A FLEXIBLE DESIGN OF FILTERBANK ARCHITECTURES FOR DISCRETE WAVELET TRANSFORMS

Patrick Longa, Ali Miri and Miodrag Bolic
School of Information Technology and Engineering
University of Ottawa
{plong034, samiri, mbolic}@site.uottawa.ca

ABSTRACT

In this paper, Distributed Arithmetic (DA) has been used to implement a fully parallel LUT-based DA wavelet filterbank with interlaced input registers. In our scheme, decimation has been seamlessly integrated into the filter structure to achieve the same throughput performance as polyphase-based filterbanks. However, because partitioning of the filters is avoided, our scheme gives more flexibility to implement the LUT-DA structure, and consequently, lets designers maximize area utilization on LUT-based FPGAs. Our architecture has been designed for orthonormal and biorthogonal wavelets, and implemented on an Altera Stratix II FPGA. Significant reduction in terms of area requirements and increased throughput performance are achieved when compared to other DWT filterbanks based on DA, convolution or the lifting scheme.

Index Terms - Wavelet transforms, distributed arithmetic, field programmable gate arrays.

1. INTRODUCTION

In the last few years, there has been a growing trend to implement DSP algorithms on Field Programmable Gate Arrays (FPGAs), because FPGAs offer higher level of parallelism than traditional processors. In this sense, Discrete Wavelet Transform (DWT), a relatively new computationally-intensive signal transform that is being increasingly applied to many areas of signal processing, has been explored and several architectures have been proposed to achieve more attractive performance/cost trade-offs on FPGAs.

However, complexity of DWT is still a major challenge for the industry and research community in applications where the traditional Discrete Cosine Transform (DCT) is the preferred tool due to its lower computing requirements. Thus, developing area-efficient and low-power DWT architectures is a key task.

In this effort, we have developed a flexible filterbank architecture for one- and two-dimensional Discrete Wavelet Transforms (1D- and 2D-DWT) based on the very well-known Distributed Arithmetic (DA) technique, which exploits the Look-Up Table (LUT)-based FPGA structure to build a multiplier-less filterbank, the core component in a DWT structure. With the help of a special memory arrangement and an interlaced structure at the input, the downsampling stage is seamlessly realized inside the filterbank structure. With this scheme, our implementation achieves the same throughput performance of regular polyphase filterbanks, getting a resultant sample at every clock cycle. However, the polyphase approach presents some disadvantages when applied on LUT-based devices such as FPGAs, given that in many cases filter partition requires the filterbank be implemented with 2- or 3-input LUTs. This typically achieves suboptimal area utilization. In that sense, our scheme offers superior flexibility, letting designers choose a more appropriate LUT unit (4- or 6-input LUT) according to the targeted device. The result is optimal area utilization and, consequently, reduced area requirements.

The proposed filterbank architecture have been designed using 8-tap and 9/7 Daubechies wavelets in Simulink (DSP Builder) and Altera Quartus II, and implemented on an Altera Stratix II FPGA. To assess performance, simulations with several standard images have been carried out in a 2D-DWT scenario.

2. PREVIOUS WORK

There are two main approaches to implement DWT: convolution-based and the lifting methods. Convolution-based methods are mainly based on Mallat's pyramid algorithm [1], which decomposes the input signal into frequency subbands. This structure can be realized as a filterbank built by means of cascaded Quadrature Mirror Filters (QMF). For a 1D-DWT, it translates to high- and low-pass filter pairs followed by a decimation stage. The basic structure can be cascaded from the low-pass filter to have several levels of transformation (figure 1(a)).

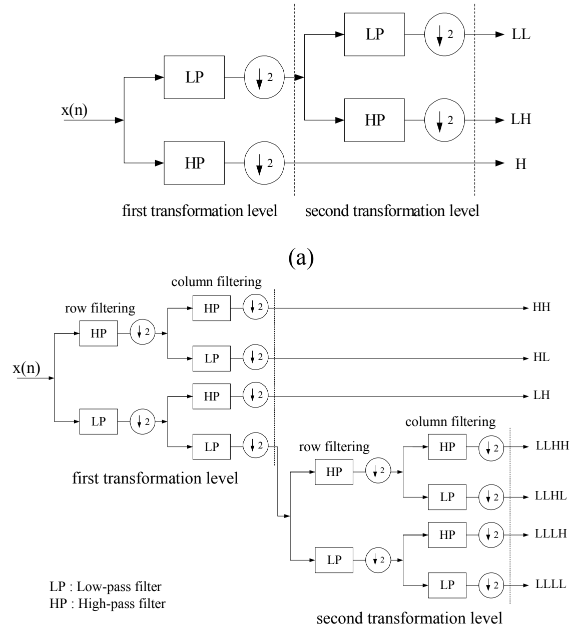
From the previous structure, 2D-DWT with N levels of transformation can be achieved by alternating row and column filtering in each level with iteration from the LL (low-pass/low-pass) subband, as shown in figure 1(b).

Based on these basic structures, several approaches have been proposed to overcome area/bandwidth limitations imposed by direct implementation of the filterbank. Representative implementations are presented by Kotteri [2] and Masud et al. [3].

Sweldens [4] presented an alternative method, known as the lifting scheme, which consists of splitting the signal into two parts and then finding a correlation between both to get rid of redundant operations. Thus, the number of computations is approximately reduced by a factor of 2 in comparison to convolution-based schemes. This is because the polyphase nature of DWT is seamlessly coupled into the lifting scheme.

Several techniques have been applied to improve the two main methods. Among them, exploiting the polyphase nature of wavelets is an optimal way of increasing the frequency and getting twice the throughput in the convolution approach [2]. Another useful technique has been the utilization of multiplier-less filterbanks to avoid expensive multipliers. For instance, Kotteri [2] used Canonic Sign Digit (CSD) representation. Zhou et al. [5] proposed the use of Distributed Arithmetic to build the filterbanks in the 2D-DWT; and Al-Haj [6] presented different versions of DA-based 1D-DWT architectures from serial to fully parallel.

1-4244-0728-1/07/$20.00 ©2007 IEEE



(a)

(b)

Fig. 1. Two-level DWT: a) 1D-DWT, b) 2D-DWT.

In this work, we show that an interlaced arrangement of the input registers can replace the traditional polyphase formulation, adding more flexibility to design the LUT structure in the DA scheme while keeping the same computing effort in terms of number of clock cycles.

3. DA-BASED DWT

As shown in figure 1, DWT can be efficiently implemented by using cascaded QMFs, which are comprised of elementary pairs of high- and low-pass FIR filters followed by a decimation stage.

However, the direct approach of discarding every other sample after filtering to decimate the result is suboptimal. An improved approach integrates the decimation into the filter structure by modifying the input accordingly, as shown in equation (1).

$$y[n] = \sum_{k=0}^{K-1} a_k x[2n-k] \tag{1}$$

Where:
x and y are the input and filtered data, respectively.
a_k is the set of constant filter coefficients.
K is the number of taps of the FIR filter.

The first remark is that equation (1) can be efficiently realized by means of an interlaced arrangement of the input registers, as shown in figure 2. Optimality in terms of throughput is easily achieved, given that two samples are injected simultaneously into the filter structure. From this, there exists a well-known approach, known as polyphase technique [2], which consists in splitting the filter *H(z)* into even and odd coefficients. This technique is useful for reducing area requirements on FPGA architectures for instance, where LUT requirement grows exponentially with the filter order.

We propose an alternative approach from figure 2. We basically avoid the polyphase filter partition by directly replacing the filter *H(z)* with a multiplier-less structure such as DA. We will show later that this simple approach may replace the traditional polyphase methodology, achieving similar throughput but reduced area requirements.

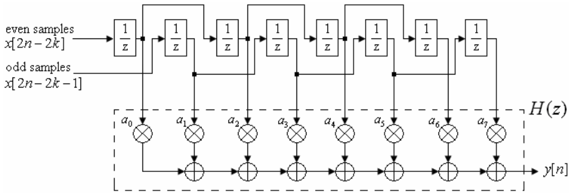


Fig. 2. 8-tap FIR filter with interlaced input registers.

In the remainder of the section, we will detail the mathematical foundations of DWT in the two-dimensional case only. A similar approach for the one-dimensional case easily follows.

Equation (1) can be modified to reflect the cascade structure of the 2D-DWT, and the approximation and detail wavelet coefficient decomposition. The latter simply corresponds to low-pass and high-pass filter outputs in the two-subband decomposition scheme. It is shown in equation (2).

$$c_{n\,(i)} = \sum_{k=0}^{K_1-1} l_k c_{2n-k\,(i-1)} \qquad d_{n\,(i)} = \sum_{k=0}^{K_2-1} h_k c_{2n-k\,(i-1)} \tag{2}$$

Where:
$i = 1,\ldots,2j$ represents the row- or column-wise stage, $\lceil i/2 \rceil$ is the current transformation level, and j is the total number of transformation levels.
l_k and h_k are sets of low-pass and high-pass filter coefficients.
$c_{n(i)}$ is the set of approximation wavelet coefficients at the i^{th} row- or column-wise stage. At the first stage: $c_{n(0)} = x[n]$, where $x[n]$ is the set of input samples of the original signal.
$d_{n(i)}$ is the set of detail wavelet coefficients at the i^{th} row- or column-wise stage.
K_1 and K_2 are the number of taps of the low- and high-pass filters, respectively. For orthonormal wavelets $K_1 = K_2$.

However, a parallel implementation of equation (2) would require (K_1+K_2) multipliers, which results in highly expensive implementations. To alleviate this problem several multiplier-less schemes such as DA have been proposed.

DA appeared as a very efficient solution especially suited for LUT-based FPGA architectures. This technique, first proposed by Croisier et al. [7], is a multiplier-less architecture based on an efficient partition of the function in partial terms using 2's complement binary representation of data. The partial terms can be pre-computed and stored in LUTs. The flexibility of this algorithm on FPGAs permits everything from bit-serial implementations to pipelined or fully parallel versions of the scheme.

In a parallel DA scheme, inputs are normally expressed in 2's complement binary representation with the sign bit to the left of the radix point. Thus, an M-bit input is expressed as follows:

$$x[n] = -b_{n,M-1} 2^{M-1} + \sum_{m=0}^{M-2} b_{n,m} 2^m \tag{3}$$

Where $b_{n,m} \in \{0,1\}$ represents the m^{th}-bit of $x[n]$.

By replacing equation (3) in the general 2D-DWT formulae in equation (2), we obtain:

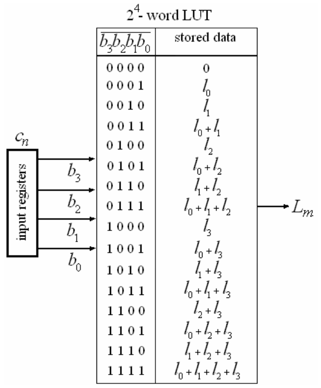
$$c_{n\,(i)} = -L_{M-1\,(i-1)}\, 2^{M-1} + \sum_{m=0}^{M-2} L_{m\,(i-1)}\, 2^m \tag{4}$$

$$d_{n\,(i)} = -H_{M-1\,(i-1)}\, 2^{M-1} + \sum_{m=0}^{M-2} H_{m\,(i-1)}\, 2^m \tag{5}$$

With: $L_{m\,(i)} = \sum_{k=0}^{K_1-1} l_k b_{2n-k,m\,(i)} \qquad H_{m\,(i)} = \sum_{k=0}^{K_2-1} h_k b_{2n-k,m\,(i)}$

Where $b_{n,m(i)} \in \{0,1\}$ represents the m^{th}-bit of the approximation wavelet coefficient $c_{n(i)}$ at the i^{th} row- or column-wise stage.

From these equations, we observe that function L_m may take one of 2^{K_1} possible values and H_m may take one of 2^{K_2} possible values, given that $b_{n,m}(i) \in \{0,1\}$, and that those values correspond to all possible sum combinations of low-pass filter coefficients, in the case of L_m, and high-pass filter coefficients, in the case of H_m. These values can be pre-computed and stored in LUTs or memories, and addressed by $b_{n,m}(i)$ (see figure 3). This way, the 2D-DWT algorithm based on FIR filters is reduced to LUT accesses and summations.



$b_3b_2b_1b_0$	stored data
0 0 0 0	0
0 0 0 1	l_0
0 0 1 0	l_1
0 0 1 1	l_0+l_1
0 1 0 0	l_2
0 1 0 1	l_0+l_2
0 1 1 0	l_1+l_2
0 1 1 1	$l_0+l_1+l_2$
1 0 0 0	l_3
1 0 0 1	l_0+l_3
1 0 1 0	l_1+l_3
1 0 1 1	$l_0+l_1+l_3$
1 1 0 0	l_2+l_3
1 1 0 1	$l_0+l_2+l_3$
1 1 1 0	$l_1+l_2+l_3$
1 1 1 1	$l_0+l_1+l_2+l_3$

Fig. 3. Basic 4-input LUT DA unit for the function L_m.

Polyphase could be applied to equations (4) and (5) to double the throughput at a small increase in the area requirements. However, instead of applying polyphase, we propose the use of the interlaced register arrangement, shown in figure 2, on inputs in equations (4) and (5). This way, the basic LUT DA filter structure would consist of the interlaced input registers, the LUT structure, and an adder-tree unit, which completes the operation with scaled summation of LUT values (see figure 4). Also, in a 2D scenario, we would have a memory block. Because images are normally available in memory before processing, it is possible to take advantage of decimation in eq. (4) and (5) and process two input samples concurrently to obtain an output sample every clock cycle. For this purpose, the memory unit has been divided in two blocks: one for even samples and another for odd samples.

Given that partition of filters is avoided in this scheme, the LUT structure can be more flexibly adapted to a given platform, such as the 4-input LUT FPGAs.

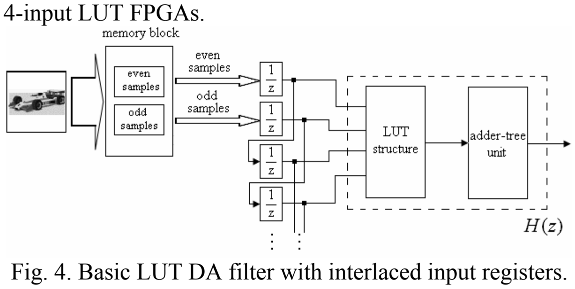


Fig. 4. Basic LUT DA filter with interlaced input registers.

The proposed LUT DA filter architecture with interlaced input registers can be efficiently applied to both orthonormal and biorthogonal wavelets. In figure 5(a), the new scheme is shown for the case of an 8-tap orthonormal wavelet. It is worth noting that, given identical order of high- and low-pass filters in orthonormal basis ($K_1 = K_2$ in equations (4) and (5)), input registers can be shared by both filters to achieve further area savings.

In figure 5(b), the proposed architecture is shown for the case of a 9-tap biorthogonal wavelet filter (e.g., the low-pass filter component of a 9/7 Daubechies wavelet). Symmetry (or anti-symmetry) of biorthogonal filters has been exploited to reduce LUT requirement by almost half.

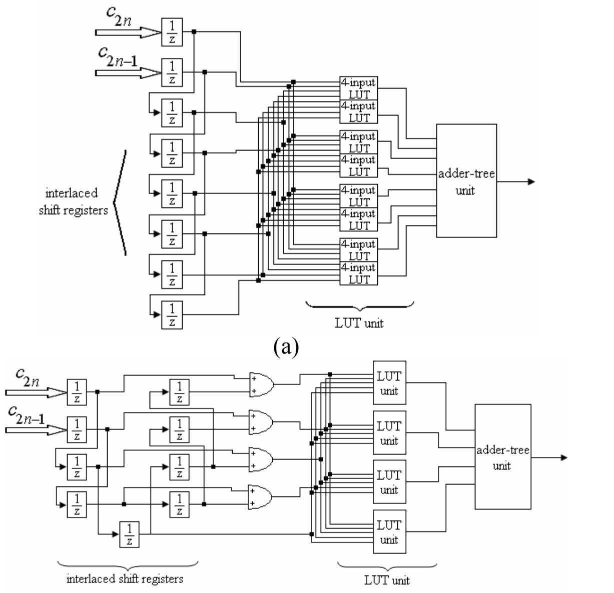


Fig. 5. Parallel DA filter structure with interlaced input registers:
a) 8-tap low- or high-pass filter of an orthonormal filterbank,
b) 9-tap low-pass filter of a biorthogonal filterbank.

Schemes in figure 5 efficiently implement the 2D-DWT of equations (4) and (5), or the 1D-DWT. In a 2D-DWT scenario, c_{2n} and c_{2n-1} would correspond to even and odd pixels of an image in the first transformation level. In following transformation levels, those values would correspond to previously computed even and odd wavelet coefficients.

The main advantage of the proposed filterbank architecture is that permits optimal area utilization because it can be more easily adapted to a given LUT-based architecture. For instance, a 4-tap orthonormal filterbank would require 2-input LUTs when partitioned with a strategy such as polyphase, whereas our architecture can be directly implemented with 4-input LUTs. In this case, the flexibility of our architecture permits easier adaptation and maximal area usage on most common FPGA devices based on 4-input LUTs. Similarly, in a 9/7 biorthogonal wavelet case, polyphase would require 2- and 3-input LUTs for the low-pass filter and 2-input LUTs for the high-pass filter, whereas our scheme needs 4- and 1-input LUTs for the low-pass filter and 4-input LUTs for the high pass filter. Again, it makes our approach ideal for 4-input LUT FPGAs when implementing low-order filterbanks, as is the general case in DWT.

4. IMPLEMENTATION

To evaluate performance, the proposed DA-based architecture for the DWT filterbank has been implemented using 8-tap and 9/7 Daubechies wavelets in Simulink (DSP Builder) and Altera Quartus II. The targeted device was the EP2S15F484C3, an FPGA of the Altera Stratix II family. Architectures shown in figure 5, corresponding to fully parallel LUT-based DA filterbanks using orthonormal and biorthogonal wavelets, were tested with several grayscale images and the results corroborated with Matlab code.

Precision of inputs and outputs was fixed to 8 bits, and coefficients were scaled to signed integer numbers and rounded to 9 bits precision. Additionally, DC level shifting was applied to the input image to have an input range of [-128, 127]. Enough bit precision in the addition of intermediate results was taken into account to avoid overflow, and the results were rounded/saturated to 8 bits. Finally, these results were shifted to the left by 9 bits to correct the initial coefficient scaling and DC level shifting.

Table 1 shows performance of proposed architecture for both orthonormal and biorthogonal Daubechies. We present area requirements for one level of transformation of 1D-DWT (see figure 1(a)) or one row filtering stage of 2D-DWT (see figure 1(b)).

		9/7 Daubechies	8-tap Daubechies
LUT-based DA	LEs	495	614
	Max. frequency	144.9 MHz	149.3 MHz

Table 1. Performance in terms of LEs (Logic Elements) and maximum frequency of proposed LUT DA wavelet filterbank.

Table 2 shows that our architecture achieves superior performance in both area and maximum operating frequency when compared against previous proposals for 9/7 Daubechies. As we can see, the closest results are presented in [8], which implemented pipelined lifting-based filterbanks. Although the area-efficient version in [8] presents area requirements comparable to our LUT-based scheme, its maximum frequency represent approximately one third of our results. Similarly, although its pipelined version achieves similar maximum frequency, it increases area by 55% in comparison to our implementation. As pointed out by [2], although the lifting scheme involves fewer operations, it requires increased coefficient and output precision to achieve a given performance level. These increased requirements finally make lifting-based implementations slower and bigger in hardware platforms. In [3] the authors presented a time-multiplexed approach that similarly to our biorthogonal scheme takes advantage of the filter decimation and coefficients symmetry. In our case, however, DA represents a more efficient multiplier-less technique that fully exploits the LUT-based FPGA structure. Nevertheless, we should note that compared designs were implemented on older Altera and Xilinx FPGA devices, which could have reduced their performance to some degree in comparison to our implementation.

	Area (LEs)	Max. frequency
LUT-based DA	495	144.9 MHz
Lifting scheme [8]	480	44 MHz
Pipelined lifting scheme [8]	766	157 MHz
Time-multiplexed [3]	785	85.5 MHz

Table 2. Comparison with previous architectures for 9/7 biortoghonal wavelet filterbank.

Additionally, we tested several 512x512 grayscale images with a forward and inverse DWT implementation based on the proposed LUT-based DA filterbank. In this case, a special address generation sequence was implemented to arrange partial results from each stage into the even and odd sample memory blocks, and in this way, to prepare data for dual access for the next stage.

Transformation and reconstruction results of the Lenna image are shown in figure 6. The performance achieved in terms of PSNR (Peak Signal-to-Noise Ratio) is 41.3dB for the first transformation level. In comparison with similar approaches, our scheme again achieves improved performance. For instance, with the pipelined lifting scheme, [8] achieves a lower image quality of 36.9dB, and [9], which presents a polyphase DA-based DWT, achieves only 30.2dB. Our higher PSNR values are due to careful fixed-point design, and show that our presented area results are not at the cost of the quality of the transform.

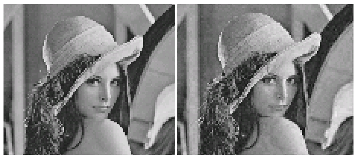


Fig 6. Reconstruction results of Lenna image after 3 transformation levels (original image is on the left).

5. CONCLUSION

Based on the very well-known Distributed Arithmetic technique, we have developed an improved filterbank architecture more flexibly adaptable to a given n-input LUT-based FPGA. By seamlessly integrating the downsampling process into the filtering stage by means of interlaced input registers and avoiding filter partition, our DA-based filterbank permits maximal area utilization on LUT-based FPGAs. Our scheme was applied to both orthonormal and biorthogonal wavelets, showing reduced area requirements and increased throughput performance. Several tests have been carried out in a 2D image compression scenario to show that our achievements were not at the expense of the quality of the transform.

REFERENCES

[1] S. Mallat, "A Theory for Multiresolution Signal Decomposition: the Wavelet Representation," in *IEEE Transactions on Pattern Anal. Machine Intell.*, Vol. II, No. 7, pp. 674-693, 1989.

[2] K. A. Kotteri, "Optimal Multiplierless Implementations of the Discrete Wavelet Transform for Image Compression Applications," *Master's Thesis*, Virginia Polytechnic Institute and State University, 2004.

[3] S. Masud and J. McCanny, "Reusable Silicon IP Cores for Discrete Wavelet Transform Applications," in *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Vol. 51, Issue 6, pp. 1114-1124, 2004.

[4] W. Sweldens, "The Lifting Scheme: a New Philosophy in Biorthogonal Wavelet Construction," in *Proc. SPIE*, 1995.

[5] Z. Zhou and D. Hung, "Digital Design of Discrete Wavelet Transform for MPEG-4," in *Proc. Annual IEEE International ASIC Conference*, pp. 333-336, 1998.

[6] Ali M. Al-Haj, "An FPGA-based Parallel Distributed Arithmetic Implementation of the 1-D Discrete Wavelet Transform," in *International Journal of Computing and Informatics (Informatica)*, Vol. 29, pp. 241-247, 2005.

[7] A. Croisier, D. J. Esteban, M. E. Levilion, and V. Rizo, "Digital Filter for PCM Encoded Signals," U.S. Patent No. 3,777,130, issued April, 1973.

[8] S.V. Silva and S. Bampi, "Area and Throughput Trade-Offs in the Design of Pipelined Discrete Wavelet Transform Architectures," in *Proc. Design, Automation and Test in Europe Conference Exhibition (DATE '05)*, 2005.

[9] A.S. Motra, P.K. Bora and I. Chakrabarti, "An Efficient Hardware Implementation of DWT and IDWT", in *Conference on Convergent Technologies for Asia-Pacific Region (TENCON '03)*, Vol. 1, pp. 95-99, 2003.