# Design and Implementation of Pipeline Architecture for High Performance 2-D Daubechies Wavelet Transform

T. Dhanya
P.G. Scholar, Department of Electronics and Communication Engineering
Adhiparasakthi Engineering College
Melmaruvathur, Tamilnadu, India

V. Nagarajan
Head of the Department, Department of Electronics and Communication Engineering
Adhiparasakthi Engineering College
Melmaruvathur, Tamilnadu, India

## ABSTRACT

Wavelet Transforms are used in number of application. They are applied in different fields such as signal processing, speech and image compression, biometrics, and so on. One of its important applications is image compression. Wavelet Transforms are preferred over other transforms, to compress image, since reconstruction is more accurate. Design of discrete wavelet transforms is complex due to large number of arithmetic operations involved. In this paper, pipeline VLSI architecture for the computation of the 2-D discrete wavelet transform (DWT) using db2 filter is proposed. The main focus of the scheme is to reduce the number of clock cycles needed for the computation. The hardware resources needed is optimized by maximizing the inter- and intra-stage parallelisms of the pipeline. The inter-stage parallelism is enhanced by optimally mapping a level of decomposition to the stages of the pipeline. The intra-stage parallelism is enhanced by decomposing the filtering operation equally into two subtasks that can be executed independently in parallel so that the delay of the critical data path for the filtering operation is minimized. This architecture requires smaller number of clock cycles and hardware resources compared to that of conventional architectures.

## General Terms

Discrete wavelet transforms (DWT), 2D-DWT computation, inter- and intra-stage parallelisms, multi-resolution filtering, parallel architecture, pipeline architecture, VLSI architecture.

## 1. INTRODUCTION

The use of the 2D Discrete Wavelet Transform (DWT) for image and video compression (lossy compression) is indisputable. For multilevel 2D DWT computation, several methods based on different input traversal patterns have been proposed. Among these, the most commonly used are: the row–column, the line-based and the block-based[1]. In my paper I use the row-column design. The pipeline architecture increases the processing speed and resource utilization. Simulation of the architecture is done using Modelsim.

Number of architectures has been proposed to provide high-speed and area-efficient implementations of DWT computation. For input image of size NxN consider a system implemented with L number of multipliers and J levels. A study of area complexities show that all the folded architectures have an area complexity of O(NLk) while the SIMD architectures have a complexity of O(N2k). The period of the folded architectures is approximately *N2* while that of the SIMD arrays is 2JL [2]. Flexible and efficient B-spline-based architecture for a one-level 2-D DWT [3] require extra buffers. Distributed arithmetic based DWT is used to reduce the number of multipliers [4]. In [5]–[7], the polyphase matrix of a wavelet filter is decomposed into a sequence of alternating upper and lower triangular matrices and a diagonal matrix to obtain the so-called lifting-based architectures with low hardware complexity. However, such architectures have a long critical path, which results in reducing the processing rate of input samples. On the other hand, the problem of low processing rate is not acute in the architectures that use convolution low and high-pass filtering operations to compute the DWT [8]–[14]. These convolution-based architectures can be categorized as single- or multistage pipeline architectures. The architectures proposed in [8]–[11] are single-stage architectures in which the DWT computation is performed using a recursive pyramid algorithm (RPA) [15] that results in a reduced memory space requirement for the architectures. Lewis and Knowles [8] have designed simple single-stage VLSI architecture to implement the computation of the DWT without multipliers where the problem of critical data path is not addressed. Chakrabarti and Vishwanath [9] have designed a SIMD architecture based on on-line algorithm. This architecture is efficient in terms of area and time but hardware resource requirement is still high since it uses systolic array and parallel filters. Grzesczak, Mandal, and Panchanathan [10] have proposed an architecture is systolic in nature and performs both high-pass and low-pass coefficient calculations with only one set of multipliers. Here too the problem of long critical path is not addressed. Cheng and Parhi [12] have proposed a high-speed single-stage architecture based on hardware-efficient parallel finite-impulse response (FIR) filter structures for the DWT computation but it does not reduce the hardware requirements. The architecture proposed in [13] is a multistage architecture in which the tasks of the various decomposition levels of the DWT computation are distributed to a number of pipeline stages. High-speed multistage pipeline architecture, with one stage to carry out the computation of one decomposition level of the DWT, has been proposed in this architecture. The pipeline architectures have the advantages of requiring a small memory space and a short computing time and are suitable for real-time computations. The proposed system is an extended work of the scheme discussed in [16]. In this pipeline architecture scheme the performance of the system is increased by reducing the no of clock cycles needed and the period of clock. This is done by satisfying the basic conditions in [17]. The proposed system is simulated using Modelsim tool.

## 2. 2-D DWT COMPUTATION

The structure of the conventional separable 2-D DWT algorithm is shown in Figure1, where h[n] and g[n] represent the low pass and high pass sub-band filters, respectively. The suffix H and V denote the horizontal and vertical filtering operation respectively. The input image is first decomposed horizontally; the resulting two bands are then decomposed vertically into four sub-bands usually denoted by LL, LH, HL, and HH. The LL sub-band can then be further decomposed by the same method.

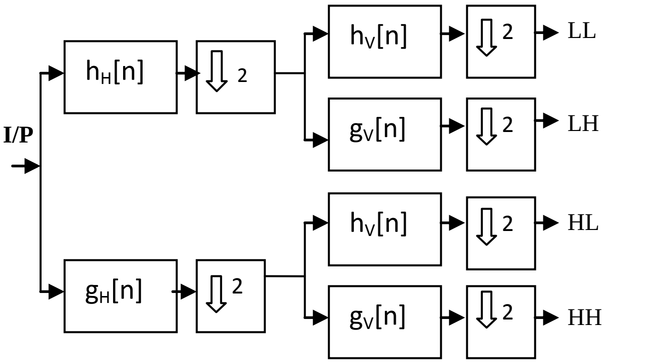

**Figure 1. Block Diagram of the 2-D Separable DWT.**

The 2D DWT can be considered as a `chain' of successive levels of decomposition as depicted in Figure 2. Because the 2D DWT is a separable transform, it can be computed by applying the 1D DWT along the rows and columns of the input image of each level during the horizontal and vertical filtering stages. Every time the 1D DWT is applied on a signal, it decomposes that signal in two sets of coefficients: a low-frequency and a high-frequency set. The low frequency set is an approximation of the input signal at a coarser resolution, while the high-frequency set includes the details that will be used at a later stage during the reconstruction phase. This procedure, presented in Figure 2, is known as the dyadic decomposition. A N level filtering operation for a 2-D DWT is illustrated in the figure given below.

## 3. PROPOSED ARCHITECTURE

First, a decomposition level is considered as two consecutive horizontal and vertical filtering operations[18], these are assigned to the stages so as to equalize the computations carried out by each stage, i.e., the hardware requirements of all the stages are kept the same. Next, the computations of the successive decomposition levels are assigned to the successive stages of a pipeline on a one-level-to-two-stage basis. Thus, in this case, the hardware requirement of the stages gets reduced as they perform the computations corresponding to higher level decompositions.

The use of the same MAC-cell network for all the stages would not be possible, unless the pipeline has only two stages [16]. For a stage-equalized pipeline structure the requirement is that all the stages should have the same hardware resource. The same MAC-cell network can be used easily for all the stages. However, in this case, the same amount of computations cannot be assigned to all the stages that are based on the same MAC-cell network, unless, again, there are only two stages in the pipeline. The two-stage version of a pipeline structure is shown in Figure 2.

An additional advantage of the two-stage pipeline is in the design flexibility of a MAC-cell network where the multiplication and accumulation operations can be furnished together by using logic gates. These logic gates could be arranged into more efficient arrays yielding a shorter propagation delay for the MAC-cell network. In this two-stage structure, stage 2 performs the vertical filtering operation by operating on the data produced by stage 1. The operations of the two stages need to be synchronized in a best possible manner.

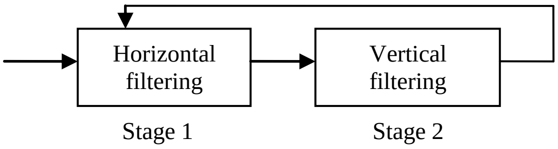

**Figure 2. Pipeline Structure with Two Stages.**

### 3.1 Design Of Stages

Since, in the stage-equalized architectures, the two stages together perform the DWT computation, with the amount and type of computations of the individual stages being the same, each of the two stages can use identical processing units. However, the control units to be employed by the stages have to be different, since, the operation of stage 1 is horizontal filtering, whereas stage 2 must always synchronize its vertical filtering operation with that of stage 1. Hence, the design of the control unit used by stage 2 would have to be a bit more tedious than that of the control unit used by stage 1.

Obviously, in order to synchronize the operation of stage 2 with that of stage 1, a buffer has to be used to store the low-pass output samples from stage 1. To synchronize the operation between consequent levels of decomposition a single 16-bit register is used as buffer at the input of stage 1. Figure 3 shows a block diagram incorporating all these requirements for the design of the proposed architecture. The two processing units are referred to as PU in stage 1 and $PU_2$ in stage 2.

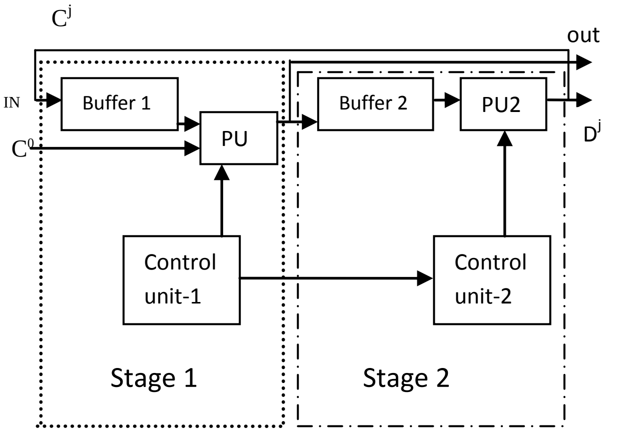

**Figure 3. Block Diagram of Two Stage Architecture**

All the output data must be synchronized. This synchronization process is facilitated by introducing a buffer in stage 1 and stage 2. The buffer in stage 1 stores the output of each level and provides input to next level of decomposition while the buffer in stage 2 which stores output data from the stage 1 and provides input data to stage 2.

### 3.2 Synchronization Of Stages

This filtering operation imposes the constraint that the first output sample at level j cannot be computed until [L-1xM]+1(Where, M is the no. of columns at output of level j-1) samples at level j-1 have already been computed and each of the subsequent samples at level j cannot be computed, unless two new samples at level j-1 have already been computed. Note that this requirement of the filtering operation imposes a constraint on the operation of stage 1 and stage 2. Under this constraint, we now give three steps of the synchronization scheme that govern the computation of the output samples at various decomposition levels by stages 1 and 2.

1.1.1 Algorithm

Initially at level 1, stage 1 operates continuously to compute the level-1 output samples sequentially until output of stage 2 is generated. Once the output of stage 2 is available it starts the computation of level 2 at time slot (L+2).

Step 2)

Stage 2 starts the computation of samples beginning at the time slot ([L-1xM]+3).

Step 3)

a) *When stage 1 or 2 is computing an output sample at the lowest incomplete level*.

After completing the computation of the present sample at this level, stage moves on to the computation of a sample at the lowest higher level, if the data required for the computation of this sample have become available; otherwise, stage continues with the computation of the next sample at the present level.

b) *When stage 1 or 2 is computing an output sample at a level other than the lowest incomplete level*.

After completing the computation of the present sample, stage moves its operation to the lowest incomplete level.

The rationale behind Step 3a) is that moving the operation of stages to a higher level allows more data from level 1, to become available, since the availability of the output samples of level 1 is crucial for the computation of the samples at higher levels. On the other hand, the rationale behind Step 3b) is that there are always more samples to be computed at lower levels than that at higher levels, and therefore, more time needs to be spent in computing lower level samples.

## 4. IMPLEMENTATION

The simulation of 2D DWT computation using the pipeline architecture was done using Modelsim_Altera 6.3g_p1, and MATLAB R2007b to convert image to data and data back to image. The scheme can be tested by implementing in altera TE270 kit. Comparison of the proposed system with other existing architectures for 2D DWT computation is given in Table 1. Figure 4 shows the 256 X 256 16 bit gray scale image that is applied as input to the system. The consolidated output image of both low and high pass filters of level 1 and 2 are as in Figure 5 and Figure 6 respectively.

**Table 1: Comparison of the Proposed System with Other Existing Architectures For 2D DWT Computation**

| ARCHITECTURE | No. of multipliers | No. of adders | Computation time |
|---|---|---|---|
| Systolic [19] | 84 | 84 | $N^2$ |
| Lifting based [7] | 22 | 24 | 21/64 $N^2$ |
| Convolution [11] | 5 | 4+4 | - |
| Proposed | 4 | 2 | [L-1xM]+N+2 |

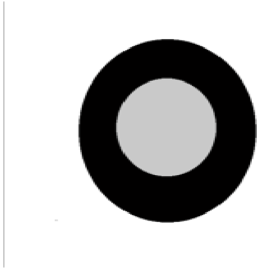

**Figure 4. Input Image**

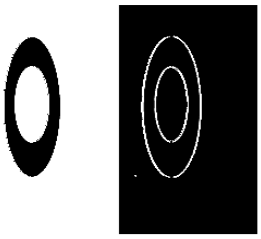

**Figure 5. Output Image For Stage 1**

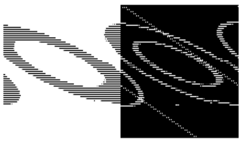

**Figure 6. Output Image For Stage 2**

## 5. CONCLUSION

The proposed architecture is designed to achieve a low computation time by reducing the number of clock cycles required for the DWT computation. This has been realized by developing a scheme for enhanced inter- and intra-stage parallelisms. It is most efficient to map the overall task of the DWT computation to only two pipeline stages [16]. This architecture is better than the existing architecture in terms of computation time and efficient use of available resources. In order to assess the effectiveness of the proposed scheme, the design is simulated using Modelsim and MATLAB. The same scheme can be extended to 3-D DWT.

## 6. REFERENCES

[1] Maria E. Angelopoulou, Peter Y. K. Cheung, Konstantinos Masselos, Yiannis AndreopouloS "Implementation and Comparison of the 5/3 Lifting 2D Discrete Wavelet Transform Computation Schedules on FPGAs" *Journal of VLSI Signal Processing* 2007.

[2] C. Chakrabarti, M. Vishwanath, and R. M. Owens, "Architectures for wavelet transforms: A survey," *J. VLSI Signal Process.*, vol. 14, no. 2, pp. 171–192, Nov. 1996.

[3] C. Huang, P. Tseng, and L. Chen, "Analysis and VLSI architecture for 1-D and 2-D discrete wavelet transform," *IEEE Trans. Signal Process.*, vol. 53, no. 4, pp. 1575–1586, Apr. 2005.

[4] Acharyya, K. Maharatna, B. M. Al-Hashimi, and S. R. Gunn, “Memory reduction methodology for distributed-arithmetic-based DWT/IDWT exploiting data symmetry,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 56, no. 4, pp. 285–289, Apr. 2009.

[5] K. A. Kotteri, S. Barua, A. E. Bell, and J. E. Carletta, “A comparison of hardware implementations of the biorthogonal 9/7 DWT: Convolution versus lifting,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 52, no. 5, pp. 256–260, May 2006.

[6] C. Wang and W. S. Gan, “Efficient VLSI architecture for lifting-based discrete wavelet packet transform,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 54, no. 5, pp. 422–426, May 2007.

[7] Zahraa Talal Abed Al-Mokhtar “Design And FPGA Implementation Of Dual Scan Two Dimensional Discrete Wavelet Transforms”, Al-Rafidain Engineering Vol.19 No.4 August 2011.

[8] A. S. Lewis and G. Knowles, “VLSI architecture for 2D Daubechies wavelet transform without multipliers,” *Electron. Lett.*, vol. 27, no. 2, pp. 171–173, Jan. 1991.

[9] C. Chakrabarti and M. Vishwanath, “Efficient realizations of the discrete and continuous wavelet transforms: from single chip implementations to mapping on SIMD array computers,” *IEEE Trans. Signal Process.*, vol. 43, no. 3, pp. 759–771, Mar. 1995.

[10] A. Grzesczak, M. K. Mandal, and S. Panchanathan, “VLSI implementation of discrete wavelet transform,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 4, no. 4, pp. 421–433, Dec. 1996.

[11] Mountassar Maamoun, Mehdi Neggazi, Abdelhamid Meraghni, And Daoud Berkani, “VLSI Design Of 2-D Discrete Wavelet Transform For Area-Efficient And High-Speed Image Computing”, World Academy Of Science, Engineering And Technology 45 2008.

[12] 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.

[13] F. Marino, D. Guevorkian, and J. Astola, “Highly efficient high-speed/low-power architectures for 1-D discrete wavelet transform,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 47, no. 12, pp. 1492–1502, Dec. 2000.

[14] C. Zhang, C. Wang, and M. O. Ahmad, “A VLSI architecture for a high-speed computation of the 1D discrete wavelet transform,” in *Proc. IEEE Int. Symp. Circuits Syst.*, Kobe, Japan, May 2005, vol. 2, pp. 1461–1464.

[15] M. Vishwanath, “The recursive pyramid algorithm for the discrete wavelet transform,” *IEEE Trans. Signal Process.*, vol. 42, no. 3, pp. 673–677, Mar. 1994.

[16] Chengjun Zhang, Chunyan Wang and M. Omair Ahmad “A Pipeline VLSI Architecture for High-Speed Computation of the 1-D Discrete Wavelet Transform*”* *IEEE transactions on circuits and systems* 1549-8328 Feb. 2010.

[17] C.V.Ramamoorthy and H.F.Li “Pipeline Architecture” Computing Surveys Vol. 9, No. 1 March 1977.

[18] Jörg Ritter and Paul Molitor, “A Pipelined Architecture for Partitioned DWT Based Lossy Image Compression using FPGA’s” *FPGA ACM 2001,* pp.11-13, February 2001.

[19] A. Grzeszczak, M. K. Mandal, S. Panchanathan, *Member, IEEE*, and T. Yeap, *Member, IEEE* “VLSI Implementation of Discrete Wavelet Transform”, *IEEE Transactions on VLSI Systems,* Dec 1996.