Tiled Interleaving for Multi-Level 2-D Discrete Wavelet Transform

Jung-Wook Kim,
Digital Printing Division
Digital Media Business, Samsung Electronics
Suwon, Korea
jw81.kim@samsung.com

Jinook Song, Seokho Lee, and In-Cheol Park
Department of EECS
Korea Advanced Institute of Science and Technology
Daejeon, Korea
{jsong, shlee, icpark}@ics.kaist.ac.kr

***Abstract*—This paper presents a new architecture of 2-dimensional discrete wavelet transform (DWT) proposed for JPEG 2000. In the proposed architecture, the image is segmented into tiles each of which is sequentially processed to minimize the size of buffers required to process 2D DWT, and multi-level DWTs are interleaved to reduce the size of the repeat buffer drastically. Compared to the conventional architecture, the overall memory size is reduced by 85% and 92% for 256x256 and 512x512 images, respectively. The proposed DWT processor needs only 5kB memory for 256x256 images, and operates at 250MHz in 0.25-μ techology.**

I. INTRODUCTION

As the discrete wavelet transform (DWT) has several advantages such as multi-resolution representation, progressive image transmission, and easy image manipulation, DWT is being increasingly used for image coding to replace discrete cosine transform (DCT). Especially the lifting-based DWT is adopted in the new still image compression standard, JPEG2000. Though JPEG2000 can provide a higher quality image in lower bit-rate than JPEG, large memories and buffers are required to process 2D DWT [1].

There are (5, 3) and (9, 7) filters adopted in JPEG 2000 standard. The (5, 3) filter consists of one predict step and one update step [2], while the (9, 7) filter can be performed by applying the predict and update steps two times. Intermediate values generated by the predict and update steps should be saved for the next step. Basically, 2D DWT is realized by applying 1D DWT two times, row-wise and column-wise processings. In this case, a memory called the middle buffer is required to save the output of the first 1D DWT to be used for the second 1D DWT. Multi-level resolution images can be generated by recursively applying 2D DWT to the output of the low-pass filter. As the low-pass filter output of the previous level should be saved for multi-level resolution, a memory called the repeat buffer is necessary.

If implemented straightforwardly, the sizes of the middle and repeat buffers are proportional to $O(N^2)$, where N is the number of one dimensional pixels of the input image. To reduce the size, a lot of works have been studied. One of the most efficient schemes is the band processing [3] which reduce the middle buffer size to $O(N)$. However, the repeat buffer size is not reduced much. In this paper, a new 2D DWT architecture is proposed to reduce the middle buffer size to $O(1)$ and the repeat buffer size to $O(N)$, drastically reducing the sizes of buffers compared to the previous works. Based on the fast pipeline architecture [4], the proposed 2D DWT processor is designed, which reduces the overall memory size by 85% and 92% for 256x256 and 512x512 images, respectively.

The rest of this paper is organized as follows. The conventional 2D DWT architecture is described in Section II, and the proposed architecture is presented in Section III. In Section IV, the proposed architecture is analyzed compared with the previous implementations. Finally, concluding remarks are made in Section V.

II. CONVENTIONAL 2D DWT ARCHITECTURE

The 1D DWT has been studied a lot, and the most efficient is the lifting-based DWT [5] that breaks up the high pass and low pass filters into a sequence of predict and update steps. However, the 2D DWT is not as simple as the 1D DWT, because the conventional 2D DWT architecture consisting of a row processor (RP) and a column processor (CP) requires a middle buffer and a repeat buffer. Given three row data, the RP generates two transformed outputs, a low-pass filter output and a high-pass filter output. Moving all the low pass filter outputs toward the left side and all the high pass filter outputs toward the right side makes an image being divided into L and H subbands, as shown in Fig. 1(a). The CP processes the outputs of the RP to generate L and H subbands to be placed at the top side and bottom side as depicted in Fig. 1(b). Therefore, the RP needs a memory called the middle buffer to save its results. If the source

1-4244-0921-7/07 $25.00 © 2007 IEEE.

image size is NxN, the middle buffer size is N^2 if implemented straightforwardly.

The multi-level resolution of images can be obtained by recursively applying 2D DWT only to the LL subband as shown in Fig. 1(c). For this purpose, a repeat buffer whose size is a quarter of the image size, $N^2/4$, is needed to save the LL subband of the previous level. As the sizes of the middle and repeat buffers are both proportional to the source image size, large memories are indispensable in straightforward implementations.

To reduce the size of the middle buffer, the band processing has been proposed by Andra in [3], which is to segment the source image into a number of bands in order to process each band sequentially from top to bottom as Fig. 2. In the 2D DWT architecture employing the band processing, the CP can process DWT with 3 data on the same column of 3 adjacent rows. For this reason, the CP can start its DWT processing as soon as the RP fills the middle buffer as many as 3 rows. Whenever the CP starts the processing using the data in the middle buffer, the RP puts new results into the middle buffer. Therefore, the size of the middle buffer can be reduced to 3 line buffers, $3N$. In this case, the CP is associated with an intermediate buffer to save the recently processed intermediate values that are reused later for the next column-wise DWT. There are 3 intermediate values in the (9, 7) filter as indicated in Fig. 3, and the intermediate values are stored for each column. Therefore, the intermediate buffer size is $3N$. Although the additional intermediate buffer is required in the band processing, the total memory size shrinks a lot as the middle buffer size is reduced to $3N$ from N^2.

The overall structure of 2D DWT employing the band processing is shown in Fig. 4. The RP and CP process the data on the same level. For multi-level representation, the repeat buffer must store the whole LL subband of the previous level, meaning that the repeat buffer size is still $N^2/4$ for multi-level representation. Note that the repeat buffer size has to be increased squarely according to the width of the source image, N.

III. Proposed Discrete Wavelet Transform Architecture

As shown in Fig. 5, the proposed DWT architecture consists of a RP, a CP, and buffers, and is similar to the conventional architecture except that the sizes of the middle and repeat buffers are significantly reduced and the RP and CP are equipped with separated overlap buffers and intermediate buffers.

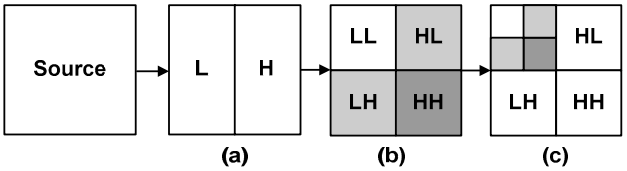

Figure 1. DWT results of (a) RP, (b) CP, and (c) multi-level resolution

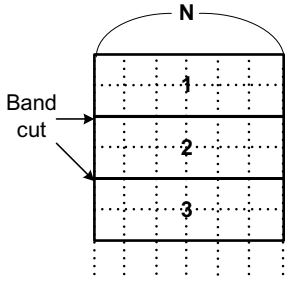

Figure 2. The processing sequence of band processing

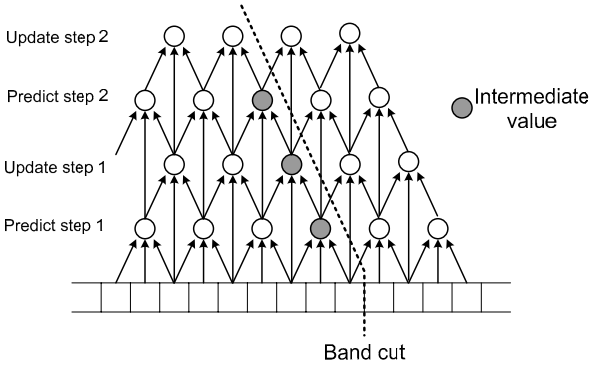

Figure 3. Band processing for the (9, 7) filter

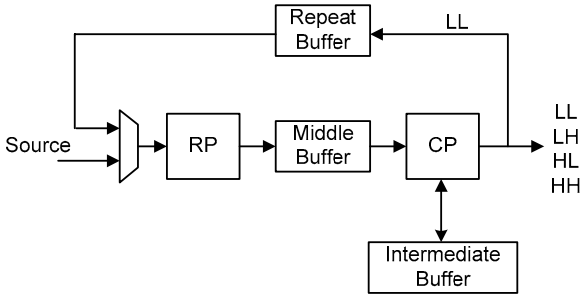

Figure 4. Conventional band processing architecture

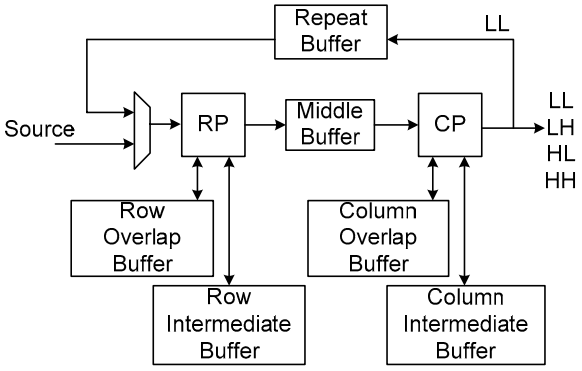

Figure 5. The architecture of the proposed DWT processor

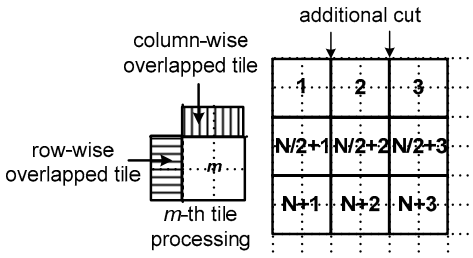

Figure 6. Tile processing using overlapped tiles

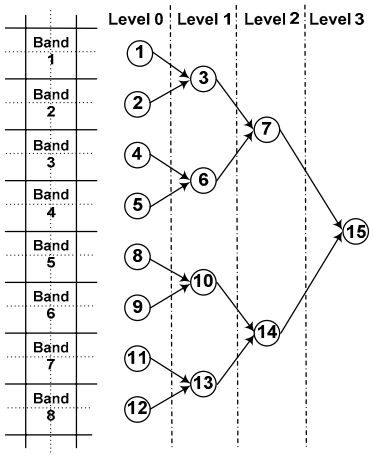

Figure 7. Processing sequence for multi-level resolution

A. *Reduction of the Middle Buffer*

The band processing [3] reduces the middle buffer size to *3N*. To further reduce the middle buffer size, additional cuts orthogonal to the band cuts are inserted as shown in Fig. 6, which renders the image being segmented into a set of rectangular tiles each size of which is *2 X 2*. For each tile, the RP performs for the partial rows and then the CP starts processing for the partial column data resulting from the RP. If the RP reaches a band cut, it will move to the next adjacent tile. Thus, the size of the middle buffer is reduced to *2 X 2*, instead of *3N* of the conventional middle buffer.

As 3 column data on the same row are needed for a row DWT processing, the RP stores the overlapped data of the previously adjacent tile into the row overlap buffer, and it processes 2 column data coming from the current tile together with the overlapped data stored previously. The intermediate values to be used for the next tile are saved into the row intermediate buffer. Therefore, the middle buffer size is reduced to constant at the cost of the additional intermediate buffer and row overlap buffer. Since the sizes of the intermediate and row overlap buffers are also proportional to 2, the additional cost is far less than the reduction of the middle buffer.

The tile processing order is depicted in Fig. 6. The tiles in a row band are processed one by one from left to right, and the next row band is processed in the same direction.

B. *Reduction of the Repeat Buffer*

The repeat buffer is indispensable in multi-level 2D DWT, but it can be removed for 1D DWT as presented in Liao's architecture [6]. Conceptually, the repeat buffer holds a quarter of the image size, that is, the size is $N^2/4$. If we start the higher level processing only when the lower level processing is completed as in the conventional approach, it is impossible to reduce the size. However, in the proposed multi-level processing, several 2D DWT processings each of which corresponds to a level are interleaved to reduce the repeat buffer size. When a lower level 2D DWT generates a band of the LL subband, namely two rows of the LL band, the higher level processing can start as in the band processing. Therefore the size of the repeat buffer can be reduced to 2 line buffers for each level, instead of a quarter of the lower level image size.

If the line size of level 0 is *N*, the line sizes of level 1, level 2 and level 3 are *N/2, N/4* and *N/8*, respectively, as the size of the LL band is reduced to half in each level. Thus, the total size of the repeat buffers can be reduced to *2N* (1).

$$2N = 2 \times (\frac{N}{2} + \frac{N}{4} + \frac{N}{8} + \frac{N}{16} + ...) \quad (1)$$

This reduction is significant when compared to the tremendous size of the previous repeat buffer, $N^2/4$.

Fig. 7 shows how to interleave the band processings of multiple levels. If two row bands for level 0 are completed, we get a row band of level 1, namely two row lines, which is enough to start the row band processing for level 1. Similarly, two row band processings for level 1 generate a row band of level 2. This relation is shown in Fig. 7 where the circled number represents the processing order of row bands each of which is belonging to a level. As indicated in the order, the multiple levels are processed in an interleaved manner.

To support the interleaved multi-level processing, the RP and the CP should be associated with multiple intermediate and overlap buffers, one for each level. For the RP, the size of the overlap buffer is 2 for each level, which is independent of the level. The number of intermediate values of the (5,3) filter to be saved into the intermediate buffer is 1 per row, and that of the (9,7) filter is 3 per row. As there are two rows in a tile, the intermediate buffer size is 6 in total for each level. Therefore the buffer size can be defined as follows.

Row overlap buffer size = 2 X (no of levels)

Row intermediate buffer size = 6 X (no of levels)

For the CP, the overlap buffer size for a level is equal to the image size of the level. Hence, the total overlap buffer size is $N + N/2 + N/4 + N/8 + \ldots = 2N$.

The column intermediate buffer size is also proportional to the image size of the level. As the intermediate buffer size needed for a column is 3, the total intermediate buffer size is 6N considering multi-level processing.

IV. Implementation and Experimental Results

In the proposed DWT processor, the CP and RP consist of 2 processing elements (PE) that are modified from the fast pipeline architecture [4] to include a separated intermediate buffer for each level. Table I compares the buffer size of the proposed architecture supporting 5-level resolution and that of the previous works. The proposed one remarkably reduces the buffer size, leading to 85 % and 92 % reduction in case of 256x256 and 512x512 images, respectively. As every coefficient of the (9, 7) filter has fraction, the precision analysis is performed to decide how many bits are sufficient to represent the fraction. Table II shows the precision analysis of the (9, 7) filter performed with varying the number of fractional bits. The results show that 4-bit fraction and 12-bit multiplication produce reasonable image quality.

The proposed DWT processor was designed in Verilog HDL and synthesized in 0.25um technology. The layout is shown in Fig. 8, and the implementation details of the proposed DWT processor are summarized in Table III. The operating frequency and logic area excluding memory are 250MHz and 43765 gates, respectively. The total memory size is 5kB for 256x256 images. The proposed processor has higher operating frequency and smaller buffer than [3] that operates at 200MHz in 0.18um technology.

TABLE I. Comparison of Buffer Sizes

	C-J Lian [7]	*K. Andra [3]*	*K-C.B Tan [8]*	**Proposed**
Middle buffer	N^2	4N	3N	4
Intermediate buffer	No	1N		6N for CP 30 for RP
Overlap buffer	No	No	No	2N for CP 2 for RP
Repeat buffer	$N^2/4$	$N^2/4$	$N^2/4$	2N
Total	$5\ N^2/4$	$N^2/4 + 5N$	$N^2/4 + 3N$	10N + 36
256 x 256 tile	160kB	34.5kB	33.5kB	5kB
512 x 512 tile	640kB	133kB	132kB	10kB

TABLE II. PSNR Analysis of the (9,7) Filter for Various Fractional Bits (unit: dB)

Images	**Fraction Bits**						
	1	*2*	*3*	*4*	*5*	*6*	*7*
Lena 512	28.65	34.46	40.24	45.20	48.58	49.64	48.89
Lena 256	28.71	34.17	39.58	45.05	48.59	49.53	48.84
Barbara 512	28.63	34.31	40.33	45.16	48.60	49.71	48.98
Barboon 256	28.28	34.46	40.16	45.15	48.29	49.58	49.02
Bridge 256	28.51	34.35	40.19	44.92	48.89	49.59	49.10
Pepper 256	28.57	34.58	40.07	45.02	48.69	49.74	48.97

TABLE III. Characteristics of the Proposed DWT Processor

Technology	0.25um CMOS
Operating Frequency	250 MHz
Logic gates	43765 gates
Memory	5kB
Tile Resolution	256 x 256

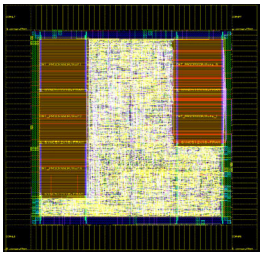

Figure 8. Layout of the proposed DWT processor

V. Conclusion

This paper has proposed an area-efficient lifting-based DWT architecture, which minimizes the buffer sizes by reducing the repeat buffer, making the middle buffer independent of image size. To achieve the reduction, the input image is segmented into tiles each of which is processed sequentially, and the multiple levels are processed in an interleaved fashion. Compared to the conventional architecture, the total memory size is reduced by 85% and 92% for 256x256 and 512x512 images, respectively. The total memory size is 5kB for 256x256 images, which is much smaller than that of the conventional DWT architecture.

Acknowledgment

This work was supported in part by Institute of Information Technology Assessment through the ITRC.

References

[1] Liang-Gee Cha, Chung-Jr Lian, Kuan-Fu Chen and Hong-Hui Chen, "Analysis and Architecture Design of JPEG2000," IEEE Int. Conf. On Multimedia and Expo, 2001.

[2] C.C. Liu, Y.H. Shiau, and J.M. Jou, "Design and Implementation of a Progressive Image coding chip based on the Lifted Wavelet Transform," Proc. 11th VLSI Design/CAD Symposium, Taiwa, 2000.

[3] Kishore Andra, Chaitali Chakrabarti and Tinku Acharya, "A VLSI Architecture for Lifting-Based Forward and Inverse Wavelet Transform," IEEE Trans. Signal Processing, vol. 50, no. 4, pp. 966–977, April 2002.

[4] Bing-Fei Wu and Chung-Fu Lin, " A Rescheduling and Fast Pipeline VLSI Architecture for Lifting-based Discrete Wavelet Transform," IEEE Int. Symp. On Circuits and Systems, vol. 2, pp. II-732–II-735, May 2003.

[5] Wim Sweldens, "The Lifting Scheme : A New Philosophy in Biorthogonal Wavelet Constructions," Proc. SPIE, vol. 2569, pp. 68–79, 1995

[6] Hongyu Liao, Mrinal K. Mandal, Bruce F. Cockburn, " Novel architectures for the lifting-based discrete wavelet transform," IEEE Canadian Cof. On Electrical and Computer Engineering, vol. 2, pp. 1020–1025, May 2002.

[7] Chung-Jr Lian, Kuan-Fu Chen, Hong-Hui Chen, and Liang-Gee Chen, "Analysis and Architecture Design of Lifting Based DWT and EBCOT for JPEG 2000," IEEE Int. Symp. On VLSI Technology, Systems and Applications, pp. 180–183, April 2001.

[8] K-C. B. Tan and T. Arslan, "Shift-accumlator ALU centric JPEG2000 5/3 lifting based discrete wavelet transform architecture," IEEE Int. Symp. On Circuits and Systems, vol. 5, pp. V-161–V-164, May 2003.