Cent. Eur. J. Comp. Sci. • 2(4) • 2012 • 389-397
DOI: 10.2478/s13537-012-0031-7

VERSITA

Central European Journal of **Computer Science**

Rationalized algorithm for computing the product of two sedenions

Research Article

Aleksandr Cariow*, Galina Cariowa†

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

Received 2 February 2012; accepted 18 September 2012

Abstract: In this paper we introduce efficient algorithm for the multiplication of sedenions. The direct multiplication of two sedenions requires 256 real multiplications and 240 real additions. We show how to compute a sedenions product with 120 real multiplications and 344 real additions.

Keywords: sedenions • multiplication of two sedenions • matrix notation

1. Introduction

Continuous scientific, technical and technological progress are continuously creating new challenges for researchers. Those tasks are difficult to solve with old, traditional methods. More and more frequently, to achieve precise goals and better results in processing data, requires the application of complex, advanced techniques and mathematical formalisms. When solving the most complicated data processing tasks it may be necessary to move beyond the arithmetic of real numbers, and apply the arithmetic of complex or hypercomplex numbers.

Hypercomplex number systems [9] are currently used in many areas, such as modeling processes in physics, construction of complicated solids and surfaces in geometry, signal and image processing, computer graphics, telecommunication, and mobile objects steering systems. Hypercomplex number systems proved useful due to their use in operations such as Discrete Fourier Transform, convolution, correlation and filtering of one-, two- and higher-dimensional signals [1–3, 5, 7, 11, 15, 18–21, 27]. "There are already many publications available on the use of hypercomplex numbers in image recognition including face recognition [17, 28] as well as audio signal recognition [14]. Wider and wider application of hypercomplex number systems can be found during the organization of data transmission channels on the base of MIMO technology (multiple input- multiple output) and also for the newest techniques of space time coding [4, 6]. Finally, the field of cryptography has seen the development of extensive research into the use of hypercomplex numbers. The use of hypercomplex systems provides, for instance, the realization of stronger cryptographic keys [13, 16], and also for a wide

* *E-mail: atariov@wi.zut.edu.pl*
† *E-mail: gtariova@wi.zut.edu.pl*

range of digital watermarking techniques [10]. Of interest here, is that the higher the dimensionality of the hypercomplex number, the greater the security of the transmitted or stored data.
The multiplication of hypercomplex numbers is the most time-consuming operations when performing calculations in hypercomplex algebra. It should be noted that floating-point multiplication is much more expensive than floating-point addition. Multiplication of two quaternions requires 16 real-valued multiplications, whereas multiplication of two octonions or two sedenions, requires 64 or 256 real multiplications, respectively. The number of real multiplications increases quadratically with the dimensionality of the hypercomplex number. It is therefore evident that finding algorithms, which reduce the number of real multiplications required for computing products of hypercomplex numbers is very important.
Efficient algorithms for the multiplication of quaternions and octonions exist [12, 23–25]. No such algorithms for the multiplication of sedenions have been proposed. The aim of the present paper is to suggest the efficient algorithm for purpose.

2. Synthesis of algorithm

A sedenion is defined as follows:

$$s = a_0 + \sum_{i=1}^{15} a_i e_i,$$

where $\{a_i\}$, $i = 0, 1, \ldots, 15$ are real numbers and $\{e_i\}$, $i = 0, 1, \ldots, 15$ are imaginary units.
Table 1 shows the multiplication table of imaginary units sedenions [8].

Table 1. Multiplication table for sedenions imaginary units

$\times$	1	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}	e_{11}	e_{12}	e_{13}	e_{14}	e_{15}
1	1	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}	e_{11}	e_{12}	e_{13}	e_{14}	e_{15}
e_1	e_1	-1	e_3	$-e_2$	e_5	$-e_4$	$-e_7$	e_6	e_9	$-e_8$	$-e_{11}$	e_{10}	$-e_{13}$	e_{12}	e_{15}	$-e_{14}$
e_2	e_2	$-e_3$	-1	e_1	e_6	e_7	$-e_4$	$-e_5$	e_{10}	e_{11}	$-e_8$	$-e_9$	$-e_{14}$	$-e_{15}$	e_{12}	e_{13}
e_3	e_3	e_2	$-e_1$	-1	e_7	$-e_6$	e_5	$-e_4$	e_{11}	$-e_{10}$	e_9	$-e_8$	$-e_{15}$	e_{14}	$-e_{13}$	e_{12}
e_4	e_4	$-e_5$	$-e_6$	$-e_7$	-1	e_1	e_2	e_3	e_{12}	e_{13}	e_{14}	e_{15}	$-e_8$	$-e_9$	$-e_{10}$	$-e_{11}$
e_5	e_5	e_4	$-e_7$	e_6	$-e_1$	-1	$-e_3$	e_2	e_{13}	$-e_{12}$	e_{15}	$-e_{14}$	e_9	$-e_8$	e_{11}	$-e_{10}$
e_6	e_6	e_7	e_4	$-e_5$	$-e_2$	e_3	-1	$-e_1$	e_{14}	$-e_{15}$	$-e_{12}$	e_{13}	e_{10}	$-e_{11}$	$-e_8$	e_9
e_7	e_7	$-e_6$	e_5	e_4	$-e_3$	$-e_2$	e_1	-1	e_{15}	e_{14}	$-e_{13}$	$-e_{12}$	e_{11}	e_{10}	$-e_9$	$-e_8$
e_8	e_8	$-e_9$	$-e_{10}$	$-e_{11}$	$-e_{12}$	$-e_{13}$	$-e_{14}$	$-e_{15}$	-1	e_1	e_2	e_3	e_4	e_5	e_6	e_7
e_9	e_9	e_8	$-e_{11}$	e_{10}	$-e_{13}$	e_{12}	e_{15}	$-e_{14}$	$-e_1$	-1	$-e_3$	e_2	$-e_5$	e_4	e_7	$-e_6$
e_{10}	e_{10}	e_{11}	e_8	$-e_9$	$-e_{14}$	$-e_{15}$	e_{12}	e_{13}	$-e_2$	e_3	-1	$-e_1$	$-e_6$	$-e_7$	e_4	e_5
e_{11}	e_{11}	$-e_{10}$	e_9	e_8	$-e_{15}$	e_{14}	$-e_{13}$	e_{12}	$-e_3$	$-e_2$	e_1	-1	$-e_7$	e_6	$-e_5$	e_4
e_{12}	e_{12}	e_{13}	e_{14}	e_{15}	e_8	$-e_9$	$-e_{10}$	$-e_{11}$	$-e_4$	e_5	e_6	e_7	-1	e_1	$-e_2$	$-e_3$
e_{13}	e_{13}	$-e_{12}$	e_{15}	$-e_{14}$	e_9	e_8	e_{11}	$-e_{10}$	$-e_5$	$-e_4$	e_7	$-e_6$	e_1	-1	e_3	$-e_2$
e_{14}	e_{14}	$-e_{15}$	$-e_{12}$	e_{13}	e_{10}	$-e_{11}$	e_8	e_9	$-e_6$	$-e_7$	$-e_4$	e_5	e_2	$-e_3$	-1	e_1
e_{15}	e_{15}	e_{14}	$-e_{13}$	$-e_{12}$	e_{11}	e_{10}	$-e_9$	e_8	$-e_7$	e_6	$-e_5$	$-e_4$	e_3	e_2	$-e_1$	-1

The following notations are used:

$$s_1 = a_0 + \sum_{i=1}^{15} a_i e_i,\ s_2 = b_0 + \sum_{i=1}^{15} b_i e_i,$$

$$s = s_1 \cdot s_2 = c_0 + \sum_{i=1}^{15} c_i e_i.$$

In vector-matrix form we can write [26]

$$\mathbf{Y}_{16\times 1} = \mathbf{B}_{16}\mathbf{X}_{16\times 1}, \tag{1}$$

where

$$\mathbf{X}_{16\times 1} = [a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, a_{14}, a_{15}]^{\mathrm{T}},$$

$$\mathbf{Y}_{16\times 1} = [c_0, c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}, c_{15}]^{\mathrm{T}},$$

$$\mathbf{B}_{16} = \begin{bmatrix} \mathbf{B}_8^{(0,0)} & \mathbf{B}_8^{(0,1)} \\ \mathbf{B}_8^{(1,0)} & \mathbf{B}_8^{(1,1)} \end{bmatrix} \tag{2}$$

and

$$\mathbf{B}_8^{(0,0)} = \begin{bmatrix} b_0 & -b_1 & -b_2 & -b_3 & -b_4 & -b_5 & -b_6 & -b_7 \\ b_1 & b_0 & b_3 & -b_2 & b_5 & -b_4 & -b_7 & b_6 \\ b_2 & -b_3 & b_0 & b_1 & b_6 & b_7 & -b_4 & -b_5 \\ b_3 & b_2 & -b_1 & b_0 & b_7 & -b_6 & b_5 & -b_4 \\ b_4 & -b_5 & -b_6 & -b_7 & b_0 & b_1 & b_2 & b_3 \\ b_5 & b_4 & -b_7 & b_6 & -b_1 & b_0 & -b_3 & b_2 \\ b_6 & b_7 & b_4 & -b_5 & -b_2 & b_3 & b_0 & -b_1 \\ b_7 & -b_6 & b_5 & b_4 & -b_3 & -b_2 & b_1 & b_0 \end{bmatrix},$$

$$\mathbf{B}_8^{(0,1)} = \begin{bmatrix} -b_8 & -b_9 & -b_{10} & -b_{11} & -b_{12} & -b_{13} & -b_{14} & -b_{15} \\ b_9 & -b_8 & -b_{11} & b_{10} & -b_{13} & b_{12} & b_{15} & -b_{14} \\ b_{10} & b_{11} & -b_8 & -b_9 & -b_{14} & -b_{15} & b_{12} & b_{13} \\ b_{11} & -b_{10} & b_9 & -b_8 & -b_{15} & b_{14} & -b_{13} & b_{12} \\ b_{12} & b_{13} & b_{14} & b_{15} & -b_8 & -b_9 & -b_{10} & -b_{11} \\ b_{13} & -b_{12} & b_{15} & -b_{14} & b_9 & -b_8 & b_{11} & -b_{10} \\ b_{14} & -b_{15} & -b_{12} & b_{13} & b_{10} & -b_{11} & -b_8 & b_9 \\ b_{15} & b_{14} & -b_{13} & -b_{12} & b_{11} & b_{10} & -b_9 & -b_8 \end{bmatrix},$$

$$\mathbf{B}_8^{(1,0)} = \begin{bmatrix} b_8 & -b_9 & -b_{10} & -b_{11} & -b_{12} & -b_{13} & -b_{14} & -b_{15} \\ b_9 & b_8 & -b_{11} & b_{10} & -b_{13} & b_{12} & b_{15} & -b_{14} \\ b_{10} & b_{11} & b_8 & -b_9 & -b_{14} & -b_{15} & b_{12} & b_{13} \\ b_{11} & -b_{10} & b_9 & b_8 & -b_{15} & b_{14} & -b_{13} & b_{12} \\ b_{12} & b_{13} & b_{14} & b_{15} & b_8 & -b_9 & -b_{10} & -b_{11} \\ b_{13} & -b_{12} & b_{15} & -b_{14} & b_9 & b_8 & b_{11} & -b_{10} \\ b_{14} & -b_{15} & -b_{12} & b_{13} & b_{10} & -b_{11} & b_8 & b_9 \\ b_{15} & b_{14} & -b_{13} & -b_{12} & b_{11} & b_{10} & -b_9 & b_8 \end{bmatrix},$$

$$\mathbf{B}_8^{(1,1)} = \begin{bmatrix} b_0 & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 \\ -b_1 & b_0 & -b_3 & b_2 & -b_5 & b_4 & b_7 & -b_6 \\ -b_2 & b_3 & b_0 & -b_1 & -b_6 & -b_7 & b_4 & b_5 \\ -b_3 & -b_2 & b_1 & b_0 & -b_7 & b_6 & -b_5 & b_4 \\ -b_4 & b_5 & b_6 & b_7 & b_0 & -b_1 & -b_2 & -b_3 \\ -b_5 & -b_4 & b_7 & -b_6 & b_1 & b_0 & b_3 & -b_2 \\ -b_6 & -b_7 & -b_4 & b_5 & b_2 & -b_3 & b_0 & b_1 \\ -b_7 & b_6 & -b_5 & -b_4 & b_3 & b_2 & -b_1 & b_0 \end{bmatrix}.$$

The straightforward multiplication of two sedenions requires 256 real multiplications and 240 additions. We shall present an algorithm, which reduces the number of multiplications to 120 at the cost of 104 more additions.
Let $\mathbf{Y}_{16\times 1} = [\mathbf{Y}_{8\times 1}^{(0)}, \mathbf{Y}_{8\times 1}^{(1)}]^{\mathrm{T}}$, $\mathbf{Y}_{8\times 1}^{(0)} = [c_0, c_1, c_2, c_3, c_4, c_5, c_6, c_7]^{\mathrm{T}}$, $\mathbf{Y}_{8\times 1}^{(1)} = [c_8, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}, c_{15}]^{\mathrm{T}}$
and $\mathbf{X}_{16\times 1} = [\mathbf{X}_{8\times 1}^{(0)}, \mathbf{X}_{8\times 1}^{(1)}]^{\mathrm{T}}$, $\mathbf{X}_{8\times 1}^{(0)} = [a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7]^{\mathrm{T}}$, $\mathbf{X}_{8\times 1}^{(1)} = [a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, a_{14}, a_{15}]^{\mathrm{T}}$.
The first step in the synthesis procedure of the proposed algorithm is to partition the original matrix $\mathbf{B}_{16}$ by four square submatrices $\mathbf{B}_8^{(0,0)}$, $\mathbf{B}_8^{(0,1)}$, $\mathbf{B}_8^{(1,0)}$, $\mathbf{B}_8^{(1,1)}$, as can be seen from equation (2), to calculate the corresponding vector-matrix products for these matrices and then to sum the respective partial products in order to obtain the final result.

Then we can write

$$\mathbf{Y}_{8\times1}^{(0)} = \mathbf{B}_8^{(0,0)}\mathbf{X}_{8\times1}^{(0)} + \mathbf{B}_8^{(0,1)}\mathbf{X}_{8\times1}^{(1)}, \tag{3}$$

$$\mathbf{Y}_{8\times1}^{(1)} = \mathbf{B}_8^{(1,0)}\mathbf{X}_{8\times1}^{(0)} + \mathbf{B}_8^{(1,1)}\mathbf{X}_{8\times1}^{(1)}. \tag{4}$$

The next step is to perform some artificial transformations, which, as we will see later, will reduce the computational complexity of the final algorithm. Multiply the first rows of matrices $\mathbf{B}_8^{(0,0)}$ and $\mathbf{B}_8^{(1,0)}$, all the columns except the first one of matrices $\mathbf{B}_8^{(0,1)}$ and $\mathbf{B}_8^{(1,1)}$ as well as all the rows of matrix $\mathbf{B}_8^{(1,1)}$ by -1. Then the calculations corresponding to (3) and (4) can be replaced by the following:

$$\mathbf{Y}_{8\times1}^{(0)} = \tilde{\mathbf{I}}_8^{(1)}\tilde{\mathbf{B}}_8^{(0,0)}\mathbf{X}_{8\times1}^{(0)} + \tilde{\mathbf{B}}_8^{(0,1)}\tilde{\mathbf{I}}_8^{(2)}\mathbf{X}_{8\times1}^{(1)}, \tag{5}$$

$$\mathbf{Y}_{8\times1}^{(1)} = \tilde{\mathbf{I}}_8^{(1)}\tilde{\mathbf{B}}_8^{(1,0)}\mathbf{X}_{8\times1}^{(0)} + \tilde{\mathbf{I}}_8^{(3)}\tilde{\mathbf{B}}_8^{(1,1)}\tilde{\mathbf{I}}_8^{(2)}\mathbf{X}_{8\times1}^{(1)}, \tag{6}$$

where

$$\tilde{\mathbf{I}}_8^{(1)} = diag\,(-1,1,1,1,1,1,1,1), \quad \tilde{\mathbf{I}}_8^{(2)} = -\tilde{\mathbf{I}}_8^{(1)}, \quad \tilde{\mathbf{I}}_8^{(3)} = -\mathbf{I}_8$$

and $\mathbf{I}_8$ – is an identity matrix.

$$\mathbf{B}_8^{(0,0)} = \begin{bmatrix} -b_0 & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 \\ b_1 & b_0 & b_3 & -b_2 & b_5 & -b_4 & -b_7 & b_6 \\ b_2 & -b_3 & b_0 & b_1 & b_6 & b_7 & -b_4 & -b_5 \\ b_3 & b_2 & -b_1 & b_0 & b_7 & -b_6 & b_5 & -b_4 \\ b_4 & -b_5 & -b_6 & -b_7 & b_0 & b_1 & b_2 & b_3 \\ b_5 & b_4 & -b_7 & b_6 & -b_1 & b_0 & -b_3 & b_2 \\ b_6 & b_7 & b_4 & -b_5 & -b_2 & b_3 & b_0 & -b_1 \\ b_7 & -b_6 & b_5 & b_4 & -b_3 & -b_2 & b_1 & b_0 \end{bmatrix},$$

$$\mathbf{B}_8^{(0,1)} = \begin{bmatrix} -b_8 & b_9 & b_{10} & b_{11} & b_{12} & b_{13} & b_{14} & b_{15} \\ b_9 & b_8 & b_{11} & -b_{10} & b_{13} & -b_{12} & -b_{15} & b_{14} \\ b_{10} & -b_{11} & b_8 & b_9 & b_{14} & b_{15} & -b_{12} & -b_{13} \\ b_{11} & b_{10} & -b_9 & b_8 & b_{15} & -b_{14} & b_{13} & -b_{12} \\ b_{12} & -b_{13} & -b_{14} & -b_{15} & b_8 & b_9 & b_{10} & b_{11} \\ b_{13} & b_{12} & -b_{15} & b_{14} & -b_9 & b_8 & -b_{11} & b_{10} \\ b_{14} & b_{15} & b_{12} & -b_{13} & -b_{10} & b_{11} & b_8 & -b_9 \\ b_{15} & -b_{14} & b_{13} & b_{12} & -b_{11} & -b_{10} & b_9 & b_8 \end{bmatrix},$$

$$\mathbf{B}_8^{(1,0)} = \begin{bmatrix} -b_8 & b_9 & b_{10} & b_{11} & b_{12} & b_{13} & b_{14} & b_{15} \\ b_9 & b_8 & -b_{11} & b_{10} & -b_{13} & b_{12} & b_{15} & -b_{14} \\ b_{10} & b_{11} & b_8 & -b_9 & -b_{14} & -b_{15} & b_{12} & b_{13} \\ b_{11} & -b_{10} & b_9 & b_8 & -b_{15} & b_{14} & -b_{13} & b_{12} \\ b_{12} & b_{13} & b_{14} & b_{15} & b_8 & -b_9 & -b_{10} & -b_{11} \\ b_{13} & -b_{12} & b_{15} & -b_{14} & b_9 & b_8 & b_{11} & -b_{10} \\ b_{14} & -b_{15} & -b_{12} & b_{13} & b_{10} & -b_{11} & b_8 & b_9 \\ b_{15} & b_{14} & -b_{13} & -b_{12} & b_{11} & b_{10} & -b_9 & b_8 \end{bmatrix},$$

$$\mathbf{B}_8^{(1,1)} = \begin{bmatrix} -b_0 & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 \\ b_1 & b_0 & -b_3 & b_2 & -b_5 & b_4 & b_7 & -b_6 \\ b_2 & b_3 & b_0 & -b_1 & -b_6 & -b_7 & b_4 & b_5 \\ b_3 & -b_2 & b_1 & b_0 & -b_7 & b_6 & -b_5 & b_4 \\ b_4 & b_5 & b_6 & b_7 & b_0 & -b_1 & -b_2 & -b_3 \\ b_5 & -b_4 & b_7 & -b_6 & b_1 & b_0 & b_3 & -b_2 \\ b_6 & -b_7 & -b_4 & b_5 & b_2 & -b_3 & b_0 & b_1 \\ b_7 & b_6 & -b_5 & -b_4 & b_3 & b_2 & -b_1 & b_0 \end{bmatrix}$$

and

$$\check{\mathbf{B}}_8^{(0,0)} = \begin{bmatrix} b_0 & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 \\ b_1 & b_0 & b_3 & b_2 & b_5 & b_4 & b_7 & b_6 \\ b_2 & b_3 & b_0 & b_1 & b_6 & b_7 & b_4 & b_5 \\ b_3 & b_2 & b_1 & b_0 & b_7 & b_6 & b_5 & b_4 \\ b_4 & b_5 & b_6 & b_7 & b_0 & b_1 & b_2 & b_3 \\ b_5 & b_4 & b_7 & b_6 & b_1 & b_0 & b_3 & b_2 \\ b_6 & b_7 & b_4 & b_5 & b_2 & b_3 & b_0 & b_1 \\ b_7 & b_6 & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 \end{bmatrix}, \quad \check{\mathbf{B}}_8^{(0,1)} = \begin{bmatrix} b_8 & b_9 & b_{10} & b_{11} & b_{12} & b_{13} & b_{14} & b_{15} \\ b_9 & b_8 & b_{11} & b_{10} & b_{13} & b_{12} & b_{15} & b_{14} \\ b_{10} & b_{11} & b_8 & b_9 & b_{14} & b_{15} & b_{12} & b_{13} \\ b_{11} & b_{10} & b_9 & b_8 & b_{15} & b_{14} & b_{13} & b_{12} \\ b_{12} & b_{13} & b_{14} & b_{15} & b_8 & b_9 & b_{10} & b_{11} \\ b_{13} & b_{12} & b_{15} & b_{14} & b_9 & b_8 & b_{11} & b_{10} \\ b_{14} & b_{15} & b_{12} & b_{13} & b_{10} & b_{11} & b_8 & b_9 \\ b_{15} & b_{14} & b_{13} & b_{12} & b_{11} & b_{10} & b_9 & b_8 \end{bmatrix},$$

$$\hat{\mathbf{B}}_8^{(0,0)} = \begin{bmatrix} b_0 & & & & & & & \\ & & & b_2 & & b_4 & b_7 & \\ & b_3 & & & & & b_4 & b_5 \\ & & b_1 & & & b_6 & & b_4 \\ & b_5 & b_6 & b_7 & & & & \\ & & b_7 & & b_1 & & b_3 & \\ & & & b_5 & b_2 & & & b_1 \\ & b_6 & & & b_3 & b_2 & & \end{bmatrix}, \quad \hat{\mathbf{B}}_8^{(0,1)} = \begin{bmatrix} b_8 & & & & & & & \\ & & & b_{10} & & b_{12} & b_{15} & \\ & b_{11} & & & & & b_{12} & b_{13} \\ & & b_9 & & & b_{14} & & b_{12} \\ & b_{13} & b_{14} & b_{15} & & & & \\ & & b_{15} & & b_9 & & b_{11} & \\ & & & b_{13} & b_{10} & & & b_9 \\ & b_{14} & & & b_{11} & b_{10} & & \end{bmatrix},$$

$$\hat{\mathbf{B}}_8^{(1,0)} = \begin{bmatrix} b_8 & & & & & & & \\ & & b_{11} & & b_{13} & & & b_{14} \\ & & & b_9 & b_{14} & b_{15} & & \\ & b_{10} & & & b_{15} & & b_{13} & \\ & & & & & b_9 & b_{10} & b_{11} \\ & b_{12} & & b_{14} & & & & b_{10} \\ & b_{15} & b_{12} & & & b_{11} & & \\ & & b_{13} & b_{12} & & & b_9 & \end{bmatrix}, \quad \hat{\mathbf{B}}_8^{(1,1)} = \begin{bmatrix} b_0 & & & & & & & \\ & & b_3 & & b_5 & & & b_6 \\ & & & b_1 & b_6 & b_7 & & \\ & b_2 & & & b_7 & & b_5 & \\ & & & & & b_1 & b_2 & b_3 \\ & b_4 & & b_6 & & & & b_2 \\ & b_7 & b_4 & & & b_3 & & \\ & & b_5 & b_4 & & & b_1 & \end{bmatrix}.$$

These transformations are performed in order to be able to represent each of the matrices $\mathbf{B}_8^{(0,0)}$, $\mathbf{B}_8^{(0,1)}$, $\mathbf{B}_8^{(1,0)}$ and $\mathbf{B}_8^{(1,1)}$ as the algebraic sum of the block-Toeplitz symmetric matrix and a certain type of sparse matrix, i.e. matrix containing only small number of non-zero elements. So we can write:

$$\begin{aligned} \bar{\mathbf{B}}_8^{(0,0)} &= \check{\mathbf{B}}_8^{(0,0)} - 2\hat{\mathbf{B}}_8^{(0,0)}, \quad \bar{\mathbf{B}}_8^{(0,1)} = \check{\mathbf{B}}_8^{(0,1)} - 2\hat{\mathbf{B}}_8^{(0,1)}, \\ \bar{\mathbf{B}}_8^{(1,0)} &= \check{\mathbf{B}}_8^{(0,1)} - 2\hat{\mathbf{B}}_8^{(1,0)}, \quad \bar{\mathbf{B}}_8^{(1,1)} = \check{\mathbf{B}}_8^{(0,0)} - 2\hat{\mathbf{B}}_8^{(1,1)} \end{aligned} \tag{7}$$

Substituting (7) into (5) and (6) we get:

$$\mathbf{Y}_{8\times1}^{(0)} = \tilde{\mathbf{I}}_8^{(1)} \left(\check{\mathbf{B}}_8^{(0,0)} - 2\hat{\mathbf{B}}_8^{(0,0)} \right) \mathbf{X}_{8\times1}^{(0)} + \left(\check{\mathbf{B}}_8^{(0,1)} - 2\hat{\mathbf{B}}_8^{(0,1)} \right) \tilde{\mathbf{I}}_8^{(2)} \mathbf{X}_{8\times1}^{(1)}$$

$$\mathbf{Y}_{8\times1}^{(1)} = \tilde{\mathbf{I}}_8^{(1)} \left(\check{\mathbf{B}}_8^{(0,1)} - 2\hat{\mathbf{B}}_8^{(1,0)} \right) \mathbf{X}_{8\times1}^{(0)} + \tilde{\mathbf{I}}_8^{(3)} \left(\check{\mathbf{B}}_8^{(0,0)} - 2\hat{\mathbf{B}}_8^{(1,1)} \right) \tilde{\mathbf{I}}_8^{(2)} \mathbf{X}_{8\times1}^{(1)}.$$

In this case the vector-matrix products $\check{\mathbf{B}}_8^{(0,0)}\mathbf{X}_{8\times1}^{(0)}$, $\check{\mathbf{B}}_8^{(0,0)}\mathbf{X}_{8\times1}^{(1)}$, $\check{\mathbf{B}}_8^{(0,1)}\mathbf{X}_{8\times1}^{(0)}$, and $\check{\mathbf{B}}_8^{(0,1)}\mathbf{X}_{8\times1}^{(1)}$ (with Toeplitz-type matrix) can now be calculated using one of the well-known fast algorithms, and the products $2\hat{\mathbf{B}}_8^{(0,0)}\mathbf{X}_{8\times1}^{(0)}$, $2\hat{\mathbf{B}}_8^{(0,1)}\mathbf{X}_{8\times1}^{(1)}$, $2\hat{\mathbf{B}}_8^{(1,0)}\mathbf{X}_{8\times1}^{(0)}$ and $2\hat{\mathbf{B}}_8^{(1,1)}\mathbf{X}_{8\times1}^{(1)}$ are calculated directly, without any tricks. For reference, note that the first term in (5), i.e. the product of $\tilde{\mathbf{I}}_8^{(1)}\bar{\mathbf{B}}_8^{(0,0)}\mathbf{X}_{8\times1}^{(0)}$ is the multiplication of octonions. Synthesis of the fast algorithm for the calculation of this product with decomposition $\bar{\mathbf{B}}_8^{(0,0)} = \check{\mathbf{B}}_8^{(0,0)} - 2\hat{\mathbf{B}}_8^{(0,0)}$ was described in detail in our previous article [24]. By analogy, we can construct algorithms for the remaining three vector-matrix products from expressions (5) and (6). It should, however be noted that the fast algorithms for computing partial products $\tilde{\mathbf{I}}_8^{(1)}\bar{\mathbf{B}}_8^{(0,0)}\mathbf{X}_{8\times1}^{(0)}$, $\bar{\mathbf{B}}_8^{(0,1)}\tilde{\mathbf{I}}_8^{(2)}\mathbf{X}_{8\times1}^{(1)}$, $\tilde{\mathbf{I}}_8^{(1)}\bar{\mathbf{B}}_8^{(1,0)}\mathbf{X}_{8\times1}^{(0)}$ and $\tilde{\mathbf{I}}_8^{(3)}\bar{\mathbf{B}}_8^{(1,1)}\tilde{\mathbf{I}}_8^{(2)}\mathbf{X}_{8\times1}^{(1)}$ have the same fragments of computing. Therefore, using these algorithms directly, followed by summation of the results, is inefficient and results in unnecessary additions. If you try to take into account and to prevent a repetition of the

same calculations, the additive complexity can be considerably reduced. In this case the resulting algorithm takes the following form:

$$\mathbf{Y}_{16\times1} = \mathbf{A}_{16\times32}\mathbf{A}_{32\times64}\mathbf{W}_{64}\mathbf{D}_{64}\mathbf{P}_{64\times48}\mathbf{W}_{48}\mathbf{P}_{48\times16}\mathbf{X}_{16\times1}, \tag{8}$$

where

$$\mathbf{P}_{48\times16} = (\mathbf{1}_{2\times1}\otimes\mathbf{I}_8)\oplus\left(\mathbf{1}_{2\times1}\otimes\tilde{\mathbf{I}}_8^{(2)}\right) = \begin{bmatrix} \begin{bmatrix}\mathbf{I}_8\\ \mathbf{I}_8\end{bmatrix} & \mathbf{0}_{16\times8} \\ \mathbf{0}_{16\times8} & \begin{bmatrix}\tilde{\mathbf{I}}_8^{(2)}\\ \tilde{\mathbf{I}}_8^{(2)}\end{bmatrix}\end{bmatrix},$$

$$\mathbf{W}_{48} = \mathbf{H}_8\oplus\mathbf{I}_{32}\oplus\mathbf{H}_8 = \begin{bmatrix}\mathbf{H}_8 & \mathbf{0}_{8\times16} & \mathbf{0}_8\\ \mathbf{0}_{16\times8} & \mathbf{I}_{16} & \mathbf{0}_{16\times8}\\ \mathbf{0}_8 & \mathbf{0}_{8\times16} & \mathbf{H}_8\end{bmatrix}, \quad \mathbf{H}_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1\\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1\\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1\\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1\\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1\end{bmatrix},$$

$$\mathbf{P}_{64\times48} = (\mathbf{1}_{2\times1}\otimes\mathbf{I}_8)\oplus\mathbf{I}_{32}\oplus(\mathbf{1}_{2\times1}\otimes\mathbf{I}_8) = \begin{bmatrix}\begin{bmatrix}\mathbf{I}_8\\ \mathbf{I}_8\end{bmatrix} & \mathbf{0}_{16\times32} & \mathbf{0}_{16\times8}\\ \mathbf{0}_{32\times8} & \mathbf{I}_{32} & \mathbf{0}_{32\times8}\\ \mathbf{0}_{16\times8} & \mathbf{0}_{16\times32} & \begin{bmatrix}\mathbf{I}_8\\ \mathbf{I}_8\end{bmatrix}\end{bmatrix},$$

$$\mathbf{D}_{64} = \mathbf{D}_{16}\oplus\hat{\mathbf{B}}_8^{(0,0)}\oplus\hat{\mathbf{B}}_8^{(1,0)}\oplus\hat{\mathbf{B}}_8^{(0,1)}\oplus\hat{\mathbf{B}}_8^{(1,1)}\oplus\mathbf{D}_{16}, \quad \mathbf{W}_{64} = (\mathbf{I}_2\otimes\mathbf{H}_8)\oplus\mathbf{D}_{32}\oplus(\mathbf{I}_2\otimes\mathbf{H}_8), \ \mathbf{D}_{32} = 2\mathbf{I}_{32},$$

$$\mathbf{D}_{16} = diag(s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}),$$

$$\mathbf{A}_{32\times64} = \left(\hat{\mathbf{1}}_{1\times2}\otimes\mathbf{I}_{16}\right)\oplus\left(\check{\mathbf{1}}_{1\times2}\otimes\mathbf{I}_{16}\right) = \begin{bmatrix}\begin{bmatrix}\mathbf{I}_{16} & -\mathbf{I}_{16}\end{bmatrix} & \mathbf{0}_{16\times32}\\ \mathbf{0}_{16\times32} & \begin{bmatrix}-\mathbf{I}_{16} & \mathbf{I}_{16}\end{bmatrix}\end{bmatrix},$$

$$\mathbf{A}_{16\times32} = \begin{bmatrix}\mathbf{I}_2\otimes\tilde{\mathbf{I}}_8^{(1)} & \mathbf{I}_8\oplus(-\mathbf{I}_8)\end{bmatrix} = \begin{bmatrix}\tilde{\mathbf{I}}_8^{(1)} & \mathbf{0}_8 & \mathbf{I}_8 & \mathbf{0}_8\\ \mathbf{0}_8 & \tilde{\mathbf{I}}_8^{(1)} & \mathbf{0}_8 & -\mathbf{I}_8\end{bmatrix},$$

$\mathbf{1}_{2\times1} = [1 \quad 1]^{\mathrm{T}}$, $\check{\mathbf{1}}_{1\times2} = [-1 \quad 1]$, $\hat{\mathbf{1}}_{1\times2} = [1 \quad -1]$, $\mathbf{0}_N$ – is order N zero matrix and "$\otimes$", "$\oplus$" – denotes tensor product and direct sum of two matrices respectively [22].

The data flow diagram for the computation of the product of two sedenions is shown in Figure 1. 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 algebraic summations. (The dash-dotted lines indicate the subtraction operation). We use the usual lines without arrows on purpose, so as not to clutter the picture. The rectangles denote the corresponding vector-matrix products, the meaning of which is clear from the context.

Note, that the multiplication of the corresponding vectors by order 8 Hadamard matrix (see Figure 1) is implemented by a well-known Fast Hadamard transform algorithm (see e.g., [23]) which, along with the other methods used in this article, allows to reduce the computational complexity.

Diagonal elements $\{s_i\}$ of the matrix $\mathbf{D}_{16}$ are calculated using the procedure (9).

$$\mathbf{S}_{16\times1} = \frac{1}{8}\mathbf{W}_{16}^2\mathbf{W}_{16}^1\mathbf{W}_{16}^0\mathbf{B}_{16\times1}, \tag{9}$$

where

$$\mathbf{S}_{16\times 1} = [s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}]^{\mathrm{T}},$$

$$\mathbf{B}_{16\times 1} = [b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}, b_{12}, b_{13}, b_{14}, b_{15}]^{\mathrm{T}},$$

$$\mathbf{W}_{16}^{0} = \mathbf{I}_2 \otimes \mathbf{H}_2 \otimes \mathbf{I}_4, \quad \mathbf{W}_{16}^{1} = \mathbf{I}_4 \otimes \mathbf{H}_2 \otimes \mathbf{I}_2, \quad \mathbf{W}_{16}^{2} = \mathbf{I}_8 \otimes \mathbf{H}_2$$

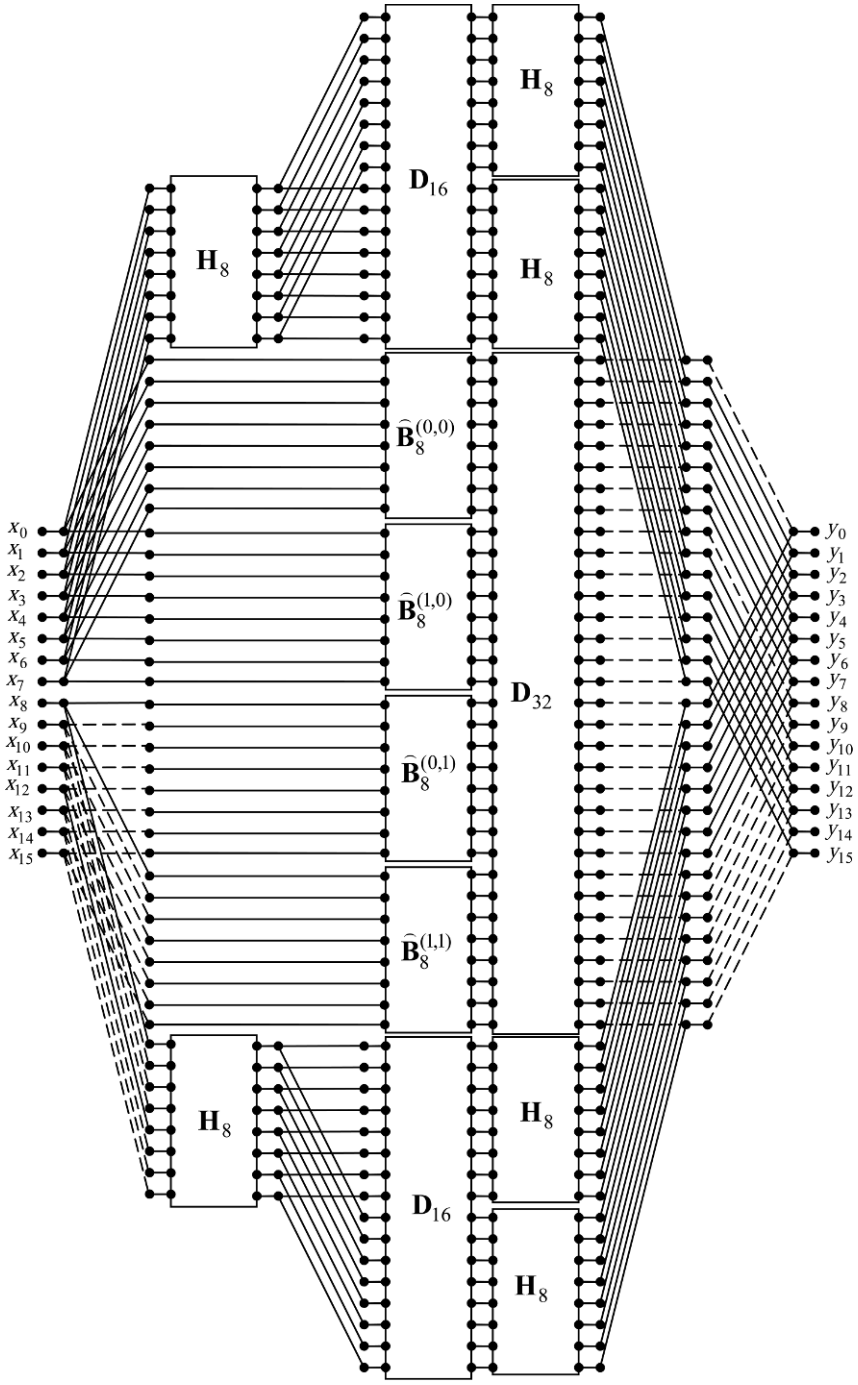

Figure 1. Data flow diagram for rationalized sedenion multiplication algorithm.

and $\mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ – is an order 2 Hadamard matrix.

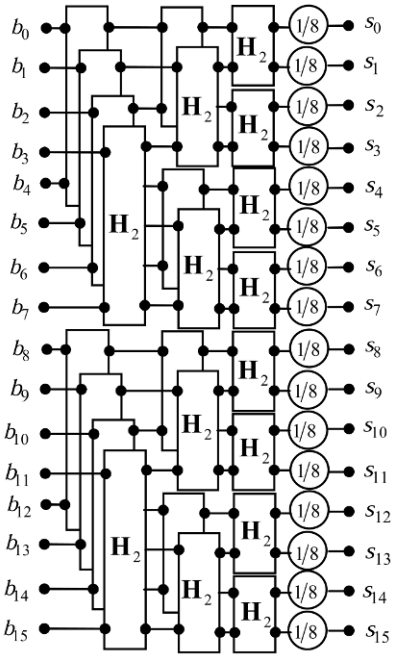

Figure 2. The signal diagram describing the process of calculating elements of the vector $\mathbf{S}_{16\times 1}$ in accordance with the procedure (9).

Figure 2 shows a data flow diagram of the process for calculating the matrix $\mathbf{D}_{16}$ elements. The circles in this figure show the operation of multiplication by a number (constant) inscribed inside a circle. In turn, the rectangles indicate the vector-matrix multiplications with the (2×2)-Hadamard matrices.

3. Conclusion

In this paper, we have presented an original algorithm allowing to multiply two sedenions with reduced multiplicative complexity. Our method reduces the number of scalar multiplications required to compute the product of two sedenions from 256 to 120. The number of additions is increased by 104, however the total number of arithmetic operations is reduced by 32.

It should be noted that in many practical applications, one of the sedenions to be multiplied contains constants. In this case, the diagonal matrix elements can be precomputed. This would reduce the number of additions in the proposed algorithm to 248.

References

[1] Alfsmann D., On families of 2^N-dimensional hypercomplex algebras suitable for digital signal processing, In: Proceedings of European Signal Processing Conf. (EUSIPCO 2006), Florence, Italy, 2006

[2] Alfsmann D., Göckler H.G., Sangwine S.J., Ell T.A., Hypercomplex Algebras in Digital Signal Processing: Benefits and Drawbacks (Tutorial), In: Proceedings of EURASIP 15th European Signal Processing Conference (EUSIPCO 2007), Poznan, Poland, 1322-1326, 2007

[3] Bülow T., Sommer G., Hypercomplex signals - a novel extension of the analytic signal to the multidimensional case, IEEE Trans. Sign. Proc., SP-49, 11, 2844-2852, 2001

[4] Calderbank R., Das S., Al Dhahir N., Diggavi S., Construction And Analysis of A New Quaternionic Space-Time Code For 4 Transmit Antennas, Commun. Inf. Syst., 5, 97-122, 2005
[5] Chan W. L., Choi H., Baraniuk R. G., Directional hypercomplex wavelets for multidimensional signal analysis and processing, In: Proceedings of ICASSP '04: IEEE International Conference on Acoustics, Speech, and Signal Processing, 3, 996-999, 2004
[6] Ertug O., Communication over Hypercomplex Kahler Manifolds: Capacity of Dual-Polarized Multidimensional-MIMO Channels, Wirel. Pers. Commun., 41, 155-168, 2006
[7] Hahn S.L., Snopek K.M., The unified theory of *n*-dimensional complex and hypercomplex analytic signals, Bull. Pol. Acad. Sci. Tech. Sci., 59, 167-181, 2011
[8] Imaeda K., Imaeda M., Sedenions: algebra and analysis, Appl. Math. Comput., 115, 77-88, 2000
[9] Kantor I., Solodovnikov A., Hypercomplex numbers (Springer-Verlag, New York, 1989)
[10] Liu L., An Introduction to Hyper-Complex on Meaningful Digital Watermarking, In: Proceedings of 6th International Conference on Wireless Communications Networking and Mobile Computing 23-25 Sept. 2010, 1-5, 2010
[11] Lu Z., Xu Y., Yang X., Song L., Traversoni L., 2D quaternion Fourier transform: the spectral properties and its application in color image representation, In: Proceedings of IEEE International Conference on Multimedia and Expo., 1715-1718, 2007
[12] Makarov O.M., An algorithm for the multiplication of two quaternions, Zh. Vychisl. Mat. Mat. Fiz., 17, 1574-1575, 1977
[13] Malekian E., Zakerolhosseini A., Mashatan A., QTRU: Quaternionic Version of the NTRU Public-Key Cryptosystems, Int. J. Inf. Secur., 3, 29-42, 2011
[14] Miron S., Le Bihan N., Mars J., Quaternion-music for vector-sensor array processing, IEEE Trans. Signal Process., 54, 1218-1229, 2006
[15] Moxey C.E., Sangwine S. J., Ell T.A., Hypercomplex correlation techniques for vector images, IEEE Trans. Signal Process., 2003. 51, 1941-1953
[16] Nagase T., Komata M., Araki T., Secure Signals Transmission Based on Quaternion Encryption Scheme, In: Proceedings of IEEE Advanced Information Networking and Applications (AINA 2004), 2, 35-38, 2004
[17] Pei S., Cheng C., A novel block truncation coding of color images by using quaternion-moment preserving principle. In: Proceedings of IEEE International Symposium on Circuits and systems. Volume 2., Atlanta, USA, 684-687, 1996
[18] Sangwine S.J., Fourier transforms of color images using quaternion or hypercomplex, numbers, Electron. Lett, 32, 1979-1980, 1996
[19] Sangwine S.J., Bihan N. Le, Hypercomplex analytic signals: extension of the analytic signal concept to complex signals, In: Proceedings of EURASIP 15th European Signal Processing Conference (EUSIPCO 2007), Poznan, Poland, 621-624, 2007
[20] Sangwine S.J., Ell T.A., Hypercomplex auto- and cross-correlation of color images, In: Proceedings of ICIP, 319-323, 1999
[21] Schütte H.D., Wenzel J., Hypercomplex numbers in digital signal processing, In: Proceedings of ISCAS '90, New Orleans, 1557-1560, 1990
[22] Steeb W.H., Hardy Y., Matrix Calculus and Kronecker Product: A Practical Approach to Linear and Multilinear Algebra (World Scientific Publishing Company, 2011)
[23] Ţariov A., Algorithmic aspects of computing rationalization in digital signal processing (West Pomeranian University of Technology Press, 2011) (In Polish)
[24] Ţariov A., Ţariova G., Algorithmic aspects of Cayley numbers multiplier organization, Elektronika, 11, 104-108, 2010 (in Polish)
[25] Ţariova G., Ţariov A., Algoritmic aspects of multiplication blocks number reduction in a two quaternion hardware multiplier, Meas. Autom. Monit., 7, 668-690, 2010 (in Polish)
[26] Ţariova G., Ţariov A., Representation of sedenions multiplication via matrix-vector product, Metody Informatyki Stosowanej, 1, 133-139, 2011
[27] Ueda K., Takahashi. S.I., Digital filters with hypercomplex coefficients, In: Proceedings of IEEE Int. Symp. Circuits Syst., 1, 479-482, 1993
[28] Villegas M., Paredes R., Face Recognition in Color Using Complex and Hypercomplex Representations. Pattern Recognition and Image Analysis, Lect. Notes Comput. Sci., 4477, 217-224, 2007