

FPGA IMPLEMENTATION OF EFFICIENT VLSI ARCHITECTURE FOR FIXED POINT 1-D DWT USING LIFTING SCHEME

Durga Sowjanya¹, K N H Srinivas² and P Venkata Ganapathi³

¹Research fellow, Sri Vasavi Engineering College, Tadepalligudem

k.durgasowjanya@gmail.com

²Head of the department in Sri Vasavi Engineering College, Tadepalligudem

Knh.tridents@gmail.com

³Venkata Ganapathi Puppala, Quartics Technologies Pvt Ltd, Pune

ganapathi.pv@gmail.com

ABSTRACT

In this paper, a scheme for the design of area efficient and high speed pipeline VLSI architecture for the computation of fixed point 1-d discrete wavelet transform using lifting scheme is proposed. The main focus of the scheme is to reduce the number and period of clock cycles and efficient area with little or no overhead on hardware resources. The fixed point representation requires less hardware resources compared with floating point representation. The pipelining architecture speeds up the clock rate of DWT and reduced bit precision reduces the area required for implementation. The architecture has been coded in verilog HDL on Xilinx platform and the target FPGA device used is Virtex-II Pro family, XC2VP7-7board. The proposed scheme requires the least computing time for fixed point 1-D DWT and achieves the less area for implementation, compared with other architectures. So this architecture is realizable for real time processing of DWT computation applications.

KEYWORDS

Discrete wavelet transform (DWT), Lifting based scheme, field-programmable gate-array (FPGA), pipeline architecture, reduced bit precision, fixed point, VLSI architecture.

1. INTRODUCTION

The advantages of the wavelet transform over conventional transforms, such as the Fourier transform, are now well recognized. Because of its excellent locality in time-frequency domain, wavelet transform is remarkable and extensively used for signal analysis, compressing and denoising. Since the development of the theory for the computation of the discrete wavelet transform (DWT) by Mallat [1] in 1989, the DWT has been increasingly used in many different areas of science and engineering mainly because of the multi resolution decomposition property of the transformed signals. Definition given for DWT by Mallat [1] provided possibility of its implementation in hardware and software. The discrete wavelet transform (DWT) performs a multi resolution signal analysis which has adjustable locality in both the space (time) and frequency domains [1]. The DWT is computationally intensive because of multiple levels of decomposition involved in the computation of the DWT. It is therefore challenging to design an efficient VLSI architecture to implement the DWT computation for real-time applications, particularly those requiring processing of high-frequency or broadband signals [2]–[4]. Using finite impulse response (FIR) filters and then sub sampling is the classical method for

implementing the DWT. Due to the large amount of computations required, there have been many research efforts to develop new algorithms [15]. Many architectures have been proposed in order to provide high-speed and area-efficient implementations for the DWT computation [5]–[8]. In [9]–[11], the poly phase 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.

The pipeline architectures have the advantages of requiring a small memory space and a short computing time and are suitable for real-time computations. However, these architectures have some inherent characteristics that have not yet been fully exploited in the schemes for their design. The computational performance of such architectures could be further improved, provided that the design with pipeline make sure of lifting steps to the maximum extent possible, synchronizes the operations of the stages optimally, and utilizes the available hardware resources optimally.

In this paper, a scheme for the design of pipeline architecture for a fast computation of the DWT is developed. The goal of fast computation is achieved by minimizing the number and period of clock cycles. The main idea used for minimizing these two parameters is to optimally distribute the task of the DWT computation among the stages of the pipeline and lifting scheme of 9/7 filter. In the study, we focus on the issues of theoretical path and internal memory size with 9/7 filters. To ease the tradeoff between the pipeline stages of 1-D architecture, a modified algorithm is proposed for the design of 1-D pipeline architecture. Based on the modified data path of lifting-based DWT, the proposed architecture achieves the one-multiplier delay constraint but uses less internal memory compared to the related architectures. Moreover, the proposed architecture implements the 9/7 filters by cascading the three main components.

Due to recent advances in the technology, implementation of the DWT on field programmable gate array (FPGA) and digital signal processing (DSP) chips has been widely developed. Based on [4], the main challenges in the hardware architectures for 1-D DWT are the processing speed and the number of multipliers. The number of multipliers in each pipeline stage determines the clock speed of the structure.

This paper is organized as follows. In Section II, Discrete Wavelet Transform is presented. In Section III, choice of pipeline for the 1-d dwt is presented, Section IV, the Lifting based scheme is presented, Section V, briefly introduces the underlying concepts of the architecture of 1-d DWT. Section VI, Presents Performance evaluation and FPGA implementation and compares the proposed architecture with other related studies. Finally, a brief conclusion is given in Section VII.

2. DISCRETE WAVELET TRANSFORM

In this section the theoretical background and algorithm development is discussed. The first recorded mention of what is now called a "wavelet" seems to be in 1909, in a thesis by Alfred Haar. An image is represented as a two dimensional (2D) array of coefficients, each coefficient representing the brightness level in that point. When looking from a higher perspective, it is not possible to differentiate between coefficients as more important ones, a lesser important ones. But thinking more intuitively, it is possible. Most natural images have smooth color variations, with the fine details being represented as sharp edges in between the smooth variations.

Technically, the smooth variations in color can be termed as low frequency components and the sharp variations as high frequency components. The low frequency components (smooth variations) constitute the base of an image, and the high frequency components (the edges which

give the detail) add upon them to refine the image, thereby giving a detailed image. Hence, the averages/smooth variations are demanding more importance than the details [4]. In wavelet analysis, a signal can be separated into approximations and detail coefficients. Averages are the high-scale, low frequency components of the signal. The details are the low scale, high frequency components. These coefficients measure the signal energy distribution in each frequency channel corresponding to the scaling parameter j at the time k .

The Discrete Time wavelet Transform (DWT) has found many applications in digital signal processing, due to the efficient computation and the sufficient properties for non-stationary signal analysis.

For the wavelet analysis, the structure is given in figure 1(a). 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 (by quality, by resolution), ease of compressed image manipulation, region of interest coding, etc.

2.1. One dimensional DWT:

Any signal is first applied to a pair of low-pass and high-pass filters. Then down sampling (i.e., neglecting the alternate coefficients) is applied to these filtered coefficients. The filter pair (h, g) which is used for decomposition is called analysis filter-bank and the filter pair which is used for reconstruction of the signal is called synthesis filter bank (g', h') . The output of the low pass filter after down sampling contains low frequency components of the signal which is approximate part of the original signal and the output of the high pass filter after down sampling contains the high frequency components which are called details (i.e., highly textured parts like edges) of the original signal.

The output from low pass filter $G(z)$ represents the approximate coefficient denoted by $S^j(n)$.

$$S^j(n) = \sum_k S^{j-1}(k)G(2n-k)$$

The output from high pass filter $H(z)$ represents the detailed coefficient denoted by $W^j(n)$.

$$W^j(n) = \sum_k S^{j-1}(K)H(2n-k)$$

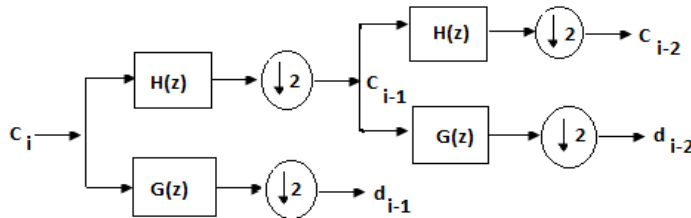


figure 1(a). 1-D DWT decomposition

3. PIPELINE FOR THE 1-D DWT COMPUTATION

In a pipeline structure for the DWT computation, multiple stages are used to carry out the computations of the various decomposition levels of the transform. Thus, the computation corresponding to each decomposition level needs to be mapped to a stage or stages of the pipeline. In order to maximize the hardware utilization of a pipeline, the hardware resource of a stage should be proportional to the amount of the computation assigned to the stage. Since the amount of computations in successive decomposition levels of the transform gets reduced by a factor of two, two scenarios can be used for the distribution of the computations to the stages of a pipeline. In the first scenario, the decomposition levels 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. In the second scenario, the computations of the successive decomposition levels are assigned to the successive stages of a pipeline on a one-level-to-one-stage basis. Thus, in this case, the hardware requirement of the stages gets reduced by a factor of two as they perform the computations corresponding to higher level decompositions.

A stage-equalized pipeline structure is the one in which the computations of all the levels are distributed equally among the stages. The process of stage equalization can be accomplished by dividing equally the task of a given level of decomposition into smaller subtasks and assigning each such sub task to a single stage and/or by combining the tasks of more than one consecutive level of decomposition into a single task and assigning it to a single stage. Note that, generally, a division of the task would be required for low levels of decomposition and a combination of the tasks for high levels of decomposition.

In a one-to-one mapped structure, the computations of decomposition levels are distributed exactly among all stages. In this structure, the computations of the first levels are carried out by the first stage, remaining levels are carried out by remaining stages respectively and those of the last levels are performed recursively by the second stage. Thus, for either pipeline structure, i.e., the one-to-one mapped or stage-equalized, a two-stage pipeline would be the best choice in terms of hardware efficiency and from the standpoint of design and implementation simplicity. Note that the five-stage version of either pipeline structure is the same but due to the flexibility of designing the architecture, stage equalized pipelining structure is preferred.

4. LIFTING BASED SCHEME

The lifting scheme has been developed as a flexible tool suitable for constructing the second generation wavelet. It is composed of three basic operation stages: splitting, predicting, and updating.

Fig.2 shows the lifting scheme of the wavelet filter computing one dimension signal:

- 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.
- 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.
- Update step: the detailed coefficients computed by the predict step are multiplied by the update factors and then the results are added to the even samples to get the coarse coefficients. Note that the details and approximation coefficients (d , s) in lifting scheme, respectively, are the same as high pass and low pass outputs. Daubechies and Sweldens first derived the lifting-based discrete wavelet transform [11], [12].

The lifting scheme can decompose DWT filter bank into several lifting steps. As $\tilde{h}^-(z)$ and $\tilde{g}^-(z)$ are the low-pass and high-pass analysis filters; the poly phase matrix $\tilde{p}^-(z)$ is defined as follows:

$$\tilde{P}(z) = \begin{pmatrix} \tilde{h}_e^-(z) & \tilde{h}_o^-(z) \\ \tilde{g}_e^-(z) & \tilde{g}_o^-(z) \end{pmatrix} \quad (1)$$

The poly phase matrix $\tilde{p}^-(z)$ can be factorized into a sequence of alternating upper and lower triangular matrices multiplied by a constant diagonal matrix.

The 9/7 filter has two lifting steps and one scaling step .The detailed algorithm of the 9/7 filter is described from (2) to (7). First, the input sequences x_i are split into even and odd parts, S_i^0 and d_i^0 .Second, the two splitting sequences are performed by two lifting steps. The outputs are denoted as S_i^n and d_i^n , where n represents the stage of lifting step. Finally, through the normalization factors k_1 and k_2 , the low-pass and high-pass wavelet coefficients S_i and d_i can be obtained.

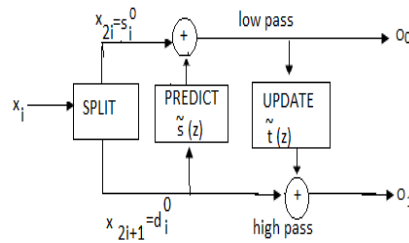


figure 2. Block diagram of the lifting scheme

1. Splitting Step:

$$d_i^0 = x_{2i+1} \quad (2)$$

$$s_i^0 = x_{2i} \quad (3)$$

2. Lifting Steps:

2.1. First lifting step

$$d_i^1 = d_i^0 + \alpha \times (s_i^0 + s_{i+1}^0) \text{ (predictor)} \quad (4)$$

$$s_i^1 = s_i^0 + \beta \times (d_{i-1}^1 + d_i^1) \text{ (updater)} \quad (5)$$

2.2. Second lifting step

$$d_i^2 = d_i^1 + \gamma \times (s_i^1 + s_{i+1}^1) \text{ (predictor)} \quad (6)$$

$$s_i^2 = s_i^1 + \delta \times (d_{i-1}^2 + d_i^2) \text{ (updater)} \quad (7)$$

Several architectures [14], [15] have been proposed to directly implement the lifting structures of the 9/7 filters. The five pipeline stages are used to improve the processing time, but the critical path is still restricted by the computation of predictor or updater (i.e., two adders and one multiplier propagation delay).

5. 1-D DWT ARCHITECTURE

By combining the functional units described in previous section we can construct one dimensional DWT. The architecture can be applied to implement the lifting-based 1-D DWT. The structure processes all input samples that arrive in pairs at consecutive clock pulses and the results for each pair are ready after five cycles. However, due to the pipelined structure, the clock frequency is higher than that of parallel architectures. There is a trade-off between the clock speed and the number of pipeline stages. Figure 3 shows the proposed architecture for the 9/7 fixed point DWT computation.

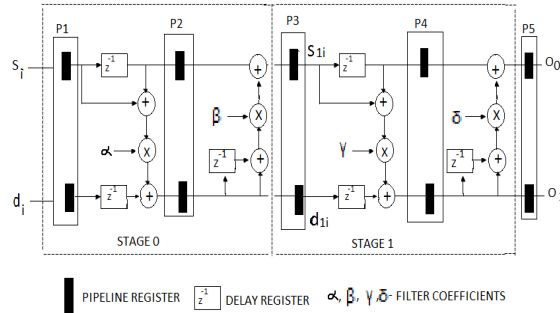


Figure3: Architecture for lifting scheme for fixed point 1-d DWT

In this architecture we advocated a five stage pipeline structure for the computation of 1-DDWT. The proposed structure is constrained by the nature of the DWT computation and is capable of optimizing the use of hardware resources.

In this five stage pipeline structure, all stages need to share the computation. Hence, all the stages need to be synchronized with one other. The pipeline registers in this proposed architecture are used in better way to optimize the use of hardware resources. Every stage performs by operating on the data produced by the previous stage. In this section we present the design of proposed five stage pipeline architecture by using 9 by 7 filter coefficients, pipeline registers and delay elements. The design of this architecture mainly focused on fixed point DWT computation.

In order to show the efficiency of our architecture, several architectures are chosen for comparison. In the proposed architecture, the clock pulses required to compute outputs are less than those in the previous architectures. This is due to the sequential states required to complete the computation of each output.

The architecture uses fixed point representation for arithmetic. The bit width of data inputs of the first stage of the 1-D DWT is 11 bits. That include 1 sign bit, 8 integral bits and the fractional bits are chosen to be 2 bits. Ideally the pixel inputs are 8 bits but having 11 bits signed input makes the hardware more generic. Whereas the coefficients inputs have 1 sign bit, 1 integral bit and 7 fractional bits. The bit precision will grow after the multiplication and addition performed in each stage of the DWT. To reduce the propagation delays in the digital circuit, the fractional part of the multiplier output is truncated by 7 bits before performing the addition. In this way the first stage of the DWT data outputs will have 19 bits (1 sign, 16 integral, 2 fractional bits). Looking at the 9/7 DWT coefficients, we can say that the output never crosses 4 times that of the inputs so we are safe to truncate 7 integral bits from the first stage output and feed to the second stage input. Similar approach is applied on the second stage also. After the second stage the outputs will have 20 bits (1 sign, 17 integral and 2 fractional bits). Saturation is applied to clip the data outputs between 0 and 255 (the 8-bit pixel range).

6. PERFORMANCE EVALUATION AND FPGA IMPLEMENTATION

In order to evaluate the performance of the architecture resulting from the proposed scheme, we need to make use of certain metrics that characterize the architecture in terms of the hardware resources used and the computation time. A five stage pipelined architecture is implemented and the simulation result of that is shown in figure 4.

For simulation of this architecture, YUV image is applied as the input. The image is given in the form of array that is represented by [0: ROWS-1] [0: COLUMNS-1]. The image is splitted into two parts as [0: ROWS/2-1] [0: COLUMNS/2-1] and [0: ROWS/2-1][0:COLUMNS/2-1]. That image can be used in the verilog code by using fopen command. We get the output image in two ports-output low pass image:[0:ROWS-1][0:COLUMNS/2-1], output high pass image [0:ROWS-1][0:COL/2-1]. In the simulation it is observed that when the input applies, the output is obtained after five clock cycles.

The hardware resources used for the filtering operation are measured by the number of multipliers and the number of adders, and that used for the memory space and pipeline latches is measured by the number of registers. The hardware resources utilization is shown in figure 5. The computation time, in general, is technology dependent. However, a metric, which is independent of the technology used but can be utilized to determine the computation time, is the number of clock cycles consumed from the instant the first sample is inputted to the last sample outputted assuming a given clock-cycle period, for example, unity, as the latency of a MAC cell.

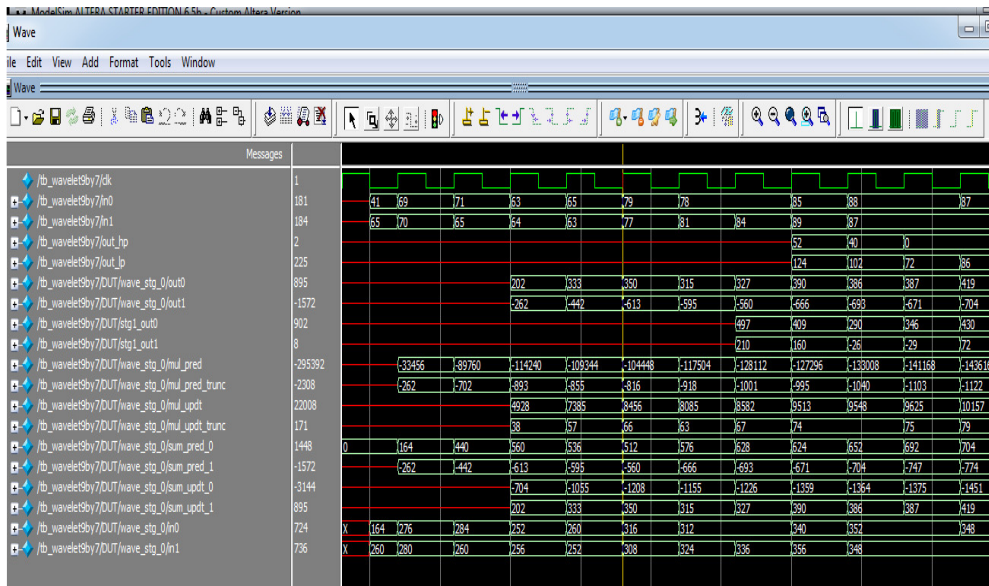


Figure 4: Simulation result for pipelining architecture

WAVE Project Status			
Project File:	wave.isc	Current State:	Synthesized
Module Name:	wavelet8ky7_top	• Errors:	No Errors
Target Device:	xc2vp7-7tp456	• Warnings:	22 Warnings
Product Version:	ISE, 8.1.03	• Updated:	24, ... 15:15:52 2012

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	158	4928	3%
Number of Slice Flip Flops	230	9856	2%
Number of 4 input LUTs	133	9856	1%
Number of bonded IOBs	70	248	28%
Number of MULT18K10s	4	44	9%
Number of GCLKs	1	16	6%

Detailed Reports					
Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	24, ... 15:15:52 2012	0	22 Warnings	48 Infos
Translation Report					
Map Report					
Place and Route Report					
Static Timing Report					
Bitgen Report					

Figure5: Hard ware resources utilization

```

Timing Summary:
-----
Speed Grade: -7

Minimum period: 8.280ns (Maximum Frequency: 120.769MHz)
Minimum input arrival time before clock: 1.418ns
Maximum output required time after clock: 5.593ns
Maximum combinational path delay: No path found

Timing Detail:
-----
All values displayed in nanoseconds (ns)

=====
Timing constraint: Default period analysis for Clock 'clk'
Clock period: 8.280ns (frequency: 120.769MHz)
Total number of paths / destination ports: 131358 / 178
-----
Delay:                8.280ns (Levels of Logic = 22)
Source:                wave_stg_1/sum_pred_1_pipe_0 (FF)
Destination:          stg1_out0_pipe_19 (FF)
Source Clock:          clk rising
Destination Clock:     clk rising

| Data Path: wave_stg_1/sum_pred_1_pipe_0 to stg1_out0_pipe_19
    
```

Figure6: Timing constraints for DWT computation

For the DWT computation, the comparison for the metrics mentioned before for various architectures are summarized in Table I. It is seen from the table that, compared to the architecture of [17], all the other architectures, including the proposed one, require approximately twice the number of clock cycles, except the architecture of [14], which requires four times as many clock cycles.

Table I: comparison of various architectures

Architecture	T _c (ns)
Parallel(13)	17.8
Systolic(14)	11.8
Pipelined(17)	11.8
DRU(18)	10.2

IP core (19)	11.8
Pipeline with parallelism (27)	8.7
Proposed	8.280

Table II: Resources Used in the FPGA devices

Resource	Used	Available in Total	Percentage used
CLB Slices	158	4928	3%
Flip Flop Slices	230	9856	2%
4-input LUTS	133	9856	1%
Bonded IOBs	70	248	28%

This performance of [17] is achieved by utilizing the hardware resources of adders and multipliers that are four times that required by the architecture of [14] and twice that required by any of the other architectures. In order to verify the estimated results, an implementation of the circuit is carried out in FPGA. Verilog is used for the hardware description and Xilinx ISE 8.2i for the synthesis of the circuit on a Virtex-II Pro XC2VP7-7 board. The implementation is evaluated with respect to the clock period (throughput) measured as the delay of the critical path of the MAC-cell network, and the resource utilization (area) measured as the number of configuration logic block slices, DFFs, lookup tables, and input/output blocks. The resources used by the implementation are listed in Table II. The circuit is found to perform well with a clock period as short as 8.280 ns. The clock period and timing constraints are shown in figure 6.

7. CONCLUSION

In this paper, a scheme for the design of pipeline architecture for a real-time computation of the fixed point 1-D DWT has been presented. The objective has been to achieve a low computation time by maximizing the operational frequency and minimizing the number of clock cycles required for the DWT computation, which, in turn, have been realized by developing a scheme for two lifting steps with 9 by 7 filtering and having five pipeline stages for the pipeline architecture.

A study has been undertaken, which suggests that, in view of the nature of the DWT computation, it is most efficient to map the overall task of the DWT computation to only five pipeline stages. There are two main ideas that have been employed for the internal design of each stage in order to enhance pipelining for DWT computation. The first idea was to decompose the filtering operation into two sub tasks that operate independently on the even- and odd-numbered input samples, respectively. This idea stems from the fact that the DWT computation is a two-sub band filtering operation, and for each consecutive decomposition level. Each subtask of the filtering operation is performed by a MAC-cell network with coefficients taken from 9 by 7 filters. The second idea employed for enhancing pipeline is to minimize the delay of the critical path.

In order to assess the effectiveness of the proposed scheme, pipeline architecture has been designed and simulated. The simulation results have shown that the architecture designed based on the proposed scheme requires the smallest number of clock cycles to compute output samples and a reduction of at least 60% in the period of the clock cycle in comparison to those required by the other architectures with a comparable hardware requirement. An FPGA based implementation of the designed architecture has been carried out, demonstrating the effectiveness of the proposed scheme for designing efficient and realizable architectures for the DWT computation.

Finally, the principle of pipelining architecture using lifting scheme presented in this paper for the design of architecture for the 1-D DWT computation is extendable to that for the 2-D DWT computation.

REFERENCES

- [1] S. Mallat, "A theory for multi resolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.11, no. 7, pp. 674–693, Jul. 1989.
- [2] J. Chilo and T. Lindblad, "Hardware implementation of 1D wavelet transform on an FPGA for infra sound signal classification," *IEEE Trans. Nucl. Sci.*, vol. 55, no. 1, pp. 9–13, Feb. 2008.
- [3] S. Cheng, C. Tseng, and M. Cole, "Efficient and effective VLSI architecture for a wavelet-based broadband sonar signal detection system," in *Proc. IEEE 14th ICECS, Marrakech, Morocco, Dec. 2007*, pp. 593–596.
- [4] K. G. Oweiss, A. Mason, Y. Suhail, A. M. Kamboh, and K. E. Thomson, "A scalable wavelet transform VLSI architecture for real-time signal processing in high-density intra-cortical implants," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 6, pp. 1266–1278, Jun. 2007.
- [5] K. Andra, C. Chakrabati, and T. Acharya, "A VLSI architecture for lifting-based forward and inverse wavelet transform," *IEEE Trans. Signal Process.*, vol. 50, no. 4, pp. 966–977, Apr. 2002.
- [6] 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.
- [7] M. Martina and G. Masera, "Multiplierless, folded 9/7–5/3 wavelet VLSI architecture," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 54, no. 9, pp. 770–774, Sep. 2007.
- [8] A. 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.
- [9] 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.
- [10] 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.
- [11] G. Shi, W. Liu, L. Zhang, and F. Li, "An efficient folded architecture for lifting-based discrete wavelet transform," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 56, no. 4, pp. 290–294, Apr. 2009.
- [12] C. T. Huang, P. C. Tseng, and L. G. Chen, "Efficient VLSI architectures of lifting-based discrete wavelet transform by systematic design method," in *Proc. IEEE ISCAS*, vol. 5, May 2002, pp. 565–568.
- [13] 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.
- [14] A. Grzeszczak, 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.
- [15] T. Acharya, C. Chakrabarti, A survey on lifting-based discrete wavelet transform architectures. *J. VLSI Signal Process.* 42, 321–339 (2006)

- [16] I. Daubechies, W. Sweldens, Factoring wavelet transform into lifting steps. *J. FourierAnal. Appl.* 4, 247–269 (1998)54 *Discrete Wavelet Transforms: Algorithms and Applications VLSI Architectures of Lifting-Based Discrete Wavelet Transform* 15
- [17] 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.
- [18] Chengjun Zhang, Chunyan Wang and M. Omair Ahmad, “A Pipeline VLSI Architecture for High-Speed Computation of the 1-D Discrete Wavelet Transform” *IEEE Trans. Circuits Systems I*,pp.1529-8328, Feb 2010.
- [19] An Efficient Architecture for 2-D Lifting-based Discrete Wavelet Transform PingpingYu, Suying Yao, JiangtaoXu School of Electronic and Information Engineering Tianjin University Tianjin, China,p.p-978-1-4244-2800-7/09-IEEE-2009
- [20] VLSI Architectures of Lifting-Based Discrete Wavelet Transform Sayed Ahmad Salehi and Rasoul Amirfattahi Isfahan University of Technology, Department of Electrical and Computer Engineering, Digital Signal Processing Research Lab., Isfahan Iran.
- [21] A High-Performance and Memory-Efficient Pipeline Architecture for the 5/3 and 9/7 Discrete Wavelet Transform of JPEG2000 Codec Bing-Fei Wu, Senior Member, IEEE, and Chung-Fu Lin, Student Member, IEEE -IEEE Transactions on Circuits and systems for video technology, vol.15,no.12,December 2005
- [22] A Rescheduling and Fast Pipeline VLSI Architecture for Lifting-based Discrete Wavelet Transform Bing-Fei Wu and Chung-Fu Lin Department of Electrical and Control Engineering National Chiao Tung University 1001 Ta Hsueh Road, Hsinchu, Taiwan, 30050, R.O.C
- [23] Lifting Based Discrete Wavelet Transform Architecture for JPEG2000 Chung-JrLian, Ktian-Ftr Chen, Hong-Hui Chen, and Liang-Gee Chen DSP/AC Design Lab., Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan, R.O.C.
- [24] 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.
- [25] T. Park, “Efficient VLSI architecture for one-dimensional discrete wavelet transform using a scalable data recorder unit,” in *Proc.ITC-CSCC*, Phuket, Thailand, Jul. 2002, pp. 353–356.
- [26] S. Masud and J. V. McCanny, “Reusable silicon IP cores for discrete wavelet transform applications,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 6, pp. 1114–1124, Jun. 2004.
- [27] C.-T. Huang, P.-C.Tseng, L.-G. Chen, Flipping structure: an efficient VLSI architecture for lifting based discrete wavelet transform, *IEEE Trans. Signal Process.* 52 (2004), pp.1080–1089
- [28] K. K. Parhi and T. Nishitani, “VLSI architectures for discrete wavelet transforms,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 1,no. 6, pp. 191–202, Jun. 1993.
- [29] M. Vishwanath, R. M. Owens, and M. J. Irwin, “VLSI architectures for the discrete wavelet transform,” *IEEE Trans. Circuits Systems II, Analog.Digit. Signal Process.*, vol. 42, no. 5, pp. 305–316, May 1995.
- [30] C. Cheng and K. K. Parhi, “High-speed VLSI implementation of 2-Ddiscrete wavelet transform,” *IEEE Trans. Signal Process.*, vol. 56, no.1, pp. 393–403, Jan. 2008.

- [31] VLSI Implementation of Discrete Wavelet Transform (DWT) for Image Compression Abdullah AlMuhit, Md. Shabiul Islam and Masuri Othman* Faculty of Engineering Multimedia University (MMU) Jalan multimedia, Cyberjaya, Selangor 63100, Malaysia.

Authors

K. Durga Sowjanya was born in Koyyalagudem (Andhra Pradesh). She received B.Tech in Electronics and Communication Engineering from Jawaharlal Nehru Technological University, Kakinada. She is currently pursuing M.Tech from Sri Vasavi Engineering College, J.N.T University, Kakinada, India. Her research interest includes image and signal processing algorithms and VLSI architecture development. She did her Master's thesis in VLSI Architecture for Computation of 1-D DWT.



Mr.K.N.H.Srinivas, received the B.Tech., degree in electronics and communication engineering from S.V.H. College of Engineering, Machilipatnam, Nagarjuna University and Completed M.Tech (Electronic Instrumentation) at National Institute of Technology, Warangal, India. He is currently working as Head of the Department of Electronics and Communication Engineering, Sri Vasavi Engineering College. He is a fellow IETE &ISTE and guided eight post graduate students so far. He published eight research papers in reputed international journal and international conferences.



Venkata Ganapathi Puppala, received the B.Tech., degree in electronics and communication engineering from JNTU University, Hyderabad, India, in 2005, and the M.Tech degree in electronics instrumentation engineering from the National Institute of Technology, Warangal, India, in 2007. In 2007, he joined iChip Technologies, Hyderabad. And worked as ASIC Design engineer and involved development of Application Specific Instruction Set Processors for Video Encoder and Decoders for H.264/MPEG-4 AVC, VC1 standards. Later he joined in Quartics Technologies, Pune where he is currently working as ASIC Design engineer and involved in ASIP architectures for computer vision, video post processing and 2D video to 3D video conversion algorithms.

