Wavelet Transform for Forward and Inverse

Karimella Vikram
De[aertment of f CSE
Medak College Engg& Tech
Siddpet , Medak (D)
Email: vikrakarimell@yaho.com

Abstract*—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. The proposed architecture consists of predict module, update module, address generation module, control unit and a set of registers to establish data communication between predict and update modules. The symmetrical extension of images at the boundary to reduce distorted images has been incorporated in our proposed architecture as mentioned in JPEG2000. This architecture has been described in VHDL at the RTL level and simulated successfully using Model Sim simulation environment. *Keywords: Lifting, DWT, DWT architecture, JPEG 2000.

I. INTRODUCTION

The discrete wavelet transform (DWT) is widely used in many fields such as image compression and signal analysis [1]. For example, JPEG2000, one of the popular image formats, includes DWT for compression [2]. Since the DWT is a very computation-intensive process, the study of its hardware implementation has gained much importance. Several DWT architectures have been proposed [3] using the filter convolution. A new scheme, termed lifting-based scheme [4] that often requires less computation, has been proposed for constructing biorthogonal wavelets. Several architectures have been proposed for the efficient computation of 1-D and 2-D DWT [5-7] base on the lifting scheme. The multilevel architecture Proposed in [5] requires changes in the architecture Design for different number of level of DWT computation even though it computes the coefficient faster. In this paper, we propose lifting-bases architecture that performs one level of DWT at a time and the architectures mentioned in [6, 7] computes DWT in the same fashion. The proposed architecture performs both forward and inverse DWT when the signal is symmetrically extended, This paper is organized as follows: in Section 2, the DWT and the lifting scheme are introduced briefly. The redundancy in arithmetic operation involved in DWT is briefly discussed in Section 3. The proposed architecture for lifting-based DWT is described in detail in Section 4. Section 5 describes the performance comparison. A conclusion is given in Section 6.

II. BACKGROUND

A. 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.*B. Lifting Scheme*

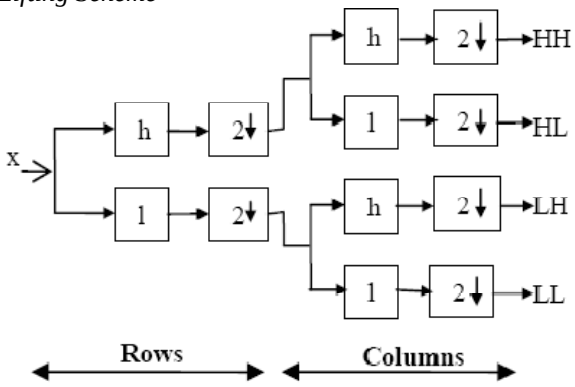

Fig. 1: 2-D Wavelet Trasform

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, 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.

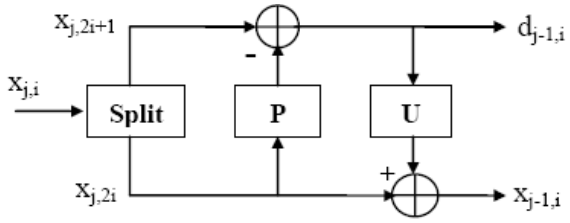

Fig. 2: Lifting-based Forward DWT

III. 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:

$dj\text{-}1,i = xj,2i+1 + \alpha\,(xj,2i) + \alpha\,(xj,2i+2)$ (1)

$dj\text{-}1,i+1 = xj,2i+3 + \alpha\,(xj,2i+2) + \alpha\,(xj,2i+4)$ (2)

$xj\text{-}1,i = xj,2i + \beta\,(dj\text{-}1,i\text{-}1) + \beta\,(dj\text{-}1,i)$ (3)

$xj\text{-}1,i+1 = xj,2i+2 + \beta(dj\text{-}1,i) + \beta\,(dj\text{-}1,i+1)$ (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 or 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. Similarly, the proposed architecture needs two multipliers each for predict and update modules in the case of (13,7) wavelet. Thus, the proposed architecture utilizes the redundancy of the above mentioned arithmetic operation reducing the number of multipliers required.

IV. THE PROPOSED ARCHITECTURE

The proposed DWT architecture consists of predict module, update module, and 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.

A. 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. Similarly, for (13,7) wavelet, only two multipliers required for predict module instead of four. For (5,3) wavelet, we can use shifters instead of multipliers. *B. 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. Similarly, for (13,7) wavelet, we need only two multipliers for update module. In this case also, we can use shifters instead of multipliers for (5,3) wavelet. *C. 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. 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 An Efficient Architecture for Lifting-based Forward and Inverse Discrete Wavelet Transform 417 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. 7 with the set of registers. In this architecture, we use dual-port inputRAM 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.

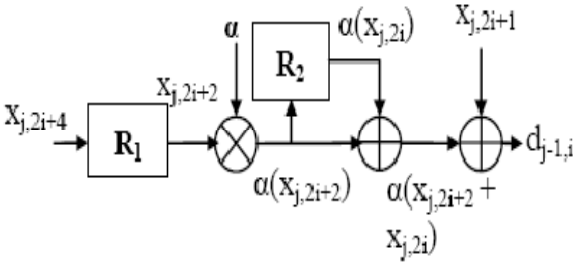

Fig. 3: Predict Module of (5,3) Wavlest

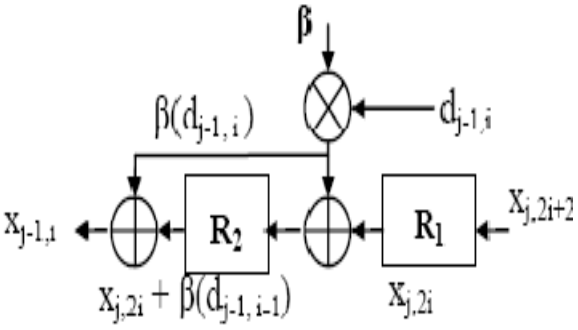

Fig. 4: Update Module of (5,3) Wavlest

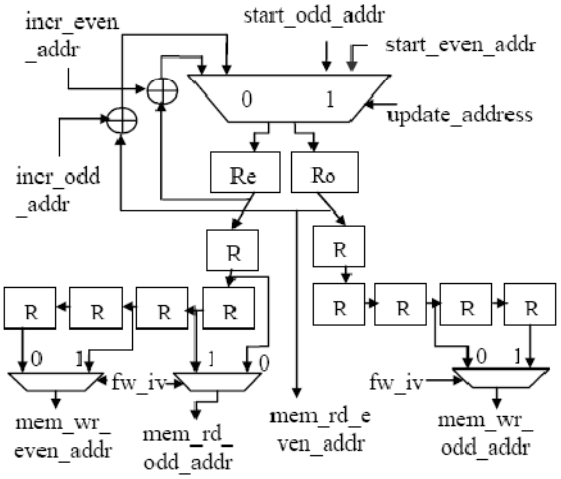

Fig. 5: AGM for (5,3) Wavelet (R-Registers)

Similarly, the mem_rd_even_addr and mem_wr_even_addr provide addresses to read even samples and to write detailed coefficients respectively.

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

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

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

V. 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

	Region1	Region2	Region3
Symm. extended signal	2, 1,	0, 1, 2, 3, 4,……, 62, 63, 64,	63
Even samples	2,	0, 2, 4, 6,……, 60, 62, 64	
Odd samples	1,	1, 3, 5, 7,……, 59, 61, 63,	63

Region1 – Sym. extended signal on the left side.
Region2 – Signal length.
Region3 – Sym. Extended signal on the right side.

Fig. 6: Addresses Required to Select Even and Odd Samples for

(5,3) Wavelet

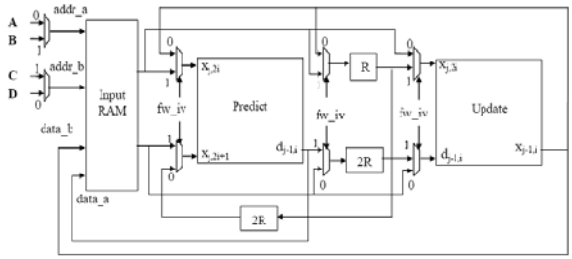

Fig. 7: Architecture of (5,3) wavlet transform (A-men_rd_even_addr, B-men_wr_even_addr, C-man_wr_odd-addr, D-men_rd_even_addr, nR-"n" number of registers connected in series)

VI. PERFORMANCE MEASURES

We have developed VHDL model of the proposed architecture at the RTL level and successfully simulated using Model Sim 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. 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 rchitecture for (13,7) wavelet. Furthermore the architecture proposed in [6] requires greater hardware area. The main advantage of the proposed architectu re is that it utilizes less number of multipliers compared to other architectures. TABLE 1: PERFORMANCE OF THE PROPOSED ARCHITECTURE

	Multipli-er/shifter	Adder	Intermedi-ate memory	Computa-tion 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)

VII. CONCLUSION

In this paper, an efficient DWT architecture utilizing less hardware area has been proposed. Based on the proposed architecture, we used registers to reduce number of multipliers required and to avoid using any external memory to store the intermediate results. We compared our architecture with other architectures and shown that the proposed architecture utilizes less hardware area. Our future work will be concerned with the development of an "FPGA-based image compression," based on the proposed DWT architecture.

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 secondgeneration
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.