

Short Paper

The Segmented-Matrix Algorithm for Haar Discrete Wavelet Transform

PO-YUEH CHEN, EN-CHI LIAO AND CHUNG-WEI LIANG

Department of Computer Science and Information Engineering

National Changhua University of Education

Changhua, 500 Taiwan

E-mail: pychen@cc.ncue.edu.tw

Discrete wavelet transform (DWT) is an efficient tool for multi-resolution decomposition of images. It has been shown to be very promising due to its high compression ratio and self-similar data structure. Conventionally a 2-D DWT is accomplished by performing two 1-D operations: one along the rows and the other along the columns of an image. Without executing ordered 1-D transforms, we develop a new algorithm to compute a 2-D Haar DWT, the simplest DWT. Two merits of this algorithm are compactness and quickness. The algorithm is implemented with a compact, regular VLSI architecture whose system throughput can be conveniently improved by appropriate parallel/pipeline methods.

Keywords: segmented-matrix algorithm, Haar discrete wavelet transform, VLSI architectures, parallel, pipeline

1. INTRODUCTION

Discrete wavelet transform (DWT) has been widely used in signal and image processing, especially in multi-resolution representation [1]. It is suitable for progressive transmission on network because it decomposes the image into several sub-bands [2] and encodes them according to their importance. Recently, the prevailing JPEG 2000 standard is also developed based on DWT [3, 4]. Moreover, there are many other vital applications for DWT such as image coding [5], image compression [6], image segmentation [7], and computer graphics [8].

Haar discrete wavelet transform (Haar DWT) is the simplest of all wavelets. It has been applied in image multi-resolution for many years [9, 10]. The computation process is extremely simple for 1-D transform (the filter coefficients are either 1 or -1). However, conventionally a 2-D DWT (including Haar DWT) is split into two sequenced 1-D operations: one along the rows and the other along the columns of the image pixels. This is detrimental to the system design flexibility because the column operations are postponed until all row operations are completed. In this paper, exploiting the non-overlapping characteristic of Haar DWT, we propose "The Segmented-matrix Algorithm" that

Received July 17, 2006; revised January 19 & March 20, 2007; accepted May 23, 2007.
Communicated by Ja-Ling Wu.

calculates a 2-D Haar DWT by one matrix multiplication. It significantly enhances the system design flexibility by applying some well designed data re-arrangements.

In recent years, matrix algorithms for Haar wavelet have been proposed in the literature. Some of them calculate 2-D Haar DWT by the Kronecker product of matrices [11, 12]. Some researchers applied matrix-vector multiplications to manipulate 2-D DWT analysis. However, the computation process still involves ordered rows and columns operations [13, 14]. Applying the proposed algorithm, we can eliminate the waiting phase and design a VLSI architecture with a lot more flexibility.

DWT is a computationally intensive process. When it is implemented in a general purpose computing system, the system performance cannot meet the requirement of some real-time applications. It is essential to develop VLSI chips for DWT applied in real-time systems [15-17]. In recent years, many researchers make efforts in FPGA (Field Programmable Gate Array) design for DWT [18-20] because the system prototypes can be implemented within a short time. In this paper, we use the Verilog Hardware Description Language (HDL) to implement the proposed VLSI architecture on an FPGA chip. Because of the parallel characteristic of the new algorithm, the system speed can be improved as much as we need at the expense of extra hardware.

In section 2, we briefly review the conventional approach and describe the proposed algorithm step by step. A regular VLSI architecture is developed in section 3. In section 4, the simulation results of both software and hardware implementation are discussed and analyzed. Finally, we conclude in section 5.

2. CONVENTIONAL APPROACH AND THE PROPOSED METHOD

2.1 Motivation

The conventional ordered 1-D DWT method is demonstrated in Fig. 1. Observing the final resulted 2-D Haar DWT coefficients in Fig. 1 (c) carefully, we find there are only four of them expressed by linear combinations of A, B, E and F. The coefficients for these four linear combinations are (1, 1, 1, 1), (1, 1, -1, -1), (1, -1, 1, -1) and (1, -1, -1, 1) respectively. Base on this observation, we can integrate these coefficients into a 2-D coefficient matrix and those four wavelet coefficients can be computed by Eq. (1):

$$\begin{aligned}
 [A \ B \ E \ F] \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} & \quad (1) \\
 = [A+B+E+F \quad A-B+E-F \quad A+B-E-F \quad A-B-E+F]. &
 \end{aligned}$$

Apply this simple idea to all pixels, we develop a new algorithm named “The Segmented-Matrix Algorithm” for computing 2-D Haar DWT. This name results from the fact we need to break the original image into some 2×2 matrices (For example, in the upper part of Fig. 1 (a), A, B, E and F form such a matrix and so do C, D, G and H.) before applying the matrix multiplication. The proposed algorithm is described step by step in the following sub-section.

$$\begin{array}{cc}
 \begin{bmatrix} A & B & C & D \\ E & F & G & H \\ I & J & K & L \\ M & N & O & P \end{bmatrix} & \begin{bmatrix} (A+B) & (C+D) & (A-B) & (C-D) \\ (E+F) & (G+H) & (E-F) & (G-H) \\ (I+J) & (K+L) & (I-J) & (K-L) \\ (M+N) & (O+P) & (M-N) & (O-P) \end{bmatrix} \\
 \text{(a) The original image.} & \text{(b) The row operation results.} \\
 \begin{bmatrix} (A+B)+(E+F) & (C+D)+(G+H) & (A-B)+(E-F) & (C-D)+(G-H) \\ (I+J)+(M+N) & (K+L)+(O+P) & (I-J)+(M-N) & (K-L)+(O-P) \\ (A+B)-(E+F) & (C+D)-(G+H) & (A-B)-(E-F) & (C-D)-(G-H) \\ (I+J)-(M+N) & (K+L)-(O+P) & (I-J)-(M-N) & (K-L)-(O-P) \end{bmatrix} \\
 \text{(c) The final results of a 2-D Haar DWT.}
 \end{array}$$

Fig. 1. Conventional 2-D Haar DWT computation.

2.2 Proposed Method

First of all, let us define some matrices in order to explain the algorithm step by step with convenience:

- (a) The original image data N : an $m \times n$ matrix which stands for the original image (assume m and n are multiples of 2).
- (b) The sub-blocks B_{ij} : 2×2 matrices, where i ranges from 1 to $m/2$ and j ranges from 1 to $n/2$. In other words, N is divided into $mn/4$ such sub-blocks. The method to obtain these B_{ij} from N will be described in details later.
- (c) The filter coefficient matrix: $C = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$. (2)
- (d) The row vectors A_{ij} : $mn/4$ such matrices of size 1×4 . They are obtained by applying some data re-arrangement to B_{ij} respectively.
- (e) The intermediate matrix M : an $\frac{mn}{4} \times 4$ matrix.
- (f) Matrix M' : an $\frac{mn}{4} \times 4$ matrix.
- (g) Matrix W : an $m \times n$ matrix. This is the final result of 2-D Haar DWT coefficients matrix.

The Segmented-Matrix Algorithm for Haar FDWT is described step by step as follows:

Step 1: The first procedure for the segmented-matrix algorithm is to form some sub-blocks B_{ij} for representing the original image data N . The representation is illustrated as in Eq. (3):

$$B_{11} = \begin{bmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{bmatrix} \quad B_{12} = \begin{bmatrix} N_{13} & N_{14} \\ N_{23} & N_{24} \end{bmatrix} \quad \dots \quad B_{1,n/2} = \begin{bmatrix} N_{1,n-1} & N_{1,n} \\ N_{2,n-1} & N_{2,n} \end{bmatrix}$$

$$\begin{aligned}
 B_{21} &= \begin{bmatrix} N_{31} & N_{32} \\ N_{41} & N_{42} \\ \vdots & \vdots \end{bmatrix} & B_{22} &= \begin{bmatrix} N_{33} & N_{34} \\ N_{43} & N_{44} \\ \vdots & \vdots \end{bmatrix} & \cdots & B_{2,n/2} &= \begin{bmatrix} N_{3,n-1} & N_{3,n} \\ N_{4,n-1} & N_{4,n} \\ \vdots & \vdots \end{bmatrix} \\
 B_{m/2,1} &= \begin{bmatrix} N_{m-1,1} & N_{m-1,2} \\ N_{m,1} & N_{m,2} \end{bmatrix} & B_{m/2,2} &= \begin{bmatrix} N_{m-1,3} & N_{m-1,4} \\ N_{m,3} & N_{m,4} \end{bmatrix} & \cdots & B_{m/2,n/2} &= \begin{bmatrix} N_{m-1,n-1} & N_{m-1,n} \\ N_{m,n-1} & N_{m,n} \end{bmatrix}
 \end{aligned} \tag{3}$$

where N_{ij} are the elements of N .

Step 2: Z-scan each B_{ij} to generate $mn/4$ row vectors A_{ij} by data rearrangement. For example,

$$B_{11} = \begin{bmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{bmatrix} \rightarrow A_{11} = [N_{11} \ N_{12} \ N_{21} \ N_{22}]. \tag{4}$$

Step 3: Express the intermediate matrix M with these A_{ij} using Eq. (5):

$$M = \begin{bmatrix} A_{11} = [N_{11} \ N_{12} \ N_{21} \ N_{22}] \\ A_{12} = [N_{13} \ N_{14} \ N_{23} \ N_{24}] \\ \vdots \\ A_{1,n/2} = [N_{1,n-1} \ N_{1,n} \ N_{2,n-1} \ N_{2,n}] \\ A_{21} = [N_{31} \ N_{32} \ N_{41} \ N_{42}] \\ \vdots \\ A_{2,n/2} = [N_{3,n-1} \ N_{3,n} \ N_{4,n-1} \ N_{4,n}] \\ \vdots \\ A_{m/2,1} = [N_{m-1,1} \ N_{m-1,2} \ N_{m,1} \ N_{m,2}] \\ \vdots \\ A_{m/2,n/2} = [N_{m-1,n-1} \ N_{m-1,n} \ N_{m,n-1} \ N_{m,n}] \end{bmatrix}. \tag{5}$$

Step 4: Apply matrix multiplication to compute the matrix $M' = \frac{1}{2}(M \times C)$. We denote the contents of M' as in Eq. (6):

$$M' = \begin{bmatrix} M'_{11} & M'_{12} & M'_{13} & M'_{14} \\ M'_{21} & M'_{22} & M'_{23} & M'_{24} \\ \vdots & \vdots & \vdots & \vdots \\ M'_{mn/4,1} & M'_{mn/4,2} & M'_{mn/4,3} & M'_{mn/4,4} \end{bmatrix}. \tag{6}$$

Step 5: In order to describe the process conveniently, let us divide W into four sub-matrices W_{LL} , W_{HL} , W_{LH} , and W_{HH} (all 4 are $\frac{m}{2} \times \frac{n}{2}$ matrices). The final result W then looks like:

$$W = \begin{bmatrix} W_{LL} & W_{HL} \\ W_{LH} & W_{HH} \end{bmatrix}. \tag{7}$$

The elements of these sub-matrices are the elements in M' rearranged. Specifically, the elements in the first column of M' are filled in W_{LL} row by row W_{HL} . Similarly, the 2nd, 3rd, 4th column of M' are row by row filled in W_{HL} , W_{LH} , and W_{HH} respectively. As a result,

$$\begin{aligned} W_{LL} &= \begin{bmatrix} M'_{1,1} & M'_{2,1} & \dots & M'_{n/2,1} \\ M'_{(n/2)+1,1} & M'_{(n/2)+2,1} & \dots & M'_{n,1} \\ \vdots & \vdots & \vdots & \vdots \\ M'_{(n/2)(m/2-1)+1,1} & M'_{(n/2)(m/2-1)+1,1} & \dots & M'_{(mn/4),1} \end{bmatrix}, \\ W_{HL} &= \begin{bmatrix} M'_{1,2} & M'_{2,2} & \dots & M'_{n/2,2} \\ M'_{(n/2)+1,2} & M'_{(n/2)+2,2} & \dots & M'_{n,2} \\ \vdots & \vdots & \vdots & \vdots \\ M'_{(n/2)(m/2-1)+1,2} & M'_{(n/2)(m/2-1)+2,2} & \dots & M'_{(mn/4),2} \end{bmatrix}, \\ W_{LH} &= \begin{bmatrix} M'_{1,3} & M'_{2,3} & \dots & M'_{n/2,3} \\ M'_{(n/2)+1,3} & M'_{(n/2)+2,3} & \dots & M'_{n,3} \\ \vdots & \vdots & \vdots & \vdots \\ M'_{(n/2)(m/2-1)+1,3} & M'_{(n/2)(m/2-1)+2,3} & \dots & M'_{(mn/4),3} \end{bmatrix}, \\ W_{HH} &= \begin{bmatrix} M'_{1,4} & M'_{2,4} & \dots & M'_{n/2,4} \\ M'_{(n/2)+1,4} & M'_{(n/2)+2,4} & \dots & M'_{n,4} \\ \vdots & \vdots & \vdots & \vdots \\ M'_{(n/2)(m/2-1)+1,4} & M'_{(n/2)(m/2-1)+2,4} & \dots & M'_{(mn/4),4} \end{bmatrix}. \end{aligned} \tag{8}$$

Assemble these 4 sub-matrices and we obtain an $m \times n$ matrix W , which is the DWT coefficients matrix for original image N .

3. VLSI ARCHITECTURE

Various architectures have been proposed for the implementation of DWT. Some are based on bit-serial architecture [21] while others are based on the systolic algorithm [15]. Applying the segmented-matrix algorithm, the computational process involves only two rows (columns) at a time. Hence, it is very convenient to improve the speed performance by exploiting parallel and/or pipeline methods.

3.1 The Proposed VLSI Architecture

To establish a regular architecture, a basic cell is designed as shown in Fig. 2. It

stores one column of C (denoted as C_i) and its input is one row of M (denoted as M_j). The output for this cell is the multiplication of this input row vector and the stored column vector C_i . Hence, the resulted output is a scalar (an element of M'). Since the 4 columns of C are all different, we need 4 different basic cells as shown in Fig. 3. However, the internal structure of each cell is almost the same except the stored column vectors are different. As a reminder, the internal circuit of these cells is pretty simple because the filter coefficients for Haar DWT are either 1 or -1 . Therefore, although we have mentioned about matrix “multiplication” many times, there is actually no multiplier required at all.

As shown in Fig. 4, we pump the elements of M into the cells at the speed of one row (4 elements) per clock cycle. As a result, 4 DWT coefficients (one row of M') are generated in one clock cycle. Four extra blocks marked as “Shr1” are applied to shift the results 1 bits right (hence the results are divided by 2).

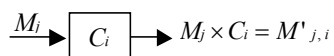


Fig. 2. The basic operational cell.



Fig. 3. The four basic cells array.

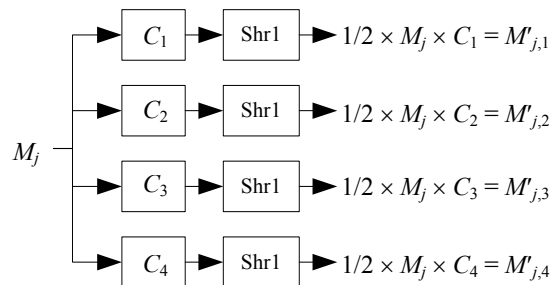


Fig. 4. Proposed VLSI architecture.

As shown in Fig. 4, the proposed architecture is regular and easily to expand. The best part of this parallel architecture is that one can always sacrifice hardware for speed performance. For example, one can achieve twice the speed by using 8 cells concurrently. Moreover, unlike the conventional approach, there is no waiting phase involved since all processes are independent.

3.2 Design Flexibility for System Architecture

One beneficial characteristic of the segmented-matrix algorithm is the Haar DWT coefficients at any sub-band can be obtained independently (no need to wait for other coefficients to be computed). Observing the matrix M' , we find the elements in the first column of M' are exactly the DWT coefficients in LL sub-band (*i.e.* W_{LL}). Similarly, the elements in the 2nd, 3rd and 4th column are the DWT coefficients in HL , LH and HH sub-bands respectively. This feature allows us to do higher-level Haar DWT with more flexibility. For instance, if we need to quickly obtain a rough version of an image, we can at

first apply all the computational resource for computing the DWT coefficients in the LL sub-band and neglect the HL, LH and HH parts temporarily.

By doing such kind of computation adjustments, the first level LL coefficients can be ready for higher level transformation before the computation of 3 detail sub-bands. The segmented-matrix algorithm is effective for implementing k -level 2-D Haar DWT and it takes only one matrix multiplication as well. As a result, a fully scaled multi-resolution system can be implemented based on a simple yet efficient algorithm. Another example to demonstrate the system design flexibility is the embedded system shown in Fig. 5. Two external memory units are used to store image data and computed results. By using two microprocessors such as MCS-8051, we can accomplish the data rearrangements of the segmented-matrix algorithm. On the other hand, the FPGA chip is in charge of the matrix multiplication. When the left microprocessor and RAM unit are dealing with preliminary data rearrangements for next input image (to obtain M from N if in DWT mode), the right microprocessor and RAM unit can deal with the post-arrangements for the computed results. Specifically, due to the new algorithm, software/hardware integrated designs are available for implementing a pipelined image processing system.

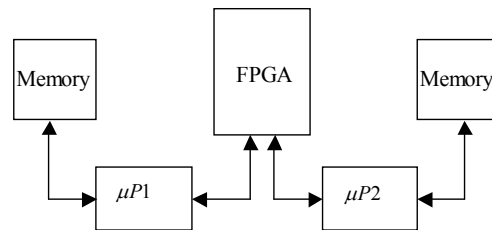


Fig. 5. The full embedded system.

4. IMPLEMENTATION AND ANALYSIS

4.1 Software Implementation

Simulated on an Intel Pentium IV 1.5G Hz CPU with VISUAL C++ codes, the conventional approach consumes 0.109 second to transform a 512×512 image while the segmented-matrix algorithm costs only 0.047 second (same amount of memory is utilized in both codes). Further analyzing the time consumption, we find the data rearrangements cost only about 33% of the total processing time. According to the Amdahl's law, we choose the most time-consuming portion (matrix multiplications) for hardware implementation.

4.2 FPGA Implementation

A Xilinx XC4010XL FPGA chip is applied to implement the proposed VLSI architecture. Verilog HDL is utilized to describe the integrated VLSI architecture and circuit in the basic cells. There are two inputs (8 bits each), two clocks (CLK1 and CLK2) and one control signal for the 4 basic cells in the integrated system. Two non-overlapped

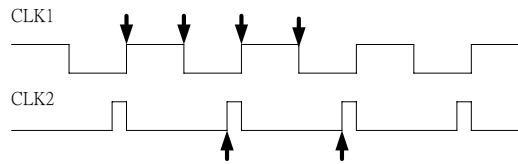


Fig. 6. The two clocks timing.

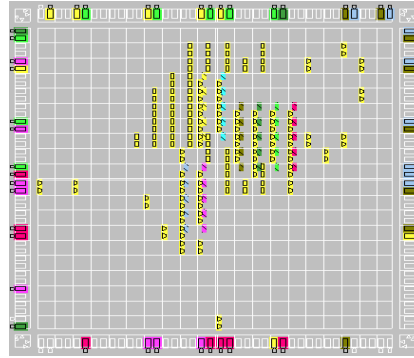


Fig. 7. The placement of the FPGA.

Table 1. FPGA resource usage.

	Used Units	Percentage
CLBs	86	20%
IOBs	51	78%
CLB Flops	64	8%
4 inputs LUTs	99	12%
3 inputs LUTs	32	8%

clocks as shown in Fig. 6 are applied. When the CLK1 is edge-triggered, either on positive or negative edges, the chip reads in two pixels. After one cycle of CLK1, we have 4 pixels stored in the buffers. When CLK2 is positive edge-triggered, the system supplies these 4 pixels to the DWT computational units. The simulation results indicate the integrated architecture functions correctly. After the FPGA implementation, the usage of hardware resource is reported as shown in Table 1 (data rearrangement is supposed to be resolved by software and hence not included). Obviously there is enough hardware resource for implementing more basic cells on the chip. Fig. 7 demonstrates the CLB placement of our design.

The system works at a rate of 37.31 MHz. It takes approximately 0.00526 sec for transforming one 512×512 image (data rearrangement is not included). Based on the new algorithm, we can decompose the original image appropriately and use more computational units to improve the system throughput. However, more expensive series of FPGA with more gates may be required when we need to design a significantly high speed system.

5. CONCLUSIONS

Haar DWT is a powerful tool which provides a wide variety of digital signal and image processing applications. In this paper, we present the segmented-matrix algorithm for Haar DWT and exploit full parallelism to speed up the overall system. A regular and expandable VLSI architecture is proposed and implemented using FPGA chips to verify the effectiveness. Furthermore, with this new scheme, the system design becomes more flexible and the throughput can be improved as much as we need at the expense of extra hardware.

REFERENCES

1. S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, 1989, pp. 674-693.
2. M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Transactions on Image Processing*, Vol. 1, 1992, pp. 205-220.
3. D. Taubman, E. Ordentlich, M. Weinberger, G. Seroussi, I. Ueno, and F. Ono, "Embedded block coding in JPEG2000," in *Proceedings of International Conference on Image Processing*, Vol. 2, 2000, pp. 33-36.
4. C. Christopoulos, J. Askelof, and M. Larsson, "Efficient methods for encoding regions of interest in the upcoming JPEG2000 still image coding standard," *IEEE Signal Processing Letters*, Vol. 7, 2000, pp. 247-249.
5. J. M. Sapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, Vol. 41, 1993, pp. 3445-2462.
6. R. A. DeVore, B. Jawerth, and B. J. Lucier, "Image compression through wavelet transform coding," *IEEE Transactions on Information Theory*, Vol. 38, 1992, pp. 719-746.
7. M. Unser, "Texture classification and segmentation using wavelet frames," *IEEE Transactions on Image Processing*, Vol. 4, 1995, pp. 1549-1560.
8. P. Schroder, "Wavelets in computer graphics," in *Proceedings of the IEEE*, Vol. 84, 1996, pp. 615-625.
9. K. Grochenig and W. R. Madych, "Multiresoultion analysis, Haar bases, and self-similar tilings of R^n ," *IEEE Transaction on Information Theory*, Vol. 38, 1992, pp. 556-568.
10. M. Fujii and J. R. W. Hofer, "Filed-singularity correction in 2-D time-domain Haar-wavelet modeling of waveguide components," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 49, 2001, pp. 685-691.
11. B. J. Falkowski, "Relationship between arithmetic and Haar wavelet transforms in the form of layered Kronecker matrices," *Electronics Letters*, Vol. 35, 1999, pp. 799-800.
12. B. J. Falkowski, "Forward and inverse transformations between haar wavelet and arithmetic functions," *Electronics Letters*, Vol. 34, 1998, pp. 1084-1085.
13. F. Marino, V. Piuri, and E. E. Swartzlander, "A parallel implementation of the 2-D discrete wavelet transform without interprocessor communications," *IEEE Transactions on Signal Processing*, Vol. 47, 1999, pp. 3179-3184.

14. K. Haapala, P. Kolinummi, T. Hamalainen, and J. Saarinen, "Parallel DSP implementation of wavelet transform in image compression," in *Proceedings of IEEE International Symposium on Circuits and Systems*, Vol. 5, 2000, pp. 89-92.
15. T. Acharya and P. Y. Chen, "VLSI implementation of a DWT architecture," *ISCAS IEEE International Symposium on Circuits and Systems*, Vol. 2, 1998, pp. 272-275.
16. F. Marion, "Two fast architectures for the direct 2-D discrete wavelet transform," *IEEE Transactions on Signal Processing*, Vol. 49, 2001, pp. 1248-1259.
17. K. C. Hung, Y. S. Hung, and Y. J. Huang, "A nonseparable VLSI architecture for two-dimensional discrete periodized wavelet transform," *IEEE Transactions on Very Large Scale Integration System*, Vol. 9, 2001, pp. 565-576.
18. M. Nibouche, A. Bouridane, O. Nibouche, D. Crookes, and S. Boussekta, "Design and FPGA implementation of orthonormal discrete wavelet transforms," in *Proceedings of the 7th IEEE International Conference on Electronics, Circuits and Systems*, Vol. 1, 2000, pp. 312-315.
19. M. Nibouche, A. Bouridane, and O. Nibouche, "Design and FPGA implementation of biorthogonal discrete wavelet transforms," in *Proceedings of International Conference on Trends in Communications*, Vol. 1, 2001, pp. 103-106.
20. M. Nibouche, A. Bouridane, and O. Nibouche, "Rapid prototyping of biorthogonal discrete wavelet transforms on FPGAs," in *Proceedings of the 8th IEEE International Conference on Electronics, Circuits and Systems*, Vol. 3, 2001, pp. 1399-1402.
21. F. Marino, "A "double-face" bit-serial architecture for the 1-D discrete wavelet transform," *IEEE Transactions on Circuits and Systems*, Vol. 47, 2000, pp. 65-71.

Po-Yueh Chen (陳伯岳) received his B.S., M.S., and Ph.D. degrees in Electrical Engineering from National Taiwan University in 1988, National Chiao Tung University in 1990 and University of Maryland at College Park in 1997 respectively. Dr. Chen is now a faculty member in Department of Computer Science and Information Engineering, National Changhua University of Education, Changhua, Taiwan. His research interests include image processing, FPGA implementations, VLSI architecture design, and digital IC design.

En-Chi Liao (廖恩吉) received his B.S. and M.S. degrees in Information and Computer Engineering from Chung Yuan Christian University in 1997, and from Chaoyang University of technology in 2003 respectively. Mr. Liao is now an engineer in Realtek Semiconductor Corp., Hsinchu, Taiwan. His major research interests include FPGA implementation, VLSI architecture, and physical design automation.

Chung-Wei Liang (梁忠璋) received his B.S. degree in Electrical Engineering from the Hsiuping Institute of Technology in 2002 and his M.S. degree in Computer Science and Information Engineering from Chaoyang University of Technology in 2004. Currently, Mr. Liang is pursuing the Ph.D. degree in Electrical Engineering at National Taiwan University of Science and Technology, Taipei, Taiwan. His research interests include document image analysis, pattern recognition, and machine vision.