The FPGA Implementation of Wavelet Transform Chip using Daubechies' 4 Tap Filter for DSP Application

Jeong, Chang-Soo and Kim, Nam-Young
Dept. of Electronic Engineering, Kwangwoon University.
E-mail : nykim@daisy.kwangwoon.ac.kr

ABSTRACT

The wavelet transform chip is implemented with Daubechies' 4 tap filter. It works at 20MHz in Field Programmable Gate Array (FPGA) implementation of Quadrature Mirror Filter(QMF) Lattice Structure. In this paper, the structure contains two-channel quadrature mirror filter, data format converter(DFC), delay control unit(DCU), and three 20×8 bits real multiplier. The structures for the DFC and DCU need to be regular and scalable, require minimum number of registers, and thereby lead to an efficient and scalable architecture for the Discrete Wavelet Transform(DWT). These results present the possibility that it can be used in Digital Signal Processing(DSP) application faster than Fourier transform at small area with low cost.

I. INTRODUCTION

The DWT with multiresolution analysis has emerged recently as a powerful signal processing tool after the discovery of the DWT by Mallat.[1] It can reduce the complexity of calculation compared with that of Fourier transform. This means that it make possibile real time DSP -- noise removal 1D or 2D digital image and speech processing for communication system.[2][3]

Due to its inherent complexity and time scale locality characteristics, the DWT has received considerable attention in DSP application. It is usually implemented based on the binary tree structured Quadrature Mirror Filter bank.[4][5]

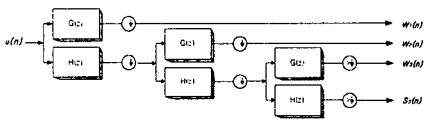

Fig 1. Three-level binary tree structured QMF banks for DWT

At its level of the tree for the forward transform in Fig. 1., the outputs of the two channel QMF bank H(z)(low-pass filter) and G(z)(high-pass filter) are evaluated and decimated by a factor of two. An attractive feature of this tree structure, which is useful for VLSI implementation, is stated as follows; in the tree, the number of filter outputs computed during each sample period is upper bounded by two, irrespective of number of levels. This property naturally leads to VLSI architecture that employ only one pair of filters and iteratively use them for all levels. All VLSI implementations for DWT introduced so far have such an architecture, which consists of a data format converter, delay control unit,

and a two-channel filter bank. The DFC controls data transfer between levels and the DCU controls data delay between Process Element(PE).

In this paper, the architectures of DWT with QMF Lattice Filter structure was studied and tested feasibility of implementing. Actually, The FPGA chip was made Very high speed integrated circuit Hardware Description Language(VHDL).

The architecture needs to L registers if input data is L bits but, our suggested architecture only need to log(L) registers. therefore it is a optimized structure for implementation of VLSI.

II. Discrete Wavelet Transform

The DWT may be calculated recursively as a series of convolutions and decimations. At each level (octave) j an input sequence $S_{j-1}(n)$ is fed into a low and high pass filter H(z) and G(z) respectively. The output from the high pass filter G(z) represents the detail information in the original signal at the given scale j, denoted by $W_j(n)$. The output from the low pass filter H(z) represents the remaining information in the original signal, denoted by $S_j(n)$.

The sequential DWT algorithm may be expressed as

$$S_j(n)=\sum_k S_{j-1}(k)H(2n-k) \qquad (1)$$

$$W_j(n)=\sum_k S_{j-1}(k)G(2n-k) \qquad (2)$$

where H(z) and G(z) are the wavelet filter coefficients. At the output of every level there is a downsampling operation which appears as the factor of two in the argument of the filter coefficients.

For implementing the DWT, the input-output relations between the PEs is needed. The relation between these inouts and outputs are derived directly from the previous discussions. For the first PE(PE_0),

$$X_0^U(n)=\begin{cases} u(n), & (n=2l) \\ Y_{M-1}^U(n-2^{j-2}), & (n=2^j l+2^{j-1}-1) \end{cases} \qquad (3\text{-a})$$

$$X_0^U(n)=\begin{cases} u(n), & (n=2l) \\ Y_{M-1}^U(n-2^{j-1}-2^{j-2}), & (n=2^j l+2^{j-1}-1) \end{cases} \qquad (3\text{-b})$$

For the i-th PE(PE_i), $1 \le i \le M-1$,

$$X_i^U(n) = Y_{i-1}^U(n) \qquad (4\text{-a})$$

$$X_i^L(n) = Y_{i-1}^L(n-2^j), \quad (n=2^j l+2^{j-1}-1) \qquad (4\text{-b})$$

An architecture for lattice structure-based DWT in Fig. 2. has the Data Format Converter (DFC) and Delay Control Unit (DCU). The DFC controls the input and the feedback sequences depending on (3). The DCU controls the delay Z^{-2^j} depending on the relation between the time index n and the resolution level j, given by (4-b)

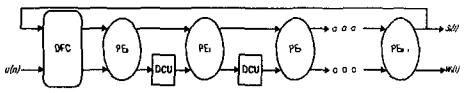

Fig. 2. An architecture for lattice structure-based DWT

In order to design DCU for Daubechies' 4 tap filter of 3 resolution level (j = 3), (4-b) is rewritten as

$$X_i^L(n)=\begin{cases} Y_{i-1}^L(n-2) & \text{for } n=2l \\ Y_{i-1}^L(n-4) & \text{for } n=4l+1 \\ Y_{i-1}^L(n-8) & \text{for } n=8l+3 \end{cases} \qquad (5)$$

This equation directly formed the structure of Fig. 3. It looks very simple but requires about 2^j word-level registers for j resolution levels. The more resolution levels the more the number of registers increases exponentially. It

is possible to reduce the number of registers by using the Parhi's method.[4] Therefore the structure of Fig. 3. could be changed as shown the Fig. 4.

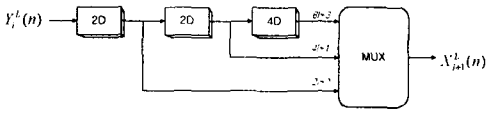

Fig. 3. Directly Designed DCU

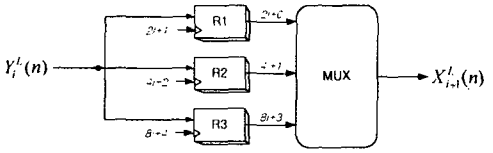

Fig. 4. Proposed DCU by Parhi's method

Next we have to design of a DFC for the DWT with 3 resolution levels (j = 3). (3-a),(3-b) can be rewritten as

$$X_0^U(n)=\begin{cases} u(n), & (n=2l) \\ Y_{M-1}^U(n-1), & (n=4l+1) \\ Y_{M-1}^U(n-2), & (n=8l+3) \end{cases} \quad (6\text{-a})$$

$$X_0^L(n)=\begin{cases} u(n-1), & (n=2l) \\ Y_{M-1}^U(n-3), & (n=4l+1) \\ Y_{M-1}^U(n-6), & (n=8l+3) \end{cases} \quad (6\text{-b})$$

Direct design of DFC with these equations formed as Fig. 5. It also has to require $3 \cdot 2^{i-2}+1$ word-level registers. It can change the structure of Parhi's method[4] if we consider to reduce the number of registers. The structure could be changed as shown Fig. 6.

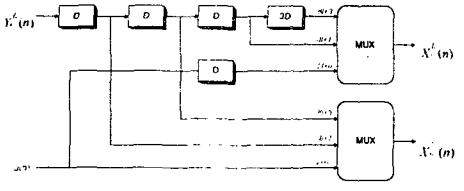

Fig. 5. Direct Designed DFC

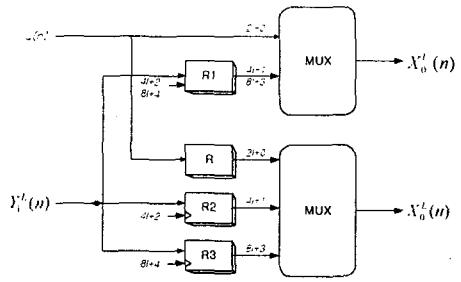

Fig. 6. Proposed DFC by Parhi's method

Now the whole architecture of DWT for VLSI implementation could be designed as follow. The all architecture is shown Fig. 7.

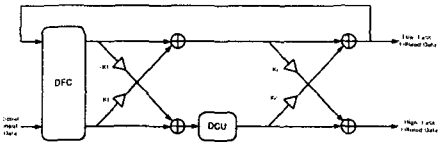

Fig. 7. QMF Lattice structured-based DWT using Daubechies' 4 Tap Filter

The QMF lattice filter is the solutions as follow:

$$F_2(n) = K_2 \cdot X_0 - K_1K_2 \cdot X_1 + K_1 \cdot X_2 + X_3 \quad (7\text{-a})$$

$$G_2(n) = X_0 - K_1 \cdot X_1 - K_1K_2 \cdot X_2 - K_2 \cdot X_3 \quad (7\text{-b})$$

The coefficients of Daubechies' 4 tap filter is as follow:

$$H_i(n) = \frac{1}{8}\{(1+\sqrt{3})X_0 + (3+\sqrt{3})X_1 + (3-\sqrt{3})X_2 + (1-\sqrt{3})X_3\} \quad (8\text{-a})$$

$$G_i(n) = \frac{1}{8}\{(1-\sqrt{3})X_0 - (3-\sqrt{3})X_1 + (3+\sqrt{3})X_2 - (1+\sqrt{3})X_3\} \quad (8\text{-b})$$

Due to Eq. (7) and (8), the K_1 and K_2 can obtain.

$$K_1 = -\sqrt{3}$$

$$K_2 = -(2 + \sqrt{3})$$

The implementation of DWT may be optimized if it is produced by full custom design of the semiconductor fab. On the other hand, it is made by FPGA chip using VHDL. The FPGA implementation could be designed and tested in a short time than full custom design. Therefore, the VHDL language, synthesis with Synopsys tool are used to design by Flex10k70RC240-3(Altera co.) chip.

III Conclusion

The FPGA implementation of QMF lattice structure-based DWT using Daubechies' 4 tap filter is designed and tested. The DWT chip works about 20MHz. It shows regular and scalable, requires few registers. The structure has two-channel QMF, DFC, DCU, and three 20×4 bits real multiplier. The results present the possibility that it can be used in DSP application faster than Fourier transform at small area with low cost.

IV References

[1] S. G. Mallat, "A theory of multiresolution signal decomposition: The wavelet representation." IEEE Trans. Pattern Anal. Machine Intell., Vol. 19, pp. 674-693, July 1989.

[2] D. L. Donoho, "De-noising by soft-thresholding ," IEEE Trans. Inform. Theory 41, pp. 613-627, May 1995.

[3] M. Vetterli and C. Herley, "Wavelets and Filter banks: Theory and Design," IEEE Trans. Acoust., Speech, Signal Processing, Vol. 40, pp. 2207-2232, Sept. 1992

[4] K. K. Parhi, "VLSI architectures for discrete wavelet transform," IEEE Trans. VLSI Syst., Vol 1, pp. 191-202, June 1993.

[5] Joon Tae Kim and Young Hoon Lee, "Scalable VLSI architectures for lattice structure-based discrete wavelet transform," IEEE Trans. Circuits and Systems-II., Vol. 45, No. 8, pp. 1031-1043, August 1998.