Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/100106

An Algorithm for Multiplication of Two Biquaternions

Aleksandr Cariow[*], *Galina Cariowa and Anna Malewicz*

Faculty of Computer Science and Information Technology, West Pomeranian University of Technology, Szczecin, Żołnierska 49, 71-210 Szczecin, Poland

Received: 9 Jun. 2015, Revised: 7 Aug. 2015, Accepted: 8 Aug. 2015
Published online: 1 Jan. 2016

Abstract: In this paper we introduce efficient algorithm for the multiplication of biquaternions. The direct multiplication of two biquaternions requires 64 real multiplications and 56 real additions. More effective solutions still do not exist. We show how to compute a product of the Pauli numbers with 24 real multiplications and 64 real additions. During synthesis of the discussed algorithm we use the fact that product of two biquaternions may be represented as vector-matrix product. The matrix that participates in the product calculating has unique structural properties that allow performing its advantageous decomposition. Namely this decomposition leads to significant reducing of the computational complexity of biquaternion multiplication.

Keywords: biquaternion, multiplication of biquaternions, fast algorithm, matrix notation

1 Introduction

The Clifford and hypercomplex algebras [1] are seeing increased application to digital signal and image processing [2,3,4], computer graphics and machine vision [5,6,7], telecommunications [8,9] and in public key cryptography [10]. Preliminary studies show that when solving problems of data processing are often used quaternions and biquaternions or complexfield quaternions [11,12,13,14,15,16,17,18].

Among other arithmetical operations in the Clifford and hypercomplex algebras, multiplication is the most time consuming one. The reason for this is, because the usual multiplication of these numbers requires $N(N-1)$ real additions and N^2 real multiplication. It is easy to see that the increasing of dimension of hypernumber increases the computational complexity of the multiplication. Therefore, reducing the computational complexity of the multiplication of Clifford and hypercomplex numbers is an important theoretical and practical task. Efficient algorithms for the multiplication of quaternions, octonions and sedenions already exist [19, 20,21,22]. No such algorithms for the multiplication of the biquaternions have been proposed. In this paper, an efficient algorithm for this purpose is suggested.

2 Formulation of the problem

A biquaternion is defined as follows [11]

$$b = b_0 + b_1 e_1 + b_2 e_2 + b_3 e_3 + b_4 e_4 + b_5 e_5 + b_6 e_6 + b_7 e_7,$$

where $\{b_i\}$, $i = 0,1,\ldots,7$ are real numbers, and $\{e_j\}$, $j = 1,2,\ldots,7$ are imaginary units whose products are defined by the following table [12]:

$\times$	e_1	e_2	e_3	e_4	e_5	e_6	e_7
e_1	-1	e_3	$-e_2$	e_5	$-e_4$	e_7	$-e_6$
e_2	$-e_3$	-1	e_1	e_6	$-e_7$	$-e_4$	e_5
e_3	e_2	$-e_1$	-1	e_7	e_6	$-e_5$	$-e_4$
e_4	e_5	e_6	e_7	-1	$-e_1$	$-e_2$	$-e_3$
e_5	$-e_4$	e_7	$-e_6$	$-e_1$	1	$-e_3$	e_2
e_6	$-e_7$	$-e_4$	e_5	$-e_2$	e_3	1	$-e_1$
e_7	e_6	$-e_5$	$-e_4$	$-e_3$	$-e_2$	e_1	1

Suppose we must to compute the product of two biquaternions $\tilde{b}_3 = \tilde{b}_1 \tilde{b}_2$, where

$$\tilde{b}_1 = x_0 + x_1 e_1 + x_2 e_2 + x_3 e_3 + x_4 e_4 + x_5 e_5 + x_6 e_6 + x_7 e_7,$$
$$\tilde{b}_2 = b_0 + b_1 e_1 + b_2 e_2 + b_3 e_3 + b_4 e_4 + b_5 e_5 + b_6 e_6 + b_7 e_7,$$
$$\tilde{b}_3 = y_0 + y_1 e_1 + y_2 e_2 + y_3 e_3 + y_4 e_4 + y_5 e_5 + y_6 e_6 + y_7 e_7.$$

[*] Corresponding author e-mail: atariov@wi.zut.edu.pl

Using “pen and paper” method we can write:

$$\begin{aligned}\tilde{b}_3 = {} & x_0b_0 + x_0b_1e_1 + x_0b_2e_2 + x_0b_3e_3 + \\ & + x_0b_4e_4 + x_0b_5e_5 + x_0b_6e_6 + x_0b_7e_7 + \\ & + x_1b_0e_1 + x_1b_1e_1e_1 + x_1b_2e_1e_2 + x_1b_3e_1e_3 + \\ & + x_1b_4e_1e_4 + x_1b_5e_1e_5 + x_1b_6e_1e_6 + x_1b_7e_1e_7 + \\ & + x_2b_0e_2 + x_2b_1e_2e_1 + x_2b_2e_2e_2 + x_2b_3e_2e_3 + \\ & + x_2b_4e_2e_4 + x_2b_5e_2e_5 + x_2b_6e_2e_6 + x_2b_7e_2e_7 + \\ & + x_3b_0e_3 + x_3b_1e_3e_1 + x_3b_2e_3e_2 + x_3b_3e_3e_3 + \\ & + x_3b_4e_3e_4 + x_3b_5e_3e_5 + x_3b_6e_3e_6 + x_3b_7e_3e_7 + \\ & + x_4b_0e_4 + x_4b_1e_4e_1 + x_4b_2e_4e_2 + x_4b_3e_4e_3 + \\ & + x_4b_4e_4e_4 + x_4b_5e_4e_5 + x_4b_6e_4e_6 + x_4b_7e_4e_7 + \\ & + x_5b_0e_5 + x_5b_1e_5e_1 + x_5b_2e_5e_2 + x_5b_3e_5e_3 + \\ & + x_5b_4e_5e_4 + x_5b_5e_5e_5 + x_5b_6e_5e_6 + x_5b_7e_5e_7 + \\ & + x_6b_0e_6 + x_6b_1e_6e_1 + x_6b_2e_6e_2 + x_6b_3e_6e_3 + \\ & + x_6b_4e_6e_4 + x_6b_5e_6e_5 + x_6b_6e_6e_6 + x_6b_7e_6e_7 + \\ & + x_7b_0e_7 + x_7b_1e_7e_1 + x_7b_2e_7e_2 + x_7b_3e_7e_3 + \\ & + x_7b_4e_7e_4 + x_7b_5e_7e_5 + x_7b_6e_7e_6 + x_7b_7e_7e_7.\end{aligned}$$

Then we have:

$$\begin{aligned}& y_0 = \\ & = x_0b_0 - x_1b_1 - x_2b_2 - x_3b_3 - x_4b_4 + x_5b_5 + x_6b_6 + x_7b_7, \\ & y_1 = \\ & = x_0b_1 + x_1b_0 + x_2b_3 - x_3b_2 - x_4b_5 - x_5b_4 - x_6b_7 + x_7b_6, \\ & y_2 = \\ & = x_0b_2 - x_1b_3 + x_2b_0 + x_3b_1 - x_4b_6 + x_5b_7 - x_6b_4 - x_7b_5, \\ & y_3 = \\ & = x_0b_3 + x_1b_2 - x_2b_1 + x_3b_0 - x_4b_7 - x_5b_6 + x_6b_5 - x_7b_4, \\ & y_4 = \\ & = x_0b_4 - x_1b_5 - x_2b_6 - x_3b_7 + x_4b_0 - x_5b_1 - x_6b_2 - x_7b_3, \\ & y_5 = \\ & = x_0b_5 + x_1b_4 + x_2b_7 - x_3b_6 + x_4b_1 + x_5b_0 + x_6b_3 - x_7b_2, \\ & y_6 = \\ & = x_0b_6 - x_1b_7 + x_2b_4 + x_3b_5 + x_4b_2 - x_5b_3 + x_6b_0 + x_7b_1, \\ & y_7 = \\ & = x_0b_7 + x_1b_6 - x_2b_5 + x_3b_4 + x_4b_3 + x_5b_2 - x_6b_1 + x_7b_0.\end{aligned}$$

We can see that the schoolbook method of multiplication of two biquaternions requires 64 real multiplications and 56 real additions.

Using the matrix notation, we can rewrite the above relations as follows:

$$\mathbf{Y}_{8\times1} = \mathbf{B}_8\mathbf{X}_{8\times1} \qquad (1)$$

where

$$\mathbf{X}_{8\times1} = [x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7]^{\mathrm{T}},$$
$$\mathbf{Y}_{8\times1} = [y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7]^{\mathrm{T}},$$

$$\mathbf{B}_8 = \left[\begin{array}{cccc:cccc} b_0 & -b_1 & -b_2 & -b_3 & -b_4 & b_5 & b_6 & b_7 \\ b_1 & b_0 & b_3 & -b_2 & -b_5 & -b_4 & -b_7 & b_6 \\ b_2 & -b_3 & b_0 & b_1 & -b_6 & b_7 & -b_4 & -b_5 \\ b_3 & b_2 & -b_1 & b_0 & -b_7 & -b_6 & b_5 & -b_4 \\ \hdashline b_4 & -b_5 & -b_6 & -b_7 & b_0 & -b_1 & -b_2 & -b_3 \\ b_5 & b_4 & b_7 & -b_6 & b_1 & b_0 & b_3 & -b_2 \\ b_6 & -b_7 & b_4 & b_5 & b_2 & -b_3 & b_0 & b_1 \\ b_7 & b_6 & -b_5 & b_4 & b_3 & b_2 & -b_1 & b_0 \end{array}\right].$$

The direct realization of (1) requires 64 real multiplications and 56 real additions too. We shall present the algorithm, which reduce arithmetical complexity to 24 real multiplications and 64 real additions.

3 The algorithm

At first, we rearrange the rows of the matrix $\mathbf{B}_8$ according to the following rule of ordering (1, 2, 3, 4, 5, 6, 7, 8) $\rightarrow$ (5, 6, 4, 3, 1, 2, 8, 7). Next, we rearrange the columns of obtained matrix according to the following rule of ordering (1, 2, 3, 4, 5, 6, 7, 8) $\rightarrow$ (1, 2, 8, 7, 5, 6, 4, 3). The next step of modification of the obtained matrix is to perform some artificial transformations which, as we see latter, will bring to minimizing the computational complexity of the final algorithm. Multiply by (-1) the fifth and sixth rows of this matrix and than multiply by (-1) the fifth and sixth columns of obtained matrix. We can easily see that this transformation leads in the future to minimize the computational complexity of the final algorithm. As a result, we obtain the following matrix:

$$\mathbf{B}'_8 = \left[\begin{array}{cccc:cccc} b_4 & -b_5 & -b_3 & -b_2 & -b_0 & b_1 & -b_7 & -b_6 \\ b_5 & b_4 & -b_2 & b_3 & -b_1 & -b_0 & -b_6 & b_7 \\ b_3 & b_2 & -b_4 & b_5 & b_7 & b_6 & b_0 & -b_1 \\ b_2 & -b_3 & -b_5 & -b_4 & b_6 & -b_7 & b_1 & b_0 \\ \hdashline -b_0 & b_1 & -b_7 & -b_6 & -b_4 & b_5 & b_3 & b_2 \\ -b_1 & -b_0 & -b_6 & b_7 & -b_5 & -b_4 & b_2 & -b_3 \\ b_7 & b_6 & b_0 & -b_1 & -b_3 & -b_2 & b_4 & -b_5 \\ b_6 & -b_7 & b_1 & b_0 & -b_2 & b_3 & b_5 & b_4 \end{array}\right].$$

Then we can write

$$\mathbf{B}'_8 = \mathbf{R}_8\mathbf{P}_8^{(1)}\mathbf{B}_8\mathbf{P}_8^{(2)}\mathbf{R}_8,$$

and

$$\mathbf{Y}_{8\times1} = \mathbf{P}_8^{(1)}\mathbf{R}_8\mathbf{B}'_8\mathbf{R}_8\mathbf{P}_8^{(2)}\mathbf{X}_{8\times1} \qquad (2)$$

where

$$\mathbf{P}_8^{(1)} = \left[\begin{array}{cccc:cccc} & & & & 1 & & & \\ & & & & & 1 & & \\ & & & 1 & & & & \\ & & 1 & & & & & \\ \hdashline 1 & & & & & & & \\ & 1 & & & & & & \\ & & & & & & & 1 \\ & & & & & & 1 & \end{array}\right],$$

$$\mathbf{P}_8^{(2)} = \left[\begin{array}{cccc|cccc} 1 &&&&&&& \\ & 1 &&&&&& \\ &&&&&&& 1 \\ &&&&&& 1 & \\ \hline &&&& 1 &&& \\ &&&&& 1 && \\ &&& 1 &&&& \\ && 1 &&&&& \end{array}\right],$$

$$\mathbf{R}_8 = \left[\begin{array}{cccc|cccc} 1 &&&&&&& \\ & 1 &&&&&& \\ && 1 &&&&& \\ &&& 1 &&&& \\ \hline &&&& -1 &&& \\ &&&&& -1 && \\ &&&&&& 1 & \\ &&&&&&& 1 \end{array}\right].$$

Now the matrix $\mathbf{B}'_8$ has a unique block structure:

$$\mathbf{B}'_8 = \left[\begin{array}{c|c} \mathbf{A}_4 & \mathbf{B}_4 \\ \hline \mathbf{B}_4 & -\mathbf{A}_4 \end{array}\right],$$

where

$$\mathbf{A}_4 = \left[\begin{array}{cc|cc} b_4 & -b_5 & -b_3 & -b_2 \\ b_5 & b_4 & -b_2 & b_3 \\ \hline b_3 & b_2 & -b_4 & b_5 \\ b_2 & -b_3 & -b_5 & -b_4 \end{array}\right], \quad \mathbf{B}_4 = \left[\begin{array}{cc|cc} -b_0 & b_1 & -b_7 & -b_6 \\ -b_1 & -b_0 & -b_6 & b_7 \\ \hline b_7 & b_6 & b_0 & -b_1 \\ b_6 & -b_7 & b_1 & b_0 \end{array}\right].$$

It is easily verify [22] that the matrix with this structure can be factorized, than the computational procedure for multiplication of the biquaternions can be represented as follows:

$$\mathbf{Y}_{8\times1} = \mathbf{P}_8^{(1)}\mathbf{R}_8\mathbf{W}_{8\times12}\mathbf{D}_{12}\mathbf{W}_{12\times8}\mathbf{R}_8\mathbf{P}_8^{(2)}\mathbf{X}_{8\times1} \qquad (3)$$

$$\mathbf{W}_{8\times12} = (\mathbf{T}_{2\times3}\otimes\mathbf{I}_4) = \left[\begin{array}{cccccccc|cccc} 1 &&&&&&&& 1 &&& \\ & 1 &&&&&&&& 1 && \\ && 1 &&&&&&&& 1 & \\ &&& 1 &&&&&&&& 1 \\ \hline &&&& 1 &&&& 1 &&& \\ &&&&& 1 &&&& 1 && \\ &&&&&& 1 &&&& 1 & \\ &&&&&&& 1 &&&& 1 \end{array}\right],$$

$$\mathbf{W}_{8\times12} = (\mathbf{T}_{2\times3}\otimes\mathbf{I}_4) = \left[\begin{array}{cccc|cccc} 1 &&&&&&& \\ & 1 &&&&&& \\ && 1 &&&&& \\ &&& 1 &&&& \\ &&&& 1 &&& \\ &&&&& 1 && \\ &&&&&& 1 & \\ &&&&&&& 1 \\ \hline 1 &&&& 1 &&& \\ & 1 &&&& 1 && \\ && 1 &&&& 1 & \\ &&& 1 &&&& 1 \end{array}\right],$$

$$\mathbf{T}_{2\times3} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \quad \mathbf{T}_{3\times2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix},$$

$$\mathbf{D}_{12} = diag\left[\begin{array}{c} \mathbf{A}_4-\mathbf{B}_4 \\ \hline -(\mathbf{A}_4+\mathbf{B}_4) \\ \hline \mathbf{B}_4 \end{array}\right],$$

$\mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ – is the order 2 Hadamard matrix, $\mathbf{I}_N$ – is the order N identity matrix, and "$\otimes$" – denotes the Kronecker product of two matrices [23].

Indeed, it is easy to see that the matrices $(\mathbf{A}_4-\mathbf{B}_4)$, $-(\mathbf{A}_4+\mathbf{B}_4)$ and $\mathbf{B}_4$ have the following structures:

$$(\mathbf{A}_4-\mathbf{B}_4) = \left[\begin{array}{cc|cc} b_4+b_0 & -b_5-b_1 & -b_3+b_7 & -b_2+b_6 \\ b_5+b_1 & b_4+b_0 & -b_2+b_6 & b_3-b_7 \\ \hline b_3-b_7 & b_2-b_6 & -b_4-b_0 & b_5+b_1 \\ b_2-b_6 & -b_3+b_7 & -b_5-b_1 & -b_4-b_0 \end{array}\right] = \left[\begin{array}{c|c} \mathbf{A}_2 & \mathbf{B}_2 \\ \hline -\mathbf{B}_2 & -\mathbf{A}_2 \end{array}\right],$$

$$-(\mathbf{A}_4+\mathbf{B}_4) = \left[\begin{array}{cc|cc} -b_4+b_0 & b_5-b_1 & b_3+b_7 & b_2+b_6 \\ -b_5+b_1 & -b_4+b_0 & b_2+b_6 & -b_3-b_7 \\ \hline -b_3-b_7 & -b_2-b_6 & b_4-b_0 & -b_5+b_1 \\ -b_2-b_6 & b_3+b_7 & b_5-b_1 & b_4-b_0 \end{array}\right] = \left[\begin{array}{c|c} \mathbf{C}_2 & \mathbf{D}_2 \\ \hline -\mathbf{D}_2 & -\mathbf{C}_2 \end{array}\right],$$

$$\mathbf{B}_4 = \left[\begin{array}{cc|cc} -b_0 & b_1 & -b_7 & -b_6 \\ -b_1 & -b_0 & -b_6 & b_7 \\ \hline b_7 & b_6 & b_0 & -b_1 \\ b_6 & -b_7 & b_1 & b_0 \end{array}\right] = \left[\begin{array}{c|c} \mathbf{E}_2 & \mathbf{F}_2 \\ \hline -\mathbf{F}_2 & -\mathbf{E}_2 \end{array}\right],$$

where

$$\mathbf{A}_2 = \left[\begin{array}{c|c} b_4+b_0 & -b_5-b_1 \\ \hline b_5+b_1 & b_4+b_0 \end{array}\right],$$

$$\mathbf{B}_2 = \left[\begin{array}{c|c} -b_3+b_7 & -b_2+b_6 \\ \hline -b_2+b_6 & b_3-b_7 \end{array}\right],$$

$$\mathbf{C}_2 = \left[\begin{array}{c|c} -b_4+b_0 & b_5-b_1 \\ \hline -b_5+b_1 & -b_4+b_0 \end{array}\right],$$

$$\mathbf{D}_2 = \left[\begin{array}{c|c} b_3+b_7 & b_2+b_6 \\ \hline b_2+b_6 & -b_3-b_7 \end{array}\right],$$

$$\mathbf{E}_2 = \left[\begin{array}{c|c} -b_0 & b_1 \\ \hline -b_1 & -b_0 \end{array}\right],$$

$$\mathbf{F}_2 = \left[\begin{array}{c|c} -b_7 & -b_6 \\ \hline -b_6 & b_7 \end{array}\right].$$

As shown in [22], the matrices having such block structures can be effectively factorized too.

$$\left[\begin{array}{c:c} \mathbf{A}_2 & \mathbf{B}_2 \\ \hdashline -\mathbf{B}_2 & -\mathbf{A}_2 \end{array}\right] = [\mathbf{I}_2 \oplus (-\mathbf{I}_2)]\,(\mathbf{H}_2\otimes\mathbf{I}_2)\times \times \frac{1}{2}diag\left[\begin{array}{c}\mathbf{A}_2+\mathbf{B}_2\\ \hdashline \mathbf{A}_2-\mathbf{B}_2\end{array}\right](\mathbf{H}_2\otimes \mathbf{I}_2) \tag{4}$$

$$\left[\begin{array}{c:c} \mathbf{C}_2 & \mathbf{D}_2 \\ \hdashline -\mathbf{D}_2 & -\mathbf{C}_2 \end{array}\right] = [\mathbf{I}_2 \oplus (-\mathbf{I}_2)]\,(\mathbf{H}_2\otimes\mathbf{I}_2)\times \times \frac{1}{2}diag\left[\begin{array}{c}\mathbf{C}_2+\mathbf{D}_2\\ \hdashline \mathbf{C}_2-\mathbf{D}_2\end{array}\right](\mathbf{H}_2\otimes \mathbf{I}_2) \tag{5}$$

$$\left[\begin{array}{c:c} \mathbf{E}_2 & \mathbf{F}_2 \\ \hdashline -\mathbf{F}_2 & -\mathbf{E}_2 \end{array}\right] = [\mathbf{I}_2 \oplus (-\mathbf{I}_2)]\,(\mathbf{H}_2\otimes\mathbf{I}_2)\times \times \frac{1}{2}diag\left[\begin{array}{c}\mathbf{E}_2+\mathbf{F}_2\\ \hdashline \mathbf{E}_2-\mathbf{F}_2\end{array}\right](\mathbf{H}_2\otimes \mathbf{I}_2) \tag{6}$$

where "$\oplus$" denotes the direct sum of two matrices [23].

Substituting (4), (5) and (6) in (3) we can write:

$$\mathbf{Y}_{8\times1} = \mathbf{P}_8^{(1)}\mathbf{R}_8\mathbf{W}_{8\times12}\mathbf{E}_{12}\mathbf{W}_{12}\times \times \mathbf{D}'_{12}\mathbf{W}_{12}\mathbf{W}_{12\times8}\mathbf{R}_8\mathbf{P}_8^{(2)}\mathbf{X}_{8\times1} \tag{7}$$

where

$$\mathbf{E}_{12} = diag(1,1,-1,-1,1,1,-1,-1,1,1,-1,-1),$$

$$\mathbf{D}'_{12} = \frac{1}{2}diag\left[\begin{array}{c}\mathbf{A}_2+\mathbf{B}_2\\ \hdashline \mathbf{A}_2-\mathbf{B}_2\\ \hdashline \mathbf{C}_2+\mathbf{D}_2\\ \hdashline \mathbf{C}_2-\mathbf{D}_2\\ \hdashline \mathbf{E}_2+\mathbf{F}_2\\ \hdashline \mathbf{E}_2-\mathbf{F}_2\end{array}\right],$$

$$\mathbf{W}_{12} = \mathbf{I}_3\otimes(\mathbf{H}_2\otimes\mathbf{I}_2) =$$

$$= \left[\begin{array}{c:c:c} \begin{array}{cc:cc}1&0&1&0\\0&1&0&1\\ \hdashline 1&0&-1&0\\0&1&0&-1\end{array} & \mathbf{0}_4 & \mathbf{0}_4 \\ \hdashline \mathbf{0}_4 & \begin{array}{cc:cc}1&0&1&0\\0&1&0&1\\ \hdashline 1&0&-1&0\\0&1&0&-1\end{array} & \mathbf{0}_4 \\ \hdashline \mathbf{0}_4 & \mathbf{0}_4 & \begin{array}{cc:cc}1&0&1&0\\0&1&0&1\\ \hdashline 1&0&-1&0\\0&1&0&-1\end{array} \end{array}\right],$$

$$\mathbf{A}_2+\mathbf{B}_2 = \left[\begin{array}{c:c} b_4+b_0-b_3+b_7 & -b_5-b_1-b_2+b_6 \\ \hdashline b_5+b_1-b_2+b_6 & b_4+b_0+b_3-b_7 \end{array}\right],$$

$$\mathbf{A}_2-\mathbf{B}_2 = \left[\begin{array}{c:c} b_4+b_0+b_3-b_7 & -b_5-b_1+b_2-b_6 \\ \hdashline b_5+b_1+b_2-b_6 & b_4+b_0-b_3+b_7 \end{array}\right],$$

$$\mathbf{C}_2+\mathbf{D}_2 = \left[\begin{array}{c:c} -b_4+b_0+b_3+b_7 & b_5-b_1+b_2+b_6 \\ \hdashline -b_5+b_1+b_2+b_6 & -b_4+b_0-b_3-b_7 \end{array}\right],$$

$$\mathbf{C}_2-\mathbf{D}_2 = \left[\begin{array}{c:c} -b_4+b_0-b_3-b_7 & b_5-b_1-b_2-b_6 \\ \hdashline -b_5+b_1-b_2-b_6 & -b_4+b_0+b_3+b_7 \end{array}\right],$$

$$\mathbf{E}_2+\mathbf{F}_2 = \left[\begin{array}{c:c} -b_0-b_7 & b_1-b_6 \\ \hdashline -b_1-b_6 & -b_0+b_7 \end{array}\right],$$

$$\mathbf{E}_2-\mathbf{F}_2 = \left[\begin{array}{c:c} -b_0+b_7 & b_1+b_6 \\ \hdashline -b_1+b_6 & -b_0-b_7 \end{array}\right].$$

Introduce the following notation:

$$\begin{aligned}
c_0 &= 1/2(b_4+b_0-b_3+b_7),\\
c_1 &= 1/2(-b_5-b_1-b_2+b_6),\\
c_2 &= 1/2(b_5+b_1-b_2+b_6),\\
c_3 &= 1/2(b_4+b_0+b_3-b_7),\\
c_4 &= 1/2(b_4+b_0+b_3-b_7),\\
c_5 &= 1/2(-b_5-b_1+b_2-b_6),\\
c_6 &= 1/2(b_5+b_1+b_2-b_6),\\
c_7 &= 1/2(b_4+b_0-b_3+b_7),\\
c_8 &= 1/2(-b_4+b_0+b_3+b_7),\\
c_9 &= 1/2(b_5-b_1+b_2+b_6),\\
c_{10} &= 1/2(-b_5+b_1+b_2+b_6),\\
c_{11} &= 1/2(-b_4+b_0-b_3-b_7),\\
c_{12} &= 1/2(-b_4+b_0-b_3-b_7),\\
c_{13} &= 1/2(b_5-b_1-b_2-b_6),\\
c_{14} &= 1/2(-b_5+b_1-b_2-b_6),\\
c_{15} &= 1/2(-b_4+b_0+b_3+b_7),\\
c_{16} &= 1/2(-b_0-b_7),\\
c_{17} &= 1/2(b_1-b_6),\\
c_{18} &= 1/2(-b_1-b_6),\\
c_{19} &= 1/2(-b_0+b_7),\\
c_{20} &= 1/2(-b_0+b_7),\\
c_{21} &= 1/2(b_1+b_6),\\
c_{22} &= 1/2(-b_1+b_6),\\
c_{23} &= 1/2(-b_0-b_7).
\end{aligned}$$

Using the above notations and combining partial decompositions in a single computational procedure we finally can write following:

$$\mathbf{Y}_{8\times1} = \tilde{\mathbf{P}}_8^{(1)}\tilde{\mathbf{W}}_{8\times12}\mathbf{W}_{12}\mathbf{A}_{12\times24}\mathbf{D}_{24}\times \times \mathbf{P}_{24\times12}\mathbf{W}_{12}\mathbf{W}_{12\times8}\tilde{\mathbf{P}}_8^{(2)}\mathbf{X}_{8\times1} \tag{8}$$

where

$$\tilde{\mathbf{P}}_8^{(1)} = \mathbf{P}_8^{(1)}\mathbf{R}_8,$$

$$\tilde{\mathbf{P}}_8^{(2)} = \mathbf{R}_8\mathbf{P}_8^{(2)},$$

$$\tilde{\mathbf{W}}_{8\times12} = \mathbf{W}_{8\times12}\mathbf{E}_{12},$$

$$\mathbf{P}_{24\times12} = \mathbf{I}_3\otimes\mathbf{I}_2\otimes\mathbf{1}_{2\times1}\otimes\mathbf{I}_2,$$

$$\mathbf{A}_{12\times24} = \mathbf{I}_3\otimes\mathbf{I}_4\otimes\mathbf{1}_{1\times2}$$

$$\mathbf{D}_{24} = diag(c_0, c_1, \ldots, c_{23}),$$

$$\mathbf{P}_{24\times 12}=$$

$$=\left[\begin{array}{c:c:c}
\begin{array}{c:c}\begin{array}{cc}1&\\&1\\1&\\&1\end{array}&\mathbf{0}_{4\times 2}\\ \hdashline \mathbf{0}_{4\times 2}&\begin{array}{cc}1&\\&1\\1&\\&1\end{array}\end{array} & \mathbf{0}_{8\times 4} & \mathbf{0}_{8\times 4}\\ \hdashline
\mathbf{0}_{8\times 4} & \begin{array}{c:c}\begin{array}{cc}1&\\&1\\1&\\&1\end{array}&\mathbf{0}_{4\times 2}\\ \hdashline \mathbf{0}_{4\times 2}&\begin{array}{cc}1&\\&1\\1&\\&1\end{array}\end{array} & \mathbf{0}_{8\times 4}\\ \hdashline
\mathbf{0}_{8\times 4} & \mathbf{0}_{8\times 4} & \begin{array}{c:c}\begin{array}{cc}1&\\&1\\1&\\&1\end{array}&\mathbf{0}_{4\times 2}\\ \hdashline \mathbf{0}_{4\times 2}&\begin{array}{cc}1&\\&1\\1&\\&1\end{array}\end{array}
\end{array}\right],$$

$$\mathbf{A}_{12\times 24}=\begin{bmatrix}1&1&&&&&&\\&&1&1&&&&\\&&&&1&1&&\\&&&&&\ddots&&\\&&&&&&1&1\end{bmatrix}.$$

We can see that the direct approach to calculation of elements $\{c_k\}$, $k=0,1,\ldots,23$ of the matrix $\mathbf{D}_{24}$ requires 56 additions. It is easy to see that the expressions for calculation of $\{c_k\}$ contain repeated algebraic sums. Therefore, the number of additions during calculation of these elements can be reduced.

So, it is easy to verify that the elements $\{c_k\}$, $k=0,1,\ldots,23$ can be calculated using the following rationalized vector-matrix procedure:

$$\mathbf{C}_{24\times 1}=\mathbf{P}_{24\times 12}\mathbf{A}_{12\times 8}\mathbf{D}_8(\mathbf{I}_4\otimes\mathbf{H}_2)\mathbf{B}_{8\times 1} \qquad (9)$$

where

$$\mathbf{B}_{8\times 1}=[b_0,b_1,b_2,b_3,b_4,b_5,b_6,b_7]^{\mathrm{T}},$$

$$\mathbf{C}_{24\times 1}=[c_0,c_1,\ldots,c_{23}]^{\mathrm{T}},\mathbf{D}_8=\frac{1}{2}\mathbf{I}_8,$$

$$\mathbf{A}_{12\times 8}=\left[\begin{array}{cccc:c}
-1&&&1&\\
&-1&1&&\\
&1&1&&\\
1&&&1&\mathbf{0}_{6\times 4}\\
&&-1&&\\
&&&-1&\\ \hdashline
&&&&\begin{array}{cccc}1&&&\\&1&&\\1&&&1\\&-1&1&\\&-1&-1&\\1&&&-1\end{array}\\
&&\mathbf{0}_{6\times 4}&&
\end{array}\right],$$

$$\mathbf{P}_{24\times 12}=\left[\begin{array}{cccccc:cccccc}
1&&&&&&&&&&&\\
1&&&&&&&&&&&\\
&1&&&&&&&&&&\\
&1&&&&&&&&&&\\
&&1&&&&&&&&&\\
&&1&&&&&&\mathbf{0}_{12\times 6}&&&\\
&&&1&&&&&&&&\\
&&&1&&&&&&&&\\
&&&&1&&&&&&&\\
&&&&1&&&&&&&\\
&&&&&1&&&&&&\\
&&&&&1&&&&&&\\ \hdashline
&&&&&&1&&&&&\\
&&&&&&-1&&&&&\\
&&&&&&&1&&&&\\
&&&&&&&-1&&&&\\
&&&&&&&&1&&&\\
&&\mathbf{0}_{12\times 6}&&&&&&-1&&&\\
&&&&&&&&&1&&\\
&&&&&&&&&-1&&\\
&&&&&&&&&&1&\\
&&&&&&&&&&-1&\\
&&&&&&&&&&&1\\
&&&&&&&&&&&-1
\end{array}\right],$$

$$\mathbf{P}_{24}=\left[\begin{array}{c:c}\mathbf{P}_{12}^{(0)}&\mathbf{P}_{12}^{(1)}\\ \hdashline \mathbf{P}_{12}^{(2)}&\mathbf{P}_{12}^{(3)}\end{array}\right],$$

$$\mathbf{P}_{12}^{(0)}=\left[\begin{array}{c:c:c:c:c:c:c:c:c:c:c:c}
&&&1&&&&&&&&\\ \hdashline
&&&&&&&&&&&\\ \hdashline
&&&&&&&&&&&\\ \hdashline
&&&&&&&1&&&&\\ \hdashline
&&&&&&1&&&&&\\ \hdashline
&&&&&&&&&&&\\ \hdashline
&&&&&&&&&&&\\ \hdashline
&&1&&&&&&&&&\\ \hdashline
&&&&&1&&&&&&\\ \hdashline
&&&&&&&&&&&\\ \hdashline
&&&&&&&&&&&\\ \hdashline
&1&&&&&&&&&&
\end{array}\right],$$

$$\mathbf{P}_{12}^{(1)} = \begin{bmatrix} & & & & & & & & & & & \\ & & & & & & & & 1 & & & \\ & & & & & & & & & & 1 & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & 1 \\ & & & & & & & & & 1 & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & 1 & & & & & \\ & & & & 1 & & & & & & & \\ & & & & & & & & & & & \end{bmatrix},$$

$$\mathbf{P}_{12}^{(2)} = \begin{bmatrix} 1 & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & 1 & & & & & & & \\ & & & & & & & & 1 & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & 1 & \\ & & & & & & & & & & & 1 \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & 1 & & \end{bmatrix},$$

$$\mathbf{P}_{12}^{(3)} = \begin{bmatrix} & & & & & & & & & & & \\ & & & & & 1 & & & & & & \\ & & & & & & & 1 & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & 1 & & & & & & & & & \\ & 1 & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ 1 & & & & & & & & & & & \\ & & & 1 & & & & & & & & \\ & & & & & & & & & & & \end{bmatrix}.$$

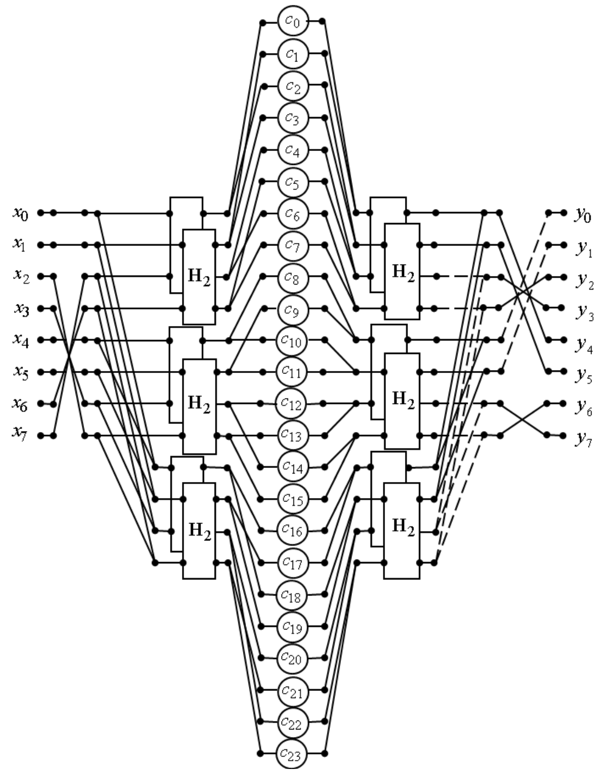


Fig. 1: Data flow diagram of rationalized algorithm for multiplication of two biquaternions

Fig. 1 shows a data flow diagram, which describes the fast algorithm for computation of the biquaternions product and Fig. 2 shows a data flow diagram of the process for calculating the vector $\mathbf{C}_{24\times 1}$ elements. In this paper, data flow diagrams are oriented from left to right.

Straight lines in the figures denote the operations of data transfer. Points where lines converge denote summation. The dash-dotted lines indicate the sign change operation. We deliberately use the usual lines without arrows on purpose, so as not to clutter the picture. The circles in these figures show the operation of multiplication by a variable (or constant) inscribed inside a circle. In turn, the rectangles indicate the matrix-vector multiplications with the order 2 Hadamard matrices. As follows from Fig. 2, calculation of elements of diagonal matrix $\mathbf{D}_8$ requires performing only trivial multiplications by the power of two. Such operations may be implemented as primitive shift operations, which have simple realization and hence may be neglected during computational complexity estimation [22].

4 Evaluation of computational complexity

We calculate how many real multiplications (excluding multiplications by power of two) and real additions are required for realization of the proposed algorithm, and compare it with the number of operations required for a direct evaluation of matrix-vector product in Eq. (1). As already mentioned the number of real multiplications required using the proposed algorithm is 24. Thus using the proposed algorithm the number of real multiplications to calculate the biquaternion product is significantly reduced. In the other hand the number of real additions required using our algorithm is 64. Thus, our proposed algorithms saves 40 multiplications but increases 8 additions compared with direct method. Therefore, the total number of arithmetic operations for proposed algorithm is approximately 27% less than that of the direct evaluation. It should be noted that in many practical applications, one of the biquaternions to be multiplied contains constant coefficients. In this case, the diagonal matrix elements can be precomputed. This would reduce the number of additions in the proposed algorithm to 56.

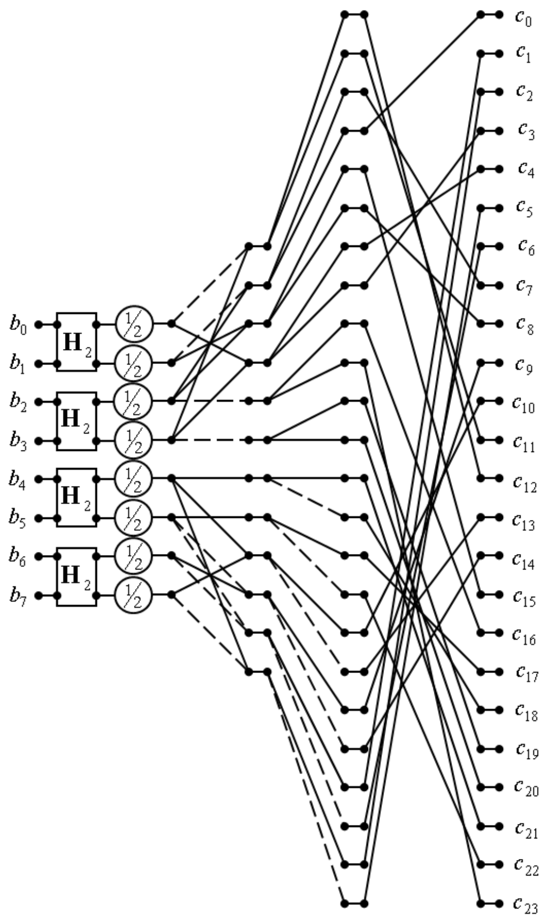


Fig. 2: Data flow diagram describing the process of calculating elements of the vector $\mathbf{C}_{24\times1}$ and in accordance with the procedure (9)

5 Conclusion

The article presents a new vectorized algorithm for the multiplication of two biquaternions. To reduce the number of real multiplications, we exploit the strategies of the synthesis of fast algorithms for the computation of the matrix-vector products [24]. Minimizing the number of multiplications is especially important in the design of specialized VLSI chips because reducing the number of two-component multipliers also reduces the power dissipation and lowers the power consumption of the entire system being implemented. This also results in a reduction in hardware implementation cost of ”biquaternion multiplier” on the one hand and allows to the effective use of parallelization of computations on the other hand. If the VLSI chip already contains embedded two-component multipliers, their number is always limited. This means that if the implemented algorithm contains a large number of multiplications, the developed processor may not always fit into the chip. So, the implementation of proposed in this paper algorithm on the base of VLSI chips, that possess embedded two-component multipliers, also allows saving the number of these blocks or realizing the biquaternion multiplier with the use of a smaller number of simpler and cheaper VLSI chips. It will enable to design of data processing units using a chips which contain a minimum required number of embedded two-component multipliers and thereby consume and dissipate least power.

So, we have presented an original algorithm which allows multiplying two biquaternions with reduced multiplicative complexity. As a result of streamlining the number of multiplications required to calculate the biquaternion product is reduced from 64 to 24 at the price of 8 more additions. Nevertheless, it should be noted that a hardware multiplier is more complicated unit than an adder and occupies much more chip area than the adder. Therefore, this solution is beneficial. Furthermore, the total number of arithmetic operations decreased by 32 compared with the naive method of calculations. Therefore, the proposed algorithm is better than the naive algorithm, even in terms of its software implementation on a conventional computer.

The authors are grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

References

[1] R. Abłamowicz (ed.), Clifford Algebras – Applications to Mathematics, Physics, and Engineering, PIM 34, Birkhauser, Basel, 2004.

[2] V. Labunets, Computational Noncommutative Algebra and Applications, NATO Science Series II: Mathematics, Physics and Chemistry **136**, 197-225, (2004).

[3] J. Mennesson, Ch. Saint-Jean, L. Mascarilla, Applied Geometric Algebras in Computer Science and Engineering **9**, 1-4, (2010).

[4] T. Batard, M. Berthier, and C. Saint-Jean. In E. Bayro-Corrochano and G. Scheuermann Eds, editors, Geometric Algebra Computing in Engineering and Computer Science, chapter **8**, 135-161, Springer Verlag, 2010.

[5] D. Hildenbrand , D. Fontijne , C. Perwass, L. Dorst, Tutorial 3, Proceedings of the 25th Annual Conference of the European Association for Computer Graphics, Interacting with Virtual Worlds, Grenoble, France, INRIA and Eurographics Association, ISSN 1017-4656.

[6] J. Ebling and G. Scheuermann, IEEE Transactions on Visualization and Computer Graphics **11**, 469-479, (2005).

[7] R. Wareham, J. Cameron, J. Lasenby, H. Li, P. J. Olver and G. Sommer (Eds.), IWMM 2004, LNCS **3519**, 329-349, (2005).

[8] S. Karmakar, B. S. Rajan, IEEE Transactions on Information Theory **55**, No. 1, 223-231, (2009).

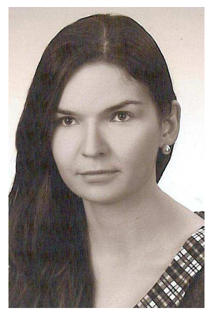
[9] M.Ye. Ilchenko, T.N. Narytnik, R.M. Didkovsky, Telecommunications and Radio Engineering **72**, 1651-1663, (2013).

[10] E. Malekian, A. Zakerolhosseini, A. Mashatan, The ISC Int’l Journal of Information Security **3**, No. 1, 29-42, (2011).

[11] A. A. Eliovich and V. I. Sanyuk, Theoretical and Mathematical Physics **162**, No. 2, 135-148, (2010).
[12] S. J. Sangwine, T. A. Ell, N. Le Bihan, Adv. Appl. Cliff ord Algebras **21**, 607-636 (2011).
[13] X.-F. Gong, Z.-W. Liu, Y.-G. Xu, Signal Processing **91**, 821-831 (2011).
[14] K. Nand, G. Hamarneh, R. Abugharbieh, 9th IEEE International Symposium on Biomedical Imaging (ISBI), 538-541 (2012).
[15] S. Said, N. Le Bihan, and S.J. Sangwine, IEEE Transactions on Signal Processing **56**, No. 4, 1522-1531, (2008).
[16] S. Miron, N. Le Bihan, Jérôme I. Mars, IEEE International Conference on Acoustics, Speech and Signal Processing, 1077-1080, (2006).
[17] E.P.J. de Haas, Apeiron **15**, No. 4, 358-381 (2008).
[18] S.-C. Pei, J.-J. Ding, 15th European Signal Processing Conference (EUSIPCO 2007), Poznan, Poland, 1337-1341, (2007).
[19] O. M. Makarov, Zh. Vychisl. Mat. Mat. Fiz. **17**, No. 6, 1574-1575 (1977).
[20] A. Cariow, G. Cariowa, Radioelectronics and Communications Systems (Allerton Press, Inc. USA) **55**, Issue 10, 464-473, (2012).
[21] A. Cariow, G. Cariowa, Information Processing Letters **113**, 324-331 (2013).
[22] A. Ţariov. Algorithmic aspects of computing rationalization in digital signal processing (in Polish), West Pomeranian University Press, 2011.
[23] W. H. Steeb, Y. Hardy, World Scientific Publishing Company. 2 edition, 2011.
[24] A. Cariow, Journal of Signal Processing Theory and Applications **3**, No. 1, 1-19 (2014).

Aleksandr Cariow 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, respectively. 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.

Galina Cariowa received the MSc degrees in mathematics from Moldavian State University, Chişinău in 1978 and PhD degree in computer science from West Pomeranian University of Technology, Szczecin, Poland in 2007. She is currently working as an assistant professor in the Department of Multimedia Systems. She is also an Associate-Editor of the World Research Journal of Transactions on Algorithms. Her scientific interests include numerical linear algebra and digital signal processing algorithms, VLSI architectures, and data processing parallelization.



Anna Malewicz received the Eng. degree in Electronics and Telecommunications and MSc degree in Computer Science from West Pomeranian University of Technology, Szczecin, Poland in 2012 and 2013 respectively. She is currently PhD student in the Department of Computer Architectures and Telecommunications. Her scientific interests include artificial intelligence, digital signal and image processing algorithms, programming languages and biomedical informatics. She is currently working on software quality assurance.