Interstage Pipeline VLSI Architecture for 2-D DWT

Ajinkya S. Bankar[1],Bhavika S. Shaha[2], P.K. Kadbe[3]
E&TC Department, Pune University[1,2,3]
VPCOE Baramati

Abstract

In this paper, a scheme for the design of a high-speed pipeline VLSI architecture for the computation of the 2-D discrete wavelet transform (DWT) is proposed. The main focus in the development of the architecture is on providing a high operating frequency and a small number of clock cycles along with an efficient hardware utilization by maximizing the inter-stage computational parallelism for the pipeline. The high-speed computation is achieved by efficiently distributing the task of the computations of multiple decomposition levels among the stages of the pipeline and by optimally configuring the data and synchronizing the operations of pipeline so as to maximize the inter-stage computational parallelism. To validate the proposed scheme, an algorithm is designed and implemented in MATLAB for the 2-D DWT computation. Then the circuit is simulated and implemented in VHDL.

1. Introduction

With the rapid progress of VLSI design technologies, many processors based on audio and image signal processing have been developed recently. The two-dimensional discrete wavelet transform (2-D DWT) plays major role in image/video compression standard. Wavelets decompose the signal at one level of approximation and detail signals at the next level. Thus subsequent levels can add more details to the information content. In addition to audio and image compression, the DWT has important applications in many areas, such as computer graphics, numerical analysis, radar target distinguishing and so forth. DWT is a computationally very intensive process and slow for many real-time applications when implemented in a general purpose computing system. It is essential to develope custom VLSI chips for DWT exploiting the underlying data parallelism to achieve high data rate.

H. Y. Liao *et al.* [2] have presented an architecture in which each of the row and columnwise filtering operations are decomposed using the so called lifting operations into a cascade of sub-filtering operations. The scheme leads to a low-complexity architecture with a large latency. C. Cheng *et al.* [3] have proposed an architecture in which a number of parallel FIR filters with a polyphase structure are used to improve the processing speed at the expense of increased hardware. F. Marino *et al.* [4] have introduced a two-stage pipeline architecture in which the first stage performs the task of the first decomposition level and the second one that of all the remaining levels, and has aimed at providing a short computation time. As the processing units employed in this architecture differ from one another, the complexity of the hardware resources is high and the design of the architecture is complicated.

A. Benkrid *et al.* [5] presents an FPGA architecture for the separable 2-D Biorthogonal Discrete Wavelet Transform (DWT) decomposition. The architecture is based on the Pyramid Algorithm Analysis, which handles computation along the border efficiently by using the method of symmetric extension. P. McCanny *et al.* [6] have given, a VLSI architecture for performing the symmetrically extended two-dimensional transform is presented. This architecture conforms to the JPEG- 2000 standard and is capable of near-optimal performance when dealing with the image boundaries.The architecture also achieves efficient processor utilization. S. Raghunath *et al.* [7] have presented an efficient architecture for a multi-resolution symmetrically extended 2-D 9/7 filter discrete wavelet transform processor is presented. Hardware complexity is greatly reduced with improved performance, due to the proposed combination of lifting scheme and line based architecture. I. S. Uzun *et al.* [8] have designed the non-separable 2-D discrete biorthogonal wavelet filter architecture which has been derived from modified-recursive-pyramid-algorithm. MRPA based architecture exploits the downsampling of output subbands and performs the first decomposition level interspersed with all other levels by means of only one processing unit. C. Zhang *et al.* [9] presents, a scheme for the design of a high-speed pipeline VLSI architecture for the computation of the 2-D discrete wavelet transform (DWT). The main focus in the development of the architecture is on providing a high operating frequency and a small number of clock cycles along with an efficient hardware utilization.

In this paper, a non-separable pipeline architecture for fast computation of the 2-D DWT with a reasonable low cost for the hardware resources is proposed. Separable approach is a simple way to compute the 2-D DWT. However, separable filters being a special class of 2-D filters are not capable to approximate well all arbitrary frequency responses. In this regard, a non-separable approach of the 2-D computation provides more flexibility. In the non-separable approach depicted in Fig. 1, the DWT of a 2-D signal $s(n_1, n_2)$ is computed by carrying out four separate 2-D filtering operations using four 2-D filters: a highpass-highpass (HH) filter $G_{HH}(z_1, z_2)$, a highpass-lowpass (HL) filter $G_{HL}(z_1, z_2)$, a lowpass-highpass (LH) filter $G_{LH}(z_1, z_2)$, and a lowpass- lowpass (LL) filter $G_{LL}(z_1, z_2)$.The output signals of these four filters are then decimated by a factor of two in the horizontal and vertical directions producing, respectively the HH, HL, LH and LL components.

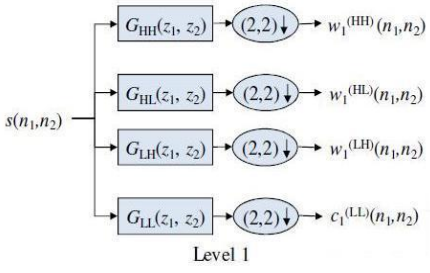


Fig.1 Computation of 1-Level 2-D DWT based on Non-Separable approach

2. Formulations for the computation of 2-D DWT

The 2-D DWT is an operation through which a 2-D signal is successively decomposed in a spatial multi-resolution domain by low- pass and highpass FIR filters along each of the two dimensions. The four FIR filters, denoted as highpass-highpass (HH), highpass-lowpass (HL), lowpass-highpass (LH) and lowpass-lowpass (LL) filters, produce, respectively, the HH, HL, LH and LL subband data of the decomposed signal at a given resolution level. The samples of the four subbands of the decomposed signal at each level are decimated by a factor of two in each of the two dimensions. For the operation at the first level of decomposition, the given 2-D signal is used as input, whereas for the operations of the succeeding levels of decomposition, the decimated LL subband signal from the previous resolution level is used as input.

2.1 Formulation for the Computation of Four Subbands

Let a 2-D signal be represented by $N_0 \times N_0$ matrix $S^{(0)}$, with its $(m,n)^{th}$ element denoted by $S^{(0)}(m,n)(0 \le m, n \le N_0\text{-}1)$, where N_0 is chosen to be 2^J , J being an integer. Let the coefficients of a 2-D FIR filter P (P=HH, HL, LH, LL) be represented by an L×M matrix $H^{(P)}$. The $(k,i)^{th}$ coefficient of the filter P is denoted by $H^{(P)}(k, i)(0 \le k \le L\text{-}1; 0 \le i \le M\text{-}1)$. The decomposition at a given level j=1, 2, , , J can be expressed as-

$$A^{(j)}(m,n) = \sum_{k=0}^{L-1}\sum_{i=0}^{M-1} H^{(HH)}(k,i).S^{(j-1)}(2m-k,2n-i) \quad (1)$$

$$B^{(j)}(m,n) = \sum_{k=0}^{L-1}\sum_{i=0}^{M-1} H^{(HL)}(k,i).S^{(j-1)}(2m-k,2n-i) \quad (2)$$

$$C^{(j)}(m,n) = \sum_{k=0}^{L-1}\sum_{i=0}^{M-1} H^{(LH)}(k,i).S^{(j-1)}(2m-k,2n-i) \quad (3)$$

$$S^{(j)}(m,n) = \sum_{k=0}^{L-1}\sum_{i=0}^{M-1} H^{(LL)}(k,i).S^{(j-1)}(2m-k,2n-i) \quad (4)$$

where $A^{(j)}$, $B^{(j)}$, $D^{(j)}$ and $S^{(j)}$, respectively, representing the HH, HL, LH and LL subbands of the 2-D input signal at the j^{th} level.

2.2 Formulation for a Four-Channel Filtering Operation

In order to facilitate parallel processing for the 2-D DWT computation, the L×M filterig operation needs to be divided into multi-channel operations, each channel processing one part of the 2-D data. It is seen from (4) that the even and odd indexed elements are always operated on the even and odd indexed filter coefficients, respectively. The matrix $S^{(j)}$ representing the LL subband at the jth level can, therefore, be divided into four (Nj/2+ L/2) × (Nj/2 + M/2) sub-matrices, $S^{(j)}ee$, $S^{(j)}oe$,$S^{(j)}eo$ and $S^{(j)}oo$, whose $(m,n)^{th}$ $(0 \le m \le Nj/2 + L/2 - 1,\ 0 \le n \le Nj/2 + M/2 - 1)$ elements are given by

$$\begin{aligned} s_{ee}^{(j)}(m,n) &= s^{(j)}(2m,2n) \\ s_{oe}^{(j)}(m,n) &= s^{(j)}(2m+1,2n) \\ s_{eo}^{(j)}(m,n) &= s^{(j)}(2m,2n+1) \\ s_{oo}^{(j)}(m,n) &= s^{(j)}(2m+1,2n+1) \end{aligned} \quad (5)$$

taking into consideration the periodic padding samples at the boundary. It is seen from (5) that the data at any resolution level are divided into four channels for processing by first separating the even and odd indexed rows of S(j), and then separating the even and odd indexed columns of the resulting two sub matrices. The

data in each channel can then be computed by an (L/2×M/2)-tap filtering operation. In order to facilitate such a 4-channel filtering operation, the filter coefficients, as used in (4), need to be decomposed appropriately. Accordingly, the matrix $H^{(P)}$ needs to be decomposed into four (L/2×M/2) sub-matrices, $H^{(p)}ee$,$H^{(p)}oe$, $H^{(p)}eo$ and $H^{(p)}oo$, whose (k,i)th ($0 \le k \le L/2\text{-}1$, $0 \le i \le M/2\text{-}1$) elements are given by respectively.

$$\begin{aligned} H_{ee}^{(P)}(m,n) &= H^{(P)}(2m,2n) \\ H_{oe}^{(P)}(m,n) &= H^{(P)}(2m+1,2n) \\ H_{eo}^{(P)}(m,n) &= H^{(P)}(2m,2n+1) \\ H_{oo}^{(P)}(m,n) &= H^{(P)}(2m+1,2n+1) \end{aligned} \quad (6)$$

By using (5) and (6) in (1- 4), any of the four subband signals, $A^{(j)}$,$B^{(j)}$, $C^{(j)}$ and $S^{(j)}$, at the j^{th} resolution level, can be computed as a sum of four convolutions using (L/2×M/2)-tap filters. For example, the LL subband given by (4) can now be expressed as

$$\begin{aligned} S^{(j)}(m,n) = &\sum_{k=0}^{L/2-1}\sum_{i=0}^{M/2-1} H_{ee}^{(LL)}(k,i).S_{ee}^{(j-1)}(m+k,n+i) \\ &+ \sum_{k=0}^{L/2-1}\sum_{i=0}^{M/2-1} H_{eo}^{(LL)}(k,i).S_{eo}^{(j-1)}(m+k,n+i) \\ &+ \sum_{k=0}^{L/2-1}\sum_{i=0}^{M/2-1} H_{oe}^{(LL)}(k,i).S_{oe}^{(j-1)}(m+k,n+i) \\ &+ \sum_{k=0}^{L/2-1}\sum_{i=0}^{M/2-1} H_{oo}^{(LL)}(k,i).S_{oo}^{(j-1)}(m+k,n+i) \end{aligned} \quad (7)$$

At any resolution level, the separation of the subband processing corresponding to even and odd indexed data as given by (7) is consistent with the requirement of decimation of the data in each dimension by a factor of two in the DWT computation. It is also seen from (7) that the filtering operations in the four channels are independent and identical, which can be exploited in the design of an efficient pipeline architecture for the 2-D DWT computation.

3. Pipeline For The 2-D DWT Computation

A straightforward mapping of the overall task of the DWT computation to a pipeline is one-level to one-stage mapping, in which the tasks of J resolution levels are distributed to J stages of the pipeline. In this mapping, the amount of hardware resources used by a stage should be one-quarter of that used by the preceding stage. Thus, the ratio λ of the hardware resource used by the last stage to that used by the first stage has a value of $1/4^{J-1}$. For images of typical size, this parameter would assume a very small value. Hence, for a structure of the pipeline that uses identical filter units, the number of these filter units would be very large. Further, since the number of such filter units employed by the stages would decrease exponentially from one stage to the next in the pipeline, it will make their synchronization very difficult. The solution to such a difficult synchronization problem, in general, requires more control units, multiplexers and registers, which result in a higher design complexity. A reasonably large value of $\lambda < 1$ would be more attractive for synchronization. In this respect, the parameter λ can be seen as a measure of design difficulty, with a smaller value of this parameter representing a greater design complexity[9].

The parameter λ can be increased from its value of $1/4^{J-1}$ in the one-level to one- stage pipeline structure by dividing the large-size stages into a number of smaller stages or merging the small-size stages into larger ones. However, dividing a stage of the one-level to one-stage pipeline into multiple stages would require a division of the task associated with the corresponding resolution level into sub-tasks, which in turn, would call for a solution of even a more complex problem of synchronization of the sub-tasks associated with divided stages. On the other hand, merging multiple small-size stages of the pipeline into one stage would not create any additional synchronization problem. As a matter of fact, such a merger could be used to reduce the overall number of filter units of the pipeline.

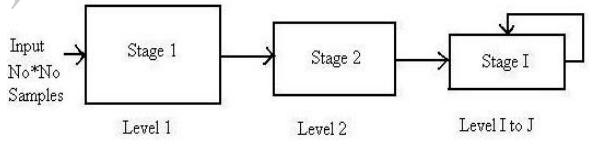


Fig.2 Pipeline structure with I stages for J-level computation

In view of the above discussion, the synchronization parameter λ can be increased by merging a number of stages at tail end of the pipeline. Fig. 2 shows the structure of a pipeline in which the stages I to J of the one-level to one-stage pipeline have been merged. In this structure, the tasks of the resolution level from j=1 to j=I - 1 are mapped to stage 1 to I - 1, respectively, whereas those of the resolution levels j=I, , J are mapped all together to the I^{th} stage. Note that the total amount of computations performed by stage I is less than one-half of that performed by stage I - 1. Considering the fact that the number of filter units employed by each stage of the pipeline is an integer, it is reasonable to have the ratio of the numbers of filter units used by the last two stages (i.e., stages I - 1 and I) to be 2:1. The value of the parameter λ is now increased from $1/4^{J-1}$ to 1/4I-1.5. However, now the resources employed by stage I would not be fully

utilized, which would lower the efficiency of the hardware utilization of the pipeline of Fig. 2

Assume that the parameter η represents the hardware utilization efficiency defined as the ratio of the resources used to that employed by the pipeline [9]. The hardware utilization efficiency η of the pipeline in Fig.2 can be shown to be equal to $(1 - 4^{-J})/(1 + 4^{-I+0.5})$. Since for images of typical size, 4^{-J} is negligibly small compared to one, the expression for η can be simplified as $1/(1 + 4^{-I+0.5})$. As the number of stages I employed by the pipeline increases, the hardware utilization efficiency increases with the parameter η approaching unity for a maximum efficiency. On the other hand, the difficulty in synchronizing the stages gets worse as the parameter λ decreases with increasing value of I. A variation in the value of I results in the values of λ and η that are in conflict from the point of view of stage synchronization and hardware utilization efficiency. Therefore, a value of I needs to be determined that optimizes the values of λ and η jointly.

Considering an example of an image of size $2^8 \times 2^8$, in which case J=8 . Table I gives the values of the parameters λ and η for the pipeline structures with *I*=2,3 and 4.

Table 1
Values of the parameters λ and η

Parameter	I=2	I=3	I=4
λ	1/2	1/8	1/32
η	89%	96%	99%

It is seen from this table that the 2-stage and 3-stage pipelines have acceptable values of λ, whereas the synchronization of the 4-stage pipeline would be very difficult because of its very low value of λ=1/32. On the other hand, the 3-stage and 4-stage pipelines have more desirable values of η in comparison to that for the 2-stage pipeline. Therefore, a 3-stage pipeline with an acceptable value for the synchronization parameter and high hardware utilization efficiency would be the best choice of a pipeline

4. Design Of Stages

In the proposed three-stage architecture, stages 1 and 2 perform the computations of levels 1 and 2, respectively, and stage 3 that of all the remaining levels. Since the basic operation of computing each output sample, regardless of the resolution level or the subband, is the same, the computation blocks in the three stages can differ only in the number of identical processing units employed by them depending on the amount of the computations assigned to the stages. As seen from (7), an (L×M)-tap filtering operation is decomposed into four independent (L/2×M/2)-tap filtering operations, each operating on the 2-D L/2×M/2 data resulting from the even or odd numbered rows and even or odd numbered columns of an L×M window of an LL- subband data[9]. An L×M window of the raw 2-D input data or that of an LL-subband data must be decomposed into four distinct L/2×M/2 sub-windows in accordance with the four decomposed terms given by the right side of (7). This decomposition of the data in an L×M window can be accomplished by designing for each stage an appropriate data scanning unit (DSU) based on the way the raw input or the LL-subband data is scanned. The stages would also require memory space (buffer) to store the raw input data or the LL-subband data prior to scanning. Fig.3 gives the block diagram of the pipeline showing all the components required by the three stages. Note that the data flow shown in this figure comprises only the LL subband data necessary for the operations of the stages

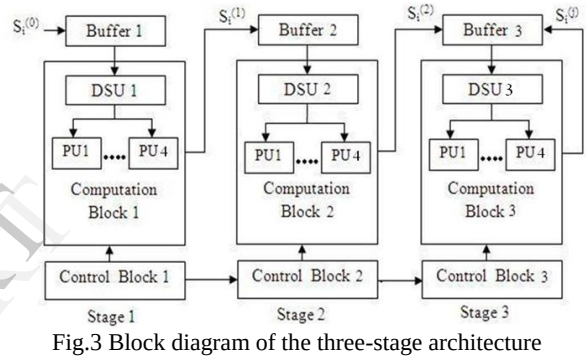
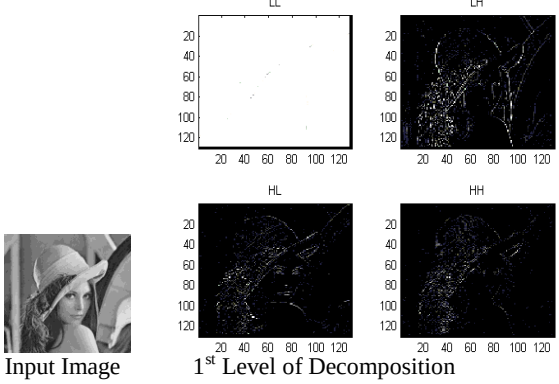


Fig.3 Block diagram of the three-stage architecture

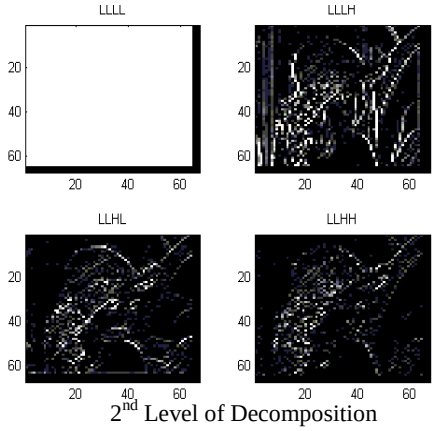
5. Performance Results

The Pipeline algorithm for decomposition of input data is implemented in MATLAB. Fig.5 shows input image and results of 1st level of decomposition and Fig.6 shows 2nd and Fig.7 shows 3rd level of decomposition.



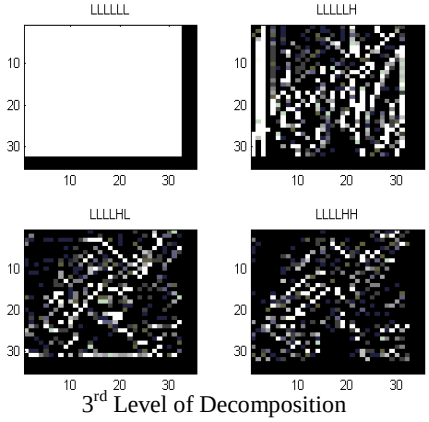
Input Image 1st Level of Decomposition

Fig.5 Results of MATLAB Implementation



2nd Level of Decomposition

Fig.6 Results of MATLAB Implementation



3rd Level of Decomposition

Fig.7 Results of MATLAB Implementation

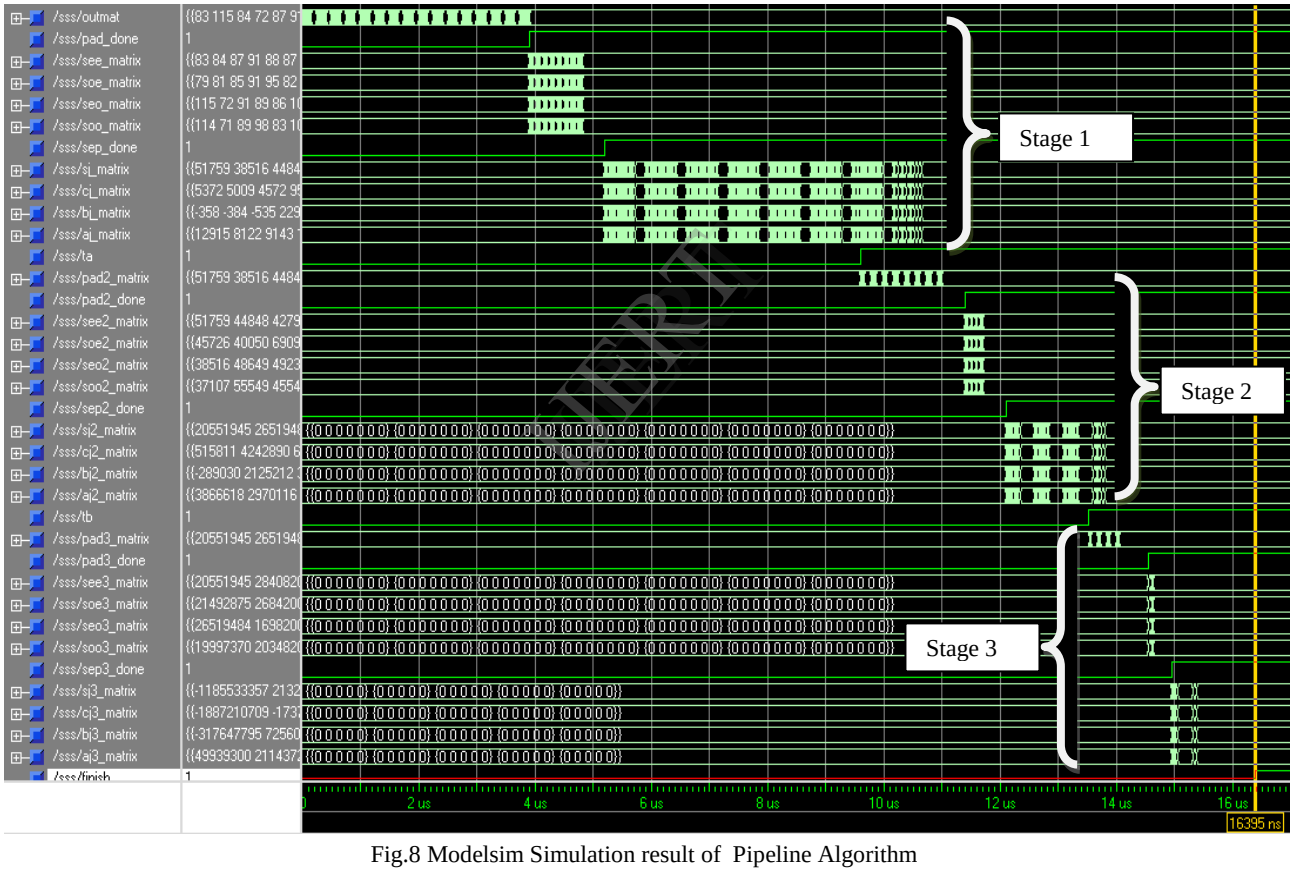


Fig.8 Modelsim Simulation result of Pipeline Algorithm

Same pipeline algorithm is implemented in VHDL. For this purpose the filter co-efficients are scaled and then they are used in the design. This digital design is simulated in Modelsim and its results are shown in Fig.8. For 100MHz of clock signal,three stages of pipeline, three levels of decomposition and image size of 16×16, it requires 16395ns.

6. Conclusion

To enhance the inter-stage parallelism, it is most efficient to map the overall task of the DWT computation to only three pipeline stages for performing the computation tasks corresponding to the decomposition level 1, level 2, and all the remaining levels, respectively. Two parameters, one specifying the synchronization of the operations of the stages and

the other representing the utilization of the hardware resources of the pipeline, have been defined. It has been shown that the best combination for the value of these parameters is achieved when the pipeline is chosen to have three stages.

7. References

[1] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.11, no. 7, pp. 674-693, Jul.1989

[2] H. Y. Liao, M. K. Mandal, and B. F. Cockburn, "Efficient architectures for 1-D and 2-D lifting-based wavelet transforms," *IEEE Trans. Signal Process.*, vol. 52, no. 5, pp 1315-1326, May 2004.

[3] C. Cheng and K. K. Parhi, "High-speed VLSI implementation of 2-D discrete wavelet transform," *IEEE Trans. Signal Process.*, vol. 56, no.1, pp. 393-403, Jan. 2008.

[4] F. Marino, "Efficient high-speed low-power pipelined architecture for the direct 2- D discrete wavelet transform," *IEEE Trans. Circuits Syst. II, Analog. Digit. Signal Process.*, vol. 47, no. 12, pp. 1476- 1491, Dec 2000.

[5] A. Benkrid, D. Crookes, and K. Benkrid, "Design and implementation of a generic 2-D orthogonal discrete wavelet transform on an FPGA," *in Proc. IEEE 9th Symp. Field-programming Custom Computing Machines (FCCM)*, Apr. 2001, pp. 190-198.

[6] P. McCanny, S. Masud, and J. McCanny, "Design and implementation of the symmetrically extended 2-D wavelet transform," *in Proc. IEEE Int. Conf. Acoustic, Speech, Signal Process.(ICASSP)*, 2002, vol. 3, pp. 3108-3111.
S. Raghunath and S. M. Aziz, "High speed area efficient multi-resolution 2-D 9/7 filter DWT processor," *Proc. Int. Conf. Very Large Scale Integration (IFIP)*, Oct. 2006, vol. 16-18. Pp, 210-215.

[7] I. S. Uzun and A. Amira, "Rapid prototyping-framework for FPGA based discrete biorthogonal wavelet transforms implementation," *IEE Vision, Image Signal Process.*, vol. 153, no. 6, pp. 721-734, Dec. 2006.

[8] C. Zhang, C. Wang, and M. O. Ahmad, "A Pipeline VLSI Architecture for Fast Computation of the 2-D Discrete Wavelet Transform", *IEEE Trans. On Circuits and Systems-I*, vol. 59, No. 8, August 2012.

[9] M. Alam,W. Badway, V. Dimitrov and G. Jullien, "An Efficient Architecture for a Lifted 2D Biorthogonal DWT", *Journal of VLSI Signal Processing* 40, 335342, 2005.

[10] R.C. Gonzalez,R. Woods, "Digital Image Processing",Prentice-Hall,3rd Edition, 2007.

[11] Principles of Digital System Design using VHDL, Charles H. Roth, Jr. & Lizy Kurian John, 1998 Cengage Learning publication

.