Hardware-efficient systolic-like modular design for two-dimensional discrete wavelet transform

Meher, Pramod Kumar; Mohanty, Basant Kumar; Patra, Jagdish Chandra

2008

Meher, P. K., Mohanty, B. K., & Patra, J. C. (2008). Hardware-Efficient Systolic-Like Modular Design for Two-Dimensional Discrete Wavelet Transform. IEEE Transactions on Circuits and Systems II: Express Briefs, 55(2), 151-155.

<https://hdl.handle.net/10356/94344>

<https://doi.org/10.1109/TCSII.2007.911801>

© 2007 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at: [DOI: <http://dx.doi.org/10.1109/TCSII.2007.911801>].

Downloaded on 26 Apr 2023 20:52:21 SGT

Hardware-Efficient Systolic-Like Modular Design for Two-Dimensional Discrete Wavelet Transform

Pramod Kumar Meher, *Senior Member, IEEE*, Basant Kumar Mohanty, *Member, IEEE*, and Jagdish Chandra Patra, *Member, IEEE*

Abstract—A systolic-like modular architecture is presented for hardware-efficient implementation of two-dimensional (2-D) discrete wavelet transform (DWT). The overall computation is decomposed into two distinct stages; where column processing is performed in stage-1, while row processing is performed in stage-2. Using a new data-access scheme and a novel folding technique, the computation of both the stages are performed concurrently for transposition-free implementation of 2-D DWT. The proposed design can offer nearly the same throughput rate, and requires the same or less the number of adders and multipliers as the best of the existing structures. The storage space is found to occupy most of the area in the existing 2-D DWT structures but the proposed structure does not require any on-chip or off-chip storage of input samples or storage/transposition of intermediate output. The proposed one, therefore, involves considerably less hardware complexity compared with the existing structures. Apart from that, it has less duration of cycle period in comparison to the existing structures, and has a latency of $O(K)$ cycles while all the existing structures have latency of $O(N)$ cycles, the filter order K being small compared to the input size N .

Index Terms—Discrete wavelet transform (DWT), systolic array, VLSI, 2-dimensional (2-D) DWT.

I. INTRODUCTION

THE two-dimensional (2-D) discrete wavelet transform (DWT) is frequently encountered in numerous applications involving image and video compression, pattern recognition, bioinformatics, and many more [1]–[5]. Due to its remarkable advantage over the discrete cosine transform (DCT), 2-D DWT has been adopted for the JPEG-2000 compression standard [6]. The implementation of this transform, however, is computation intensive and most of its applications demand real-time processing. High-speed calculation of 2-D DWT to meet the timing requirement of real-time applications is, consequently, considered as an important task. Keeping this in view, several architectures have been suggested in the last decade for efficient realization of 2-D DWT in dedicated VLSI systems [7]–[14]. The 2-D DWT can be implemented either by non-separable direct approach or by separable indirect approach. The direct approach [13], [14] of 2-D DWT

involves latency of a few clock cycles only, but requires $(K/2)$ times more computation (where K is the order of 2-D filter bases) compared with the separable approach; and requires proportionately more hardware to achieve the same throughput performance as the latter. Although the nonseparable devices have very small latency, they are not suitable for cost-effective realization due to their large hardware complexity. In the indirect approach, the 2-D DWT architectures are designed by separating the transform into row- and column-transformations. A straightforward implementation of each level of decomposition by separable approach consists of three distinct stages. In the first stage, 1-D DWT is performed on each of the N rows of the input of size $(N \times N)$ to obtain two intermediate matrices of size $(N \times N/2)$. In the second stage, the intermediate matrices are transposed and in the third stage, N number of 1-D DWT are performed again on the N rows of the transposed intermediate matrices to obtain the desired 2-D DWT. The area-time complexity of a computing structure for 2-D DWT thus amounts to that of its constituent 1-D modules along with the overheads pertaining to the transposition of intermediate output matrix.

Although the performance and complexity of a 2-D DWT architecture are substantially dependent on the implementation of its 1-D modules, the data-access mechanism along with the size and location of the input and intermediate buffers greatly affects the overall performance of the computing structure. According to the data-access mechanism, the 2-D DWT architectures are categorized into three main classes, e.g., block-based architectures [15], level-by-level architectures [16] and line-based recursive architectures [9], [17]. It has been observed that none of the architectures has definite advantage in all respects over the others, but each one may defeat the others in terms of storage requirements, processing speed, and energy dissipation under specific conditions, e.g., number of levels of decomposition, size and location of on-chip and off-chip memory, and the order of the 2-D wavelet filter bases [17], [18]. Comparisons of various aspects of 2-D DWT architectures are presented in [17]–[19]. In this brief, we propose a new data-access scheme and a novel folding approach to have a transposition-free concurrent row-column processing in order to eliminate the requirement of storage for intermediate output.

Necessary mathematical formulation is presented in Section II and the proposed architecture is described in Section III. The hardware complexity and performance of the proposed structure are discussed in Section IV. Conclusion of the paper is presented in Section V.

Manuscript received May 31, 2007; revised August 31, 2007. This paper was recommended by Associate Editor E. A. B. da Silva.

P. K. Meher and J. C. Patra are with the School of Computer Engineering, Nanyang Technological University, Singapore, 639798 (e-mail: aspkmeh@ntu.edu.sg; aspatra@ntu.edu.sg).

B. K. Mohanty is with the Department of Electronics and Communication Engineering, Jaypee Institute of Engineering and Technology, Raghuagarh 473226, India (e-mail: bk.mohanti@jiet.ac.in).

TABLE I
INPUT DATA-ACCESS-SCHEME USED BY THE PROPOSED STRUCTURE

cycle no.	input block	cycle no.	input block	cycle no.	input block	...	cycle no.	input block
1	$\mathbf{I}(1, 1)$	$r + 1$	$\mathbf{I}(2, 1)$	$2r + 1$	$\mathbf{I}(3, 1)$	...	$(P - 1)r + 1$	$\mathbf{I}(P, 1)$
2	$\mathbf{I}(1, 2)$	$r + 2$	$\mathbf{I}(2, 2)$	$2r + 2$	$\mathbf{I}(3, 2)$	...	$(P - 1)r + 2$	$\mathbf{I}(P, 2)$
3	$\mathbf{I}(1, 3)$	$r + 3$	$\mathbf{I}(2, 3)$	$2r + 3$	$\mathbf{I}(3, 3)$	...	$(P - 1)r + 3$	$\mathbf{I}(P, 3)$
...	...	...	...	...	...	...	...	...
N	$\mathbf{I}(1, N)$	$r + N$	$\mathbf{I}(2, N)$	$2r + N$	$\mathbf{I}(3, N)$	...	$(P - 1)r + N$	$\mathbf{I}(P, N)$
$N + 1$	$\mathbf{I}(1, 1)$	$r + N + 1$	$\mathbf{I}(2, 1)$	$2r + N + 1$	$\mathbf{I}(3, 1)$	...	$(P - 1)r + N + 1$	$\mathbf{I}(P, 1)$
...	...	...	...	...	...	...	...	...
r	$\mathbf{I}(1, K')$	$2r$	$\mathbf{I}(2, K')$	$3r$	$\mathbf{I}(3, K')$	...	Pr	$\mathbf{I}(P, K')$

$r = (N + K - 2)$, $P = N/2$ and $K' = (K - 2)$. The input image blocks are given by $\mathbf{I}(j, k) = \{A((2j - 2)_N, (k - 1)_N), A((2j - 1)_N, (k - 1)_N), A((2j)_N, (k - 1)_N), \dots, A((2j + K - 3)_N, (k - 1)_N)\}$ for $1 \leq j \leq P$ and $1 \leq k \leq r$. The symbol $(\cdot)_N$ denotes modulo N operation.

II. MATHEMATICAL FORMULATION

The 2-D DWT with separable wavelet bases of any decomposition level can be obtained from the scaling coefficients of its previous level using the pyramid algorithm (PA) [20] as

$$A^l(m, n) = \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} h_1(i) \cdot h_2(j) \cdot A^{l-1}(2m - j, 2n - i) \quad (1a)$$

$$B^l(m, n) = \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} h_1(i) \cdot g_2(j) \cdot A^{l-1}(2m - j, 2n - i) \quad (1b)$$

$$C^l(m, n) = \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} g_1(i) \cdot h_2(j) \cdot A^{l-1}(2m - j, 2n - i) \quad (1c)$$

$$D^l(m, n) = \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} g_1(i) \cdot g_2(j) \cdot A^{l-1}(2m - j, 2n - i). \quad (1d)$$

$A^l(m, n)$, $B^l(m, n)$, $C^l(m, n)$ and $D^l(m, n)$ are the coefficients of l th stage of 2-D DWT. K represents the length of the low-pass and high-pass filters. For the first level decomposition, the input matrix $[A^0(m, n)]$ is of size $(N \times N)$, while each of the four sub-bands is of size $(N/2) \times (N/2)$. Without loss of generality of the design, we may assume N to be a power-of-two number, for simplicity of presentation. The computations of (1) may be decomposed into two distinct stages of computation as

$$A^l(m, n) = \sum_{j=0}^{K-1} h_2(j) \cdot V_L^{l-1}(m, 2n - j) \quad (2a)$$

$$B^l(m, n) = \sum_{j=0}^{K-1} g_2(j) \cdot V_L^{l-1}(m, 2n - j) \quad (2b)$$

$$C^l(m, n) = \sum_{j=0}^{K-1} h_2(j) \cdot V_H^{l-1}(m, 2n - j) \quad (2c)$$

$$D^l(m, n) = \sum_{j=0}^{K-1} g_2(j) \cdot V_H^{l-1}(m, 2n - j) \quad (2d)$$

for $0 \leq m, n \leq M - 1$ and $M = N/2^l$, where

$$V_L^{l-1}(m, n) = \sum_{i=0}^{K-1} h_1(i) \cdot A^{l-1}(2m - i, n) \quad (3a)$$

$$V_H^{l-1}(m, n) = \sum_{i=0}^{K-1} g_1(i) \cdot A^{l-1}(2m - i, n) \quad (3b)$$

for $0 \leq n \leq 2M - 1$ and $0 \leq m \leq M - 1$.

Using (2) and (3), the 2-D DWT of any given level of decomposition can, therefore be computed in two distinct stages.

- In stage-1, 1-D DWT is performed on the columns of the input matrix $[A^{l-1}(m, n)]$ according to (3) for obtaining the low-pass intermediate matrix $[V_L^{l-1}(m, n)]$, and the high-pass intermediate matrix $[V_H^{l-1}(m, n)]$.
- In stage-2, 1-D DWT is again performed on the rows of two intermediate output matrices $[V_L^{l-1}(m, n)]$ and $[V_H^{l-1}(m, n)]$ according to (2) for obtaining the desired 2-D DWT coefficients $\{A^l(m, n)\}$, $\{B^l(m, n)\}$, $\{C^l(m, n)\}$, and $\{D^l(m, n)\}$.

The key idea used here is to generate the intermediate output of stage-1 in row-wise manner, which could directly be used for the computation of stage-2 without transposition. To achieve this, we have proposed a new data access scheme which is discussed in detail in the following section.

III. PROPOSED ARCHITECTURE

From the 2-D input matrix $[A(m, n)]$ of size $N \times N$, we derive rP number of input-blocks $\mathbf{I}(j, k)$, for $1 \leq j \leq P$ and $1 \leq k \leq r$, where $P = N/2$ and $r = N + K - 2$. Each block of input contains K values of a column of $[A(m, n)]$, such that: $\mathbf{I}(j, k) = \{A((2j - 2)_N, (k - 1)_N), A((2j - 1)_N, (k - 1)_N), A((2j)_N, (k - 1)_N), \dots, A((2j + K - 3)_N, (k - 1)_N)\}$. The symbol $(\cdot)_N$ denotes modulo N operation on the indexes, which takes care of cyclic extension of $[A(m, n)]$ by $(K - 2)$ rows and $(K - 2)$ columns. Note that cyclic extension of input matrix by $(K - 2)$ rows and $(K - 2)$ columns is required to prevent the loss of boundary information. The data-access scheme to be used in the proposed structure is shown in Table I. As shown in the table, r number of input blocks, $\mathbf{I}(1, k)$ for $1 \leq k \leq r$, are used as input during the first r cycles, out of which the first N blocks are derived from the first K rows of the input matrix $[A(m, n)]$, such that each of those input blocks belongs to a different column

Fig. 1. Proposed structure for 2-D DWT.

Fig. 2. Internal structure of subcell-1 for $K = 4$. (a) Structure for subcell-1. (b) Structure of each MU. X_0, X_1, X_2 and X_3 are the four current input samples fed to the MUs of subcell-1.

of $[A(m, n)]$. The last $(K - 2)$ blocks are repetitions of the first $(K - 2)$ blocks to realize the cyclic extension of input by $(K - 2)$ columns. The second set of r input blocks, $\mathbf{I}(2, k)$ for $1 \leq k \leq r$, to be used in the next r cycles, are similarly derived from the third row to $(K + 2)$ th row of $[A(m, n)]$. Other sets of input blocks $\mathbf{I}(j, k)$ for $3 \leq j \leq P$ and $1 \leq k \leq r$, are also derived similarly, where the adjacent input blocks pertaining to the same column of input matrix overlap by $(K - 2)$ samples.

Proposed structure for the computation of 2-D DWT is shown in Fig. 1. It consists of two subcells working in separate pipelined stages. Subcell-1 performs the computation of stage-1, while subcell-2 performs the computation of stage-2. The internal structure of subcell-1 for $K = 4$ is shown in Fig. 2. It performs two filtering operations given by (3a) and (3b) of stage-1 of the 2-D DWT computation. Subcell-1 for $K = 4$ has four multiplication units (MUs). The structure of an MU is described in Fig. 2(b). Each of the MUs stores a pair of filter coefficients of the low-pass, and the high-pass filters such that the $(k + 1)$ th MU stores the pair of coefficients $h_1(k)$ and $g_1(k)$ for $k = 0, 1, 2$ and 3 . During each computational cycle an input block of four sample values are fed in parallel to the four MUs. Four pairs of multiplications corresponding to the high- and the low-pass filtering are performed on the input samples concurrently by four MUs as shown in Fig. 2(a). The outputs of the MUs are added concurrently by six adders in a pipelined-adder-tree (PAT) to obtain a pair of filter outputs in three pipelined stages. The duration of cycle period T_C of the structure is the same as the time required to perform a multiplication in an MU. During each cycle, after a latency of three cycles, subcell-1 produces a low-pass and a high-pass intermediate output. Note that subcell-1 processes the successive input blocks in successive cycles, and generates the pair

Fig. 3. Structure and function of subcell-2 for $K = 4$. (a) Structure for subcell-2. (b) Structure of LC. (c) Function of AC and DC.

of intermediate matrices $[V_L(m, n)]$ and $[V_H(m, n)]$ row-wise which can be used directly (without transposition) as the input for subcell-2.

The computation of stage-2 given by (2) requires four filtering operations to process the output of stage-1 in order to produce four output sub-bands. All the four filtering operations can be mapped to subcell-2, where processing of the low-pass and the high-pass intermediate outputs ($[V_L(m, n)]$ and $[V_H(m, n)]$, respectively) are multiplexed together in subcell-2 to take advantage of the down-sampling of subband components followed by wavelet transformation of the rows. The internal structure of subcell-2 for $K = 4$ along with the functions of its components is depicted in Fig. 3. It consists of four MUs, three adder cells (ACs), one delay cell (DC), and a line changer (LC). During every cycle, a pair of output from subcell-1 ($V_L(m, n)$, $V_H(m, n)$) is fed to subcell-2 through the LC. The structure and function of LC are shown in Fig. 3(b). LC has two output lines, LINE-1 and LINE-2. In each cycle, the pair of inputs of LC exchange their output lines, such that if the low-pass and the high-pass intermediate outputs $V_L(m, n)$ and $V_H(m, n)$, respectively, are on LINE-1 and LINE-2 of LC during the current cycle period, then in the next cycle $V_L(m, n+1)$ and $V_H(m, n+1)$ would, respectively, be on LINE-2 and LINE-1. Input samples are fed to the MUs in subcell-2 through these pair of output lines from the LC. The sample values on LINE-1 are fed to the odd-numbered MUs (MU-1 and MU-3), while the sample values on LINE-2 are fed to the even-numbered MUs (MU-2 and MU-4). The schedule of arrival of the rows of intermediate low-pass and high-pass values is shown in Table II. Decimation pertaining to row-transformation is performed indirectly in subcell-2 by this sample loading through the LC; because, the

TABLE II
SCHEDULE OF INPUT ARRIVAL TO THE MUS OF SUBCELL-2 THROUGH THE LC

cycle no.	MU-1	MU-2	MU-3	MU-4
$ri + 1$	$V_L(i, 0)$	$V_H(i', r''')$	$V_L(i, 0)$	$V_H(i', r''')$
$ri + 2$	$V_H(i, 0)$	$V_L(i, 1)$	$V_H(i, 0)$	$V_L(i, 1)$
$ri + 3$	$V_L(i, 2)$	$V_H(i, 1)$	$V_L(i, 2)$	$V_H(i, 1)$
$ri + 4$	$V_H(i, 2)$	$V_L(i, 3)$	$V_H(i, 2)$	$V_L(i, 3)$
$ri + 5$	$V_L(i, 4)$	$V_H(i, 3)$	$V_L(i, 4)$	$V_H(i, 3)$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
$ri + r - 1$	$V_L(i, r')$	$V_H(i, r'')$	$V_L(i, r')$	$V_H(i, r'')$
$ri + r$	$V_H(i, r')$	$V_L(i, r''')$	$V_H(i, r')$	$V_L(i, r''')$

$r = (N + K - 2)$. r is even, when N and K are assumed to be even.

$r' = (r - 2)$, $r'' = r' - 1$, $r''' = r - 1$ and $i' = (i - 1)$. $0 \leq i \leq P - 1$.

computations to process the low-pass and the high-pass intermediate output ($[V_L(m, n)]$ and $[V_H(m, n)]$, respectively) are performed only once in every pair of consecutive cycles.

From Table II, it may be observed that, the computation of filter output for stage-2 for low-pass input subband $[V_L(m, n)]$ is initiated at MU-1 in odd-number cycles, while those of high-pass input subband $[V_H(m, n)]$, is initiated on even-numbered cycles. The function of the MUs in this case is the same as that of the subcell-1 as depicted in Fig. 2(b). Each of the four MUs of subcell-2 stores a pair of filter coefficients ($h_2(k)$ and $g_2(k)$) of the low-pass, and the high-pass filters of stage-2, and performs a pair of multiplications of the intermediate output value with the pair of filter coefficients. Four pairs of multiplications corresponding to the high- and the low-pass filtering are thus implemented concurrently by the four MUs as shown in Fig. 3(a) for $K = 4$. The output of the MUs are fed to the DC and three ACs. The function of DC and ACs are depicted in Fig. 3(c). The addition operations pertaining to the low-pass (LL and HL) and the high-pass (LH and HH) output subbands are computed by the ACs and a DC in a pair of separate systolic pipelines. The additions pertaining to the low-pass and the high-pass intermediate output ($[V_L(m, n)]$ and $[V_H(m, n)]$, respectively) are multiplexed in the ACs. After a latency of five cycles, subcell-2 produces two subband coefficients in every cycle period, such that during two consecutive cycles it computes four 2-D DWT coefficients corresponding to four subbands.

IV. HARDWARE AND TIME COMPLEXITIES

The entire computations of the 2-D DWT are performed in two stages. Subcell-1 performs computation of stage-1 and subcell-2 performs the computation of stage-2. Each of the subcells is comprised of K MUs, where each MU consists of $2K$ multipliers. Apart from that, each subcell consists of $2(K - 1)$ adders. Subcell-1 involves $(1 + \log_2 K)$ pipeline stages (one stage for multiplication in the MUs and $\log_2 K$ stages for the adder-tree). Subcell-2 involves $(K + 1)$ pipeline stages (one stage for the multiplications by the MUs and K stages in DC and ACs). The latency of stage-1 and stage-2, therefore, amounts to $(1 + \log_2 K)$ and $(K + 1)$ cycles, respectively. The overall latency of the structure thus amounts to $(K + \log_2 K + 2)$ cycles. After the latency period, a pair of 2-D DWT coefficients are obtained in every cycle from the structure so that $(N + K - 1)$ subband outputs corresponding to a row of intermediate result

Fig. 4. Plot of area-time complexities of proposed structure and some of the existing structures for different input sizes.

are obtained in $(N + K - 1)$ cycles. The complete DWT of input size $N \times N$ thus obtained by processing the $N/2$ rows of intermediate result in every $N(N + K - 1)/2$ cycles. The hardware complexity in terms of the number of multipliers, adders and storage words along with the time complexities of the proposed structure and the existing structures [7]–[11] in terms of average computation time (ACT), and latencies are listed in Table III.

As shown in the table, the proposed design requires the same number of cycles of ACT, and involves the same number of multipliers and adders as the structures of [7]–[9], but the structures of [7]–[9] have lower throughput rate due to their higher cycle period and require an additional storage space $O(KN)$. The structure of [10] has $(N/4)$ times less ACT and higher throughput rate, but involves higher cycle period and $(3N/8)$ times more number of multipliers and adders than the proposed structure. The structure of [11] for tile-factor $l = 1$ requires twice the multipliers and adders than the proposed design, and involves nearly twice the ACT along with storage of $KN/2$ words. The structure of [11] has the same operating frequency as the proposed one but yields lower throughput rate. The memory-complexity of the tile-based design falls by 2^l , where l is the tile-factor, but on the other hand, its ACT increases by nearly the same factor. It may be noted that the number of multipliers and adders are $O(K)$, while storage space of the existing structures is $O(KN)$. Moreover, in most practical situations, N is very large compared to K , e.g., for many image compression problems $N = 512$, while $K = 4$. The storage space, consequently, makes major contribution to the hardware complexities of the existing structures. The proposed structure, therefore, involves significantly less hardware complexity compared with the existing structures.

The area of the multipliers, adders, and storage cells along with the addition-time and multiplication-time are determined using the Synopsys DesignWare 0.18- μm TSMC library for 8-bit data-width [21]. Accordingly, we have estimated the area-complexity and the ACT of proposed structure and the existing structures of [9]–[11]. The area-time complexities for different values of N and $K = 4$ thus obtained are plotted in Fig. 4. The proposed structure is found to involve significantly low area-time complexity compared with the others. The proposed design is coded in VHDL and simulated using Xilinx ISE 9.1i tool for validation. The simulation results obtained for {2-D DWT} of different sizes, is found to be as expected from the theoretical result. The area-complexity in terms of

TABLE III
COMPARISON OF HARDWARE AND TIME COMPLEXITIES OF THE PROPOSED STRUCTURE AND THE EXISTING STRUCTURES

structures	storage words	multipliers	adders	cycle period	latency (cycles)	ACT
Pan <i>et al</i> [7]	$K(N + K/2 + 5/2)$	$4K$	$4K$	$T_M + 2T_A$	$2K + (K - 1)N/2$	$(N + K - 1)N/2$
Wu <i>et al</i> [8]	$KN + K$	$4K$	$4K$	$T_M + T_A$	$K/2 + (K - 1)N/2$	$(N + K - 1)N/2$
Huang <i>et al</i> [9]	$KN + 3K - N/2$	$4K$	$4K$	$T_M + T_A$	$K + (K - 1)N/2$	$(N + K - 1)N/2$
Pan <i>et al</i> [10]	$N(N + 3K/2 - 3/2)$	$3NK/2$	$3N(K - 1)/2$	$T_M + 2T_A$	$2(N + K + 1)$	$2(N + K - 1)$
Dimitroulakos <i>et al</i> [11]	$(KN)/(2^l)$	$8K$	$8(K - 1)$	T_M	$N(K - 1)/4$	$[2^{l+1}N + (K - 1)]N/4$
proposed structure	$K + 2$	$4K$	$4(K - 1)$	T_M	$(K + 2 + \log_2 K)$	$(N + K - 1)N/2$

The throughput rate of the structures can be estimated as $1/[\text{ACT} \times T_C]$, where ACT is in terms of number of cycles and T_C is the cycle period of the respective structures. Operating frequencies of the structures can be obtained as the reciprocal of their respective cycle-times. The ACT of the tile-based structure of [11] $[2^l(N^2/2 - 1) + N(K - 1)]/4 + 1$ is approximated to $[2^l(N^2/2 - 1) + N(K - 1)]/4$ assuming N^2 to be large compared 1.

TABLE IV
IMPLEMENTATION RESULTS FOR DIFFERENT FPGA DEVICES

devices	slices	4-input LUTs	max. frequency
Virtex 4 XC4VLX15	163	77	281.2 MHz
Virtex E XCV50E	326	160	173.6 MHz
Virtex 2P (XC2VP2)	303	173	230.3 MHz

the number slices, number of LUTs, and maximum usable frequencies obtained from the synthesis report for devices of different speed grades are listed in Table IV.

V. CONCLUSION

Using a new data-access scheme and a novel folding approach, we have designed a hardware-efficient systolic-like architecture for 2-D DWT. The overall computation is decomposed into two distinct stages; and both the stages are implemented concurrently in a modular structure consisting of two fully-pipelined subcells for transposition-free realization of 2-D DWT. Unlike the existing structures, the proposed one does not involve any on-chip memory for storage of intermediate data. The area-time complexity of proposed design is found to be significantly less compared with the existing designs. Apart from these, the proposed structure has a latency of $O(K)$ while all the other structures have latency $O(N)$. It would, therefore, be very much suitable for VLSI implementation of 2-D DWT for real-time reactive systems and low latency applications. The proposed single-level DWT can be extended further for multi-level DWT computation either by level-level or by recursive approach. The proposed data-access scheme may be utilized further to derive an area-time efficient design for the computation of lifting-based 2-D DWT, as well.

REFERENCES

- [1] Y. Meyer, *Wavelets: Algorithms and Applications*. Philadelphia, PA: SIAM, 1993.
- [2] A. S. Lewis and G. Knowles, "Image compression using 2-D wavelet transform," *IEEE Trans. Imag. Process.*, vol. 1, no. 2, pp. 244–250, Apr. 1992.
- [3] G. M. Davis, "A wavelet-based analysis of fractal image compression," *IEEE Trans. Imag. Process.*, vol. 7, no. 2, pp. 141–154, Feb. 1998.
- [4] X. H. Wang, R. S. H. Istepanian, and H. S. Yong, "Application of wavelet modulus maxima in microarray spots recognition," *IEEE Trans. Nanobiosci.*, vol. 2, no. 4, pp. 190–192, Dec. 2003.
- [5] L. Shutao, L. Chen, and J. T. Kwok, "Wavelet-based feature extraction for microarray data classification," in *Proc. Int. Joint Conf. Neural Networks, (IJCNN '06)*, Jul. 2006, pp. 5028–5033.
- [6] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 36–58, Sep. 2001.
- [7] S. B. Pan and R. H. Park, "VLSI architectures of the 1-D and 2-D discrete wavelet transforms for JPEG 2000," *Signal Process.*, vol. 82, no. 7, pp. 981–992, Jul. 2002.
- [8] P.-C. Wu and L.-G. 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.
- [9] 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 Syst. Video Technol.*, vol. 15, no. 7, pp. 910–920, Jul. 2005.
- [10] S. B. Pan and R. H. Park, "Systolic array architectures for computation of the discrete wavelet transform," *J. Visual Commun. Imag. Representation*, vol. 14, no. 3, pp. 217–231, Sep. 2003.
- [11] G. Dimitroulakos, M. D. Galanis, A. Milidonis, and C. E. Goutis, "A high-throughput, memory efficient architecture for computing the tile-based 2-D discrete wavelet transform for the JPEG2000," *Integration*, vol. 39, pp. 1–11, 2005.
- [12] F. Marino, "Efficient high-speed/low-power pipelined architecture for the direct 2-D discrete wavelet transform," *IEEE Trans. Circuit. Syst. II, Analog Digit. Signal Process.*, vol. 47, no. 12, pp. 1476–1491, Dec. 2000.
- [13] F. Marino, "Two fast architectures for the direct 2-D discrete wavelet transform," *IEEE Trans. Signal Process.*, vol. 49, no. 6, pp. 1248–1259, Jun. 2001.
- [14] C. Yu and S. J. Chen, "VLSI implementation of 2-D discrete wavelet transform for real-time video signal processing," *IEEE Trans. Consum. Electron.*, vol. 43, no. 4, pp. 1270–1279, Nov. 1997.
- [15] G. Lafruit, L. Nachtergaele, J. Bormans, M. Engels, and I. Bolsens, "Optimal memory organizations for scalable texture codecs in MPEG-4," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 3, pp. 218–242, Mar. 1999.
- [16] S.-K. Paek and L.-S. Kim, "2D DWT VLSI architecture for wavelet Imag. Process.," *Electron. Lett.*, vol. 34, no. 6, pp. 537–538, Mar. 1998.
- [17] N. D. Zervas, G. P. Anagnostopoulos, V. Spiliotopoulos, Y. Andreopoulos, and C. E. Goutis, "Evaluation of design alternatives for the 2-D-discrete wavelet transform," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 12, pp. 1246–1262, Dec. 2001.
- [18] C.-T. Huang, P.-C. Tseng, and L.-G. Chen, "Analysis and VLSI architecture for 1-D and 2-D discrete wavelet transform," *IEEE Trans. Signal Process.*, vol. 53, no. 4, pp. 1575–1586, Apr. 2005.
- [19] C. Chakrabarti, M. Vishwanath, and R. M. Owens, "Architectures for wavelet transform: A survey," *J. VLSI Signal Process.*, vol. 4, pp. 171–192, Mar. 1996.
- [20] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 7, pp. 674–693, Jul. 1989.
- [21] "Synposys, DesignWare," Foundry Libraries, Mountain View, CA [Online]. Available: <http://www.synposys.com/>