Design and Analysis of Discrete Wavelet Transform(DWT) for Image Compression using VHDL.

Conference Paper · January 2005

Source: DBLP

3 authors, including:

Abdullah Al Muhit

Royal Perth Hospital

Md Shabiul Islam

Multimedia University, Cyberjaya, Malaysia.

International Conference on Parallel and Distributed Processing Techniques and Applications
June 27-30, 2005 Las Vegas, Nevada, USA

Design and Analysis of Discrete Wavelet Transform (DWT) for Image Compression using VHDL

Abdullah Al Muhit
Faculty of Engineering
Multimedia University(MMU)
Jalan multimedia, Cyberjaya
63100 Selangor
Malaysia
E.mail:Muhit@mmu.edu.my

Md. Shabiul Islam
Faculty of Engineering
Multimedia University(MMU)
Jalan multimedia, Cyberjaya
63100 Selangor
Malaysia

Masuri Othman
Institute of Microengineering and Nanoelectronics(IMEN)
University Kebangsaan Malaysia(UKM)
43600 UKM, Bangi, Selangor DE, Malaysia.

Abstract - This paper presents the use of Electronic Design Automation (EDA) for implementing the 2-D Discrete Wavelet Transform (DWT) with JPEG2000 standard for image compression using Aldec Active HDL 3.5 tool. The design flow starts with specifications of the system behavior for implementation on silicon and it takes several steps before hardware implementation to meet the specifications. The software allows the analysis to ensure the designed chip satisfy the required specifications.

Key words: EDA, DWT, IDWT, image compression, thresholding and JPEG2000.

1. Introduction

This paper describes the step of designing a DWT chip using Aldec Active HDL 3.5. The rest of the paper is organized as follows: In section 2, we describe DWT algorithm. In section 3, we describe the implementation of the DWT algorithm based on MATLAB and VHDL. In section 4, we describe the simulation results of the algorithm based on MATLAB and Aldec Active HDL 3.5. The discussion is given in section 5. Finally, we give the conclusion in section 6.

2. Overview of DWT

Digital images can be compressed by eliminating redundant information present in the image, such as spatial redundancy, spectral redundancy and temporal redundancy [1]. The removal of spatial and spectral redundancy is often accomplished by transform coding, which uses some reversible linear transform to decorrelate the image data [2]**.** JPEG which has many limitations is the most commonly used image compression standard in today's world supported by ISO and ITU-T. In order to overcome all those limitations, ISO and ITU-T has come up with new image compression standard, JPEG2000.

The JPEG2000 is intended to provide a new image coding/decoding system using state of the art compression techniques, based on wavelet technology. The two important parts in the coding/decoding processes of JPEG2000 are Wavelet Transform and Arithmetic Coding [3].

In wavelet analysis, a signal is decomposed into approximation and detail which are used later to reconstruct the original signal [4]. The approximation is the high-scale, low frequency component of the signal. The detail is the low-scale, high frequency component. Most natural images have smooth color variations (low frequency components), with the fine details (high frequency components) being represented as sharp edges in between the smooth variations. Hence, the smooth variations (low frequency components) are demanding more importance than the details (high frequency components) [5]. The DWT algorithm consists of Forward DWT (FDWT) and Inverse DWT (IDWT) shown in fig. 1 and 2 respectively:

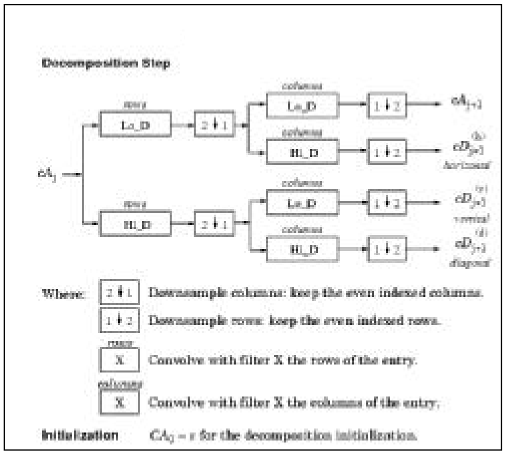

Fig. 1: Block diagram of FDWT Algorithm.

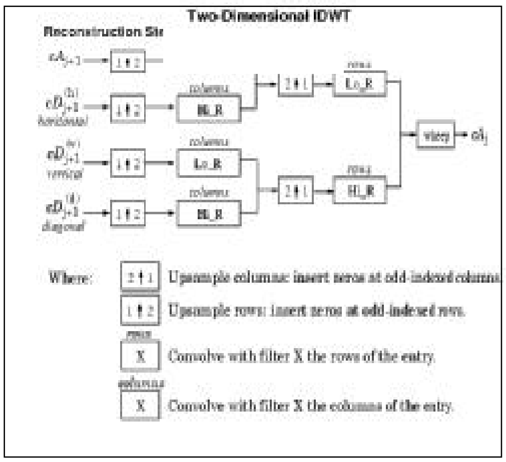

Fig. 2: Block diagram of IDWT Algorithm.

The FDWT can be performed on a signal using different types of filters such as db7, db4 or Haar. The Forward transform can be done in two ways, such as matrix multiply method and linear equations. In the FDWT, each step calculates a set of wavelet averages (approximation or smooth values) and a set of details. If a data set s_0, s_1, ... s_{N-1} contains N elements, there will be N/2 averages and N/2 coefficient values. The averages are stored in the upper half and the details are stored in the lower half of the N element array.

In the matrix multiply method [6] of the forward transform, the first average is calculated by the inner product of the signal [s_0, s_1, ... s_{N-1}] and the scaling vector, of the same size, [h_0, h_1, 0, 0,...,0]. The first detail is calculated by the inner product of the signal and the wavelet vector [g_0, g_1, 0, 0….,0].

The next average and coefficient values are calculated by shifting the scaling and wavelet vectors by two and then calculating the inner products. In the case of the wavelet transform using Haar filter, the Scaling function coefficients are $h_0 = 0.5$, $h_1 = 0.5$ and Wavelet function coefficients are $g_0 = 0.5$, $g_1 = -0.5$. The scaling and wavelet values for the forward transform are shown in the fig. 3 in matrix form.

$$\begin{array}{ccccc} h_0 & h_1 & 0 & 0 & \cdots \\ g_0 & g_1 & 0 & 0 & \cdots \\ 0 & 0 & h_0 & h_1 & \cdots \\ 0 & 0 & g_0 & g_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{array}$$

Fig. 3: Transform matrix of vector coefficients.

On the other hand, the following linear equations no (i) and (ii) can be used to calculate an average (a_i) and a detail (d_i) from an odd and even element in the input data set :

$$a_i = (s_i + s_{i+1})/2 \qquad \text{(i)}$$

$$d_i = (s_i - s_{i+1})/2 \qquad \text{(ii)}$$

After the FDWT stage, the resulting average and coefficient values can be compressed using thresholding method [7].

In the IDWT process, to get the reconstructed image, the wavelet details and averages can be used in the matrix multiply method and linear equations. For the matrix multiply method, the Scaling function coefficients are $h_0 = 1$, $h_1 = 1$ and Wavelet function coefficients are $g_0 = 1$, $g_1 = -1$.

The IDWT process can be performed using the following linear equations (iii) and (iv).

$$s_i = a_i + d_i \qquad \text{(iii)}$$

$$s_{i+1} = a_i - d_i \qquad \text{(iv)}$$

The pseudo-code for the DWT algorithm in MATLAB environment is given in the fig. 4 with the input image file “barbara.png”

```
/* Pseudo-code  for the Computation of the
DWT*/

Load Image File();/*input file is Barbara.png*/

Initialize Input matrix ();   /* Input matrix   */

Convolution with scaling vector ();/* FDWT */

Down sample by_2 ();

/*Thresholding */

Initialize threshold value ();

Calculate dead value ();

Perform thresholding on the transform
matrix();

Up sample by_2 ();

Convolution with wavelet vectors () ;
/*IDWT*/
```

Fig. 4: Pseudo-code based on MATLAB environment.

3. Implementation of DWT Algorithm

The flow for implementing the DWT algorithm in the MATLAB environment and the Aldec Active HDL 3.5 environment is shown in Fig. 5. Initially the code of the algorithm was written in PC using MATLAB because it provides powerful numerical computation and advanced visualization with an extremely easy-to-write syntax. The DWT algorithm has been tested on the "barbara.png" image file with satisfactory result. After achieving confidence on the result, we move to VHDL (VHSIC (Very High Speed Integrated Circuit) Hardware Description Language) language using Aldec Active HDL 3.5 environment.

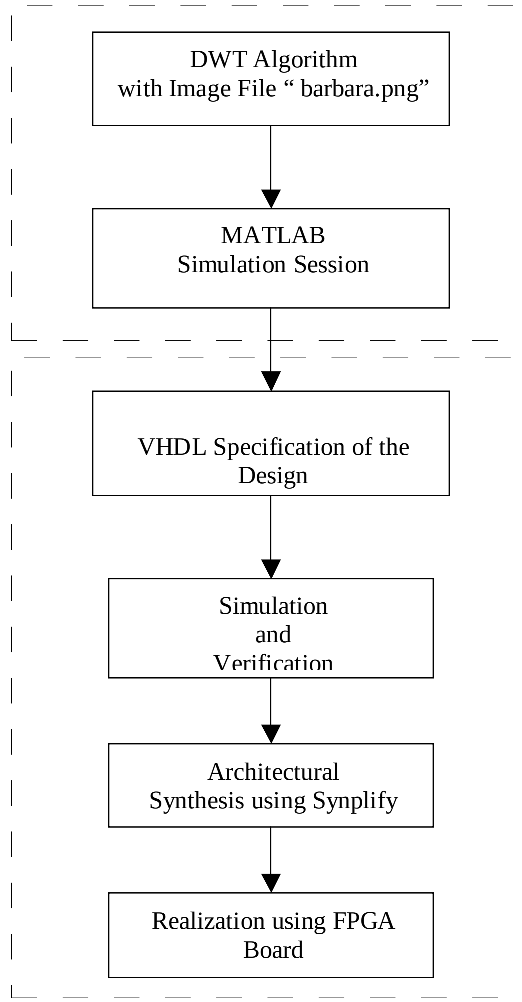

Fig. 5: Design flow of the DWT Algorithm in PC Environment.

The specification of the algorithm at the behavioral level is carried out using the VHDL. After compilation, the VHDL algorithm was simulated using Aldec Active HDL 3.5 to get the satisfactory results for real time implementation. Next, the VHDL codes will be synthesized using the synthesis tool "Synplify" which will produce "gate-level architecture" for VLSI implementation. Finally, the design codes of DWT will be downloaded into FPGA board for verifying the functionality of DWT. However, in this paper we only present the simulation results for the DWT using MATLAB and Aldec environment.

4. Simulation Results

After performing the DWT algorithm on "barbara.png" file using MATLAB environment, we were able to reconstruct image shown in Fig. 6. The few 8 random coefficients used to simulate using both MATLAB and VHDL and their simulation results are shown in table 1. As we can see, the two outputs of coefficient given by MATLAB and VHDL are almost equal. In this way, the design is tested to validate the DWT algorithm for a future implementation.

Fig. 6: MATLAB Simulation results on "barbara.png" image file.

5. Discussion

A single wavelet transform step using a matrix multiply method involves the multiplication of the signal vector by a transform matrix, which is an (N X N) operation (where N is the data size for each transform step) while the use of linear equations needs only N operations. In practice matrices are not used to calculate the wavelet transform. The matrix form of the wavelet transform is both computationally inefficient and impractical in its memory consumption. That is why, linear equations are preferred over the matrix multiply method. In order to reduce the complexity and increase computation speed, linear equations of DWT were used for VHDL implemention.

TABLE 1: COMPARISON OF MATLAB AND VHDL SIMULATION

Input coefficient	MATLAB simulation result	VHDL simulation result
1100	1100	1100
1200	1200	1200
1000	1000	1000
1200	1200	1200
1400	1400	1400
1200	1200	1200
1400	1400	1400
1200	1200	1200

6. Conclusion

In this paper, an efficient DWT algorithm with "barbara.png" input image file is simulated by MATLAB sesion and VHDL laguage using Aldec Active HDL 3.5 tool. The simulation results for both environments are satisfactory for VLSI implementation.

7. References

[1] "Compressing still and moving images with wavelets" by M L Hilton, B D Jawerth and Ayan Sengupta, published in the journal "Multimedia Systems" vol – 2 no-3,1994.

[2] Rabbani M, Jones P (1991) Digital Image Compression Techniques.(SPIE tutorial texts in optical engineering, vol TT7) SPIE press, Washington.

[3] JPEG official website, www.jpeg.org/jpeg2000.html

[4] MATLAB Wavelet Toolbox.

[5] Tutorial by Satish http://www.debugmode.com/imagecmp/

[6] Bear Products International website http://www.bearcave.com/misl/misl_tech/wavelets/daubechies

[7] "Image Compression Using The Haar Wavelet Transform" by Colm Mulcahy, Spelman College Science & Mathematics Journal, Vol 1, No 1, April 1997, 22-31