

# HDL Implementation of the DWT Algorithm for Image Compression

Ms.Pragya Mishra, Mr.Parag Parandkar, Prof.H.R.Singh,

Electronics and Communication Engineering

Oriental University, Indore (M.P.) India

mishrapragya25@gmail.com

**Abstract** - The Discrete Wavelet Transform (DWT) algorithm is well known and commonly used for data and image compression. This paper presents an approach towards HDL implementation of the Discrete Wavelet Transform (DWT) for image compression. DWT is designed in HDL using filter method to compressed image. Daubechies (db2)-4 tap filter is used in this design. Coefficients of Daubechies filters for DWT algorithm are fixed. This design can be used for image compression in robotics.

**Keywords:** DWT (Discrete Wavelet Transform), Daubechies (db2) wavelet filter, VLSI Design.

## INTRODUCTION

Data compression is the technique to reduce the redundancies in data representation in order to decrease data storage requirements and hence communication costs. Reducing the storage requirement is equivalent to increasing the capacity of the storage medium and hence communication bandwidth. Thus the development of efficient compression techniques will continue to be a design challenge for future communication systems and advanced multimedia applications.

Data is represented as a combination of information and redundancy. Information is the portion of data that must be preserved permanently in its original form in order to correctly interpret the meaning or purpose of the data. Redundancy is that portion of data that can be removed when it is not needed or can be reinserted to interpret the data when needed. Most often, the redundancy is reinserted in order to generate the original data in its original form. A technique to reduce the redundancy of data is defined as Data compression. The redundancy in data representation is reduced such a way that it can be subsequently reinserted to recover the original data, which is called decompression of the data.

Applications of data compression are primarily in transmission and storage of information. Image transmission applications are in broadcast television, remote sensing via satellite, military communication via aircraft, radar and sonar, teleconferencing, and computer communications.

### Wavelet Transform

The transform of a signal is just another form of representing the signal. It does not change the information content present in the signal. The Wavelet Transform provides a time-frequency representation of the signal. It was developed to overcome the short coming of the Short Time Fourier Transform (STFT), which can also be used to analyze non-stationary signals. While STFT gives a constant resolution at all frequencies, the Wavelet

Transform uses multi-resolution technique by which different frequencies are analyzed with different resolutions.

The wavelet analysis is done similar to the STFT analysis. The signal to be analyzed is multiplied with a wavelet function just as it is multiplied with a window function in STFT, and then the transform is computed for each segment generated. However, unlike STFT, in Wavelet Transform, the width of the wavelet function changes with each spectral component. The Wavelet Transform, at high frequencies gives good time resolution and poor frequency resolution, while at low frequencies the Wavelet Transform gives good frequency resolution and poor time resolution.

### DWT and Filter Banks

The Wavelet Series is just a sampled version of CWT and its computation may consume significant amount of time and resources, depending on the resolution required. The Discrete Wavelet Transform (DWT), which is based on sub-band coding, is found to yield a fast computation of Wavelet Transform. It is easy to implement and reduces the computation time and resources required.

In CWT, the signals are analyzed using a set of basis functions which relate to each other by simple scaling and translation. In the case of DWT, a time-scale representation of the digital signal is obtained using digital filtering techniques. The signal to be analyzed is passed through filters with different cutoff frequencies at different scales.

Filters are one of the most widely used signal processing functions. Wavelets can be realized by iteration of filters with rescaling. The resolution of the signal, which is a measure of the amount of detail information in the signal, is determined by the filtering operations, and the scale is determined by up sampling and down sampling (sub sampling) operations.

The DWT is computed by successive low pass and high pass filtering of the discrete time-domain signal. This is called the Mallat algorithm or Mallat-tree decomposition. Its significance is in the manner it connects the continuous time multiresolution to discrete-time filters. In the figure, the signal is denoted by the sequence  $x[n]$ , where  $n$  is an integer. The low pass filter

is denoted by  $g[n]$  while the high pass filter is denoted by  $h[n]$ . At each level, the high pass filter produces detail information, while the low pass filter associated with scaling function produces coarse approximations.

The reconstructions of the original signal from the wavelet coefficients. Basically, the reconstruction is the reverse process of decomposition. The approximation and detail coefficients at every level are up sampled by two, passed through the low pass and high pass synthesis filters and then added. This process is continued through the same number of levels as in the decomposition process to obtain the original signal.

*One level of the transform*

The DWT of a signal  $x$  is calculated by passing it through a series of filters. First the samples are passed through a low pass filter with impulse response  $g$  resulting in a convolution of the two:

$$y[n] = (x * g)[n] = \sum_{k=-\infty}^{\infty} x[k]g[n - k]$$

The signal is also decomposed simultaneously using a high-pass filter  $h$ . The output giving the details of coefficients (from the high-pass filter) and approximation coefficients (from the low-pass). It is important that the two filters are related to each other and they are known as a quadrature mirror filter. However, since half the frequencies of the signal have now been removed, half the samples can be discarded according to Nyquist's rule. The filter outputs are then sub sampled by 2 (Mallat's and the common notation is the opposite,  $g$ - high pass and  $h$ - low pass)

$$y_{low}[n] = \sum_{k=-\infty}^{\infty} x[k]g[2n - k]$$

$$y_{high}[n] = \sum_{k=-\infty}^{\infty} x[k]h[2n - k]$$

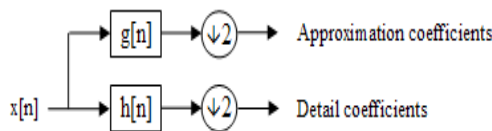


Figure 1. One level of transform

**Multi-Resolution Analysis using Filter Banks**

Filters are one of the most widely used signal processing functions. Wavelets can be realized by iteration of filters with rescaling. The resolution of the signal, which is a measure of the amount of detail information in the signal, is determined by the filtering operations, and the scale is determined by up sampling and down sampling (sub sampling) operations.

The DWT is computed by successive low pass and high pass filtering of the discrete time-domain signal as shown in figure 02. This is called the Mallat algorithm or Mallat-tree decomposition.

Its significance is in the manner it connects the continuous time multiresolution to discrete-time filters. In the figure, the signal is denoted by the sequence  $x[n]$ , where  $n$  is an integer. The low pass filter is denoted by  $G_0$  while the high pass filter is denoted by  $H_0$ . At each level, the high pass filter produces detail information;  $d[n]$ , while the low pass filter associated with scaling function produces coarse approximations,  $a[n]$ .

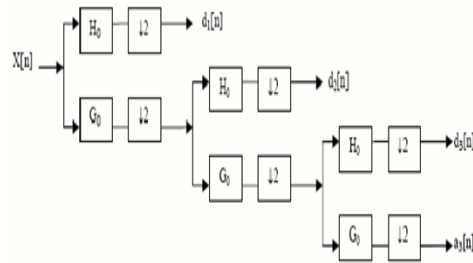


Figure 2. Three-level wavelet decomposition trees

*Discrete Wavelet Transform Operation*

When a signal passes through these filters, it is split into two bands. The low pass filter, which corresponds to an averaging operation, extracts the coarse information of the signal. The high pass filter, which corresponds to a differencing operation, extracts the detail information of the signal. The output of the filtering operations is then decimated by two.

A two-dimensional transform (see Figure 03) can be accomplished by performing two separate one-dimensional transforms. First, the image is filtered along the  $x$  dimension and decimated by two. Then, it is followed by filtering the sub-image along the  $y$ -dimension and decimated by two. Finally, we have split the image into four bands denoted by LL, HL, LH and HH after one-level decomposition.

The Multiresolution decomposition approach in the two-dimensional signal is demonstrated in figures. After the first level of decomposition, it generates four sub bands LL1, HL1, LH1, and HH1. Considering the input signal is an image, the LL1 sub bands can be considered as a 2:1 sub sampled (both horizontally and vertically) version of image. The other three sub bands HL1, LH1, and HH1 contain higher frequency detail information. These spatially oriented (horizontal, vertical or diagonal) sub bands mostly contain information of local discontinuities in the image and the bulk of the energy in each of these three sub bands is concentrated in the vicinity of areas corresponding to edge activities in the original image. Since LL1 is a coarser approximation of the input, it has similar spatial and statistical characteristics to the original image. As a result, it can be further decomposed into four sub bands LL2, LH2, HL2 and HH2 as shown in figure 03 based on the principle of Multiresolution analysis. Accordingly the image is decomposed into any number of levels. The

same computation can continue to further decompose LL2 into higher levels.

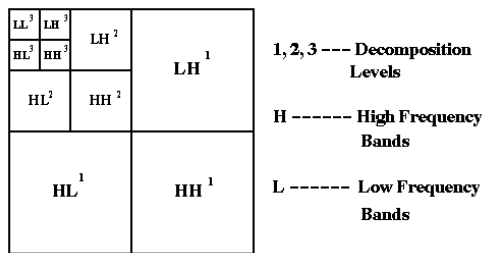


Figure 3. Extension of DWT in two - dimensional signals  
 Implementation of DWT Algorithm

The DWT Algorithm is implemented using Verilog HDL on Altera's Quartus II 9.0 Web Edition tool.

In DWT module it consists of low pass, high pass filter and a down sampler module. Low pass filter, high pass filter and down sample are coded in different module. After designing these modules a top level module is designed using low pass, high pass and down sample module to DWT. This top level module is the DWT module. The pixel value of the image is taken as a input of the DWT algorithm and this input is processed by low pass and high pass filter followed by a down sampler. The output of the high pass filter is wavelet coefficients and the output of the low pass filter is the scaling coefficients. Now in second stage the outputs of the low pass filter become the inputs for the second stage of DWT algorithm. In a similar way further DWT scaling and wavelet coefficients are computed.

RTL of DWT Algorithm

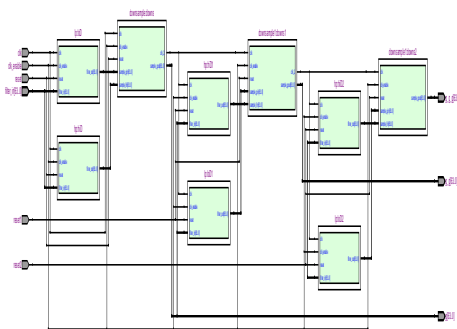


Figure 4. RTL of DWT Algorithm

**Results**

The grayscale cameraman image of 64x64 pixels is taken as an input (generated with the help of MATLAB Tool) for the DWT algorithm designed in HDL and is compressed at two levels. After two level compressions, the image is represented by 16x16 pixels. The compressed simulated data is in binary format thus to visualize these compressed data image plotting tool of MATLAB is used. Figure 05 and figure 06 shows the original image and compressed image using HDL simulation and MATLAB tool. The compressed image is basically the LL-2 part of the DWT

algorithm. DWT algorithm for image compression can be used at any level of compression.

The DWT algorithm is designed and verified with the help of simulation. This algorithm is designed in verilog HDL so it can be implemented on the hardware. This DWT algorithm gives the compressed result of input image. This result is verified with the help of simulation tool.

**Compression using HDL Simulation**

Test Image 1



Original Image                      Compressed Image after  
 (a)    Two level DWT  
 (b)

Figure .5 Compression using HDL Simulation

**Compression using HDL Simulation**

Test Image 2



Original Image                      Compressed Image after Two level  
 (a)    DWT  
 (b)

Figure 6. Compression using HDL Simulation

**CONCLUSION**

The 2D – DWT algorithm code was written in the Verilog HDL. It is then synthesized and simulated successfully. Pixels value of input images are taken, and DWT coefficients is calculated through DWT algorithm designed in Verilog HDL and the same images are used for calculating DWT using MATLAB. Image compression with the help of HDL simulation and using MATLAB tool is compared. It is found that the value of DWT is approximately same in both the implementation methodology; however, the intensity of the image compressed with MATLAB tool is higher than that of the HDL simulation. This is due to shifting of the values in HDL coding.

Variable levels of compression can be easily achieved using HDL. The number of DWT stages can be varied, resulting in different number of sub bands. Different filter banks with different characteristics can be used.

Efficient fast algorithm (pyramidal computing scheme) for the computation of discrete wavelet coefficients makes a wavelet transform based encoder computationally efficient.

## REFERENCES

1. Tze-Yun Sung, Hsi-Chin Hsin Yaw-Shih Shieh and Chun-Wang Yu, "Low-Power Multiplierless 2-D DWT and IDWT Architectures Using 4-tap Daubechies Filters", Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies, December 2006.
2. Abdullah AL Muhit, Md. Shabiul Islam and Masuri Othman, "VLSI Implementation of Discrete Wavelet Transform (DWT) for Image Compression", 2nd International Conference on Autonomous Robots and Agents, 13-15 December 2004.
3. Motra A. S. Bora P. K. and Chakrabarti I. "An efficient hardware implementation of DWT and IDWT", Conference on Convergent Technologies for Asia-Pacific Region, 15-17 October 2003.
4. Yun-Nan Chang and Yan-Sheng Li, "Design of highly efficient VLSI architectures for 2-D DWT and 2-D IDWT", IEEE Workshop on Signal Processing Systems, 26-28 September 2001.
5. Keshab K. Parhi and Takao Nishitani, "VLSI architectures for discrete wavelet transforms", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 1, No. 2, pp. 191-202, June 1993.
6. David S. Taubman, Michael W. Marcellin, "JPEG 2000 – Image compression, fundamentals, standards and practice", Kluwer academic publishers, Second Edition, 2002.
7. Simone Borgio, Davide Bosisio, Fabrizio Ferrandi, Matteo Monchiero, Marco D. Santambrogio, Donatella Sciuto and Antonino Tumeo, "Hardware DWT accelerator for MultiProcessor System-on-Chip on FPGA", Embedded Computer Systems: Architectures, Modeling and Simulation, 2006. IC-SAMOS 2006, July 2006, pp. 107 - 114.
8. Mel Tsai, BDTI, "Designing Low-Power Signal Processing Systems", <http://www.dspdesignline.com/showArticle.jhtml?articleID=187002923>.
9. Tze-Yun, "Low-power and high-performance 2-D DWT and IDWT architectures based on 4-tap Daubechies filters", Proceedings of the 7th WSEAS International Conference on Multimedia Systems and Signal Processing, Hangzhou, China, pp. 50-55, 2007.
10. S. Baloch, I. Ahmed, T. Arslan, A. Stoica, "Low power domain-specific reconfigurable array for discrete wavelet transforms targeting multimedia applications", "International Conference on Field Programmable Logic and Applications", pp. 618-621, International Conference on Field Programmable Logic and Applications, 2005.
11. Anirban Das, Anindya Hazra, and Swapna Banerjee, "An Efficient Architecture for 3-D Discrete Wavelet Transform" IEEE Transactions on circuits and systems for video technology, vol. 20, no. 2, February 2010.
12. Chin-Fa Hsieh, Tsung-Han Tsai, Neng-Jye Hsu, and Chih-Hung Lai, "A Novel, Efficient Architecture for the 1D, Lifting-Based DWT with Folded and Pipelined Schemes" Department. of Electronics Engineering, China Institute of Technology, Taipei, Taiwan and Department. of Electrical Engineering, National Central University, Chung-Li, Taiwan, IEEE Trans 2004.
13. Kuan-Hung Chen and Yuan-Sun Chu, "A Low-Power Multiplier with the Spurious Power Suppression Technique," IEEE Trans. On Very Large Scale Integration (VLSI) Systems, vol. 15, no. 7, July 2007, pp. 846-850.
14. Kishore Andra, Chaitali Chakrabarti and Tinku Acharya, "A VLSI Architecture for Lifting-Based Forward and Inverse Wavelet Transform" IEEE Transaction on signal processing, VOL. 50, NO. 4, pp. 966-977, April 2002.
15. Movva S and Srinivasan S "A novel architecture for lifting-based discrete wavelet transform for JPEG2000 standard suitable for VLSI Implementation" VLSI Design, 2003. Proceedings. 16th International Conference On 4-8, Jan 2003 Page(s): 202 – 207. Washington, DC, USA
16. Dusan Suvakovic, C. Andre, Salama "A Pipelined Multiply-Accumulate Unit Design for Energy Recovery DSP Systems" IEEE International Symposium on Circuits and Systems, May 28-31, 2000.
17. Jung- Yup Kang, Jean-Luc Gaudiot, "A Simple High-Speed Multiplier Design" IEEE Transactions on computers, Vol 55, No 10, October 2006, pp 1253 – 1258.
18. Zhijun Huang, Milos D. Ercegovac, "High-performance Low-power Left-to-Right Array Multiplier Design" IEEE Transactions on computers, Vol 54, No. 3, Mar 2005, pp 272 – 283.
19. C. S. Wallace, "A Suggestion for a Fast Multiplier", IEEE Trans. computers, vol 13, no. 2, pp 14 – 17, 1964.
20. Vojin G. Oklobzija, Bart R. Zaydel, Hoang Q. Dao, Sanu Mathew and Ram Krishnamurthy, "Comparison of high-performance VLSI adders in the energy-delay space", IEEE Trans. VLSI Systems, vol. 13, no. 6, pp. 754-758, June 2005.
21. J-Y Kang and J-L Gaudiot, "A Fast and Well structured Multiplier," EUROMICRO Symp. Digital System Design, pp. 508 – 515, Aug 2004.
22. Nagabushanam, M., Cyril prasanna Raj, and Ramachandran, S. 'Modified VLSI implementation of DA-DWT for image compression', International Journal of Signal and Imaging Systems Engineering, (to be published by inderscience) Vol. x, No. x, pp. xxx-xxx.
23. Nagabushanam, M., Cyril prasanna Raj, and Ramachandran, S. (2009), 'Design and implementation of parallel and pipelined Distributive arithmetic based discrete wavelet transform IP core'. EJSR International Journal, Vol. 35. No. 3, pp. 379-392.
24. Nagabushanam, M., Cyril Prasanna Raj, and Ramachandran, S. (2011), 'Design and FPGA implementation of Modified Distributive arithmetic based DWT-IDWT processor for image compression', International Conference on Communication and Signal Processing, February, NIT, Calicut, India, p. 69.
25. Nagabushanam, M. and Ramachandran, S. 'Fast Implementation of Lifting based DWT Architecture for Image Compression', Global Journal of Computer Science and Technology (F), Volume – XII, Issue XI, Version I, p. 23-29., June 2012.
26. M. Mottaghi-Dastjerdi, A. Afzali-Kusha, and M. Pedram BZ-FAD: A Low-Power Low-Area Multiplier Based on Shift-and-Add Architecture, IEEE Transactions on Very large Scale Integration (VLSI) systems, VOL. 17, no. 2, Feb 2009.