An Efficient VLSI Architecture for 3D DWT Using Lifting Scheme

A Karthikeyan
Assistant Professor, ECE
SNS College of Technology
Coimbatore, India
karthikeyan1686@gmail.com

P Saranya
Assistant Professor, ECE
Sri Guru Institute of Technology
Coimbatore, India
digisaran@gmail.com

N Jayashree
Assistant Professor, ECE
Sri Guru Institute of Technology
Coimbatore, India
Jai.ece@sriguru.ac.in

***Abstract*- The role of the compression is to reduce bandwidth requirements for transmission and memory requirements for storage of all forms of data as it would not be practical to put images, audio, video alone on websites without compression. The medical community has many applications in image compression often involving various types of diagnostic imaging. The use of wavelet transform is now well established due to its multi resolution and scaling property. Among the various techniques, we have used the lifting scheme, as the lifting scheme allows perfect reconstruction by its structure. The system is fully compatible with JPEG 2000 standard. The most important property of this concept seems to be the possibility of simple and fast applications into FPGA chip. It requires fever operations and provides in-place computation of the wavelet coefficients. This paper presents a method which implements 3-D lifting wavelet by FPGA. This architecture has an efficient pipeline structure to implement high-throughput processing without any on-chip memory/first in first out access. The proposed VLSI architecture is more efficient than the previous proposed architectures in terms of memory access, hardware regularity and simplicity and throughput.**

Keywords- Discrete Wavelets Transform, Very-Large-Scale integration (VLSI), high-speed, lifting scheme, multi-input / multi-output.

I. INTRODUCTION

Recent advances in medical imaging and telecommunication systems require efficient speed, resolution and real-time memory optimization with maximum hardware utilization. The 3D Discrete Wavelet Transform (DWT) is widely used method for these medical imaging systems because of perfect reconstruction property. DWT can decompose the signals into different sub bands with both time and frequency information and facilitate to arrive at high compression ratio. DWT architecture, in general, reduces the memory requirements and increases the speed of communication by breaking up the image into the blocks. A methodology for implementing lifting based DWT has been proposed because of lifting based DWT has many advantages over convolution based one. The lifting structure largely reduces the number of multiplication and accumulation where filter bank architectures can take advantage of many low power constant multiplication algorithms. FPGA is used in general in these systems due to low cost and high computing speed with reprogrammable property.

II. RELATED WORK

Wavelet transform has gained widespread acceptance in image compression research in particular. In addition, several VLSI architectures have been proposed for computing the 3D-DWT. They are mainly based on convolution scheme and lifting scheme. The lifting scheme can reduce the computational complexity by exploiting the similarities between high and low pass filters and it usually requires fewer multipliers and adders than the convolution scheme. Architecture, presented by Knowles [5], uses many large multiplexers for storing intermediate results. Wu and Chen proposed a 2-D architecture that employs a folding technique [6]. Masud et al. proposed an efficient architecture implemented by filter banks [8].

Xixin presented an efficient VLSI implementation of Distributed Architecture for DWT in order to minimize area requirement, but they have a computation time which is proportional to input data N [7].

Andra proposed a 2-D DWT architecture which composes of simple processing units and computes one stage of DWT at a time [9]. Dillen presented a combined architecture for the (5, 3) and (9, 7) transforms with minimum area [9].

III. Efficient Mimoa For 3-D Lifting-Based DWT

Lifting scheme is a relatively new method to construct wavelet bases. This scheme is called the second-generation wavelet which leads to a fast in-place implementation of the DWT. A 2-D DWT consists of horizontal filtering along the rows followed vertical filtering along the columns. Convolution-based DWT or lifting-based DWT is used to implement the filters traditionally. The lifting- based DWT is considered. The proposed architecture is suitable for any lifting-based DWT. The throughput rate is denoted by a one-dimensional variable. The odd (even) indexed data samples are represented by $x_{2n+1}(x_{2n})$.

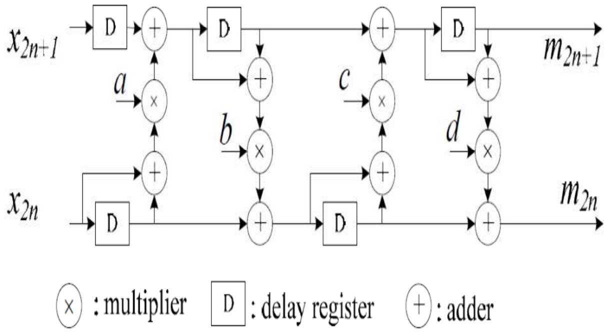


Figure 1. MIMO Architecture

In Figure 1, the intermediate values computed during lifting steps are denoted as m^{k}_{2n+1} and m^{k}_{2n} (k=0,1,2,3), and the high- and low- frequency coefficients are expressed as the sequence m_{2n+1} and m_{2n}, respectively.

A. Horizontal filtering architecture

The elements from each row are processed by one SISO module. We can accordingly get output data flow of the architecture for the horizontal filtering as shown in Figure .2, where $m_{i,j}$ denote the computational results after we apply CDF97 to the row dimension. An efficient SISO architecture is proposed by employing the fold technique. In which every single notation, it applies in Parallel Manner, which would be suitable to provide the simple and simplest. It thus promotes the much simpler architecture for the simpler implementation of its design.

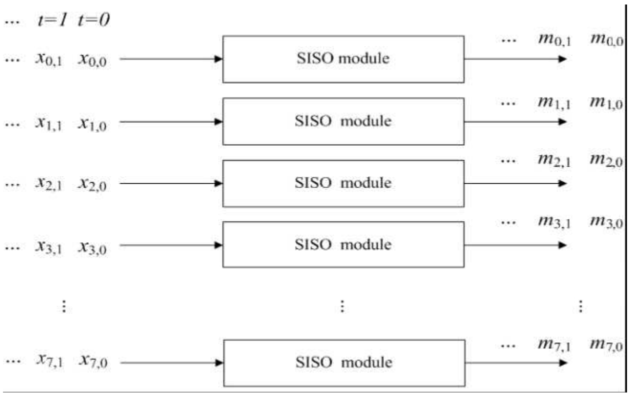


Figure 2. Architecture for Horizontal Filtering

B. Vertical filtering architecture

Eight elements(for example, $m_{0,0}$, $m_{1,0}\ldots m_{7,0}$) from each column arrive simultaneously. For example, a four-input/four-output architecture by directly mapping (3) is which process four elements from each column per clock can cycle as in Figure 3.

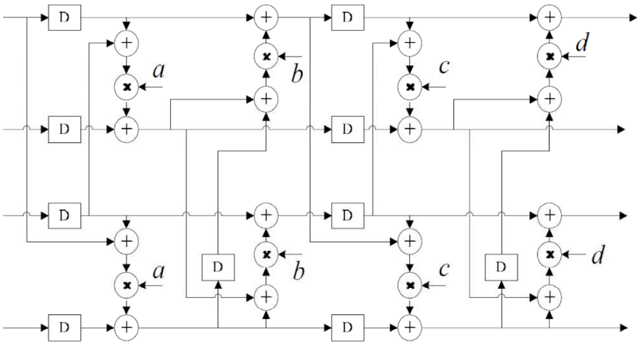


Figure 3. Architecture for vertical filtering

The time interval is measured by the number of clock cycles (ccs). We assume that the length of the image is *N* pixels, the throughput rate of proposed architecture is *M* pixels/ccs, and the computation time interval between the first pair and the second pair of the elements from one column is ccs. Therefore, the number of the input in the architecture for the vertical filtering is *M*, then the computing time interval between the first pair and the (*M*/2+1)th pair(such as $m_{0,0}$, $m_{1,0}$ and $m_{8,0}$,$m_{9,0}$) is *M*/2h ccs. On the other hand, the time interval between the first pair and the (*M*/2+1)th pair arriving at the same TITO module is equal to

$$\frac{N\ \text{pixels} \times M}{M\ \text{pixels}/\text{ccs}} = N\ \text{ccs}$$

IV. 3D Wavelet Transform Structure

The 3D DWT can be considered as a combination of three 1D DWT in the *x*, *y* and *z* directions. The preliminary work in the DWT processor design is to build 1D DWT modules, which are composed of high-pass and low-pass filters that perform a convolution of filter coefficients and input pixels. After a one-level of 3D discrete wavelet transform, the volume of image is decomposed into HHH, HHL, HLH, HLL, LHH, LHL, LLH and LLL signals as shown in figure 4.

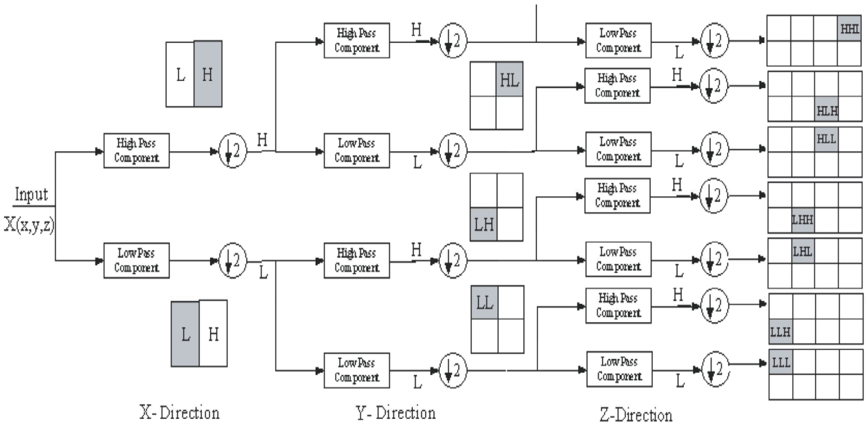


Figure 4. One Level 3D-DWT Structure

A. *Lifting scheme in 3D-DWT :*

Split the data into two sets (split phase) i.e., odd samples and even samples as shown in Figure 5. Because of the assumed smoothness of the data, we predict that the odd samples have a value that is closely related to their neighboring even samples. We use *N even* samples to predict the value of a neighboring odd value (predict phase). With a good prediction method, the chance is high that the original odd sample is in the same range as its prediction. We calculate the difference between the odd sample and its prediction and replace the odd sample with this difference.

As long as the signal is highly correlated, the newly calculated odd samples will be on the average smaller than the original one and can be represented with fewer bits. The odd half of the signal is now transformed. To transform the other half, we will have to apply the predict step on the even half as well. Because the even half is merely a sub-sampled version of the original signal, it has lost some properties that we might want to preserve. In case of images we would like to keep the intensity (mean of the samples) constant throughout different levels. The third step (update phase) updates the even samples using the newly calculated odd samples such that the desired property is preserved. Now the circle is round and we can move to the next level. We apply these three steps repeatedly on the even samples and transform each time half of the even samples, until all samples are transformed.

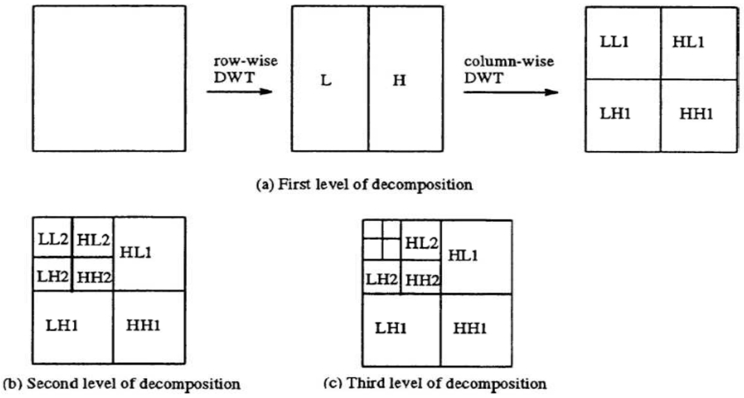

row-wise
DWT
L
H
column-wise
DWT
LL1
HL1
LH1
HH1
(a) First level of decomposition
LL2
HL2
HL1
LH2
HH2
LH1
HH1
(b) Second level of decomposition
HL2
HL1
LH2
HH2
LH1
HH1
(c) Third level of decomposition

Figure 5. Decomposition Levels For 3D DWT

These coefficients correspond to H and L respectively. Now these coefficients are passed through the 1-D processor 3 times. Here *z*-coordinate processor gives the final output as the eight subsets of original image. These coefficients are then stored in external memory in the form of binary file. For the multiple level of decomposition this binary file can be invoked iteratively to obtain further sublevels. By performing the down sampling operation before multiplication by filter co-efficients, this filter breaks the input stream up into odd and even streams at half the input rate. Note that there is a delay unit be- fore one of the down samplers that create the odd stream. One data stream would carry the inputs for one lowpass (or highpass) filter, whereas the second stream would carry the inputs for the other lowpass (or highpass) filter. Doubling the amount of registers in each MAC is the obvious solution.

B. 3D-DWT Architecture:

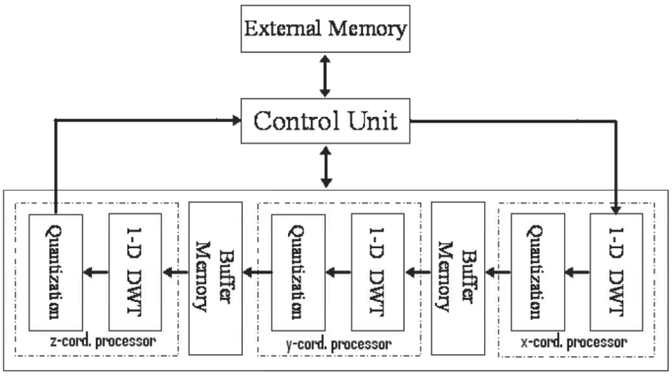


Figure 6. 3D DWT Processor Architecture

3-D DWT is simple extension of 1-D DWT. The input data in case of 1-D DWT (*x*-coordinate processor), picked from image file is in binary format. Once it generates the output set of coefficients it stores the result into buffer memory.

V. Experimental Result

The input image used for the compression is shown in Figure 7.

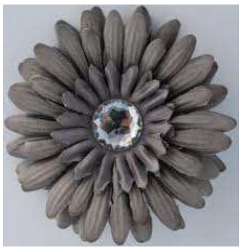
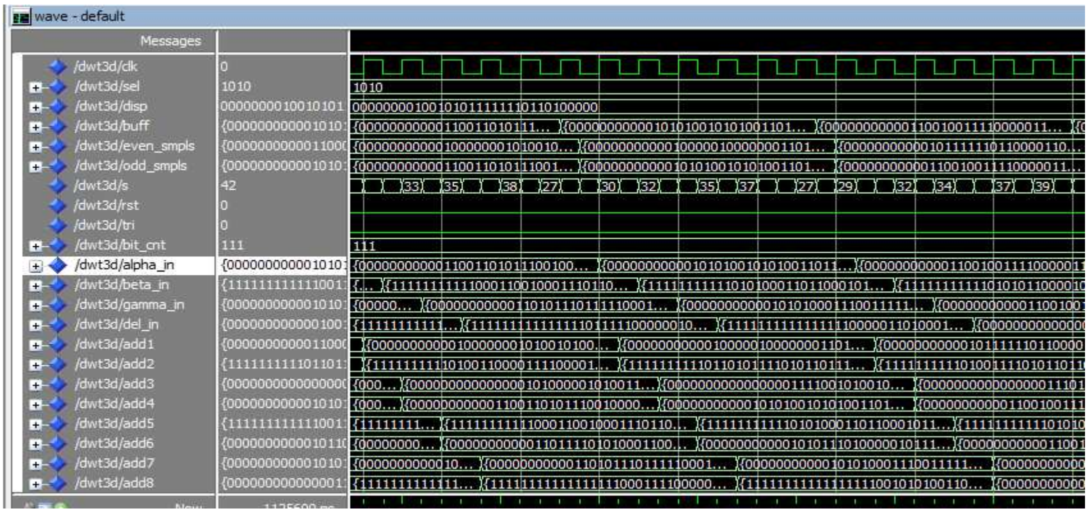


Figure 7. Input image

A. *Simulated Results*

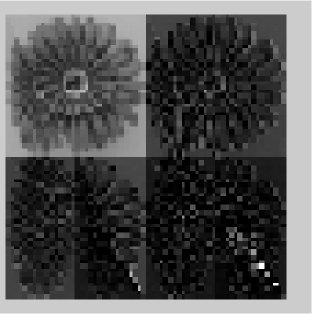
The given input image could be converted into corresponding binary values by using MATLAB Software. The main module is 3D_DWT. The simulation result of proposed architecture is shown in Figure 8 This result is simulated in ModelSim using VHDL coding.



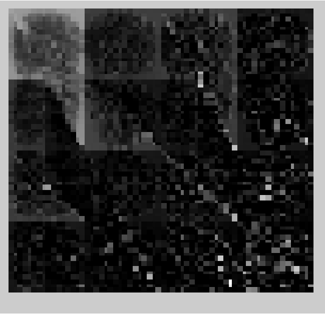
```
wave - default
Messages
/dwt3d/clk          0
/dwt3d/sel          1010
/dwt3d/disp         00000000100101011
/dwt3d/buff         {00000000000010101
/dwt3d/even_smpls   {000000000000110000
/dwt3d/odd_smpls    {00000000000010101
/dwt3d/s            42
/dwt3d/rst          0
/dwt3d/tri          0
/dwt3d/bit_cnt      111
/dwt3d/alpha_in     {00000000000010101
/dwt3d/beta_in      {11111111111110011
/dwt3d/gamma_in     {00000000000010101
/dwt3d/del_in       {000000000000001001
/dwt3d/add1         {000000000000110000
/dwt3d/add2         {11111111111011011
/dwt3d/add3         {0000000000000000000
/dwt3d/add4         {00000000000010101
/dwt3d/add5         {11111111111110011
/dwt3d/add6         {000000000000101110
/dwt3d/add7         {00000000000010101
/dwt3d/add8         {00000000000000011
Now                 1125600 ns
```

Figure 8. Simulation Result

B. *Output Image*



a) 2D DWT output



b)3D DWT output

Figure 9. Output Image

C. *Synthesis Report*

Selected Device	: 3s1600efg320-4
Number of Slices	: 1792 out of 14752 (12%)
Number of Slice Flip Flops	: 160 out of 29504 (0%)
Number of 4 input LUTs	: 3491 out of 29504 (11%)
Number of bonded IOBs	: 35 out of 250 (14%)
Number of GCLKs	: 1 out of 24 (4%)

D. Timing Summary

Speed Grade	: -4
Minimum period	: 92.807ns (Maximum Frequency: 10.775MHz)
Minimum input arrival time before clock	: No path found
Maximum output required time after clock	: 98.232ns
Maximum combinational path delay	: 8.023ns

E. MAP Report

Design Summary

Number of errors	: 0
Number of warnings	: 0

Logic Utilization

Number of Slice Flip Flops	: 160 out of 29,504 (1%)
Number of 4 input LUTs	: 3,475 out of 29,504 (11%)

Logic Distribution

Number of occupied Slices	: 1,852 out of 14,752 (12%)
Number of Slices containing only related logic	: 1,852 out of 1,852 (100%)
Number of Slices containing unrelated logic	: 0 out of 1,852 (0%)
Total Number 4 input LUTs	: 3,483 out of 29,504 (11%)
Number used as logic	: 3,475
Number used as a route-thru	: 8
Number of bonded IOBs	: 35 out of 250 (14%)
Number of GCLKs	: 1 out of 24 (4%)
Total equivalent gate count for design	: 22,946
Additional JTAG gate count for IOBs	: 1,680
Peak Memory Usage	: 247 MB

F. Comparison between proposed and existing system

Proposed Report:

Timing Summary

92.807 ns (Maximum Frequency	: 10.775MHz)
Total equivalent gate count for design	: 22,946
Number of occupied Slices	: 1852

Existing Report:

Timing Summary

13.025µs (Maximum Frequency	: 8.784MHz)
Total equivalent gate count for design	: 23,963
Number of occupied Slices	: 1905

V. Conclusion

The architecture is developed for lossy compression, which is based on the lifting algorithm of Daubechies (9, 7) filters and CDF97 filters. The advantages of the proposed architecture are saving embedded memories, fast computing time, low power consumption, and low control complexity. This hardware is designed to be used as part of a complete high performance and low power JPEG2000 encoder system for digital cinema applications. In Future, this Architecture could be implemented in FPGA. It is also possible to provide multilevel decomposition for 3D-DWT.Also, Delay in the design would be optimized. This Architecture could also extend to multi level DWT.

References

[1] Barua.S, Carletta.J.E, Kotteri.K.A, and Bell.A.E(2005),”An efficient architecture for lifting-based two-dimensional discrete wavelet transform,”Integr.VLSI J.,Vol.38,no.3,pp.341-352.

[2] Cheng.C and Keshab.K Parhi(2008),”High-speed VLSI implementation of 2-D Discrete wavelet transform,”IEEE Trans.Signal Processing,Vol.56,no.1,pp.393-403.

[3] Christopoulos.C, Skodras.A, and Ebrahimi.T (2000),”The JPEG2000 still image coding system: An overview,”IEEE Trans.Consumer Electronics, Vol.46, pp.1103-1127.

[4] Chrysafis.C and Ortega. A (2000),”Line-based, reduced memory,wavelet image compression,”IEEE Trans.Image Processing,Vol.9,no.3,pp.378-389.

[5] Daubechies.I and Sweldens.W (1998),”Factoring wavelet transforms into lifting steps,”Journal Fourier Anal.Applicat.,Vol.4, pp247-269.

[6] Dillen.G, Georis.B, et al.(2003),”Combined line-based architecture for the 5-3 and 9-7 wavelet transform of the JPEG2000,”IEEE.Trans.Circuits and Systems for video technology,Vol.13,no.9,pp.944-950.

[7] Grangetto.M,Magli.E,Martina.M and Olmo.G(2002),”Optimization and implementation of the integer wavelet transform for image coding,”IEEE Trans.Image .Processing,Vol.11,no.6,pp.596-604.

[8] Huang.C.T,Tseng.P.C, and Chen.L.G(2004),”Flipping structure:An efficient VLSI architecture for the lifting based discrete wavelet transform,”IEEE Trans.Signal.Processing,Vol.52,no.4,pp.1080-1089.

[9] Wang.C, Gan .W.S, (2007),”Efficient VLSI architecture for lifting based discrete wavelet packet transform,”IEEE Trans.Circuits and Systems-II:express briefs,Vol.54.no. 5, pp.422-426.

Mr. A. KARTHIKEYAN was born in karaikal, Pondicherry, India in 1986. He received his B.Tech degree specialized in Electronics and Communication Engineering from Bharathiyar College of Technology, Karaikal in the year 2008, under Pondicherry University, Pondicherry, and M.E., degree in VLSI Design in SNS College of Technology, Coimbatore in the year 2012 from Anna University, Chennai. He is now working as Assistant Professor in the Department of Electronics and Communication Engineering, SNS College of Technology, Coimbatore, Tamilnadu, India. His area of interest includes VLSI Design Techniques and Computer networking.

Ms. P. SARANYA was born in Coimbatore, Tamilnadu(TN),India in 1989. She received her B.E., Degree in Electronics and Communication Engineering from SNS College of Technology, Coimbatore in the year 2010, under Anna University, Chennai, Tamilnadu and M.E., degree in VLSI Design from SNS College of Technology, Coimbatore in the year 2012 from Anna University, Chennai. She is now working as Assistant Professor in the Department of Electronics and Communication Engineering, Sriguru Institute of Technology, Coimbatore, Tamilnadu, India. Her area of interest includes Low power VLSI.

Ms. N. JAYASHREE was born in Tirupur, Tamilnadu(TN),India in 1985. She received her B.E., Degree specialized in Electronics and Communication Engineering from Maharaja Prithivi Engineering college, Coimbatore in the year 2007,under Anna University, Chennai, Tamilnadu and M.E., degree in VLSI Design from Anna University of Technology, Coimbatore in the year 2011 from Anna University, Coimbatore. She is now working as Assistant Professor in the Department of Electronics and Communication Engineering, Sriguru Institute of Technology, Coimbatore, Tamilnadu, India. Her area of interest includes Microprocessors, VLSI Design.