International Journal of Multimedia and Ubiquitous Engineering
Vol. 12, No. 7 (2017), pp. 31-42
http//dx.doi.org/10.14257/ijmue.2017.12.7.04

Performance Analysis of Modified Architecture of DA-DWT and Lifting based Scheme DWT for Image Compression.

Chetan H[1] and Indumathi G[2]

Research Scholar[1], Visveswarayya Technological University, Belgaum, India
Professor and HOD[2], Cambridge Institute of Technology, Bengaluru

Abstract

The purpose of this study is performance analysis of modified DA DWT1- architecture and Advance Lifting scheme architecture for Image Compression techniques. The information of the form image or video are transmitted as an array of data in terms of signal. Due to limited channel bandwidth, the data is compressed which in return reduces the quality of the image. An algorithmic concept of encoding information is given by wavelets in a manner that is layered according to level of detail. The analysis of this implementation includes speed optimization, accuracy, and power reduction. This study uses modified DA and optimized lifting based scheme, and architectures are modelled using digital systems, which is used for studying different performance on compressed image data. The study was done using signal simulation tool and VLSI cad tools. By implementing the proposed algorithm and modelling the architecture for image compression using DWT, we analyzed the timing wrt clock speed, area consumed and power consumed by both the architectures. Our study shows higher speed can be achieved by using DWT and better encoders for image compression and system can be modelled using digital systems, the study can be optimized to any further extent.

Keywords*: IDWT, DWT, Lifting algorithm, Low power, compression, DA- Distributive Arithmetic*

1. Introduction

Today's electronic equipment comes with user friendly interfaces such as keypads and graphical displays. As images convey more information to a user, it is many of the equipment today have image displays and interfaces. Image storage on these smaller, handled devices is a challenge as they occupy huge storage space; also, image transmission requires higher bandwidth [1]. Hence most of the signal processing technologies today has dedicated hard ware that act as co-processors to compress and decompress images. Image compression is one of the major image processing techniques that is widely used in medical, automotive, consumer and military applications. Discrete wavelet transforms are the most popular transformation technique adopted for image compression. Complexity of DWT is always high due to large number of arithmetic operations [2]. This approach is termed distributed arithmetic (DA), a bit serial method of computing the inner product of two vectors with a fixed number of cycles. The original DA architecture stores all the possible binary combinations of the coefficients w[k] in a memory or lookup table. It is evident that for large values of L, the size of the memory containing the pre-computed terms grows exponentially too large to be practical [2,3]. The memory size can be reduced by dividing the single large memory (2Lwords) into m multiple smaller sized memories each of size 2k where L = m × k. The memory size can be further reduced to 2L−1 and 2L−2 by applying offset binary coding and exploiting resultant symmetries found in the contents of the memories. This technique is based on using 2's complement binary representation of data, and the data can be pre-computed and stored in LUT. As DA is a very efficient solution especially suited for LUT-based FPGA

ISSN: 1975-0080 IJMUE

architectures, many researchers put great effort in using DA to implement FIR filters in FPGA. Patrick Longa introduced the structure of the FIR filter using DA algorithm and the functions of each part [3].

[4] proposed a modified DA architecture that gradually replaces LUT requirements with multiplexer/adder pairs. But the main problem of DA is that the requirement of LUT capacity increases exponentially with the order of the filter, given that DA implementations need 2Kwords (K is the number of taps of the filter). And if K is a prime, the hardware resource consumption will cost even higher. To overcome these problems, this paper presents a hardware-efficient DA architecture. This method not only reduces the LUT size, but also modifies the structure of the filter to achieve high speed performance.

Several recently proposed DWT-based video coders have achieved coding efficiency similar to or slightly better than block-based hybrid video coders. An important recent development in wavelet-related research is the design and implementation of 2-D multi scale transforms that represent edges more efficiently than does the separable DWT. Kingsbury‟s dual-tree wavelet transform (DT-WT) [5] is an outstanding example. The 2-D DWT is an over complete transform with limited redundancy (2m:1 for m-dimensional signals). This transform has good directional selectivity and its sub band responses are approximately shift-invariant. The 2-D DWT has given superior results for image processing applications compared to the separable DWT. Selesnick and Li introduced a 3-D model of the dual-tree wavelet transform [6] and proved that it has superior motion selectivity. The major challenge to apply the 3-D DWT for video coding is it is over completeness transform with 8:1 redundancy. By choosing the real parts of the wavelet coefficients, perfect reconstruction is obtained with the motion selectivity retained. This reduces the redundancy to 4:1. To reduce the number of coefficients, Kingsbury proposed an iterative projection-based noise shaping (NS) scheme, which modifies previously chosen large coefficients to compensate for the loss of small coefficients. Wang et al., found that noise shaping applied to 3-D DWT can yield a more compact set of coefficients than from the 3-D DWT [6].

2. Background Technology

2.1. Discrete Wavelet Transform

The discrete wavelet transforms became a very versatile signal processing tool after Mallat proposed [7] the multiresolution representation of signals based on wavelet decomposition. The method of multi-resolution is to represent a function (signal) with a collection of coefficients, each of which provides information about the position as well as the frequency of the signal (function). The advantage of the DWT over Fourier transformation is that it performs multi- resolution analysis of signals with localization both in time and frequency, popularly known as time-frequency localization [7]. As a result, the DWT decomposes a digital signal into different sub bands so that the lower frequency sub bands have finer frequency resolution and coarser time resolution compared to the higher frequency sub bands. The DWT is being increasingly used for image compression due to the fact that the DWT supports features like progressive image transmission, ease of compressed image manipulation region of interest coding, etc. Figure 1 shows the basic block diagram of DWT based image compression.

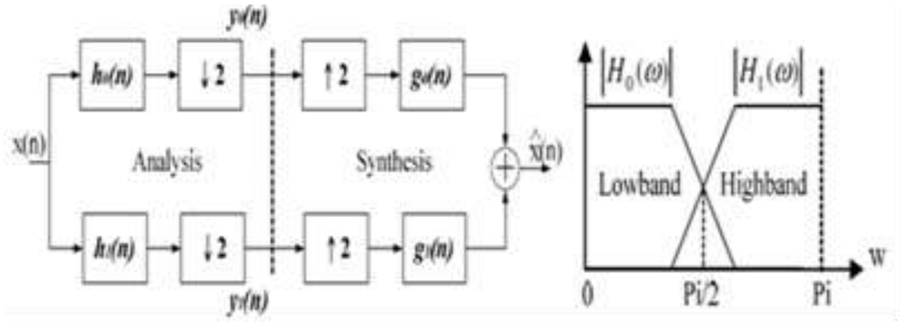


Figure 1. Basic Block Diagram of Image Compression using [7,8]

2.2 Distributive Arithmetic

Distributed Arithmetic is one of the most well-known methods of implementing FIR filters. The DA solves the computation of the inner product equation when the coefficients are pre-knowledge, as happens in FIR filters [9]. An FIR filter of length K is described as:

$$Y(n) = \sum_{k=1}^{K} h(k).x(n-k) \qquad \text{...(2.1)}$$

Where h[k] is the filter coefficient and x[k] is the input data. For the convenience of analysis, x'[k] =x [n - k] is used for modifying the equation (1) and we have:

$$Y(n) = \sum_{k=1}^{K} h(k).x(k) \qquad \text{..(2.2)}$$

The implementation of digital filters using this arithmetic is done by using registers, memory resources and a scaling accumulator. Original LUT-based DA implementation of a 4-tap (K=4) FIR filter is shown in figure. The DA architecture includes three units: the shift register unit, the DA-LUT unit, and the adder/shifter unit.

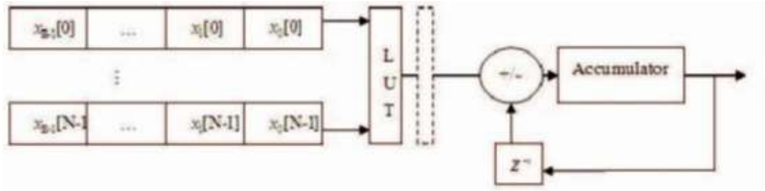


Figure 2. Basic DA MAC Architecture [9]

2.3 Lifting Scheme

The Lifting Scheme found its roots in a method to improve a given wavelet transform to obtain some specific properties. Later it was extended to a generic method to create so called `Second Generation' wavelets.

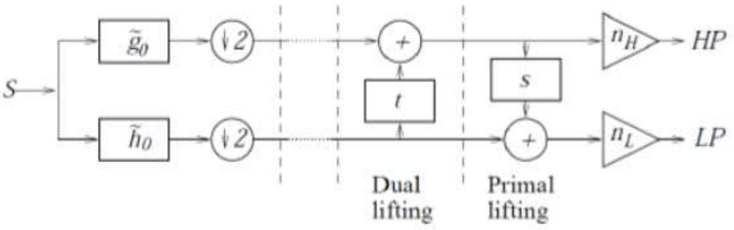


Figure 3. Lifting Scheme [9]

The Generic scheme is represented in fig 3. The filters and the successive subsampling are the “lazy’ Wavelet transform. It splits the original signal S into two sequences containing the odd and even samples. The transform is lifted to a transform with the wanted properties by using one or more of the filer operations s and t. Finally, the results are normalized by multiplying with the factors nh and nl.

3. Principle Method

The data flowing through a sequence of steps for processing is given by Data flow models. The data is moved from one step to another in sequential manner. The flow of bits is explained with the help of blocks as shown in fig below

3.1 Distributive Arithmetic Architecture

Figure 4 shows the block diagram which describes the operation of distributed arithmetic. LUT stores all the possible combinations of the sums of the coefficients in three-dimensional order. Addition operation is to implement the Filter functionality by using LUT contents. The register is used to shift the values again to arithmetic operation block. The major drawback of this method is that as the order of filter increased then look up table size also increased. Since we use three-dimensional lookup table, large amount of values is stored when compared to conventional lookup table.

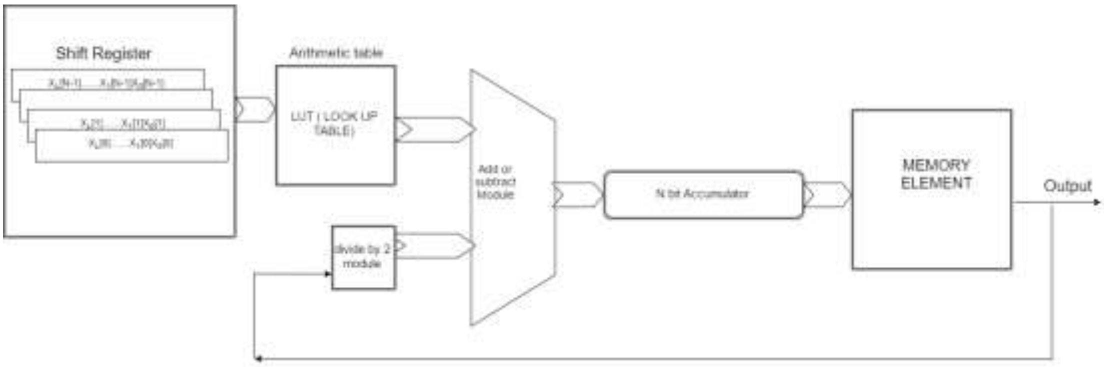


Figure 4. Proposed Architecture for Distributive Arithmetic Algorithm

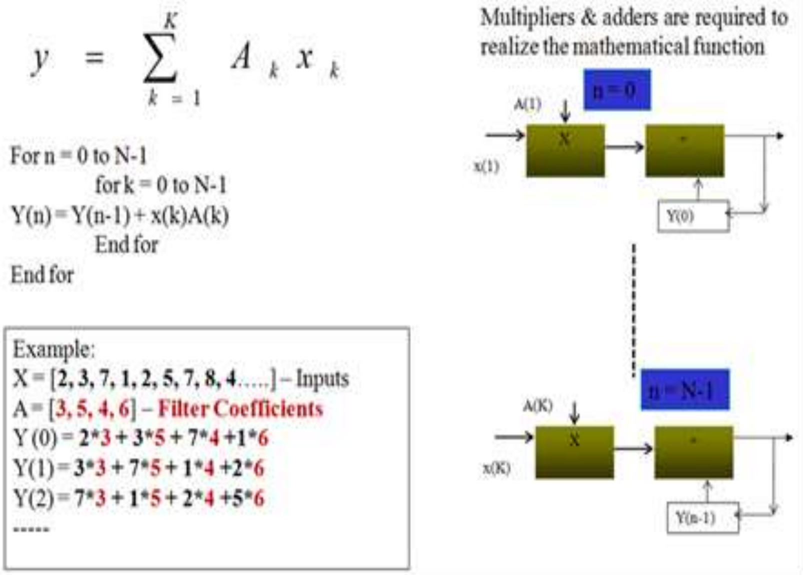
3.2 Formulation of DAA

Consider $y = \sum_{k=1}^{K} A_k x_k$ (3.1)

- Let X_k be an N-bit scaled two’s complement number. In other words,

  $| x_k | < 1$

  $x_k : \{b_{k0}, b_{k1}, b_{k2} \ldots\ldots, b_{k(N-1)}\}$
- To create a look up table, initially consider convolution operation.
- The equation for convolution operation is as shown below.
- Where X values are the inputs and A values are the filter coefficients.



$$y = \sum_{k=1}^{K} A_k x_k$$

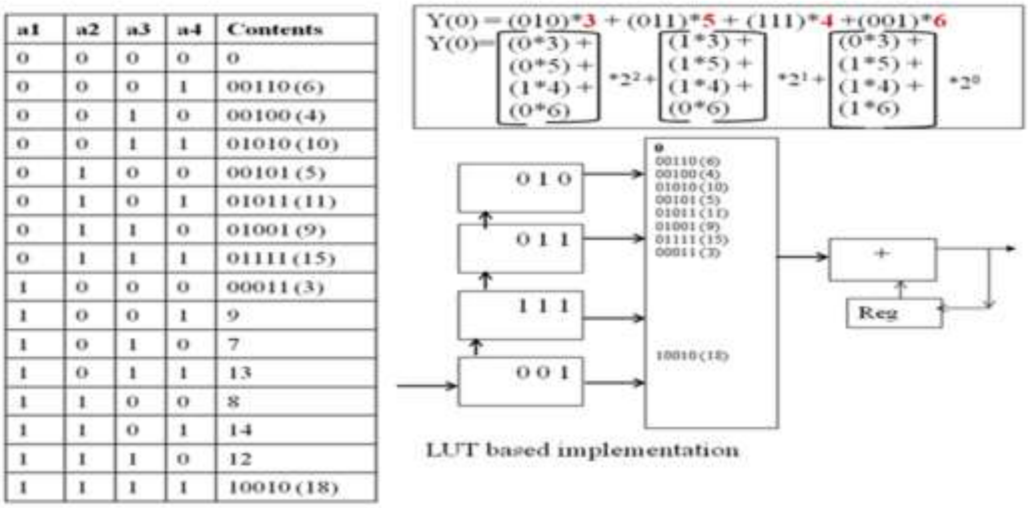
```
For n = 0 to N-1
        for k = 0 to N-1
Y(n) = Y(n-1) + x(k)A(k)
        End for
End for
```

Example:
X = [2, 3, 7, 1, 2, 5, 7, 8, 4.....] – Inputs
A = [3, 5, 4, 6] – Filter Coefficients
Y (0) = 2*3 + 3*5 + 7*4 +1*6
Y(1) = 3*3 + 7*5 + 1*4 +2*6
Y(2) = 7*3 + 1*5 + 2*4 +5*6

Multipliers & adders are required to realize the mathematical function
n = 0
A(1)
x(1)
X
Y(0)
n = N-1
A(K)
x(K)
X
Y(n-1)

Figure 6. Look Up Table Creation

- As per the formula the calculations are carried out and the input values are written in terms of binary and the weights are given.
- Based on the weights and binary values the calculated answers are entered in LUT.



a1	a2	a3	a4	Contents
0	0	0	0	0
0	0	0	1	00110 (6)
0	0	1	0	00100 (4)
0	0	1	1	01010 (10)
0	1	0	0	00101 (5)
0	1	0	1	01011 (11)
0	1	1	0	01001 (9)
0	1	1	1	01111 (15)
1	0	0	0	00011 (3)
1	0	0	1	9
1	0	1	0	7
1	0	1	1	13
1	1	0	0	8
1	1	0	1	14
1	1	1	0	12
1	1	1	1	10010 (18)

$$Y(0) = (010)*3 + (011)*5 + (111)*4 + (001)*6$$

$$Y(0) = \begin{bmatrix}(0*3) + \\ (0*5) + \\ (1*4) + \\ (0*6)\end{bmatrix} *2^2 + \begin{bmatrix}(1*3) + \\ (1*5) + \\ (1*4) + \\ (0*6)\end{bmatrix} *2^1 + \begin{bmatrix}(0*3) + \\ (1*5) + \\ (1*4) + \\ (1*6)\end{bmatrix} *2^0$$

Figure 7. LUT based Implementation Block Diagram

4. Architecture Design

4.1 Architecture of DA based DWT

Proposed architecture is developed by combining both DWT and distributive arithmetic architecture. DWT is modified with new filter coefficients as listed in architecture design. Modified DWT has been developed with new filter coefficients.

Proposed architecture is as shown in figure 8. It consists DWT system and modified DAA system. Initially input image is converted to a grey scale and resize into scalable values i.e. powers of 2. Before processing the image through filter, extra bits will be added to image to protect the information during transmission. Image must be converted into 1 dimensional signal before passing through filter because filter analysis can only be done on signals Image padding block helps to add extra bits which helps to retain the quality information during reconstruction.

After processing the information through the filter unit of DWT, the signal values will be stored into the registers. Shift registers helps to save all the data into LUT. Here ROM based DA has been proposed to increase the speed of the computation complexity. All the multiplier operation of the convolution is replaced by the LUT based design to reduce the power consumption of image compression. The proposed architecture provides better results in terms of area, delay and power.

For IDWT implementation output of quantization has given to subsystem shown below the output of subsystem is added to get two outputs then it is transpose and up-sampled by 2 so as to obtain original size then followed by same procedure as in the subsystem then the two outputs are added to get denoised image and viewed using video viewer.

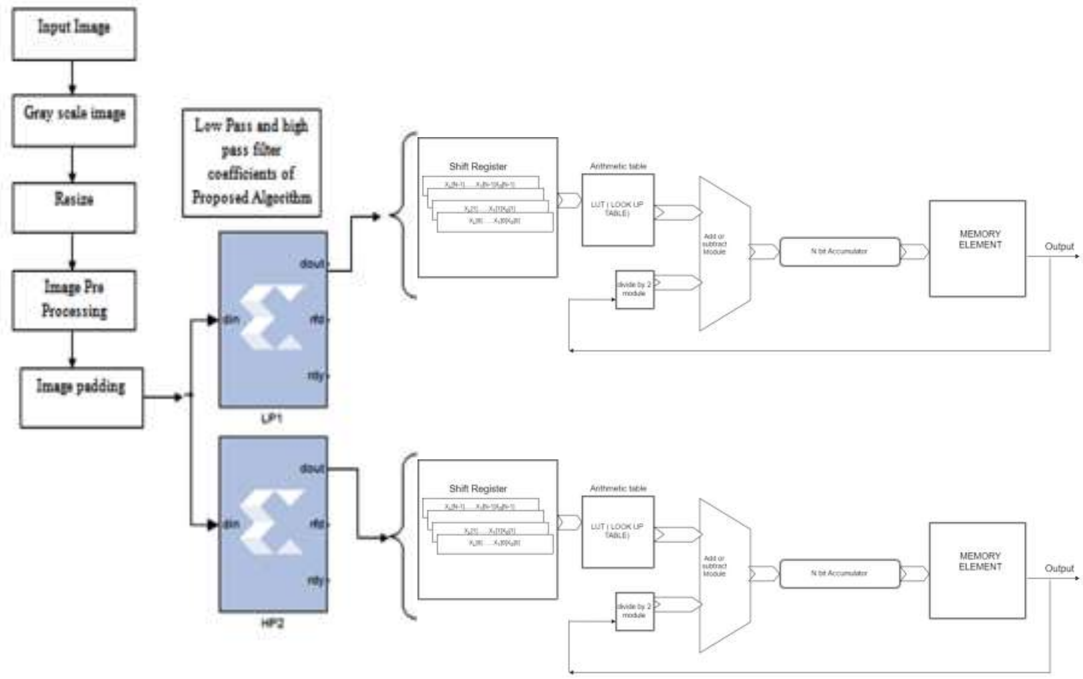


Figure 8. Proposed Architecture of Modified Distributive Arithmetic Architecture based DWT for Image Compression

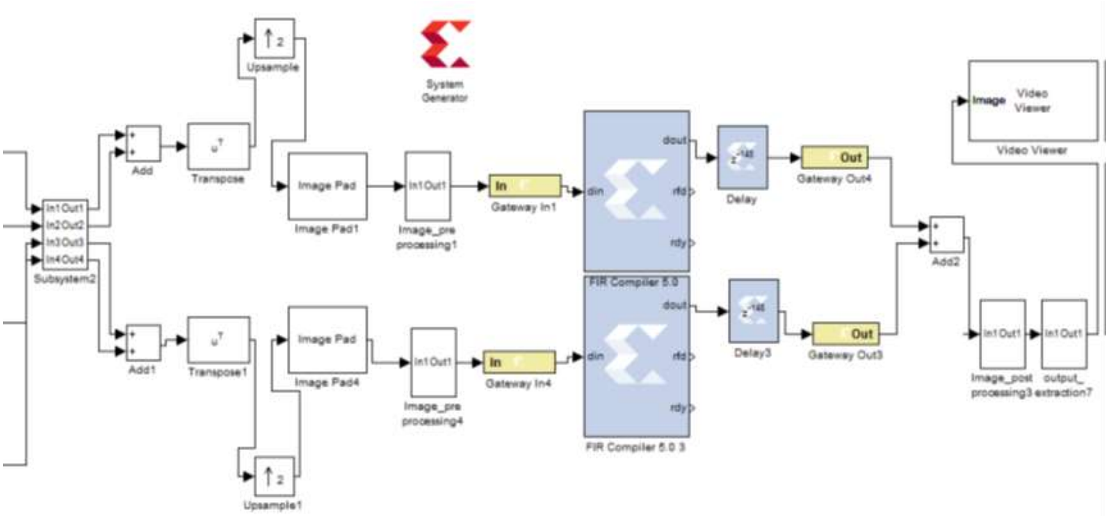


Figure 9. System Generator Model IDWT

4.2 Lifting Scheme Architecture

A lifting-based DWT architecture capable of performing filters with one lifting step, i.e., one predicts and one update step as show in Fig 5.4. The outputs are generated in an interleaved fashion. The basic principle of the lifting scheme is to factorize the poly phase matrix of a wavelet filter into a sequence of alternating upper and lower triangular matrices and a diagonal matrix. This leads to the wavelet implementation by means of banded-matrix multiplications.

Factorization consists of three steps:
1. **Split Step**: where the signal is split into even and odd points, because the maximum correlation between adjacent pixels can be utilized for the next predict step.
2. **Predict Step**: The even samples are multiplied by the predict factor and then the results are added to the odd samples to generate the detailed coefficients.
3. **Update Step**: the detailed coefficients computed by the predict steps are multiplied by the update factors and then the results are added to the even samples to get the coarse coefficients.

4.3 Mathematical Modelling of Lifting DWT

The lifting scheme algorithm can be described as Split step in this original signal, $X(n)$, is split into odd and even samples like if input $x(n) = [1\ 2\ 3\ 4\ 5\ 6\ 7\ 8]$ then $x(2n) = [2\ 4\ 6\ 8]$ then $x(2n+1) = [1\ 3\ 5\ 7]$. Lifting step This step is executed as N sub-steps into odd and even samples are filtered by the prediction and update filters, $Pn(n)$ and $Un(n)$. In Scaling step after lifting steps, a scaling coefficients K and 1/K are applied respectively to the odd and even samples in order to obtain the $(YL(i))$, and $(YH(i))$.

The lifting scheme algorithm to the (9,7) filter is follows

Forward transform

LDWT forward transform equations are

a) Split step

$Xe \leftarrow X(2n)$ Even Samples ....(4.1)

$Xo \leftarrow X(2n+1)$ Odd Samples ....(4.2)

Input will get divided into even sample and odd sample in split step then that outputs given to prediction and update in lifting step.

b) lifting step

$Y(2n+1) = X(2n+1) + a\,(X(2n) + X(2n+2))$ ……(4.3)

In lifting step of 1st equation takes odd samples from equation 4.2 split step. This step is called prediction1.

$Y(2n) = X(2n) + b\,(Y(2n-1) + Y(2n+1))$ ………(4.4)

lifting step of 2nd equation takes even samples from equation 4.1 split step. This step is called update1.

$Y'(2n+1) = Y(2n+1) + c\,(Y(2n) + Y(2n+2))$ …..(4.5)

In lifting step of 3rd equation takes odd samples from equation 4.3 1st equation output of lifting step. This step is called pridiction2.

$Y'(2n) = Y(2n) + d\,(Y'(2n-1) + Y'(2n+1))$ ….(4.6)

In lifting step of 4th equation takes even samples from equation 4.4 2nd equation output of lifting step. This step is called update2.

c) Scaling step

$Y''(2n+1) = -\,k\,Y'(2n+1)$ ………………………(4.7)

In this equation inputs are from eqn 4.5 odd samples of forward transform output.

$Y''(2n) = Y'(2n)/\,k$……………………………(4.8)

In this equation inputs from eqn (4.6) even samples of forward transform output. These outputs given to inverse transform to get back original signal. The lifting-based wavelet transform basically consists of three steps, which are called split, lifting, and scaling, respectively, these steps is done by above equations

Inverse transform

a) Scaling step

$X(2n) = k\,Y''(2n)$ ………………………………(4.9)

Input of this equation is Even samples from eqn 4.8.

$X(2n+1) = (-\,Y''(2n+1))\,/k$…………..………4.10

Input of this equation is odd samples from eqn 4.7

b) Lifting step

$X'(2n) = X(2n) - d\,(X(2n-1) + X(2n+1))$……….4.11

Inputs of this equation from scaling step of 4.9

$X'(2n+1) = X(2n+1) - c\,(X'(2n) + X'(2n+2))$…....4.12

Inputs of this equation from equation 6.3.10

$X''(2n) = X'(2n) - b\,(X'(2n-1) + X'(2n+1))$……..4.13

Input of this equation from lifting step of equation 4.11

$X''(2n+1) = X'(2n+1) - a(X''(2n) + X''(2n+2))$….4.14

Input of this equation from lifting step of equation 4.12

c) Combine step

$Xe \leftarrow X''(2n)$ Even Samples 4.15

$Xo \leftarrow X''(2n+1)$ Odd Samples 4.16

where

$a = -1.586134342, b = -0.0529801185, c = 0.882911076, d = -0.443506852,$

and $K = 1.149604398$. These fractional values are multiplied by a factor of 128 to convert them to decimal values.

Modified Mathematical Equation

$$di2 = di1 + \Upsilon\,[\,si2 + \beta d_{i-1} + \beta d_1 + s0i + 1 + \beta di + \beta d_{i+1}]$$
$$di2 = di1 + \Upsilon\,[\,si1 + s_{0i+1} + \beta\,(d_{i+1} + di2 + d1i + 1)]$$

4.4 Architecture of the Project

The lifting based scheme consists of registers and adders. The input is of 20 bits each. Whenever the input is send, the data divided into even data and odd data. The even data and odd data is stored in the temporary registers. When the reset is high the temporary register value consists of zero whenever the reset is low the input data split into the even data and odd data.

The pixel values of the image, that is, the input data will be given to this block and hence these values will be split in to even and odd pixel values. In the design, this even and odd were taken as a array which will store its pixel values in it and once all the input pixel values over, then load will be made high which represents that the system is ready for the further process.

The Low Pass Coefficients will be achieved from the addition process of multiplied output and the odd pixel value. Again, this Low Pass Coefficient will be taken and it will be multiplied with the filter coefficients. The resultant will be added with the even pixel value which gives the High Pass Coefficient. Hence all the values from even and odd array will be taken and then above said process will be carried out in order to achieve the High and Low Pass Coefficients of the image. Now these low pass coefficients and the high pass coefficients are taken as the input for the further process.

The lifting scheme coefficients are fractions and thus require fixed point or floating-point number representation. The arithmetic units such as multipliers and adders need to be designed to operate on fixed or floating-point numbers. In order to improve the throughput and reduce latency, a modified lifting scheme algorithm is used. The modified algorithm eliminates the use of fractional coefficients and thus requires integer based adders and multipliers.

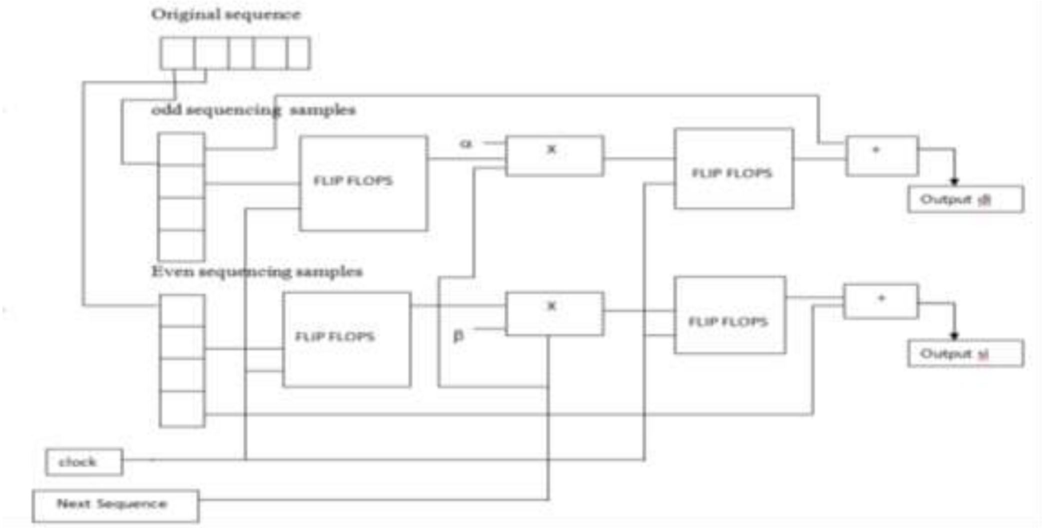


Figure 10. Proposed Architecture for Lifting based Image Compression

5. Simulation Results

5.1 Simulation Results for Image Compression

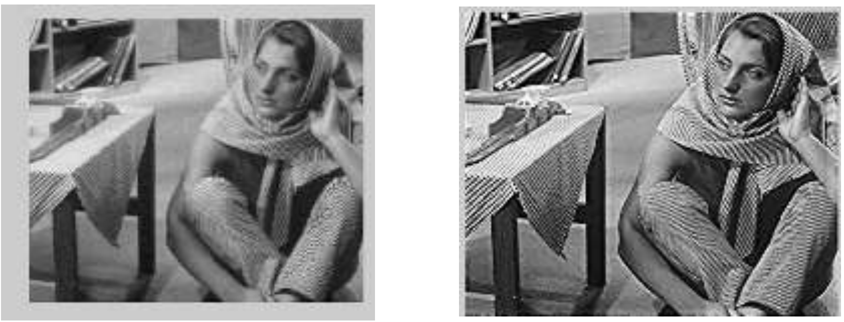


Figure 11. Simulation Results of Original Image and Reconstructed Image DAA-DWT

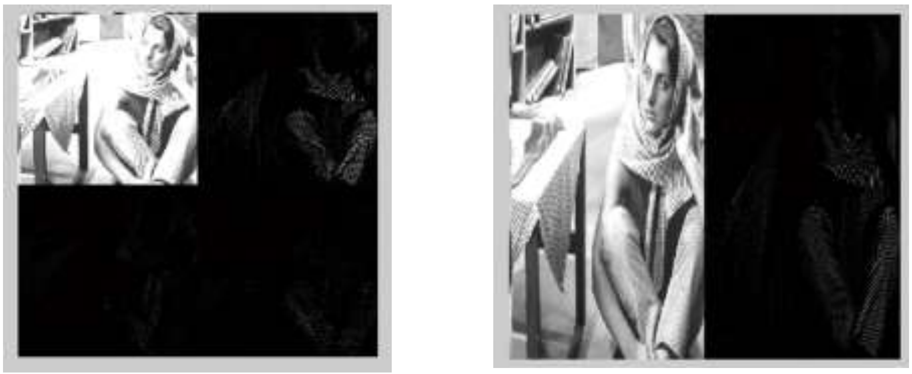


Figure 12. Simulation Results DAA -1D & 2D DWT

5.2 Simulation Results for Decompression algorithm using DAA-IDWT

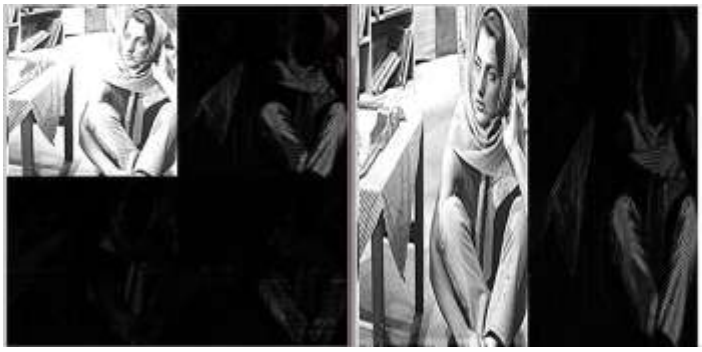
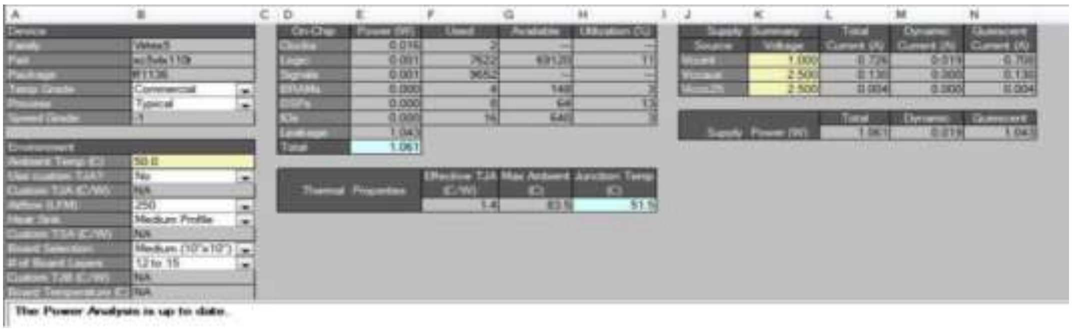


Figure 16. Simulation Result DAA-2D & 1D IDWT

Figure 17 below shows the power analysis reported from the Xpower analyzer in Xilinx 13.1. It gives an estimate of the power consumption for Xilinx devices and reports thermal information.



Device	
Family	Virtex5
Part	xc5vlx110t
Package	ff1136
Temp Grade	Commercial
Process	Typical
Speed Grade	-1

Environment	
Ambient Temp (C)	50.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	250
Heat Sink	Medium Profile
Custom TSA (C/W)	NA
Board Selection	Medium (10"x10")
# of Board Layers	12 to 15
Custom TJB (C/W)	NA
Board Temperature (C)	NA

On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.016	2	---	---
Logic	0.001	7622	69120	11
Signals	0.001	3652	---	---
BRAMs	0.000	4	148	3
DSPs	0.000	8	64	13
IOs	0.000	16	640	3
Leakage	1.043			
Total	1.061			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	1.4	83.5	51.5

Supply Source	Summary Voltage	Total Current (A)	Dynamic Current (A)	Quiescent Current (A)
Vccint	1.000	0.726	0.019	0.708
Vccaux	2.500	0.130	0.000	0.130
Vcco25	2.500	0.004	0.000	0.004

	Total	Dynamic	Quiescent
Supply Power (W)	1.061	0.019	1.043

The Power Analysis is up to date.

Figure17. Power Analysis of DAA-DWT Architecture

5.3 Lifting based (Bi-orthogonal filter output)

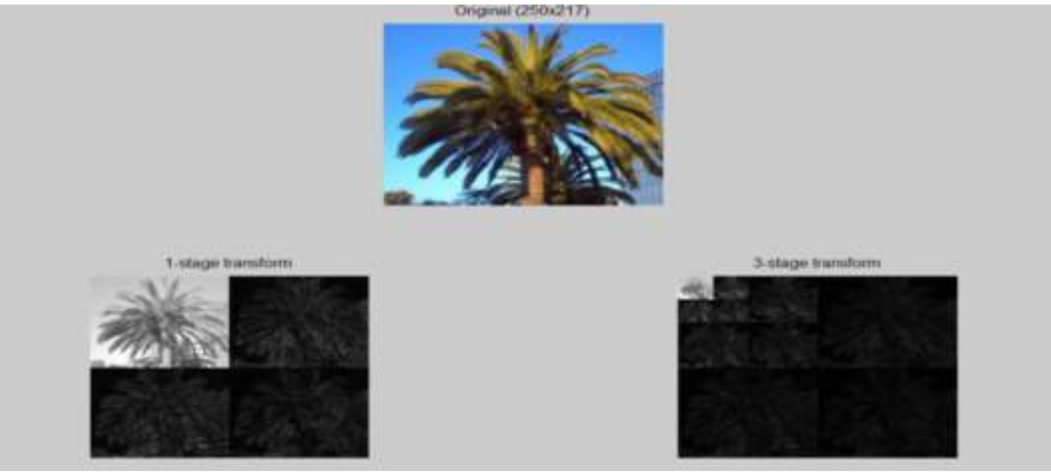
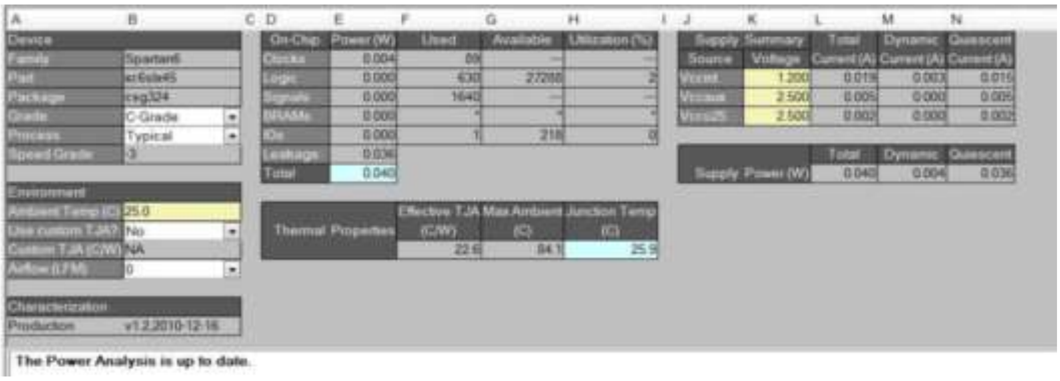


Figure 18. Wavelet Decomposition using Lifting based (Bi-orthogonal) DWT



Device	
Family	Spartan6
Part	xc6slx45
Package	csg324
Grade	C-Grade
Process	Typical
Speed Grade	-3

Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	0

Characterization	
Production	v1.2,2010-12-16

On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.004	69	---	---
Logic	0.000	630	27288	2
Signals	0.000	1640	---	---
BRAMs	0.000	[illegible]	[illegible]	[illegible]
IOs	0.000	1	218	0
Leakage	0.036			
Total	0.040			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	22.6	84.1	25.9

Supply Source	Summary Voltage	Total Current (A)	Dynamic Current (A)	Quiescent Current (A)
Vccint	1.200	0.019	0.003	0.016
Vccaux	2.500	0.005	0.000	0.005
Vcco25	2.500	0.002	0.000	0.002

	Total	Dynamic	Quiescent
Supply Power (W)	0.040	0.004	0.036

The Power Analysis is up to date.

Figure 19. Power Analysis of Bi-Orthogonal lifting based-DWT Architecture

9. Conclusion

The wavelet approximation error is less when compared to that of the DAA architecture. The 9/7 biorthogonal filter is designed which the suitable test vectors are chosen to verify the logic correctness. The verified model is synthesized using the Verilog and simulated using Modelsim. The results obtained show that the proposed design operates at maximum frequency of 107.665MHz, and consumes power less than 0.040W and the area of suitable multipliers and adders can be adopted to the design the modified design. Application such as medical, web based image could be compressed, real time video transmission can be possible with high accuracy, large bitrate and minimum bandwidth.

References

[1] M. L. Hilton, B. D. Jawerth and A. Sengupta "Compressing still and moving images with wavelets" journal of Multimedia Systems, vol. 2, no. 3, **(1994)**.

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

[3] L. Mintzer, "FIR filters with the Xilinx FPGA "FPGA '92 ACM/SIGDA Workshop on FPGAs pp. 129-134.

[4] K.K. Parhi and T. Nishitani "VLSI Architecture for Discrete Wavelet Transform", IEEE Trans. VLSI Systems, vol. 1, **(1993)**, pp. 191-202.

[5] I. W. Selesnick, R. G. Baraniuk, and N. G. Kingsbury, "The Dual-Tree Complex Wavelet Transform," 1053-5888/05, IEEE Signal Processing Magazine, **(2005)**, pp. 123-151.

[6] M. Vishwanath, R.M. Owens and MJ. Irwin, "VLSI Architecture for The Discrete Wavelet Transform", IEEE Trans. Circuits and Systems, vol. 42, **(1996)**, pp. 305-316.

[7] Third edition of the book A Wavelet Tour of Signal Processing, 3rd edition, The Sparse Way, of Stéphane Mallat.

[8] C. Chakrabarti and M. Vishwanath, "Architectures for Wavelet Transforms: A Survey", Journal of VLSI Signal Processing, Kulwer, vol. 10, (**1995)** , pp. 225-236.

[9] C. Nagabushanam, P. Raj, P. Ramachandran, "Design and implementation of Parallel and Pipelnined Distributive Arithmetic based Discrete Wavelet Transform IP core", EJSR, vol. 35, no. 3, (**2009)**, pp. 378-392,

[11] K.B. Sowmya, S. Sonali, M. N. Bhushanam, "Optimized DA based DWT-IDWT for Image Compression", International Journal of Conceptions on Electrical and Electronics Engineering, vol. 1, no. 1, **(2013)**, pp. 2345 – 9603.

[12] T. Vijayakumar & S. Ramachandran, "Design and FPGA Implementation of High Speed DWT-IDWT Architecture with Pipelined SPIHT Architecture for Image Compression," Global Journal of Computer Science and Technology: Graphics & Vision Volume 14 Issue 1 Version 1.0 Year 2014 Type: Double Blind Peer Reviewed International Research Journal Publisher: Global Journals Inc., (**2014)**.

[13] M. Goparaju, S. Mohan, "Design & Implementation of DWT – IDWT Algorithm for Image Compression by using FPGA", International Journal of Scientific and Research Publications, vol. 3, no. 3, **(2013)**.

[14] Z. Guangjun, C. Lizhi and C. Huowang, "A Simple 9/7-TAP Wavelet filter based on Lifting Scheme", IEEE transaction, **(2001)** , pp. 349-352.

[15] C. Jiazhong, G. Weixue J. Zengwei, X. Tao, L. Hefei, C. Changnian, W. Xian, "A New Design Method of 9-7 Biorthogonal Filter Banks Based on Odd Harmonic Function", Circuits Syst Signal Process (2012) 31:1245–1255, Received: 8 March 2011 / Revised: 20 October 2011 / Published online: 16 November 2011 © Springer Science+Business Media, LLC **(2011).**

[16] B. Anju, S. Manimurugan, "An Approach to Medical Image Compression Using Filters Based on Lifting Scheme," IOSR Journal of VLSI and Signal Processing (IOSR-JVSP), vol. 1, no. 2, **(2012)**, pp. 09-16.

[17] H. Chetan, G. Indumathi, "VLSI Implementation of Low Power and High Speed Architecture of DWT-IDWT using Lifting based Algorithm", Indian Journal of Science and Technology, vol. 10, no. 2, **(2017)**.

[18] R. Pathak, A. Katariya, G.S. Tomar, K.K. Prajapati "A DWT Based Multilevel Digital Watermarking Technique for Images", International Journal of Signal and Image Processing, vol. 2, no. 2, **(2011)**, pp-36-41.