International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878
Volume 1 Issue 1, September 2013

An Efficient VLSI Architecture for Lifting Based Discrete Wavelet Transform

Sasirekha.K[1], P. M. Francis[2], B. Prasad Kumar[3]

[1]M. Tech, GITAS College, Bibbili, A.P, India

[2]Department of ECE, Head of Department, Assistant Professor, GITAS College, Bobbili, A.P, India

[3]Department of ECE, Assistant Professor, GITAS College, Bobbile, A.P, India

Abstract: ***High-speed and reduced-area 2-D discrete wavelet transform (2-D DWT) architecture is proposed. Previous DWT architectures are mostly based on the modified lifting scheme or the flipping structure. In order to achieve a critical path with only one multiplier, at least four pipelining stages are required for one lifting step, or a large temporal buffer is needed. In this brief, modifications are made to the lifting scheme, and the intermediate results are recombined and stored to reduce the number of pipelining stages. As a result, the number of registers can be reduced to 18 without extending the critical path. In addition, the two-input/two-output parallel scanning architecture is adopted in our design. For a 2-D DWT with the size of N × N, the proposed architecture only requires three registers between the row and column filters as the transposing buffer, and a higher efficiency can be achieved.***

Keywords: Images, DWT, lifting scheme

1. Introduction

The discreet wavelet transform (DWT) is a multire solution analysis tool with excellent characteristics in the time and frequency domains. Through the DWT, signals can be decomposed into different subbands with both time and frequency information [2] [3]. The coding efficiency and the quality of image restoration with the DWT are higher than those with the traditional discrete cosine transform. Moreover, it is easy to obtain a high compression ratio. As a result, the DWT is widely used in signal processing and image compression, such as MPEG-4, JPEG2000, and so on. Traditional DWT architectures [7] [8] are based on convolutions. Then, the second-generation DWTs, which are based on lifting algorithms, are proposed.

Compared with convolution-based ones, lifting-based architectures not only have lower computation complexity but also require less memory. Nevertheless, directly mapping these algorithms to hardware leads to relatively long data path and low efficiency. Recently, several novel architectures based on the lifting scheme have been proposed. Shi *et al.* achieved an efficient folded architecture (EFA) with low hardware complexity. However, its critical path delay is $Tm + Ta$, where Tm and Ta are the delay of a multiplier and an adder, respectively, and the computation time of EFA is quite long. Through optimizing the lifting scheme, Wu and Lin proposed a pipelined architecture to reduce the critical path to one multiplier and limit the size of the temporal buffer to $4N$, but it has one input and one output and cannot achieve high processing speed. Based on Wu and Lin's design, Lai *et al.* [10] implemented the parallel 2-D DWT. The design is a pipelined two-input/two output architecture, and a 2 × 2 transposing module with four registers was developed. In addition, the critical path delay is one Tm. Nevertheless, it needs eight pipelining stages to complete the 1-D DWT and makes the total number of registers reach.

The flipping structure is another important DWT architecture that was proposed by Huang *et al.* [11]. With a five-stage pipeline, the critical path can be also reduced to one multiplier. However, the flipping structure has a large temporal buffer, and fewer pipelining stages lead to longer critical path delay. In this brief, further optimization on the lifting scheme is proposed to overcome shortages in previous works and minimize sizes of the logic units and the memory without loss of the throughput. By recombining the intermediate results of the row and column transforms, the number of pipelining stages and registers is reduced, while keeping the critical path delay as Tm. In addition, a novel architecture is developed to implement the 2-D DWT based on the above modified scheme. The parallel scanning method is employed to reduce the size of the transposing buffer. As a result, our design achieves higher efficiency. The rest of this brief is organized as follows. Section II reviews the lifting scheme and the flipping structure of the DWT, and then, we proposed our modified algorithm for DWT. Section III presents the proposed architecture for the 2-D DWT, and Section IV provides implementation results and comparisons with previous architectures. Conclusion is drawn in Section V.

2. Proposed Algorithm

The lifting scheme was first proposed by Daubechies and Sweldens in 1996 [5], [6]. It shows that every finite-impulse response wavelet or filter bank can be factored into a cascade of lifting steps. That means the polyphase matrices for the wavelet filters can be decomposed into a sequence of alternating upper and lower triangular matrices multiplied by a diagonal normalization matrix. The whole lifting scheme of the 9/7 filter has two lifting steps and one scaling step. In order to optimize the critical path of the lifting-based hardware implementation, a modified algorithm is employed

International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878
Volume 1 Issue 1, September 2013

by changing the coefficients in lifting formulas [11], shown as follows:

$$\frac{1}{\alpha}y(2n+1) = \frac{1}{\alpha x(2n+1)}[x(2n)+x(2n+2)] \quad (1)$$

$$\frac{1}{\beta}y(2n) = \frac{1}{\beta}x(2n) + y(2n-1) + y(2n+1) \quad (2)$$

$$\frac{1}{\gamma}H(2n+1) = \frac{1}{\gamma}y(2n+1) + y(2n) + y(2n+1) \quad (3)$$

$$\frac{1}{\delta}L(2n) = \frac{1}{\delta}(2n) + H(2n-1) + H(2n+1) \quad (4)$$

Based on (1)–(4), the flipping structure can achieve one multiplier delay by pipelining. However, the above flipping based algorithm also shows obvious limitations. It needs the temporal buffer with the size of 11*N* to cache the intermediate data. Substituting (1) into (2) and reordering the expression with the associative law, we can get

$$\begin{aligned}\frac{1}{\alpha\beta}y(2n) &= \frac{1}{\alpha\beta}x(2n) + \frac{1}{\alpha}y(2n-1) + \frac{1}{\alpha}y(2n+1) \\ &= \left[\left(\frac{1}{\alpha\beta}+1\right)x(2n) + \frac{1}{\alpha}x(2n-1) + x(2n-2)\right] \\ &\quad + \left[\frac{1}{\alpha}x(2n+1) + x(2n) + x(2n+2)\right].\end{aligned} \quad (5)$$

Four intermediate variables, namely, *Dk* 1 (*n*), *Dk* 2 (*n*), *Dk*3 (*n*),and *Dk*4 (*n*), are defined as below, where *k* stands for different values in the row and column transforms. In the row transform, *k* means the number of rows in progress, whereas in the column transform, *k* represents the number of scans, and one scan means finishing parallel scan of two adjacent rows in the column transform. Thus

D1(n) = 1/ a x (2n+1)+x(2n) (6)

$$D_1^k(n) = \frac{1}{\alpha}x(2n+1) + x(2n) \quad (7)$$

$$D_3^k(n) = \frac{1}{\gamma}y(2n+1) + y(2n) \quad (8)$$

$$D_4^k(n) = \left(\frac{1}{\delta\gamma}+1\right)y(2n) + \frac{1}{\gamma}y(2n-1) + y(2n-2) \quad (9)$$

Rearranging (4) with the method deriving to (5), then substituting (6)–(9) into (1), (3), (5), and the expression rearranged from (4), the following expressions can be derived

$$\frac{1}{\alpha}y(2n+1) = D_1^k(n) + x(2n+2) \quad (10)$$

$$\frac{1}{\alpha\beta}y(2n) = D_2^k(n) + D_1^k(n) + x(2n+2) \quad (11)$$

$$\frac{1}{\gamma}H(2n+1) = D_3^k(n) + y(2n+2) \quad (12)$$

$$\frac{1}{\delta\gamma}L(2n) = D_4^k(n) + D_3^k(n) + y(2n+2) \quad (13)$$

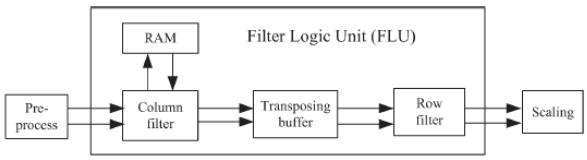


Figure 1: Proposed overall architecture of a 2D-DWT

Compared with the flipping-based lifting scheme, our modified algorithm suggests a novel way in data combination with different coefficients in even data and simplifies the computation process. The proposed algorithm combines the predictor with the updater. The high-pass signal and the low-pass signal can be calculated in parallel through the two-input/two-output architecture. At the same time, the coefficients of even items are changed by inversion of the factors. As a result, (5) can be divided into two asymmetrical parts, and data with coefficients $(1 + 1/\alpha\beta)$ and $(1/\alpha)$ can be obtained from the first pipelining stage by dividing multiplication and addition into two pipelining stages. Intermediate variables *Dk* 1 (*n*) and *Dk* 2 (*n*), as shown in (6) and (7), are introduced into (5) to calculate $y(2n + 1)/\alpha$ and $y(2n)/\alpha\beta$ together in the second pipelining stage with a critical path delay of *Tm*. Simultaneously, *Dk* 1 (*n*) and *Dk* 2 (*n*) are updated. The high-pass signal and the low-pass signal can be directly outputted from the third pipelining stage. Therefore, the pipelining stages of the 1-D processing element (PE) are limited to three, and the number of registers is further reduced.

3. Proposed Architecture for the 2-D Dwt

3.1 Overall Architecture

Based on the proposed modified algorithm, a novel architecture for the 2-D DWT shown in Fig. 1 is proposed. First, the serial-parallel conversion for the original data in the preprocessing module is carried out. After that, data are sent into the column filter for the column transform. Next, the output data of the column filter are sent into the transposing buffer, where the data transposition is operated to meet the order of the data flow required by the row filter. Then, the row filter begins to read the data from the transposing buffer for the row transform. Finally, the scaling module is used to finish the scaling computation.

3.2 Parallel Scan of Preprocessing and Raw Image Data

Since parallel scanning is adopted, as shown in Fig. 2, the data of each even row and odd row of the columns are alternately read. This way, the column filter can process the column transform for the data of adjacent columns alternately. With the preprocessing module, raw image data are factorized into odd and even parts for the following filter logic unit (FLU) module.

3.3 One-Dimensional PE

The architecture of the proposed 1-D PE is shown in Fig. 3.This architecture can be applied in the column and row filters by selecting the RAM or Buffer properly.

International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878
Volume 1 Issue 1, September 2013

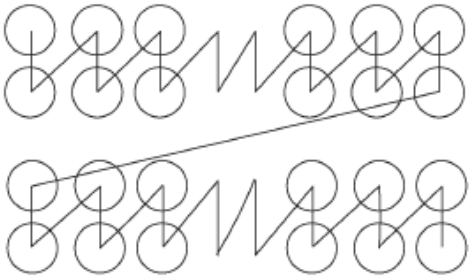


Figure 2: Parallel scanning input data

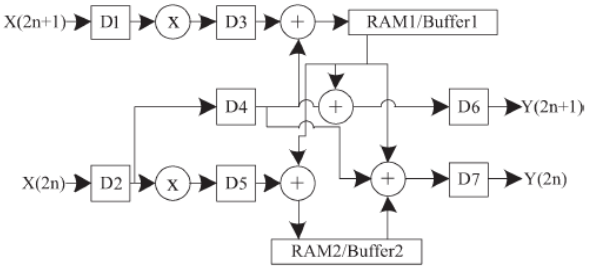


Figure 3: Proposed 1D PE

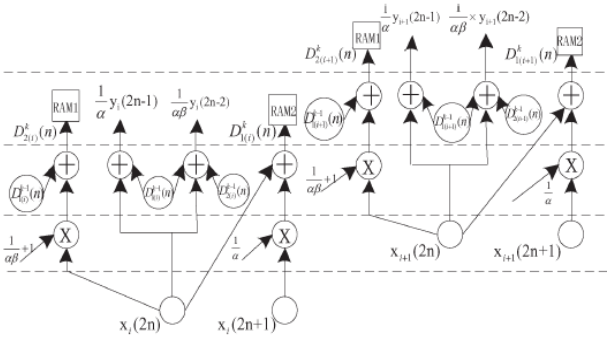


Figure 4: Data flow of first lifting step in column filter

In order to reduce the size of the transposing buffer between the column and row filters and to improve the processing speed, our design adopts the architecture of two-input/two-output. When the column filter begins to work, the input data from the preprocessing module, the odd term $xi(2n + 1)$ and the even term $xi(2n)$, are sent into the column filter in each clock cycle simultaneously. In this case, i mean the column index, and n has the same meaning as k, which represents the number of scans in the column. Then, each input is multiplied by the corresponding coefficient. The result of multiplication is used in the computation of the intermediate variables $Dk\ 1(i)(n)$ and $Dk\ 2(i)(n)$, which will be stored in the temporal buffers RAM1 and RAM2, respectively. At the same time, if $k > 1$, $Dk-1\ 1(i)\ (n)$ and $Dk-1\ 2(i)\ (n)$ will be read for the computation of $yi(2n - 1)$ and $yi(2n - 2)$. Otherwise, the boundary extension will be processed. Because of parallel scanning, $Dk-1\ 1(i)\ (n)$ and $Dk-1\ 2(i)\ (n)$ read from RAM1 and RAM2 in current clock cycle are calculated and saved at the corresponding places in the previous two rows. Then, they are used by the new data to carry out the current column transform. RAM1 and RAM2 are both dual-port RAM devices with the depth of $2N$. Fig. 4 shows the data flow of the first lifting step in the column filter. Such architecture not only has low hardware complexity but also reduces the critical path between every two registers to one multiplier. For the row filter, two buffers with the size of two registers are enough to store the intermediate variables. In the row filter, the input data will take part in the computation of intermediate variables $Dk\ 1\ (n + 1)$ and $Dk\ 2\ (n + 1)$ after two clock cycles. Then, $Dk1\ (n + 1)$ and $Dk\ 2\ (n + 1)$ are fed into Buffer1 and Buffer2, which are used to meet the requirement of the alternate reading intermediate variables in the odd and even rows and to make the computation in the odd and even rows independent without disturbing each other. $Dk\ 1\ (n)$ and $Dk\ 2\ (n)$, which are stored in another addresses in Buffer1 and Buffer2 are read out to calculate $y(2n + 1)$ and $y(2n)$ at the same time. It should be noted that n is the variable to judge the boundary extension in the row filter instead of the variable k in the column filter. Table I shows the data flow of the first lifting step in the row filter, where subscripts e and o represent the even and odd rows, respectively. In the row filter, k represents the number of row in which the current calculation is going on. We set $k = 0$ for the first even row and $k = 1$ for the first odd row. The following cases can be obtained in the same manner as above. Note that the aforementioned "row" refers to the flow order of the input data for the transposing buffer.

3.4 Transposing Module

The architecture of the transposing module is shown in Fig. 4. Three registers and two multiplexers are used to make the output data meet the order of the data flow required by the row filter. Fig. 5 also shows the orders of the input and the output for the transposing module, where L and H represent the low-pass signal and the high-pass signal of the column transform output, respectively.

4. Performance Analysis and Comparison

Tables II and III show comparisons of the proposed architecture with several previous designs for the 2- and 1-D 9/7 DWTs. As for the 2-D architecture, the RAM-based architecture [12] reduces the size of the transposing buffer to $1.5N$. However, the critical path delay is not optimized, and it requires a large memory with the size of $4N$. In the fast architecture (FA) and the high-speed architecture (HA) [13], the hardware requirements are reduced by reusing the predictor and updater circuits, while it needs the feedback results of the predicting process for the updating process. Thus, the pipelined structure cannot be achieved, and the critical path delay is $Tm + 2Ta$. In addition, although these two architectures employ parallel filters to implement the transposing buffer with four registers, they need additional buffers to cache the input data, and the overall size of the on-chip memory is large. Another architecture employing parallel filters is the dual-scan architecture (DSA) [14]. This design reduces the transposing buffer to four registers. However, the critical path delay of $4Tm + 8Ta$ limits its applications. The parallel-based lifting scheme architecture (PLSA) [15] is a modified flipping structure with a critical path of one multiplier and the temporal buffer with $4N$ size, while its transposing buffer size is $1.5N$. Cheng and Parhi [16] proposed a VLSI architecture that has high speed at the cost of large hardware requirements. If the parallel level of the 9/7 filter is one, the computation time will be $N2/2$.

International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878
Volume 1 Issue 1, September 2013

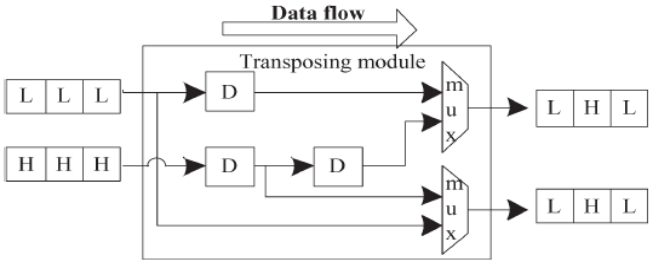
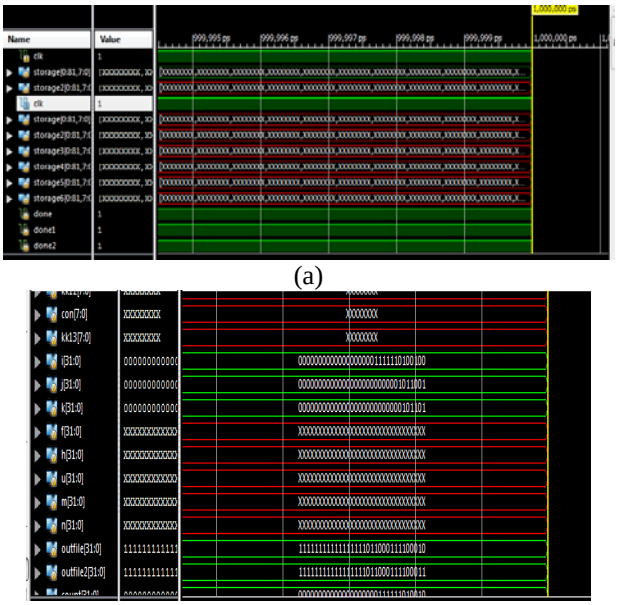


Figure 5: Detail architecture of transporting model and order of the input and output

At the same time, the number of adders will increase to 40 with the temporal buffer of 14*N* size. Besides, the critical path delay in this design is *Tm* + 4*Ta*. Based on the pyramid algorithm, throughput-scalable hybrid-pipeline architecture was proposed by Mohanty and Meher [17], whereas high throughput can only be achieved in case of the high degree of parallelism. According to [17], if the throughput reaches 64*/Tm* + 2*Ta*, the numbers of multipliers, adders, and registers will be 388, 688, and 98, respectively. Moreover, its critical patch delay is *Tm* + 2*Ta*. In [18], a two-input/two-output architecture based on the flipping structure was proposed. The design reconstructs the flipping structure by substituting the traditional five-stage flipping structure with a two-stage structure. All multipliers are included in the first stage, and adders are included in the second stage. This way, the number of registers can be reduced. However, there are accumulated operations in the second stage. The critical path will be either five adders or one multiplier, which depends on the specific process technology adopted by the ASIC implementation. For deep sub-micrometer processes, the critical path delay is more likely to be 5*Ta*, much larger than one *Tm*. Thus, the throughput for the chip is impacted. As for the 1-D architecture, the flipping structure [11] provides a new method to shorten the delay. The critical path is dominated by *Tm* + 5*Ta*, and the 1-D DWT only needs four registers. When this design is pipelined into five stages, the critical path delay is *Tm*. Nevertheless, if this 1-D processor is adopted to construct the 2-D DWT, each intermediate variable of the column transform must be stored, and the five-stage pipelined architecture needs 11 registers. For an image of $N \times N$, the size of the temporal buffer will reach 11*N*.

A different pipelined architecture was proposed in [19] by Wu and Wang. Except that the multiplication is replaced by shifted additions, this architecture does not make any modifications on the lifting algorithm. Although the critical path delay is 2*Ta*, 38 registers are needed. Two other architectures without multipliers are bit-serial (BS) and bit-parallel (BP) structures [20]. These two designs are based on the distributed arithmetic. By presorting the intermediate variables related to multiplication in ROM, multipliers can be also eliminated. However, in the BS architecture, a critical patch delay of *LTa* is produced by the calculation of partial outputs, where *L* is the bit width of the intermediate variables that depends on the magnitude of the input data. Generally, *L* is larger than 8. Although the critical patch delay in BP can be reduced to *Ta*, the number of pipeline registers and adders will significantly increase. Based on Wu and Lin's modified lifting scheme, Lai *et al.* [10] proposed a two-input/two-output architecture with a critical path delay of *Tm*. However, Lai *et al.* used the method of symmetrical component combination. Thus, more delay and pipeline registers are required to store the intermediate variables. To implement the 1-D DWT architecture, 22 registers are needed. Table IV shows the synthesis result of the proposed design and Lai *et al.*'s design. Our work is based on SMIC 0.18 μm technology, and the clock frequency is 100 MHz. obviously, the area of the proposed architecture is about 10% smaller than that of Lai *et al.*'s. Furthermore, the transposing module in [10] needs four registers to store data in every two clock cycles. The control circuit to adjust the flow of the data is also more complicated than ours, which leads to additional hardware requirements. Based on our modified lifting scheme, the proposed two input/ two-output 2-D DWT architecture is implemented with a parallel scanning method. The one step lifting circuit adopts three pipelining stages, and the number of registers is 7.



(a)

(b)

Figure 6: (a) Scanning the first order of the image; (b) finding the error and correction

5. Conclusion

In this brief, we have proposed a novel architecture for the 1- and 2-D DWTs. The modified one lifting step circuit can work within three pipelining stages with fewer registers, and the critical path delay is *Tm*. For the 2-D DWT architecture, only the temporal buffer with 4*N* size is used in the column filter. A detailed analysis is performed to compare the proposed architectures with other previous architectures in terms of hardware complexity, critical path delay, storage size, computation time, and throughput. According to the results, the proposed architecture can achieve high speed with lower hardware complexity and smaller storage size.

References

[1] G. Xing, J. Li, and Y. Q. Zhang, "Arbitrarily shaped video-object coding by wavelet," *IEEE Trans. Circuits*

Syst. Video Technol., vol. 11, no. 10, pp. 1135–1139, Oct. 2001.

[2] S. C. B. Lo, H. Li, and M. T. Freedman, "Optimization of wavelet decomposition for image compression and feature preservation," *IEEE Trans. Med. Imag.*, vol. 22, no. 9, pp. 1141–1151, Sep. 2003.

[3] K. K. Parhi and T. Nishitani, "VLSI architecture for discrete wavelet transforms," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 1,no. 2, pp. 191–202, Jun. 1993.

[4] P.Wu and L. Chen, "An efficient architecture for two-dimensional discrete wavelet transform," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 4, pp. 536–545, Apr. 2001.

[5] W. Sweldens, "The new philosophy in biorthogonal wavelet constructions," in *Proc. SPIE.*, 1995, vol. 2569, pp. 68–79.

I. Daubechies and W. Sweldens, "Factoring wavelet transform into lifting steps," *J. Fourier Anal. Appl.*, vol. 4, no. 3, pp. 245–267, Mar. 1998.

[6] J. M. Jou, Y. H. Shiau, and C. C. Liu, "Efficient VLSI architectures for the biorthogonal wavelet transform by filter bank and lifting scheme," in *Proc. IEEE ISCAS*, May 2001, vol. 2, pp. 529–532.

[7] G. Shi, W. Liu, and L. Zhang, "An efficient folded architecture for liftingbased discrete wavelet transform," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 56, no. 4, pp. 290–294, Apr. 2009.

[8] B. F. Wu and C. F. Lin, "A high-performance and memory-efficient pipeline architecture for the 5/3 and 9/7 discrete wavelet transform of JPEG2000 codec," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 12, pp. 1615–1628, Dec. 2005.

[9] Y. K. Lai, L. F. Chen, and Y. C. Shih, "A high-performance and memory-efficient VLSI architecture with parallel scanning method for 2-D lifting-based discrete wavelet transform," *IEEE Trans. Consum. Electron.* vol. 55, no. 2, pp. 400–407, May 2009.

[10] C.-T. Huang, P.-C. Tseng and L.-G. Chen, "Flipping structure: An efficient VLSI architecture for lifting-based discrete wavelet transform," *IEEE Trans. Signal Process.*, vol. 52, no. 4, pp. 1080–1089, Apr. 2004.

[11] P.-C. Tseng, C.-T. Huang and L.-G. Chen, "Generic RAM-based architecture for two dimensional discrete wavelet transform with line based method," in *Proc. Asia-Pacific Conf. Circuits Syst.*, 2002, vol. 2, pp. 363–366.

[12] C. Xiong, J. Tian, and J. Liu, "Efficient architectures for two-dimensional discrete wavelet transform using lifting scheme," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 607–614, Mar. 2007.

[13] H. Liao, M. K. Mandal, and B. F. Cockburn, "Efficient architectures for 1-D and 2-D lifting-based wavelet transforms," *IEEE Trans. Signal Process.*, vol. 52, no. 5, pp. 1315–1326, May 2004.

[14] C.-Y. Xiong, J.-W. Tian, and J. Liu, "A note on 'flipping structure: An efficient VLSI architecture for lifting-based discrete wavelet transform'," *IEEE Trans. Signal Process.*, vol. 54, no. 5, pp. 1910–1916, May 2006.

[15] C. Cheng and K. K. Parhi, "High-speed VLSI implement of 2-D discrete wavelet transform," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 393–403, Jan. 2008.

[16] B. K. Mohanty and P. K. Meher, "Throughput-scalable hybrid-pipeline architecture for multilevel lifting 2-D DWT of JPEG 2000 coder," in *Proc. IEEE Int. Conf. Appl.-Specific Syst., Archit. Processors*, 2008, pp. 305–309.

[17] J. Song and I.-C. Park, "Novel pipelined DWT architecture for dual-line scan," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2009, pp. 373–376.

[18] Z. G.Wu and W.Wang, "Pipelined architecture for FPGA implementation of lifting-based DWT," in *Proc. Int. Conf. Elect. Inform. Control Eng.*, 2011, pp. 1535–1538.

[19] B. K. Mohanty and P. K. Meher, "Efficient multiplier less designs for 1-D DWT using 97 filters based on distributed arithmetic," in *Proc. Int. Symp. Integr. Circuits*, 2009, pp. 364–367.