Available online at http://www.ijims.com
ISSN: 2348 – 0343

# Design of DWT Module

Prabha S
VLSI Division,SENSE School, VIT University,Vellore,TN,India

## Abstract

Due to the advancements in information and communication technologies, the electronic transfer of formal documents such as business deals, etc is inevitable. Therefore a DWT based authentication method is proposed in this paper. The formal text document is watermarked by embedding the fingerprint image of the authenticated personnel. DWT is being widely considered in watermarking applications because of its efficient multi-resolution in frequency domain. The lifting based 2D-DWT with two levels of computation is implemented. The 2D-DWT can be implemented by performing 1D DWT row-wise and column wise. Hence with the help of DWT filters both document and the fingerprint image are compressed and it's divided into sub-bands and the fingerprint is watermarked in the document. The complete system is implemented on FPGA.

**Keywords-**multi-resolution, watermarking, lifting scheme, 2-D DWT

## Introduction

Every intelligent system like fingerprint eye invention, galaxy, goggle glasses, automatic trolley controller and detection of infrastructure damage caused by earthquake is solely based upon image processing, which is closely related to human and computer vision. Image manipulations such as image compression, enhancement, extraction any operation can be performed in a very precise manner with the help of DSP tool like MATLAB, Xilinx System Generator which provides in-built block sets, to increase speed, reduce occupied area and cost. These operations are all about mathematical operation on image. Henceforth first image is converted into pixel value and after that some linear mathematical transform techniques are applied so that pixel is mapped onto a set of filter coefficients. DCT and DWT are mainly used for this purpose in image processing.

DWT is preferred over other transforms due to the multi resolution characteristics[1] i.e. DWT represents image on different multi-resolution so when used in watermarking the identity of owner is embedded into a part of image. If a person publishes a paper his authentication is preserved by embedding his fingerprint as a copyright which is invisible to another person unless the document is tampered. Discrete wavelet transform is operated on host image and fingerprint image and then fingerprint image is embedded onto the lower resolution part of the host image so that it is invisible. By applying inverse discrete wavelet transform the fingerprint image is extracted from watermarked image.

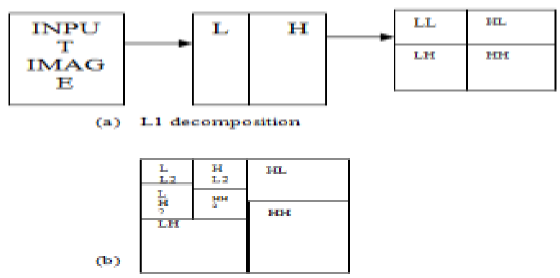
**Discrete Wavelet Transform**

A matrix containing pixel value of an image in the range from 0 to 256. Pixel values are given as an input to the filter. If the pixel values are very close to each other, it means that they are having the same intensity so it is very easy to compress that part of the image. Therefore images having large areas of uniform colour will have large redundancies and conversely images that has frequent and large changes in colour will be less redundant and harder to compress.

To embed one image into another we need to go into further information about host image i.e. image having higher resolution and part of an image having lower resolution is analysed which is called approximation sub-signal or coarse (overall information of an image).To get the approximation sub-signal low pass filter is used. For getting the lossless watermarked image the image is embedded into the part of the host image where it contains less information. Second information is the change in the image in three directions vertically horizontally and diagonally called detailed sub-signal (coefficient that effect the function at various signal). High pass filter is used to get the detailed sub-signal. Daubechies wavelet filter[3] is preferred over other filters because it gives (1) Compact support condition (2) regularity condition (3) orthogonality .For Daubechies wavelet filter for high pass filter a0=0.483, a1=0.837, a2=0.224, a3=-0.129 and for low pass filter coefficient values are just reversed i.e. g0=h3, g1=h2, g2=h2, g3=h0.where g and h are low pass filter co-efficient and high pass filter co-efficient.

*A.Two Dimensional Discrete Wavelet Transform*

In DWT at first the input samples are divided into high pass and low pass samples. Then these samples are sub-sampled and the alternate samples are taken to produce low-pass and high-pass outputs. This is called analysis [2] and the filter used in this process is called analysis filters. The inverse process, i.e the synthesis is done by up-sampling the low-pass and high-pass outputs by inserting zeros and filtered by low pass and high pass filters in order to produce the original input signal. The filters used here are called synthesis filters. The low pass sub-band is further decomposed to obtain the second level of decomposition. The 2-D DWT is done by applying 1-D DWT first row wise, to produce L and H in each row and then LL, LH, HL, HH are produced by applying 1-D DWT column wise.




L2 decomposition

Fig. 1 Decomposition of input Image

The condition for filter bank is given by

$$h(z)\tilde{h}(z^{-1}) + g(z)\tilde{g}(z^{-1}) = 2 \quad (1)$$

$$h(z)\tilde{h}(-z^{-1}) + g[(z)\tilde{g}(-z^{-1}) = 0 \quad (2)$$

where h(z) and g(z) are low pass and high pass coefficients and h(z) is given by

$$h(z) = \sum_{i=0}^{p} hi\ z^{-1} \quad (3)$$

and the polyphase equations are derived by

$$h(z) = he(z^2) + z^{-1}ho(z^2) \quad (4)$$

$$g(z) = ge(z^2) + z^{-1}go(z^2) \quad (5)$$

$$\tilde{h}(z) = \tilde{h}e(z^2) + z^{-1}\tilde{h}o(z^2) \quad (6)$$

$$\tilde{g}(z) = \tilde{g}e(z^2) + z^{-1}\tilde{g}o(z^2) \quad (7)$$

where he ,ge and ho, go represent even and odd coefficients of FIR filters h and g respectively .Using the above formulae , polyphase matrices are given as

$$\tilde{P}(z) = \begin{bmatrix} \tilde{h}e(z) & \tilde{h}o(z) \\ \tilde{g}e(z) & \tilde{g}o(z) \end{bmatrix} \quad (8)$$

$$P(z) = \begin{bmatrix} he(z) & ho(z) \\ ge(z) & go(z) \end{bmatrix} \quad (9)$$

Where $P(z)\ \tilde{P}(z^{-1})^T = I$ .When P(z) is unity, if the synthesis and analysis are equal then the filter pair orthogonal and if the filter pair is complementary, $\tilde{P}(z)$ can be factorized into upper and lower triangular matrices with the help of Euclidean algorithm.

$$\tilde{P}(z) = \left\{\prod_{i=1}^{m} \begin{bmatrix} 1 & \tilde{s}2(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \tilde{t}i(z) & 1 \end{bmatrix}\right\} \begin{bmatrix} K & 0 \\ 0 & \frac{1}{K} \end{bmatrix} \quad (10)$$

Where K and $\frac{1}{K}$ are the scaling factors and $\tilde{S}i(z)$ and $\tilde{t}i(z)$ are Laurent polynomials of lower orders.Thus the computation of upper triangular matrix is called as primal lifting or update and that of lower triangular matrix is called dual lifting or predict[3].

*B.Lifting Based DWT*

The lifting-based DWT is preferred over DWT because of the usage of less memory due to in-place computation, which uses limited on-chip memory. It also offers lossless image compression, due to integer to integer transformation which helps in exact reconstruction of the original image. The input image size is of size N X N and J and K are the number of stages of high pass and low pass filters with T latency cycles.

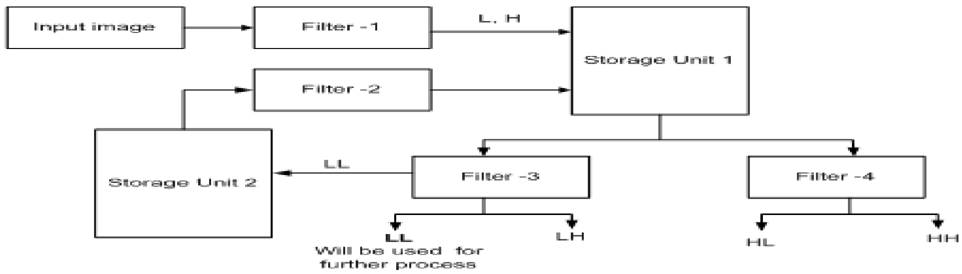



Fig. 2 Architecture for L2-2D-DWT

The latency of analysis and synthesis filter is referred to as the forward-latency and reconstructive-latency.In computing 2-D DWT, along the row (LL) H and (LL) L, computations are done after $(a + 1)^{th}$ stage. The (LL) HH and (LL) HL are computed along the columns of (LL) H and similarly (LL)LH and (LL)LL are computed along the columns of (LL)L.

**Implementation of Analysis Filter**

The input image is first converted into matrix using MATLAB.The image coefficients are fed row-wise as input to the filter 1, where the high pass(H) and low pass(L) coefficients are computed and stored in the storage unit 1.(LL)L and (LL)H coefficients are computed along row-wise by filter2 and stored in the storage unit 1.These coefficients are fetched from the storage unit 1 by filter 3 to compute (LL)LL and (LL)HL components along the row and by filter 4 to compute (LL)LH and (LL)HH are computed along the column. The first level of decomposition is done by polyphase decomposition[4] where the coefficients LH, HH, LL, HL are produced. The dataflow for this is described in table 1,

TABLE 1:Dataflow for first level of decomposition

| CLK | SW | IN | ODD | EVEN | OUT |
|---|---|---|---|---|---|
| 0 | 0 | $x[0]$ | $a_1\,x[0]$ | | |
| 1 | 1 | $x[1]$ | | $a_0\,x[1]$ | $a_0x[1] + a_1x[0]$ |
| 2 | 0 | $x[2]$ | $a_1x[2] + a_3x[0]$ | | |
| 3 | 1 | $x[3]$ | | $a_0x[3] + a_2x[1]$ | $a_0x[3] + a_1x[2] + a_2x[1] + a_3x[0]$ |
| 4 | 0 | $x[4]$ | $a_1x[4] + a_3x[2]$ | | |
| 5 | 1 | $x[5]$ | | $a_0x[5] + a_2x[3]$ | $a_0x[5] + a_1x[4] + a_2x[3] + a_3x[2]$ |
| 6 | 0 | $x[6]$ | $a_1x[6] + a_3x[4$ | | |
| 7 | 1 | $x[7]$ | | $a_0x[7] + a_2x[5$ | $a_0x[7] + a_1x[6] + a_2x[5] + a_3x[4]$ |
| 8 | 0 | $x[8]$ | $a_1x[8] + a_3x[6$ | | |
| 9 | 1 | $x[9]$ | | $a_0x[9] + a_2x[7$ | $a_0x[9] + a_1x[8] + a_2x[7] + a_3x[6]$ |

The second level of decomposition is done by coefficient folding technique where the input given is LL to compute (LL)LH, (LL)HH,(LL)HL,(LL)HH coefficients.The dataflow for the second level of decomposition is given by table[2].

TABLE2: Dataflow for second level of decomposition

| CLK | SW | IN | ODD | EVEN | OUT |
|---|---|---|---|---|---|
| 0 | 0 | $x[0]$ | $a_3\,x[0]$ | $a_1\,x[0]$ | |
| 1 | 1 | $x[1]$ | $a_2x[1] + a_3x[$ | $a_0x[1] + a_1x[0$ | $a_0x[1] + a_1x[0]$ |
| 2 | 0 | $x[2]$ | $a_3\,x[2]$ | $a_1x[2] + a_2x[1$ | |
| 3 | 1 | $x[3]$ | $a_2x[3] + a_3x[$ | $a_0x[3] + a_1x[2$ | $a_0x[3] + a_1x[2] + a_2x[1] + a_3x[0]$ |
| 4 | 0 | $x[4]$ | $a_3\,x[4]$ | $a_1x[4] + a_2x[3$ | |

| | | | | | |
|---|---|---|---|---|---|
| 5 | 1 | $x[5]$ | $a_2x[5] + a_3x[$ | $a_0x[5] + a_1x[4$ | $a_0x[5] + a_1x[4] + a_2x[3] + a_3x[2]$ |
| 6 | 0 | $x[6]$ | $a_3\, x[6]$ | $a_1x[6] + a_2x[5$ | |
| 7 | 1 | $x[7]$ | $a_2x[7] + a_3x[$ | $a_0x[7] + a_1x[6$ | $a_0x[7] + a_1x[6] + a_2x[5] + a_3x[4]$ |
| 8 | 0 | $x[8]$ | $a_3\, x[8]$ | $a_1x[6] + a_2x[5$ ) | |
| 9 | 1 | $x[9]$ | $a_2x[9] + a_3x[$ | $a_0x[9] + a_1x[8$ | $a_0x[9] + a_1x[8] + a_2x[7] + a_3x[6]$ |

**Implementation of Synthesis Filter**

In the synthesis filter implementation, the reverse process of analysis filter i.e. inverse discrete wavelet transform, is performed in order to obtain the input image x[n].At first the coefficients (LL)LL and (LL)HH are fed into the column filter 1 which computes (L'L')H' by reading the input coefficients twice. Similarly (L'L')L' coefficient is computed by column filter2 by feeding (LL)LH and (LL)LL. The coefficients (L'L')H' and (L'L')L' are stored in the synthesis storage unit 1 which is fetched by the row filters ,filter 3 and filter 4.The row filters reads the data from the storage unit1 at the same time to produce (L'L') coefficients. Thus the output is scheduled in such a way that 2 x'[n] are computed every two cycles, so that the reconstructed output does not undergo any buffering or data conversion. If the complexity of the control unit is reduced, then the reconstruction-latency is increased and vice versa.

## Implementation and Results

The two dimensional DWT was performed using the built in block sets of simulink.The bi-orthogonal filter was used for obtaining the low pass and high pass co-efficients and the results were analysed. The polyphase decomposition and coefficient folding techniques were implemented using verilog and the time and area consumption was studied.And a comparison of hardware datapath and number of cycles is done with the references[5],[6],[7],[8]and [9] is tabulated in table[3].

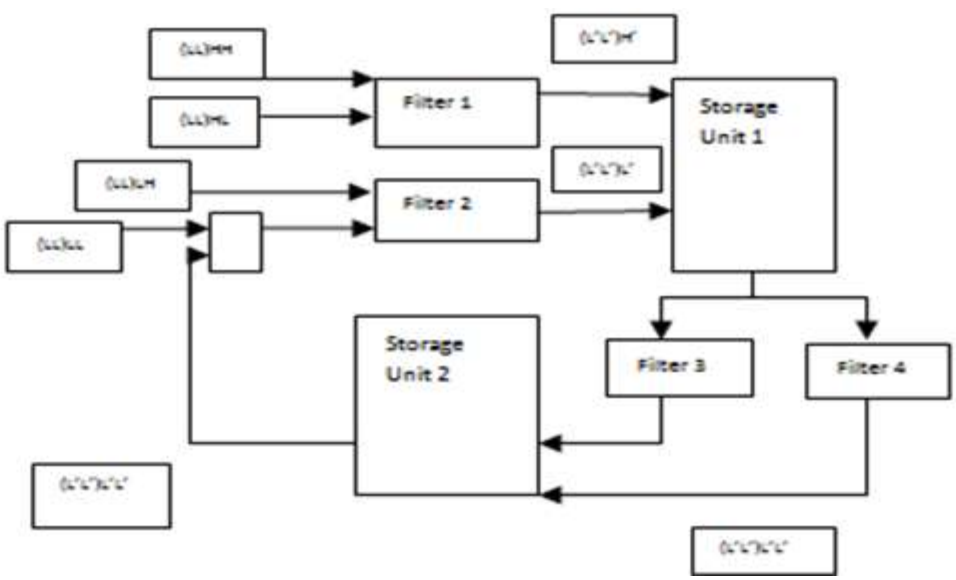



Fig 3. Architecture for 2-D Inverse DWT

TABLE3.Comparison of architectures for 2D DWT

| Architecture | Hardware Data path | No of cycles |
|---|---|---|
| (4,2) filter[5] | 10mult,6mux,<br>4 shifters,4data reg,8 control reg,8 adders | Not provided |
| Generalized<br>(9,7) [6] | 16 reg,<br>4 mult,<br>8 adders,<br>4 shifters,<br>2 scaling mult | $4/3\ N^2 + 2L$ |
| Recursive(9,7)[7] | 14 reg,4 scaling mult,8 mult,8 adders | $N^2+N+2L$ |
| Folded RPA(9,7)[8] | 2adders,mux,regs,9 mul | Similar to [9] |
| Ours | 5 adders,6 multipliers,6 registers | N/2 |

## Conclusion and Future Work

An efficient architecture for 2-D DWT is designed and its output is analysed using MATLAB and Verilog.Further the image coefficients obtained from the 2-D DWT of finger print and the text document, is added together using image addition and the watermarking is performed. The coefficients added can be extracted by performing inverse discrete wavelet transform on the watermarked image.

## Acknowledgment

The author sincerely thanks Prof.V.Arunachalam, Assistant Professor(Senior Grade) of VLSI Division,VIT University, for his invaluable guidance and support to finish this project.

## References

[1]T.Acharya and C.Chakrabarti, A Survey on Lifting-based Discrete Wavelet Transform Architectures", Journal of VLSI Signal Processing, Netherlands, 2006,42 321–339.

[2]C.Chakrabarti and C.Mumford,Efficient realizations of analysis and synthesis filters based on the 2-D discrete wavelet transform, in Proc. IEEE ICASSP, pp. 3256-3255.

[3] I. Daubechies and W. Sweldens, Factoring Wavelet Transforms into Lifting Schemes,*The J. of Fourier Analysis and Applications*, 1998,vol. 4, pp. 247–269.

[4]U.Bhanu and A.Chiambuchelvan,Efficient VLSI Architecture for Discrete Wavelet transform, International Journal of Computer Science Issues, Special Issue, ICVCI-2011, November 2011, Vol. 1, Issue 1.

[5] M. Ferretti and D. Rizzo, "A Parallel Architecture for the 2-D Discrete Wavelet Transform with Integer Lifting Scheme," *Journal of VLSI Signal Processing*, 2001,vol. 28, pp.165–185.

[6] K. Andra, C. Chakrabarti, and T. Acharya, "A VLSI Architecture for Lifting-Based Forward and InverseWavelet Transform,"*IEEE Trans. of Signal Processing*, 2002,vol. 50, no. 4, pp. 966–977.

[7] H. Liao, M.K. Mandal, and B.F. Cockburn, "Efficient Architectures for 1-D and 2-D Lifting-BasedWavelet Transform," *IEEE Transactions on Signal Processing*, 2004,vol. 52, no. 5, pp.1315–1326.

[8] G.C. Jung, D.Y. Jin, and S.M. Park, "An Efficient Line Based VLSI Architecture for 2-D Lifting DWT," in *The 47th IEEE International Midwest Symposium on Circuits and Systems*, 2004.

[9] H. Liao, M.K. Mandal, and B.F. Cockburn, "Novel Architectures for Lifting-Based Discrete Wavelet Transform,"in *Electronics Letters*, 2002,vol. 38, no. 18, pp. 1010–1012.