International Journal of Engineering and Technical Research (IJETR)
ISSN: 2321-0869, Volume-3, Issue-5, May 2015

Design and Synthesis of 3D Discrete Wavelet Transform Architecture for Real Time Application

Vijay Pandey, Ms. Saroj Behal

***Abstract*— 3D discrete wavelet transforms (dwt) is a compute-intensive task that is usually implemented on specific architectures in many real-time medical imaging systems. In this work, novel area-efficient high-throughput 3d dwt architecture is proposed based on distributed arithmetic. a tap-merging technique is used to reduce the size of DA lookup tables. The proposed architectures were designed in VHDL and mapped to a Xilinx vertex-e FPGA. The synthesis results show the proposed architecture has a low area cost. Wavelet analysis is an exciting new method for solving difficult problems in mathematics, physics, and engineering, with modern applications as diverse as wave Propagation, data compression, signal processing, image processing, pattern recognition, computer graphics, the detection of aircraft and submarines and other medical image technology. Wavelets allow complex information such as music, speech, images and patterns to be decomposed into elementary forms at different positions and scales and subsequently reconstructed with high precision. Signal transmission is based on transmission of a series of numbers. The series representation of a function is important in all types of signal transmission. The wavelet representation of a function is a new technique. Wavelet transform of a function is the improved version of Fourier transform. Fourier transform is a powerful tool for analyzing the components of a stationary signal. But it is failed for analyzing the non stationary signal where as wavelet transform allows the components of a non-stationary signal to be analyzed. In this dissertation, our main goal is to find out the advantages of wavelet transform compared to Fourier transform**

***Index Terms*—DWT, DA, FPGA.**

I. INTRODUCTION

The wavelet transform is similar to the Fourier transform (or much more to the windowed Fourier transform) with a completely different merit function. The main difference is this: Fourier transform decomposes the signal into sines and cosines, i.e. the functions localized in Fourier space; in contrary the wavelet transform uses functions that are localized in both the real and Fourier space. Generally, the wavelet transform can be expressed by the following equation:

$$F(a,b) = \int_{-\infty}^{\infty} f(x)\, \psi^{*}_{(a,b)}(x)\, \mathrm{d}x$$

Vijay Pandey (Student), CBS Group Institution Haryana
Ms. Saroj Behal (Assistant Professor), CBS Group Institution Haryana

It is seen, the Wavelet transform is in fact an infinite set of various transforms, depending on the merit function used for its computation. This is the main reason, why we can hear the term "wavelet transforms" in very different situations and applications. There are also many ways how to sort the types of the wavelet transforms

(a)Discrete Wavelet Transform

The discrete wavelet transform (DWT) is an implementation of the wavelet transform using a discrete set of the wavelet scales and translations obeying some defined rules. In other words, this transform decomposes the signal into mutually orthogonal set of wavelets, which is the main difference from the continuous wavelet transform (CWT), or its implementation for the discrete time series sometimes called discrete-time continuous wavelet transform (DT-CWT).

The wavelet can be constructed from a scaling function which describes its scaling properties. The restriction that the scaling functions must be orthogonal to its discrete translations implies some mathematical conditions on them which are mentioned everywhere, e.g. the dilation equation

$$F(a,b) = \int_{-\infty}^{\infty} f(x)\, \psi^{*}_{(a,b)}(x)\, \mathrm{d}x$$

Where *S* is a scaling factor (usually chosen as 2). Moreover, the area between the function must be normalized and scaling function must be orthogonal to its integer translations, i.e.

$$F(a,b) = \int_{-\infty}^{\infty} f(x)\, \psi^{*}_{(a,b)}(x)\, \mathrm{d}x$$

After introducing some more conditions (as the restrictions above does not produce unique solution) we can obtain results of all these equations, i.e. the finite set of coefficients a_k that define the scaling function and also the wavelet. The wavelet is obtained from the scaling function as *N* where *N* is an even integer. The set of wavelets then forms an orthonormal basis which we use to decompose the signal. Note that usually only few of the coefficients a_k are nonzero, which simplifies the calculations.

In the following figure, some wavelet scaling functions and wavelets are plotted. The most known family of orthonormal wavelets is the family of Daubechies. Her wavelets are usually denominated by the number of nonzero coefficients a_k, so we usually talk about Daubechies 4, Daubechies 6, etc. wavelets. Roughly said, with the increasing number of wavelet coefficients the functions become smoother. See the comparison of wavelets Daubechies 4 and 20 below. Another mentioned wavelet is the simplest one, the Haar wavelet, which uses a box function as the scaling function.

$$F(a,b) = \int_{-\infty}^{\infty} f(x)\,\psi^*_{(a,b)}(x)\,\mathrm{d}x$$

Haar scaling function and wavelet (left) and their frequency content (right).

$$F(a,b) = \int_{-\infty}^{\infty} f(x)\,\psi^*_{(a,b)}(x)\,\mathrm{d}x$$

Daubechies 4 scaling function and wavelet (left) and their frequency content (right).

$$F(a,b) = \int_{-\infty}^{\infty} f(x)\,\psi^*_{(a,b)}(x)\,\mathrm{d}x$$

Daubechies 20 scaling function and wavelet (left) and their frequency content (right).

There are several types of implementation of the DWT algorithm. The oldest and most known one is the Mallat (pyramidal) algorithm. In this algorithm two filters – smoothing and non-smoothing one – are constructed from the wavelet coefficients and those filters are recurrently used to obtain data for all the scales. If the total number of data $D = 2^N$ is used and the signal length is L, first $D/2$ data at scale $L/2^{N-1}$ are computed, then $(D/2)/2$ data at scale $L/2^{N-2}$, up to finally obtaining 2 data at scale $L/2$. The result of this algorithm is an array of the same length as the input one, where the data are usually sorted from the largest scales to the smallest ones. Within Gwyddion the pyramidal algorithm is used for computing the discrete wavelet transform. Discrete wavelet transform in 2D can be accessed using DWT module.

Discrete wavelet transform can be used for easy and fast denoising of a noisy signal. If we take only a limited number of highest coefficients of the discrete wavelet transform spectrum, and we perform an inverse transform (with the same wavelet basis) we can obtain more or less Denoise signal. There are several ways how to choose the coefficients that will be kept. Within Gwyddion, the universal thresholding, scale adaptive thresholding [2] and scale and space adaptive thresholding [3] is implemented. For threshold determination within these methods we first determine the noise variance guess given by

$$F(a,b) = \int_{-\infty}^{\infty} f(x)\,\psi^*_{(a,b)}(x)\,\mathrm{d}x$$

Where Y_{ij} corresponds to all the coefficients of the highest scale sub band of the decomposition (where most of the noise is assumed to be present). Alternatively, the noise variance can be obtained in an independent way, for example from the AFM signal variance while not scanning. For the highest frequency sub band (universal thresholding) or for each sub band (for scale adaptive thresholding) or for each pixel neighborhood within sub band (for scale and space adaptive thresholding) the variance is computed as

$$F(a,b) = \int_{-\infty}^{\infty} f(x)\,\psi^*_{(a,b)}(x)\,\mathrm{d}x$$

Threshold value is finally computed as

$$F(a,b) = \int_{-\infty}^{\infty} f(x)\,\psi^*_{(a,b)}(x)\,\mathrm{d}x$$

Where

$$F(a,b) = \int_{-\infty}^{\infty} f(x)\,\psi^*_{(a,b)}(x)\,\mathrm{d}x$$

When threshold for given scale is known, we can remove all the coefficients smaller than threshold value (hard thresholding) or we can lower the absolute value of these coefficients by threshold value (soft thresholding).

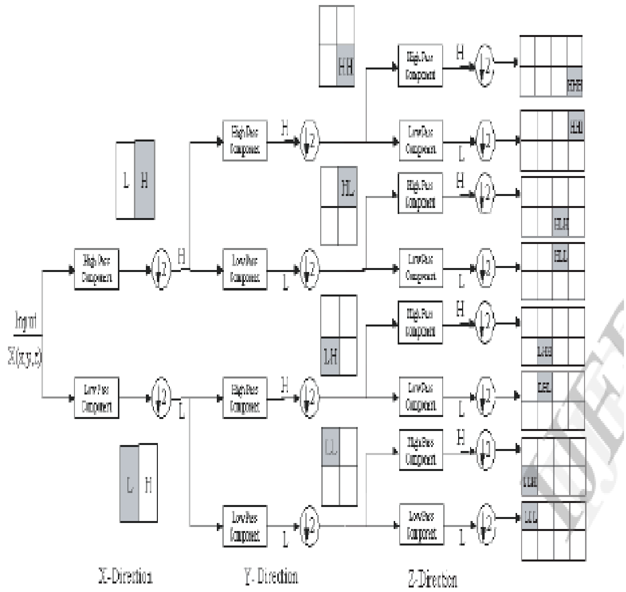
DWT denoising can be accessed with *Data Process → Integral Transforms → DWT Denoise*.

(b) Continuous Wavelet Transform

Continuous wavelet transform (CWT) is an implementation of the wavelet transform using arbitrary scales and almost arbitrary wavelets. The wavelets used are not orthogonal and the data obtained by this transform are highly correlated. For the discrete time series we can use this transform as well, with the limitation that the smallest wavelet translations must be equal to the data sampling. This is sometimes called Discrete Time Continuous Wavelet Transform (DT-CWT) and it is the most used way of computing CWT in real applications. In principle the continuous wavelet transform works by using directly the definition of the wavelet transform, i.e. we are computing a convolution of the signal with the scaled wavelet. For each scale we obtain by this way an array of the same length N as the signal has. By using M arbitrarily chosen scales we obtain a field $N \times M$ that represents the time-frequency plane directly. The algorithm used for this computation can be based on a direct convolution or on a convolution by means of multiplication in Fourier space (this is sometimes called Fast Wavelet Transform).The choice of the wavelet that is used for time-frequency decomposition is the most important thing. By this choice we can influence the time and frequency resolution of the result. We cannot change the main features of WT by this way (low frequencies have good frequency and bad time resolution; high frequencies have good time and bad frequency resolution), but we can somehow increase the total frequency of total time resolution. This is directly proportional to the width of the used wavelet in real and Fourier space.

II. Model of Proposed Work

(A)Combination of three 1D DWT in the *x*, *y* and *z* directions

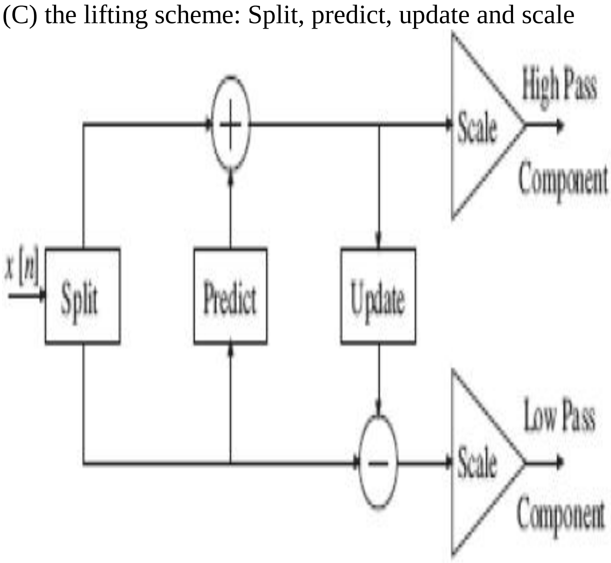


Be considered as a combination of three 1D DWT in the *x*, *y* and *z* directions, as shown in Fig. 1. The preliminary work in the DWT processor design is to build 1D DWT modules, which are composed of high-pass and low-pass filters that perform a convolution of filter coefficients and input pixels. After a one-level of 3D discrete wavelet transform, the volume of image is decomposed into HHH, HHL, HLH, HLL, LHH, LHL, LLH and LLL signals

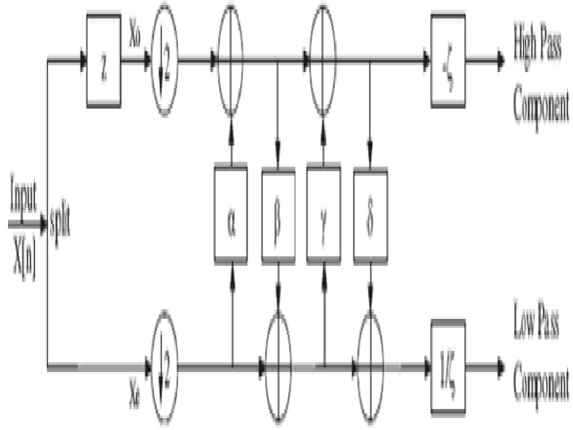
(B) Lifting Scheme

Lifting Scheme The basic idea behind the lifting scheme is very simple; try to use the correlation in the data to remove redundancy [4, 5]. First split the data into two sets (split phase) i.e., odd samples and even samples as shown in Figure

(C) the lifting scheme: Split, predict, update and scale



(D) 1-D lifting scheme of daubechies 9/7 for forward Wavelet DWT



Because of the assumed smoothness of the data, we predict that the odd samples have a value that is closely related to their neighboring even samples. We use *N* even samples to predict the value of a neighboring odd value (predict phase). With a good prediction method, the chance is high that the original odd sample is in the same range as its prediction. We calculate the difference between the odd sample and its prediction and replace the odd sample with this difference. As long as the signal is highly correlated, the newly calculated odd samples will be on the average smaller than the original one and can be represented with fewer bits. The odd half of the signal is now transformed. To transform the other half, we will have to apply the predict step on the even half as well. Because the even half is merely a sub-sampled version of the original signal, it has lost some properties that we might want to preserve. In case of images we would like to keep the intensity (mean of the samples) constant throughout different levels. The third step (update phase) updates the even samples using the newly calculated odd samples such that the desired property is preserved. Now the circle is round and we can move to the next level.

We apply these three Steps repeatedly on the even samples and transform each time half of the even samples, until all samples are transformed.

III. CONCLUSION & FUTURE WORKS

Due to their capability to localize in time, wavelet transforms readily lend themselves to no stationary signal analysis. Detection of short duration events, on the other hand, is limited in Fourier analysis by the width of the windowing function used in the short-time Fourier transform. Wavelet transforms exist that project a finite energy function onto to an orthonormal basis of *L2(R).* The corresponding multi resolution analysis decomposes the function into a set of details at different resolutions and a smoothed version of the original function. As with the Fourier transform, a "fast wavelet transform" exists. However, the fast wavelet transform generates a multi resolution analysis in O (*n)* time; Whereas, a fast Fourier transform takes (*n* log *n)* time. Our intent in this paper was to present the basic concept of the wavelet transform from a viewpoint that targets signal analysis applications. Much of the current literature utilizes a high level of mathematical terminology. Our hope was to provide an brief introduction to the primary underlying ideas in a relatively intuitive manner. For those who are interested, we provide an annotated bibliography that includes some of the key papers in the field. With each listing is a short description of the contents. Most of the papers require an understanding of Fourier analysis and sometimes an understanding of more general functional analysis principles. To help specify the mathematical sophistication of a paper, we adopt a relative rating scale, based upon our experience of reading the papers, 1 meaning little or no mathematical sophistication

REFERENCES

[1]http://www.mathworks.com/access/helpdesk/help/toolbox/wavelet/wavelet.html Robi Polikar, The Wavelet Tutorial, http://users.rowan.edu/~polikar/WAVELETS/WTpart1.html Robi Polikar, Multi resolution Wavelet Analysis of Event Related Potentials for the Detection of Alzheimer's
Disease, Iowa State University, 06/06/1995.

[2] Amara Grapes, An Introduction to Wavelets, IEEE Computational Sciences and Engineering, Vol. 2, No 2, Summer 1995, pp 50-61. Resonance Publications, Inc. Wavelets. http://www.resonancepub.com/wavelets.htmR. Crandall, Projects in Scientific Computation, Springer-Verilag, New York, 1994, pp. 197-198, 211-212.

[3] Y. Meyer, Wavelets: Algorithms and Applications, Society for Industrial and Applied Mathematics, Philadelphia, 1993, pp. 13-31, 101-105.

[4] G. Kaiser, A Friendly Guide to Wavelets, Birkhauser, Boston, 1994, pp. 44-45. W. Press et al., Numerical Recipes in FORTRAN, Cambridge University Press, New York, 1992, pp. 498-499,584-602.

[5]M. Petteril and C. Herley, "Wavelets and Filter Banks: Theory and Design," IEEE Transactions on Signal Processing, Vol. 40, 1992, pp. 2207-2232.

[6] Daubechies, "Orthonormal Bases of Compactly Supported Wavelets," Comm. Pure Appl. Math., Vol 41, 1988, pp. 906-966.

[7] V. Wickerhauser, Adapted Wavelet Analysis from Theory to Software, AK Peters, Boston, 1994, pp. 213-214, 237, 273-274, 387.

[8] M.A. Cody, "The Wavelet Packet Transform," Dr. Dobb's Journal, Vol 19, Apr. 1994, pp. 44-46, 50-54.

[9] Bradley, C. Brislawn, and T. Hopper, "The FBI Wavelet/Scalar Quantization Standard for Gray-scale Fingerprint Image Compression," Tech. Report LA-UR-93-1659, Los Alamos Nat'l Lab, Los Alamos, N.M. 1993.

[10] D. Donoho, "Nonlinear Wavelet Methods for Recovery of Signals, Densities, and Spectra from Indirect and Noisy Data," Different Perspectives on Wavelets, Proceeding of Symposia in Applied Mathematics, Vol 47, I. Daubechies ed. Amer. Math. Soc., Providence, R.I., 1993, pp. 173-205.

[11] B. Vidakovic and P. Muller, "Wavelets for Kids," 1994, unpublished. Part One, and Part Two.

[12] J. Scargle et al., "The Quasi-Periodic Oscillations and Very Low Frequency Noise of Scorpius X-1 as Transient Chaos: A Dripping Handrail?," Astrophysical Journal, Vol. 411, 1993, L91-L94.

[13] M.V. Wickerhauser, "Acoustic Signal Compression with Wave Packets," 1989. Available by TeXing this TeX

[14] J. Stolarek, "Realization of Daubechies transform using lattice structure", The Collection of Scientific Works of ISDMCI 2, 188–192 (2009).

[15] Burrus, C.S., Goperath, R.A., and Guo, H., "Introduction to Wavelet and Wavelets Transforms", A primer Upper Saddle, NS (U.S.A), Prantice Hall, Inc., 1998.

[16] Misite, M., Misite, Y., Oppenheim, G., and Pog., J.M., "Wavelet Toolbox : Computation, Visualization, and Programming", User's Guide, version 1, Math works, Inc., 1997.

[17] Donoho, D.L., and Johnston, I.M., "Adapting to unknown smoothness via wavelet shrinkage", Journal of American Statistic Assn., Stanford Statistics, Dep., report TR-U25, June 1995