International Journal of Electronics and Electical Engineering

Volume 3 | Issue 3 Article 3

January 2015

A NOVEL VLSI ARCHITECTURE OF HIGH SPEED 1D DISCRETE WAVELET TRANSFORM

POOJA GUPTA
Dept. of ECE, MITS (Deemed University), Lakshmangarh, Rajasthan, India, guptapooji@gmail.com

Saroj Kumar Lenka
Dept. of IT, MITS (Deemed University), Lakshmangarh, Rajasthan, India, lenka.sarojkumar@gmail.com

A NOVEL VLSI ARCHITECTURE OF HIGH SPEED 1D DISCRETE WAVELET TRANSFORM

POOJA GUPTA[1], SAROJ KUMAR LENKA[2]

[1]Dept. of ECE, [2]Dept. of IT, MITS (Deemed University), Lakshmangarh, Rajasthan, India
E-mail: guptapooji@gmail.com, lenka.sarojkumar@gmail.com

Abstract- This paper describes an efficient implementation for a multi-level convolution based 1-D DWT hardware architecture for use in FPGAs. The proposed architecture combines some hardware optimization techniques to develop a novel DWT architecture that has high performance and is suitable for portable and high speed devices. The first step towards the hardware implementation of the DWT algorithm was to choose the type of FIR filter block. Firstly we design the high speed linear phase FIR filter using pipelined and parallel arithmetic methods. This proposed filter employs efficiently distributed D-latches and multipliers. Furthermore this filter is used in the proposed DWT architecture. Thus, the new VLSI architecture based on combining of fast FIR filters for reducing the critical path delay and data interleaving technique for lower chip area. We synthesized the final design using Xilinx 9.1i ISE tool. We illustrate that a DWT design using a pipelined linear phase FIR filter coupled with data-interleaving gives the best combination of the performance metrics when compared to other DWT structures.

Keywords- *Discrete Wavelet Transform, Fast Convolution, Finite impulse response filter, Field-programmable gate array, Very Large Scale Integration.*

I. INTRODUCTION

The implementation of Finite Impulse Response (FIR) filtering could be a basic demand in several Digital signal and image processing systems. Last decade, significant attention has been given to the VLSI implementation of signal processing algorithms, starting from full custom VLSI to general purpose digital signal processors. The essential Discrete Wavelet Transform (DWT) is often accomplished by convolution-based implementation using the FIR-filters to do the transform [1]. FIR filters are unit applicable for improving the performance of the DWT hardware design [2]. A range of approaches for high speed implementation of FIR filters are conferred [3-10]. Novel area-efficient architectures of FIR filters are proposed by Benkrid and Benkrid [11]. P. Bougas designed the pipelined multiplier factor array based FIR filter and reduced hardware [12].In order to achieve high performance, pipeline and parallel implementation ways are applied. Pipelining reduces the effective critical path by introducing pipelining latches on the info path [3].

The Discrete wavelet transform (DWT) is a multiresolution tool with excellent characteristics in the time and frequency domains. In the DWT, signals can be decomposed into different subbands with both time and frequency. Since the introduction of the DWT by Mallet [1] in 1989 has become one of the most widely used mathematical tools in image compression, speech analysis and pattern recognition. In recent years, a large number of algorithms and implementation structures for the computation of the DWT have been proposed [2], [14-17]. These implementations use efficient VLSI architectures in order to meet the high-speed requirement of real-time applications of the DWT computations and to improve the hardware performance. Most hardware implementations optimized two or more essential designs to improve their performance in terms of area, speed or power dissipation. Lower area and high speed VLSI implementation is the prime concern in the portable and real-time DSP application.

The basic DWT can be realized by convolution based implementation using the FIR filters The input sequence x(n) is convolved with the filters H(z) and G(z) and the outputs obtained at each level are decimated by a factor of two. After down sampling, alternate samples of the output sequence from the low pass filter and high pass filter are dropped. This reduces the time resolution by half and conversely doubles the frequency resolution of two. The computation of the output sequences YL and y_H from the low pass filter and the high pass filter can be represented with the following equations:

$$y_H(n) = \sum_{i=0}^{T_h-1} G(i).x(2n-i) \qquad (1)$$

$$y_L(n) = \sum_{i=0}^{T_l-1} H(i).x(2n-i) \qquad (2)$$

Baganne [13] presents the three level DWT design, modeled at the register transfer level (RTL), was implemented using binary tree structure. The hardware implementation of the multilevel DWT architecture as shown in Figure 1 is realized by a number of cascaded filter blocks followed by scaling. The advantage of this design is the minimal clock latency and less complexity. The drawbacks of this design are large critical path delay and large design area.

DWT architecture can be designed by employing a single processing element, in which computations are performed by interleaving the data from successive levels of decomposition [14]. Interleaving method reduces the design area but may increase the critical path delay depending on the filter structure and length. Pipelining reduces the critical path delay but may increase the latency resulting degradation in throughput performance for real-time applications. A similar method to reduce area and improve throughput was proposed in [15] wherein the highpass and lowpass filter coefficients were interleaved to reduce the number of multipliers rather than interleaving the data. A folded DWT structure has been proposed [16] for storing the intermediate outputs from each level of decomposition in a memory block before processing once more by the same DWT block.

The folded structures exhibit a large latency due to the successive levels of decomposition are interleaved with the preceding levels and having high complexity in the control logic. The advantage of such designs is the less critical path that performs the multiply-accumulate operations and a reduction in chip area.

Two processing units for the multi level DWT architecture have been proposed in [17] and applied the folding scheme to the second level of decomposition and beyond. An efficient VLSI implementation of the 1D DWT architecture is presented in [2] in order to improve throughput, latency and power dissipation but design is more complex.

We propose novel high speed architecture for convolution based 1D DWT. We were applying, the multiple low level optimizations in an organized way to improve the performance and reduce the complexity of the design. The first optimization is for reducing the critical path delay by using high speed FIR filters in place of direct form FIR. In this filter for reducing the critical path delay, pipeline and parallel arithmetic techniques are used. For reducing area, we have to take into account the symmetric filter coefficients for FIR filter and to scale back the latency; we have used the pipelined binary adder tree in the place of conventional one. And the second optimization introduces data interleaving to improve the hardware area utilization of overall DWT architecture.

The rest of this paper is organized in this way: In Section two optimizations of FIR filter for DWT architecture discussed, Section three presents the new VLSI design for convolution based 1D DWT architecture, Section four includes the performance evaluation of various DWT architectures and Section five presents the conclusion.

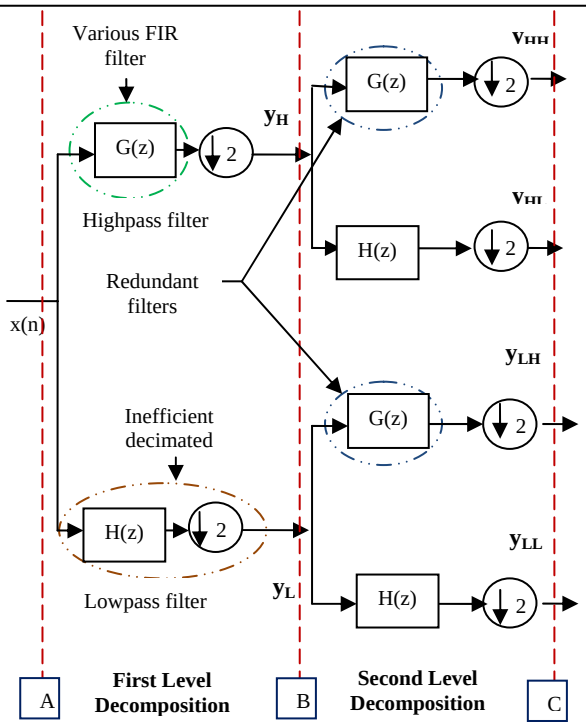

Figure 1. 2-level Convolution based DWT decomposition

II. OPTIMIZATION OF FIR FILTER STRUCTURES

FIR filters are essential building blocks for DSP algorithms such as signal and image compression techniques that rely on DWT blocks. The filter blocks generated by the Performance Analysis Framework (PAF) [7] provided a starting point for the implementation of the DWT. It gave an estimate of the latency, throughput and the area utilized by the filter which assisted us in the selection and analysis process.

A. Basic Architectures

An FIR filter is defined by its impulse response coefficients, h(n), with h(n) = 0 for n < 0 and n > M, where M is the filter order. The filter output y(n) is given by the convolution of h(n) with the input sequence x(n):

$$y(n) = h(n) * x(n) \qquad (3)$$

Expanding the convolution gives the difference equation of the so-called direct form [5]:

$$y(n) = \sum_{k=0}^{M} h(k).x(n-k) \qquad (4)$$

Where h(k) is the impulse response of the filter, and the filter has M +1 coefficients. The direct form is shown in Figure 2. If T_A, T_M and T_{at} are the delays of the adder, multiplier and adder tree, respectively, then the critical path of the direct form filter is T_M+T_{at}. The graph reversal theorem can be exploited to obtain the so-called transposed form [5], which is shown in Figure 3. The transposed form has the significant advantage that its critical path is greatly reduced to T_M+T_A, since its adders have only two operands and

are followed by a register. In designing frequency-selective digital filters, it is usually desirable to have approximately constant frequency-response magnitude and minimum phase distortion [6].

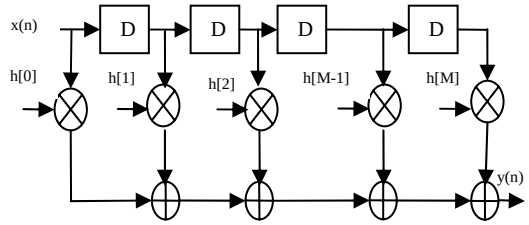

Figure 2. M-tap Direct form (DF) FIR filter structure

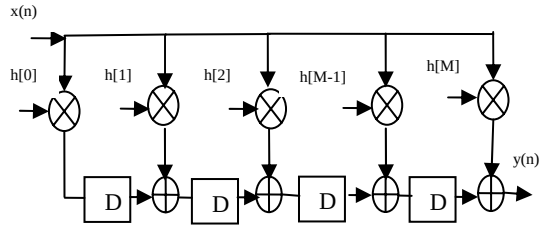

Figure 3. M-tap Transpose form (TF) FIR filter structure

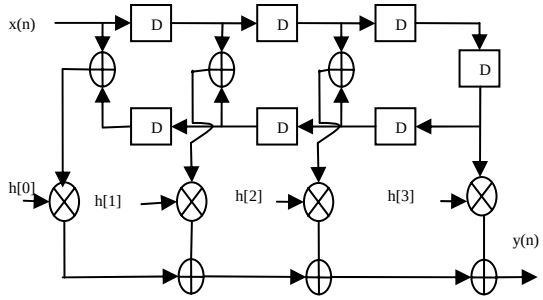

Figure 4. 8-tap Linear–phase FIR filter structure

A linear phase with integer slope is replaced by a simple delay in the time domain and it reduces the phase distortion to a minimum in the frequency domain. Therefore, it is required to design digital filters with exactly or approximately linear phase. Figure 4 shows the linear phase FIR filter. A linear-phase FIR filter of order N is characterized by symmetric impulse response [6]

$$h[n] = h[N-n], \quad (3)$$

For 8-tap FIR filter, direct and transpose form required 8 multipliers and 7 adders, whereas linear-phase requires 4 multipliers and 7 adders.

B. Pipelined Architectures

Pipelining reduces the effective critical path by introducing pipelining latches along the data path. Pipelining transformation leads to a reduction in the critical path, which can increase the clock speed (or sample speed). K.Parhi [5] presented pipelined FIR filter by using feed-forward cutset method. The critical path is now reduced from $T_M + 2T_A$ to T_M+T_A, where T_M is the time taken for multiplication and T_A is the time needed for an addition operation. The pipelining latches can only be placed across any feed-forward cutset of the graph. Ramsey S. Hourani [7] used the FIR architecture in low power equalizer, in which the feed forward cutset is used in the multiplier data path, by placing a register at the output of each multiplier. This decouples the multiplication and addition delays, and it allows a more equal comparison with the direct form. The timing of each multiplier may differ therefore, for giving the data to the adder at the same time, D latches are used, as a result, the speed of the circuit is improved.

Direct form (DF) architectures are suitable for area sensitive small filter orders while transposed structures are suitable for large filter orders. However, according to [9] replacing the multi-operand adder in the direct form with the pipelined binary adder tree will achieve a very high speed FIR filter. This arrangement is most efficient when the number of filter coefficients is an integer power of two. Otherwise, some terms must be delayed prior to addition.

M. Maamoun [10] presented very high speed FIR filter architecture, which is based on combining pipelining and parallel arithmetic methods. In this architecture, efficiently distributed D-latches and multipliers are used, thus this critical path is reduced to Tm/2 (if $T_A < T_M/2$).

We propose a new high speed linear phase FIR filter in which pipelining and parallel arithmetic methods are used [10].Thus, using equal amount of multipliers high speed architecture can be achieved compared to DF. This fast FIR filter is a combination of alternate multiplication and binary adder tree as shown in Figure 5. In the alternate multiplication for a given clock, the input signal is divided into even and odd signal, D_{m1} is triggered for odd part and D_{m2} is triggered for even part. To increase the processing frequency, D_{m1} and D_{m2} D-latches work at the same frequency ($F_{m1} = F_{m2}$) and the input and output D-latches work with the same phase and frequency (F_d).Where,

$$F_d = 2 * F_{m1} \quad (5)$$

TABLE I COMPARISON OF 8-TAP FIR STRUCTURES

Architectures	Mul.(s)	Add.(s)	Reg.(s)	Delay (ns)
DF FIR	8	7	9	30.904
Linear Phase FIR	4	7	9	21.120
FIR [9]	8	7	23	11.034
Fast FIR [10]	16	7	59	5.045
Proposed FIR	8	7	23	5.045

Table I depict the requirement of multipliers, adders, registers and the critical path delay for the individual filter structure. The proposed design needs equal amount of multipliers and adders in comparison to direct form but requires an approx twice number of more registers.

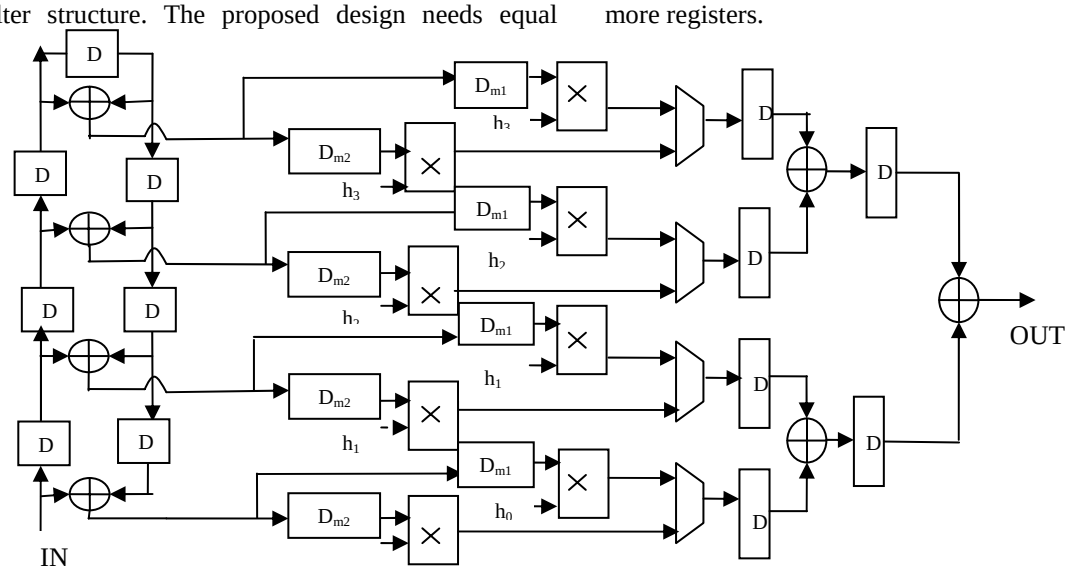

Figure 5. The proposed fast finite response (FIR) architecture

We observe that the DF FIR filter has highest delay. In FIR filter [9], the multiple products are rearranged with binary tree of adders. This decouples the multiplication and addition delays, by this delay is reduced by 64% Fast FIR filter [10] and proposed architecture uses pipelining and parallelism thus the delay is reduced by 83% as compared to DF.

C. Data- Interleaving

Data interleaving helps process multiple independent signals using a single filter structure [18]. If two signals are filtered independently by identical filters, we can replace two filters by a single filter. Though interleaving will increase the registers in the filter, while using the same number of multipliers and adders. To reduce the redundant filter blocks in the second level decomposition in Figure 1 we employ a data-interleaving scheme wherein multiple inputs from different sources are combined in an alternating manner to generate a single input stream. But this must need perfect timings to confirm that the interleaved signals are processed correctly. The same methodology can be applied for third level decomposition. This technique may reduce the throughput of the filter but the decimated FIR filters in the DWT algorithm makes data-interleaving possible with negligible change to the overall design throughput.

III. PROPOSED 1D DWT ARCHITECTURE

We construct the FIR filter structures for each level of decomposition by using the framework discussed in [8]. In the framework user has to enter the design parameters which are essential in the construction of specialized filter. The first step is the input signal is divided into highpass and lowpass signal. In the previous work [2] direct form FIR filters are used in the polyphase structures, but we are using the proposed fast FIR filter so the overall high speed design can be achieved. Decimation by two discards every alternate sample computed by the filter. Thus interleaved the two signals by alternating each signal same as input data rate. Then the interleaved signal was processed by single lowpass and highpass filter shown in Figure 6.

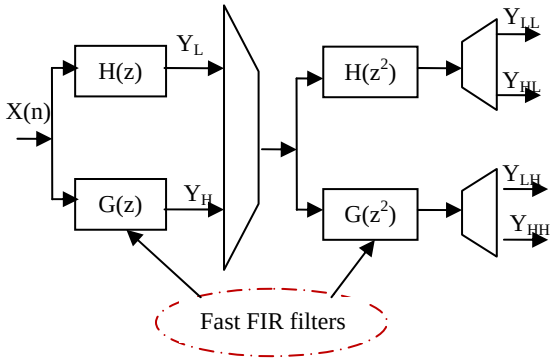

Figure 6. Proposed Data-Interleaved Structure for 1D DWT with Fast FIR filter

The same interleaved technique is used at the output of the second level decomposition and then used for the third level decomposition. At the final stage final output was deinterleaved and obtained eight separate channels, each at one eighth the original data rate. Further, this method can be extended to higher levels of decomposition. Figure 1 uses six filter structures to implement a 2-level DWT. For level J, the number of filters required are 2^J . However, for levels two and higher, we observe that multiple inputs are filtered through the same low pass and high pass filters. For example, y_L and y_H are filtered by the same filter G(z) in level 2. Hence, there are redundant filters within the design, two redundant filters for level 2 and six

for level 3; that is (2^J - 2) redundant filters to each J level. We can remove these redundant filters within the design by using data-interleaving techniques. This results in significant savings in the hardware utilization. The architecture after implementing data-interleaving is illustrated in Figure 6.

Since y_L and y_H outputs of level 1 pass through the same filters, H(z) and G(z), we interleave these signals using a multiplexer and then proceed to filter them. In Fig.1 the outputs of the low pass filters in level 2 are y_{HL} and y_{LL}. The same outputs are obtained in Figure 6 by splitting the interleaved output using a demultiplexer. Thus data interleaving method transforms the binary tree structure comprising of 14 filters into a structure comprising of 6 filters by using simple control logic. In Figure 6 G(z) and H(z) blocks presents the proposed fast FIR filter. As a result overall design is efficient in the terms of area and speed.

IV. PERFORMANCE EVALUATION

We synthesized all the design in different Xilinx FPGA Devices for the verification [19, 20]. The basic metrics, we considered in the implementation of length-8 FIR filter, are critical path delay and resource utilization.

A. Delay Analysis

As we described that R. Hourani [2] presents the efficient DWT architecture in the terms of area and delay. In this architecture pipelined direct form FIR filter is used in the each level. Table II displays the delay results for each design.

TABLE II DELAY ANALYSIS OF VARIOUS DWT ARCHITECTURES ON XILINX FPGAS DEVICES

Devices Architectures	Spartan-3 XC3S2000	Spartan-3E XC3S500	Virtex-5 XCVLX
DWT using pipelined DF filter	17.071 ns	16.635 ns	15.569 ns
DWT using pipelined TF filter	9.611 ns	8.854 ns	5.643 ns
Proposed DWT	4.033 ns	3.521 ns	2.189 ns

In the proposed design we have used linear phase fast FIR having a minimum critical path delay as shown in Table I. DWT using DF filter has critical path T_M+7T_A, where as DWT using TF has T_M+T_A. And proposed design has only $T_M/2$. Figure 7 shows the graph of delay. From the graph we can conclude that the proposed design has minimum delay and design using pipelined direct form FIR has the largest delay. If a pipelined TF filter is used then delay can be reduced by 56.29 % in Spartan-3, 50.69 % in Spartan-3E and 63.62 % in Virtex-5. When comparing proposed design from the first design then delay is reduced by 76.40 % in Spartan-3, 78.84 % in Spartan-3E, 85 % in Virtex-5 and comparison to the second design our design is fast by 41.93 % in Spartan-3, 39.77 % in Spartan-3E, 38.63 % in Virtex-5. Thus we obtain very high speed DWT architecture.

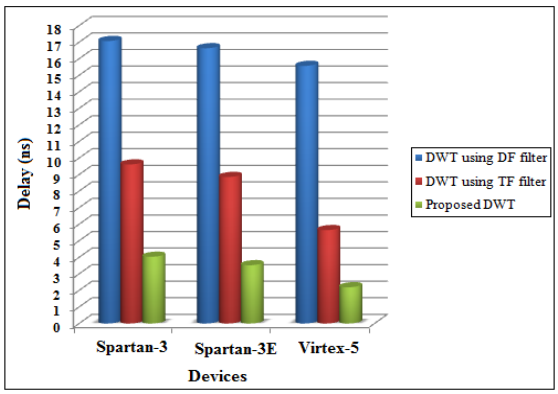

Figure 7. Delay Analysis of DWT designs at different devices

B. Area Analysis

For area analysis we implemented all the designs for 16-tap, 2-level, 4 channel on the Xilinx Virtex-5 FPGA. Table III displays the number of slice registers, LUTs, bonded IOBs and DSPs. LUT stand for Look-Up Table, and a LUT is a one bit wide memory array. Virtex-5 FPGA function generators are implemented as 4-input LUTs. IOBs are input, output blocks that provides interface between the package pins and the internal logic.

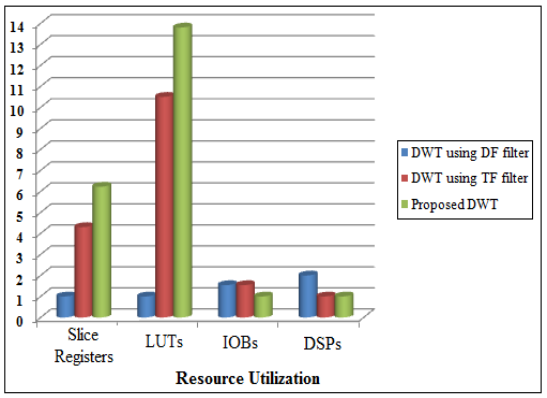

Figure 8. Normalized Area Graph for various DWT Architectures

TABLE III RESOURCE UTILIZATION FOR 16-TAP, 2-LEVEL DWT ARCHITECTURES

Architectures	Slice Registers	LUTs	IOBs	DSP48Es
DWT using pipelined DF filter	196	192	182	64

DWT using pipelined TF filter	848	2012	182	32
Proposed DWT	1224	2657	118	32

Figure 8 shows a graph of the normalized area. The resources are normalized individually for registers, LUTs, IOBs and DP48Es. We observe that polyphase DWT architecture using pipelined DF requires minimum number of Slice registers and LUTs.

Proposed design requires 35 % less IOBs and 50 % less DSPs than the first design. This is a significant reduction since there are limited numbers of DSP48 slices on an FPGA and should be prudently used. Design using pipelined transpose form has the same number of DSPs as the proposed but also required 35 % more IOBs. The second design reduces critical path to around 55 % on an average but requires approximately four times more registers and ten times more LUTs than the first design. The proposed design reduces critical path to 80 % on an average but it needs approximately 6 times more registers and 12 times more LUTs than first design and 30 % more registers, 24 % more LUTs than the second design. There is a tradeoff between area and delay. If we required high speed then we have to compromise to the slightly more area.

V. CONCLUSION

This paper presents the fast FIR filter and proposed the VLSI architecture of a multi-level 1D DWT using some optimization techniques. The optimizations considered for this work includes the data interleaving for the reduced hardware optimization techniques and applied pipelining and parallel methods on the FIR filter structure for reducing the critical path delay. Thus the presented architecture for 1D discrete wavelet transform is efficient in the terms of speed and area. We implemented a 16-tap, two-level, four channel convolution based DWT architecture using Xilinx ISE 9.1i tool. We compared our implementation to designs with high-throughput and low area. As a result we obtain that a new VLSI architecture for DWT is intended for high speed signal and image processing. The 2-D DWT can be implemented on an image by a succession of row-wise and column-wise 1-D filtering. Therefore, for future work an effective methodology to optimize the 2-D hardware architecture could be explored using the methodology we developed as a basis.

REFERENCES

[1] S. G. Mallat, “A theory for multiresolution signal decomposition: the wavelet representation," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 11, no.7, pp. 674-693, July 1989.

[2] R. Hourani, I. Dalal, W.R. Davis, “An Efficient VLSI Implementation for the 1D Convolution Discrete Wavelet Transform”, Circuits and Systems, MWSCAS 2008, 51st Midwest Symposium, 10-13 Aug, pp. 870-873, 2008.

[3] J. B. Evans, “An efficient FIR filter architecture,” IEEE International Symposium on Circuits and Syst., pp.627-630, May 1993.

[4] R. Hourani, Y. Kim, S. Ocloo, and W. Alexander, “Automated hardware IP generation for digital signal processing applications," in Signals, Systems and Computers, 2006. ACSSC '06. Fortieth Asilomar Conference, pp. 1156-1160, Nov. 2006.

[5] Keshab K. Parhi, VLSI Digital Signal processing systems, Published by John Wiley & Sons Inc., ISBN 978-81-265-1098-6, 2009.

[6] Håkan Johansson, “Fractional-Delay and Supersymmetric Mth-Band Linear-Phase FIR Filters Utilizing Partially Symmetric and Antisymmetric Impulse Responses”. IEEE Trans. on Circuits And Syst. II,Express Briefs, vol. 59, no. 6, June 2012.

[7] Ramsey S. Hourani, A Performance Analysis Framework for the Design of DSP Systems, published by ProQuest, 2008.

[8] Kumm, M. Zipf, P. “High Speed Low Complexity FPGA-based FIR Filters Using Pipelined Adder Graphs,” IEEE International Conference on Field-Programmable Technology (FPT), pp. 1-4, Dec. 2011.

[9] J.M. Pierre Langlois “Design and Implementation of High Sampling Rate Programmable FIR Filters in FPGAs,” IEEE Circuits and System, Northeast Workshop 2006, pp.37-40, June 2006.

[10] Maamoun, M.; Bradai,R.; Meraghni,A.; Beguenane, R. “Low Cost VLSI Discrete Wavelet Transform and FIR Architecture for Very High-Speed Signal and Image Processing,” Cybernetic Intelligent Systems (CIS), IEEE 9th International Conference 2010, pp. 1-6, DOI:10.1109/UKRICIS.2010.5898088, Sept.2010.

[11] A. Benkrid and K.Benkrid, "Novel Area-Efficient FPGA Architectures for FIR Filtering With Symmetric Signal Extension" IEEE Trans. on Very Large Scale Integration (VLSI) Systems, vol. 17, no. 5, pp. 709-722, May 2009.

[12] P.Bougas and P.Kalivas “Pipelined Array-Based FIR Filter Folding” IEEE Trans. on Circuits and Systems-I, vol. 52, no.1, pp. 108-118, Jan. 2005.

[13] A. Baganne, I. Bennour, M. Elmarzougui, R. Gaiech, and E. Martin, “A multi-level design flow for incorporating IP cores: case study of 1d wavelet IP integration,” Design, Automation and Test in Europe Conference and Exhibition, 2003, pp. 250–255, 2003.

[14] T. C. Denk and K. K. Parhi, “Systolic VLSI architectures for 1-d discrete wavelet transforms,” in Signals, Systems & Computers, 1998. Conference Record of the Thirty-Second Asilomar Conference on, vol. 2, (Pacific Grove, CA), pp. 1220–1224, Nov. 1–4, 1998.

[15] F. Marino, D. Guevorkian, and J. T. Astola, “Highly efficient highspeed/ low-power architectures for the 1-d discrete wavelet transform,” IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 47, pp. 1492–1502, Dec. 2000.

[16] A. B. Premkumar and A. S. Madhukumar, “An efficient VLSI architecture for the computation of 1-d discrete wavelet transform,” in International Conference on Information, Communications and Signal Processing, 1997 ICICS., pp. 1180–1184, Sept. 9–12, 1997.

[17] C. Zhang, C. Wang, and M. O. Ahmad, “A VLSI architecture for a highspeed computation of the 1d discrete

wavelet transform," in Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on, pp. 1461–1464, May 23–26, 2005.

[18] M. Dabbagh and W. Alexander, "Pipelining of digital filter structures for VLSI implementation," in Southeastcon '89. Proceedings. 'Energy and Information Technologies in the Southeast'., IEEE, (Columbia, SC), pp. 1185–1189, Apr. 1989.

[19] "Xilinx, Virtex-5 Multiplatform FPGA." http://www.xilinx.com/virtex5/".

[20] Virtex-5 FPGA Family, Data Sheet, DS100 (v5.0), February 6, 2009 .Online www.xilinx.com/support/documentation/data_sheets/ds100.pdf.