

Verilog Implementation of Image Compression Using Discrete Wavelet Transform

P.Sreekanth

CVR College of Engineering

Email: sreekanth.isoft@gmail.com

Abstract: As our use of computer continues to grow, need for storing large amount of data efficiently increases day by day. Image compression has become a necessary step for making better use of available storage and transmission bandwidth. This paper presents image compression using discrete wavelet transform(DWT) implementation using VERILOG. Decompression algorithm is implemented using inverse wavelet transform (IDWT). Here we used sub-band coding to implement wavelet transform. By using DWT we get high compression ratios. The results of VERILOG are discussed in this paper.

Keywords: Image compression, Discrete Wavelet Transform (DWT), Inverse Discrete Wavelet Transform (IDWT), Compression Ratio(CR) and sub-band coding

I. INTRODUCTION

Data compression is the process of converting data files into smaller files for efficiency of storage and transmission. As one of the enabling technologies of the multimedia revolution, data compression is a key to rapid progress being made in information technology. It would not be practical to put images, audio and video alone on websites without compression. Data compression algorithms are used in those standards to reduce the number of bits required to represent an image or video sequence. Compression is necessary and essential method for creating image files with manageable and transmittable sizes. In order to be useful, a compression algorithm has a corresponding decompression algorithm that, given the compressed file, reproduces the original file. There are many types of compression algorithms developed. These algorithms fall into two broad types: lossless algorithms and lossy algorithms. A lossless algorithm reproduces the original exactly. A lossy algorithm loses some data. Text compression must be lossless because a very small difference can result in statements with totally different meanings. There are also many situations where loss may be either unnoticeable or acceptable. In image compression, for example the exact reconstruction of each sample of the image is not necessary. Depending on the quality required of each sample of the image, varying amounts of loss of information can be accepted.

Since the input signal (e.g., a digital image) is processed by a digital computing machine, it is prudent to define the discrete version of the wavelet transform. To define the wavelet in terms of discrete values of the dilation and translation parameters ‘a’ and ‘b’ instead of being continuous, make *a* and *b* discrete.

$$a = a_0^m, \quad b = nb_0 a_0^m \quad (1)$$

Where *m* and *n* are integers. Substituting *a* and *b* in Eq. 3.1 by Eq. 4.1 the discrete wavelets can be represented by Eq.(4.2).

$$\Psi_{m,n}(t) = a_0^{-m/2} \Psi(a_0^{-m}t - nb_0) \quad (2)$$

In general, the wavelet coefficients for function *f* (*t*) are given by

$$C_{m,n}(f) = a_0^{-m/2} \int f(t) \Psi(a_0^{-m}t - nb_0) dt \quad (3)$$

II. ONE DIMENSIONAL DWT ARCHITECTURE

The discrete wavelets transform (DWT), which transforms a discrete time signal to a discrete wavelet representation. The 1-D DWT is given as the inner product of the signal *f* (*t*) being transformed with each of the discrete basis functions. [2], [7]

$$C_{m,n} = \langle f(t) \Psi_{m,n}(t) \rangle \quad (4)$$

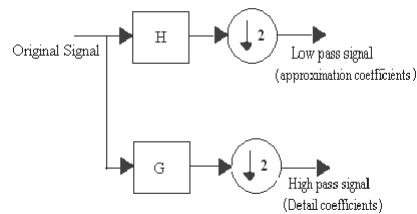


Figure 1: One Dimensional DWT

The generic form of 1 D DWT is shown in above figure1. Here the discrete signal is passed through low pass and high pass filters H and G then down sampled by factor 2 completes forward wavelet transform [6], [7] . Here the low pass filtering and down sampling means averaging operation. Low pass signal contains approximation coefficients which contain important information of original signal. High pass filtering and down sampling means differencing operation. High pass signal consists of detail coefficients which contain very less information of original signal.

The low pass filter (H) and high pass filter (G) in one dimensional forward dwt combined and it called as analyze filter bank [6], [7].

III. ONE DIMENSIONAL IDWT ARCHITECTURE

The 1-D Inverse DWT given as

$$f(t) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} C_{m,n}(f) \psi_{m,n}(t) \tag{5}$$

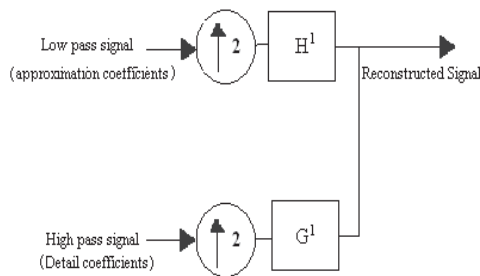


Figure 2 : One Dimensional IDWT

The generic form of 1 D IDWT is shown in figure 2. The low pass filter (H1) and high pass filter (G1) in one dimensional Inverse DWT combined and it is called as decomposition filter bank [2], [7].

Here input is output of one dimensional DWT. The low pass signal and high pass signal passes through up sampler first followed by decomposition filter bank next. By combining the outputs of low pass filter (H1) and high pass filter (G1), we can get original signal back. [8]

IV.DWT FOR TWO – DIMENSIONAL SIGNAL

The two-dimensional extension of DWT is essential for transformation of two-dimensional signals, such as a digital image. A two-dimensional digital signal can be represented by a two-dimensional array $x(m, n)$ with m rows and n columns, where m and n are nonnegative integers. The simple approach for two dimensional implementation of the DWT is to perform the one-dimensional DWT row-wise to produce an intermediate result and then perform the same one-dimensional DWT column-wise on this intermediate result to produce the final result [11]. This is shown in Figure4. 3. [2],[3],[4].

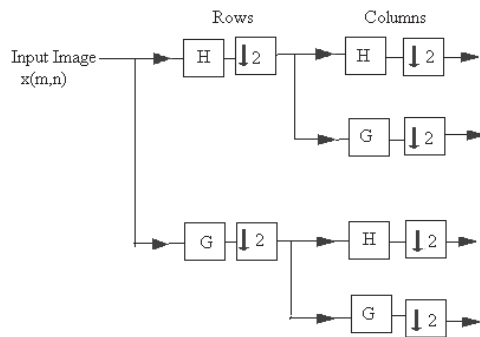


Figure 3: Level One DWT for Image

The above diagram shows forward wavelet transform of level 1 [8]. In this diagram ‘H’ represents low pass filter, ‘G’ represents high pass filter. The both filters are combined called as analyze filter bank. Here input is image which is 2 dimensional signals denoted by $x(m, n)$. Then apply one dimensional DWT row wise and down sampling divides input image into two bands. Low pass filtering band contains very important information which looks like image and high pass filtering part contain very less information which look like noise. Then similarly applying one dimensional DWT column wise completes the level 1 DWT for image. [3] Here input is original image and output is compressed image.

V. IDWT FOR TWO – DIMENSIONAL SIGNAL

The below figure.4 shows Inverse wavelet transform of level 1. In this diagram ‘H1’ represents low pass filter, ‘G1’ represents high pass filter. Both filters are combined called as synthesize filter bank. First we have to apply 1 D DWT column wise to compressed image then apply followed by row wise .This completes the level 1 IDWT.Input to this block is output of level 1 DWT image means compressed image. This compressed image then apply to up sampler first then followed by synthesize filter bank in column wise. This output is again given to up sampler followed by synthesis filter bank in row wise. We can interchange the order of columns and rows. The output is reconstructed image. Reconstructed image size is same as original image size.

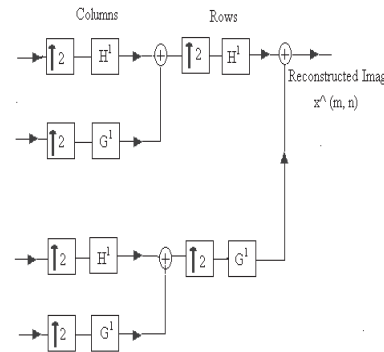


Figure 4: Level One IDWT for Image

VI. HAAR WAVELET TRANSFORM

Consider a, b are pixel values of image. Then forward HAAR wavelet transform contain two parts which are smoothing coefficients and detail coefficients, denoted by S and D respectively[9],[10]

$$\left. \begin{aligned} S &= (a + b)/2 \\ D &= (a - b)/2 \end{aligned} \right\} \tag{6}$$

We have to apply above algorithm to all pixel values of the image; it completes image compression using forward DWT. Smoothing coefficients obtained by low pass filtering and down sampling by factor of ‘2’. Low pass

filtering coefficients contain very important information of image. We can reconstruct whole image with only low pass filtering coefficients. Detail coefficients obtained by high pass filtering and down sampling by factor of '2'. High pass filtering coefficients contain very less information of image. We neglect these coefficients if want high compression.

The image reconstruction is obtained by Inverse Discrete Wavelet Transform (IDWT).

$$\left. \begin{aligned} a &= (S + D) \\ b &= (S - D) \end{aligned} \right\} \quad (7)$$

VII. VERILOG IMPLEMENTATION OF DWT

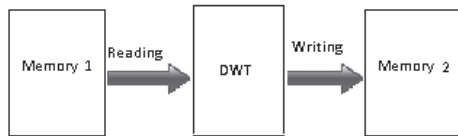


Figure 5: Block diagram for image Compression

Fig. 5 shows Block diagram of image compression using Discrete Wavelet Transform (DWT) in VERILOG. Here memory 1 is input memory where we store input image in text format. Generally image is two dimensional signal of size (m x n), where 'm' is number of rows and 'n' is number of columns. But VERILOG HDL language does not support two dimensional memories, but supports only one dimensional memory. So we need to convert two dimensional memories to one dimensional memory. By using MATLAB, input image is converted to binary text file. If input image is of size (8x8) which is two dimensional then by using MATLAB we convert it to (64x1) binary text file which is one dimensional, used for DWT operations in VERILOG. Each pixel value of the image is represented by 9 bits.

We apply above HAAR wavelet transform DWT & IDWT algorithm in between two pixels of entire image row wise followed by column wise completes compression of image using Discrete Wavelet Transform (DWT).

The outer view of DWT Block which is implemented in VERILOG is shown in fig 6. Here 'clk' is clock signal which is common for all inner blocks of DWT, 'memrd' is used for reading pixel values from memory to perform operations according to equations and 'memwr' is used for writing the resultant pixel values in memory. Here mem2 is output memory to display output pixel values.

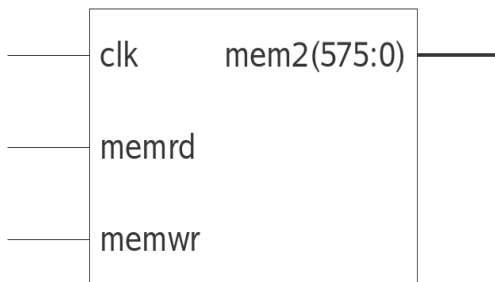


Figure 6: DWT block

DWT Block consists two sub blocks, Pixel Averaging Unit and Pixel Subtraction Unit.

A) Pixel Averaging Unit



Figure 7: Pixel Averaging Unit

Pixel averaging unit is to perform average of two pixels. In above figure, p1 and p2 are input pixel values, clk is clock signal and pout is output pixel value after performing averaging operation. Here input pixel values in form of binary and '9' bits are required to represent each pixel value.

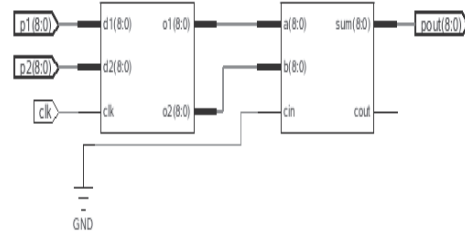


Figure 8: Inner view of Pixel Averaging Unit

Pixel average unit consists two blocks in it as shown in figure 8. The first block is 'Right shifter' which helps to perform dividing operation. And the second block is '9 bit binary' adder which helps to add two right shifted versions of pixel values. These two blocks implement

$$\left(\frac{p_1}{2} + \frac{p_2}{2} \right), \text{ same as smoothing coefficient calculation given in equation 6.}$$

B) Pixel Subtraction Unit

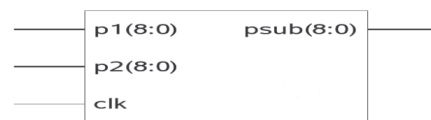


Figure 9: Pixel Subtraction Unit

Pixel subtraction unit is to perform subtraction of two pixels. In above figure, p1 and p2 are input pixel values, 'clk' is clock signal and pout is output pixel value after performing averaging operation and 'psub' is output of pixel subtraction unit. Here input pixel values in the form of binary and '9' bits are required to represent each pixel value.

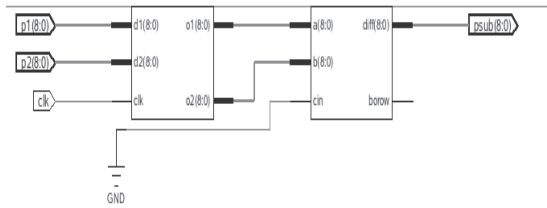


Figure 10: Inner view of Pixel Subtraction Unit

Pixel subtraction unit contain two blocks in it. Right shifter is followed by 9 bit sub tractor. The output of pixel subtractor unit is $(\frac{p_1}{2} - \frac{p_2}{2})$ where p1 and p2 are pixel values of image, same as equation 6.

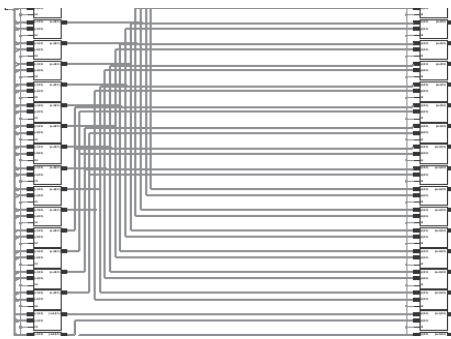


Figure 11: Connections inside the DWT block

This pixel averaging and subtraction units are connected in DWT block to complete the Forward DWT for two dimensional signals (image) for rows and columns separately. The figure.11 shows the connections of Pixel averaging and subtraction units in DWT block.

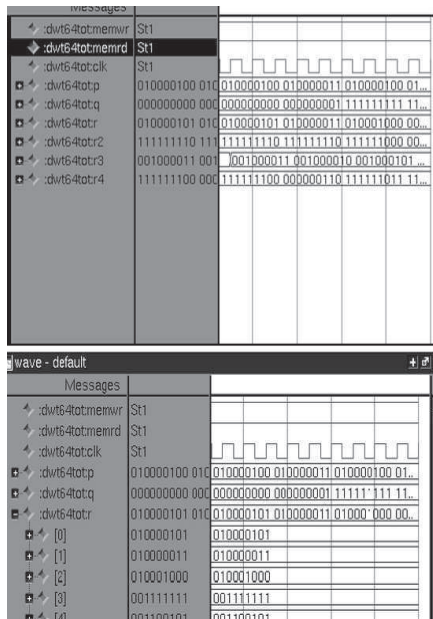


Figure 12: Simulation results of DWT block

C) IDWT Implementation In VERILOG

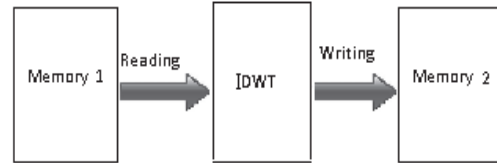


Figure 13: Block diagram for image Reconstruction

IDWT block includes one binary adder and binary sub tractor to perform operations on the output of DWT block (smoothing and detail coefficients) pixel values as given in equation 3 and 4.

The pixel values of reconstructed image are stored in binary text file in memory as one dimensional. We see the results using MATLAB by converting one dimensional memory to two dimensional memory. Have been taken (64X64) Lena image as shown fig 14(a) After applying DWT algorithm, the compressed DWT image is as shown in fig 14 (b). Then has been taken only LL part of image to reconstruct the original image it is shown in fig 14 (c). The error between original and reconstructed is as shown in fig 14 (d). Here compression ratio of 4 was obtained.

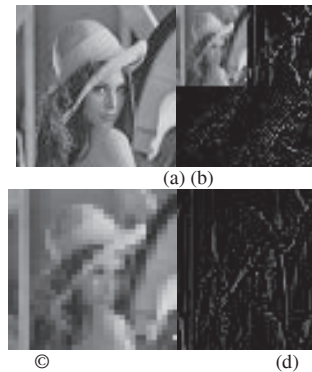


Figure 14: image compression results

Taking total parts of fig 14 (b), then the reconstructed image is shown in fig 15. Here image quality increases but compression ratio is decreased to 1.



Figure 15: reconstructed image using all parts of DWT

Similarly, The same (64 X64) DWT and IDWT algorithms to (128 X 128) image with 4 iterations has been taken. The results are shown in fig 16. Figure 16(a) shows the original image and figure 16(b) shows the reconstructed image with compression ratio of 4. The figure 16 (c) shows the error between original and reconstructed image

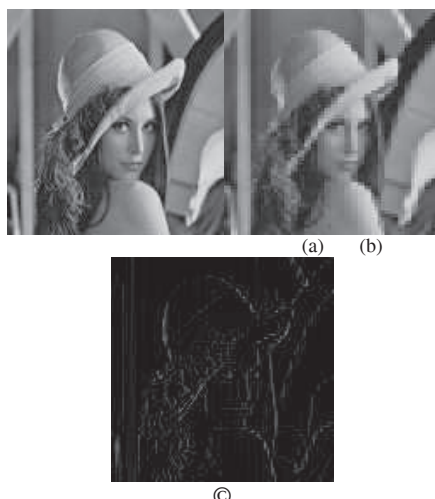


Figure 16: results for Lena (128X128)

VII CONCLUSIONS

The need for efficient ways of storing large amount of data is increasing day by day. For example, if we want to have a web page or online catalog having hundreds of images, we essentially look for some kind of image compression to have those images stored. This is because the amount of space required holding uncompressed image costs more. In this paper we discussed how image compression is done in VERILOG HDL language. Image compression and reconstruction are displayed in this paper.

We observe that as the size of image is increasing, we get better reconstructed image quality.

REFERENCES

- [1.] RatchakitSakuldee and SomkaitUdomhunsakul, "Objective Performance of Compressed Image Quality Assessments", World Academy of Science, Engineering and Technology 35 2007
- [2.] http://en.wikipedia.org/wiki/Discrete_wavelet_transform.
- [3.] Jyotishman Das, "Lossless performance of image compression using 2D DWT"
- [4.] Amir Averbuch, "wavelet Compression", School of Computer Science Tel- Aviv University
- [5.] Kamrul Hasan Talukder and Koichi Harada, "Discrete Wavelet Transform for Image Compression and A Model of Parallel Image Compression Scheme for Formal Verification", WCE 2007, July 2 - 4, 2007, London, U.K.
- [6.] Chao-Tsung Huang, Po- Chih Tseng, and Liang-Gee Chen, "Analysis and VLSI Architecture for 1-D and 2-D Discrete Wavelet Transform". IEEE Transactions on Signal Processing, vol. 53, No. 4, April 2005
- [7.] Sanjay kumarMitra, "Digital Signal Processing", Fourth edition, pp 15-56, 2010
- [8.] Bryan E. Usevitch, "A Tutorial on Modern Lossy Wavelet Image Compression: Foundations of JPEG 2000".
- [9.] P. Raviraj and M.Y. Sanavullah, "The Modified 2D-Haar Wavelet Transformation in Image Compression", Middle-East Journal of Scientific Research 2 (2): 73-78, 2007.
- [10.] Nidhi Sethi, Ram Krishna and Prof R.P. Arora, "Image Compression Using Haar Wavelet Transform", Computer Engineering and Intelligent Systems, ISSN 2222-171