Novel Polymorphic Reconfigurable Hardware Support for Discrete Wavelet Transform

Amit Pande and Joseph Zambreno
Department of Electrical and Computer Engineering
Iowa State University, Ames, IA -50011, USA
Email: amit@iastate.edu, Telephone: (515) 294–8343, Fax: (515) 294–2667

***Abstract*—Real-time multimedia processing empowers many sensitive multimedia services including video surveillance, telemedicine, e-learning etc. While several hardware architectures have been proposed in the research literature to support these primitives, they fail to address applications whose performance and resource requirements have a dynamic aspect. We introduce a Polymorphic Wavelet Architecture (Poly-DWT) as a crucial building block that can potentially make dynamic resource allocation decisions including power requirements, internal bit representation, number of processing kernels, according to application requirements. We introduce a folded switching filter architecture. It allows for dynamic switching between 5/3 and 9/7 filters and leads to power-efficient design. Further, a multiplier-free design with few adder requirements demonstrates the potential of Poly-DWT for embedded systems. Through an FPGA prototype, we perform a quantitative analysis of our Poly-DWT architecture. Expermiental results and comparisons with systems proposed in existing literature demonstrates the superior performance of our approach in static and dynamic aspects.**

I. Introduction

DWT has emerged as the popular transform step in many recent image and multimedia processing applications [1], [2], [3], [4]. While many hardware implementations (both VLSI and FPGA based designs) of DWT exist in literature [5], [6], they fail to address the dynamic requirements of real-time multimedia processing applications like traffic management systems, security cameras, tele-medicine over scarce resource networks, distant education classes etc.

In this paper, we introduce a new layout and reconfiguration scheme for multimedia applications, which we call the Polymorphic Wavelet Architecture (Poly-DWT). We define *polymorphism* as the capacity of an architecture to adapt its hardware usage to meet the desired dynamic specifications. In the image processing domain, these specifications would be in terms of throughput, reconstruction quality, and power consumption, among others. Our Poly-DWT architecture allows the individual processing kernels to modify their hardware resources to suit the instantaneous application requirements. At its highest level, the Poly-DWT provisions for optimal device usage under the given performance and quality requirements. It allows for a fine granularity of run-time reconfiguration even when implemented over commodity FPGA platforms. More details of implementation are given in [7], [8].

II. Motivation

The requirements of embedded system design for real-time DWT-based image processing are as follows:

- Modern embedded multimedia systems would require transmission of a higher quality signal over a potentially sparse resource network. Thus, good compression is a desired feature of an efficient implementation.
- High system throughput and good perceptual quality are desired features and impose constraints on system design.
- Embedded systems have hardware and power constraints because they are typically mobile battery-driven devices.
- Hardware reconfiguration of the filters is the enabling technology to realize these tradeoffs. Intelligent allocation of hardware resources can achieve a run-time tradeoff between hardware resources and performance constraints.

The contributions of this paper can be summarized as follows:

- We introduce the concept of the Polymorphic Discrete Wavelet Transform (Poly-DWT) architecture. The Poly-DWT architecture enables dynamic reconfiguration of hardware resources to efficiently create a dynamic response to changing external conditions.
- We discuss the development of a family of parameterized bi-orthogonal 9/7 filters and the derivation of binary coefficient filters for hardware-efficient implementation.
- A multiplier-free binary 9/7 wavelet filter is introduced to obtain a faster and more efficient implementation.
- A switching scheme to allow runtime flipping between 5/3 and 9/7 wavelet structures with hardware reuse is presented.
- We present a quantitative analysis of the various factors and tradeoffs involved in a Poly-DWT implementation.
- We present a detailed area/performance tradeoff analysis for the sample Poly-DWT filters.

III. Insight

This work proposes a reconfigurable design for DWT (Poly-DWT). The two most common DWT filters used in image compression are Le Gall's 5/3 filter and the Daubechies 9/7 filter [3]. The Le Gall's filter has rational coefficients and its hardware implementation requires less resources. The Daubechies 9/7 (also commonly known as CDF 9/7) filter has better compression performance. However, it has irrational

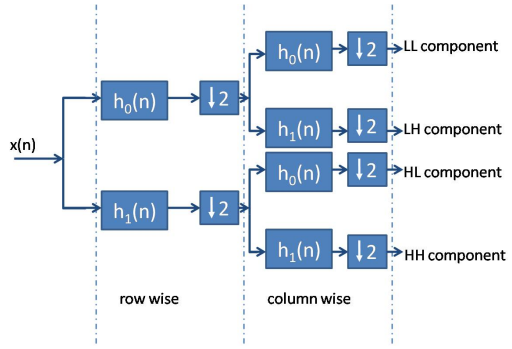

Fig. 1. Basic stages of a one level 2-D wavelet transform operation

coefficients therefore its hardware requirements are very large. A binary-coefficients 9/7 filter is first implemented to allow cheaper implementation cost, higher throughput and on the fly switching to 5/3 filter architecture. The design is multiplier-free and constructed over 5/3 filter enabling on-the-fly switching between 9/7 and 5/3 filter structures. A folded design is then derived to further reduce hardware requirements by utilizing redundancy in DWT computations.

IV. Development

There are four filters that comprise the two-channel bi-orthogonal wavelet system. 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 1.

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

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) to get derivation of Daubechies 9/7 filter (see [9]). While this filter has high compression performance, it will lead to lossy compression due to truncation involved in filter coefficients in a fixed point hardware representation. On the other hand floating point implementation implies a higher hardware cost. Moreover hardware multipliers would be needed to implement this in our design with reasonable precision.

A parameterized filter design allows us to obtain a family of 9/7 filters. This new design is then searched for rational coefficients to get new filters to alleviate the above mentioned problems.

A. Parameterized Filter Design

A parameterized design alleviates the problem of irrational coefficients. Tay et al. [10] impose this constraint on $D(z)$ to derive the binary rational coefficients and achieve new sets of

TABLE I
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

TABLE II
Analysis low pass filter (H_0) coefficients for the bi-orthogonal 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	$19/32$	$5/8$	$23/32$

9/7 filters by adding more degrees of freedom to the original LHBF equation (by introducing a free parameter α):

$$H_0(Z) = K_h(Z+1)(Z^3 + AZ^2 + VZ + C) \tag{3}$$
$$G_0(Z) = K_g(Z+1)^2(Z+\alpha) \tag{4}$$

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.

B. Numerical Study

The parameter α can be varied to achieve a family of bi-orthogonal 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). We observed local minima of quantization error around two points in the vicinity of $\alpha = -1.6848$ (at $\alpha = -1.66$ and $\alpha = -1.8$ approximately and at $\alpha = -2$ we obtain zero error). We also derive approximate filter coefficients from these minimas to obtain a binary coefficients 9/7 filter. These filter coefficients are reported in Tables I and II. The design with $\alpha = -2$ was selected for implementation because of zero quantization error and low hardware requirements [7].

C. Folded Binary Filter

The filter coefficients for selected design are as follows:

$$\begin{aligned} low(i) =& \frac{23}{32}.x(i) + \frac{1}{4}.(x(i-1) + x(i+1)) \\ & - \frac{1}{8}.(x(i-2) + x(i+2)) + \frac{1}{64}.(x(i-4) + x(i+4)) \end{aligned} \tag{5}$$

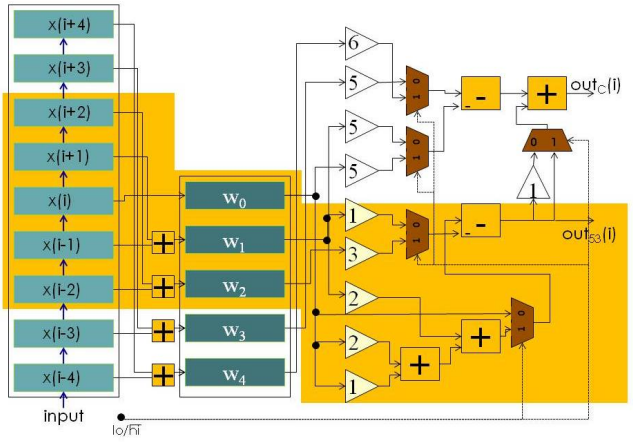

Fig. 2. New hardware architecture for 9/7 filter

$$high(i) = x(i) - \frac{9}{16}.(x(i-1) + x(i+1)) + \frac{1}{16}.(x(i-3) + x(i+3)) \quad (6)$$

This is rewritten in terms of the coefficients of Le Gall's filter as:

$$low(i) = low_{53}(i) - 1/32 \times x(i) + 1/64 \times x(i-4+x(i+4) \quad (7)$$

$$high(i) = 1/2 \times high_{53}(i) - 1/32 \times x(i-1+x(i+1) + 1/32 \times x(i-3+x(i+3) \quad (8)$$

where low_{53} and $high_{53}$ are the high and low pass outputs of Le Gall's filter respectively. The output low and high pass signals are redundant and are downsampled by 2 before any subsequent operation. We exploit this redundancy to design a folded design where we alternate the use of adders and otherusers between alternate cycles. Thus the output switches between low and high in subsequent cycles which can be demultiplexed if required. This feature reduces the number of adders in our circuit to merely 9. The final architecture is as shown in figure 2.

The original Daubechies 9/7 filter has $\alpha = -1.68$. From the available rational coefficients architectures, we select $\alpha = -2$ architecture which leads to DWT implementation in just 12 adders. Further we note that DWT computations give output which are redundant and downsampled by factor of 2 at the end of computation. We utilize this property to build a folded design which performs high and low pass computations in alternate cycles. Thiss design requires merely nine adders in the design.

D. Design Features

We performed several optimization steps to reduce the cost of underlying hardware. The following optimization steps were performed:

- The coefficients of $x(i \pm k), k \in [1, 4]$ are the same. Thus they can be added together to reduce the hardware complexity.

TABLE III
HARDWARE ACCELERATION ON A VIRTEX-5 XC5VLX30 FPGA(TIME IN μS)

Image	Le Gall 5/3 Filter			Daubechies 9/7 Filter			Poly-DWT 9/7 Filter	
	SW	HW	Speedup	SW)	HW	Speedup	HW	Speedup
CIF	1420	197	7.06×	2780	288	9.65×	230	12.01×
Q-CIF	370	68	5.45×	790	91	8.68×	77	10.26×

- Division by rational coefficients (e.g. $1/64, 1/16, 3/4 = 1/2 + 1/4$) was performed using arithmetic shift operations. This eliminates the need of multipliers in the circuits.
- The input stream was pipelined. Thus, as shown in Fig. 2 our architecture takes one pixel (or channel input) as the input and outputs the low and high pass signal coefficients with a finite latency. This help us to achieve a good throughput and a higher clock frequency.
- Folded design was implemented to further reduce the number of adders to merely 9.
- The yellow (shaded) region of the architecture corresponds to the implementation of Le Gall's filter. Therefore, all the architectures are based on Le Gall's design providing us the feature of 'on the fly' switching to Le Gall's mode of operation (power saving feature).
- We propose other parameters like change in bitwidth of internal registers of the FPGA etc. to add more dimensions of polymorphism.

V. EXPERIMENTS

We evaluate our approach on the Xilinx Virtex-V XC5VlX30 FPGA.The Polymorphic architecture presented in this paper has been analyzed in terms of image reconstruction and kernel area considerations. Table III gives the speedup by using a FPGA based approach over a software implementation on a Dual-core 1.86 GHz CPU. Poly-DWT filter utilizes a simpler hardware and hence reaches more speed up than original Daubechies' filter.

The image compression performance of Poly-DWT is evaluated and reported in Table IV. We obtained the results over DWT-based SPIHT image compression format. The results indicate the better PSNR performance of Poly-DWT over existing works. Results are reported over two bitrates to illustrate the superior performance of Poly-DWT over a larger range of bitrate (size of compressed video).

Table V 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. Some key observations from the table are:

- The proposed filters provide efficient hardware implementations requiring fewer adders and no multipliers in the design.
- The $\alpha = -2$ filter requires least hardware. These designs are reconfigurable and can be switched on-the-fly to 5/3

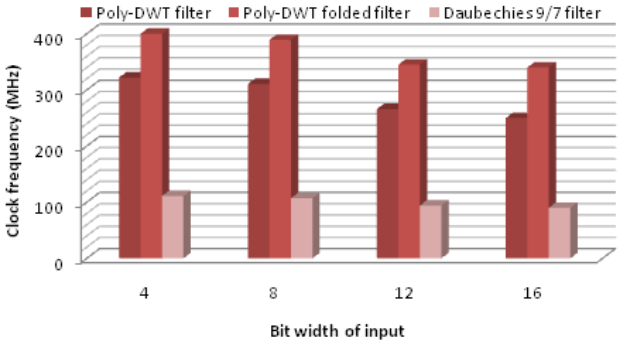

Fig. 3. Change in FPGA clock frequency(MHz) for variable bit widths for various filters

TABLE IV
IMAGE COMPRESSION PERFORMANCE ON SPIHT CODER (PSNR VALUES).

Image	Bitrate=0.5 bpp			Bitrate=2 bpp		
	Daub. 9/7	Poly-DWT	Martina,07	Daub. 9/7	Poly-DWT	Martina,07
lena	28.213	29.46	27.7	38.47	38.17	36.5
surveillance	26.1	28.1	26.54	38.41	42.21	39.21
lecture	34.35	33.8	32.73	48.3	51.25	43.71
helicoptor	33.75	35.7	35.01	48.59	54.72	47.14

filter.

- For $\alpha = -2$ design, the clock frequency of 390 MHz was achieved. Moreover, the new hardware architectures have good PSNR and compression performance.
- Poly-DWT design outperforms existing designs in literature in terms of image compressions performance, hardware usage, design complexity etc while adding the novel dimensions of on-the-fly switching, reconfigurability, folded design etc to the architecture.

Our initial non-pipelined design had a long critical path. The critical path of the circuit lies from the w_i (see figure 2 registers to the final output $low(i)$ or $high(i)$. 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 (see Figure 3). This design requires less FPGA resources (registers and LUTs) than the $\alpha = -1.67$ and $\alpha = -1.8$ architectures. Thus the overall area requirements are less.

Performance comparison of the proposed architecture demonstrates the promises of a fast and efficient image processing over reconfigurable systems. We also performed analysis of bitwidth of internal representation and other reconfigurable design aspects in our work which has not been included for brevity. The details are available at: $www.ece.iastate.edu/\ amit/PolyDWT.html$

VI. RESULTS & FUTURE WORK

This work presents an attempt to build an architecture to provide Polymorphic implementation of DWT for real-time needs. Various aspects of polymorphism including on-the-fly swiching structures, reconfiguration, bit-variation for internal registers etc were studied in this work and a multiplier-free architecture was presented. As a future work, such polymorphic architectures can be developed for more image compression modules.

TABLE V
COMPARISON OF BINARY FILTER FEATURES AND HARDWARE RESOURCES REQUIREMENTS

Features	Daub. 9/7	$\alpha = -2^*$	$\alpha = -2$	$\alpha = -1.8$	$\alpha = -1.67$	[11]	[5]	[12]	[13]	[14]
Adders	15	9	12	17	19	19	15	8	19	21
Multipliers	16	0	0	0	0	0	0	4	0	0
PSNR	A	B	B	A	A	C	C	A	B	B
Reconf.	N	Y	Y	Y	Y	N	N	N	Y	N
Registers	144	208	213	253	294	-	-	-	-	-
LUTs	80	175	194	217	289	-	-	-	-	-
Bit Slices	210	245	259	311	375	-	-	-	-	-
Clock(MHz)	107	389	317	311	310	-	-	-	200	-

* Folded design

REFERENCES

[1] W. Yang, F. Wu, Y. Lu, J. Cai, K. N. Ngan, and S. Li, "Scalable multiview video coding using wavelet," *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, pp. 6078–6081 Vol. 6, May 2005.

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

[3] 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.

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

[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] P. Tseng, Y. Chang, Y. Huang, H. Fang, C. Huang, and L. Chen, "Advances in hardware architectures for image and video coding - a survey," *Proc. IEEE*, vol. 93, no. 1, pp. 184–197, Jan. 2005.

[7] A. Pande and J. Zambreno, "Design and analysis of efficient reconfigurable wavelet filters," in *Proceedings of the IEEE Intl. Conf. on Electro Information Technology (EIT)*, 2008, pp. 337–342.

[8] amit Pande and J. Zambreno, "Polymorphic wavelet architecture over reconfigurable hardware," in *Field Programmable Logic and Applications (FPL)*. IEEE, 2008.

[9] R. Ansari, C. Guillemot, and J. Kaiser, "Wavelet construction using Lagrange halfband filters," *IEEE Trans. Cir. Sys.*, vol. 38, no. 9, pp. 1116–1118, Sept. 1991.

[10] 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, Sept. 2000.

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

[12] 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.

[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, Sept. 2007.

[14] ——, "Low-complexity, efficient 9/7 wavelet filters implementation," in *Proceedings of the Intl. Conference on Image Processing (ICIP)*, Sept. 2005.