Aleksandr CARIOW, Galina CARIOWA
WEST POMIERANIAN UNIVERSITY OF TECHNOLOGY, SZCZECIN
49 Żołnierska St., 71-210 Szczecin

Hardware-efficient algorithms for implementation of the GHM discrete multiwavelet transform kernels

Abstract

In this correspondence, we discuss two efficient algorithms for the execution of forward (FDMWT) and inverse (IDMWT) discrete multiwavelet transform basic operations with reduced computational complexities. We used multiwavelet basis proposed by Geronimo, Hadrin, and Massopust (GHM). The direct implementation of GHM-FDMWT basic operation requires 23 multiplications and 19 additions. The direct implementation of GHM-IDMWT basic operation requires 23 multiplication and 16 additions. At the same time, our solutions allow designing the computation procedures, which take only 10 multiplications plus 15 additions for GHM-FDMWT basic operation and 10 multiplications plus 10 additions for GHM-IDMWT basic operation.

Keywords: multiwavelets, GHM, fast algorithms, implementation complexity reduction, FPGA implementation.

1. Introduction

Recently, discrete wavelets transform (DWT) [1] and discrete multiwavelet transform (DMWT) [2] have been used in numerous signal and image processing applications [3-13]. The multiwavelet transform is a new concept in the framework of wavelet transform but it has some important differences. Whereas the wavelet transform has one scaling function and a wavelet function, the multiwavelet transform has two or more scaling functions and mother wavelets for data representation. Multiwavelet transforms possess more superior properties than any single wavelet function such as short support, symmetry, vanishing moments and smoothness. Due to these important properties, they can be used for advanced signal and image processing. Different variations of multiwavelet bases have been developed and presented in the scientific literature. One of the well-known multiwavelet basis obtained with the help of fractal interpolation has been proposed by Geronimo, Hardin, and Massopust (GHM) [10]. GHM basis has become a popular tool for data processing in many computer science applications. Unfortunately, the GHM multiwavelet transform typically requires a significant computational effort [11-14]. This problem becomes extremely challenging for applications requiring real-time processing at high throughput especially in digital signal and image processing. Hence, to meet the stringent requirements to throughput, latency, and power-consumption constraints of real-time DSP systems, the development of dedicated hardware implementations, such as application specific integrated circuits (ASICs) or field-programmable gate arrays (FPGAs), is of paramount importance.

2. Statement of the problem

The "basic operations" of the forward and inverse discrete multiwavelet transform (similar to a "butterfly" operation in the FFT algorithm) are the GHM-FDMWT and GHM-IDMWT kernels, respectively.

In the matrix-vector form, the GHM-FDMWT basic operation can be defined in the following way:

$$\mathbf{Y}_{4\times1}^{(f)} = \mathbf{F}_{4\times8}\mathbf{X}_{8\times1}^{(f)}, \tag{1}$$

while the inverse operation used to reconstruct signal from GHM-DMWT coefficients (inverse GHM-DMWT base operation) is as follows:

$$\mathbf{X}_{8\times1}^{(i)} = \mathbf{F}_{4\times8}^{(i)}\mathbf{Y}_{4\times1}^{(i)} \tag{2}$$

where $\mathbf{X}_{8\times1}^{(f)} = [x_0^{(f)}, x_1^{(f)}, ..., x_7^{(f)}]^T$ - is an 8-element input data vector and $\mathbf{Y}_{4\times1}^{(f)} = [y_0^{(f)}, y_1^{(f)}, y_2^{(f)}, y_3^{(f)}]^T$ - is a 4-element output data vector for GHM-FDMWT kernel (1), $\mathbf{Y}_{4\times1}^{(i)} = [y_0^{(i)}, y_1^{(i)}, y_2^{(i)}, y_3^{(i)}]^T$ - is an 4-element input data vector and $\mathbf{X}_{8\times1}^{(i)} = [x_0^{(i)}, x_1^{(i)}, ..., x_7^{(i)}]^T$ - is a 8-element output data vector for GHM-IDMWT kernel (2), and superscripts f and i indicate the forward and inverse transforms, respectively.

$$\text{Matrix } \mathbf{F}_{4\times8}^{(f)} = \begin{bmatrix} \mathbf{H}_2^{(0)} & \mathbf{H}_2^{(1)} & \mathbf{H}_2^{(2)} & \mathbf{H}_2^{(3)} \\ \mathbf{G}_2^{(0)} & \mathbf{G}_2^{(1)} & \mathbf{G}_2^{(2)} & \mathbf{G}_2^{(3)} \end{bmatrix} \tag{3}$$

in (1) is a GHM-FDMWT base matrix with dimensions of (4×8) whose submatrices $\mathbf{H}_2^{(l)}$ and $\mathbf{G}_2^{(l)}$ contain the coefficients of the high-pass and low-pass matrix-valued filter impulse responses, respectively.

In turn, the matrix:

$$\mathbf{F}_{8\times4}^{(i)} = \begin{bmatrix} [\mathbf{H}_2^{(0)}]^T & [\mathbf{G}_2^{(0)}]^T \\ [\mathbf{H}_2^{(1)}]^T & [\mathbf{G}_2^{(1)}]^T \\ [\mathbf{H}_2^{(2)}]^T & [\mathbf{G}_2^{(2)}]^T \\ [\mathbf{H}_2^{(3)}]^T & [\mathbf{G}_2^{(3)}]^T \end{bmatrix} \tag{4}$$

in (2) is a GHM-IDMWT base matrix with dimensions of (8×4), and submatrices $[\mathbf{H}_2^{(l)}]^T$ and $[\mathbf{G}_2^{(l)}]^T$ are the transposed matrices of the matrices $\mathbf{H}_2^{(l)}$ and $\mathbf{G}_2^{(l)}$.

For the GHM basis these submatrices are as follows:

$$\mathbf{H}_2^{(0)} = \begin{bmatrix} 3/5\sqrt{2} & 4/5 \\ -1/20 & -3/10\sqrt{2} \end{bmatrix}, \ \mathbf{H}_2^{(1)} = \begin{bmatrix} 3/5\sqrt{2} & 0 \\ 9/20 & 1/\sqrt{2} \end{bmatrix},$$

$$\mathbf{H}_2^{(2)} = \begin{bmatrix} 0 & 0 \\ 9/20 & -3/10\sqrt{2} \end{bmatrix}, \ \mathbf{H}_2^{(3)} = \begin{bmatrix} 0 & 0 \\ -1/20 & 0 \end{bmatrix}$$

$$\mathbf{G}_2^{(0)} = \begin{bmatrix} -1/20 & -3/10\sqrt{2} \\ 1/10\sqrt{2} & 3/10 \end{bmatrix}, \ \mathbf{G}_2^{(1)} = \begin{bmatrix} 9/20 & -1/\sqrt{2} \\ -9/10\sqrt{2} & 0 \end{bmatrix},$$

$$\mathbf{G}_2^{(2)} = \begin{bmatrix} 9/20 & -3/10\sqrt{2} \\ 9/10\sqrt{2} & -3/10 \end{bmatrix}, \ \mathbf{G}_2^{(3)} = \begin{bmatrix} -1/20 & 0 \\ -1/10\sqrt{2} & 0 \end{bmatrix}.$$

Formulas (1), (2) define the GHM-FDMWT and GHM-IDMWT basic kernels, respectively.

Substituting the numerical values of $\mathbf{H}_2^{(l)}$ and $\mathbf{G}_2^{(l)}$ into the formula (3) and numerical values of $[\mathbf{H}_2^{(l)}]^T$ and $[\mathbf{G}_2^{(l)}]^T$ into the formula (4) we obtain:

$$\mathbf{F}_{4\times8} = [\mathbf{F}_4^{(0)} \mid \mathbf{F}_4^{(1)}] \tag{5}$$

$$\mathbf{F}_4^{(0)} = \begin{bmatrix} 3/5\sqrt{2} & 4/5 & 3/5\sqrt{2} & 0 \\ -1/20 & -3/10\sqrt{2} & 9/20 & 1/\sqrt{2} \\ -1/20 & -3/10\sqrt{2} & 9/20 & -1/\sqrt{2} \\ 1/10\sqrt{2} & 3/10 & -9/10\sqrt{2} & 0 \end{bmatrix} \tag{6}$$

$$\mathbf{F}_4^{(1)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 9/20 & -3/10\sqrt{2} & -1/20 & 0 \\ 9/20 & -3/10\sqrt{2} & -1/20 & 0 \\ 9/10\sqrt{2} & -3/10 & -1/10\sqrt{2} & 0 \end{bmatrix} \tag{6}$$

$$\mathbf{F}_{8\times4} = \begin{bmatrix} 3/5\sqrt{2} & -1/20 & -1/20 & 1/10\sqrt{2} \\ 4/5 & -3/10\sqrt{2} & -3/10\sqrt{2} & -3/10 \\ 3/5\sqrt{2} & 9/20 & 9/20 & -9/10\sqrt{2} \\ 0 & 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 0 & 9/20 & 9/20 & 9/10\sqrt{2} \\ 0 & -3/10\sqrt{2} & -3/10\sqrt{2} & -3/10 \\ 0 & -1/20 & -1/20 & -1/10\sqrt{2} \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{7}$$

From expressions (1) and (2) it is clear that the implementation of the GHM-FDMWT basic operations required carrying out 23 multiplications and 19 additions whereas the implementation of the GHM-IDMWT basic operations required making 23 multiplications and 16 additions. We show two algorithms that require a much smaller amount of computations for the implementation of the GHM-FDMWT and GHM-IDMWT kernels.

3. The algorithms

The first algorithm (for implementation of the GHM-FDMWT kernel) can be written with the help of the following matrix-vector calculating procedure:

$$\mathbf{Y}_{4\times1}^{(f)} = \mathbf{W}_{4\times6}\mathbf{D}_6\mathbf{W}_{6\times9}\mathbf{D}_9\mathbf{W}_{9\times8}\mathbf{X}_{8\times1}^{(f)} \tag{3}$$

where

$$\mathbf{W}_{9\times8} = \begin{bmatrix} 1 & 1 & & & & & & \\ & 1 & & & & & & \\ 1 & & & & & & 1 & \\ 1 & & & & & & & -1 \\ & & & -1 & 1 & & & \\ & & & 1 & 1 & & & \\ & & & & & 1 & & \\ 1 & & & & & & & -1 \\ 1 & & & & & & & 1 \end{bmatrix}$$

$$\mathbf{D}_9 = diag(\frac{3}{5\sqrt{2}}, \frac{4}{5}, 1, 1, 9, 9, 1, \frac{3}{10}, \frac{3}{10}),$$

$$\mathbf{D}_6 = diag(1, \frac{1}{20}, \frac{1}{10\sqrt{2}}, \frac{1}{2}, \frac{1}{2}, 1),$$

$$\mathbf{W}_{4\times6} = \begin{bmatrix} 1 & & & & & \\ & 1 & & -1 & & \\ & 1 & -1 & & & \\ & 1 & & & 1 & \end{bmatrix},$$

$$\mathbf{W}_{6\times9} = \begin{bmatrix} 1 & 1 & & & & \\ & -1 & & 1 & & \\ & 1 & 1 & & & \\ & & & & 1 & 1 \\ & & & & -1 & 1 \\ & & & & & 1 \end{bmatrix}.$$

The second algorithm (for implementation of the GHM-IDMWT kernel) can be written with the help of the following matrix-vector calculating procedure:

$$\mathbf{X}_{8\times1}^{(i)} = \mathbf{W}_{8\times9}\mathbf{D}_9\mathbf{W}_{9\times6}\mathbf{D}_6\mathbf{W}_{6\times4}\mathbf{Y}_{4\times1}^{(i)} \tag{4}$$

where

$$\mathbf{W}_{8\times9} = [\mathbf{W}_{9\times8}]^T, \quad \mathbf{W}_{9\times6} = [\mathbf{W}_{6\times9}]^T, \quad \mathbf{W}_{6\times4} = [\mathbf{W}_{4\times6}]^T.$$

Fig. 1 shows a data flow diagram of the proposed algorithm for realization of the GHM-FDMWT kernel, whereas Fig. 2 shows a data flow diagram of the proposed algorithm for realization of the GHM-IDMWT kernel. In this paper, the data flow diagrams are oriented from left to right. The straight lines in the figures denote the operations of data transfer. The circles in these figures show the operation of multiplication by a number inscribed inside the circle. The points where the lines converge denote summation the dotted lines indicate the sign-change operations. We use the usual lines without arrows purposefully so as not to clutter the picture.
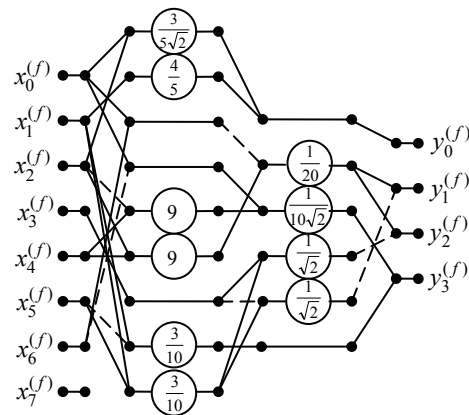

Fig. 1. The data flow diagram of the proposed algorithm for realization of the GHM-IDMWT kernel

Taking into account the presence of zero elements in the matrices $\mathbf{F}_{4\times8}^{(f)}$ and $\mathbf{F}_{8\times4}^{(i)}$, the actual number of multiplications needed to implement the GHM-FDMWT/GHM-IDMWT kernels is 23 instead 32. To implement our algorithms, it is required to carry out only 10 multiplications. This is more than twice reduction.

Minimizing the number of multiplications is especially important in the design of specialized VLSI on-board DSP processors because reducing the number of multipliers also reduces the power dissipation and lowers the implementation cost of the entire system being implemented.

Moreover, a hardware multiplier is a more complicated unit than an adder and occupies much more chip area than the adder. Even if the VLSI chip already contains embedded multipliers, their number is always limited. This means that if the implemented algorithm has a large number of multiplications, the projected processor may not always fit into the chip and the problem of minimizing the number of multiplications remains relevant [15].
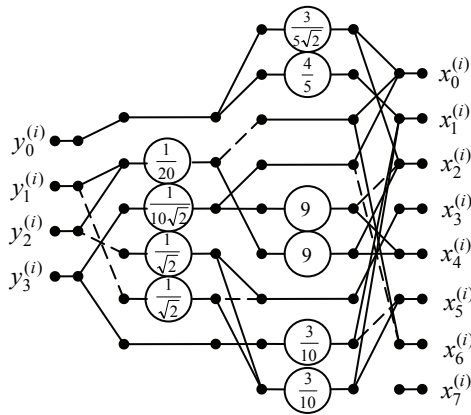

Fig. 2. The data flow diagram of the proposed algorithm for realization of the GHM-IDMWT kernel

4. Conclusions

The paper presents two new FPGA-oriented algorithm for the execution of GHM-FDMWT and GHM-IDMWT kernels with reduced computational complexities. To reduce the hardware complexity (number of embedded multipliers), we exploit the specific structural properties of the matrix-vector products that represent GHM-FDMWT and GHM-IDMWT basic operations. So, the fully parallel implementation of the GHM-FDMWT kernel requires 8 multipliers by fractional numbers, two multipliers by integer numbers and 15 adders. The fully parallel implementation of the GHM-IDMWT kernel requires also 8 multipliers by fractional numbers, two multipliers by integer numbers and only 10 adders. For example, a fully parallel implementation of the GHM-FDMWT kernel or the GHM-IDMWT kernel using schoolbook (direct) method requires two FGPA-chips Spartan-3 XC3S200, while the implementation of these kernels with the help of our proposed algorithms occupies only one such chip.

5. References

[1] Ţariov A., Ţariova G., Majorkowska-Mech D.: Algorytmy wielopoziomowej dekompozycji oraz rekonstrukcji sygnałów cyfrowych. Wydawnictwo Komisji Informatyki GO PAN. 2012.

[2] Strela V., Heller P. N., Strang G., Topiwala P. and Heil C.: The application of multiwavelet filterbanks to image processing. IEEE Trans. on Image Processing, vol. 8, no. 4, pp. 548-563, 1999.

[3] Cotronei M., Montefusco L. B. and Puccio L.: Multiwavelet analysis and signal processing. IEEE Trans. Circuits Syst. II, vol. 45, no. 8, pp. 970-985, 1998.

[4] Sawant V.P.: Fusion Algorithm for Images based on Discrete Multi-wavelet Transform. IOSR Journal of VLSI and Signal Processing (IOSR-JVSP), vol. 2, No 3, pp. 22-27, 2013.

[5] Saleem M. R., Taha Walid A. Mahmood: New techniques for Daubechies wavelets and multiwavelets implementation using quantum computing. Facta Universitatis Series: Electronics and Energetics vol. 26, No 2, August 2013, pp. 145 – 156, 2013.

[6] Mohammed Aboud Kadhim, Saleim Hachem Farhan, Maher Ibraheem Gamaj: Design and Improvement of HiperLAN/2 Physical Layer Model Based Multiwavelet Signals. Innovative Systems Design and Engineering, vol.5, No.6, 2014, pp. 20-27, 2014.

[7] Sameer A. Dawood, Malek F., Anuar M. S., Rashid A. Fayadh, Farah Salwani Abdullah and Mohd Hariz Mohd Fakri: Improve the Performance of Multi-users MC-CDMA Based on Critically Sampling Multi-wavelet Transform over Wireless Propagation Channel. PIERS Proc., Guangzhou, China, August 25-28, 2014, pp. 1958-1962, 2014.

[8] Mayurakshi Roy Medhi and Kandarpa Kumar Sarma: Multiwavelet based MC-CDMA System to Track MIMO Channel Variations. Advances in Automatic Control: Proceedings of the 16th International Conference on Automatic Control Modelling & Simulation, Brasov, (Romania), June 26-28, 2014, pp. 257-261, 2014.

[9] Xiao-Bing Zhang, Yun-Hui Li and Xiao-Meng Cui: Active Power Measurement Based on Multiwavelet Transforms. Mathematical Problems in Engineering (Hindawi Publishing Corporation), vol. 2014, pp. 1-6, 2014.

[10] Geronimo J., Hardin D. and Massupost P.R.: Fractal Function and Wavelet Expansions Based on Several Functions. J. Approx. Theory, 78: pp. 373-401, 1994.

[11] Kiran S. P., Vinu T.: An algorithm for accelerating GHM Multiwavelet transformation on FPGA. International Conference on Data Science & Engineering (ICDSE), 2012, pp. 123 – 127, 2012.

[12] Rieder P., Simon S., Schimpfle C. V.: Application specific efficient VLSI architectures for orthogonal single- and multiwavelet transforms. Journal of VLSI signal processing systems for signal, image and video technology, June 1999, vol. 21, No 2, pp. 77-90, 1999.

[13] Anitha.K, Dharmistan K. Varugheese: Area Optimized High Throughput IDMWT/DMWT Processor for OFDM on Virtex-5 FPGA. I.J. Image, Graphics and Signal Processing, No 9, pp. 23-29, 2012.

[14] Veena M. B and Shanmukha Swamy M. N.: Design & Implementation of Optimized DMWT Architecture for OFDM on FPGA. Intern. Conf. on Innovations in Electrical and Electronics Engineering (ICIEE'2012) Oct. 6-7, 2012 Dubai (UAE), pp. 263-267, 2012.

[15] Cariow A., Cariowa G.: Algorithmic tricks for reducing the complexity of FDWT/IDWT basic operations implementation. I. J. Image, Graphics and Signal Processing, No 1, pp. 25-32, 2009.

Prof. Aleksandr CARIOW, DSc, PhD

He received the Candidate of Sciences (PhD) and Doctor of Sciences degrees (DSc or Habilitation) in Computer Sciences from LITMO of St. Petersburg, Russia in 1984 and 2001. In September 1999, he joined the faculty of Computer Sciences at the West Pomeranian University of Technology, Szczecin, Poland, where he is currently a professor and chair of the Department of Computer Architectures and Telecommunications. His research interests include digital signal processing algorithms, VLSI architectures, and data processing parallelization.

e-mail: acariow@wi.zut.edu.pl

Galina CARIOWA, PhD

She received the MSc degree in mathematics from Moldavian State University, Chişinău in 1976 and PhD degree in computer science from West Pomeranian University of Technology, Szczecin, Poland in 2007. She is currently working as an assistant professor of the Department of Multimedia Systems. She is also an Associate-Editor of World Research Journal of Transactions on Algorithms. Her scientific interests include numerical linear algebra and digital signal processing algorithms, VLSI architectures.

e-mail: gcariowa@wi.zut.edu.pl