International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:3, No:6, 2009

High Performance VLSI Architecture of 2D Discrete Wavelet Transform with Scalable Lattice Structure

Juyoung Kim, Taegeun Park

***Abstract*—**In this paper, we propose a fully-utilized, block-based 2D DWT (discrete wavelet transform) architecture, which consists of four 1D DWT filters with two-channel QMF lattice structure. The proposed architecture requires about 2MN-3N registers to save the intermediate results for higher level decomposition, where M and N stand for the filter length and the row width of the image respectively. Furthermore, the proposed 2D DWT processes in horizontal and vertical directions simultaneously without an idle period, so that it computes the DWT for an N×N image in a period of $N^2(1-2^{-2J})/3$. Compared to the existing approaches, the proposed architecture shows 100% of hardware utilization and high throughput rates. To mitigate the long critical path delay due to the cascaded lattices, we can apply the pipeline technique with four stages, while retaining 100% of hardware utilization. The proposed architecture can be applied in real-time video signal processing.

***Keywords*—**discrete wavelet transform, VLSI architecture, QMF lattice filter, pipelining.

I. Introduction

RECENTLY, discrete wavelet transform (DWT), based on time-scale representations, has received a considerable attention as one of the most promising alternatives to the discrete cosine transform (DCT) in video compression applications such as JPEG2000 and MPEG4. Wavelet transforms are closely related to tree structured digital filter banks. Therefore, the wavelet transform can be considered as a multi-resolution signal analysis, which decomposes a signal into its components in different frequency bands. DWT does not have the blocking effect inherent to discrete cosine transform (DCT) and provides a relatively high compression rate, so that it can be applied to low bit-rate image compression. The applications of DWT include, but are not limited to image compression, speech analysis, pattern recognition, and computer vision. Since DWT demands massive computations, an implementation using general-purpose computing system is not adequate for the real-time applications.

In recent years, many researchers have proposed a number of VLSI architectures on DWT to achieve real-time signal processing. There are two approaches to compute the 2D DWT: separable and non-separable. A simple separable approach to compute the 2D DWT is processing the horizontal direction followed by the vertical direction or vice versa by cascading 1D DWT devices. However, this approach requires a transposition memory to keep an intermediate result of 1D DWT [7]. A non-separable approach for the 2D DWT directly decomposes an image into four quadrants (subimages) without row and column processes one after another [6][8]. This can improve the performance because it computes the 2D DWT outputs directly, however the dedicated four 2D filters require considerably more hardware resources. Efficient architectures employing the two-channel QMF lattice have been proposed [5][9]. The architecture in [9] is not scalable to the filter length and the resolution level, and requires much more resources as the resolution level increases. The lattice structure for the 1D DWT proposed by Kim [5] is regular and scalable but the structure itself cannot be readily applied to the 2D DWT due to its complex folded scheduling control. The recursive 1D DWT architectures (systolic, semi-systolic) [7] show a good performance and hardware utilization, but still require complex routing networks to feed data into systolic array according to the DWT scheduling. Yu and Chen [10] proposed a separable approach for 2D DWT by data dependency analysis that reduced the intermediate hardware requirements between resolution levels. Parallel approaches have been suggested to increase performance, but these require more resources due to their multiple filter banks [6][7][10]. The architectures employing the polyphase decomposition technique show 100% hardware utilization, however they require a RAM module to save an intermediate data [11]. The architecture in [12] shows high performance with relatively less hardware but the hardware is not fully utilized.

In this paper, we propose a scalable, high performance VLSI architecture for the 2D DWT with 100% hardware utilization. The proposed lattice architecture minimizes the storage necessary for the transposition in the separable approach and improves the performance by simultaneously processing the 2D DWT in horizontal and vertical directions. The control design for the data flow is scalable to extend to an arbitrary 2D DWT with M taps and J levels. The proposed lattice structure is so regular that it is suitable for VLSI implementation and requires $N^2(1-2^{-2J})/3$ of period to compute an $N\times N$ image. This paper

Juyoung Kim is with Nexuschips, Seoul, Korea.
Taegeun Park is with department of Information, Communication, and Electronics Engineering, The Catholic University of Korea, Bucheon-shi, Kyungki-do, Korea (corresponding author; e-mail: parktg@catholic.ac.kr).

outlines as follows. Section 2 introduces to the DWT. Section 3 describes the proposed 2D lattice architecture. The performance analysis and simulation for the architecture are discussed in Section 4. Finally, concluding remarks are given in Section 5.

II. Discrete Wavelet Transform

In wavelet analysis, signals are represented using a set of basis functions derived by shifting and scaling a single prototype function, referred to as "mother wavelet", in time [1]. Wavelet transforms are closely related to tree structured digital filter banks and multiresolution analysis. A set of wavelet basis functions can be generated by translating and dilating the mother wavelet. Whereas the classical short-time Fourier transform uses the same size of filter on the entire time-frequency domain, the wavelet transform takes different size of windows at variable scale, so that it is suitable to analyze spatial and spectral locality [2]. The wavelet transform can be viewed as a decomposition of a signal in the time-scale (frequency) plane. Therefore an arbitrary signal *f(t)* can be expressed as

$$f(t)=\sum_k c_{j_0}(k)\varphi_{j_0,k}(t)+\sum_{j\ge j_0}\sum_k d_j(k)\psi_{j,k}(t)\ , \quad (1)$$

$$c_j =< f(t),\varphi_{j,k}(t) >= \int f(t)\varphi_{j,k}(t)dt\ , \quad (2)$$

$$d_j =< f(t),\psi_{j,k}(t) >= \int f(t)\psi_{j,k}(t)dt\ , \quad (3)$$

where c_j and d_j are the wavelet coefficients at level *j*, and $\varphi(t)$ and $\psi(t)$ are the father and the mother wavelet functions respectively.

A simple iterative algorithm, known as Mallat's recursive pyramid algorithm (RPA) has been proposed [3]. In this algorithm, the DWT coefficients of any level in analysis stage can be defined from the DWT coefficients of the previous level and calculated by decimation and filtering. The equations to compute a 1D DWT can be expressed as follows:

$$c_j(k)=\sum_{m=0}^{N-1}h(m-2k)c_{j-1}(m)$$

$$d_j(k)=\sum_{m=0}^{N-1}g(m-2k)c_{j-1}(m)$$

where *N* is the number of samples, $c_j(k)$ and $d_j(k)$ are the wavelet coefficients at level *j*, and $h(m)$ and $g(m)$ are low-pass and high-pass filters obtained from the wavelet. The synthesis of discrete-time signal can be done by interpolation and filtering, which is expressed as follows:

$$c_j(k)=\sum_{m=0}^{N-1}h(m-2k)c_{j+1}(m)+\sum_{m=0}^{N-1}g(m-2k)d_{j+1}(m)$$

The reconstructed signal differs from the original data due to aliasing, amplitude distortion, and phase distortion. Filters can be designed in such a way that some or all of these distortions are eliminated. Therefore, the wavelet basis functions are designed to satisfy the perfect reconstruction (PR) property for computing the original signal correctly. The DWT is implemented using a tree-structured filter bank, where the *M* wavelet coefficients are computed through $\log_2 M$ octave levels. At each octave level *j*, an input sequence is fed into the low-pass filter $h(m)$ and the high-pass filter $g(m)$. A number of filterbanks are cascaded to produce a multi-resolution wavelet analysis. Figure 1 shows the analysis stage of three-level wavelet decomposition in binary tree. Here, the filters $H(z)$, $G(z)$ satisfy the PR property [1].

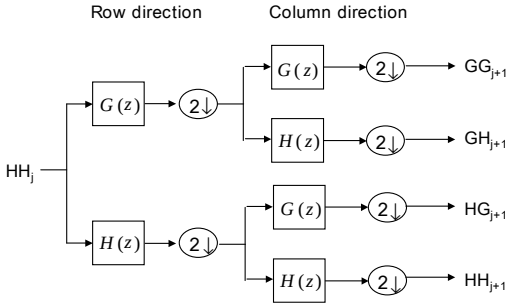
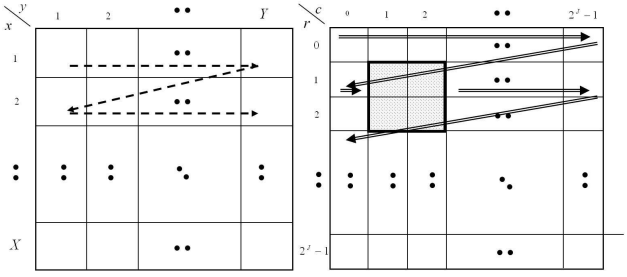


Fig. 1 One level of separable 2D DWT (Analysis stage).

III. The Proposed Architecture

A. New Data Scheduling

In this paper, we propose a new data scheduling to design a high performance, fully-utilized 2D DWT filter. Fig. 2 explains the new scheduling which scans an image in block basis. As depicted in Fig. 2(a), an *M*x*N* image is divided into *X*x*Y* of 2^Jx2^J blocks, where *J* is the resolution level and *M* and *N* is a multiple of 2^J. For each block of size 2^Jx2^J showed in Fig. 2(b), we input four contiguous data showed in the rectangle at the same time to the filter and calculate the 2D DWT with *J* resolution level so that the hardware efficiency of the architecture reaches to 100% by eliminating an idle period.



(a) *M* x *N* image (b) 2^J x 2^J block

Fig. 2 The proposed scan method

Before we explain the detailed scheduling, we will set some notations and definitions. An input image data is defined as $u^{x,y}(r,c)$, where (x,y) and (r,c) stand for the block location in the image and the pixel location in the block (x,y) respectively, where $1\le x\le X$, $1\le y\le Y$, and $0\le r,c\le 2^J$-1. And the results of 2D DWT with *J* resolution level are defined as $LL_j^{x,y}(t_r,t_c)$, $LH_j^{x,y}(t_r,t_c)$, $HL_j^{x,y}(t_r,t_c)$, and $HH_j^{x,y}(t_r,t_c)$, where (t_r, t_c) stands for the operation time in row and column direction at its

corresponding level *j* respectively. We compute the first level of 2D DWT with the four inputs of $u^{x,y}(2t_r,2t_c)$, $u^{x,y}(2t_r,2t_c\text{-}1)$, $u^{x,y}(2t_r\text{-}1,2t_c)$, and $u^{x,y}(2t_r\text{-}1,2t_c\text{-}1)$ for $2^{2(J-1)}$ cycles and then output $2^{2(J-1)}$ of $LL_1^{x,y}(t_r,t_c)$, $LH_1^{x,y}(t_r,t_c)$, $HL_1^{x,y}(t_r,t_c)$, and $HH_1^{x,y}(t_r,t_c)$, where $0\le t_r,t_c\le 2^{J-1}\text{-}1$. In the same method, we can compute the *j*-level of 2D DWT with the four input of $LL_{j-1}^{x,y}(2t_r,2t_c)$, $LL_{j-1}^{x,y}(2t_r,2t_c-1)$, $LL_{j-1}^{x,y}(2t_r-1,2t_c)$, and $LL_{j-1}^{x,y}(2t_r-1,2t_c-1)$ for $2^{2(J-j)}$ cycles and then output $2^{2(J-j)}$ of $LL_j^{x,y}(t_r,t_c)$, $LH_j^{x,y}(t_r,t_c)$, $HL_j^{x,y}(t_r,t_c)$ and $HH_j^{x,y}(t_r,t_c)$, where $2\le j\le J$ and $0\le t_r,t_c\le 2^{J-j}\text{-}1$. When t_r and t_c are equal to zero, $2t_r$-1 and $2t_c$-1 become -1 which are beyond the block boundary. In other words, when t_r is equal to zero, the contiguous data in the upward block to the current block is referenced. Also, when t_c is equal to zero, the contiguous data in the left block to the current block is referenced. This ensures the correct DWT operation in the block boundary.

j	t_r \ t_c	0	1	2	3
1	0	H_1 V_1 / H_1	H_1 V_1 / H_1	H_1 V_1 / H_1	H_1 V_1 / H_1
	1	H_1 V_1 / H_1	H_1 V_1 / H_1	H_1 V_1 / H_1	H_1 V_1 / H_1
	2	H_1 V_1 / H_1	H_1 V_1 / H_1	H_1 V_1 / H_1	H_1 V_1 / H_1
	3	H_1 V_1 / H_1	H_1 V_1 / H_1	H_1 V_1 / H_1	H_1 V_1 / H_1
2	0	H_2 V_2 / H_2	H_2 V_2 / H_2		
	1	H_2 V_2 / H_2	H_2 V_2 / H_2		
3	0	H_3 V_3 / H_3			

Fig. 3 Block scheduling scheme for three level 2D DWT

Fig. 3 shows the proposed scheduling in a block of 8x8 size, when the level *J* is equal to 3. The parameters t_r and t_c are used to show the scan order and the computation timing. Also the figure shows that the proposed scheduling computes the 2D DWT without any idle period, thus the hardware can be fully utilized.

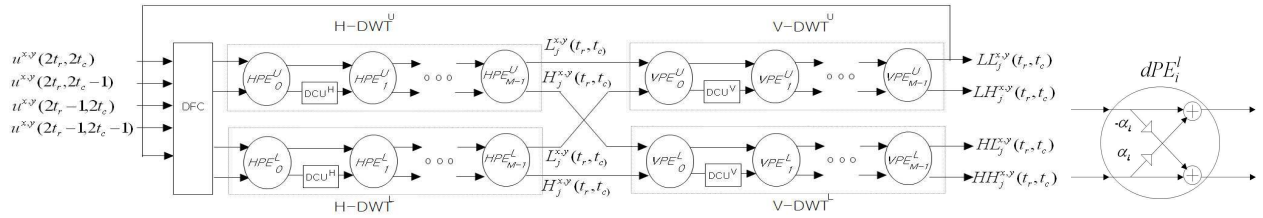
B. Proposed 2D DWT Architecture

The QMF lattice structure requires only *K* lattices for filter length 2*K*, each of which consists of two multipliers and two adders [4]. Therefore the required hardware complexity is about half of that of a pair of direct approach FIR filters. In addition, lattice digital filters are implemented as a cascade of regular modules so that these filters are suitable for VLSI implementation. A scalable folded 1D DWT architecture based on lattice structure computes outputs of the other resolution levels during idle periods by decimations [5]. However, the 1D architecture itself is difficult to apply to compute the 2D DWT directly due to the complex scheduling control. Moreover, if we adopt a separable approach (row scan followed by column scan or vice versa), the memory module is inevitably necessary to transpose the elements of the 2D DWT. In this paper, we propose a high performance 2D DWT architecture, based on the modified separable iterative approach [6], which interspersed the higher-level computations between the first level computations. Fig. 4 shows the structure of the proposed 2D DWT, which consists of four 1D lattice filters of length 2*M* and its basic processing element (PE). The proposed architecture processes the horizontal and vertical directions at the same time to minimize the storage for an intermediate data. Moreover, it is scalable to extend to an arbitrary 2D DWT with *M* taps and *J* levels.

First two lattice filters, H-DWT^U and H-DWT^L, which take two rows as inputs, are dedicated to compute 1D DWT in a horizontal direction, whereas the next two lattice filters, V-DWT^U and V-DWT^L compute 1D DWT in a vertical direction. The lattices for the horizontal direction process four inputs simultaneously to maximize the hardware utilization and improve the performance. The outputs of the horizontal filters are fed into the vertical filters; lowpass- and highpass-filtered outputs are going to the upper filter (V-DWT^U) and the lower filter (V-DWT^L) respectively. The filter output LL_j is going back to the input of the filter for higher-level decompositions.

The input and output relations of each PE are defined as depicted in Fig. 5. Each PE is defined as dPE_i^l, where *d* can be *H* for the horizontal filter and *V* for the vertical filter, *i* stands for the *i*-th lattice, and *l* stands for the filter location which can be either U (upper) or L (lower) respectively.

Now we explain the structure of the Data format Converter (DFC) to control the input for the H-DWT filters which includes the primary inputs $u(t_r,t_c)$ and the intermediate feedback result $VY_{M-1}^{UU}(x,y,j,t_r,t_c)$. The intermediate result can be also considered as $LL_j(t_r,t_c)$ which is the lowpass-filtered output at *j*-level. The DFC stores the intermediate results for higher resolution level computations, due to the time difference between the available moment and the necessary moment of the data. Fig. 6 shows the structure of DFC for 3-level 2D DWT. The four outputs in Fig. 6 are fed into the HPE_0^U and HPE_0^L which are the first lattices in the horizontal filters H-DWT^U and H-DWT^L. The block diagram tells us which inputs should be available to the corresponding PE at the specific time. The triple notation (j,t_r,t_c) in front of the registers explains which data among the intermediate *j*-level lowpass-filtered output $LL_j(t_r,t_c)$ is going to be stored at time (t_r,t_c) for higher level decomposition. Another triple notation (j,t_r,t_c) in front of the multiplexers shows the timing to be output through the multiplexers, which is used to design multiplexer controls. The required hardware for the DFC to compute the *J*-level 2D DWT is $2^{2(J-1)}-2^{J-1}+(Y+1)(2^J-2)$ of registers and four multiplexers, where *J* is the resolution level and *Y* is the number of blocks in the horizontal direction.



(a) Proposed 2D DWT lattice architecture of length 2*M* (b) PE structure

Fig. 4 Proposed 2D DWT architecture and its PE structure

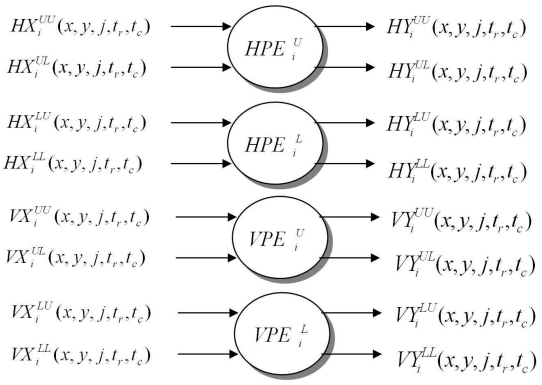


Fig. 5 Inputs and outputs of the PE

The Delay Control Unit (DCU) controls the delays between lattices to adjust the data rate due to the decimation. Fig. 7(a) and 7(b) shows the structures of the DCU^H and DCU^V for 3-level 2D DWT respectively. The outputs in Fig. 7(a) are fed into the HPE_i^U and HPE_i^L in the H-DWT^U and H-DWT^L, where $1 \le i \le M-1$ and M is the filter length. The DCU^H is located between the PEs and controls the delay z^{-2^j} depending on the resolution level j. The DCU^H for the J-level 2D DWT requires $2^J - 1$ of registers and a multiplexer. When the H-DWT^H filter computes the j-level DWT at time (k,l), the content of the register $R_{j,l}$ is output to $HX_{i+1}^{UL}(x, y, j, t_r, t_c)$. When $k \neq 0$, $HY_i^{UL}(x, y, j, t_r, t_c)$ is stored in the register $R_{j,l-1}$. Otherwise, $HY_i^{UL}(x, y-1, j, t_r, 2^{J-j}-1)$ stored in the register $R_{j-1,2^{J-j}-1}$ which is computed from the contiguous block on the left is output to $HX_{i+1}^{UL}(x, y, j, t_r, t_c)$. The operation and the structure of the DCU^H in the H-DWT^L filter are identical. The output of the H-DWT and the input of the V-DWT are crossed to process the 2D DWT in horizontal and vertical directions simultaneously.

The operation of the DCU^V can be explained in the same way as for the DCU^H. The DCU^V for the J-level 2D DWT requires $Y(2^J - 1)$ of registers and a multiplexer. When the V-DWT^H filter computes the j-level DWT at time (k,l), the content of the register $R_{y,j,l}$ is output to $VX_{i+1}^{UL}(x, y, j, t_r, t_c)$. When $k \neq 0$, $VY_i^{UL}(x, y, j, t_r, t_c)$ is stored in the register $R_{y,j,l-1}$. Otherwise, $VY_i^{UL}(x-1, y, j, 2^{J-j}-1, t_c)$ stored in the register $R_{y,j-1,2^{J-j}-1}$ which is computed from the contiguous upward block is output to $VX_{i+1}^{UL}(x, y, j, t_r, t_c)$. The operation and structure of the DCU^V in the V-DWT^L filter are identical.

The proposed 2D DWT architecture may suffer from the long critical path due to the cascaded lattice structure. In this case we can apply the pipelining technique to the proposed architecture. Since the $(j-1)$-level lowpass-filtered output $LL_{j-1}^{x,y}(0,0)$ to compute $LL_j^{x,y}(0,0)$ is available 4 clock cycle ahead, we can apply up to four-stage of pipeline at maximum. Otherwise, the hardware efficiency will be less than 100%.

IV. Performance Analysis

The proposed 2D DWT architecture is designed and verified with VerilogHDL and Modelsim design environment. Various architectures to compute the 2D DWT, such as the direct architecture [7], parallel architecture [10], systolic-parallel architecture [7], non-separable architecture [6], SIMD architecture [6], polyphase architecture [11], 2D lattice architecture [12], and RAM-based architecture [13] have been proposed recently. In Table I, we compare the performance of the proposed architecture and these architectures in terms of the number of multipliers, the number of adders, storage size, computing time, hardware utilization, and control complexity. A straightforward direct approach uses the minimum hardware, but results in very poor performance [7]. The parallel approaches in [6][10] use more resources with relatively less gain in computation time. The recursive DWT architectures (systolic and semi-systolic) [7] show a good performance and hardware utilization, but still require complex routing networks and scheduling. The architecture with the polyphase decomposition technique show 100% hardware utilization, however it requires a considerable storage size since a RAM module is necessary to save an intermediate data [11]. The architecture in [12] shows high performance with relatively less hardware but the hardware is not fully utilized. The RAM-based architecture [13] needs more hardware and relatively longer computation time compared to the proposed architecture. The computing time for the $N \times N$ image in the proposed architecture is $N^2(1-2^{-2J})/3$ because the horizontal and vertical directions are processed simultaneously in the proposed block-based scheduling. The proposed architecture may suffer from long critical delay path due to the cascade of processing elements. The problem can be overcome by inserting pipeline registers between processing elements using cut-set retiming method [4].

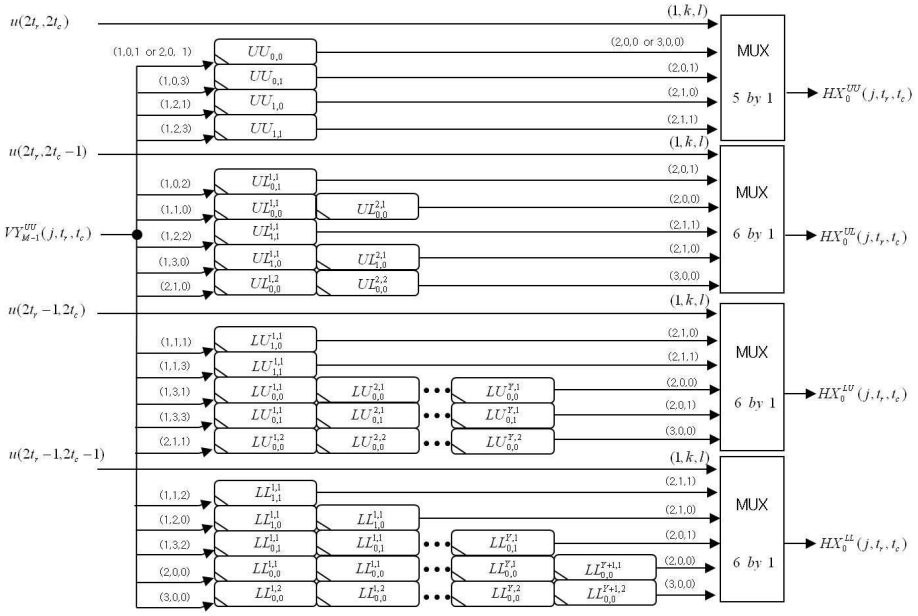
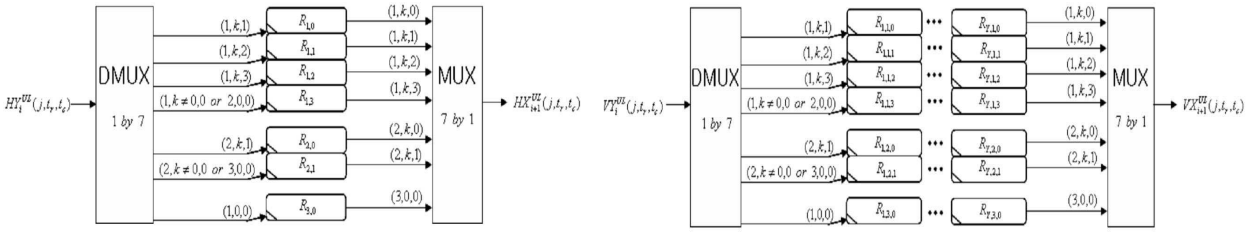


Fig. 6 The structure of DFC (J = 3)



(a) The structure of DCUH (b) The structure of DCUV

Fig. 7 The structure of DCUH and DCUV (J = 3)

TABLE I
PERFORMANCE COMPARISON OF 2D DWT ARCHITECTURE
(N^2: image size, M: filter length, J: decomposition level)

	Multipliers	Adders	Storage size	Computing time	Utilization	Control
Direct [7]	M	M	N^2	$4N^2$	100%	simple
Systolic-parallel [7]	$4M$	$4M$	$2N(M+2)$	N^2-N	high	complex
Non-separable [6]	$2M^2$	$2(M^2-1)$	$2(M^2+MN)$	N^2	high	complex
SIMD [6]	$2N^2$	$2N^2$	$3N^2$	M^2J	low	complex
Parallel [10]	$3M$	$3(M-1)$	$2N(M-1)$	N^2	low	simple
Polyphase [11]	$4M$	$4M$	$N^2/4+K(N+1)$	$0.5N^2 \sim 0.67N^2$	100%	moderate
2D Lattice [12]	$4M$	$4M$	$M(N+J)$	$N^2/2$	67%	complex
RAM-based [13]	$6M$	$6M$	$N(2M-0.5)$	$N^2/2$	89%	complex
Proposed	$4M$	$4M$	$2MN-3N$	$N^2(1-2^{-2J})/3$	100%	complex

Compared to conventional approaches, the proposed architecture requires relatively less hardware while providing high performance with 100% hardware utilization.

V. CONCLUSIONS

In this paper, we proposed a fully-utilized, high-performance 2D DWT architecture which performs the horizontal and vertical wavelet filtering simultaneously so that it computes the 2D DWT for an N×N image within a period of $N^2(1-2^{-2J})/3$. Four inputs are fed to the proposed architecture at the same time to maximize the throughput. We also introduce the new block-based scheduling with the block size of 2^Jx2^J to reach the hardware efficiency to 100%, where J is the resolution level. The required memory is about $2MN$-$3N$, where M and N stand

for the filter length and the row width of the image respectively. To mitigate the delay due to long critical path, we can apply the pipeline technique up to four stages without sacrificing hardware efficiency. The proposed architecture can be applied for various real-time image/video applications such as JPEG-2000 and MPEG4.

ACKNOWLEDGMENT

We are grateful to IC Design Education Center that provides us with a design environment.

REFERENCES

[1] P. P. Vaidyanathan, *Multirate systems and Filter Banks*, Englewood Cliffs, Prenctice-Hall, 1993
[2] O. Rioul and M. Vetterli, "Wavelets and signal processing," *IEEE Signal Process. Magazine*, vol.8, no.4, pp.14-38, 1991
[3] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol.11, no.7, pp.674-693, 1989
[4] K. Pahri, *VLSI digital signal processing systems*, John Wiley & Sons, 1999
[5] J. T. Kim, Y. H. Lee, T. Isshiki, and H. Kunieda, "Scalable VLSI architectures for lattice structure-based discrete wavelet transform," *IEEE Trans. Circuits Systems II, Analog Digit. Process.*, vol.45, no.8, pp.1031-1043, 1998
[6] C. Chakrabarti and M. Vishwanath, "Efficient realizations of the discrete and continuous wavelet transforms: from single chip implementations to mappings on SIMD array computers," *IEEE Trans. Signal Process.*, vol.43, no.3, pp.759-771, 1995
[7] M. Vishwanath, R. M. Owens, and M. J. Irwin, "VLSI architectures for the discrete wavelet transform," *IEEE Trans. Circuits Systems II, Analog Digit. Process.*, vol.42, no.5, pp.305-316, 1995
[8] F. Marino, "Efficient high-speed/low-power pipelined architecture for the direct 2-D discrete wavelet transform," *IEEE Trans. Circuits Systems II, Analog Digit. Process.*, vol.47, no.12, pp.1476-1491, 2000
[9] T. C. Denk and K. K. Pahri, "Architectures for lattice structure based orthogonal discrete wavelet transform," *IEEE Trans. Circuits Systems II, Analog Digit. Process.*, vol.44, no.2, pp.129-132, 1997
[10] C. Yu and S.-J. Chen, "Design of an efficient VLSI architecture for 2-D discrete wavelet transforms," *IEEE Trans. Consumer Elect.*, vol.45, no.1, pp.135-140, 1999
[11] P. Wu and L.-G. Chen, "An efficient architecture for two-dimensional discrete wavelet transform," *IEEE Trans. Circuits Systems, Video Technol.*, vol.11, no.4, pp.536-545, 2001
[12] T. Park and S. Jung, "High speed lattice based VLSI architecture of 2D discrete wavelet transform for real-time video signal processing," *IEEE Trans. Consumer Elect.*, vol.48, no.4, pp.1026-1032, 2002
[13] C.-T. Huang, P.-C. Tseng, and L.-G. Chen, "Generic RAM-based architectures for two-dimensional discrete wavelet transform with line-based method," *IEEE Trans. Circuits Systems, Video Technol.*, vol.15, no.7, pp.910-920, 2005.