Sensing and Imaging (2020) 21:53
https://doi.org/10.1007/s11220-020-00317-z

ORIGINAL PAPER

An Efficient Architecture for Modified Lifting-Based Discrete Wavelet Transform

Rohan Pinto[1] · Kumara Shama[2]

Received: 31 March 2019 / Revised: 8 August 2020 / Accepted: 10 October 2020 /
Published online: 26 October 2020

Abstract

A high speed and memory efficient lifting based architecture for one-dimensional (1-D) and two-dimensional (2-D) discrete wavelet transform (DWT) is proposed in this paper. The lifting algorithm is modified in this work to achieve a critical path of one multiplier delay with minimum pipeline registers. A 1-D DWT structure with two-input/two-output and four-input/four-output is developed based on the modified lifting scheme. The proposed 2-D DWT architecture for the Daubechies 5/3 and 9/7 filter comprises of two 1-D processors, together with a transpose and a temporal memory. An efficient transpose block is presented, which utilizes three registers to transpose the output sequence of the 1-D DWT block. The transpose block is independent of the size of the image read for the transform. The scanning process of an $N \times N$ image for a one-level 2-D transform is in Z fashion to minimize the temporal buffer to $4N$ and $2N$ for the 9/7 and 5/3 mode DWT respectively. The comparison results show that the proposed structure is hardware cost-effective and memory efficient, which is favorable for real-time visual operations. The model is described in VHDL and synthesized using the Cadence tool in 90 nm technology.

Keywords Discrete wavelet transform (DWT) · Lifting scheme · 1-D DWT · 2-D DWT · Pipeline · VLSI architecture

1 Introduction

The discrete wavelet transform (DWT) has established itself as an efficient tool for many applications, such as speech analysis, signal analysis, numerical analysis, video processing and compression due to its time-frequency localization characteristics [1]. It has been adapted in the JPEG-2000 standard due to its ability to

✉ Rohan Pinto
rohanp@sjec.ac.in

[1] St Joseph Engineering College, Mangalore, Karnataka, India

[2] Manipal Institute of Technology, Manipal, Karnataka, India

decorrelate large images. The DWT is computationally intensive, which make it a challenge to implement. It requires a sizable quantity of arithmetic resources and memory. Therefore, to reduce the operational complexity Daubechies and Sweldens proposed the lifting scheme [2] to construct the DWT. The lifting based architectures have many advantages over the classical convolution based structures [3] in computational complexity, power consumption, and memory.

At present, many VLSI architectures for the 2-D DWT based on both the convolution and lifting scheme are available for real-time processing. However, designing a highly efficient structure at low hardware cost is an exacting task. In this work, two DWT architectures for the biorthogonal 9/7 and 5/3 wavelet are proposed based on the modified lifting scheme. The advantages of the proposed structure are 100% hardware utilization, low critical path delay, hardware efficient, and low control complexity. The structure is flipping based, with a modest memory requirement.

The rest of this paper is organized as follows: Sect. 2 briefly reviews the different DWT structures based on the lifting scheme. Section 3 describes the classical lifting scheme and modified lifting scheme algorithm. The proposed DWT architectures are depicted in Sect. 4. Implementation and performance analysis is done in Sect. 5. Finally, in Sect. 6 conclusion is drawn.

2 Related Work

The traditional DWT architectures are based on the convolution scheme [4–7]. These structures are not suitable for VLSI implementation because of the requirement of larger chip area. Hence, in recent years, a number of novel architectures based on the lifting algorithm are proposed to reduce the operations involved in computing the DWT. Jou et al. [8] proposed a structure, which is a direct implementation of the lifting steps. It has the advantage of low implementation complexity. In order to increase the hardware utilization, Lian et al. [9] proposed a folded architecture based on the lifting algorithm, but both these architectures have limitation on the critical path delay. Chen [10] implemented a 1-D, 5/3 and 9/7 wavelet transform. Since the structure is multi-level lifting based transform, it cannot be extended directly to the 2-D transform. The structure is folded to achieve higher hardware utilization. It avoids the use of external memory to store the intermediate results, and thus reducing the delay required to access the memory. Barua et al. [11] developed an architecture for 2-D DWT of the biorthogonal 9/7 wavelet. The update and the predict steps are modified to change the computation for the first and the last pixel respectively in each row and column. The structure can achieve 100% hardware utilization. It has a regular data flow, low complexity and high throughput. It utilizes 30% less memory than the conventional filter bank scheme. Shi et al. [12] proposed an efficient architecture employing the folding technique. The hardware complexity in the structure is moderate. Although it requires minimum hardware resources, the critical path delay is the sum of T_m and T_a, where, T_m and T_a are multiplier and adder delay respectively. The throughput rate of the structure is one, with 100% hardware utilization. To speed up the computation, Lai et al. [13] introduced a two-input/two-output pipelined architecture. The predict and update stages are merged to reduce the critical path delay. The registers required for a 1-D DWT are 22, with eight pipelined stages.

The flipping based structure is another prominent architecture for DWT. It has the advantage of reducing the critical path delay. Huang et al. [14] proposed a flipping structure for DWT. It has a minimum critical path of one multiplier delay ($1T_m$) with five pipeline stages. Daubechies 9/7 filter, integer 9/7 filter, and 6/10 filter are used in the design to demonstrate the performance of the flipping structure. Wu and Lin [15] proposed a memory efficient pipelined architecture by modifying the lifting algorithm. The predictor and updater are merged to minimize the critical path delay to $1T_m$. The throughput of the structure is one-input/one-output, and this reduces the processing speed. Inverse DWT is also proposed adopting the same architecture of the forward transform. Xiong et al. [16] presented a parallel based lifting scheme (PLS) to build a VLSI architecture for DWT. This scheme reduces the critical path delay and the number of registers used to implement a 1-D DWT. Using this scheme, implementation of forward and inverse DWT can be identical. Xiong et al. [17] proposed a novel 1-D and 2-D DWT architecture. Embedded decimation technique is utilized to optimize the DWT structure. A line based 2-D DWT structure is developed using parallel and pipelined technique called the fast architecture (FA), and another line based DWT structure called the high-speed architecture (HA) is proposed based on the parallelism technique. Cao et al. [18] presented the decomposed lifting scheme (DLS) algorithm to perform a 1-D DWT. Three structures are proposed with a throughput of one-input/one-output, two-input/two-output, and four-input/four-output. Two memory efficient structures, FA and HA are constructed for 2-D transform. The computing time is reduced with these structures at the cost of multipliers and adders. Lin et al. [19] proposed a pipeline architecture for 2-D DWT. The 1-D DWT structure requires fewer pipeline registers to achieve a critical path of $1T_m$. The rescheduling algorithm is proposed to merge the lifting steps to attain a critical path same as that of the other architectures, but with fewer pipeline registers. Tian et al. [20] proposed a multi-input/multi-output structure for 2-D DWT. This architecture is used for high speed application with the computing time of N^2/M for an $N \times N$ image, and M is the throughput. It has a simple control procedure, and hardware cost is minimum. Zhang et al. [21] developed a high-speed DWT architecture for 2-D transform. The critical path delay is $1T_m$ with three pipeline stages. It is a two-input/two-output structure. The lifting scheme is modified, and the intermediate data is recombined to reduce the pipeline. A novel transpose module is developed to meet the input order of the data flow for the row processor. Hsia et al. [22] built an efficient architecture for 2-D DWT to address the issues of memory requirement and critical path delay. To reduce the transpose memory, interlaced read scan method is introduced. The structure has the advantage of regular signal flow, low transpose memory, and latency. The multiplication and accumulation cell (MAC) unit is used in the 1-D DWT architecture. Nagesh [23] developed a lifting based DWT structure for 1-D transform. The structure is developed by applying the folding and pipelining method. The serial-in parallel-out (SIPO) shift registers are used at the input of the architecture and parallel-in serial-out (PIPO) shift register at the output. Darji et al. [24] presented three architectures for multi-level 2-D

DWT. These structures use different types of input scanning methods. They reduce the total computing cycles, and the hardware utilization efficiency ranges from 60 to 100%. Darji et al. [25] proposed a flipping architecture for 2-D DWT. The structure optimizes the lifting algorithm by simultaneously computing the intermediate data beforehand. The structure has a minimum critical path delay and 100% hardware utilization. Ang et al. [27] developed a scalable architecture that supports multiple DWT blocks. Each block operates independently. The throughput rate in the design is m, where m is the number of parallel DWT blocks. The advantages of the architecture are simple control complexity, low memory, fast computation, and minimum power consumption. Basiri et al. [28] proposed lifting based 1-D and 2-D DWT architecture for the 5/3 and 9/7 filters to optimize the area, power, and delay requirement. The area is reduced by using only one processing element (PE) to perform the entire DWT operation. Multiple PE's are processed in parallel to reduce the delay. The PE consists of floating point adders, floating point multipliers, and fused multiply-add (FMA) unit. Wang and Choy [29] presented a novel data scan method for 2-D DWT. The stripe based method accesses the adjacent even and odd rows simultaneously to increase the throughput rate. The systolic array structure is proposed for the row transform, and the conventional two-input/two-output lifting based architecture is adopted for the column transform. Pipelining technique is employed to reduce the critical path delay to $1T_m$. The structure gives good area, power, and delay results for the stripe size of 2, 4 and 8.

A multi-level 2-D DWT structure using the Haar wavelet is presented by Al-Azawi [30]. The architecture operates at 209 MHz for a three-level 2-D DWT. The hardware resources and memory requirement is constant. The transposition memory is required at every level of decomposition. The throughput of the structure is one-input/one-output. A five-level 2-D DWT structure for the 5/3 filter is presented by Aziz and Pham [26]. Multiple processing blocks operate in parallel for a five-level transform. The designs in [31–33] present a memory efficient structure for multi-level 2-D DWT. The proposed architectures are based on the lifting scheme with different scanning methods being employed. Inverse DWT architecture is presented in [34, 35] for the Daubechies 5/3 and 9/7 wavelet. It is a high performance memory efficient structure which supports JPEG 2000 decoder.

The prominence of lifting-based DWT has led to the advancement of few designs in recent time. The architectures range from the parallel type, flipping type to the folded type. In this work, a pipelined structure with a low critical path delay of $1T_m$ is presented with least hardware usage based on the modified lifting scheme.

3 DWT Based on Lifting Scheme

The lifting scheme [2] is a simple and efficient algorithm to compute the DWT. This scheme is an alternate way of building the wavelet filters by the lifting steps. The block diagram in Fig. 1 depicts the three steps in the lifting scheme.

The input samples are first split into even and odd samples. The predict function is then applied on the even samples to predict the odd samples. The difference between the predicted odd samples and the actual odd samples form the high-pass

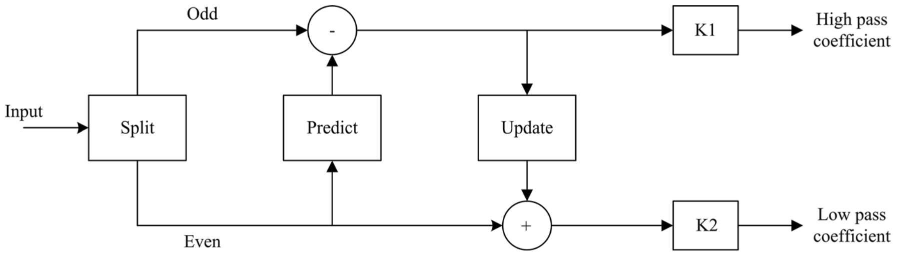

Fig. 1 Block diagram of the lifting scheme [2]

coefficients or the detail coefficients. Applying the update function on the detail coefficient and combining it with the even samples, it forms the low-pass coefficients or the approximate coefficients. Finally, the coefficients are scaled by the normalization factors, K1 and K2, to obtain the high-pass and the low-pass coefficients respectively. The following section details the conventional lifting and the modified lifting scheme.

3.1 Conventional Lifting Scheme for Discrete Wavelet Transform

The Euclidean algorithm can be used to factorize every wavelet transform into the lifting scheme [2]. The polyphase matrix of a DWT filter is resolved into a sequence of alternating upper triangular matrix and lower triangular matrix, and a diagonal matrix. The polyphase matrix of the wavelet transform can be represented as

$$P(z) = \begin{bmatrix} g_e(z) & g_o(z) \\ h_e(z) & h_o(z) \end{bmatrix} \tag{1}$$

where $h(z)$ and $g(z)$ are low-pass and high-pass filters respectively. $P(z)$ of the 9/7 filter can be factorized as

$$\begin{aligned} P(z) = &\begin{bmatrix} 1 & \alpha(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \beta(1+z) & 1 \end{bmatrix} \begin{bmatrix} 1 & \gamma(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \\ &\times \begin{bmatrix} 1 & 0 \\ \delta(1+z) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} \end{aligned} \tag{2}$$

where $\alpha(1+z^{-1})$ and $\gamma(1+z^{-1})$ are predict polynomials, $\beta(1+z)$ and $\delta(1+z)$ are update polynomials and K is a constant.

Given the input sequence $x(n)$ with $n = 0, 1, \ldots, N-1$, the lifting scheme of the 9/7 filter is mathematically described in the following three steps [15].

1. Split step

$$d_i^0 = x_{2n+1} \tag{3}$$

$$s_i^0 = x_{2n} \tag{4}$$

2. Lifting step

   (a) First lifting step

$$d_i^1 = d_i^0 + \alpha\left(s_i^0 + s_{i+1}^0\right) ------ predict \tag{5}$$

$$s_i^1 = s_i^0 + \beta\left(d_{i-1}^1 + d_i^1\right) ------ update \tag{6}$$

   (b) Second lifting step

$$d_i^2 = d_i^1 + \gamma\left(s_i^1 + s_{i+1}^1\right) ------ predict \tag{7}$$

$$s_i^2 = s_i^1 + \delta\left(d_{i-1}^2 + d_i^2\right) ------ update \tag{8}$$

3. Scaling step

$$d_i = 1/k \times d_i^2 \tag{9}$$

$$s_i = k \times s_i^2 \tag{10}$$

d_i and s_i are the high-pass and the low-pass coefficients respectively, $0 \le i \le M - 1$, where M is data length. The 9/7 filter coefficients are $\alpha = -1.586134342$; $\beta = -0.05298011854$; $\gamma = 0.8829110762$; $\delta = 1.149604398$ and $k = 1.149604398$ is the scaling coefficient. Similarly, the 5/3 filter lifting algorithm consists of one lifting step (first lifting) along with the split step and the scaling step. The two filter coefficients are $\alpha = -0.5$ and $\beta = 0.25$.

The design bottleneck of 9/7 DWT is the large critical path of one multiplier and two adder ($T_m + 2T_a$) delay. This can be improved by modifying the lifting algorithm. The data flow graph of the conventional lifting scheme for the 9/7 filter shown in Fig. 2 requires a maximum of four adders and two multipliers for the computation in a clock cycle. The latency in the scheme to produce 2-D DWT is 11 cycles delay. The high-pass and low-pass coefficients are obtained after scaling by a factor of $1/k$ and k respectively.

3.2 Modified Lifting Scheme for Discrete Wavelet Transform

In the conventional lifting scheme, intermediate data is processed serially, resulting in an extended critical path delay. Pipeline register can be employed to reduce the critical path delay to one multiplier, but the internal memory size of the 2-D DWT structure increases. Hence, the lifting algorithm is redesigned by modifying the *predictor* and the *updater* to bring down the critical path delay, and to reduce the latency. Two modified lifting schemes are discussed in the following section.

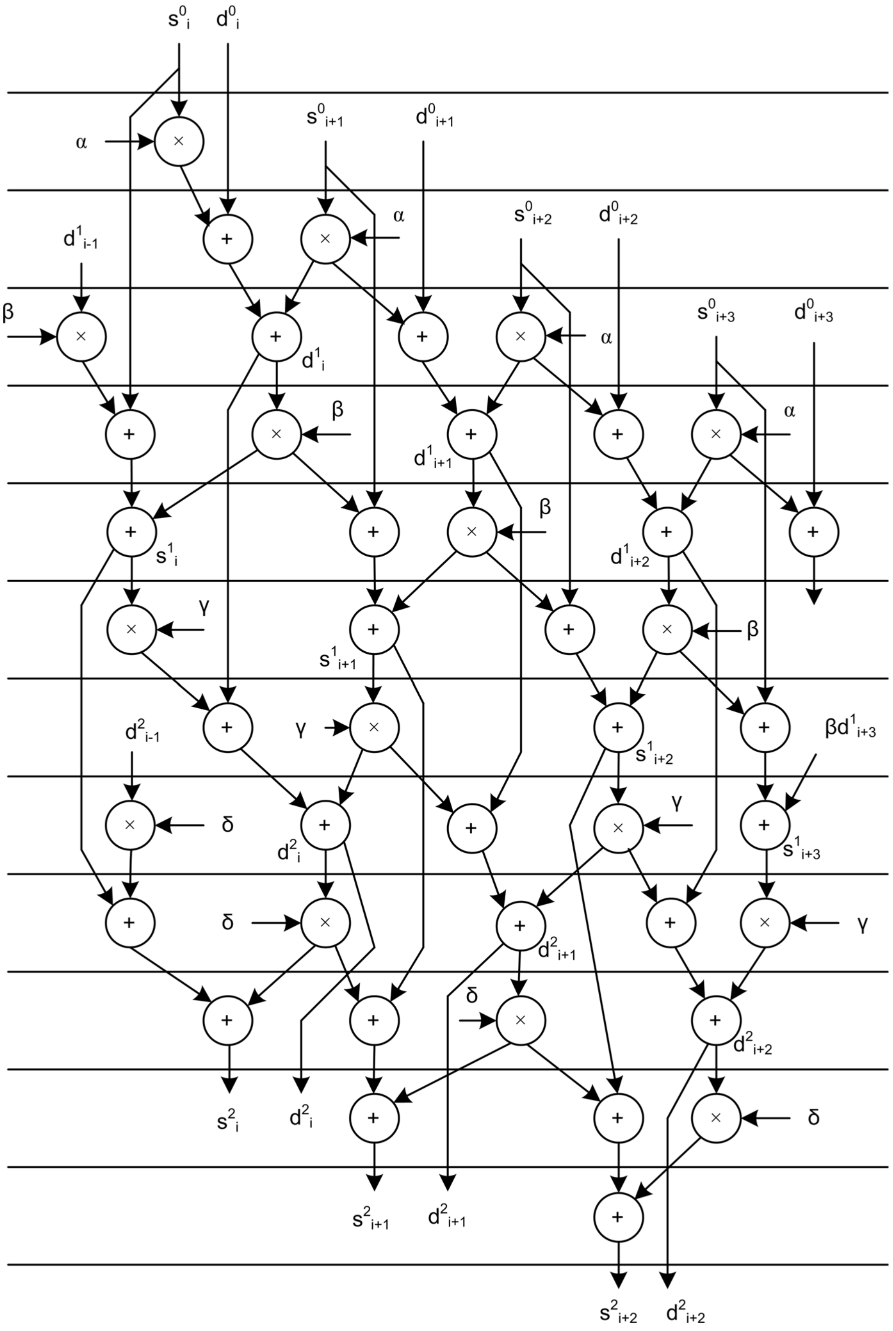

Fig. 2 Data flow graph of the conventional lifting scheme

3.2.1 Scheme-1

In order to minimize the critical path delay in the lifting based DWT structure, the predict function (Eq. 11) of the first lifting step is modified, and merged with the update function (Eq. 12) to obtain a modified equation of the low-pass coefficient at the first lifting step. In the second lifting step, both the predict and the update functions are modified and merged. Later, the coefficients are scaled with the appropriate scaling function to obtain the high-pass and the low-pass coefficients.

1. First lifting step

$$d_i^1 = d_i^0 + \alpha\left(s_i^0 + s_{i+1}^0\right) \tag{11}$$

$$s_i^1 = s_i^0 + \beta\left(d_{i-1}^1 + d_i^1\right) \tag{12}$$

Modifying Eq. 11, and substituting for βd_i^1 and βd_{i-1}^1 in Eq. 12

$$s_i^1 = s_i^0 + \left(\beta d_{i-1}^0 + \alpha\beta\left(s_{i-1}^0 + s_i^0\right)\right) + \left(\beta d_i^0 + \alpha\beta\left(s_i^0 + s_{i+1}^0\right)\right) \tag{13}$$

Similarly βd_{i+1}^1 and s_{i+1}^1 can be given as

$$\beta d_{i+1}^1 = \beta d_{i+1}^0 + \alpha\beta\left(s_{i+1}^0 + s_{i+2}^0\right) \tag{14}$$

$$s_{i+1}^1 = s_{i+1}^0 + \left(\beta d_i^0 + \alpha\beta\left(s_i^0 + s_{i+1}^0\right)\right) + \left(\beta d_{i+1}^0 + \alpha\beta\left(s_{i+1}^0 + s_{i+2}^0\right)\right) \tag{15}$$

2. Second lifting step

$$d_i^2 = d_i^1 + \gamma\left(s_i^1 + s_{i+1}^1\right) \tag{16}$$

$$s_i^2 = s_i^1 + \delta\left(d_{i-1}^2 + d_i^2\right) \tag{17}$$

Modifying Eq. 17 and replacing d_{i-1}^2 and d_i^2 by Eq. 16

$$s_i^2/\delta\gamma = s_i^1/\delta\gamma + \left(d_{i-1}^1/\gamma + s_{i-1}^1 + s_i^1\right) + \left(d_i^1/\gamma + s_i^1 + s_{i+1}^1\right) \tag{18}$$

Similarly d_{i+1}^2/γ and $s_{i+1}^2/\delta\gamma$ can be given as

$$d_{i+1}^2/\gamma = d_{i+1}^1/\gamma + s_{i+1}^1 + s_{i+2}^1 \tag{19}$$

$$s_{i+1}^2/\delta\gamma = s_{i+1}^1/\delta\gamma + \left(d_i^1/\gamma + s_i^1 + s_{i+1}^1\right) + \left(d_{i+1}^1/\gamma + s_{i+1}^1 + s_{i+2}^1\right) \tag{20}$$

3. Scaling step

$$d_i = \gamma/k \times d_i^2/\gamma \tag{21}$$

$$s_i = k\delta\gamma \times s_i^2/\delta\gamma \tag{22}$$

The data flow graph of the modified lifting scheme for the 9/7 filter is shown in Fig. 3. Two input samples are fetched in each cycle for processing. The filter coefficients in this method are β, $\alpha\beta$, $1/\beta\gamma$ and $1/\delta\gamma$. The outputs of the first lifting step are βd_i^1 and s_i^1. Due to the modification in the lifting step, one cycle is saved, each in the first and the second lifting step. The maximum number of cycles required to produce one level 2-D DWT in this method are nine. The output coefficients after the second lifting step are d_i^2/γ and $s_i^2/\delta\gamma$. The coefficients are later scaled by a factor of γ/k and $k\delta\gamma$ to obtain the high-pass and low-pass wavelet coefficients respectively.

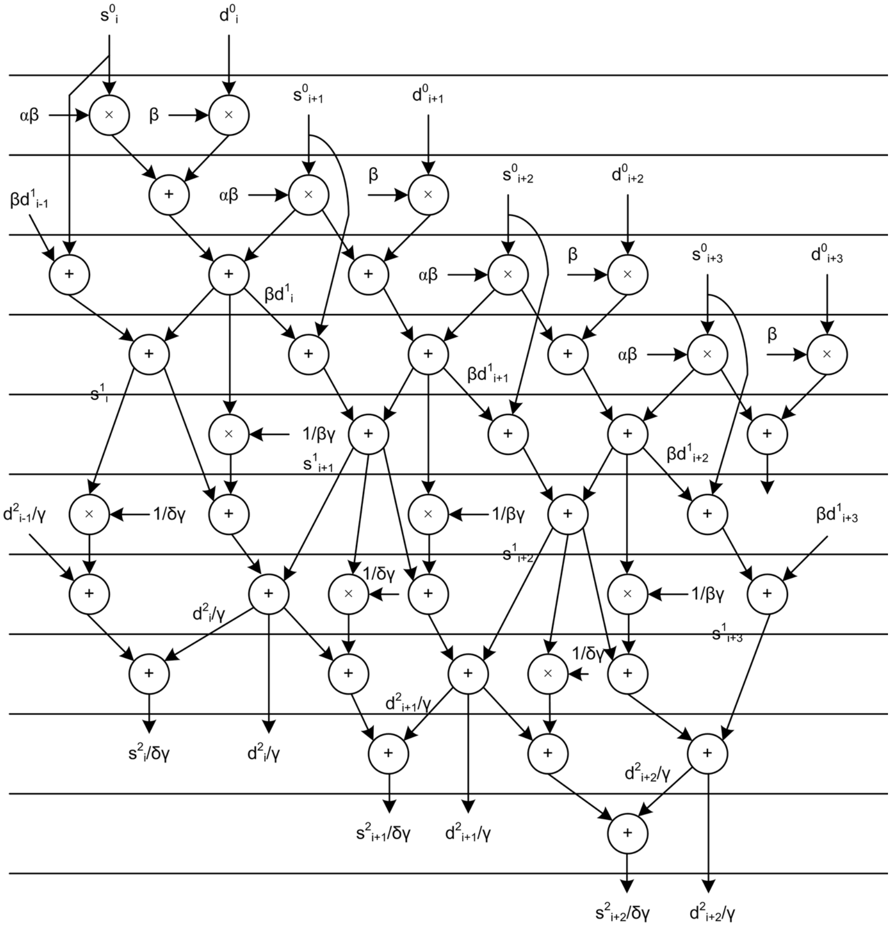

Fig. 3 Data flow graph of the modified lifting scheme with two-input/two-output

3.2.2 Scheme-2

In this method, at the first lifting step, the predict function (Eq. 23) is merged with the modified update function (Eq. 24) to obtain an equation for the low-pass coefficient. In the second lifting step, the update function (Eq. 29) is modified, and the predict function (Eq. 28) is merged with it to obtain a modified low-pass coefficient equation. Later, the high-pass and the low-pass coefficients are scaled appropriately.

1. First lifting step

$$d_i^1 = d_i^0 + \alpha\left(s_i^0 + s_{i+1}^0\right) \tag{23}$$

$$s_i^1 = s_i^0 + \beta\left(d_{i-1}^1 + d_i^1\right) \tag{24}$$

Modifying Eq. 24, and substituting d_{i-1}^1 and d_i^1 by Eq. 23

$$s_i^1/\beta = s_i^0/\beta + \left(d_{i-1}^0 + \alpha\left(s_{i-1}^0 + s_i^0\right)\right) + \left(d_i^0 + \alpha\left(s_i^0 + s_{i+1}^0\right)\right) \tag{25}$$

Similarly d_{i+1}^1 and s_{i+1}^1 can be given as

$$d_{i+1}^1 = d_{i+1}^0 + \alpha\left(s_{i+1}^0 + s_{i+2}^0\right) \tag{26}$$

$$s_{i+1}^1/\beta = s_{i+1}^0/\beta + \left(d_i^0 + \alpha\left(s_i^0 + s_{i+1}^0\right)\right) + \left(d_{i+1}^0 + \alpha\left(s_{i+1}^0 + s_{i+2}^0\right)\right) \tag{27}$$

2. Second lifting step

$$d_i^2 = d_i^1 + \gamma\left(s_i^1 + s_{i+1}^1\right) \tag{28}$$

$$s_i^2 = s_i^1 + \delta\left(d_{i-1}^2 + d_i^2\right) \tag{29}$$

Modifying Eq. 29 and substituting Eq. 28 for d_{i-1}^2 and d_i^2

$$s_i^2/\beta = s_i^1/\beta + (\delta/\beta)\left[d_{i-1}^1 + \gamma\left(s_{i-1}^1 + s_i^1\right)\right] + (\delta/\beta)\left[d_i^1 + \gamma\left(s_i^1 + s_{i+1}^1\right)\right] \tag{30}$$

Similarly d_{i+1}^2 and s_{i+1}^2 can be given as

$$d_{i+1}^2 = d_{i+1}^1 + \gamma\left(s_{i+1}^1 + s_{i+2}^1\right) \tag{31}$$

$$s_{i+1}^2/\beta = s_{i+1}^1/\beta + (\delta/\beta)\left[d_i^1 + \gamma\left(s_i^1 + s_{i+1}^1\right)\right] + (\delta/\beta)\left[d_{i+1}^1 + \gamma\left(s_{i+1}^1 + s_{i+2}^1\right)\right] \tag{32}$$

3. Scaling step

$$d_i = 1/k \times d_i^2 \tag{33}$$

$$s_i = k\beta \times s_i^2/\beta \tag{34}$$

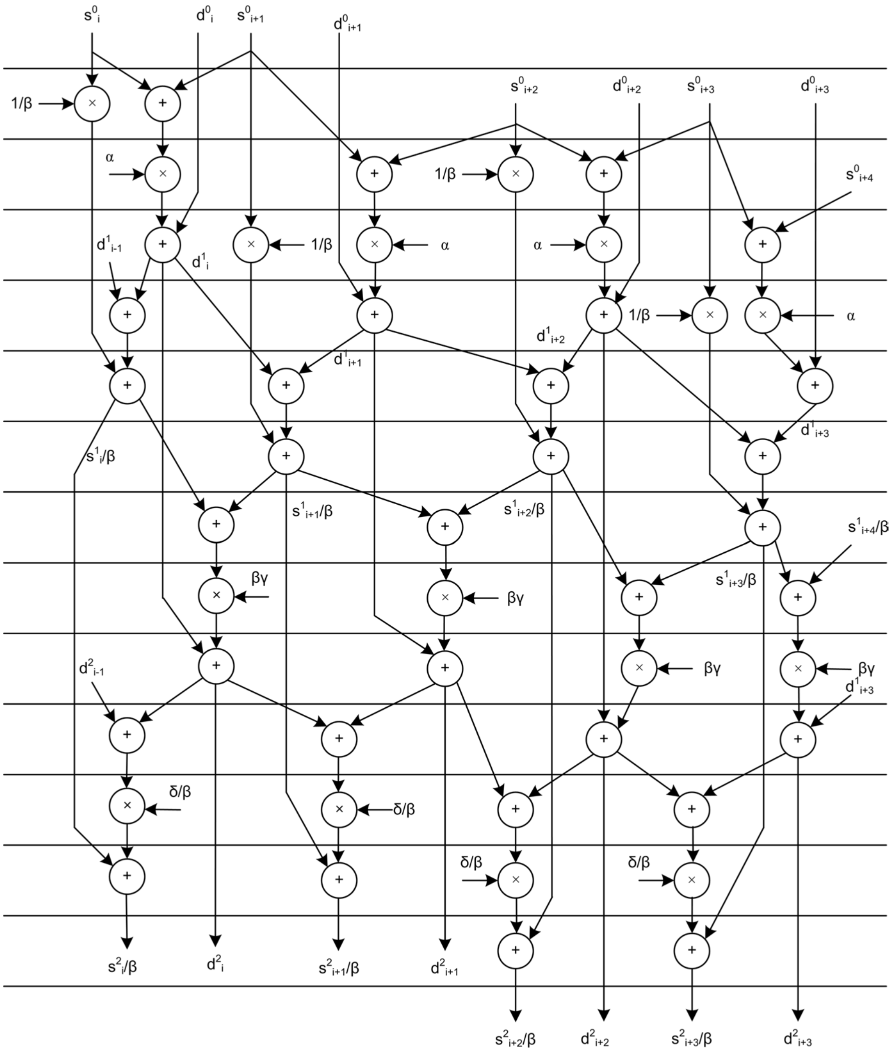

Fig. 4 Data flow graph of the modified lifting scheme with four-input/four-output

The data flow graph of the modified lifting scheme for the 9/7 filter is shown in Fig. 4. The filter coefficients in this method are α, $1/\beta$, $\beta\gamma$ and δ/β. Four input coefficients are fetched simultaneously to give two, high-pass and two, low-pass coefficients. The outputs of the first lifting step are d_i^1 and s_i^1/β. The high-pass and low-pass coefficients after the second lifting step are d_i^2 and s_i^2/β respectively. The latency to produce the output is 13 cycles delay. The high-pass and low-pass coefficients are scaled by a factor of $1/k$ and $k\beta$ respectively in the scaling step.

4 Proposed Architecture for the 2-D Lifting Based DWT

4.1 Overall Architecture

The overall block diagram of the proposed 2-D DWT architecture is shown in Fig. 5. The structure consists of row processor, column processor, transpose block and temporal memory. The row and the column processor are designed to perform 2-D transform with row-column wise scanning scheme. The intermediate data is stored in the temporal memory during column processing. For the 9/7 filter, the first and the second lifting step together constitute a row/column processor, and for the 5/3 filter, row/column processor comprises of only the first lifting step.

4.2 Data Scanning Method

Z-scanning [24] is employed to process the row and column elements simultaneously. The data scanning method for an $N \times M$ image is shown in Fig. 6. Two elements, viz., odd and even are read in unison at every rising edge of the clock. All the elements in the image are read in a Z fashion. This type of scanning method results in reducing the latency and developing a transpose block, independent of the number of elements in an image.

4.3 First Lifting Step Processing Architecture

The first lifting step processing block shown in Fig. 7 is based on the data flow graph described in Fig. 3. It consists of two input lines and two output lines with four adders and two multipliers. The structure has four pipeline stages to reduce the critical path delay to $1T_m$. Input scanning is done in a Z fashion with a throughput rate of two. The first and the second element of the odd row is read at the rising edge of the clock. Next, when the elements of the even row are read and processed, intermediate results of the odd row are stored in first-in first-out (FIFO) registers, $P3$ and $P5$. The intermediate results are later utilized when the third and the fourth element of the odd row is read. $P3$ and $P5$ registers are used to store the processed data of the odd and the even row elements. The data flow of the first lifting step is shown in Table 1, where subscript i is the element in an image. The outputs obtained at the end of the cycle are βd_i^1 and s_i^1.

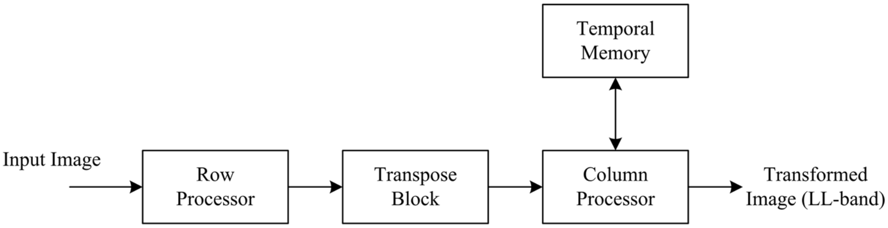

Fig. 5 Block diagram of the proposed 2-D DWT structure

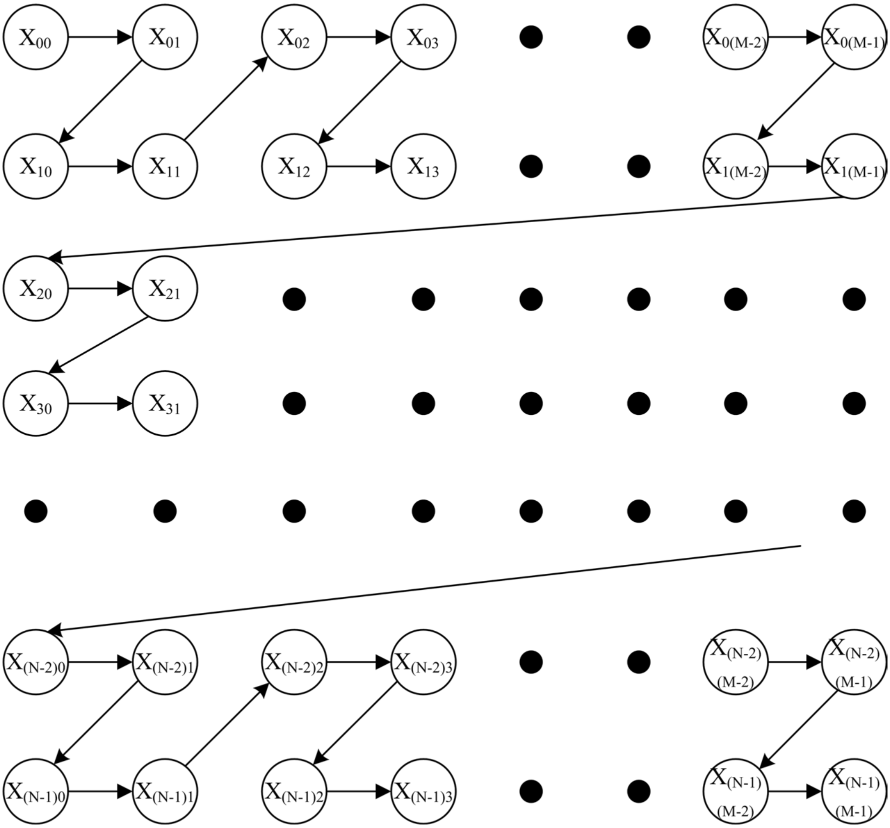

Fig. 6 Z-scan method [24]

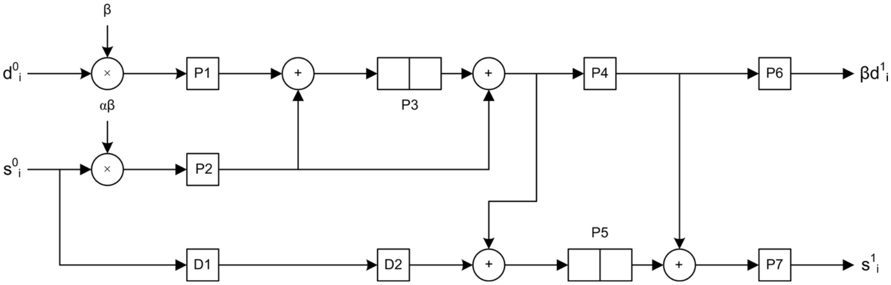

Fig. 7 Proposed first lifting step processing block

4.4 Second Lifting Step Processing Architecture

Based on the data flow graph in Fig. 3, the second lifting step processing block is constructed as shown in Fig. 8. The inputs to this block are the outputs of the

Table 1 Data flow of the first lifting step

d_n	s_n	P1	P2	P3	P4	D1	D2	P5	P6	P7
d_i^0	s_i^0									
d_{i+1}^0	s_{i+1}^0	βd_i^0	$\alpha\beta s_i^0$			s_i^0				
d_{i+2}^0	s_{i+2}^0	βd_{i+1}^0	$\alpha\beta s_{i+1}^0$	$\beta d_i^0 + \alpha\beta s_i^0$		s_{i+1}^0	s_i^0			
d_{i+3}^0	s_{i+3}^0	βd_{i+2}^0	$\alpha\beta s_{i+2}^0$	$\beta d_{i+1}^0 + \alpha\beta s_{i+1}^0$	βd_i^1	s_{i+2}^0	s_{i+1}^0	$\beta d_{i-1}^1 + s_i^0$		
d_{i+4}^0	s_{i+4}^0	βd_{i+3}^0	$\alpha\beta s_{i+3}^0$	$\beta d_{i+2}^0 + \alpha\beta s_{i+2}^0$	βd_{i+1}^1	s_{i+3}^0	s_{i+2}^0	$\beta d_i^1 + s_{i+1}^0$	βd_i^1	s_i^1
d_{i+5}^0	s_{i+5}^0	βd_{i+4}^0	$\alpha\beta s_{i+4}^0$	$\beta d_{i+3}^0 + \alpha\beta s_{i+3}^0$	βd_{i+2}^1	s_{i+4}^0	s_{i+3}^0	$\beta d_{i+1}^1 + s_{i+2}^0$	βd_{i+1}^1	s_{i+1}^1

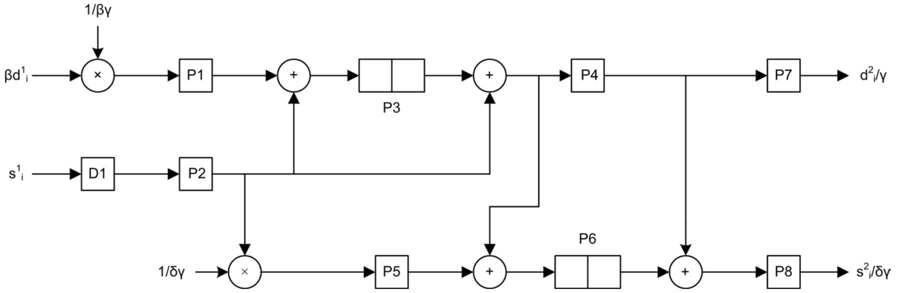

Fig. 8 Proposed second lifting step processing block

Table 2 Data flow of the second lifting step

d_n	s_n	P1	D1	P3	P4	P2	P5	P6	P7
βd_i^1	s_i^1								
βd_{i+1}^1	s_{i+1}^1	d_i^1/γ	s_i^1						
βd_{i+2}^1	s_{i+2}^1	d_{i+1}^1/γ	s_{i+1}^1	$d_i^1/\gamma + s_i^1$	d_{i-1}^2/γ	$s_i^1/\delta\gamma$			
βd_{i+3}^1	s_{i+3}^1	d_{i+2}^1/γ	s_{i+2}^1	$d_{i+1}^1/\gamma + s_{i+1}^1$	d_i^2/γ	$s_{i+1}^1/\delta\gamma$	$d_{i-1}^2/\gamma + s_i^1/\delta\gamma$		
βd_{i+4}^1	s_{i+4}^1	d_{i+3}^1/γ	s_{i+3}^1	$d_{i+2}^1/\gamma + s_{i+2}^1$	d_{i+1}^2/γ	$s_{i+2}^1/\delta\gamma$	$d_i^2/\gamma + s_{i+1}^1/\delta\gamma$	d_i^2/γ	$s_i^2/\delta\gamma$
βd_{i+5}^1	s_{i+5}^1	d_{i+4}^1/γ	s_{i+4}^1	$d_{i+3}^1/\gamma + s_{i+3}^1$	d_{i+2}^2/γ	$s_{i+3}^1/\delta\gamma$	$d_{i+1}^2/\gamma + s_{i+2}^1/\delta\gamma$	d_{i+1}^2/γ	$s_{i+1}^2/\delta\gamma$

first lifting step processing architecture. The intermediate results are stored in the FIFO registers, $P3$ and $P5$. Similar to the first lifting step, the four pipeline stages reduce the critical path delay to $1T_m$. The outputs of this block are d_i^2/γ and $s_i^2/\delta\gamma$. Table 2 shows the data flow of the second lifting step.

The first and the second lifting architectures are cascaded to form the row/column processing block. This forms a structure for 1-D DWT. In the column processing block the FIFO registers are replaced by an internal memory of size N. For the 2-D transform, the transposed output of 1-D DWT is fed again to the processing block. Since the scanning is in Z fashion, the intermediate results of

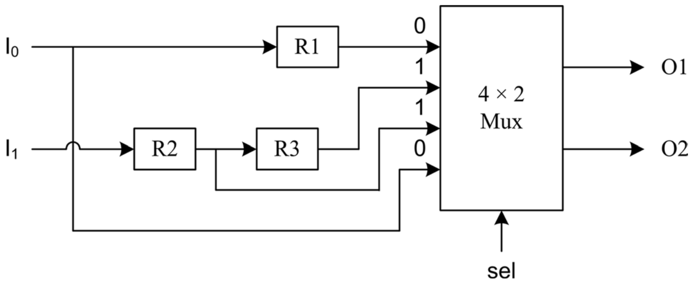

Fig. 9 Block diagram of the transpose block

Table 3 Data flow of the transpose block

I_0	I_1	R1	R2	R3	sel	O1	O2
l_{00}	h_{00}						
l_{01}	h_{01}	l_{00}	h_{00}				
l_{10}	h_{10}	l_{01}	h_{01}	h_{00}	0	l_{00}	l_{01}
l_{11}	h_{11}	l_{10}	h_{10}	h_{01}	1	h_{00}	h_{01}
l_{02}	h_{02}	l_{11}	h_{11}	h_{10}	0	l_{10}	l_{11}

the two rows are stored in the temporal memory of size 4N and 2N during column processing of the 9/7 and 5/3 filters respectively.

4.5 Transpose Block

The column processing block accepts the column-wise data managed by the transpose block. An efficient transpose block is developed to rearrange the output sequence of the row processor for the use of the column processor. The transpose block, as shown in Fig. 9, consists of three registers ($R1-R3$) and one 4-to-2 multiplexer. The input and output data flow in the transpose block is shown in Table 3. h and l are the high-pass and the low-pass coefficients after 1-D transform. The structure of the transpose block is independent of the size of the image.

4.6 Four-Input/Four-Output 1-D DWT Architecture

The 1-D DWT architecture for the 9/7 filter shown in Fig. 10 processes four elements per clock cycle. This structure is designed from the data flow graph outlined in Fig. 4. The input data flow is line-based, with four pixels being read simultaneously from the row of an image. The structure consists of sixteen adders and eight multipliers with two stage pipeline. For the 2-D DWT, the column-wise decomposition is done after the row-wise decomposition with the same structure. However, temporal buffer and transpose block are necessary to perform the 2-D transform.

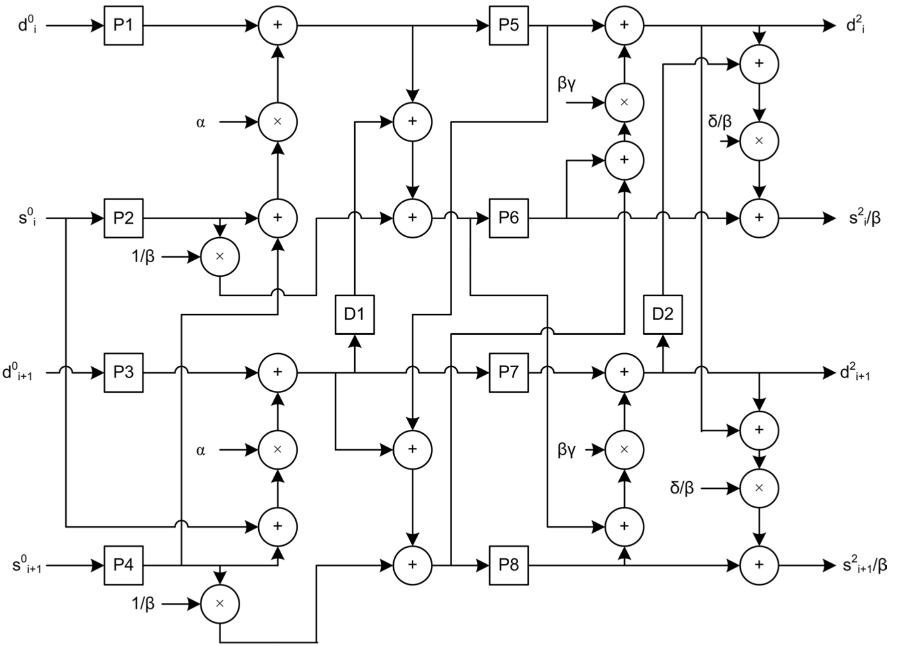

Fig. 10 Proposed four-input/four-output 1-D DWT architecture

5 Implementation and Performance Analysis

5.1 Implementation

In order to build a systematic and standard description, the proposed structures are described in structural VHDL, following uniform coding guidelines. The structures are synthesized using Cadence RTL compiler over the commercial 90 nm technology of low power library *slow_lib* for the lowest possible voltage and worst-case delay. Power consumption for the design is obtained by performing the power simulation using Cadence NCsim, simulating 256 × 256 vectors to obtain the toggle count format (TCF) file. Full hierarchy TCF dumps are performed, and the data is fed to the Cadence tool for power simulation. MATLAB is used to read the input image, and convert the transformed coefficients back to an image.

Derived netlist and design constraints are fed to the Cadence Encounter tool to generate a layout, from which RC parasitic information is extracted. The interconnect stack consists of one polysilicon layer and six metal layers. The layout is optimized and nano routed. Post route static timing analysis and hold timing analysis are done. It is observed that the worst negative slack (WNS) is positive and the total negative slack (TNS) is zero.

The implementation summary of the proposed single-level 2-D DWT architecture with two-input/two-output is given in Table 4. For an image size of 256 × 256, the power dissipation and delay of the structure for the 5/3 filter is 8.45 mW and 4.26 ns respectively. Similarly, the power and delay parameters for the 9/7 filter design is

Table 4 Design specification of the proposed structure

Specification	5/3 Filter	9/7 Filter
Technology	90 nm CMOS	90 nm CMOS
Die area (mm^2)	0.105	0.273
Core area (mm^2)	0.092	0.251
Gate count	8282	22,575
Power supply (V)	0.9	0.9
Clock frequency (MHz)	100	100

35.93 mW and 7.64 ns respectively. Further, four 256 × 16 two-port RAM and two 256 × 16 two-port RAM are employed for 9/7 and 5/3 filter respectively to store the intermediate results from the column processor.

In the hardware implementation, the floating point number computation needs larger chip area and more computation time. Hence, these numbers are converted into fixed point representation [15]. The 9/7 filter coefficients are irrational numbers; they are converted into a fixed-point representation of 12-bits to resolve the overflow issues [13]. The 16-bit data path is allocated to 1 sign bit, 10 integer bits and 5 fractional bits to preserve the image quality. Although the truncation compromises the quality of the image, its effect is negligible.

5.2 Performance Analysis

The comparison of the proposed DWT structure with other architectures for 9/7 and 5/3 filter are presented in Tables 5, 6 and 7. The performance evaluation is done in terms of hardware requirement, memory, critical path delay and throughput. The proposed two-input/two-output 1-D 9/7 DWT architecture hardware requirement is 4 multipliers and 8 adders. The memory requirement of the structure is 22 registers. It has a critical path delay of $1T_m$ with simple control complexity in contrast to the architecture proposed by Lai et al. [13] and Zhang et al. [21]. The effective folded architecture [12] reduces the hardware cost and memory requirement by re-utilizing the hardware in the structure. However, the critical path and throughput limits its application. It can be observed that the proposed structure is better than the direct structure [8] and the flipping structure [14] in terms of critical path delay. High-performance structure [15] has minimum adder and multiplier requirement due to the merging of the updater and the predictor steps, but this benefit is at the cost of throughput. The pipeline structure [19] and multi-input/multi-output structure [20] have reasonable hardware and memory requirement, but the critical path delay is high compared to the proposed structure. 1-D DWT architecture [28] with single processing element (PE) utilizes minimum hardware resource, but has a high critical path delay, which is the sum of fused multiply-add unit delay and floating point adder delay. The PE's are connected in parallel for higher throughput rate. The proposed four-input/four-output 1-D 9/7 DWT architecture based on modified lifting scheme can be used for high speed application. It has a critical path delay of $T_m + 2T_a$ and hardware requirement of 8 multipliers and 16 adders. The quad-input/

Table 5 Comparison of 1-D DWT architectures for 9/7 filter (Thrp: throughput, T_m: multiplier delay, T_a: adder delay, T_{fma}: fused multiply-add unit delay, T_{fadd}: floating point adder delay, $N \times N$: image size)

Architecture	Multiplier	Adder	Register	Critical path	Thrp
Jou et al. [8]	4	8	6	$4T_m + 8T_a$	2
Lian et al. [9]	2	4	10	$T_m + 2T_a$	2
Flipping + no pipeline [14]	4	8	4	$T_m + 5T_a$	2
Flipping + 5-stage pipeline [14]	4	8	11	T_m	2
Wu and Lin [15]	2	4	20	T_m	1
Shi et al. [12]	2	4	10	$T_m + T_a$	1
Lai et al. [13]	4	8	22	T_m	2
Lin et al. [19]	4	8	10	$T_m + 2T_a$	2
Cao et al. [18]	2	4	$2N + 5$	T_m	1
Cao et al. [18]	8	16	28	T_m	4
Tian et al. [20]	4	8	5	$T_m + 2T_a$	2
Zhang et al. [21]	4	8	18	T_m	2
Darji et al. [25]	4	8	20	T_m	2
Basiri et al. [28]	2	1	8	$T_{fma}+T_{fadd}$	1
Proposed structure (2-input/2-output)	4	8	21	T_m	2
Proposed structure (4-input/4-output)	8	16	10	$T_m + 2T_a$	4

Table 6 Comparison of 2-D DWT architectures for 9/7 filter (Thrp: throughput, T_m: multiplier delay, T_a: adder delay, T_{fma}: fused multiply-add unit delay, T_{fadd}: floating point adder delay, $N \times N$: image size)

Architecture	Multiplier	Adder	Transpose buffer	Temporal buffer	Critical path	Thrp
Flipping + no pipeline [14]	10	16	$1.5N$	$4N$	$T_m + 5T_a$	2
Flipping + 5-stage pipeline [14]	10	16	$1.5N$	$11N$	T_m	2
Liao et al. [38]	12	16	4	$4N$	$4T_m + 8T_a$	2
Huang et al. [36]	10	16	$1.5N$	$11N$	$4T_m + 8T_a$	2
Wu and Lin [15]	6	8	$1.5N$	$4N$	T_m	1
Xiong et al. [16]	10	16	$1.5N$	$4N$	T_m	2
Fast architecture [17]	10	16	–	$5.5N$	$T_m + 2T_a$	2
Lai et al. [13]	10	16	4	$4N$	T_m	2
Lin et al. [19]	10	16	$1.5N$	$4N$	$T_m + 2T_a$	2
Cao et al. [18]	12	16	–	$4N + 32$	T_m	2
Hsia et al. [37]	12	16	–	$4N$	$2T_m + 4T_a$	2
Tian et al. [20]	8	16	–	$5N + 8$	$T_m + 2T_a$	2
Zhang et al. [21]	10	16	3	$4N$	T_m	2
Hsia et al. [22]	–	16	–	$4N$	$2T_m + 4T_a$	2
Darji et al. [25]	10	16	5	$4N$	T_m	2
Darji and Limaye [39]	14	24	12	$2N$	T_m	2
Ang et al. [27]	4	8	$2N$	$4N$	T_a	2
Proposed structure	10	16	3	$4N$	T_m	2

Table 7 Comparison of 2-D DWT architectures for 5/3 filter (Thrp: throughput, T_m: multiplier delay, T_a: adder delay, $N \times N$: image size)

Architecture	Multiplier	Adder	Transpose buffer	Temporal buffer	Critical path	Thrp
Diou et al. [40]	6	12	$2N$	$1.5N$	$T_m + T_a$	2
Andra et al. [3]	4	8	$4N$	5	$T_m + 2T_a$	2
Chen and Wu [41]	4	6	–	$2.5N$	$T_m + T_a$	1
Lan et al. [42]	4	8	–	$3N$	T_m	2
Wu and Lin [15]	2	4	$1.5N$	$2N$	T_m	1
Hsia et al. [37]	12	16	–	$2N$	$2T_m + 4T_a$	2
Hsia et al. [22]	–	8	–	$2N$	$2T_m + 4T_a$	2
Proposed structure	4	8	3	$2N$	T_m	2

quad-output structure [18] needs the same amount of adders and multipliers as that of the proposed structure, but the memory requirement is high.

As for the 2-D DWT, the proposed structure for 9/7 filter requires 16 adders, 8 multipliers for processing and 2 multipliers for scaling. The critical path delay of the structure is $1T_m$ with a throughput rate of two. The computation time for $N \times N$ image is $N^2/2$. The temporal memory needed is $4N$ and the transpose buffer required is 3 registers. Compared with other structures [14, 19, 36, 37], the proposed structure has the least hardware cost and memory. The critical path delay defined in flipping structure [14] with 5-stage pipeline is same as the proposed structure, but the transpose buffer and temporal buffer requirement is in excess. Dual-scan architecture [38] reduces the transpose buffer to 4 registers, but the critical path delay limits its application. Even though the high-performance structure [15] is hardware cost effective, it is evident that the proposed 2-D structure has better throughput. The parallel based lifting scheme structure [16] for DWT aims at reducing the critical path to $1T_m$. However, the transposing buffer is $1.5N$. The FA [17] reduces the hardware requirement by utilizing the same hardware for predict and update steps. Because of the hardware reuse, pipelining cannot be done, and the critical path delay surges to $T_m + 2T_a$. The memory-efficient structure [18] requires 32 register in excess along with $4N$ temporal memory to bring down the critical path delay to $1T_m$. The two-input/two-output module [20] has a high memory requirement, and the critical path delay is $T_m + 2T_a$. Although the hardware and memory utilized in [21] are same as that of the proposed structure, the control procedure is complex. Even though the 2-D dual mode architecture [22] is hardware efficient, the critical path delay of $2T_m + 4T_a$ and computing time of $(3/4)N^2 + (3/2)N + 7$ does not make it a fast and effective structure for real-time application. Shifters are used instead of multipliers in the design. The structure consists of multiply accumulate unit (MAC), multiplexers and de-multiplexers in the 1-D DWT structure. The proposed design reduces the transpose buffer to 3 registers as compared to the dual-scan parallel flipping structure [25], where, 5 registers are used to rearrange the output from the row processor in an order required for the column processor. Darji and Limaye [39] proposed a memory efficient structure requiring $2N$ temporal memory, but the hardware resources essential for processing is high. The arithmetic resources required in the

2-D DWT architecture [27] is minimum when compared with all the other architectures, but the computation time required for the two DWT engines in parallel is $3N^2/32 + 3N^2/8$, which is more than other structures. N-parallel DWT architecture [28] uses single PE for the transform. For 9-pin 2-D transform, nine architectures are connected in parallel, containing 9 PEs and 18 FMAs. The transpose memory is not needed in this structure. However, the critical path delay is $T_{fma} + T_{fadd}$.

A trade-off exits between the critical path delay and the complexity in the design of the 2-D DWT architecture. The proposed 2-D 5/3 filter DWT structure utilizes 4 multipliers and 8 adders. The temporal memory required is $2N$ and the computation time is $N^2/2$. The critical path delay is limited to $1T_m$ with simple control complexity. Diou et al. [40] presented a 2-D DWT structure for 5/3 filter by interleaving technique. Compared to the proposed structure, arithmetic resources required are high, and the critical path delay is the sum of T_m and T_a. Chen and Wu [41] presented a folded structure for the 2-D 5/3 filter DWT. The throughput rate is one-input/one-output, and the temporal memory required to store the intermediate data is $2.5N$ for $N \times N$ image. The high-speed VLSI architecture [42] requires 4 multipliers and 8 adders for computation. The critical path delay is minimum but the temporal memory required is $3N$.

6 Conclusion

In this paper, an efficient 1-D DWT architecture of two-input/two-output and four-input/four-output is proposed based on the modified lifting scheme. A 2-D DWT structure for the 9/7 and 5/3 filter is also proposed with a throughput rate of two per cycle. The DWT architecture is energy efficient, and has a low critical path delay of $1T_m$. The structure has shown desired quality performance in terms of hardware usage and memory. Based on the comparison analysis, the proposed structure can achieve a good speed with low hardware cost, which will be an efficient alternative for the high speed application. The proposed one-level 2-D DWT can be easily extended to multilevel by storing the LL-band in an external memory for the next level transformation. From the ASIC synthesis result, it can be observed that the proposed structure is area and power efficient, which is suitable for real-time image and video processing.

References

1. Acharya, T., & Chakrabarti, C. (2006). A survey on lifting-based discrete wavelet transform architectures. *Journal of VLSI Signal Processing*, *42*, 321–339.
2. Daubechies, I., & Sweldens, W. (1998). Factoring wavelet transforms into lifting schemes. *Journal of Fourier Analysis and Applications*, *4*(3), 247–269.
3. Andra, K., Chakrabarti, C., & Acharya, T. (2002). A VLSI architecture for lifting-based forward and inverse wavelet transform. *IEEE Transactions on Signal Processing*, *50*(4), 247–269.
4. Parhi, K. K., & Nishitani, T. (1993). VLSI architectures for discrete wavelet transforms. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *1*(2), 191–202.
5. Wu, P. C., & Chen, L. G. (2001). An efficient architecture for two-dimensional discrete wavelet transform. *IEEE Transactions on Circuits and Systems for Video Technology*, *11*(4), 536–545.

6. Cheng, C., & Parhi, K. K. (2008). High-speed VLSI implementation of 2-D discrete wavelet transform. *IEEE Transactions on Signal Processing*, *56*(1), 393–403.
7. Zhang, X., & Jindapetch, N. (2010). An efficient VLSI architecture for discrete wavelet transform based on the Daubechies architecture. In *International conference on electrical engineering/electronics computer telecommunications and information technology (ECTI-CON)*, Thailand.
8. Jou, J. M., Shiau, Y. H., & Liu, C. C. (2001). Efficient VLSI architectures for the bi-orthogonal wavelet transform by filter bank and lifting scheme. *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, *2*, 529–532.
9. Lian, C.-J., Chen, K.-F., Chen, H.-H., & Chen, L.-G. (2001). Lifting based discrete wavelet transform architecture for JPEG2000. *Proceedings of IEEE International Symposium on Circuits and Systems*, *2*, 445–445.
10. Chen, P.-Y. (2004). VLSI implementation for one-dimensional multilevel lifting-based wavelet transform. *IEEE Transactions on Computers*, *53*(4), 386–398.
11. Barua, S., Carletta, J. E., Kotteri, K. A., & Bell, A. E. (2005). An efficient architecture for lifting-based two-dimensional discrete wavelet transforms. *Integration, the VLSI Journal*, *38*, 341–352.
12. Shi, G., Liu, W., Zhang, L., & Li, F. (2009). An efficient folded architecture for lifting-based discrete wavelet transform. *IEEE Transactions on Circuits and System II: Express Briefs*, *56*(4), 290–294.
13. Lai, Y.-K., Chen, L.-F., & Shih, Y.-C. (2009). A high-performance and memory: Efficient VLSI architecture with parallel scanning method for 2-D lifting- based discrete wavelet transform. *IEEE Transactions on Consumer Electronics*, *55*(2), 400–407.
14. Huang, C. T., Tseng, P. C., & Chen, L. G. (2004). Flipping structure: An efficient VLSI architecture for lifting-based discrete wavelet transform. *IEEE Transactions on Signal Processing*, *52*(4), 1080–1089.
15. Wu, B., & Lin, C. (2005). A high performance and memory efficient pipeline architecture for 5/3 and 9/7 discrete wavelet transform of JPEG2000 codec. *IEEE Transactions on Circuits and Systems for Video Technology*, *15*(12), 1615–1628.
16. Xiong, C.-Y., Tian, J.-W., & Liu, J. (2006). A note on flipping structure: An efficient VLSI architecture for lifting-based discrete wavelet transform. *IEEE Transactions on Signal Processing*, *54*(5), 1910–1916.
17. Xiong, C., Tian, J., & Liu, J. (2007). Efficient architectures for two-dimensional discrete wavelet transform using lifting scheme. *IEEE Transactions on Image Processing*, *163*, 607–614.
18. Cao, P., Wang, C., & Shi, L. X. (2010). Memory-efficient and high-speed VLSI implementation of two-dimensional discrete wavelet transform using decomposed lifting scheme. *Journal of Signal Processing System*, *61*, 219–230.
19. Lin, C. F., Huang, P. K., & Wu, B. F. (2010). An efficient pipeline architecture and memory bitwidth analysis for discrete wavelet transform of the 9/7 filter for JPEG 2000. *Journal of Signal Processing System*, *59*, 245–253.
20. Tian, X., Wu, L., Tan, Y. H., & Tian, J. W. (2011). Efficient multi-input/multi-output VLSI architecture for two-dimensional lifting-based discrete wavelet transform. *IEEE Transactions on Computers*, *60*(8), 1207–1211.
21. Zhang, W., Jiang, Z., Gao, Z., & Liu, Y. (2012). An efficient VLSI architecture for lifting-based discrete wavelet transform. *IEEE Transactions on Circuits and Systems—II: Express Briefs*, *59*(3), 158–162.
22. Hsia, C.-H., Chiang, J.-S., & Guo, J.-M. (2013). Memory-efficient hardware architecture of 2-D dual-mode lifting-based discrete wavelet transform. *IEEE Transactions on Circuits and Systems for Video Technology*, *25*(4), 671–683.
23. Bhat N.S. (2013) Implementation of lifting scheme based DWT architecture on FPGA. In M. Aswatha Kumar et al. (Eds.), *Proceedings of international conference on advances in computing* (pp. 361–369). Springer, India.
24. Darji, A. D., Kushwah, S. S., Merchant, S. N., & Chandorkar, A. N. (2014). High-performance hardware architectures for multi-level lifting-based discrete wavelet transform. *EURASIP Journal on Image and Video Processing*, *2014*, 47.
25. Darji, A., Agrawal, S., Oza, A., Sinha, V., Verma, A., Merchant, S. N., et al. (2014). Dual-scan parallel flipping architecture for a lifting-based 2-D discrete wavelet transform. *IEEE Transactions on Circuits and Systems—II: Express Briefs*, *61*(6), 433–437.
26. Aziz, S. M., & Pham, D. M. (2012). Efficient parallel architecture for multi-level forward discrete wavelet transform processors. *Computers and Electrical Engineering*, *38*, 1325–1335.

27. Ang, B. H., Sheikh, U. U., & Marsono, M. N. (2015). 2-D DWT system architecture for image compression. *Journal of Signal Processing System*, *78*, 131–137.
28. Basiri, M. A., & Noor, M. Sk. (2016). An efficient VLSI architecture for lifting based 1D/2D discrete wavelet transform. *Microprocessors and Microsystems*, *47*, 404–418.
29. Wang, H., & Choy, C. S. (2016). Systolic array based VLSI architecture for high throughput 2-D discrete wavelet transform. In *Proceedings of IEEE international conference on electron devices and solid-state circuits (EDSSC)*, Hong Kong, China.
30. Al-Azawi, S. (2017). Low-power, low-area multi-level 2-D discrete wavelet transform architecture. *Circuits System Signal Processing*, *37*(1), 444–458.
31. Mohanty, B. K., & Meher, P. K. (2011). Memory efficient modular VLSI architecture for high-throughput and low-latency implementation of multilevel lifting 2-D DWT. *IEEE Transactions on Signal Processing*, *59*(5), 2072–2084.
32. Hu, Y., & Jong, C. C. (2013). A memory-efficient high-throughput architecture for lifting-based multi-level 2-D DWT. *IEEE Transactions on Signal Processing*, *61*(20), 4975–4987.
33. Zhang, Y., Cao, H., Jiang, H., & Li, B. (2016). Memory-efficient high-speed VLSI implementation of multi-level discrete wavelet transform. *Journal of Visual Communication and Image Representation*, *38*, 297–306.
34. Nath, P. K., & Banerjee, S. (2016). A high speed, memory efficient line based VLSI architecture for the dual mode inverse discrete wavelet transform of JPEG2000 decoder. *Microprocessors and Microsystems*, *40*, 181–188.
35. Savić, G., Prokin, M., Rajović, V., & Prokin, D. (2017). High-performance 1-D and 2-D inverse DWT 5/3 filter architectures for efficient hardware implementation. *Circuits, Systems and Signal Processing*, *36*, 3674–3701.
36. Huang, C.-T., Tseng, P., & -C, and Chen L.-G., (2005). Generic RAM-based architectures for two-dimensional discrete wavelet transform with line-based method. *IEEE Transactions on Circuits and Systems for Video Technology*, *15*(7), 910–920.
37. Hsia, C. H., Li, W. M., & Chiang, J. S. (2011). Memory-efficient architecture of 2-D lifting-based discrete wavelet transform. *Journal of the Chinese Institute of Engineers*, *34*(5), 629–643.
38. Liao, H., Mandal, M. K., & Cockburn, B. F. (2004). Efficient architectures for 1-D and 2-D lifting-based wavelet transforms. *IEEE Transactions on Signal Processing*, *52*(5), 1315–1326.
39. Darji, A. D. and Limaye, A., (2014). Memory efficient VLSI architecture for lifting-based DWT. In *Proceedings of 57th IEEE international midwest symposium on circuits and systems (MWSCAS)*, TX, USA.
40. Diou, C., Torres, L., & Robert, M. (2001). An embedded core for the 2-D wavelet transform. In *Proceedings of 8th IEEE international conference on emerging technologies and factory automation* (vol. 2, pp. 179–186), Antibes-Juan les Pins, France.
41. Chen, S.-C., & Wu, C.-C. (2002). An architecture of 2-D 3-level lifting-based discrete wavelet transform. In: *Proceedings of VLSI design CAD symposium* (pp. 351–354).
42. Lan, X., Zheng, N., & Liu, Y. (2005). Low-power and high-speed VLSI architecture for lifting-based forward and inverse wavelet transform. *IEEE Transactions on Consumer Electronics*, *51*(2), 379–385.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.