A Detailed Survey on VLSI Architectures for Lifting based DWT for efficient hardware implementation

Usha Bhanu.N[1] and Dr.A.Chilambuchelvan[2]

[1]Research Scholar, Anna University, Chennai-25, INDIA
bhanu.usha@rediffmail.com
[2]Professor, R.M.D. Engineering college ,Chennai-601 206 , INDIA
hodeie@rmd.ac.in

Abstract

Evaluating the previous work is an important part of developing new hardware efficient methods for the implementation of DWT through Lifting schemes. The aim of this paper is to give a review of VLSI architectures for efficient hardware implementation of wavelet lifting schemes. The inherent in place computation of lifting scheme has many advantages over conventional convolution based DWT. The architectures are represented in terms of parallel filter, row column, folded, flipping and recursive structures. The methods for scanning of images are the line-based and the block-based and their characteristics for the given application are given. The various architectures are analyzed in terms of hardware and timing complexity involved with the given size of input image and required levels of decomposition. This study is useful for deriving an efficient method for improving the speed and hardware complexities of existing architectures and to design a new hardware implementation of multilevel DWT using lifting schemes.

Keywords

Discrete Wavelet Transform, Lifting schemes, VLSI architectures, image compression.

1. INTRODUCTION

The Discrete Wavelet Transform (DWT) plays a major role in the fields of signal analysis, computer vision, object recognition, image compression and video compression standard. The advantage of DWT over other traditional transformations is that it performs multi resolution analysis of signals with localization both in time and frequency as described by Mallat [3]. At present, many VLSI architectures for the 2-D DWT have been proposed to meet the requirements of real-time processing. The implementation of DWT in practical system has issues. First, the complexity of wavelet transform is several times higher than that of DCT. Second, DWT needs extra memory for storing the intermediate computational results. Moreover, for real time image compression, DWT has to process massive amounts of data at high speeds. The use of software implementation of DWT image compression provides flexibility for manipulation but it may not meet timing constraints in certain applications. Hardware implementation of DWT has practical obstacles. First, is that the high cost of hardware implementation of multipliers. Filter bank implementation of DWT contains two FIR filters. It has traditionally been implemented by convolution or the finite impulse response (FIR) filter bank structures. Such implementations require both large number of arithmetic computations and storage, which are not desirable for either high speed or low power image/video processing applications.

DOI : 10.5121/vlsic.2012.3213

Therefore a new approach called the lifting scheme based wavelet transform was proposed by Sweldens based on a spatial construction of the second generation wavelet and a very versatile scheme for its factorization has been suggested in Sweldens[6].The lifting scheme has many advantages over the previous approaches. In particular, all the interesting properties of wavelets, such as bi-orthogonality and regularity, are defined by linear relationships between the filter bank coefficients. As a consequence, it is easier to design wavelet filters. Unlike convolutional wavelets, lifting scheme does not depend on Fourier transform of the wavelets. As a consequence, wavelets can be designed on arbitrary lattices in spatial domain. Since the lifting scheme makes optimal use of similarities between the high and low pass filters to speed up the calculation of wavelet transform, it has been adopted in the image compression standard JPEG2000. The various architectures differ in terms of required numbers of the multipliers, adders and registers, as well as the amount of accessing external memory, and leads to decrease efficiently the hardware cost and power consumption of design.Inspite of improving the efficiency of existing architectures, the present requirement is to improve the hardware utilization and capable of handling multiple data streams for the calculation of 2D DWT.

This paper focuses to give brief survey on 1D and 2D DWT hardware architectures with their implementation VLSI structures and computational complexities. The paper is structured as follows: Section 2 reviews the filter bank implementation or conventional convolution method of DWT and the basics of lifting algorithm along with their mathematical background. In Section 3, various one-dimensional lifting-based DWT architectures suitable for VLSI implementation and comparison of the hardware and timing complexities of all the architectures are discussed. Section 4. gives the memory requirement for 2-dimensional DWT architectures, followed by representative architectures and a comparison of their hardware and timing complexities with the possibility of extending to multilevel input signals. Finally, in section 5 concluding remarks are made.

2. DWT AND LIFTING SCHEME IMPLEMENTATION

2.1 DISCRETE WAVELET TRANSFORM USING CONVOLUTION

The inherent time-scale locality characteristics of the discrete wavelet transforms (DWT) have established as powerful tool for numerous applications such as signal analysis, signal compression and numerical analysis. This has led numerous research groups to develop algorithms and hardware architectures to implement the DWT. Discrete wavelet transform (DWT) is being increasingly used for image coding. This is due to the fact that DWT supports features like progressive image transmission, ease of compressed image manipulation, region of interest coding etc. The VLSI architectures proposed in [1-5] for hardware implementations of DWT are mainly convolution-based. In the conventional convolution method of DWT, a pair of Finite Impulse Response filters (FIR) is applied in parallel to derive high pass and low-pass filter coefficients.Mallat's pyramid algorithm [3] can be used to represent the wavelet coefficients of image in several spatial orientations. The architectures are mostly folded and can be broadly classified into serial and parallel architectures. The architecture in [5] implements filter bank structure efficiently using digit serial pipelining. The architecture proposed in [1] employs polyphase decomposition and coefficient folding technique for efficient implementation of discrete wavelet transform. A general fashion in which DWT decomposes the input image is shown below in Fig.1.

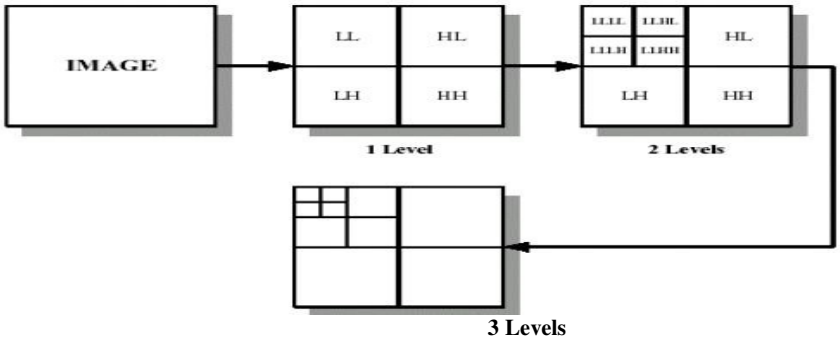


Figure. 1 Three level decomposition of an image

Each decomposition level shown in Fig.1 comprises two stages: stage 1 performs horizontal filtering and stage 2 performs vertical filtering. In the first-level decomposition, the size of the input image is N* N, and the outputs are the three sub bands LH, HL, and HH, of size N/2*N/2. In the second-level decomposition, the input is the LL band and the outputs are the three sub bands LLLH, LLHL, and LLHH, of size N/4*N/4.The multi-level 2-D DWT can be extended in an analogous manner. The arithmetic computation of DWT can be expressed as basic filter convolution and down sampling. Two dimensional discrete wavelet transform (DWT) is defined as:

$$x_{LL}^{J}(n_1,n_2)=\sum_{i_1=0}^{K-1}\sum_{i_2=0}^{K-1}g(i_1).g(i_2).x_{LL}^{J-1}(2n_1-i_1)(2n_2-i_2) \quad (1)$$

$$x_{LH}^{J}(n_1,n_2)=\sum_{i_1=0}^{K-1}\sum_{i_2=0}^{K-1}g(i_1).h(i_2).x_{LL}^{J-1}(2n_1-i_1)(2n_2-i_2) \quad (2)$$

$$x_{HL}^{J}(n_1,n_2)=\sum_{i_1=0}^{K-1}\sum_{i_2=0}^{K-1}h(i_1).g(i_2).x_{LL}^{J-1}(2n_1-i_1)(2n_2-i_2) \quad (3)$$

$$x_{HH}^{J}(n_1,n_2)=\sum_{i_1=0}^{K-1}\sum_{i_2=0}^{K-1}h(i_1).h(i_2).x_{LL}^{J-1}(2n_1-i_1)(2n_2-i_2) \quad (4)$$

Where x_{LL} (n1, n2) is the input image, J is 2-D DWT level is filter length, g (n) is impulse responses of the low-pass filter and h (n) impulse responses of the high-pass filter.

The input sequence X (n) in Fig. 2 is convolved with the quadrature mirror filters H (z) and G (z) and the outputs obtained 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 by two.

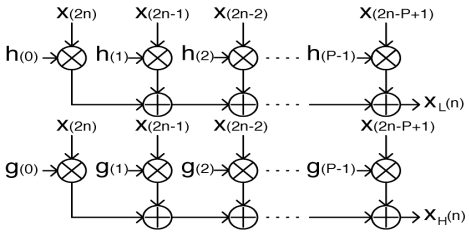


Figure.2 General convolution-based architecture

The 1D-DWT is a two channel sub-band decomposition of an input signal X (n) that produces two sub-band coefficients Y_L (n) and Y_H(n).A separable two dimensional DWT can be obtained

by taking 1D DWT along rows and 1D DWT along the columns as in equations[1-4].There have been several efficient architectures proposed for convolution-based DWT, such as [1-5,27]. However the simplest parallel architecture as shown in Fig.2.can be used if the throughput is set as two-input/two-output per clock cycle with minimum latency and few registers. The critical path is, Tm+(K-1)Ta where T_m is the time taken for a multiplication operation, Ta is the time needed for an addition operation, and K is the filter length. The hardware cost for the filter is, $2KC_m+2(K-1)\ C_a$ where C_m and C_a are the hardware cost of a multiplier and an adder, respectively. The Coefficient folding technique outperforms parallel filter structure with respect to hardware utilization and speed.[27]

The low pass filter has 5 taps and the high pass has 3 taps and hence it is (5, 3) wavelet. The Filter bank analysis of wavelet transforms is in the frequency domain and not in the time domain and the filter coefficients are not integer numbers so they are not appropriate for hardware implementation. In addition, the number of arithmetic computations in the FB method is very large.

2.2. LIFTING BASED DWT

The lifting scheme is a new method to construct wavelet basis, which was first introduced by Swelden's [6]. The lifting scheme entirely relies on the spatial domain, has many advantages compared to filter bank structure, such as lower area, power consumption and computational complexity. The lifting scheme can be easily implemented by hardware due to its significantly reduced computations. Lifting has other advantages, such as "in-place" computation of the DWT, integer-to-integer wavelet transforms which are useful for lossless coding. The lifting scheme has been developed as a flexible tool suitable for constructing the second generation wavelets. It is composed of three basic operation stages: split, predict and update. Fig.3. shows the lifting scheme of the wavelet filter computing one dimension signal. The three basic steps in Lifting based DWT are:

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. For each pair of given input samples x(n) split into even x(2n) and odd coefficients x(2n+1).

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 (dj).Detailed coefficients results in high pass filtering.

$$HP[2n+1] = X[2n-1] - \left[\frac{X[2n]+X[2n+2]}{2}\right] \quad (5)$$

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 (sj).The coarser coefficients gives low pass filtered output.

$$LP[2n] = X[2n] + \left[\frac{HP[2n-1]+HP[2n+1]+2}{4}\right] \quad (6)$$

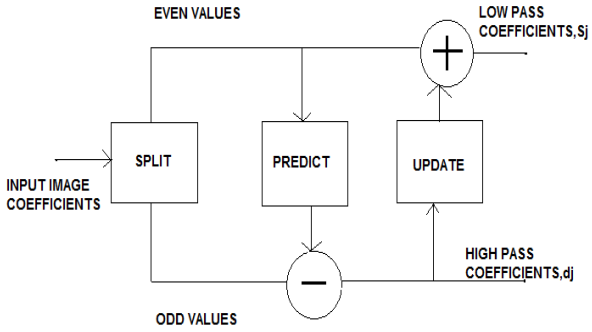


Figure. 3 Block diagram of forward Lifting scheme

The inverse transform could easily be found by exchanging the sign of the predict step and the update step and apply all operations in reverse order as shown in Fig.4.The implementation of lifting based inverse transform (IDWT) is simple and it involves order of operations in DWT to be reversed. Hence the same resources can be reused to define a general programmable architecture for forward and inverse DWT.

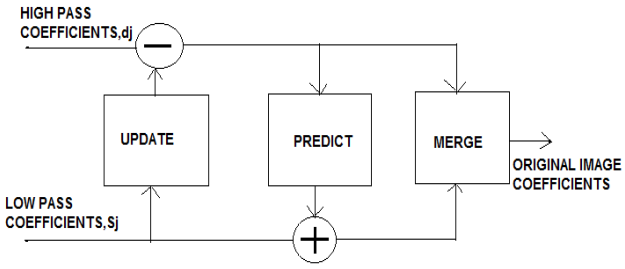


Figure. 4 Block diagram of inverse lifting scheme

2.3 Mathematics in DWT and lifting implementation

DWT of perfect reconstruction filters can be decomposed into a finite sequence of lifting steps [6]. The decomposition corresponds to a factorization of the poly-phase matrix of the target wavelet filter into a sequence of alternating upper and lower triangular matrices and a constant diagonal matrix,

$$h(z)=h_e(z^2)+Z^{-1}h_o(z^2) \quad (5)$$

$$g(z)=g_e(z^2)+z^{-1}g_o(z^2)$$

$$p(z)=\begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix} \quad (6)$$

P(z) is called the dual (synthesis) of $\tilde{p}(z)$ and for perfect reconstruction $p(z)\,\tilde{p}(z^{-1})^T = I$ where I is the 2x2 identity matrix.Wavelet transforms in terms of poly-phase matrix is,

$$\begin{bmatrix} y_L(z) \\ y_H(z) \end{bmatrix} = \tilde{p}(z) \begin{bmatrix} X_e(z) \\ z^{-1}X_o(z) \end{bmatrix}$$

$$\begin{bmatrix} X_e(z) \\ z^{-1}X_o(z) \end{bmatrix} = p(z) \begin{bmatrix} y_L(z) \\ y_H(z) \end{bmatrix} \quad (7)$$

When the determinant of p(z) is unity synthesis filter pair $(\tilde{h}, \tilde{g})$ and analysis filter pair (h,g) are both complementary. When (h,g)=($\tilde{h}, \tilde{g}$), the wavelet transformation is orthogonal, otherwise it is biorthogonal.The filter pair can be factorized into finite sequence of alternating upper and lower triangular matrices, the relation between filter bank coefficients and lifting schemes is,

$$E(z) = \begin{bmatrix} h_e(z) & h_o(z) \\ g_e(z) & g_o(z) \end{bmatrix} = \begin{bmatrix} k & 0 \\ 0 & 1/k \end{bmatrix} \prod_{i=1}^{m} \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \quad (8)$$

Where h_e and h_o are even and odd taps of LPF, ge and g_o are even and odd taps of HPF $S_i(z)$ and $t_i(z)$ are the filter coefficients in filter bank structure and K is scale normalization factor

2.3.1 Example [12]

For JPEG2000 (5,3) lossless standard. It consists of one lifting step. For this wavelet the prediction of each odd sample and signal is the average of two adjacent even samples. Then p block calculates the difference between the real value of signal sample and its prediction. Matrix equation for (5, 3) wavelet with a=-1/2 and b=1/4 is given by,

$$\begin{bmatrix} HP(z) \\ LP(z) \end{bmatrix} = \begin{bmatrix} \sqrt{2} & 0 \\ 0 & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 0 & b(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ a(1+z) & 1 \end{bmatrix} \begin{bmatrix} X_e(z) \\ X_o(z) \end{bmatrix} \quad (9)$$

$$\begin{bmatrix} HP(z) \\ LP(z) \end{bmatrix} = \begin{bmatrix} 1 & a(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ b(1+z) & 1 \end{bmatrix} \begin{bmatrix} 1 & c(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ d(1+z) & 1 \end{bmatrix} \begin{bmatrix} k_0 & 0 \\ 0 & k_1 \end{bmatrix} \quad (10)$$

Equation (10) represents matrix computation of (9,7) lossy lifting scheme adopted in JPEG 2000 with a, b, c and d are lifting coefficients and k_0,k_1 are scale normalization coefficients where a=-1.586134342, b=-0.05298011854, c=0.8829110762,d=0.4435068522, k_1=1.149604398.

2.3.2. Multilevel implementation of (9, 7) lifting schemes

The odd (even) indexed data samples are represented by $x_{2n+1}(2n)$. The intermediate values computed during the lifting steps are denoted as $m^k_{2n+1}$and m^k_{2n}, (k=0,1,2,3…..) levels and the high and low frequency coefficients are given by m_{2n+1} and m_{2n} with these mathematical notation, the implementation of CDF(9,7) can be rewritten as follows[20].

$$\begin{aligned}
m^0_{2n+1} &= x_{2n+1} \\
m^0_{2n} &= x_{2n} \\
m^1_{2n+1} &= m^0_{2n+1} + a(m^0_{2n} + m^0_{2n+2}) \\
m^1_{2n} &= m^0_{2n} \\
m^2_{2n+1} &= m^1_{2n+1} \\
m^2_{2n} &= m^1_{2n} + b(m^1_{2n-1} + m^1_{2n+1}) \\
m^3_{2n+1} &= m^2_{2n+1} + c(m^2_{2n} + m^2_{2n-2}) \\
m^3_{2n} &= m^2_{2n} \\
m_{2n+1} &= m^3_{2n+1} \\
m_{2n} &= m^3_{2n} + d(m^3_{2n-1} + m^3_{2n+1})
\end{aligned} \quad (11)$$

Multilevel decomposition can be done with the intermediate values. The steps shown above can be used for efficient realization of multilevel wavelet architecture through lifting schemes. The first six steps are used for implementing (5, 3) lossless standard of lifting scheme. It is very clear

that the Lifting scheme based realization allows Integer Wavelet Transform (IWT).The transform coefficients of the IWT are exactly represented by finite precision numbers, thus allowing for truly lossless encoding. This helped in reducing number of bits for the sample storage and to use simpler filtering units and no multipliers are required. The complexity of conventional convolution and lifting schemes are compared in terms of basic components needed for its implementation. The Table.1 shows the number of multiplications, additions and shifts needed for (5, 3) and (9, 7) for both methods.

Table 1: Complexity comparison of convolution and lifting DWT [8]

Filter	Multiplication/Shifts		Additions	
	Convolution	Lifting	Convolution	Lifting
(5,3) Lossless	4	2	6	4
(9,7) Lossy	9	5	14	8
(2,6)	2	3	6	4
(2,10)	3	4	10	6

From the Table.1 it is clear that the lifting scheme will be more suitable for hardware implementation of DWT with limited on-chip memory, lower computational complexity, small area and low power.

2.3.3 Boundary treatment of signals

For both modes convolution-based and a lifting-based the signals [17, 21] should first be extended periodically as shown in Fig.5 .This periodic symmetric extension is used to ensure that for the filtering operations that take place at both boundaries of the signal, one signal sample exists and spatially corresponds to each coefficient of the filter mask. The number of additional samples required at the boundaries of the signal is therefore filter-length dependent. In this method the signal is extended so as it becomes periodic and symmetric.

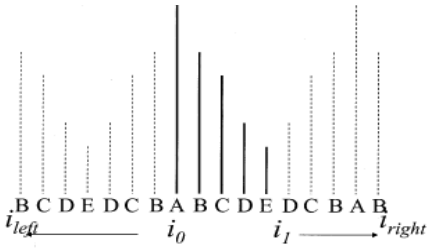


Figure 5. Boundary treatment of signals

The mirror extension can be seen on the left and on the right of the original line (A-B-C-D-E). The values of I_{left} and I_{right} parameters are chosen in relation to the filter length. This table explains, for example, that if I0 is even, we have to extend the line with 1 coefficient to the left when (5, 3) transform is used, or 3 coefficients to the left with the (9, 7) transform. In VLSI implementation the embedded mirror symmetric is implemented into the data by changing the operation process at the beginning and end of the lifting operation using multiplexers.

Table 2. Periodic extension of signals

Filter type	I_{left}		I_{right}	
	I0even	I0odd	I1even	I1odd
(5,3)	1	2	2	1
(9,7)	3	4	4	3

3. HARDWARE REQUIREMENTS OF LIFTING SCHEME

3.1 Basic functional units for lifting scheme[12]

Different kinds of lifting-based DWT architectures can be constructed by combining the three basic lifting elements. Most of the applicable DWTs like (9, 7) and (5 ,3) wavelets consist of processing units, as shown in Fig.8, which is simplified as Fig.7. This unit is called the processing element (PE). The processing nodes A, B and C are input samples which arrive successively. To implement the predict unit, A and C receive even samples while B receives odd samples. On the other hand, for the update unit, A and C are odd samples and B receives even samples. Now, the structure can be used to implement (5, 3) and (9, 7) wavelets is shown in Fig.7 & Fig.8. In this architecture each white circle represents a PE.

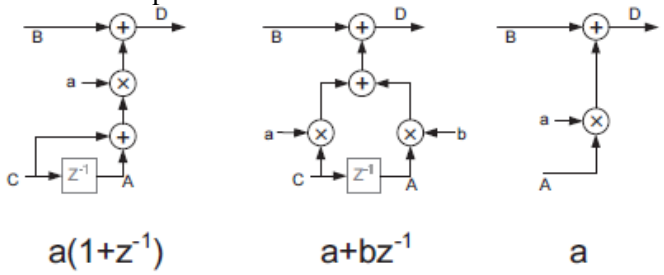


Figure 6. Basic functional units of lifting schemes

The input and output layers are essential (basic) layers and are fixed for each wavelet type, while by changing the number of extended layers, the type of wavelet can be changed accordingly. For example, omission of a single extended (added) layer in the Fig.8 structure will change the related architecture from (9, 7) type to (5, 3) type as in Fig.7. The black circles represent needed stored data for computing outputs (s, d). R0, R1 and R2, are registers that get their values from new input samples and are called data memory. The other three black circles which store the results of previous computations are known as temporary memory.

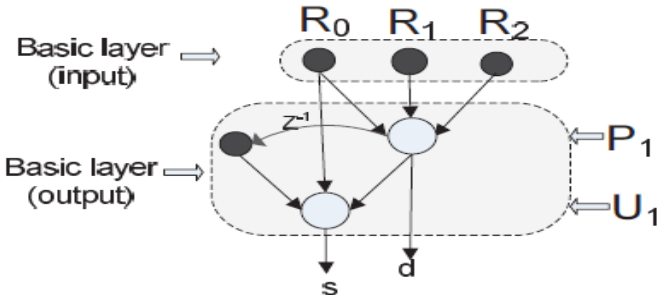


Figure 7. Lifting structure for (5, 3) wavelet

The number of data memory registers is constant and is equal to 3, while the number of temporary memory registers is (2*e* + 1), where *e* is the number of extended layers [22]. This structure can be implemented by using combinatorial circuits so that, when the input samples are fed to the architecture, outputs are ready to be used after a delay time. Also, the implementation of the structure can be performed via a pipelined structure by adding some registers. The number of pipeline stages depends on the added registers. Increasing the pipeline stages increases the clock frequency, system latency and number of required registers [7]. So, the sum of the data and temporary memories in the column-wise DWT unit determines the amount of needed internal memory [12]. The pipeline registers do not affect the required internal memory [7].

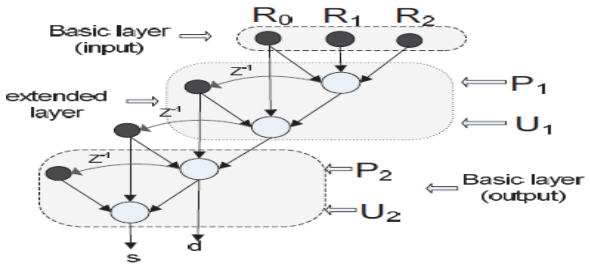


Figure 8. Lifting structure for (9, 7) wavelet

The data dependencies in the lifting scheme can be explained with the help of an example of DWT filtering with four factors as in equations [10, 11].The intermediate results generated in the first two stages for the first two lifting steps are subsequently processed to produce the high-pass (HP) outputs in the third stage, followed by the low-pass (LP) outputs in the fourth stage. For (9, 7) filter is an example of a filter that requires four lifting steps. For the DWT filters requiring only two factors, such as the (5, 3) filter, the intermediate two stages can simply be bypassed.

3.1.1 Minimizing hardware architectures-parallel and direct mapped architectures [23]

A direct mapping of the data dependency diagram into a pipelined architecture was proposed by Liu et al. in [23, 9] .For lifting schemes that require only 2 lifting steps, such as the (5,3) filter consists of two pipeline stages whereas for (9,7) it requires four pipeline stages reducing the hardware utilization to be only 50% or less. The architecture can be sequentially pipelined by combining the previous output of predict stage to current output.

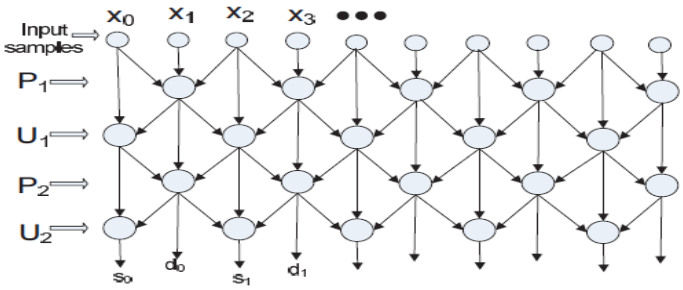


Figure 9. 1 D DWT architecture based on parallel filter method.

In this structure, U1 (0) represents the current output of the U1 unit and P1 (-1) represents the previous output of the P1 unit, and so on. The control signal **S**, which has four states, selects the inputs of the multiplexers sequentially. In the first state, two consecutive input samples arrive and the P1 function with **a** coefficient is performed on them. In the second state, the U1 function with

b coefficient will be imposed on the result of the previous state (first state's output). Similarly, in the third and fourth states, computations for P2 and U2 units will be performed on the results of the previous states. Thus, P2 and U2 produce final outputs for the structure. These steps can be used for parallel processing of inputs.

Table 3. Data flow for (9, 7) hardware architecture

S	IN1	IN2	OUT	F (factor)
0	I1	I0	P1	a
1	I0(-1)	P1	U1	b
2	P1(-1)	U2	P2	c
3	U1(-1)	P2	U2	d

The conventional lifting architectures for (5, 3) and (9, 7) consists of basic processing elements as shown in Fig.10.The cascaded blocks differ only in multiplier's coefficients. The delay unit represented by z^{-1} is implemented by one register. Each delay unit contains two consecutive registers. As shown in Fig.10 the architecture contains one P and one U unit for (5, 3) wavelet.

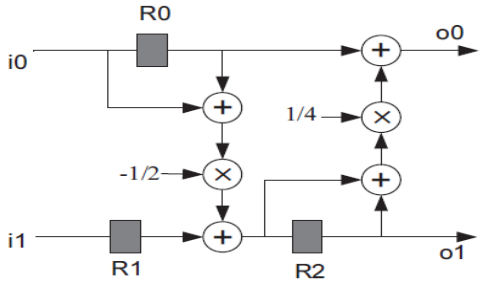


Figure 10. Lifting based hardware architecture for (5, 3) wavelet

From the (5,3) wavelet implementation of the proposed architecture it is clear that only the number of coefficients and delay block registers, that is, the z^{-1} blocks, have been modified from four to two. So, changing the wavelet type changes these two quantities, coefficients and registers, only. The architecture for lossy (9, 7) wavelet is shown in Fig.11.

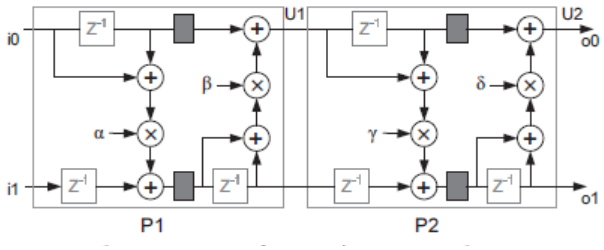


Figure 11. Direct architecture for (9, 7) wavelet

3.1.2 Folded architectures [24, 9, 10]

The folded structure is an alternative for the direct mapped architectures as in Fig.12 .by which the lifting-based structures can be designed systematically. In folded structure, the output of the PE unit is fed back through the delay registers to the PE's input. By adding different numbers of delay registers and coefficients with PE, the structure for different wavelets can be designed.

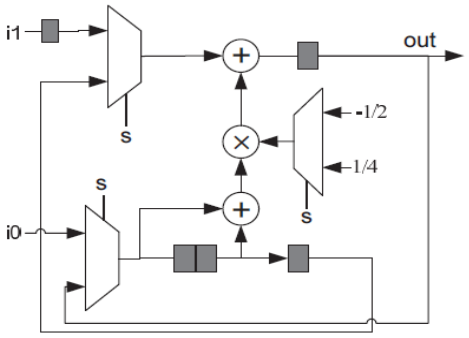


Figure.12 Folded architecture for (5, 3) wavelet [10]

For example the folded structure for (5, 3) and (9, 7) wavelets have two and four delay registers, respectively. Also the coefficients for (5, 3) wavelet are $-1/2$ and $1/4$ while for (9, 7) they are a,b,c,d. The architecture can be reconfigured so that computation of the two phases can be interleaved by selection of appropriate data by the multiplexers. As a result, two delay registers (D) are needed in each lifting step in order to properly schedule the data in each phase. Based on the phase of interleaved computation, the coefficient for multiplier M1 is either a or c, and similarly the coefficient for multiplier M2 is b or d. The hardware utilization of this architecture is always 100% and the critical path in the multiplier can be reduced. The architecture for (9, 7) filter is shown in Fig.13 .

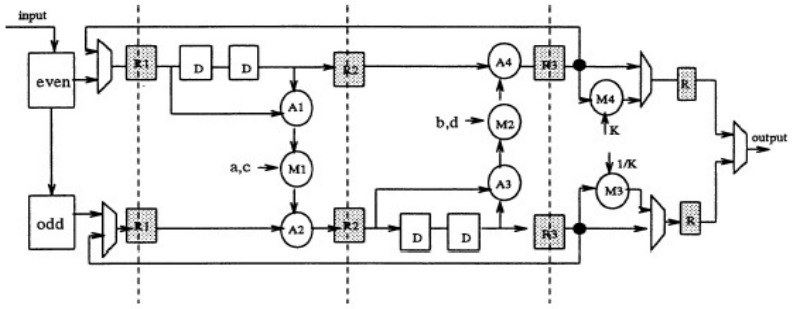
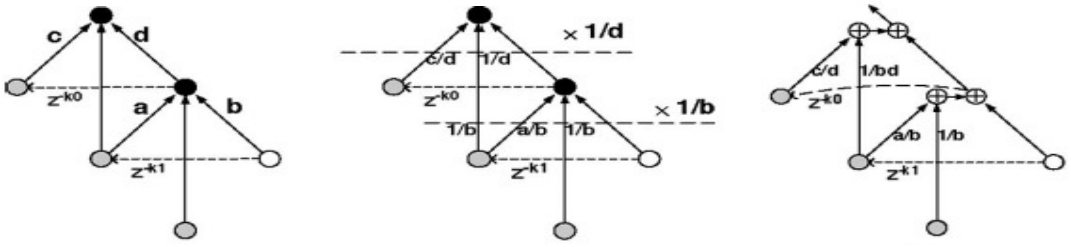


Figure 13. Folded architecture for (9,7) wavelet.[9]

3.1.2 Flipping Architecture [7, 9]

While conventional lifting-based architectures require fewer arithmetic operations, they sometimes have long critical paths.The critical path of the lifting-based architecture for the (9,7) filter is 4Tm +8Ta while that of the convolution implementation is Tm + 4Ta. One way of improving this is by pipelining which results in a significant increase in the number of registers. For instance, to pipeline the lifting-based (9,7) filter such that the critical path is Tm + 2Ta, 6 additional registers are required. Huang et al. [7] proposed a very efficient way of solving the timing accumulation problem. The basic idea is to remove the multiplications along the critical path by scaling the remaining paths by the inverse of the multiplier coefficients. In the Fig.14 shown the critical path is reduced from 2Tm+3Ta to Tm+3Ta and the reduction rate will increase as the number of serially connected computing units becomes larger.



(a) (b) (c)

Figure .14: Flipping architecture [7]. (a) Two connected computing units (b) Flipping computing units(c) After splitting the computing units and merging the multipliers.

3.1.3 Recursive Architecture [11]

The conventional DWT architectures compute the i^{th} level of decomposition upon completion of the $(i-1)^{th}$ level of decomposition. For a finite-length input signal, the number of input samples is always greater than the total number of intermediate low-frequency coefficients to be processed at the second and higher stages. The same data path can be used to interleave the calculation of the higher stage DWT coefficients while the first-stage coefficients are being calculated.. This is the basic principle of recursive architecture [11]. Here computations in higher levels of decomposition are initiated as soon as enough intermediate data in low-frequency sub band is available for computation. The basic circuit elements used in this architecture are delay elements, multipliers and MAC units which are in turn designed using a multiplier, an adder and two shifters. The multiplexers M1 and M2 select the even and odd samples of the input data as needed by the lifting scheme. S1, S2 and S3 are the control signals for data flow of the architecture. The select signal (S1) of the multiplexers is set to 0 for the first level of computation and is set to 1 during the second or third level computation. The switches S2 and S3 select the input data for the second and third level of computation. The multiplexer M3 selects the delayed samples for each level of decomposition based on the clocked signals shown in Fig.15. A recursive architecture for multilevel 1D implementation of the (5, 3) filter has been proposed in [15]. The architecture has hardware complexity and the control signals are very complex and it is regular.

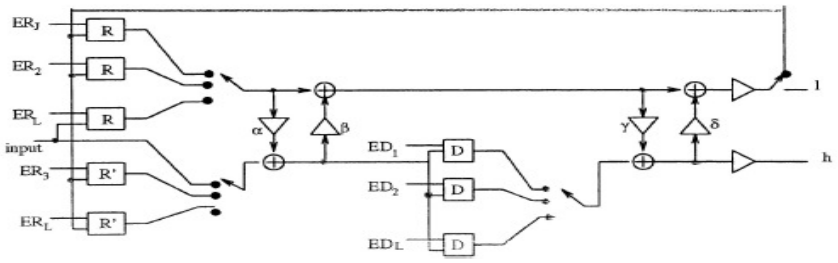


Figure.15. Recursive architectures [14, 15]

3.1.4 Other 1D DWT architectures

In [11], Liao et al. presented a 1D dual scan architecture (DSA) for DWT that achieves 100% data path hardware utilization by processing two independent data streams together using shared functional blocks in an interleaved fashion. During the DWT computation, the input samples are shifted in and the low-frequency coefficients are stored in the internal memory. After all the input samples in one stage are processed, the stored coefficients are retrieved to start computation in the

next stage. Since DSA performs useful calculation in every clock cycle, its hardware utilization for the processing element is effectively 100%.

The architecture proposed by Andra et al. [8, 9] is an example of a highly programmable architecture that can support a large set of filters. These include filters (5,3), (9,7), C(13,7), S(13,7), (2,6), (2,10), and (6,10).Since the data dependencies in the filter computations can be represented by at most four stages, the architecture consists of four processors, where each processor is assigned computations of one stage. The processor architecture consists of adders, multipliers and shifters that are interconnected in a manner that would support the computational structure of the specific filter.

A filter independent DSP type parallel architecture has been proposed by Martina et al. in [25]. The architecture consists of Nt = maxi{k_{si}, k_{ti}} number of MAC (Multiply-Accumulate) units, where k_{si} and k_{ti} are length of the primal and dual lifting filters si and ti respectively, in step i of the lifting factorization. The architecture is designed to compute nt simultaneous partial convolution products selected by the MUX, where nt is the length of filter tap for the lifting step being currently executed in the architecture. The architecture [17] can be programmed to support a wide range of filters including (9,7), (10,18),(13,11), (6,10), (5,3) and (9,3).sThe architecture in [15, 20] can be used for implementation of multilevel lifting schemes sharing the resources.

3.1.5 Comparison of Performance of the 1D Architectures

A summary of the hardware and timing requirements of the different (9, 7) filter implementations for data size N is presented in table.4. The hardware complexity has been compared with respect to the data path. An estimate of the controller complexity has also been included. The timing performance has been compared with respect to two parameters: the number of clock cycles to compute L levels of decomposition and the delay in the critical path. The term Tm stands for the delay of a multiplier, Ta the delay of an adder . In terms of hardware complexity, the folded architecture in [24,15] is the simplest and the DSP-based architecture in [25] is the most complex. All other architectures have comparable hardware complexity and primarily differ in the number of registers and multiplexer circuitry. The control complexity of the architecture in [23] is very simple. In contrast, the number of switches, multiplexers and control signals used in the architectures of [11, 14, 28] is quite large. The control complexity of the remaining architectures is moderate. In terms of timing performance, the architectures in [7,8,17,19,24,] are all pipelined architectures having the highest throughput (1/Tm). The architecture in [11] has fewer cycles since it is RPA based, but its clock period is higher. Finally, all the architectures with the exception of [11] compute all the outputs of one level before starting computations of the next level. The architecture in [11] is the only one that adopts an RPA based approach and intersperses the computations of the higher levels with those of the first level. So it is likely that the memory requirements of [11] would be lower than the others.

Table 4 .Comparison of various 1D DWT architectures

Architecture	**Datapath**	**Timing Requirements**		**Control complexity**
		No. of cycles	**Clock period**	
Direct mapped[23]	4 Mul,2 scaling MUL,8 A,8 R,6 D	$4+2N(1-1/2^L)$	T_m+2T_a	Simple
Folded[24]	4 Mul,2 scaling MUL,8 A,8 R,6 D	$4+2N(1-1/2^L)$	$2T_m+2T_a$	Moderate
Flipping[7]	4 Mul,2 scaling MUL,8 A,10 R,6 S	$5+2N(1-1/2^L)$	T_m	Moderate
Generalized[8]	4 processors(each with 1 Mul,2 A,2 R),2 scaling Mul	$12+2N(1-1/2^L)$	T_m	Moderate
Recursive[11]	4 Mul,2 scaling Mul,4 A,7 R,3 D,6 Mux	$N+L+2^L$	4Tm+8Ta	Complex
MIMOA[20] M levels(2,4,8…)	8 M adders,4M Mul,10 R for 1 SISO,	N^2/M	MTm+MTa	Simple

(A: adders: Registers: Delay units, S: Shifters, SISO: Single input single Output, L: decomposition levels)

4.2. DWT ARCHITECTURES

4.1. Two-Dimensional Discrete Wavelet Transform

The main challenges in the hardware architectures for 1-D DWT are the processing speed and the number of multipliers and adders while for 2-D DWT it is the memory issue that dominates the hardware cost and the architectural complexity. A 2-D DWT is a separable transform where 1-D wavelet transform is taken along the rows and then a 1-D wavelet transform along the columns as shown in Fig.16. The 2-D DWT operates by inserting array transposition between the two 1-D DWT.The rows of the array are processed first with only one level of decomposition. This essentially divides the array into two vertical halves, with the first half storing the average coefficients, while the second vertical half stores the detail coefficients. This process is repeated again with the columns, resulting in four sub-bands within the array defined by filter output.as in three-level decomposition

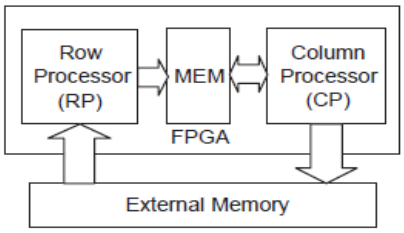


Figure 16. A Generic 2D DWT processor

The LL sub-band represents an approximation of the original image, the LL1 sub-band can be considered as a 2:1 sub-sampled version of the original image. The other three sub-bands HL1, LH1, and HH1 contain higher frequency detail information. This process is repeated for as many levels of decomposition as desired. The JPEG2000 standard specifies five levels of decomposition [21], although three are usually considered acceptable in hardware. In order to extend the 1-D filter to compute 2-D DWT in JPEG2000, two points have to be taken into account:

Firstly, the 1-D DWT generates the control signal memory to compute 2-D DWT and manages the internal memory access. Secondly, we need to store temporary results generated by 2-D column filter. The amount of the external memory access and the area occupied by the embedded internal buffer are considered the most critical issues for the implementation of 2D-DWT. As the cache is used to reduce the main memory access in the general processor architectures, in similar way, the internal buffer is used to reduce the external memory access for 2D-DWT. However, the internal buffer would occupy much area and power consumption.

Three main architecture design approaches were proposed in the literature with the aim to implement efficiently the 2D-DWT [26] level by level, line-based and block based architectures. These architectures address this difficulty in different ways. A typical **level-by-level architecture** as in Fig.17 uses a single processing module that first processes the rows, and then the columns. Intermediate values between row and column processing are stored in memory. Since this memory must be large enough to keep wavelet coefficients for the entire image, external memory is usually used. Access to the external memory is sometimes done in row-wise order, and sometimes in column-wise order, so high-bandwidth access modes cannot be used. As a result,

external memory access can become the performance bottleneck of the system for the given J level of decomposition.

$$4 \times (1 + \frac{1}{4} + \frac{1}{16} + \cdots + (\frac{1}{4})^{J-1}) \times N^2$$

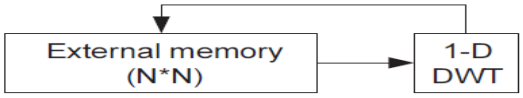


Figure 17. Level by level method

A line-based architecture scans input image row by-row manner to produce the wavelet coefficients.The line based architecture needs only few lines of the image to be stored whereas traditional methods almost need the whole image or tile of image to be memorized. Thus, this technique does not require external memory to store the intermediate data. Instead, some internal buffers are used to store the intermediate data, and the required memory size is proportional to image width or height. It can be implemented in two forms: single level and multilevel. In the line-based single level method, which is shown in Fig.18 each level of DWT is performed by a 2-D DWT block. In this method, only internal memory is used to compute one level DWT for both the row and column directions, hence, there is no external memory access during the computation of one level 2-D DWT. The required internal memory is the sum of the data memory and the temporal memory (black circles shown in Fig. 7 & Fig.8) for each line. So the amount of needed internal memory is 6N for (9, 7) wavelet and 4N for (5, 3) wavelet [26]. But the results of the DWT in the row direction for even rows can be used for the computation of the DWT in the column direction without storing them. Hence, the required internal memory for (9, 7) and (5, 3) 2-D wavelets is reduced to 5N and 3N, respectively. For higher-level 2-D DWT, only LL coefficients of the previous level are used, so the total number of external memory access for a J- level 2-D DWT on an N × N image is $2 \times (1 + \frac{1}{4} + \frac{1}{16} + \cdots + (\frac{1}{4})^{J-1}) \times N^2$

Figure.18. Single level line based method

The total number of external memory accesses for a J-level 2-D DWT is limited to $2N^2$, which corresponds to reading the input image for the first level and writing the final DWT results. The line-based multilevel structure as in Fig.19 is much faster than the previous structures, but it needs a larger amount of hardware and hence hardware utilization is low but the 2-D recursive architecture proposed in [11] improves the hardware utilization for the J-level 2-D DWT. The required internal memory for the (9, 7) wavelet is obtained from equation below.

$$5N \times (1 + \frac{1}{2} + \frac{1}{4} + \cdots + (\frac{1}{2})^{J-1})$$

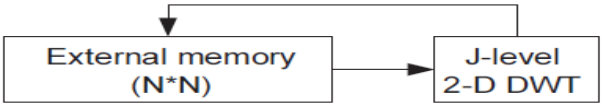


Figure 19. Multilevel line based method

In block-based architecture, the image is broken into blocks small enough to fit in an embedded memory that is processed separately. A typical Block based architecture scans the external memory block-by block, and the DWT coefficients are also computed block by-block. To perform the block-based wavelet transforms, it is necessary to store into memory an additional

row of coefficients and one additional column. The additional row is located at the top of the block, and the additional column is located to the left of the block, As a result, the input block has a size of (N +1) x (N +1) pixels. At the top and to the left of the image, the additional row or column is extracted from the extended signal. In a block-based approach, the filtering of the image boundaries should be taken into account. The treatment of this filtering has a potential impact on the visual artifacts near the boundaries. Image is divided into input blocks. Both line-based and block-based approaches have been proposed to improve the issues of memory usage and memory-access of the conventional level-by-level approach. There is a trade-off between the size of the internal memory and the number of external memory accesses in the 2-D DWT structures. The overlapped block base scanning overcomes the above mentioned problem. The block-based structure is similar to the line-based method, but instead of considering the total length of a row for DWT in the row direction, only a part of it with length M pixels is considered (Fig.20). It means that the first M columns of the main frame (the gray area in Fig. 20) are used as the input frame and 2-D DWT coefficients are computed for them. So the required internal memory, which is determined by the length of the rows, is decreased. As an example, for the (9, 7) wavelet the internal memory size will be decreased from 5N to 5M and M is a fraction of N. It is possible to consider a block of image by partitioning the image in both the row and column directions. In this method, the block (or window) slides across the image and both the row- and column-wise 1-D DWT will be performed on them [25]. The size of tile windows may be reduced to 2× 2 pixels [26].

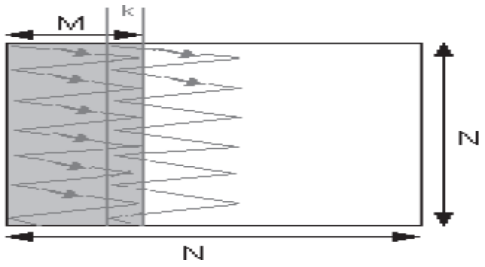


Figure 20. Scan method in Block based technique

4.2 Scan methods for block-based structure

However, in the overlapped block base method, there is a problem in the boundary region between two M-pixel sections. To compute the DWT for the beginning pixel of the next M-pixel section, values of K previous pixels are needed. These K pixels produce values of temporary memory black circles in Fig.8 for (9, 7) lifting. K is equal to $n_t - 2$, where n_t is the number of filter taps corresponding to the desired DWT and K=7. To solve the boundary problem, the overlapped scan method has been proposed in [27]. A new M-pixel section begins from the last K pixels in the previous section. So two sections are overlapped in K pixels, and this causes the number of external memory reads to be $\frac{N^2 M}{M-K}$ instead of N^2. The number of external memory writes is limited to writing the output results and is equal to N^2. The storage of temporary memory may be fulfilled in internal or external memory. If internal memory is used to save temporary data, the number of external memory accesses is equal to N^2 read and N^2 write operations.

Table.5 Comparison of memory requirement for 2D (9, 7) DWT

Type	Direct	Single level line based	Block based
Internal memory size	0	5N	5M

External memory reads	$2N^2$	N^2	N^2M ------- M-K
External memory write	$2N^2$	N^2	N^2
Control complexity	Low	Medium	Medium

From the Table.5 it is clear that the block-based structure with the overlap scan method needs only about one-half of the external memory accesses in comparison with the direct method. It is suitable for J level DWT decomposition with single level and multilevel architecture. Due to the shorter access time for internal memory, the clock pulse frequency will increase, and the the power consumption will decrease. In comparison with other methods, the new method remarkably decreases the needed internal memory at the cost of a slight increase in the number of external memory accesses. The internal memory size can be decreased by saving temporary data of overlapped region in external memory or the number of external memory access can be decreased by saving data in internal memory.

4.3.1. 2D DWT Architecture in [8]

The architecture proposed by Andra et al. [8] is more generalized and can compute a large set of filters for both the 2D forward and inverse transforms. It supports two classes of architectures based on whether lifting is implemented by one or two lifting steps. The M2 architecture corresponds to implementation using one lifting step or two factorization matrices, and the M4 architecture corresponds to implementation by two lifting steps or four factorization matrices.

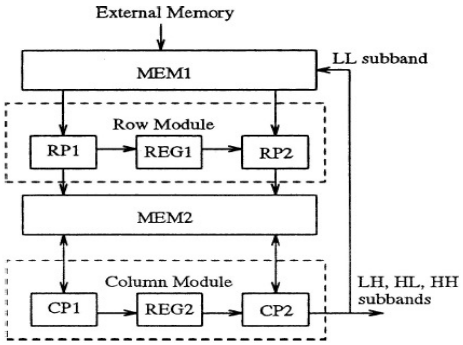
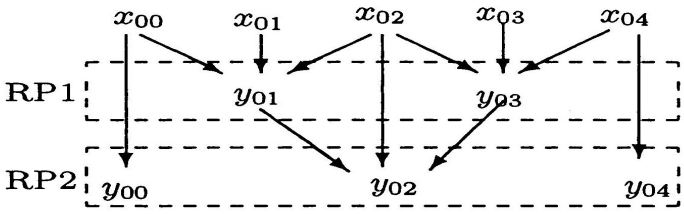


Figure 21. Block diagram of generalized M2 architecture [8]

It consists of the row and column computation modules and two memory units, MEM1 and MEM2. The row module consists of two processors RP1 and RP2 along with a register file REG1, and the column module consists of two processors CP1 and CP2 along with a register file REG2. All the four processors RP1, RP2,CP1, CP2 in the proposed architecture consists of 2 adders, 1 multiplier and 1 shifter as shown in Fig.21.



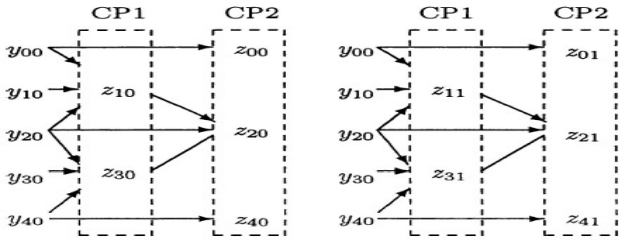


Figure.22 Data access patterns for the row (RP) and column modules(CP) for the (5,3) filter as in [8].

For the M2 architecture, RP1 and CP1 are predict filters and RP2 and CP2 are update filters .Fig.22 illustrates the data access pattern for the (5 ,3) filter with N = 5. The column processor CP1 and CP2 start computations as soon as the required elements are generated by row processorRP1 and RP2. The memory modules, MEM1 and MEM2, are both dual port with one read and one write port, and support two simultaneous accesses per cycle. MEM1 architecture consists of two banks and MEM2 architecture consists of four banks. The multi-bank structure increases the memory bandwidth and helps support highly pipelined operation.

4.3.2. 2D Recursive Architecture[11]

As in the 1D [11] case, the computations of all the lifting stages are interleaved to increase the hardware utilization. A simplified block diagram of this architecture is shown in Fig. 23. The column processor and the row-processor are similar to the 1D recursive architecture processor. The image is input to the row-processor in raster scan format. When the row-processor processes the even rows, the high- and low-frequency DWT coefficients of the odd rows are shifted into their corresponding first-in first-out FIFO registers. The use of two FIFO is to separately store high frequency and low frequency components results in lower controller complexity. When the row processor processes the odd lines, the low-frequency DWT coefficients of the current line and lines previously stored in the FIFOs are sent to the column processors. The column-processor starts calculating the vertical DWT in zigzag scan format after one row delay. The computations are arranged in a way that the DWT coefficients of the first stage are interleaved with the other stages. The arrangement is done with the help of the data arrangement switches at the input to the row and column processors, and the exchange switch.

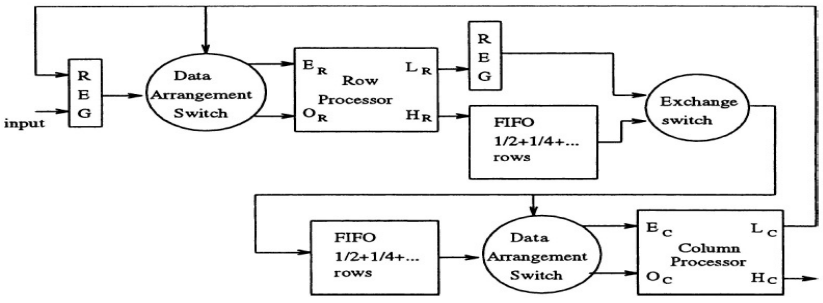


Figure.23 Block diagram of the 2D recursive architecture in [15].

4.3.3. Other 2D DWT Architectures

The architecture proposed by Andra et al. [8] is more generalized and can compute a large set of filters for both the 2D forward and inverse transforms. MEM1 architecture consists of two banks and MEM2

architecture consists of four banks. The multi-bank structure increases the memory bandwidth and helps support highly pipelined operation.

The 2D recursive architecture proposed by Liao et al. [11] .As in the 1D case, the computations of all the lifting stages are interleaved to increase the hardware utilization. The column processor and the row-processor are similar to the 1D recursive architecture processor. The arrangement is done with the help of the data arrangement switches at the input to the row and column processors, and the exchange switch and the memory requirement is low and control complexity is higher.

A mix of the principles of recursive pyramid algorithm (RPA) [11] and folded architecture has been adopted by Jung et al. to design a 2D architecture for lifting based DWT in [14]. The row-processor is a 1D folded architecture and does row-wise computations in the usual fashion. The column processor is responsible for filtering along the columns at the first level and filtering along both the rows and the columns at the higher levels. It does this by employing RPA scheduling and achieves very high utilization. The utilization of the row processor is 100%, and that of the column processor is 83% for 5-level decomposition.Wu *et al.* [1] have proposed a line-based scanning scheme and a folded architecture for the computation of multilevel 2-D DWT level-by-level. By line-based scanning, a few rows of intermediate output matrix are buffered to perform the column-wise processing. Consequently, the transposition buffer size is reduced from O (M*N) to O (N) , where M and N are, respectively, the height and width of the input image. The folded structure of [1], therefore, requires a small internal buffer, involves simple control circuitry and performs multilevel DWT computation using frame-buffer of size with 100% HUE. An external frame-buffer is used in this structure to store the low-low sub band components of the current decomposition level for the calculation of sub band coefficients of the next higher levels. Several other folded structures also have been proposed [4, 7, 10, 17, 19, 20 ,27, 29] for efficient implementation of lifting 2-D DWT. All these structures differ in terms of size of arithmetic-unit, on-chip memory, cycle period and average computation time (ACT). The dual scan 2-D architecture proposed by Liao *et al.* [11] requires two streams of input samples and computes the folded multilevel DWT computation with 100% HUE. The folded design proposed by Barua et al. [10] uses an embedded symmetric data extension scheme for 9/7 DWT computations. The structures of [19, 29, 17] have used the embedded decimation technique for low-complexity implementation of 2-D DWT using 9/7 filters.Cheng *et al.* [15, 20] have proposed the multiple lifting scheme and -scan method to reduce the memory (line-buffer) access for minimizing the overall power consumption at the cost of a few local registers.

The existing folded designs and recursive designs have some inherent difficulty for efficient realization of 2-D DWT. The folded designs require small internal buffer and simple control circuitry but involve large frame-buffer, where the frame-buffer not only contributes to the power budget, but also introduces significant delay in multilevel DWT computation. A significant amount of memory bandwidth is also wasted by accessing the external buffer which ultimately limits the throughput rate. The recursive structures eliminate the requirement of external buffer but involve complex control circuits and more internal memory than the folded structures. The HUE of the recursive designs is also not favorable for efficient realization of the 2-D DWT core. The line-buffer and frame-buffer are found to contribute almost 90% of the chip area and power dissipation in the existing structures. The requirement is to design a new method to overcome the above mentioned short comings.

4.3.4 MULTILEVEL ARCHITECTURES FOR DWT

The steps given in [Equation 11] are used to construct an efficient multi input/ multi-output VLSI architecture (MIMOA) based on lifting scheme. It has high processing speed requirement with controlled increase of hardware cost and simple control signals. High processing speed can be achieved when multiple row data samples are processed simultaneously. And time multiplexing technique is adopted to control the increase of the hardware cost for the MIMOA. Furthermore, the control signals are simple, since the regular architecture is a combination of simple single-input/ single-output (SISO) modules and two-input/two-output (TITO).modules. It provides a

variety of hardware implementations to meet different processing speed requirements by selecting different throughput rates.

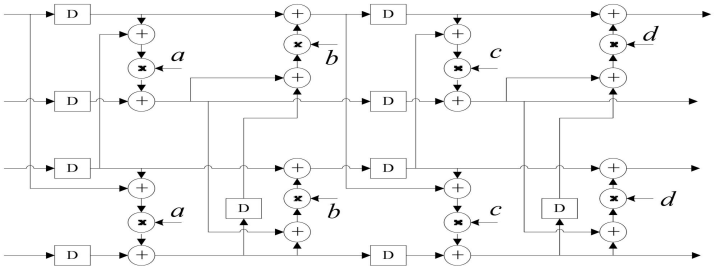


Figure. 23 Four-input/four-output architecture for processing four elements per clock cycle.[20]

The multiple parallel processing implementation results in increasing processing element number that raises the cost of DWT core. Folding of the coefficients is done to minimize hardware and it can be extended to 2D DWT.

4.3.5 Comparison of 2D DWT Architectures

A summary of the hardware and timing requirements of a few representative architectures is presented in shown in Table 4 for (5,3) and (9,7) wavelet transform for the given J level of decomposition. From the Table.4 it is clear that the internal memory requirement of recursive architectures is almost zero but the control complexity is high. Folded architectures are efficient in hardware and it can be extended to parallel processing of multilevel input samples.

Table 4: Comparison of various 2D DWT architectures

Architecture	**Multiplie-rs/ Shifters**	**Adders**	**Memory**	**Compu-ting Time**	**Out-put Late -ncy**	**DWT Mode**	**HUE %**	**Bound-ary Proces-sing**	**Control Comple-xity**
Wu[1] (5, 3) (9, 7)	16	16	$N^2/4$	$2N^2(1-4^J)/3$	2N	CB	100	ZP	Medium
	32	32	$N^2/4$	$2N^2(1-4^J)/3$	4N	CB	100	ZP	
Liao+RA[11,28] (5, 3) (9, 7)	4	8	0	N^2	2N	LB	66.7	ZP	Complex
	12	16	0	N^2	4N	LB	66.7	ZP	
Barua[10] (5, 3	4	8	$N^2/4$	$2N^2(1-4^J)/3$	5N	LB	100	EBDE	Simple
(9, 7)	12	16	$N^2/4$	$2N^2(1-4^J)/3$	7N	LB	100	EBDE	
Andra [8] (5, 3) (9, 7)	4	8	0	$2N^2(1-4^J)/3$	2N	LB	100	SSO	Medium
	6	8	0	$4N^2(1-4^J)/3$	$N^2/4$	LB	100	SSO	
FA [29] (5, 3) (9, 7)	4	8	$N^2/4$	$2N^2(1-4^J)/3$	$N^2/4$	LB	100	EBDE	Simple
	10	16	$N^2/4$	$2N^2(1-4^J)/3$	$N^2/4$	LB	100	EBDE	
FA[19]	1 shifters	2	$N^2/4$	$]N^2/4$	N^2	LB	100	EBDE	Simple
XinTian[20] MIMOA (M=2)	8	16	$N^2/2$	N^2/M	M	LB	100	Not discussed	Simple
MIMOA (M=8)	32	64	$N^2/8$	N^2/M					

5. CONCLUSION

In this paper, we have analyzed the existing Lifting based 1-dimensional and 2-dimensional Discrete Wavelet Transform based on the hardware complexities and computational time for the different architectures using Lifting schemes. The architectures represented vary from direct mapped, folded, recursive to multilevel folded architectures. This review is useful for exploring a new method of pipelined architectures capable of handling multiple data streams suitable for application in image and video processing multimedia real time applications.

REFERENCES

[1] P.C. Wu and L.G. Chen ,(2001), "An efficient architecture for two-dimensional discrete wavelet transform",IEEE Transaction on Circuit and Systems and Systems for Video Technology, Volume 11, No. 4, Pages 536-545.

[2] MiladGhantous, MagdyBayoumi, (2011), "P^2E-DWT: A Parallel and Pipelined Efficient VLSI Architecture of 2-D Discrete Wavelet Transform",. IEEE.

[3] S.Mallat ,(1989), "A theory for multiresolution signal decomposition: the wavelet representation", IEEE Trans. Pattern Anal. Mach. Intell. 11, 674–693.

[4] Chao Cheng and Keshab K. Parhi,(2008), ".High-Speed VLSI Implementation of 2-D Discrete Wavelet Transform",IEEE Transactions on Signal Processing, Vol. 56, No. 1.

[5] Keshab K. Parhi and Takao Nishitani,(1993), "VLSI Architectures for Discrete Wavelet Transforms".IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. **1,** No. 2.

[6] I. Daubechies, W. Sweldens,(1998), " Factoring wavelet transform into lifting steps", J. Fourier Anal. Appl. 4, 247–269.

[7] Chao-Tsung Huang, Po-Chih Tseng, and Liang-Gee Chen,(2004), ".Flipping Structure: An Efficient VLSI Architecture for Lifting-Based Discrete Wavelet Transform", IEEETransactions On Signal Processing, Vol. 52, No. 4.

[8] Kishore Andra,ChaitaliChakrabathi, and Tinku Acharya(2002), " A VLSI architecture for Lifting based Forward and inverse wavelet transform",IEEE Trans, on Signal processing,VOL.50.No.4.

[9] T. Acharya, C. Chakrabarti,(2006), " A survey on lifting-based discrete wavelet transforms architectures",J.VLSI Signal Process.42, 321–339.

[10] S. Barua, J.E. Carletta, K.A. Kotteri and A.E. Bell,(2004), " An efficient architecture for lifting-based two-dimensional discrete wavelet transforms*"* .The VLSI Journal 38, 341-352.

[11] L.Hongyu, M.K. Mandal, B.F. Cockburn,(2004), "Efficient architectures for 1-D and 2-D lifting-based wavelet transforms",IEEE Trans. Signal Process.52 (5), 1315–1326.

[12] S.A. Salehi, S. Sadri,(2009), ".Investigation of Lifting-Based Hardware Architectures for Discrete Wavelet Transform",Journal of Circuits Systems and Signal Processing, vol. 28.

[13] M.S.Bhuyan,Nowshad Amin, MD.Azrul Hasni Madesa,(2007), ".An Efficient VLSI Implementation of Lifting Based Forward Discrete Wavelet Transform Processor for JPEG2000", Proceedings of the 7th WSEAS International Conference on Signal, Speech and Image Processing, Beijing, China.

[14] Gab Cheon Jung, Duk Young Jin,Seong Mo Park,(2004), "An Efficient Line based VLSI Architecture for 2D Lifting DWT".IEEE.

[15] P.Y Chen,(2004), "VLSI implementation for one-dimensional multilevel lifting based wavelet transform",IEEE Trans. on Computers, vol. 53, pp.386-398, 2004.

[16] Peng Cao, XinGuo, and Chao Wang,(2007), "Efficient architecture for two dimensional discrete wavelet transform based on lifting scheme", IEEE 7th International Conference on, pp. 225-228.

[17] B.-F.Wu and C.F. Lin,(2005), "A high-performance and memory-efficient pipeline architecture for the5/3 and 9/7 discrete wavelet transform of JPEG2000 codec", IEEE Trans. on Circuit and Systems for Video Technology, vol. 15, pp. 1615-1628.

[18] A. Mansouri, A. Ahaitouf, and F. Abdi ,(2009), " An Efficient VLSI Architecture and FPGA Implementation of High-Speed and Low Power 2-D DWT for (9, 7) Wavelet Filter", International Journal of Computer Science and Network Security, VOL.9 No.3.

[19] Ping ping Yu, Suying yao, Jiangtao Xu,(2009),"An Efficient architecture for 2D lifting based discrete wavelet transform",ICIEA , PP 3667-3670.

[20] XinTian, Lin Wu, Yi-Hua Tan, and Jin-WenTian (2011),"Efficient Multi-Input/Multi-Output VLSI Architecture for Two-Dimensional lifting-Based Discrete Wavelet Transform", IEEE Transactions On Computers, Vol. 60, No. 8.

[21] Charilaos Christopoulos, Athanassios Skodras and Touradj Ebrahimi,(2000), "The JPEG2000 Still Image Coding System: an overview",IEEE .
[22] W.-H. Chang, Y.-S.Lee, W.S. Peng, C.-Y. Lee,(2001), "A line-based, memory efficient and programmable architecture for 2D DWT using lifting scheme" in IEEE International Symposium on Circuits and Systems, vol. 4.
[23] C.C. Liu,Y.H. Shiau, and J.M. Jou,(2000), "Design and Implementation of a Progressive Image Coding Chip Based on the Lifted Wavelet Transform" in Proc. of the 11th VLSI Design/CAD Symposium, Taiwan.
[24] C.J Lian, K.F. Chen, H.H. Chen, and L.G. Chen,(2001), "Lifting Based Discrete Wavelet Transform Architecture for JPEG2000", IEEE International Symposium on Circuits and Systems, Sydney, Australia.
[25] M.Martina,G.Masera,G.Piccinini,and M.Zamboni,(2003),"Novel JPEG 2000 Compliant DWT and IWT VLSI implementations",Journal of VLSI Signal Processing, vol.34. pp. 137–153.
[26] Sayed Ahmad Salehi, Rasoul Amir fattahi,(2011), "A block- based 2D DWT structure with new scan method for overlapped sections", IEEE.
[27] UshaBhanu. N, Dr.A.Chilambuchelvan,(2011) , "Efficient VLSI Architecture for Discrete Wavelet Transform", IJCSI International Journal of Computer Science Issues. ISSN (Online): 1694-0814.
[28] H. Liao, M.K. Mandal, and B.F. Cockburn,(2002), "Novel Architectures for Lifting-Based DWT" in Electronics Letters, vol. 38, no. 18,pp. 1010–1012.
[29] ChengyiXiong, Jin wen Tian, and Jian Liu,(2007), "Efficient Architectures for Two-Dimensional DWT using Lifting Scheme", IEEE Transactions on Image Processing, Vol.16, No.3.
[30] K.K.Parhi.(1999), "*VLSI Digital Signal Processing Systems: Design and implementation",*New York Wiley Publications.

Authors Biography

Ms.UshaBhanu.N. received the B.E. degree in E.C.E from Bharathiar University in 1996 and M.E. degree in VLSI Design in 2006 from Anna University. She is currently pursuing her Ph.D. in the faculty of Information and Communication Engg. in Anna University, Chennai, India. She is working as Assistant Professor in Department of E.C.E in Meenakshi College of Engineering, Chennai and has more than 14 years of teaching experience. Her research area includes VLSI signal processing and image processing.

Dr.A.Chilambuchelvan received his Ph.D. degree in Embedded systems in the faculty of Information and Communication Engineering in the year 2008 from College of Engineering, Anna University, Chennai. He has completed his B.E. (E.C.E) and M.E. (Applied Electronics) in the year 1989 and 1994 respectively. He has more than 21 years of teaching experience. He is currently working as Professor and Head of the Department in EIE in R.M.D. Engineering College affiliated to Anna University, Chennai. His research area of interest includes Embedded systems, VLSI design, Soft computing and Bio medical engineering. He has published 12 papers in international journals and conferences.