efficient design. We also presented a multiplier-free binary coefficient filter design. This architecture allows for on-the-fly switching between the filter structures to meet dynamic application requirements.

The folded implementation requires only nine adders which is a significant improvement over existing works. Performance comparison and simulation results over Xilinx Virtex-V FPGA demonstrate the promise of a robust reconfigurable design.

For brevity, we have discussed only the design and implementation of the filters only for this paper. A more descriptive design involving layout issues and other constraints for reconfigurable architectures will be included in future work.

REFERENCES

[1] M. Alam, C. Rahman, W. Badawy, and G. Jullien, "Efficient distributed arithmetic based dwt architecture for multimedia applications," in *Proceedings of the Intl. Workshop on SoC for Real Time Applications*, 2003, pp. 333–336.

[2] A. Benkrid, K. Benkrid, and D. Crookes, "Design and implementation of a generic 2D orthogonal discrete wavelet transform on FPGA," in *Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM)*, Apr. 2003, pp. 162–172.

[3] A. Benkrid, D. Crookes, and K. Benkrid, "Design and implementation of a generic 2D biorthogonal discrete wavelet transform on an FPGA," in *Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 2001, pp. 190–198.

[4] J. Ritter and P. Molitor, "A pipelined architecture for partitioned dwt based lossy image compression using FPGAs," in *Proceedings of the Intl. symposium on Field Programmable Gate Arrays (FPGA)*, 2001, pp. 201–206.

[5] K. Kotteri, S. Barua, A. Bell, and J. Carletta, "A comparison of hardware implementations of the biorthogonal 9/7 DWT: convolution versus lifting," *IEEE Transactions on Circuits and Systems II*, vol. 52, no. 5, pp. 256–260, May 2005.

[6] M. Martina and G. Masera, "Low-complexity, efficient 9/7 wavelet filters implementation," in *Proceedings of the Intl. Conference on Image Processing (ICIP)*, Sep. 2005.

[7] J. Villasenor, B. Belzer, and J. Liao, "Wavelet filter evaluation for image compression," *IEEE Transactions on Image Processing*, vol. 4, no. 8, pp. 1053–1060, Aug. 1995.

[8] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG2000 still image coding system: an overview," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 1103–1127, Nov 2000.

[9] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3445–3462, Dec. 1993.

[10] A. Said and W. Pearlman, "An image multiresolution representation for lossless and lossy image compression," *IEEE Transactions on Image Processing*, vol. 5, pp. 1303–1310, 1996.

[11] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1158–1170, Jul. 2000.

[12] P. Kumar, A. Mittal, and P. Kumar, "Fusion of thermal infrared and visible spectrum video for robust surveillance," in *ICVGIP*, 2006, pp. 528–539.

[13] M. Martina and G. Masera, "Multiplierless, folded 9/7 5/3 wavelet VLSI architecture," *IEEE Transactions on Circuits and Systems II*, vol. 54, no. 9, pp. 770–774, Sep. 2007.

[14] D. Tay, "Rationalizing the coefficients of popular biorthogonal wavelet filters," *IEEE Transactions on Circuits and Systems for Video Technology,*, vol. 10, no. 6, pp. 998–1005, Sep. 2000.

[15] D. I. Cohen, A. and J. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Commun. Pure Appl. Math.*, vol. 45, pp. 485–560, 1992.

[16] D. L. Gall and A. Tabatabai, "Sub-band coding of digital images using symmetric short kernel filters and arithmetic coding techniques," in *Proceedings of the Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1988, pp. 761–764.

[17] D. Tay, "A class of lifting based integer wavelet transform," in *Proceedings of the Intl. Conference on Image Processing (ICIP)*, 2001, pp. 602–605.

[18] C. Huang, P. Tseng, and L. Chen, "Flipping structure: an efficient VLSI architecture for lifting-based discrete wavelet transform," *IEEE Transactions on Signal Processing*, vol. 52, no. 4, pp. 1080–1089, Apr. 2004.