Systolic VLSI Architectures for 1-D Discrete Wavelet Transforms

Tracy C. Denk
Broadcom Corporation
Irvine, CA 92618
tdenk@broadcom.com

Keshab K. Parhi
Department of ECE
University of Minnesota
Minneapolis, MN 55455
parhi@ece.umn.edu

Abstract

This paper presents systolic VLSI architectures for the discrete wavelet transform (DWT) and inverse discrete wavelet transform (IDWT) which operate on one-dimensional signals. Recently, a dependence graph (DG) of the DWT has been presented which enables systolic mapping techniques to be used to derive DWT architectures. We use this DG to systematically derive new DWT architectures. In addition, we present a DG for the IDWT and use it to systematically derive new IDWT architectures. The resulting DWT and IDWT architectures are scalable with filter length and number of octaves, modular, have high hardware utilization, and use fixed-coefficient multipliers. These properties make them well-suited for VLSI implementation.

1. Introduction

Wavelet transforms have proven to be useful tools for signal processing [1, 5, 7]. Applications of wavelet transforms include signal compression, statistical analysis, adaptive filtering, and solving linear equations.

The one-dimensional (1-D) discrete wavelet transform (DWT) recursively decomposes the input signal, $S_0(n)$, into approximation and detail at the next lower resolution. Let $S_i(n)$ and $W_i(n)$ be the approximation and detail, respectively, of the signal at level i. The approximation of the signal at level $i+1$ is computed using

$$S_{i+1}(n) = \sum_{k=0}^{L-1} h(k) S_i(2n-k), \qquad (1)$$

and the detail of the signal at level $i+1$ is computed using

$$W_{i+1}(n) = \sum_{k=0}^{L-1} g(k) S_i(2n-k). \qquad (2)$$

The equations (1) and (2) describe the computation of the DWT. This technique for computing the DWT is often referred to as the pyramid algorithm or Mallat's algorithm [5].

The input signal $S_0(n)$ can be synthesized from its wavelet coefficients using the inverse DWT (IDWT) equation

$$S_{i-1}(n) = \sum_k \{S_i(k)h'(n-2k) + W_i(k)g'(n-2k)\}. \qquad (3)$$

The wavelet filters $G(z)$, $H(z)$, $G'(z)$ and $H'(z)$ used in (1), (2), and (3) are assumed to be 4-tap FIR filters throughout this paper.

2. Systolic DWT Architectures

Several VLSI architectures for the 1-D DWT have been proposed in the past [4, 6], including systolic architectures [2, 8, 3]. In this section, systolic architecture design is described and then used to design systolic architectures for the DWT and IDWT.

2.1. Systolic Mapping Techniques

Systolic architecture design requires a space representation of the algorithm, which is also called a dependence graph (DG). For example, consider the convolution equation

$$y_n = w_0 x_n + w_1 x_{n-1} + w_2 x_{n-2} + w_3 x_{n-3} \qquad (4)$$

of a 4-tap FIR filter. A DG for this algorithm is shown in Figure 1(a). Each node in the DG represents a multiply-accumulate operation, and each node has an index in the ij-plane.

The DG is mapped to a systolic architecture using a processor space vector $\mathbf{p} = \begin{bmatrix} p_1 & p_2 \end{bmatrix}^T$ and a schedule vector $\mathbf{s} = \begin{bmatrix} s_1 & s_2 \end{bmatrix}^T$. An edge $\mathbf{e}$ in the DG leads to

0-7803-5148-7/98/$10.00©1998 IEEE

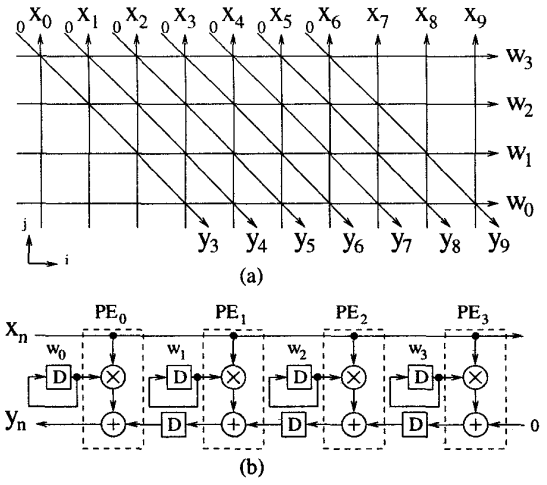

Figure 1. Systolic mapping of the FIR filter described in (4). (a) The dependence graph. (b) The systolic architecture.

Table 1. Systolic mapping of Figure 1(a).

Edge Description	Edge Vector $\mathbf{e}$	$\mathbf{p}^T\mathbf{e}$	$\mathbf{s}^T\mathbf{e}$
Inputs	$[\ 0\ \ 1\ ]^T$	1	0
Coefficients	$[\ 1\ \ 0\ ]^T$	0	1
Outputs	$[\ 1\ \ -1\ ]^T$	-1	1

an interconnection in the systolic architecture in the direction $\mathbf{p}^T\mathbf{e}$ with $\mathbf{s}^T\mathbf{e}$ delay elements.

To demonstrate systolic architecture design, the processor space vector $\mathbf{p} = [\ 0\ \ 1\ ]^T$ and the schedule vector $\mathbf{s} = [\ 1\ \ 0\ ]^T$ are applied to the DG in Figure 1(a). Table 1 shows the values of $\mathbf{p}^T\mathbf{e}$ and $\mathbf{s}^T\mathbf{e}$ for all edges in the DG. This table shows that the inputs move from PE_i to PE_{i+1} with zero sample delays, the outputs move from PE_i to PE_{i-1} with one sample delay, and the filter coefficients move from PE_i to PE_i with one sample delay. The systolic architecture is shown in Figure 1(b).

2.2. DWT Architectures

The dependence graph for the computation of $S_1(n)$ and $W_1(n)$, according to (1) and (2) for $i = 0$, is shown in Figure 2(a) [2]. The output quantities in brackets, $[S_1(n)]$, are computed using the filter coefficients in brackets, $[h_k]$, and the output quantities in parentheses, $(W_1(n))$, are computed using the filter coefficients in parentheses, (g_k). Each node in the dependence graph represents two multiply-accumulate (MAC) operations. As an example, both $S_1(2)$ and $W_1(2)$ are computed along the vertical line corresponding to $n = 4$. Examining the signals that intersect along this line reveals $S_1(2) = h(0)S_0(4) + h(1)S_0(3) + h(2)S_0(2) + h(3)S_0(1)$ and $W_1(2) = g(0)S_0(4) + g(1)S_0(3) + g(2)S_0(2) + g(3)S_0(1)$, which correspond to (1) and (2), respectively, for $i = 0$ and $n = 2$. Figure 2(b) represents (1) and (2) for $i = 1$, and Figure 2(c) represents (1) and (2) for $i = 2$.

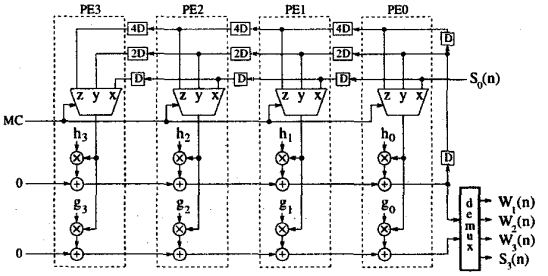

Figure 3. The three-level DWT architecture A1. The mux control (MC) signal repeats the sequence $x, y, x, z, x, y, x, \phi$ starting at time $= 0$, where ϕ is denotes "don't care."

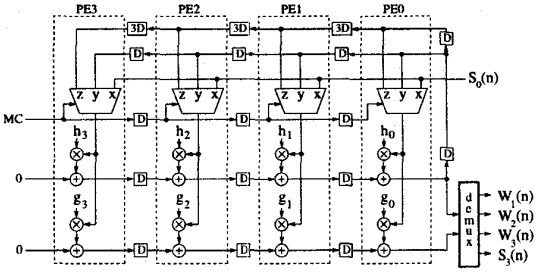

Figure 4. The three-level DWT architecture A2. The mux control (MC) signal repeats the sequence $x, y, x, z, x, y, x, \phi$.

The dependence graph for the three-octave DWT is formed by superimposing parts (a), (b), and (c) of Figure 2. In this superimposed dependence graph, octave 1 outputs are computed along $n = 2l$, octave 2 outputs are computed along $n = 4l + 1$, and octave 3 outputs are computed along $n = 8l + 3$. This implies that only two MAC operations are performed at each node in the superimposed dependence graph.

The edges of the superposition of the three dependence graphs in Figure 2 are described and listed in the

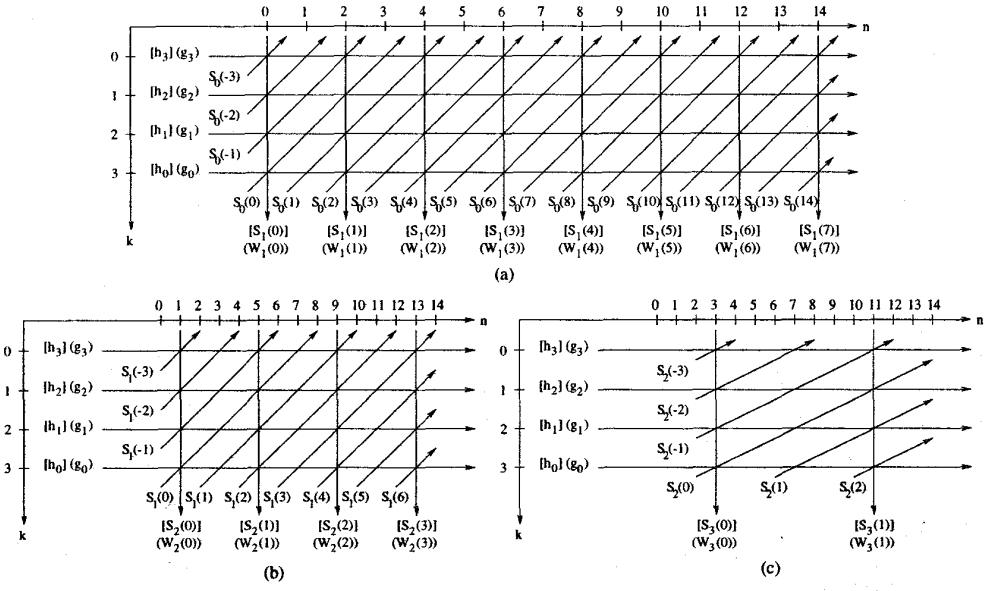

Figure 2. The dependence graph for a 1-D DWT broken down into (a) octave 1, (b) octave 2, (c) octave 3. The three-level DWT dependence graph is the superposition of (a), (b), and (c).

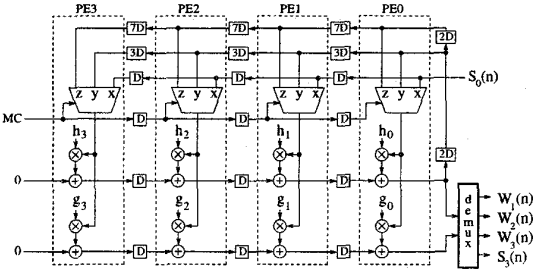

Figure 5. The fully systolic three-level DWT architecture A3. The mux control (MC) signal repeats the sequence $x, \phi, y, \phi, x, \phi, z, \phi, x, \phi, y, \phi, x, \phi, \phi, \phi$.

first two columns of Table 2. The inputs to octave i correspond to the values $S_{i-1}(n)$, which can be found in parts (a), (b), and (c) of Figure 2 for $i = 1$, 2, and 3, respectively. The direction of the weights (h_k and g_k) and the direction of the outputs are the same for all octaves. The final two entries in the table are required because the inputs to octave 2 are also the outputs of octave 1, and the inputs to octave 3 are the outputs of octave 2. For example, the output $S_1(2)$ is available at $(n, k) = (4, 3)$ in Figure 2(a), and this sample is first used at $(n, k) = (5, 3)$ in Figure 2(b), so the direction of this vector is $\begin{bmatrix} 5 & 3 \end{bmatrix}^T - \begin{bmatrix} 4 & 3 \end{bmatrix}^T = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$. This corresponds to the entry "Out 1 $\rightarrow$ In 2" in the table. Similarly, the inputs to octave 3, which are also the outputs of octave 2, correspond to the entry "Out 2 $\rightarrow$ In 3" in the table.

Based on the dependence graph edges in Figure 2, three systolic architectures, A1, A2, and A3, are developed for the DWT. Architecture A1 uses $\mathbf{p}_1^T = \begin{bmatrix} 0 & 1 \end{bmatrix}$ and $\mathbf{s}_1^T = \begin{bmatrix} 1 & 0 \end{bmatrix}$, architecture A2 uses $\mathbf{p}_2^T = \begin{bmatrix} 0 & 1 \end{bmatrix}$ and $\mathbf{s}_2^T = \begin{bmatrix} 1 & 1 \end{bmatrix}$, and architecture A3 uses $\mathbf{p}_3^T = \begin{bmatrix} 0 & 1 \end{bmatrix}$ and $\mathbf{s}_3^T = \begin{bmatrix} 2 & 1 \end{bmatrix}$ to map the DWT DG to the systolic architectures in Figures 3, 4 and 5, respectively. The mux control signal is determined from the knowledge that the computation at the location $\mathbf{I} = \begin{bmatrix} i & j \end{bmatrix}^T$ in the DG is executed by the processing element $\mathbf{p}^T\mathbf{I}$ at the time instance $\mathbf{s}^T\mathbf{I}$ in the architecture.

2.3. IDWT Architectures

Figure 6 is the DG for a three-level IDWT. Table 3 shows how the edges of the DG map to the architectures S1, S2, and S3 using the same vectors $\mathbf{p}_i^T$ and $\mathbf{s}_i^T$, $i = 1, 2, 3$, as used for the DWT architectures in Section 2.2. Architectures S1 and S2 are shown in Figures 7 and 8, respectively, and architecture S3 is omitted due to lack of space.

Table 2. The DG edges and architectures for the three-octave DWT.

Edge Description	Edge Vector $\mathbf{e}$	Arch A1		Arch A2		Arch A3	
		$\mathbf{p}_1^T\mathbf{e}$	$\mathbf{s}_1^T\mathbf{e}$	$\mathbf{p}_2^T\mathbf{e}$	$\mathbf{s}_2^T\mathbf{e}$	$\mathbf{p}_3^T\mathbf{e}$	$\mathbf{s}_3^T\mathbf{e}$
Inputs Octave 1	$[\ 1\ \ -1\ ]^T$	-1	1	-1	0	-1	1
Inputs Octave 2	$[\ 2\ \ -1\ ]^T$	-1	2	-1	1	-1	3
Inputs Octave 3	$[\ 4\ \ -1\ ]^T$	-1	4	-1	3	-1	7
Weights	$[\ 1\ \ 0\ ]^T$	0	1	0	1	0	2
Outputs	$[\ 0\ \ 1\ ]^T$	1	0	1	1	1	1
Out 1 $\rightarrow$ In 2	$[\ 1\ \ 0\ ]^T$	0	1	0	1	0	2
Out 2 $\rightarrow$ In 3	$[\ 2\ \ 0\ ]^T$	0	2	0	2	0	4

Table 3. The DG edges and architectures for the three-octave IDWT.

Edge Description	Edge Vector $\mathbf{e}$	Arch S1		Arch S2		Arch S3	
		$\mathbf{p}_1^T\mathbf{e}$	$\mathbf{s}_1^T\mathbf{e}$	$\mathbf{p}_2^T\mathbf{e}$	$\mathbf{s}_2^T\mathbf{e}$	$\mathbf{p}_3^T\mathbf{e}$	$\mathbf{s}_3^T\mathbf{e}$
Inputs Octave 1	$[\ 1\ \ -1\ ]^T$	-1	1	-1	0	-1	1
Inputs Octave 2	$[\ 2\ \ -1\ ]^T$	-1	2	-1	1	-1	3
Inputs Octave 3	$[\ 4\ \ -1\ ]^T$	-1	4	-1	3	-1	7
Weights	$[\ 1\ \ 0\ ]^T$	0	1	0	1	0	2
Outputs	$[\ 0\ \ 1\ ]^T$	1	0	1	1	1	1
Out 2 $\rightarrow$ In 1 (even)	$[\ 1\ \ 0\ ]^T$	0	1	0	1	0	2
Out 2 $\rightarrow$ In 1 (odd)	$[\ 3\ \ 0\ ]^T$	0	3	0	3	0	6
Out 3 $\rightarrow$ In 2 (even)	$[\ 2\ \ 0\ ]^T$	0	2	0	2	0	4
Out 3 $\rightarrow$ In 2 (odd)	$[\ 6\ \ 0\ ]^T$	0	6	0	6	0	12

3. Conclusions

Systolic architectures for the one-dimensional DWT and IDWT have been systematically derived from dependence graphs. These architectures are scalable with filter length and number of octaves, are modular, have efficient memory utilization, have high hardware utilization efficiency (close to 100%), have simple routing and control, and have silicon area which is independent of input signal length.

References

[1] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Comm. in Pure and Applied Math.*, 41(7):909–996, November 1988.

[2] J. Fridman and E. Manolakos. Discrete wavelet transform: Data dependence analysis and synthesis of distributed memory and control array architectures. *IEEE Transactions on Signal Processing*, 45(5):1291–1308, May 1997.

[3] A. Grzeszczak, M. Mandal, S. Panchanathan, and T. Yeap. VLSI implementation of the discrete wavelet transform. *IEEE Transactions on VLSI Systems*, 4(4):421–433, December 1996.

[4] G. Knowles. VLSI architecture for the discrete wavelet transform. *Electronics Letters*, 26(15):1184–1185, July 1990.

[5] S. G. Mallat. Multifrequency channel decompositions of images and wavelet models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(12):2091–2110, December 1989.

[6] K. K. Parhi and T. Nishitani. VLSI architectures for discrete wavelet transforms. *IEEE Transactions on VLSI Systems*, 1(2):191–202, June 1993.

[7] O. Rioul and M. Vetterli. Wavelets and signal processing. *IEEE Signal Processing Magazine*, pages 14–38, October 1991.

[8] M. Vishwanath, R. M. Owens, and M. J. Irwin. VLSI architectures for the discrete wavelet transform. *IEEE Transactions on Circuits and Systems–II: Analog and Digital Signal Processing*, 42(5):305–316, May 1995.

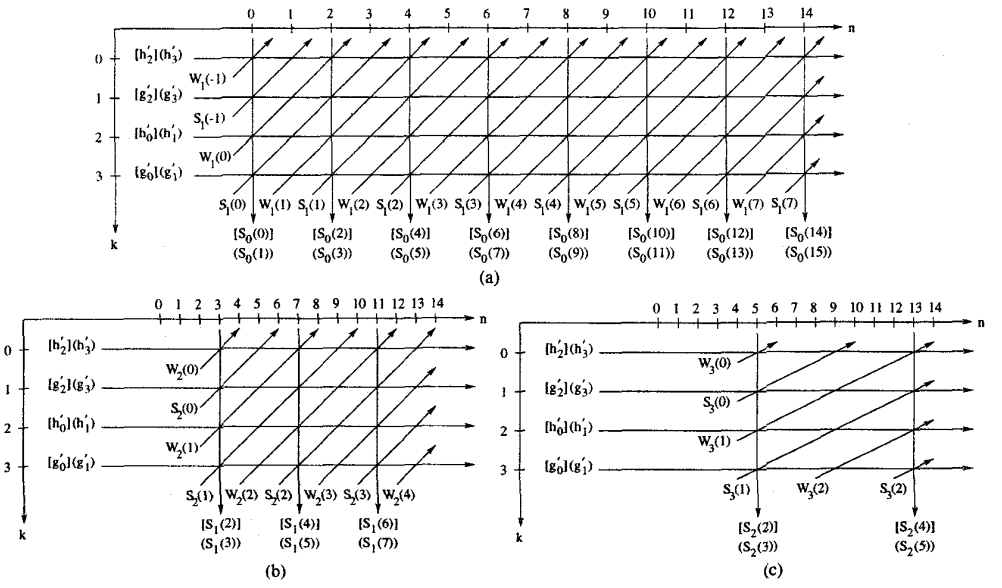

Figure 6. The dependence graph for a 1-D IDWT broken down into (a) octave 3, (b) octave 2, (c) octave 1. The three-level IDWT dependence graph is the superposition of (a), (b), and (c).

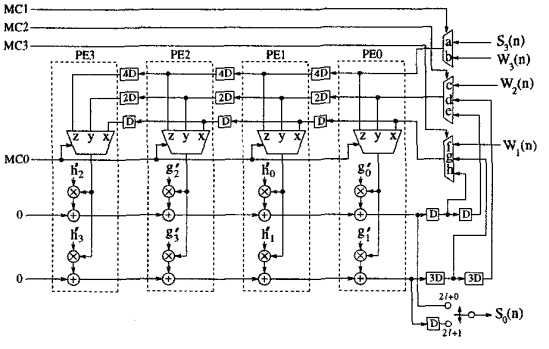

Figure 7. The three-level IDWT architecture S1. The multiplexer control signals are periodic with period 8 and have the values $MC0 = x, \phi, x, y, x, z, x, y$ and $MC1 = \phi, b, \phi, \phi, \phi, a, \phi, \phi$ and $MC2 = \phi, c, \phi, d, \phi, c, \phi, e$ and $MC3 = h, f, g, f, h, f, g, f$.

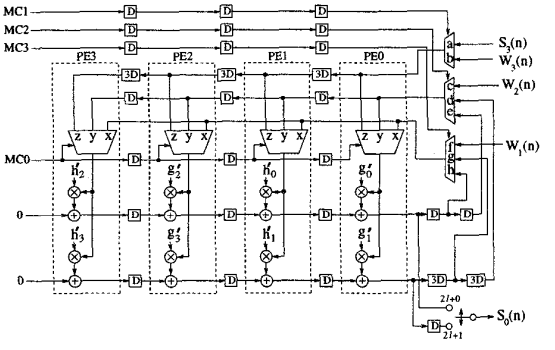

Figure 8. The three-level IDWT architecture S2. The multiplexer control signals are the same as in Figure 7.