Efficient VLSI architecture for one-dimensional discrete wavelet transform using a scalable data reorder unit

Taegeun Park
Department of Computer and Electronic Engineering, Catholic University of Korea,
San 43-1, Yokok2-dong, Wonmi-gu, Bucheon-shi, Kyongki-do, 420-743, Korea
Tel. +82-32-340-3378, Fax.: +82-32-340-3777
e-mail : parktg@catholic.ac.kr

Abstract: In this paper, we design an efficient, scalable one-dimensional discrete wavelet transform (1D DWT) filter using data reorder unit (DRU). At each level, the required hardware is optimized by sharing multipliers and adders because the input rate is reduced by a factor of two at each level due to decimation. The proposed architecture shows 100% hardware utilization by balancing hardware with input rate. Furthermore, sharing the coefficients of the highpass and the lowpass filters using the mirror filter property reduces the number of multipliers and adders by half. We designed a scalable DRU that efficiently reorders and feeds inputs to highpass and lowpass filters. The proposed DRU-based architecture is so regular and scalable that it can be easily extended to an arbitrary 1D DWT structure with M taps and J levels. Compared to other architectures, the proposed DWT filter shows efficiency in performance with relatively less hardware.

1. Introduction

There have been significant achievements in algorithms and architectures in processing digital signals, such as image, video and audio in multimedia. The discrete wavelet transform (DWT), based on time-scale representation, is receiving considerable attention in signal and image processing areas. Wavelet transform can be considered as a multi-resolution signal analysis, which decomposes a signal into its components in different frequency bands. DWT does not have the blocking effect inherent to discrete cosine transform (DCT) and provides a relatively high compression rate, so that it can be applied to low bit-rate image compression. The applications of DWT include, but are not limited to image compression, speech analysis, pattern recognition, and computer vision [2],[3],[4],[5]. One drawback of a system based on DWT is that it demands massive computations.

In recent years, many researchers have proposed a number of VLSI architectures on discrete wavelet transform (DWT) to achieve real-time signal processing [6],[7],[8],[9],[10],[12]. The lattice structure for the 1D DWT is regular and scalable with a complex folded scheduling control [6]. Three types of the recursive 1D DWT architectures (systolic, semi-systolic, and RAM-based) [7] show a good performance and hardware utilization but require complex routing networks and scheduling. Parallel approaches have been proposed to increase the performance by using more hardware [7],[8]. High-speed parallel architecture with dyadic structure proposed in [9] applies the pipeline structure with two sets of filter banks (highpass and lowpass) at each level. In this paper, we propose an efficient 1D DWT with fewer hardware resources, employing a mirror filter property in orthogonal wavelets.

This paper outlines as follows. Section 2 describes a brief introduction to the DWT and the Daubechies filter used in this paper. Section 3 describes the proposed 1D DWT architecture using data reordering. The structure of the data reorder unit (DRU) is generalized. The performance analysis and simulation for the architecture are discussed in Section 4. Finally, concluding remarks are given in Section 5.

2. Discrete Wavelet Transform

In wavelet analysis, signals are represented using a set of basis functions derived by shifting and scaling a single prototype function. The family of wavelet basis functions can be generated by translating and dilating the mother wavelet corresponding to the wavelet. Given a fixed scale m, one can find a "mother" scaling function $\phi(t)$ such that the family of functions

$$\phi_{mn}(t) = 2^{-m/2}\phi(2^{-m}t - n) \qquad (1)$$

forms an orthogonal basis. If $f_m(t)$ denotes the value of $f(t)$ at resolution level m, then $f_m(t)$ can be represented as

$$f_m(t) = \sum_n c_{(m+1)n}\phi_{(m+1)n} + \sum_n d_{(m+1)n}\varphi_{(m+1)n} \qquad (2)$$

In this equation, $c_{(m+1)n}$ and $d_{(m+1)n}$ are the scaling coefficient and the wavelet coefficient respectively, mathematically represented as

This work was partially supported by IC Design Education Center.

$$c_{mn} = \sum_k h(k-2n)c_{(m-1)k}$$
$$d_{mn} = \sum_k g(k-2n)d_{(m-1)k} \quad (3)$$

In equation (3), $h()$ and $g()$ are the lowpass and the highpass filters obtained from the wavelet. Therefore, if $\phi(t)$ is considered a lowpass function, and the wavelet $\varphi(t)$ is considered a bandpass function, then the function can be synthesized as the sum of low and high frequency components where m is the resolution level.

Whereas the classical short-time Fourier transform uses the same size of filter on the entire time-frequency domain, the wavelet transform takes different size of windows at variable scale, so that it is suitable to analyze spatial and spectral locality. Mallat showed that this multiresolution representation of a signal could be computed using a pyramid filter structure of quadrature mirror filter (QMF) pairs [1]. The implementation of the DWT can be realized in the form of filterbanks. Each filterbank comprises lowpass and highpass filters succeeded by decimation by two. In this paper we used Daubechies wavelets with mirror filter property. The coefficients of Daubechies wavelets are summarized in Table 1.

Table 1. The Daubechies filter coefficients

Daubechies $M = 6$		
m	H (lowpass)	G (highpass)
0	0.33267055295008	-0.03522629188571
1	0.80689150931109	-0.08544127388203
2	0.45987750211849	0.13501102001025
3	-0.13501102001025	0.45987750211849
4	-0.08544127388203	-0.80689150931109
5	0.03522629188571	0.33267055295008
Daubechies $M = 4$		
0	0.48296291314453	0.12940952255126
1	0.83651630373781	0.22414386804201
2	0.22414386804201	-0.83651630373781
3	-0.12940952255126	0.48296291314453

The Daubechies filters used in this research satisfies the following relationship between the filter coefficients.

$$g_{M-1-m} = (-1)^m h_m \qquad (M > 2) \qquad (4)$$

If we apply this equation to the lowpass and highpass filter, the transfer function $H(z)$ and $G(z)$ can be rewritten.

$$H(z) = h_0 + h_1 z^{-1} + \cdots + h_{M-1} z^{-(M-1)}$$
$$G(z) = (-1)^{M-1} h_{M-1} + (-1)^{M-2} h_{M-2} z^{-1} + \cdots + (-1)^0 h_0 z^{-(M-1)} \quad (5)$$

Therefore, 4-tap 1D Daubechies filter can be computed as follows;

$$v_n = a_{2n}h_0 + a_{2n-1}h_1 + a_{2n-2}h_2 + a_{2n-3}h_3: \text{ low}$$
$$u_n = -a_{2n}h_3 + a_{2n-1}h_2 - a_{2n-2}h_1 + a_{2n-3}h_0: \text{ high} \quad (6)$$

$$v_0 = a_0h_0 \qquad [l_0^j]$$
$$u_0 = -a_0h_3 \qquad [h_0^j]$$
$$v_1 = a_2h_0 + a_1h_1 + a_0h_2 \qquad [l_1^j]$$
$$u_1 = -a_2h_3 + a_1h_2 - a_0h_1 \qquad [h_1^j]$$
$$v_2 = a_4h_0 + a_3h_1 + a_2h_2 + a_1h_3 \qquad [l_2^j]$$
$$u_2 = -a_4h_3 + a_3h_2 - a_2h_1 + a_1h_0 \qquad [h_2^j] \quad (7)$$
$$\vdots$$

3. Proposed 1D DWT Architecture

The proposed 1D DWT architecture in Figure 1 shows a cascade of DWT blocks at each level, where N is the number of inputs and L^i and H^i are the outputs of lowpass and highpass filters at level i. Since the input data rate is reduced by half at each level, the required hardware to finish the DWT process for a certain level can be half of the previous level. In Figure 1, P_i stands for the required hardware at level i; for example, a single M-tap filter is required for level 1. In M-tap Quadrature Mirror Filter (QMF) structure, $g_{M-1-m} = (-1)^m h_m$ holds, where g and h are the coefficients of highpass and lowpass filters and $0 \le m \le M-1$. We propose a parallel 1D DWT architecture only with one set of filters employing the relation stated above. The data reorder unit (DRU) is designed to rearrange data for lowpass and highpass filters.

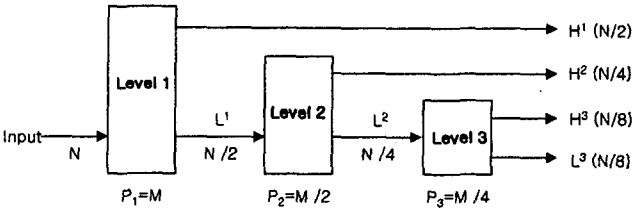

Figure 1. The proposed 1D DWT architecture

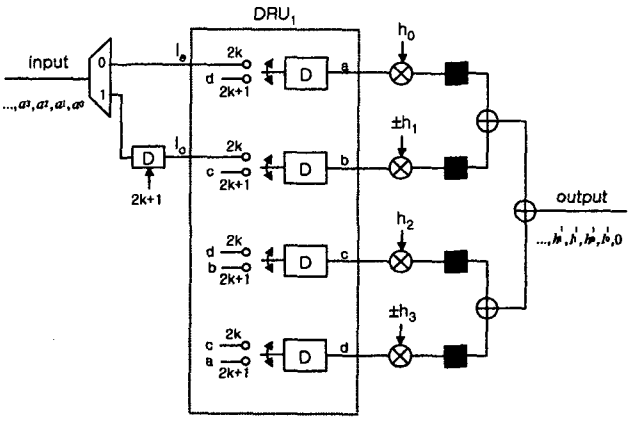

Figure 2. Block diagram of level 1

Figure 2 shows a block diagram of the 1D DWT (level 1) with 4 taps. The input sequence I is split into even and odd and fed into the DRU block. The proposed structure produces lowpass and highpass filtering outputs at odd and even clock cycles respectively. To make the filtering possible, the DRU reorders the stored data (with inputs at even clock cycles) and feeds them to the filter block. The DRU-based architecture is quite regular and can be easily extended to an arbitrary 1D DWT structure with M taps and J levels. The DRU block of 1D DWT with M-tap filter length consists of M registers, and each register in DRU selects one of two inputs.

The following explains the DRU structure of level 1. In even clock cycles, the first two registers select I_e and I_o as inputs, whereas the rest of registers rearrange the data among them in reverse order. In odd clock cycles, however, all of M registers rearrange the data in reverse order. We can design the DRU of j-level in a similar way, where $1 \le j \le J$. The number of registers in DRU has been minimized by lifetime analysis [11]. Table 2 depicts the data flow of level 1 in the proposed architecture that shows the rearrangement of data for filtering at each clock cycle.

Table 2. Data flow of level 1

clk		0	1	2	3	4	5	6	7	8
I		a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
I_e		a_0	-	a_2	-	a_4	-	a_6	-	a_8
I_o		-	-	a_1	a_1	a_3	a_3	a_5	a_5	a_7
DRU	a	-	a_0	-	a_2	-	a_4	a_1	a_6	a_3
	b	-	-	-	a_1	a_0	a_3	a_2	a_5	a_4
	c	-	-	-	a_0	a_1	a_2	a_3	a_4	a_5
	d	-	-	a_0	-	a_2	a_1	a_4	a_3	a_6
O		0	l^1_0	h^1_0	l^1_1	h^1_1	l^1_2	h^1_2	l^1_3	h^1_3

Figure 3 and Figure 4 show the block diagrams of level 2 and level 3 respectively. The shaded squares in the figures are nothing but the latches for pipelining for high performance and low power. The offset of the switching time in DRUs of level 2 and 3 is slightly adjusted due to the output delay of the previous levels. If we ignore the pipeline latches, the output delay (latency) Δ_j at level j can be estimated as $\Delta_j = \Delta_{j-1} + 2^{j-1}$ $(j \ge 1, \Delta_0 = 0)$. In this case, the register in the output of level j ($j \ge 2$) should be reset to accumulate the partial sums at $T_j = \Delta_j + 2^{j-1} \cdot k$ clocks (k=0,1,2,3,...).

The proposed architecture is well balanced with data rate at each level and therefore shows 100 % hardware utilization. The data flow table for levels 2 and 3 can be acquired in a similar way to Table 2. The required number of multipliers and adders for the J-level 1D DWT is approximately $\sum_{k=1}^{J} \lceil M/2^{k-1} \rceil$, where M is the filter length.

The proposed DRU structure is so regular that it can be easily extended to the 1D DWT with M taps and J levels. At $2^{j-1} \cdot k + \Delta_{j-1}$, data in registers is swapped on the axis of $M/2$ and $(M/2)+1$ when M is odd, while swapped on the axis of $\lfloor M/2 \rfloor + 1$ when M is even. On the other hand, at $2^{j-1} \cdot k + 2^{j-1} + \Delta_{j-1}$, data in registers is swapped on the axis of $(M/2)-1$ and $M/2$ when M is odd, while swapped on the axis of $\lfloor M/2 \rfloor$ when M is even.

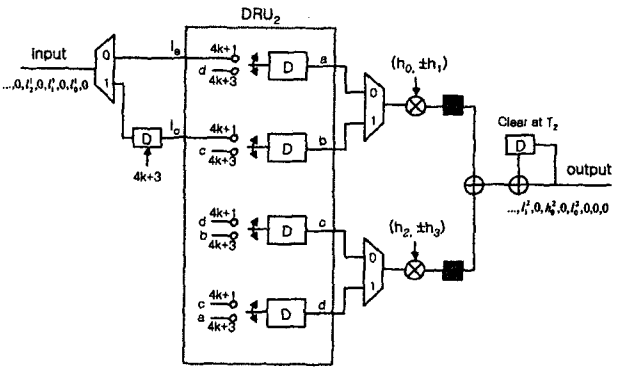

Figure 3. Block diagram of level 2

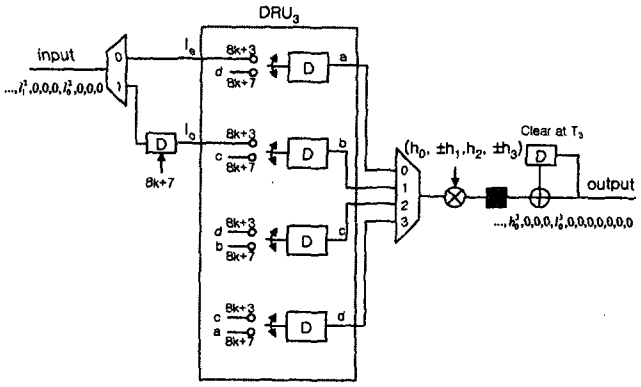

Figure 4. Block diagram of level 3

4. Simulations and Analysis

The proposed 1D DWT was designed in VHDL and verified under Modelsim environment. Figure 5 shows the timing diagram for the proposed 1D DWT. The signal (H^1,L^1), (H^2,L^2), and (H^3,L^3) represent the outputs of levels 1, 2, and 3 respectively. The outputs of the lowpass and highpass filters at each level are overlapped by half DWT cycle and produced every 2 clock cycles for (H^1,L^1), every 4 clock cycles for (H^2,L^2), and every 8 clock cycles for (H^3,L^3). The filter coefficients and data of the proposed 1D DWT are fixed-point numbers. The Daubechies filter coefficients and data are encoded with 8 bits and 16 bits respectively. The lower 5 bits of data are assigned for the fractional part and the remaining is assigned for integer and sign part.

In Table 3, we compare the performance of our architecture with that of the various 1D DWT architectures in terms of multipliers, adders, period, hardware utilization,

and scheduling. The systolic architecture in [7] has a period of $2N$ to complete the DWT and requires complex routing networks to provide the filter cells' right values. In another systolic architecture [12], the hardware efficiency decreases as the decomposition level increases. The parallel approach in [8] needs a similar amount of hardware; however, it additionally requires $J \bullet M$ storage to keep the intermediate data for higher levels. The carefully balanced pipeline architecture in [9] shows higher throughput by utilizing two filter banks in parallel. The folded architecture that has shorter latency does not achieve maximum hardware utilization and requires complex routing and control [10].

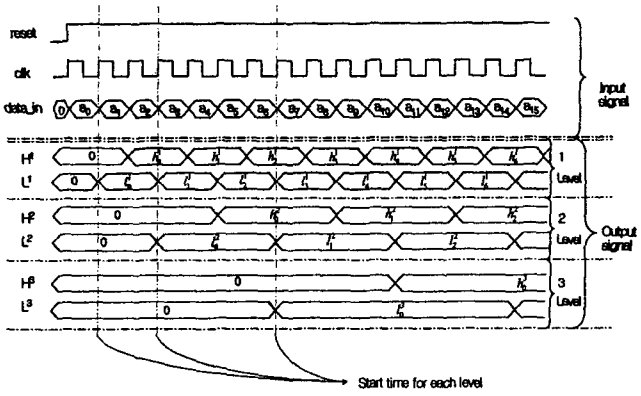

Figure 5. The timing diagram for the proposed 1D DWT

Table 3. The performance comparisons of the various 1D DWT architectures (M=4, J=3)

Architecture	Mult.	Adder	Period	H/W util.	Scheduling
Systolic [7]	*8*	*8*	*2N*	*100%*	complex
Parallel [8]	*8*	*6*	*N*	*100%*	complex
Pipelined [9]	*14*	*14*	*N/2*	*100%*	simple
Folded [10]	*8*	*6*	*N*	*87.5%*	complex
Systolic [12]	*12*	*9*	*N*	*58.3%*	simple
Proposed	*7*	*6*	*N*	*100%*	simple

5. Conclusion

In this paper, efficient 1D DWT architecture in orthogonal wavelets is proposed. Highpass and lowpass filtering outputs are produced in a single filter by virtue of data reordering in DRU. The required hardware at each decomposition level is reduced by half to balance with input data rate, and therefore the proposed architecture shows 100% hardware utilization. The DRU-based architecture is quite regular and scalable so that it fits into a VLSI implementation and can be extended to an arbitrary 1D DWT structure with M taps and J levels.

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, 1989

[2] T. W. Ryan, L. D. Sanders, H. D. Fisher, and A. E. Iverson, "Image compression by texture modeling in the wavelet domain," *IEEE Trans. Image Process.*, vol. 5, pp. 26-36, 1996

[3] G. M. Davis, "A wavelet-based analysis of fractal image compression," *IEEE Trans. Image Process.*, vol. 7, pp. 141-154, 1998

[4] R. Kronland-Martinet, J. Morlet, and A. Grossman, "Analysis of sound patterns through wavelet transform," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 1 no. 2, pp. 273-302, 1987

[5] M. Unser, "Texture classification and segmentation using wavelet frames," *IEEE Trans. Image Processing*, vol. 4, no. 11, pp. 1549-1560, 1995

[6] J. T. Kim, Y. H. Lee, T. Isshiki, and H. Kunieda, "Scalable VLSI architectures for lattice structure-based discrete wavelet transform," *IEEE Trans. CAS-II*, vol. 45, no. 8, pp. 1031-1043, 1998

[7] M. Vishwanath, R. M. Owens, and M. J. Irwin, "VLSI architectures for the discrete wavelet transform," *IEEE Trans. CAS-II*, vol. 42, no. 5, pp. 305-316, 1995

[8] C. Chakrabarti and M. Vishwanath, "Efficient realization of the discrete and continuous wavelet transforms: from single chip implementations to mappings on SIMD array computers," *IEEE Trans. SP*, vol. 43, no. 3, pp. 759-771, 1995

[9] F. Marino, D. Guevorkian, and J. Astola, "Highly efficient high-speed/low-power architectures for the 1D discrete wavelet transform," *IEEE Trans. CAS-II*, vol. 47, no. 12, pp. 1492-1502, 2000

[10] K. Pahri and T. Nishitani, "VLSI architectures for discrete wavelet transforms', *IEEE Trans. VLSI Syst.*, vol. 1, no. 2, pp. 191-202, 1993

[11] K. Pahri, *VLSI digital signal processing systems*, John Wiley & Sons, 1999

[12] S. Pan and R. Park, "An efficient VLSI architecture for the discrete wavelet transform," *Journal of IEEK*, vol. S-36, no. 6, pp. 96-103, 1999