A Novel Architecture for Compression Using Lifting Scheme

1
B.Sreedevi.
prof of ECE

2
K.Anush kumar
Student of VLSI

vaagdevi college of Engineering and technology

Abstract —**a new approach for Discrete Wavelet Transform (DWT) has been proposed recently under the name of lifting scheme. This scheme presents many advantages over the convolution-based approach. In this paper, a high speed 9/7 lifting DWT algorithm which is implementation on FPGA with multi-stage pipelining structure and rational 9/7 coefficients is presented. Compared with the architecture without multi-stage pipeline, the proposed architecture has higher operating frequency, the design raises operating frequency around 3 times more fast, at the expense of about 40% more hardware area. The hardware architecture**
is suitable for high speed implementation.

Keywords-Discrete wavelet transforms (DWT); lifting scheme; multi-stage pipelining; FPGA

I. INTRODUCTION

For the last two decades the wavelet theory has been studied by many researchers [1-3] to answer the demand of better and more appropriate functions to represent signals than the one offered by the Fourier analysis.
Wavelets study each component of the signal on different resolutions and scales. One of the most attractive features that wavelet transformations provide is their capability to analyze the signals which contain sharp spikes and discontinuities.

Early implementations of the wavelet transform were based on filters' convolution algorithms. This approach requires a huge amount of computational resources. In fact at each resolution, the algorithm requires the convolution of the filters used with the approximation image. A relatively recent approach uses the lifting scheme for the implementation of the discrete wavelet transform (DWT). This method still constitutes an active area of research in mathematics and signal processing.

The lifting-based DWT scheme presents many advantages over the convolution-based approach such ascomputational efficiency, saving of memory, "in-place" computation of the DWT, integer-to-integer wavelet transform (IWT), symmetric forward and inverse transform, etc[4]. Real-time, high quality, image transmission from mobile wireless sensors requires a low-power, low-cost, high-speed hardware implementation of a state-of-the-art codec like the JPEG2000 lossy coder [5]. This coder uses the 9/7 discrete wavelet transform (DWT). The high-speed implementation of lifting-based 9/7 DWT on Field-programmable gate array (FPGA) using multi-stage pipelining and rational wavelet coefficients are presented in this paper.

The rest of the paper is organized as follows. Insection 2 the theoretical basis of the lifting-based discrete wavelet transforms are briefly presented. Section 3 describes the design of the 9/7 lifting DWT architectures. The performance evaluation of the architecture are presented in section 4, finally, section 5 presents a conclusion for this paper.

II. LIFTING-BASED WAVELET TRANSFORM

The second generation of wavelets, which is under the name of lifting scheme, was introduced by Sweldens[6]. The main feature of the lifting-based discrete wavelet transform scheme is to break up the high-pass and low-pass wavelet filters into a sequence of smaller filters that in turn can be converted into a sequence of upper and lower triangular matrices. The basic idea behind the lifting scheme is to use data correlation to remove the redundancy. Some of the advantages of this reformulation of the DWT includes "in-place" computation of the DWT, integer-to-integer wavelet transform (IWT), symmetric
Forward and inverse transform. The lifting algorithm can be computed in three main phases, namely: the split phase, the predict phase and the update phase, as illustrated in
Fig.1. 978-1

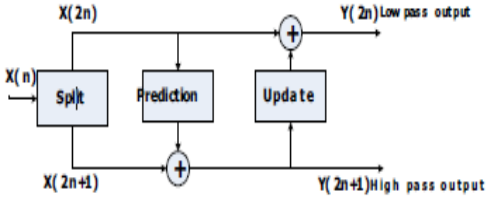


Fig.1. Split, predict and update phases of the lifting based DWT

A. Split phase.

In this split phase, the data set x(n) is split into two subsets to separate the even samples from the odd ones:

$X = X(2n), X = X(2n+1)$ *e o* (1)

B. Prediction phase.

For image data, the two sets of data have great similarity after the decomposition of the first step. In another word, there exists data redundancy. In the prediction stage, the main step is to eliminate redundancy left and give a more compact data representation. At this point, we will use the even subset x(2n) to predict the odd subset x(2n+1) using a prediction function P. The difference between the predicted value of the subset and the original value is processed and replaces this latter:

(2 1) *o* (2 1) (*e*) $Y n+ = X n+ - P X$ (2)

C. Update phase**.**

The third stage of the lifting scheme introduces the update phase. In this stage the coefficient x(2n) is lifted with the help of the neighboring wavelet coefficients. This phase is referred as the primal lifting phase or update

Phase:

(2) (2 1) (*e*) $Y n = Y n + + U X$ (3)

Where, U is the new update operator.

Since the image signals are two-dimensional, the two-dimensional wavelet transform are required. The two-dimensional wavelet transform is computed by recursive application of one-dimensional wavelet transform. After the two-dimensional wavelet transform of the first level, the image is divided into four parts. There is the horizontal low frequency-vertical low frequency component (LL), the horizontal low

Frequency-vertical high frequency component (LH), the horizontal high frequency-vertical low frequency (HL), and the horizontal high frequency-vertical high frequency (HH), respectively. The 2D-DWT is show in Fig.2.

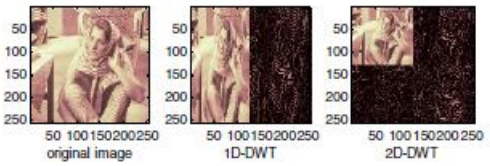


Fig.2. 2D-DWT

III. DESIGN OF THE ARCHITECTURE

The design of the 2D-DWT has 5 blocks: wavelet level select, 1D-DWT, boundary process, memory and memory control, as shown in Fig.3.

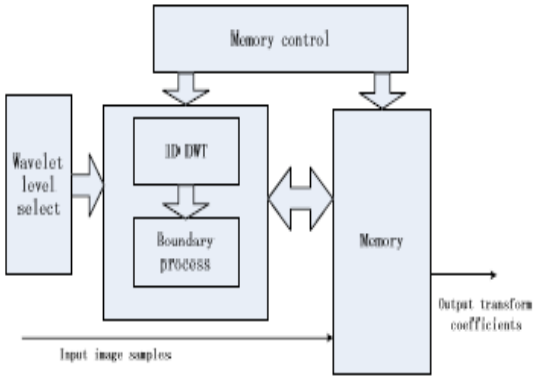


Fig.3. 2D-DWT architecture.

The input image samples are stored in memory. The memory control addresses the coefficients of band to 1D-DWT and addresses the transformed coefficients back to the memory. The finite samples filtering present a

Problem of discontinuities its boundaries. So, the image boundary information could be lost if it was not treated properly. The boundary process module is to handle this problem. A simple method to eliminate this problem consists in mirroring the boundaries of the samples. After computation of the four parts, the transformed coefficients of the LL part are transferred to next stage.

The standard LS for the 9/7 wavelet filters is shown in Fig.4, where the four lifting coefficients ⍰, ⍰, ⍰, ⍰ and the scaling factor k are evidenced. In table1 the second column summarizes the original 9/7 lifting coefficient

Values whereas the third column is the rational 9/7 lifting factorization.

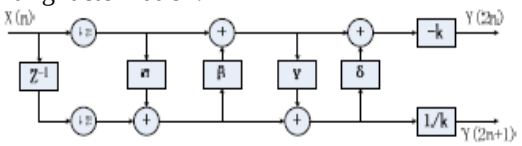


Fig.4. 9/7 lifting DWT

TABLE1. Lifting coefficients: original9/7, rational9/7 and fixed point binary of rational9/7

Coefficient	Original 9/7	Rational 9/7	Fixed point binary of rational 9/7
α	-1.586134342	-3/2	11.1
β	-0.052980118	-1/16	10.0001
γ	0.882911075	4/5	00.1100110011001100
δ	0.443506852	15/32	00.0111100000000000
k	-1.230174105	4/5	00.1100110011001100
1/k	0.812893066	5/4	01.01

The calculation of the rational 9/7 lifting factorization proposed in [7] is relatively easier than the original 9/7 lifting factorization. The rational 9/7 architecture requires only 2 float-point multipliers, 1 integer multiplier, 10 integer adders and 5 shifters. The original 9/7 architecture requires 6 float-point multipliers and 8 float-point adders. The floating operation of the former is only 1/7 of the latter, but the image compression performance of them are almost the same Table2 shows the peak signal to noise ratio (PSNR) obtained for two standard 512×512 images (‘lena’-img1, ‘barbara’-img2) of different bit rates (0.25, 0.5 and 1 bit per pixel)at wavelet decomposition level 5.

TABLE2. PSNR values of the original and rational 9/7 wavelet

Image	L	Original 9/7			Rational 9/7		
		0.25	0.5	1	0.25	0.5	1
Img1	5	34.16	37.29	40.38	34.16	37.29	40.38
Img2	5	28.37	32.31	37.19	28.46	32.32	37.21

The fourth column of table1 is the 16 binary express of the rational 9/7 lifting factorization. Here the decimal turn into binary is firstly multiple the decimal by 216, and then converted into binary. There are some differences between this method and the method of direct use of negative power of 2 to approximate. The conversion coefficient and the original coefficient have a certain
degree of deviation, but it will not impact the transform results. This approximation error can be measured by PSNR. When the fractional part of the binary is 12bit, the PSNR values of the reconstructed image have some
differences between the two methods under a small compression ratio, but the maximum value will not exceed 0.1dB. When the fractional part is 14bit, the results of the two methods have almost no difference.

1D-DWT architecture can be designed as a pipelined structure following the lifting scheme. This basic design is shown in Fig.5. This basic architecture can be designed with 6 multipliers, 8 adders and 14 registers.

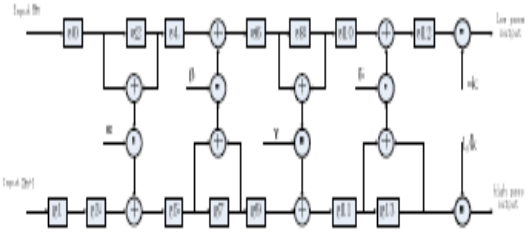


Fig.5. Basic pipeline architecture of 1D-DWT

Generic multipliers usually have high area cost. Multiplication by constant can be performed by shifted additions when the number of bits in the multiplication is large. The decimal point showed in binary representation in table1 is for documenting the design of the control, as it is not considered in the hardware multiplier, which is fix point. According to the binary representation of multiplier constants presented in table1, the multiplication by needs 7 adders, the first one performs c6+c8, the next five ones are to perform the sum of shifted partial products of c6+c8, and the last one performs the sum with c9. The multiplication by needs 4 adders, needs 3
Adders and the multiplication by needs 5 adders. The k equivalent constant has 8 high bits and this stage needed a simple multiplication, so 7 adders can perform the
Multiplication by k. The 1/k equivalent has 2 high bits, so 1 adder can perform the multiplication by 1/k. Hence, this design promotes only integer sums, reducing the area cost thus increasing the maximum
operating frequency.

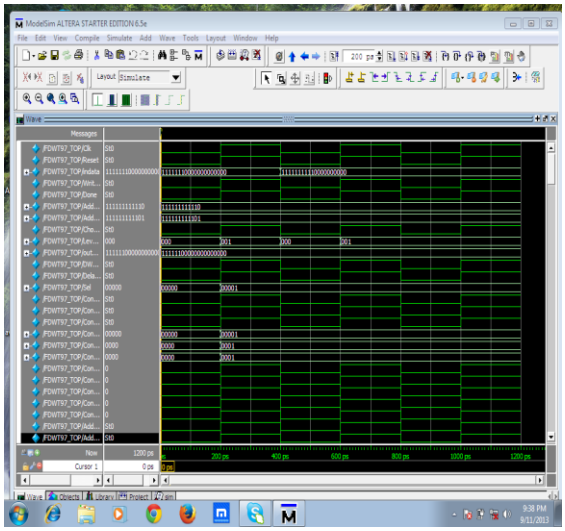
The architecture of lifting DWT can be naturally pipelined, but the add/shift stages represent the worst delay path between registers. Pipelining these stages increases the data throughput (operating frequency). The original arithmetic stage structure to perform the multiplication by is shown in Fig.6(a). The arithmetic could be improved by combination of tree-type arrangement and Horner’s rule for the accumulation of partial products in multiplication, as presented in expression (4). Horner’s rule is used for partial product accumulation to reduce the truncation error, Tree-Height Reduction is used for Latency Reduction [8]. If x=c6+c8, the multiplication by can be expressed as
TABLE1. Lifting coefficients: original9/7, rational9/7 and fixed point binary of rational9/7.

V. IMPLEMENTATION RESULTS

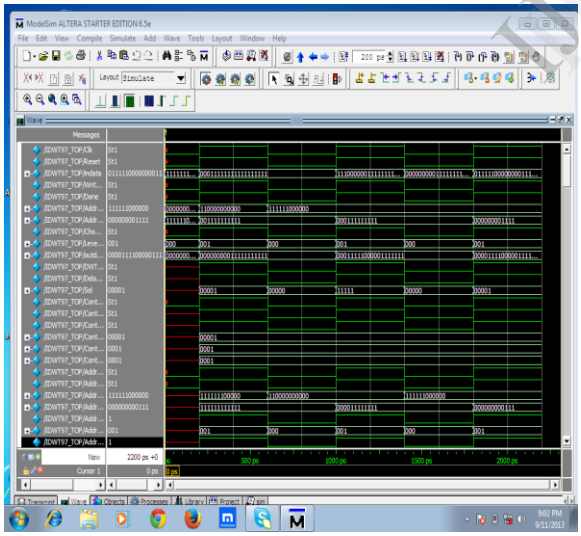
Here we present the implementation results of 1D-DWT architectures presented in section 3. The two designs were compiled and simulated in a Stratix FPGA device from Altera. Table3 shows the results of area cost,
maximum data throughput (maximum operating frequency) and number of pipeline stages. The design 1 is the architecture showed in Fig.6(a). The design 2 is an implementation with pipelined integer shifted adders, resulting in a 18 stages pipeline, which showed in Fig.6(b). This is an architecture that has a 3 times larger operating frequency of design 1. So, the architecture of design 2 definitely shows a better area-throughput

compromise than the first architecture. Compared with the architectures presented in [9], our architecture presents a larger data throughput (1.2 times of the fastest design presented in [9]) due to use of the rational 9/7 DWT coefficients, tree-type arrangement and Horner's rule for the accumulation of partial products in multiplication.

VI. Simulation Results



Dwt waveforms



IDwt waveforms

VII. CONCLUSIONS

This paper presents an efficient FPGA implementation of the DWT algorithm with lifting scheme. For the purpose of reducing the resource consumption and speeding up the performance, the rational 9/7 DWT coefficients and multi-stage pipeline technique architecture are used. By use of the rational 9/7 DWT coefficients, the PSNR values of the reconstructed image will not exceed 0.1dB. At the same time, the Horner's rule and the Tree-Height are used to reduce the multiplication truncation error and latency. The performance evaluation has proven that the proposed architecture has much higher operating frequency than the architecture which without multi-stage pipelining.

REFERENCES

[1] David S. Taubman, Michael W. Marcellin - JPEG 2000 – Image compression, fundamentals, standards and practice", Kluwer academic publishers, Second printing - 2002.
[2] G. Knowles, "VLSI Architecture for the Discrete Wavelet Transform," Electronics Letters, vo1.26, pp. 1184-1185,1990.
[3] M, Vishwanath, R. M. Owens, and M. 1. Irwin, "VLSI Architecturesfor the Discrete Wavelet Transform," IEEE Trans. Circuits And Systems II, vol. 42, no. 5, pp. 305-316, May. 1995.
[4] AS. Lewis and G. Knowles, "VLSI Architectures for 2-D Daubechies wavelet
Wavelet Transform without MUltipliers". Electron Letter, vo1.27, pp. 171-173, Jan 1991.