

Approximation Method for High Speed Multiplier-Less Dwt Architecture

Mrs. Geetha K¹

¹(Department of Instrumentation Technology, JSSATE, Bangalore, India)

Abstract : This paper presents a VLSI design approach for a high speed and real time Discrete Wavelet Transform computing. The hardware requirement is a major concern in the computation of discrete wavelet transform. There are many multiplier-less architecture for DWT for reducing the hardware requirement. But it is observed that the approximation method for constant multiplier implementation in DWT can increase the speed and reduce the hardware requirement for the computation of Discrete Wavelet Transform.

Keywords – DWT, BBRF, Fast-convolution, Lifting.

I. INTRODUCTION

The discrete wavelet transform (DWT) has gained wide popularity due to its excellent decorrelation property. Many modern image and video compression systems embody the DWT as the transform stage. It is widely recognized that the 9/7 filters are among the best filters for DWT-based image compression. In fact, the JPEG2000 image coding standard employs the 9/7 filters as the default wavelet filters for lossy compression. Several recent publications describe efficient implementation of JPEG2000 encoders and decoders [1].

Moreover, many research works have faced the problem of reducing the DWT complexity. This issue has been investigated mainly from two perspectives: 1) reducing the memory access overhead 2) reducing the DWT computational complexity.

Lifting and convolution present the two computing approaches to achieve the discrete wavelet transform. While conventional lifting based architectures require fewer arithmetic operations compared to the convolution-based approach for DWT, they sometimes have long critical paths. If T_a and T_m are the delays of the adder and multiplier, respectively, then the critical path of the lifting based architecture for the (9, 7) filter is $(4 \times T_m + 8 \times T_a)$, while that of the convolution implementation is $(T_m + 2 \times T_a)$. In addition to this and for the reason to preserve proper precision, intermediate variables widths are larger in lifting-based computing. As a result, the lifting multiplier and adder delays are longer than the convolution ones. Hence convolution is a best method to reduce the delays in the computation of DWT [2]. But, the traditional advantages of lifting implementations are 1. Lifting leads to a speed-up when compared to the standard implementation. 2. Lifting allows for an in-place implementation of the fast wavelet transform, a feature similar to the Fast Fourier Transform. This means the wavelet transform can be calculated without allocating auxiliary memory. 3. All operations within one lifting step can be done entirely parallel while the only sequential part is the order of the lifting operations. 4. Using lifting it is particularly easy to build non linear wavelet transforms. Typical examples are wavelet transforms that map integers to integers. Such transforms are important for hardware implementation and for lossless image coding. 5. Using lifting and integer-to-integer transforms, it is possible to combine biorthogonal wavelets with scalar quantization and still keep cubic quantization cells which are optimal like in the orthogonal case. In a multiple description setting, it has been shown that this generalization to biorthogonality allows for substantial improvements. 6. Lifting allows for adaptive wavelet transforms [3].

In the present era, efficient implementation of DWT using 9/7 filters in resource-constrained hand-held devices with capability for real-time processing of the computation-intensive multimedia applications is, a necessary challenge.

In computation of DWT the conventional multipliers adds delay to the architecture and also utilizes more hardware. Multiplier-less hardware implementation approach provides a kind of solution to this problem due to its scope for lower hardware-complexity and higher throughput of computation [4].

Several designs have been proposed for the multiplier-less implementation of DWT based on the principle of distributed arithmetic (DA) [5] - [9]. The structure of [5] and [6] distributes the bits of the fixed coefficients instead of the bits of input samples. Consequently, the adder complexity of the structure of [5] and [6] depends on the DA-matrix of the fixed coefficients. Longa et al [9] have suggested an LUT-less DA-based design for the implementation of 1-D DWT. They have eliminated the ROM cells required by the DA based structures at the cost of additional adders and multiplexers. The adder complexity of this structure is significantly higher than the other multiplier-less structures [4]. Gaurav Tewari et al [10] has proposed scalable Polyphase Structure with DA for JPEG2000, and achieved 129.8 MHz speed. In [11] the Filter coefficients of the biorthogonal 9/7-5/3 wavelet low-pass filter are quantized before implementation in the high speed

computation hardware. In this proposed architectures, all multiplications are performed using less shifts and additions, which significantly reduced the adder complexity of the 9/7 DWT. In the pipelined architecture of lifting scheme Zhigang et al [12] showed that the arithmetic could be improved by combination of tree-type arrangement and Horner’s rule for the accumulation of partial products in multiplication. Horner’s rule is used for partial product accumulation to reduce the truncation error; Tree-Height Reduction is used for Latency Reduction.

Conventionally, programmable DSP chips are used to implement DWT algorithms for low-rate applications and the VLSI application specific integrated circuits (ASICs) for higher rates. The FPGAs are programmable logic devices that provide sufficient quantities of logic resources that can be adapted to support a large parallel distributed architecture [2]. In the present paper the approximation method proposed by Martina et al [1] is used to reduce the hardware requirement and to increase the speed of the architecture in both convolution and lifting based DWT.

The rest of the paper is organized as follows. Section II & III gives the basics of convolution and lifting based DWT respectively. Section IV describes the multiplier-less architecture for fast convolution based DWT and section V describes the multiplier-less architecture for pipelined architecture of Lifting-Based DWT. The performance evaluation of the architecture are presented in section VI, finally, section VII presents a conclusion for this paper.

II. CONVOLUTION BASED DWT

The basic DWT can be realized by convolution-based implementation using the FIR-filters to do the transform.

The input discrete signal X(n) is filtered by a low-pass filter (h) and a high-pass filter (g) at each transform level. The two output streams are then sub-sampled by simply dropping the alternate output samples in each stream to produce the low pass sub band YL and high-pass sub band YH. The associated equations can be written as (1) [2]. Fig 1 shows the signal analysis in one dimensional (1-D) Discrete Wavelet Transform.

$$y_L(n) = \sum_{i=0}^{N/2-1} h(2n - i) x(i), \quad y_H(n) = \sum_{i=0}^{N/2-1} g(2n - i) x(i) \quad (1)$$

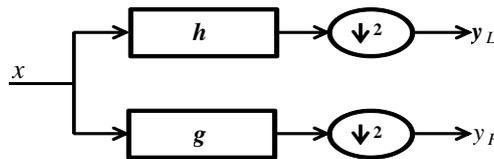


Fig 1. 1 D DWT decomposition

III. LIFTING BASED DWT

The main feature of the lifting-based discrete wavelet transform scheme is to break up the high-pass and low-pass wavelet filters into a sequence of smaller filters that in turn can be converted into a sequence of upper and lower triangular matrices [12]. The basic idea behind the lifting scheme is to use data correlation to remove the redundancy. The lifting algorithm can be computed in three main phases, namely: the split phase, the predict phase and the update phase, as illustrated in Fig 2.

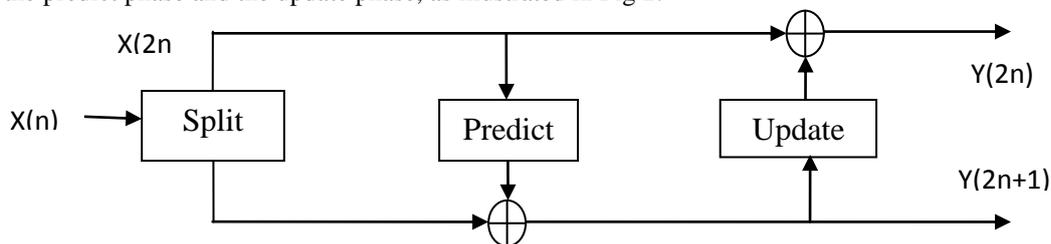


Fig.2. Split, predict and update phases of the lifting based DWT

3.1. Split phase

In this split phase, the data set x(n) is split into two subsets to separate the even samples from the odd ones:

$$X_e = X(2n), X_o = X(2n+1) \quad (2)$$

3.2. Prediction phase.

In the prediction stage, the main step is to eliminate redundancy left and give a more compact data representation. At this point, we will use the even subset x(2n) to predict the odd subset x(2n+1) using a

prediction function P. The difference between the predicted value of the subset and the original value is processed and replaces this latter:

$$Y(2n+1) = Xo(2n+1) - P(Xe) \tag{3}$$

3.3 Update phase.

The third stage of the lifting scheme introduces the update phase. In this stage the coefficient x(2n) is lifted with the help of the neighboring wavelet coefficients. This phase is referred as the primal lifting phase or update phase:

$$Y(2n) = Y(2n+1) + U(Xe) \tag{4}$$

Where, U is the new update operator.

Since the image signals are two-dimensional, the two-dimensional wavelet transform are required. The two-dimensional wavelet transform is computed by recursive application of one-dimensional wavelet transform. After the two-dimensional wavelet transform of the first level, the image is divided into four parts. There are the horizontal low frequency-vertical low frequency component (LL), the horizontal low frequency-vertical high frequency component (LH), the horizontal high frequency-vertical low frequency (HL), and the horizontal high frequency-vertical high frequency (HH), respectively. The standard LS for the 9/7 wavelet filters is shown in Fig.3, where the four lifting coefficients $\alpha, \beta, \gamma, \delta$ and the scaling factor k are evidenced.

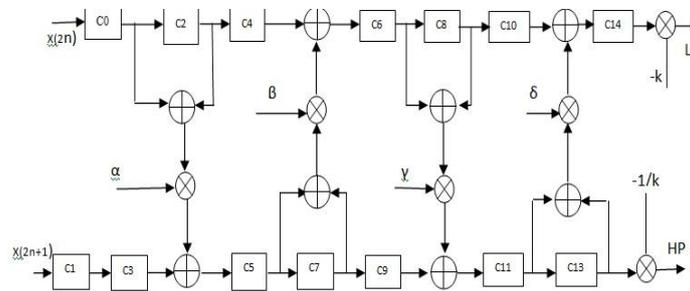


Fig.3. Basic pipeline architecture of 1D-DWT

IV. FAST CONVOLUTION APPROACH

The proposed architecture is based on new and fast convolution approach. It presents an implementation of a very high-speed discrete wavelet transform with reduced hardware complexity and memory [2]. Fig 4 shows the proposed architecture for DWT in [2]. The main principle of this architecture can be applied to implement any symmetric filter. The (9, 7) wavelet filter presents the developed example. These (9, 7) filter has 9 low-pass filter coefficients $h = \{h_{-4}, h_{-3}, h_{-2}, h_{-1}, h_0, h_1, h_2, h_3, h_4\}$ and 7 high-pass filter coefficients $g = \{g_{-2}, g_{-1}, g_0, g_1, g_2, g_3, g_4\}$ and present symmetry,

$$\begin{aligned} h_{-4} &= h_4 & g_{-2} &= g_4 \\ h_{-3} &= h_3 & g_{-1} &= g_3 \\ h_{-2} &= h_2 & g_0 &= g_2 \\ h_{-1} &= h_1 & g_1 & \\ h_0 & & & \end{aligned}$$

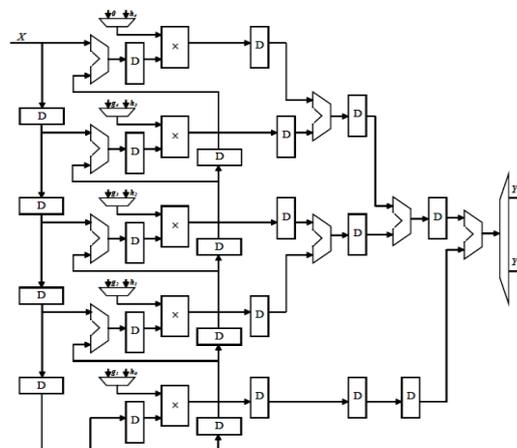


fig 4. Fast convolution architecture for DWT [2]

In the architecture shown in figure 4 a multiplexers are added to send either LPF or HPF coefficient. The proposed architecture to compute the YL(n) and YH(n) is shown in fig 5 and 6 respectively. This proposed method makes the architecture of LPF and HPF suitable for implementing a multiplier-less architecture, as the coefficients to be multiplied in LPF and HPF are different. And also it is observed that as there are only 4 coefficients are present in HPF the architecture of HPF requires only 6 adders, 4 multipliers and 4 D flip flops. To reduce the hardware requirement the conventional multipliers in figure 3 and 4 are replaced by a shift and adders which acts as a constant multipliers as described in section C. Furthermore, the outputs YLi and YHi are obtained alternately at the trailing edges of the even and odd clock cycles. (e.g., YL0, YL1, YL2....., are obtained at clock cycles 9, 11, 13, . . . and YH0, YH1, YH2, ... are obtained at clock cycles 8, 10, 12,..... respectively).

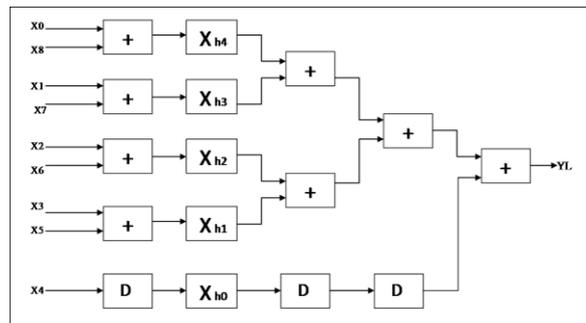


fig 5. Fast convolution architecture for LPF

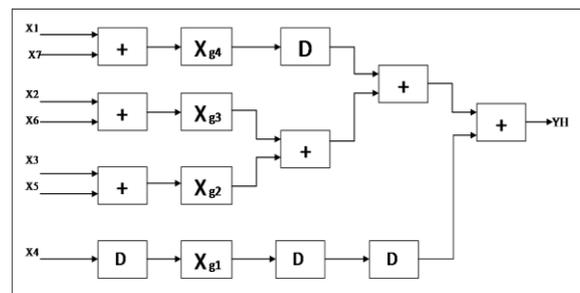


fig 6. Fast convolution architecture for HPF

4.1 Multiplier-less architecture for high-pass and low pass FIR filters

A digital filter is most important and most frequently used element in processing a digital signal and includes delays, multipliers and adders. The simplest form of digital filter is multiplier with number of delays. This type of filter is generally used for processing a signal such as controlling gains. The digital filter is generally comprised of plurality of multipliers, which occupy large areas and consume much power, impose constraints on a one-chip solution when circuits are integrated. In this aspect, efforts have been expanded to reduce the associated hardware complexity by simplifying multipliers used in such digital filters. To avoid multipliers we give preference to BBRF form in FIR filter structure.

The present invention relates to method for processing image in a filter employing BBRF and circuit suitable for the method, which can improve performance of the filter and can be adapted to many kinds of filters by increasing resolution of scaling factors [17].

TABLE I
JPEG2000 standard biorthogonal 9/7 wavelet [19].

k	Lowpass Filter (h_k)	Highpass Filter (g_k)
0	0.6029490182363579	--
1	0.2668641184428723	-1.115087052456994
2	-0.0782232665289878	0.5912717631142470
3	-0.0168641184428749	0.05754352622849957
4	0.02674875741080976	-0.0912717631142494

TABLE II
BBRF for low pass and high pass coefficients

Coefficients	BBRF	Coefficients	BBRF
h_0	1.01'1'001'	--	--
h_1/h_{-1}	0.010001	g_1	1'.001'001
h_2/h_{-2}	0.001'011	g_2/g_0	0.10101'
h_3/h_{-3}	0.00001'1	g_3/g_{-1}	0.000101'
h_4/h_{-4}	0.00011'	g_4/g_{-2}	0.001'01

Table (I) gives the JPEG2000 standard biorthogonal 9/7 wavelet used for image compression application. After approximating, the coefficients can be represented as a Booth binary recoded format as given in table II [18].

V. MULTIPLIER-LESS ARCHITECTURE FOR PIPELINED LIFTING BASED DWT

1D-DWT architecture can be designed as a pipelined structure following the lifting scheme. This basic design is shown in Fig.3. This basic architecture can be designed with 6 multipliers, 8 adders and 14 registers. Here the 6 multipliers are replaced by constant multiplier using shift and add method. The BBRF codes for the coefficients after approximation are as given in table III.

TABLE III
BBRF for lifting coefficients

Coefficients		BBRF
α	1.58613	1'.1'001'1'0
β	0.05298	0.0001'0010
γ	0.88291	0.111000
δ	0.0.443506	0.1001'00010
k	0.81289	0.1110001'0
$1/k$	1.152344	1.0101'1'001'0

VI. RESULTS

The proposed architectures are simulated using Xilinx ISE9.1i & ModelSim XE III 6.3c simulator. Fig 7 and 8 shows the simulation results of 1D_DWT fast convolution block and pipelined lifting based architecture block with $X_i=64$. In each cycle the input bytes ($X_i=64$) are loaded to the 1D_DWT module. The $Y_{L0}, Y_{L1}, Y_{L2}, \dots$, and $Y_{H0}, Y_{H1}, Y_{H2}, \dots$ are obtained at alternate clock cycles. The implementation of this architecture in Xilinx Virtex-II FPGA shows that the proposed architecture requires a gate count of 3222 and provides a speed of 281.373MHz. The multiplier-less pipelined architecture gives a speed of so and so. The table IV shows comparison of hardware complexities of existing multiplier-less architecture for 9/7 filter and also for proposed multiplier-less lifting scheme. The embedded approximation method results in less gate count compared to other available architectures. The table V gives the Comparison between fast convolution DWT & multiplier-less fast convolution based architecture.

TABLE IV
Comparison between other multiplier-less architectures

Structures	Add	MUX	Reg	Gate
Alam et al	43	0	9	9120
Cao at al	27	0	9	6048
Martina et al	19	8	9	4896
Longa et al	35	40	9	9504
BS	9	11	11	3696
BP	21	0	21	9120
Pipeline lifting Scheme	26	0	42	5500
Approximation in convolution	28	0	17	3222
Approximation in lifting scheme	22	0	17	4250

TABLE V

Comparison between fast convolution DWT & multiplier-less fast convolution based DWT (FPGA Device: xc2v1000-6-ff896)

Parameters	Proposed	Fastconvolution architecture[2]
Gate count	3222	8130
Frequency	281.373MHz	270 MHz
Parameters	Proposed	Pipelined architecture
Gate count	4250	5500
Frequency	74.6MHz	69 MHz

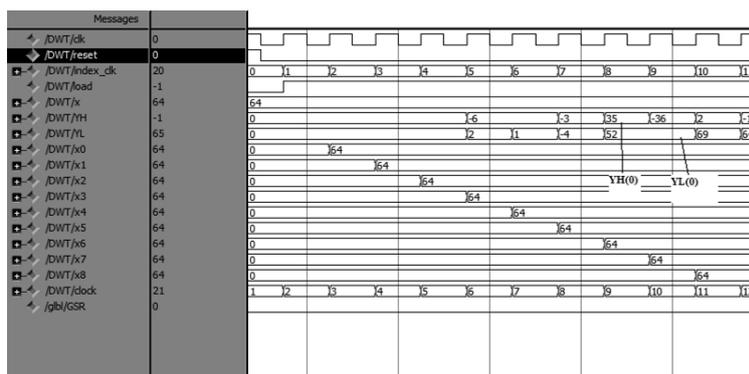


fig 7: simulation results for multiplier-less fast convolution dwt

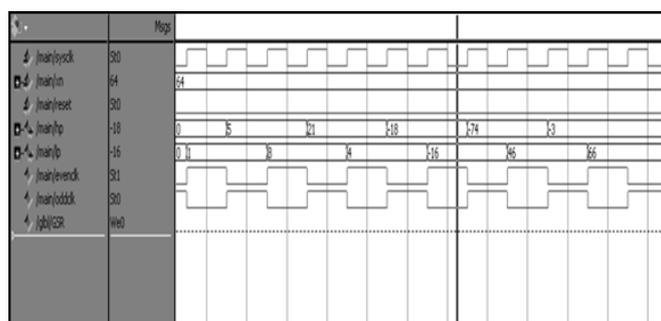


fig 8. Simulation results of pipelined dwt

VII. CONCLUSION AND FUTURE SCOPE

The multiplier-less architecture is used for the fast convolution based 1D DWT. This architecture reduces the gate count from 8130 to 3222 (overall 60% reduction). In FPGA device: xc2v1000-6-ff896 a high speed of about 281.373 MHz is attained for DWT computation so there is an increase of 4% speed compared with the multiplier fast convolution architecture (270MHz). And for the pipelined lifting scheme the speed increased by 5% and the gate count is reduced by 22%. Hence, the approximation method for constant multipliers is efficient in improving the speed and area requirement for real time DWT computation.

As the filter coefficients are quantized, the computation provides approximated output. Hence, the PSNR is low when compared with multiplier based architecture. The quality of the image can be improved by reducing the error produced during multiplication.

REFERENCES

- [1] Maurizio Martina: "Multiplier-less, Folded 9/7– 5/3 Wavelet VLSI Architecture", IEEE transactions on circuits and systems ii: express briefs, vol. 54, no. 9, September 2007
- [2] M Maamoun et al, Low cost VLSI discrete wavelet transform and FIR filters architectures for very high-speed signal and image processing,," Cybernetic Intelligent Systems (CIS), 2010 IEEE 9th International Conference on,
- [3] Ingrid Daubechies et al, "Factoring wavelet transforms into lifting steps", November 2007.
- [4] Basant K. Mohanty: "Efficient Multiplier-less Designs for 1-D DWT using 9/7 Filters Based on Distributed Arithmetic."
- [5] M. Alam et al., "Efficient distributed arithmetic based DWT architecture for multimedia applications," in Proc. IEEE Int. Workshop on SoC for Real-Time Applications, 2003, pp. 333–336.

- [6] X. Cao et al., "An efficient VLSI implementation of distributed architecture for DWT," in Proc. IEEE Workshop Multimedia Signal Process., 2006, pp. 364–367.
- [7] K. A. Kotteri, A. E. Bell, and J. E. Carletta, "Design of multiplier-less, high-performance, wavelet filter banks with image compression applications," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 51, no. 3, pp. 483–494, Mar. 2004.
- [8] M. Martina and G. Masera, "Low-complexity, efficient 9/7 wavelet filters VLSI implementation," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 53, no. 11, pp. 1289–1293, Nov. 2006.
- [9] P. Longa, A. Miri, and M. Bolic, "Modified distributed arithmetic based architecture for discrete wavelet transforms," Electronics Letters vol. 44, no. 4, Feb. 2008.
- [10] Gaurav Tewari, Santu Sardar, K. A. Babu, "High-Speed & Memory Efficient 2-D DWT on Xilinx Spartan3A DSP using scalable Polyphase Structure with DA for JPEG2000 Standard," IEEE, 2011.
- [11] Tze-Yun Sung et al, Reconfigurable VLSI Architecture for 9/7-5/3 Discrete Wavelet Transform.
- [12] Zhiqiang WU et al, "Pipelined Architecture for FPGA Implementation of Lifting-Based DWT", IEEE, 2011.
- [13] Andra, K., Acharya, T., Chakrabarti, C.: "A High- Performance JPEG2000 Architecture. IEEE Transactions on Circuits and Systems for Video Technology," vol. 13.No. 3. (2003) 209-218
- [14] Mallat, S.: "A Theory for Multi resolution Signal Decomposition: The Wavelet Representation. IEEE Trans. on Pattern Analysis and Machine Intelligence," Vol. 11, No. 7. (1989) 674-693
- [15] K. Z. Mei, N. N. Zheng, C. Huang, Y. Liu, and Q. Zeng "VLSI Design of a High-Speed and Area-Efficient JPEG2000 Encoder," IEEE Trans. Circuits Syst. Video Technol., vol. 17, no. 8, pp. 1065–1078, Aug. 2007
- [16] Basant K. Mohanty: Concurrent Systolic Architecture for High-Throughput Implementation of 3-Dimensional Discrete Wavelet Transform. 1-4244-1898-5/08/ ©2008 IEEE
- [17] Mr. M.M. Aswale: "VHDL Implementation of Multiplier-less, High Performance DWT Filter Bank." Proceedings of the World Congresson Engineering 2007 Vol I.
- [18] Tze-Yun Sung: "Reconfigurable VLSI Architecture for 9/7-5/3Discrete Wavelet Transform." WSEAS TRANSACTIONS on CIRCUITS and SYSTEMS. Issue 5, Volume 9, May 2010
- [19] R. F. Gonzalez: "Digital image processing using mat lab. Pearson Education."
- [20] Nazeih M. Botros: "HDL Programming VHDL and Verilog."