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

An FPGA-oriented fully parallel algorithm for multiplying dual quaternions

Abstract

This paper presents a low multiplicative complexity fully parallel algorithm for multiplying two dual quaternions. The "pen-and-paper" multiplication of two dual quaternions requires 64 real multiplications and 56 real additions. More effective solutions still do not exist. We show how to compute a product of two dual quaternions with 24 real multiplications and 64 real additions. During synthesis of the discussed algorithm we use the fact that the product of two dual quaternions can be represented as a matrix–vector product. The matrix multiplicand that participates in the product calculating has unique structural properties that allow performing its advantageous factorization. Namely this factorization leads to significant reducing of the multiplicative complexity of dual quaternion multiplication. We show that by using this approach, the computational process of calculating dual quaternion product can be structured so that eventually requires only half the number of multipliers compared to the direct implementation of matrix-vector multiplication.

Keywords: dual quaternion product, fast algorithms, hardware complexity reduction, FPGA implementation.

1. Introduction

Today, hypercomplex algebras are increasingly being used to enhance the effectiveness of the solution of problems in various scientific and technological areas. Dual quaternions, in particular, are widely used in biomechanics, robotics, skeletal animation and many other applications of 3D computer graphics applications that require data processing in real time [1-9].

It should be noted that in the implementation of numerical algorithms using hypercomplex representation of the data, the multiplication is the most time-consuming and labor-intensive. This is because the multiplication of two hypercomplex numbers requires performing many real multiplications and real additions. It is easy to verify that the complexity of such a multiplication is proportional to the square of its dimension. In particular, the multiplication of the dual quaternion requires 64 real multiplications and 56 real additions. Therefore to speed up the calculations, it is appropriate to use hardware FPGA-accelerators.

Most modern-day high-end FPGA targets contain a number of embedded dedicated multipliers. Thus, instead of mapping a multiplier into several logic gates, dedicated multipliers provided on the FPGA fabric can be used. So, all multiplications involved in an implementation of the fully parallel algorithm can efficiently be implemented using these embedded multipliers. However, their number may be simply not enough to meet demanded fully parallel implementation of the algorithm. The designer uses embedded multipliers to implement multiply operations until he occupies all the embedded multipliers. If the FPGA target runs out of embedded multipliers, the designer uses generic logic gates instead, and the multiplication implementation becomes expensive in terms of FPGA resource usage. In some cases, therefore, available logic has to be exploited to implement multipliers, seriously restricting the maximum number of real multiplications that can be implemented in parallel on a target device. This will lead to significant difficulties during implementation of the computation unit. Therefore the problem of reducing the number of multiplications in the fully-parallel hardware-oriented algorithms is critical.

2. Statement of the problem

We can represent any dual quaternion Q as an 8-tuple $(q_0, q_1, q_2, q_3, \tilde{q}_0, \tilde{q}_1, \tilde{q}_2, \tilde{q}_3)$. The following representation of Q reveals the products of the quaternion units and the dual units associated with each component of the 8-tuple.

$$Q = q_0 + iq_1 + jq_2 + kq_3 + \varepsilon\tilde{q}_0 + \varepsilon i\tilde{q}_1 + \varepsilon j\tilde{q}_2 + \varepsilon k\tilde{q}_3$$

where

$$i^2 = j^2 = k^2 = -1\,,\ ij = k = -ji\,,\ jk = i = -kj\,,$$

$$ki = j = -jk\,,\ \varepsilon^2 = 0\,.$$

The results of all possible products of dual quaternion imaginary units can be summarized in the following table [6]:

Tab. 1. Table of multiplication of dual quaternion imaginary units

	1	i	j	k	ε	εi	εj	εk
1	1	i	j	k	ε	εi	εj	εk
i	i	-1	k	$-j$	εi	$-\varepsilon$	εk	$-\varepsilon j$
j	j	$-k$	-1	i	εj	$-\varepsilon k$	$-\varepsilon$	εi
k	k	j	$-i$	-1	εk	εj	$-\varepsilon i$	$-\varepsilon$
ε	ε	εi	εj	εk	0	0	0	0
εi	εi	$-\varepsilon$	εk	$-\varepsilon j$	0	0	0	0
εj	εj	$-\varepsilon k$	$-\varepsilon$	εi	0	0	0	0
εk	εk	εj	$-\varepsilon i$	$-\varepsilon$	0	0	0	0

Suppose we have to compute the product of two dual quaternions

$$Q_3 = Q_1 Q_2\,,$$

where

$$Q_1 = x_0 + ix_1 + jx_2 + kx_3 + \varepsilon x_4 + \varepsilon ix_5 + \varepsilon jx_6 + \varepsilon kx_7\,,$$
$$Q_2 = b_0 + ib_1 + jb_2 + kb_3 + \varepsilon b_4 + \varepsilon ib_5 + \varepsilon jb_6 + \varepsilon kb_7\,,$$
$$Q_3 = y_0 + iy_1 + jy_2 + ky_3 + \varepsilon y_4 + \varepsilon iy_5 + \varepsilon jy_6 + \varepsilon ky_7\,.$$

We can see that the "pen-and-paper" method of multiplication of two dual quaternions requires 64 real multiplications and 56 real additions.

We affirm that the multiplication of two dual quaternions can be represented by the following matrix-vector product:

$$\mathbf{Y}_{8\times1} = \mathbf{B}_8 \mathbf{X}_{8\times1}\,, \tag{1}$$

where

$$\mathbf{X}_{8\times1} = [x_0, x_1, x_2, x_3.x_4, x_5, x_6, x_7]^{\mathrm{T}}\,,$$

$$\mathbf{Y}_{8\times1} = [y_0, y_1, y_2, y_3.y_4, y_5, y_6, y_7]^{\mathrm{T}}\,,$$

$$\mathbf{B}_8 = \left[\begin{array}{c|c} \mathbf{B}_4^{(0,0)} & \mathbf{B}_4^{(0,1)} \\ \hline \mathbf{B}_4^{(1,0)} & \mathbf{B}_4^{(1,1)} \end{array}\right],$$

$$\mathbf{B}_4^{(0,0)}=\mathbf{B}_4^{(1,1)}=\begin{bmatrix} b_0 & -b_1 & -b_2 & -b_3 \\ b_1 & b_0 & b_3 & -b_2 \\ b_2 & -b_3 & b_0 & b_1 \\ b_3 & b_2 & -b_1 & b_0 \end{bmatrix},$$

$$\mathbf{B}_4^{(1,0)}=\begin{bmatrix} b_4 & -b_5 & -b_6 & -b_7 \\ b_5 & b_4 & b_7 & -b_6 \\ b_6 & -b_7 & b_4 & b_5 \\ b_7 & b_6 & -b_5 & b_4 \end{bmatrix},\ \mathbf{B}_4^{(0,1)}=\mathbf{0}_4,$$

and $\mathbf{0}_{N\times M}$ is an $M\times N$ matrix of zeros (a matrix where every element is equal to zero).

Taking into account that the $\mathbf{B}_4^{(0,1)}=\mathbf{0}_4$, direct realization of (1) requires only 48 real multiplications and 40 real additions. Despite the fact that the computational complexity is reduced, the number of multiplications is still large. Below we shall present the algorithm, which reduces arithmetical complexity to 24 real multiplications and 64 real additions.

3. The algorithm

The proposed algorithm can be written with the help of the following matrix-vector calculating procedure:

$$\mathbf{Y}_{8\times1}=\mathbf{D}_8\mathbf{\Sigma}_{8\times16}\mathbf{W}_{16}\mathbf{\Sigma}_{16\times24}\mathbf{D}_{24}\mathbf{P}_{24\times20}\mathbf{W}_{20}\mathbf{P}_{20\times8}\mathbf{X}_{8\times1} \qquad (2)$$

where

$$\mathbf{P}_{20\times8}=[\mathbf{P}_{4\times8},\mathbf{I}_8,\mathbf{I}_8]^{\mathrm{T}},\ \mathbf{P}_{4\times8}=(\mathbf{I}_4\blacksquare|\blacksquare\mathbf{0}_4),$$

$$\mathbf{W}_{20}=\mathbf{I}_4\oplus\mathbf{\Delta}_8\oplus\mathbf{I}_8,\ \mathbf{\Delta}_8=\mathbf{H}_4\oplus\mathbf{H}_4,\ \mathbf{P}_{24\times20}=\mathop{\frac{\blacksquare}{\blacksquare}}_{i=0}^{2}\mathbf{P}_{8\times20}^{(i)},$$

$$\mathbf{P}_{8\times20}^{(0)}=(\mathbf{I}_8\blacksquare|\blacksquare\mathbf{0}_{12}),\ \mathbf{P}_{8\times20}^{(1)}=(\mathbf{0}_4\blacksquare|\blacksquare\mathbf{I}_8\blacksquare|\blacksquare\mathbf{0}_8),$$

$$\mathbf{P}_{8\times20}^{(2)}=(\mathbf{0}_{12}\blacksquare|\blacksquare\mathbf{I}_8),$$

$$\mathbf{\Sigma}_{16\times24}=\mathbf{I}_8\oplus(\mathbf{1}_{1\times2}\otimes\mathbf{I}_4)\oplus(\mathbf{1}_{1\times2}\otimes\mathbf{I}_4),$$

$$\mathbf{W}_{16}=\mathbf{I}_4\oplus\mathbf{\Delta}_8\oplus\mathbf{I}_4,\ \mathbf{\Sigma}_{8\times16}=(\mathbf{1}_{1\times2}\otimes\mathbf{I}_4)\oplus(\mathbf{1}_{1\times2}\otimes\mathbf{I}_4),$$

where $\mathbf{H}_4$ - is a Hadamard matrix of order 4, $\mathbf{1}_{M\times N}$ - is an $M\times N$ matrix of ones (a matrix where every element is equal to one), $\mathbf{I}_N$ - is an identity $N\times N$ matrix, signs „$\otimes$”, “$\oplus$”denote the Kronecker product and direct sum of two matrices, respectively, signs $\frac{\blacksquare}{\blacksquare}$, $\blacksquare|\blacksquare$ denote vertical and horizontal concatenation of the two or more matrices, respectively [10].

$$\mathbf{D}_8=\mathrm{diag}(-1,1,1,1,-1,1,1,1),\ \mathbf{D}_{24}=\mathrm{diag}(s_0,s_1,...,s_{23}).$$

If the elements of $\mathbf{D}_{24}$ are placed vertically without disturbing the order and written in the form of the vector $\mathbf{S}_{24\times1}$, then they can be calculated using the following vector-matrix procedure:

$$\mathbf{S}_{24\times1}=\widetilde{\mathbf{D}}_{24}\mathbf{P}_{24\times20}\widetilde{\mathbf{W}}_{20}\mathbf{P}_{20\times8}\mathbf{B}_{8\times1} \qquad (3)$$

$$\mathbf{B}_{8\times1}=[b_0,b_1,b_2,b_3,b_4,b_5,b_6,b_7]^{\mathrm{T}},\ \widetilde{\mathbf{W}}_{20}=\mathbf{P}_4\oplus\mathbf{\Delta}_8\oplus\mathbf{P}_8,$$

$$\mathbf{P}_4=\begin{bmatrix} 1 & & & \\ & & & 1 \\ & 1 & & \\ & & 1 & \end{bmatrix},\ \mathbf{P}_4=\begin{bmatrix} & & & & 1 & & & \\ & & & & & & & 1 \\ & & & & & 1 & & \\ & & & & & & 1 & \\ 1 & & & & & & & \\ & & & 1 & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \end{bmatrix}.$$

$$\widetilde{\mathbf{D}}_{24}=\mathrm{diag}(2\mathbf{I}_4,\tfrac{1}{4}\mathbf{I}_8,2\mathbf{I}_4).$$

Fig. 1 shows a data flow diagram of the new algorithm for multiplying two dual quaternions and Fig. 2 depicts a data flow diagram of the process for calculating the matrix $\widetilde{\mathbf{D}}_{24}$ entries. 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 lines converge denote summation. We use the usual lines without arrows on purpose, so as not to clutter the picture.

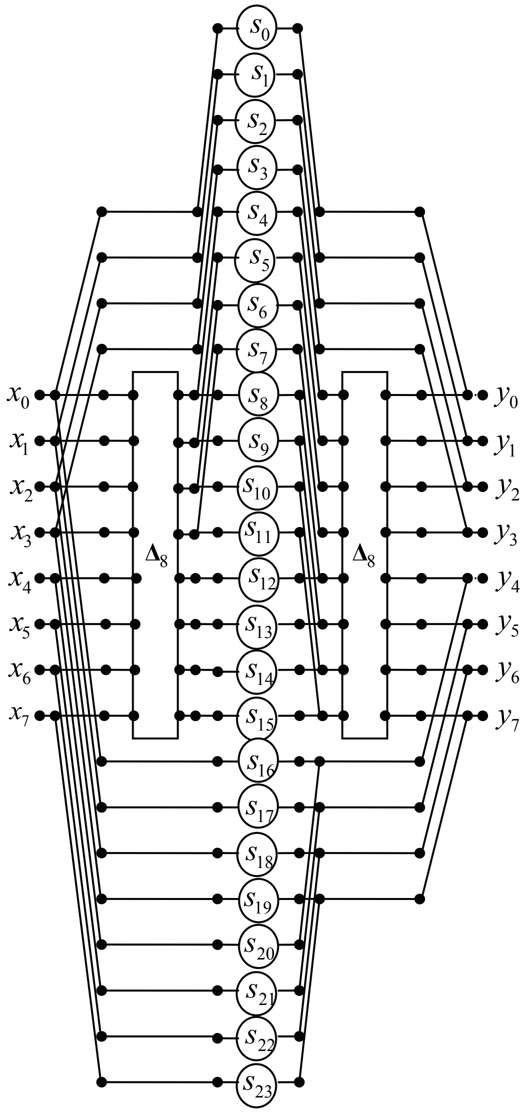

Fig. 1. The data flow diagram of the proposed algorithm

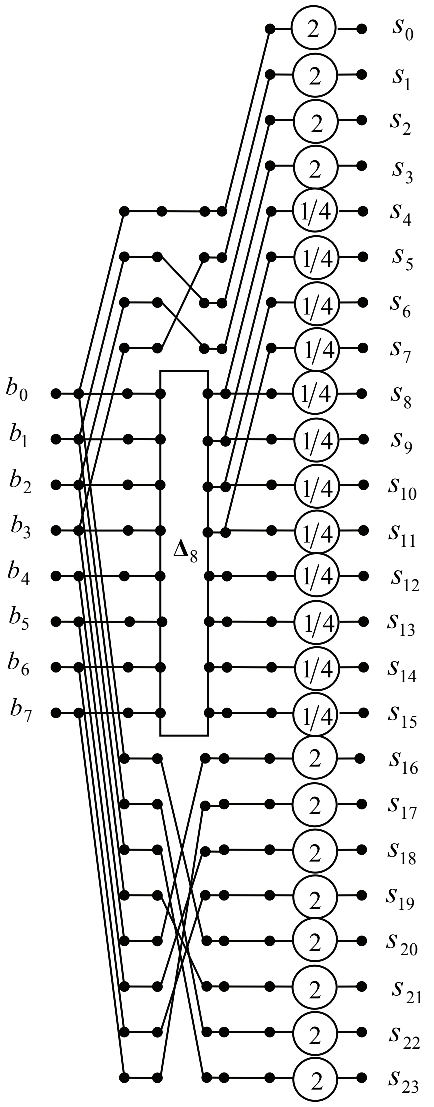

Fig. 2. The data flow diagram for calculating elements of the diagonal matrix $\mathbf{D}_{24}$

4. Conclusions

The paper presents a new FPGA-oriented algorithm for multiplying two dual quaternions. To reduce the hardware complexity (number of embedded multipliers), we exploit the specific properties of the matrix-vector product representation of dual quaternions multiplication. So, the algorithm requires 24 real multiplications and only 64 real additions (because multiplication of any vector by the matrix $\mathbf{H}_4$ takes only 8 real additions). A completely parallel implementation of dual quaternion multiplier using the schoolbook (direct) method requires three FGPA-chips Spartan-3 XC3S1000-4FT256, while the implementation of the proposed algorithm occupies only one such chip.

5. References

[1] Pennestrì E., Valentini P. P.: Dual quaternions as a tool for rigid body motion analysis: a tutorial with an application to biomechanics. Archive of Mechanical Engineering, vol. LVII, No 2, pp. 187–205. 2010.

[2] Jiang F., Wang H.-N, Huang Ch. S.: Algorithm for Relative Position and Attitude of Formation Flying Satellites Based on Dual Quaternion. Chinese Space Science and Technology, v32(3): pp. 20-26. 2012.

[3] Feng X., Wan W.: Real time skeletal animation with dual quaternion. Journal of Theoretical and Applied Information Technology. v. 49, no.1, pp. 356-362, 2013.

[4] Pham H. L, Perdereau V., Adorno B.V., and Fraisse P.: Position and orientation control of robot manipulators using dual quaternion feedback. IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, China. 18-22 October 210, pp. 658–663. 2010.

[5] Torsello A.: Point Invariance of the Screw Tension Minimizer. Università Ca'Foscari Venezia, Dipartimento di Scienze Ambientali Informatica e Statistica, Technical Report Series, Rapporto di Ricerca DAIS-2011-2, pp. 1-7, 2011.

[6] Kavan L., Collins S., Žára J. J., and O'Sullivan C.: Geometric skinning with approximate dual quaternion blending. ACM Trans. Graph., 27(4):105, pp. 2442-2443, 2008.

[7] Torsello A., Rodolà E., and Albarelli A.: Multiview Registration via Graph Diffusion of Dual Quaternions. IEEE Conference on Computer Vision and Pattern Recognition, 20-25 June 2011, pp. 2441-2448, 2011.

[8] Mukundan R.: Advanced Methods in Computer Graphics: With examples in OpenGL. Springer-Verlag London Limited, 2012.

[9] Kenwright B.: Dual-Quaternions: From Classical Mechanics to Computer Graphics and Beyond, 1-11, Source:www.xbdev.net

[10] Ţariov A.: Algorytmiczne aspekty racjonalizacji obliczeń w cyfrowym przetwarzaniu sygnałów, Wydawnictwo Zachodniopomorskiego Uniwersytetu Technologicznego, 2011.

Received: 02.04.2015 *Paper reviewed* *Accepted: 02.06.2015*

Prof. Aleksandr CARIOW, DSc, PhD

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

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

Galina CARIOWA, PhD

She received the MSc degrees 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. Her scientific interests include numerical linear algebra and digital signal processing algorithms, VLSI architectures, and data processing parallelization.

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

Mira WITCZAK, eng.

Mira is a fourth-year undergraduate student in West Pomeranian University of Technology, Faculty of Computer Science and Information Technology, Szczecin. She is a member of the Students' Scientific Circle "Quaternion", which studies the problems of processing and recognition signals and images. Her research interests lie in the development of theory and practice of digital signal and image processing algorithms, computer graphics, and multimedia programming. She expects to graduate in the spring of 2015.

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