EFFICIENT VLSI ARCHITECTURE FOR 2-D INVERSE DISCRETE WAVELET TRANSFORMS

Chu Yu and *Sao-Jie Chen*

Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan, R.O.C.

ABSTRACT

In this paper, we present a high-performance VLSI architecture for 2-D inverse discrete wavelet transforms (IDWT). The architecture is designed based on a computation-schedule scheme to process the input signals in real-time, and uses two efficient filter structures to minimize the hardware cost. For the computation of an $N \times N$ 2-D image with a filter length L, this architecture spends near N^2 clock cycles, and requires about NL storage unit, $3\frac{1}{2}L$ multipliers, as well as $7(\frac{L}{2}-1)+4$ adders.

1. INTRODUCTION

In recent years, there has been a number of studies on wavelet transforms for signal analysis and synthesis [1]-[3]. In the applications of 2-D discrete wavelet transforms (DWT) to image [4]-[7] and video [8]-[9] processing, the inverse DWT (IDWT) is essentially used to reconstruct the original signals and the forward DWT to decompose the input signals. Similar to 2-D inverse transforms, a 2-D IDWT needs the same large amount of computation; in order to meet the requirements of fast computation in real-time applications, dedicated hardware realizations are required.

Several VLSI architectures [10]-[12] have been proposed for 2-D IDWT's. For instance, Lewis and Knowles [10] firstly proposed multiplierless 2-D forward and inverse architectures for the 4-tap Daubechies wavelet transform. Rumian [11] presented a practical implementation of 2-D wavelet decomposition based on a pair of 1-D QMF FIR filters. Chakrabarti and Mumford [12] devised some 2-D DWT folded architectures and scheduling algorithms for the analysis and synthesis filters.

This work was partly supported by the National Science Council, ROC, under Grant NSC-88-2215-E002-037.

2. 2-D IDWT ARCHITECTURES

2.1 Preliminaries

In order to reconstruct the original signal, the 2-D IDWT has to synthesize the signals from the coarsest resolution level to the finest one. For example, a 3-level 2-D IDWT firstly generates a x^2_{HH} band by upsampling and filtering four input-signal bands, x^3_{HH} (low-low), x^3_{HG} (low-high), x^3_{GH} (high-low), and x^3_{GG} (high-high), in a separable 2-D manner as shown in Fig. 1, where x^m_{xx} represent external input samples for the xx band at the m-th level. Then, the foregoing procedure will be iterated to construct the next finer level of HH synthesis band, until the original signal is restructured.

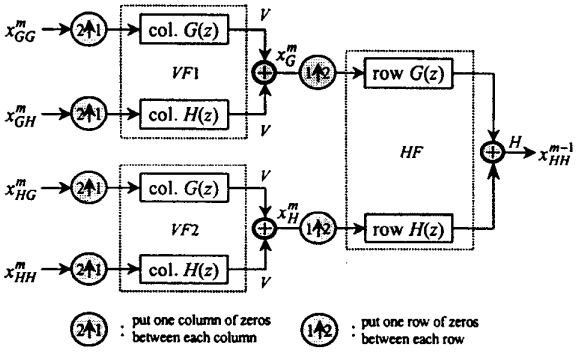


Fig. 1 One level of separable 2-D inverse DWT.

2.2 Proposed Architecture

Our proposed 2-D IDWT architecture is shown in Fig. 2. The architecture consists mainly of two vertical filters (VF), one horizontal filter (HF), a small-size temporary buffer, and two register-banks. Assume that the horizontal

0-7803-5471-0/99/$10.00©1999 IEEE

(row-major) filtering was performed first and then the vertical (column-major) filtering by a 2-D forward DWT to generate the decomposed signals. Given these signals as input, an inverse architecture has to perform the vertical filtering first and then the horizontal filtering for restoring the decomposed signals to the original ones.

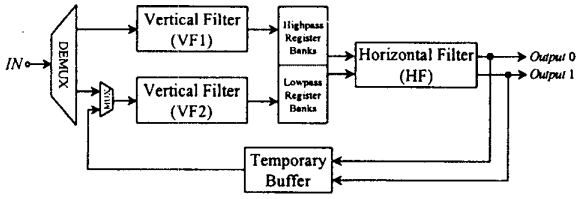


Fig. 2 Proposed architecture for 2-D inverse DWT.

A three-level 2-D IDWT computation-schedule scheme for the proposed architecture is shown in Fig. 3. The amount of X's marked on each band in the figure denotes the number of IDWT computations needed in one pass with a sequence of 3-2-1111-2-1111-2-1111-2-1111, where each number n represents the computation at the n-th resolution level. Thus, the total number of passes needed is equal to one quarter of the amount of IDWT computations at the coarsest level.

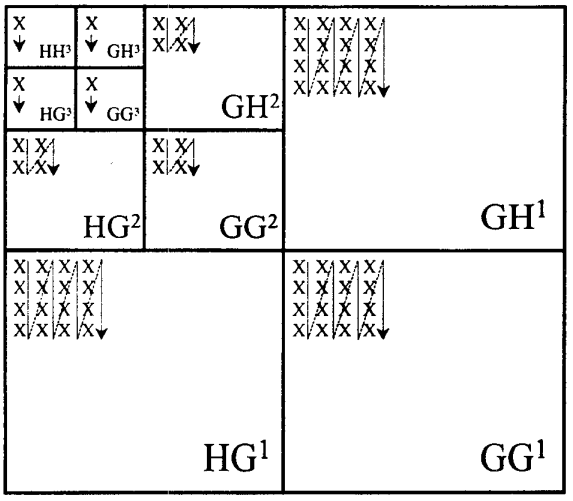


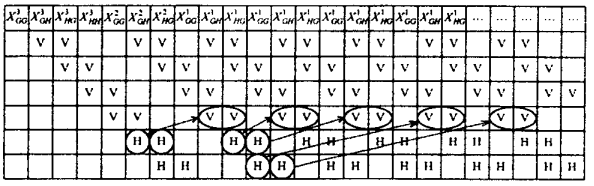
Fig. 3 Diagram a three-level computation-schedule scheme.

At each level of the IDWT computation, we need four points of input-band samples to be fed into this architecture. Then, the architecture synthesizes these input samples into four output samples, which will be stored in the temporary buffer and served as the input data for the next finer-level computation through a multiplexer. For example, at the third-level computation, we synthesize the four input samples (HH^3, HG^3, GH^3, GG^3) into four points of HH^2. Except for the coarsest level (which uses four external input samples), we need three external samples and one internal sample that comes from a previous-level computation (the coarser level) to synthesize a finer level of output samples.

According to the above discussion, the dataflow of the whole architecture is shown in Table I, where V and H denote output samples generated by the vertical and the horizontal filters, respectively. From the table, the dash-line with arrow exhibits a relation that the horizontal output sample (H) of the previous level will become the internal input of the current level to generate the two vertical output samples (V).

TABLE I

DATAFLOW OF THE PROPOSED ARCHITECTURE



In the following, we will describe the main components of our proposed architecture in more details.

2.3 Filter Structure

For simplicity, we show only four taps of filter coefficients for all the filters in our proposed architecture in the following discussion. Moreover, since the structures of the vertical filters VF1 and VF2 are the same, we only describe VF1 in this paper.

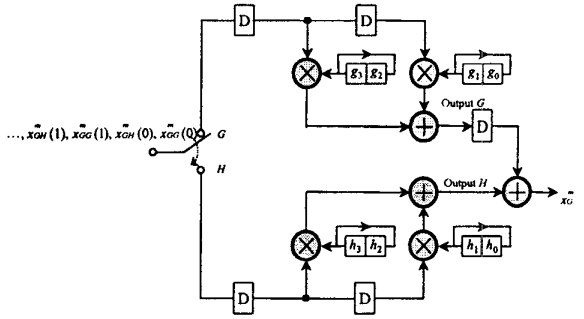


Fig. 4 Vertical filter (VF1).

The vertical filter VF1 of our IDWT architecture is shown in Fig. 4. The filter contains a pair of interpolation FIR filters, which is composed of a lowpass and a highpass

filters. The input sequence of this filter takes a highpass-band sample followed by a lowpass-band sample, and this sequence is repeated until the end of input data.

The dataflow of the vertical filter is shown in Table II, where outputs G and H are a highpass and a lowpass filtering outputs corresponding to the pairs of V's in Table I, respectively. Since output H delays one more clock cycle than output G, a delay element is added behind output G to generate the accurate synthesis samples. Clearly, the proposed vertical filter structure is efficient because it requires approximately half the hardware cost than the corresponding direct-form structures.

TABLE II

DATAFLOW OF VERTICAL FILTER

Time	*I/P*	*Output G*	*I/P*	*Output H*	*O/P*
1	$x_{GG}^m(0)$	0		0	0
2		$g_3x_{GG}^m(0)$	$x_{GH}^m(0)$	0	0
3	$x_{GG}^m(1)$	$g_2x_{GG}^m(0)$		$h_3x_{GH}^m(0)$	$x_G^m(0)$
4		$g_3x_{GG}^m(1)+g_1x_{GG}^m(0)$	$x_{GH}^m(1)$	$h_2x_{GH}^m(0)$	$x_G^m(1)$
5	$x_{GG}^m(2)$	$g_2x_{GG}^m(1)+g_0x_{GG}^m(0)$		$h_3x_{GH}^m(1)+h_1x_{GH}^m(0)$	$x_G^m(2)$
6		$g_3x_{GG}^m(2)+g_1x_{GG}^m(1)$	$x_{GH}^m(2)$	$h_2x_{GH}^m(1)+h_0x_{GH}^m(0)$	$x_G^m(3)$
7	$x_{GG}^m(3)$	$g_2x_{GG}^m(2)+g_0x_{GG}^m(2)$		$h_3x_{GH}^m(2)+h_1x_{GH}^m(1)$	$x_G^m(4)$
8		$g_3x_{GG}^m(3)+g_1x_{GG}^m(2)$	$x_{GH}^m(3)$	$h_2x_{GH}^m(2)+h_0x_{GH}^m(2)$	$x_G^m(5)$
9	⋮	$g_2x_{GG}^m(3)+g_0x_{GG}^m(2)$		$h_3x_{GH}^m(3)+h_1x_{GH}^m(2)$	$x_G^m(6)$
10		⋮	⋮	$h_2x_{GH}^m(3)+h_0x_{GH}^m(2)$	$x_G^m(7)$
⋮	⋮	⋮		⋮	⋮

Analogous to the case of vertical filter VF, the horizontal filter HF contains also a pair of interpolation FIR lowpass and highpass filters. But, since the output of the VF generates two samples for each input sample, the amount of input data to process in the HF is twofold to the input data of VF. This means that we need a pair of HF filters for filtering the input data concurrently to avoid losing any input data. Thus, in order to reduce the hardware cost, we devise another more efficient filter structure, which consists mainly of three sub-filters each enclosed in a dash-box, as shown in Fig. 5.

By correctly turning on the switches in Fig. 5, the HF will generate exactly the wanted output data. The dataflow of this HF is shown in Table III, where input samples $x_H^m(1)$ and $x_H^m(2)$ come from the VF1 via the lowpass register-bank, and samples $x_G^m(1)$ and $x_G^m(2)$ come from the VF2 via the highpass register-bank. After input $x_H^m(1)$, the sub-filter ❸ in Fig. 5 generates only two output samples $H11$ and $H12$. However, other output samples ($H21$, $G11$, $G21$) and ($H22$, $G12$, $G22$) are generated by the sub-filters ❶ and ❷, respectively. Moreover, in Table III, the dash-lines with arrow indicate which two filtering samples will be summated to obtain the accurate outputs, i.e., x_{HH}^{m-1} which are equivalent to H in Table I.

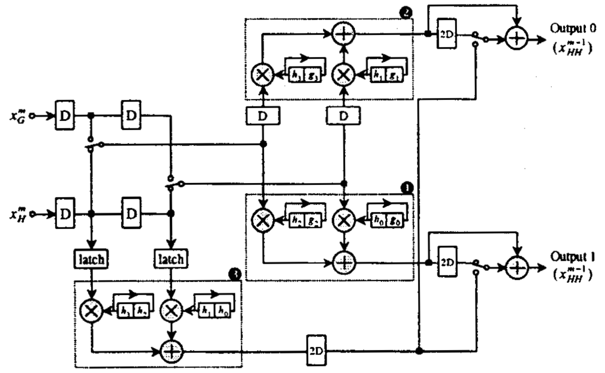


Fig. 5 Horizontal filter.

TABLE III

DATAFLOW OF HORIZONTAL FILTER

sub-filters \ data	$x_H^m(1)$	$x_H^m(2)$	$x_G^m(1)$	$x_G^m(2)$	$x_H^m(1)$	$x_H^m(2)$	$x_G^m(1)$	$x_G^m(2)$ $x_H^m(1)$	$x_H^m(2)$	$x_G^m(1)$	$x_G^m(2)$ $x_H^m(1)$	$x_H^m(2)$	$x_G^m(1)$	$x_G^m(2)$	...	...
❶			H21	G11	G21		H21	G11	G21	H21	G11	G21	H21	G11	G21	
❷				H22	G12	G22		H22	G12	G22	H22	G12	G22	H22	G12	G22
❸		H11	H12			H11	H12		H11	H12		H11	H12			

2.4 Storage Unit

In order to deal with all the resolution levels of 2-D IDWT's, the proposed architecture needs a temporary buffer, which size is $4(m-1)$ for m levels. In addition, the architecture uses a highpass and a lowpass register banks in a separable 2-D filtering manner. The total size of these two register banks is approximately NL for processing an $N{\times}N$ 2-D image with L filter length.

3. PERFORMANCE EVALUATION

The performance data of two 2-D IDWT architectures is summarized in Table IV. For the computation of an $N{\times}N$ 2-D image with a filter length L, a direct implementation needs about $4N^2$ clock cycles to compute all the levels of 2-D IDWT's, and requires $2L$ multiplier-and-accumulator cells (MAC's) and at least N^2 storage unit. On the other hand, our proposed architecture spends near N^2 clock cycles, and requires about NL storage unit, $3\frac{1}{2}L$ multipliers, as well as $7(\frac{L}{2}-1)+4$ adders.

In addition, the function of the proposed architecture has correctly been verified by Verilog logic-level simulation.

TABLE IV

COMPARISONS OF TWO 2-D IDWT ARCHITECTURES

Architectures	Direct Approach	Ours
MAC's	$2L$	-
Multipliers	-	$3\frac{1}{2}L$
Adders	-	$7(\frac{L}{2}-1)+4$
Registers	N^2	NL
Period	$4N^2$	N^2

4. CONCLUSION

Realization of a high-performance VLSI architecture for 2-D IDWT that favors the single-chip VLSI implementation has been described in this paper. The architecture utilized efficient filter structures and a computation-schedule scheme to accomplish the computations in all resolution levels. Since this architecture has a low latency, a low hardware cost, and ability to process 2-D digital signals in real-time, it can be applied very well to many real-time video/image applications, such as MPEG-4 and JPEG-2000.

5. REFERENCES

[1] 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, July 1989.

[2] O. Rioul and M. Vetterli, "Wavelets and signal processing", *IEEE Signal Processing Magazine*, vol. 8, no.4, pp. 14-38, Oct. 1991.

[3] I. Daubechies, *Ten Lectures on Wavelets, vol. 61 of CBMS-NSF Regional Conferences Series in Applied Mathematics*, SIAM, Philadelphia, PA, 1992.

[4] S. Mallat, "Multifrequency channel decompositions of images and wavelet models", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 12, pp. 2091-2110, Dec. 1989.

[5] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform", *IEEE Trans. Image Processing*, vol. 1, no. 2, pp. 205-220, Apr. 1992.

[6] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients", *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3445-3462, Dec. 1993.

[7] A. Averbuch, D. Lazar, and M. Israeli, "Image compression using wavelet transform and multiresolution decomposition", *IEEE Trans. Image Processing*, vol. 5, no. 1, pp. 4-15, Jan. 1996.

[8] A. S. Lewis and G. Knowles, "Video compression using 3-D wavelet transforms", *Electron. Lett.*, vol. 26, no. 6, pp. 396-398, Mar. 1990.

[9] K. H. Goh, J. J. Soraghan, and T. S. Durrani, "New 3-D wavelet transform coding algorithm for image sequences", *Electron. Lett.*, vol. 29, no. 4, pp. 401-402, Feb. 1993.

[10] A. S. Lewis and G. Knowles, "VLSI architecture for 2-D daubechies wavelet transform without multipliers", *Electron. Lett.*, vol. 27, no. 2, pp. 171-173, Jan. 1991.

[11] R. Rumian, "An architecture for real-time wavelet image decomposition", in *Proc. IEEE Int. Symp. on Circuits and Systems*, London, England, May 1994, pp. 73-76.

[12] C. Chakrabarti, C. Mumford, "Efficient realizations of analysis and synthesis filters based on the 2-D discrete wavelet transform", in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, 1996, pp. 3256-3259.