A High-Performance VLSI Architecture for Image Compression Technique Using 2-D DWT

***S. Jayachandranath **P. Suresh Babu,**
*Assistant Professor, SV College of Engineering, Tirupati, **Assistant Professor, SV College of Engineering, Tirupati,

Abstract

Image compression is the application of Data compression on digital images. A fundamental shift in the image compression approach came after the Discrete Wavelet Transform (DWT) became popular. To overcome the inefficiencies in the JPEG standard and serve emerging areas of mobile and Internet communications, the new JPEG2000 standard has been developed based on the principles of DWT. In this research, an architecture that performs both forward and inverse lifting-based discrete wavelet transform is proposed. The proposed architecture reduces the hardware requirement by exploiting the redundancy in the arithmetic operation involved in DWT computation. The proposed architecture does not require any extra memory to store intermediate results. Using Verilog HDL, the encoder for the image compression employing DWT was implemented. This architecture has been described in VHDL at the RTL level and simulated successfully using ModelSim simulation environment.

In this paper, high-efficient lifting-based architectures for the 5/3 discrete wavelet transform (DWT) are proposed. The proposed parallel and pipelined architecture consists of a horizontal filter (HF) and a vertical filter (VF). The system delays of the proposed architectures are reduced. Filter coefficients of the biorthogonal 5/3 wavelet low-pass filter are quantized before implementation in the high-speed computation hardware. In the proposed architecture, all multiplications are performed using less shifts and additions. The proposed architecture is 100% hardware utilization and ultra low-power. The architecture has regular structure, simple control flow, high throughput and high scalability. Thus, it is very suitable for new generation image compression systems, such as JPEG- 2000.

Keywords: 5/3 discrete wavelet transform (DWT), IDWT, horizontal filter (HF), vertical filter (VF), lifting-based architecture, JPEG-2000.

1. INTRODUCTION

Data compression is the technique to reduce the redundancies in data representation in order to decrease data storage requirements and hence communication costs. Reducing the storage requirement is equivalent to increasing the capacity of the storage medium and hence communication bandwidth. Thus the development of efficient compression techniques will continue to be a design challenge for future communication systems and advanced multimedia applications. The data compression algorithms can be broadly classified in two categories – lossless and lossy. Usually lossless data compression techniques are applied on text data or scientific data.

The discrete wavelet transform (DWT) is being increasingly used for image coding. It is due to the fact that DWT supports superior features like progressive image transmission by quality or by resolution. The DWT is the key component of the JPEG2000 system, [1] and it also has been adopted as the transform coder in MPEG-4 still texture coding. However, the DWT requires much more computation than the discrete cosine transform (DCT) because of filter computation. Recently, lifting scheme widely used for DWT leads a speed-up and a fewer computation compared to the classical convolution-based method. Daubechies and Sweldens first derive the lifting-based discrete wavelet transform to reduce complex operations [2][3]. The lifting-based DWT has several advantages including entire parallel operations, “in-place” computations of the DWT, integer-to-integer transform, symmetric forward and inverse transform, etc. Hence, the lifting scheme is deservedly adopted in the JPEG 2000 image standard. Several efficient DWT architectures are presented by using the lifting-scheme. In general, 2-D DWT is realized by directly executing the 1-D DWT row by row and then column by column [4][5]. However, the huge frame memory is required to store the intermediate coefficients [4][5]. Due to the *N*2 size of the frame memory, it is usually devised to be the external memory of the DWT chip. Thus, the high external memory bandwidth leads to great power consumption in the 2-D DWT architecture.

The line-based architecture proposed in [6] is an alternative method to eliminate the frame memory by performing 1-D DWT in both directories simultaneously. In order to reduce the external memory access, the line-based method requires some internal line buffer to store the intermediate coefficients. Tseng *et al.* [7] focus their idea on optimizing the internal line buffer size for the line-based 2-D DWT architectures. Moreover, several line-based architectures are proposed based on the factorized lifting scheme [8]-[15]. Andra *et al.*[8] present a lifting-based forward and inverse DWT architecture with a general 1-D DWT core to support various DWT filters in JPEG 2000. A systematic method with systolic array mapping is proposed to construct several efficient architectures for 1-D and 2-D lifting-based DWT [9]. Liao *et al.* [10] propose two DWT architectures with recursive and dual scan methods for multi-level and single-level 2-D DWT decomposition, respectively.

2. BACKGROUND

2.1. DWT

DWT analyzes the data at different frequencies with different time resolutions [1]. Fig. 1 shows the DWT decomposition of the image. The DWT decomposition involves low-pass 'l' and high-pass 'h' filtering of the images in both horizontal and vertical directions. After each filtering, the output is down-sampled by two. Further decomposition is done by applying the above process to the LL sub-band.

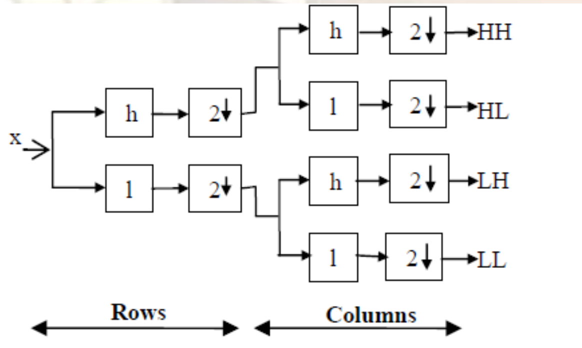


Fig.1: 2-D Wavelet Transform

2.2 Lifting Scheme

The lifting scheme has been developed by Sweldens [4] as an easy tool to construct the second generation wavelets. The scheme consists of three simple stages: split, predict (P) and update (U). In the split stage, the input sequence xj,i is divided into two disjoint set of samples, even indexed samples (even samples) xj,2i and odd indexed samples (odd samples) xj,2i+1. In the predict stage, even samples are used to predict the odd samples based on the correlation present in the signal. The differences between the odd samples and the corresponding predicted values are calculated and referred to as detailed or high-pass coefficients, dj-1,i. The update stage utilizes the key properties of the coarser signals i.e. they have the same average value of the signal. In this stage, the coarse or low-pass coefficient xj-1,i is obtained by updating the even samples with detailed coefficient. The block diagram of the lifting based DWT is shown in Fig. 2

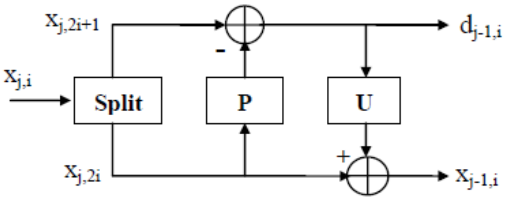


Fig.2: Lifting-based Forward DWT

3. ARITHMETIC IN LIFTING DWT

The lifting scheme provides many advantages, such as fewer arithmetic operations, in-place implementation and easy management of boundary extension compared to convolution based DWT architectures. For simplicity, we use the popular bi-orthogonal wavelet (5,3) filter, adopted in JPEG 2000, in order to explain the redundancy in the arithmetic operation involved in the calculation of the lifting-based DWT computation. The calculation of high-pass and low-pass coefficients for two consecutive values for (5,3) wavelet is shown below:

$$d_{j-1,i} = x_{j,2i+1} + \alpha\,(x_{j,2i}) + \alpha\,(x_{j,2i+2}), \qquad (1)$$

$$d_{j-1,i+1} = x_{j,2i+3} + \alpha\,(x_{j,2i+2}) + \alpha\,(x_{j,2i+4}), \qquad (2)$$

$$x_{j-1,i} = x_{j,2i} + \beta\,(d_{j-1,i-1}) + \beta\,(d_{j-1,i}), \qquad (3)$$

$$x_{j-1,i+1} = x_{j,2i+2} + \beta(d_{j-1,i}) + \beta\,(d_{j-1,i+1}), \qquad (4)$$

where α and β are the (5,3) filter coefficients. From the equations (1) and (2), it is found that the product value of α times xj,2i+2 calculated at the particular clock cycle is required at the next clock cycle. Similarly from equations (3) and (4), the product value of β times dj-1,I at the particular clock cycle is required at the next clock cycle. Therefore, in the proposed architecture for the predict module calculation, we perform one multiplication in each cycle for calculating [α(xj,2i+2)] and the other value [α(xj,2i)] can be obtained from previous clock cycle, instead of performing two multiplications in every clock cycle as mentioned in [7]. Also, the proposed architecture needs only one multiplier in the update module. Thus, the proposed architecture utilizes the redundancy of the above mentioned arithmetic operation reducing the number of multipliers required.

4. THE PROPOSED ARCHITECTURE

The proposed DWT architecture consists of predict module, update module, address generation module, control unit and a set of registers to establish data communication between the modules. This architecture can be used to carry out both forward and inverse discrete wavelet transform.

4.1 Predict Module

The predict module for (5,3) wavelet is shown in Fig. 3. Initially, the input register R1 is loaded with the even sample from the input RAM. In the meantime, the predict filter coefficient ‘α’ and the corresponding odd sample are made available to calculate the detailed coefficient dj-1,i. The second register R2 stores the output of the multiplier in the current cycle and in the meantime the register R2 supplies the multiplier output obtained in the previous cycle. Thus, we reduce the number of multipliers required for predict operation for (5,3) wavelet to one whereas the number required for the architecture described in [7] is two. For (5,3) wavelet, we can use shifters instead of multipliers

.

4.2 Update Module

The structure of the update module for (5,3) wavelet is shown in Fig. 4. The input register R1 is loaded with the even sample. In the next clock cycle, the multiplier is fed with the detailed coefficient and the update

coefficient ‘β’ and the output of the multiplier is fed to both the adders as shown in Fig. 4 . In this case also, we can use shifters instead of multipliers for (5,3) wavelet.

4.3 Address Generation Module (AGM)

The AGM generates appropriate read and write addresses for both even and odd samples to the input RAM as shown in Fig. 5. As mentioned in JPEG2000 [2], the signal is symmetrically extended by two signal values to the left side and by one signal value on the right side for (5,3) wavelet to reduce artifacts at the boundary. The boundary treatment problem is solved by passing proper start address (start_odd_addr and start_even_addr) of the input signal and increment values (incr_even_addr and incr_odd_addr) to the AGM. Let us assume a signal of length 64 and perform the first level of DWT computation. If the signal is symmetrically extended as mentioned in JPEG2000, the signals start_even_addr and start_odd_addr are set to two and one respectively. The update_address signal is set to one for the first clock cycle.

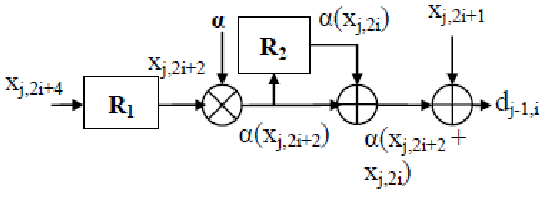


Fig.3: Predict Module of (5,3) wavelet

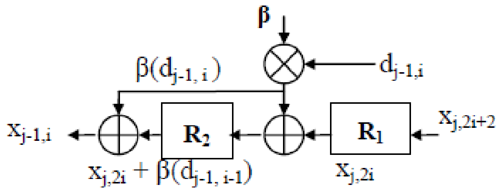


Fig.4: Update Module of (5,3) wavelet

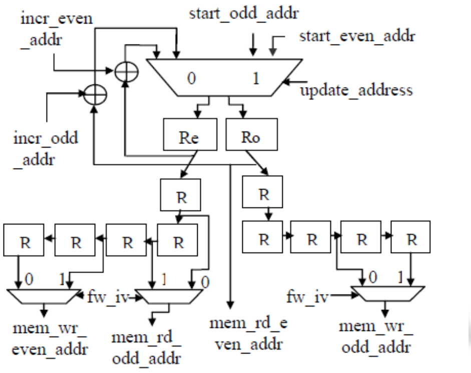


Fig.5: AGM for (5,3) Wavelet (R – Registers)

In this clock cycle, the registers Re and Ro are set with the start_even_addr and start_odd_addr respectively. These registers store the output of the adders from the next clock cycle onwards. The incr_even_addr and incr_odd_addr signals are set to ‘-2’ and ‘0’ when DWT computation is carried out on the symmetrically extended signals on the left side of the signal. The incr_even_addr and incr_odd_addr signals are set to two when the decomposition is carried out on the signal and both set to zero when the DWT computation is carried out on the symmetrically extended signals at the right side of the signal. The selection of even and odd samples from the symmetrically extended signal to complete first level of DWT computation for this example is shown in Fig. 6. The LL subband is decomposed in each level of decomposition in DWT.

The modules are integrated for (5,3) wavelet as shown in Fig. 6 with the set of registers. In this architecture, we use dual-port input RAM and it operates twice as fast as the system clock frequency to obtain the detailed coefficient and the coarse coefficient at every clock cycle. The forward or inverse DWT is selected based on the value of the fw_iv signal (1 or 0). The mem_rd_odd_addr and mem_wr_odd_addr provide addresses to read odd samples and to write coarse coefficients respectively. Similarly, the mem_rd_even_addr and mem_wr_even_addr provide addresses to read even samples and to write detailed coefficients respectively.

5. PERFORMANCE MEASURES

We have developed VHDL model of the proposed architecture at the RTL level and successfully simulated using ModelSim simulation environment. The performance analysis is performed in terms of hardware (number of multipliers required) requirement and computation time for (5,3), (9,7) and (13,7) wavelets. Because the set of registers controlled by the clock is employed, the architecture does not require any extra memory/FIFO to store the intermediate results. Table 1 provides the comparative evaluation of the proposed architecture with other architectures [6], [7] in terms of area and computation time for one level of decomposition of the signal of size NxN.

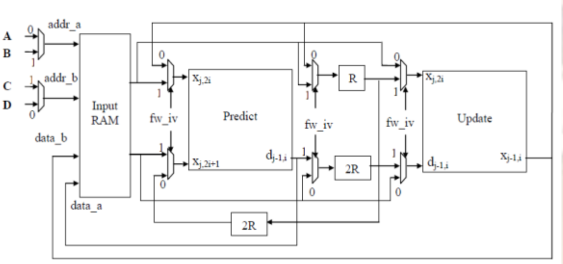


Fig.6: Architecture of (5,3) wavelet transform (A – mem_rd_even_addr, B – mem_wr_even_addr, C – mem_wr_odd-addr, D – mem_rd_even_addr, nR – 'n' number of registers connected in series)

The proposed architecture needs less number of multipliers compared to other architectures proposed in[6], [7]. The architecture proposed in [6] has better computation time than the proposed architecture in the case of (5,3) and (9,7) wavelets but it requires same computation time approximately as the proposed architecture for (13,7) wavelet. Furthermore the architecture proposed in [6] requires greater hardware area. The main advantage of the proposed architecture is that it utilizes less number of multipliers compared to other architectures.

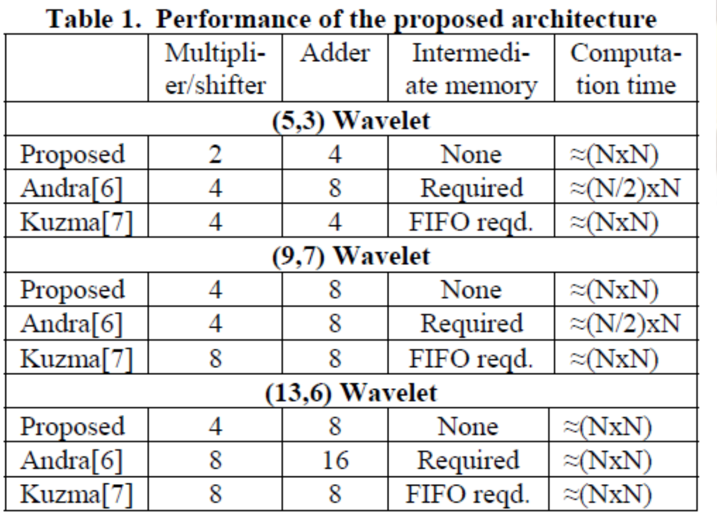
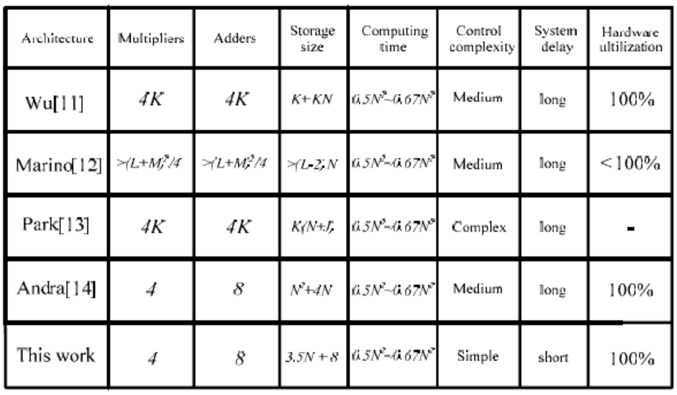


Table 1. Performance of the proposed architecture

	Multiplier/shifter	Adder	Intermediate memory	Computation time
(5,3) Wavelet				
Proposed	2	4	None	≈(NxN)
Andra[6]	4	8	Required	≈(N/2)xN
Kuzma[7]	4	4	FIFO reqd.	≈(NxN)
(9,7) Wavelet				
Proposed	4	8	None	≈(NxN)
Andra[6]	4	8	Required	≈(N/2)xN
Kuzma[7]	8	8	FIFO reqd.	≈(NxN)
(13,6) Wavelet				
Proposed	4	8	None	≈(NxN)
Andra[6]	8	16	Required	≈(NxN)
Kuzma[7]	8	8	FIFO reqd.	≈(NxN)

Table2. The comparison between previous works and this work in 2- DWT



Architecture	Multipliers	Adders	Storage size	Computing time	Control complexity	System delay	Hardware ultilization
Wu[11]	$4K$	$4K$	$K+KN$	$0.5N^2-0.67N^2$	Medium	long	100%
Marino[12]	$>(L+M)^2/4$	$>(L+M)^2/4$	$>(L-2)N$	$0.5N^2-0.67N^2$	Medium	long	<100%
Park[13]	$4K$	$4K$	$K(N+1)$	$0.5N^2-0.67N^2$	Complex	long	-
Andra[14]	4	8	N^2+4N	$0.5N^2-0.67N^2$	Medium	long	100%
This work	4	8	$3.5N+8$	$0.5N^2-0.67N^2$	Simple	short	100%

6. Conclusions and Discussions

Filter coefficients are quantized before implementation using the biorthogonal 5/3 wavelet. The hardware is cost-effective and the system is high-speed. The architecture reduces power dissipation by *m* compared with conventional architectures in *m*-bit operand (lowpower utilization). In this paper, the high-efficient and low-power architectures for 2-D DWT and IDWT have been proposed. The architectures for DWT and IDWT perform compression and decompression in $(4N\ 2\ (1-4\ j\ ) + 9N)\ /\ 6$ computation time, where the time unit is an addition operation. The control complexity is simple. The comparison between previous works and this work in 2-D DWT is shown in Table 1 [11], [12], [13], [14]. So far, the papers of 5/3 2-D lifting-based IDWT are rarely published. The proposed architecture for 2-D IDWT is very suitable for VLSI implementation. The design analysis of the proposed architecture is valuable for the future research. The proposed architectures have been verified by Verilog-HDL and implemented on FPGA. The hardware code can be reused and become as IP. The advantages of the proposed architectures are 100% hardware utilization and ultra low-power. The architectures have regular structures, simple control flows, high throughputs and high scalabilities.

Thus, it is very suitable for new-generation image compression systems, such as JPEG-2000.

7. REFERENCES

[1] O. Rioul and M. Vetterli, "Wavelets and Signal Processing," *IEEE Signal Processing*, vol. 8, issue: 4, pp. 14-38, Oct. 1991.

[2] ISO/IEC. International Standard, 15444-1: 2000(E), JPEG2000 Image Coding System – Part I Core coding system.

[3] C. Chakrabarti, M. Vishwanath, and R. M Owens, "Architectures for wavelet

transforms: A survey," *J. VLSI Signal Process.*, vol. 14, pp.171– 192, 1996.

[4] W. Sweldens, "The lifting scheme: A construction of second-generation wavelets," *SIAM J. Mathematical Analysis*, vol. 29, no.2, pp. 511–46, 1997.

[5] P. Chen, "VLSI implementation for one-Dimensional multilevel lifting-based wavelet transform," IEEE Trans. on Computers, vol.53, no.4, pp.386-398, April 2004.

[6] K. Andra, C. Chakrabati and T. Acharya, "A VLSI Architecture for Lifting-based Forward and Inverse Wavelet Transform," *IEEE Trans. On Signal Processing*, vol.50, no.4, pp966-977, April 2002.

[7] D. Taubman , "High performance scalable image compression with EBCOT," *IEEE Trans. Image Processing*, vol. 9, pp. 1158-1170, July 2000.

[8]R. Kronland-Martinet, J. Morlet, and A. Grossman "Analysis of sound patterns through wavelet transform," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 1, no. 2, 1987, pp. 273-302.

[9] M. A. Stoksik, R. G. Lane, D. T. Nguyen, "Accurate synthesis of fractional Brownian motion using wavelets," *Electronic Letters*, vol.30, no. 5, Mar. 1994, pp. 384-284.

[10] K. Parhi and T. Nishitani, "VLSI architectures for discrete wavelet transforms," *IEEE Trans.VLSI Systems*, vol. 1, no. 2, 1993, pp. 191-202.

[11] P.Wu"An efficient architecture for two dimensional discrete wavelet transform," *IEEE Trans. on Circuits and System for video Tech.*, 11, 2001, pp.536-545.

[12] F. Marino, "Two fast architectures for the direct 2-D discrete wavelet transform-Signa l Processing," *IEEE Trans. on Signal Processing*, 49, 2001, pp.1248-1347.

[13] T . Park and S.Jung, "High speed lattice based VLSI architecture of 2d discrete wavelet transform for real time video signal processing," Consumer Electronics, *IEEE Trans. On Consumer Electronics* ,48, 2002, pp..1026-1032.

[14] K. Andra, C. Chakrabarti and T. Acharya, " A VLSI architecture for lifting-based forward and inverse wavelet transform," *IEEE Trans. on Signal Processing*, 50, 2002, pp.966-977.

[15] T. Y. Sung, Y. S. Shieh, "A High-Speed / Ultra Low-Power Architecture for 2-D Discrete Wavelet Transform," *2005 IEEE International Conference on Systems and Signals (ICSS2005)*, I-Shou University, Kaohsiung, Taiwan, April 28-29,2005,pp.326-331.