BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES,
DOI: 10.24425/bpasts.2025.153827

The Fast Type-IV Discrete Sine Transform Algorithms for Short-Length Input Sequences

Marina POLYAKOVA[1*], **Anna WITENBERG**[2**], **Aleksandr CARIOW**[3***]

[1] Institute of Computer Systems, Odesa Polytechnic National University, Shevchenko 1, Odesa, 65044, Ukraine
[2] Institute of Telecommunications and Computer Science, University of Science and Technology;
Al. prof. S. Kaliskiego 7, 85-796 Bydgoszcz, Poland
[3] Faculty of Computer Science and Information Technology, West Pomeranian University of Technology in Szczecin,
Zołnierska 49, 71-210 Szczecin, Poland

Abstract. The fast algorithms of discrete sinusoidal transform of the fourth type (DST-IV) for small-length input data in the range of lengths from 2 to 9 are developed. Fast algorithms for short input data sequences are subsequently used as building blocks for designing fast algorithms of large-sized discrete transforms. Applying the fast DST-IV algorithms for small-size block processing can reduce overall system complexity and delay, allowing detailed signal processing. As a result of the literature review, two main approaches to developing fast discrete sine transform (DST) algorithms were identified, namely, the polynomial algebraic approach and the matrix factorization approach. In the paper, the last approach is exploited. A matrix-vector product expression of the DST-IV is the starting point for designing the fast algorithms. Then based on the repetition and arranging of the matrix elements, the factorization of the matrices of coefficients of DST-IV is produced to reduce computational complexity. The correctness of the obtained algorithmic solutions was justified theoretically using a strict mathematical background of each of them. The elaborated algorithms were then further tested using MATLAB R2023b software to finally confirm their performance. The resulting factorizations of the DST-IV matrices reduce the number of multiplications by 63% but increase the number of additions by 8% on average in the range of signal sample numbers from 3 to 9. It has been observed that for even-length input sequences, the reduction in the number of multiplications is not as significant as for odd-length sequences. For some other well-known discrete trigonometric transforms (discrete Fourier transform, discrete Hartley transform) the opposite situation holds. The proposed DST-IV fast algorithms do not limit the length of the input data sequence to powers of two or three. The data flow graphs constructed for the proposed algorithms reveal their modular space-time structure suitable for VLSI implementation.

Key words: discrete sine transform; matrix factorization; fast algorithms; computational complexity; digital signal processing

1. INTRODUCTION

Discrete orthogonal transforms are widely used in digital signal and image processing [1, 2]. One of the important, but not the main advantages of such transforms is a huge number of so-called fast algorithms that have been developed for them [3, 4]. Fast algorithms allow the implementation of discrete orthogonal transforms with high computational efficiency [5, 6]. Among other orthogonal transforms, the DST is a well-known and well-tested tool in digital signal processing, specifically, in video steganography [7], medical image fusion [8], filter design [9, 10], video coding [11, 12], image transmission [13], image denoising [14]. However undesirable high-frequency artifacts may often appear when using DST for image compression [1].

In the signal and image processing community, the reasonableness of applying the DST versus discrete cosine transform (DCT) and discrete Fourier transform (DFT) are discussed from time to time. If an input signal is mirrored by a symmetric reflection then the FFT input does not have a discontinuity in the middle. In this case, the DCT is roughly equivalent to the DFT. At the same time, the DST is roughly equivalent to a DFT after an antisymmetric mirrored extension which results in discontinuities both in the middle and around the circle. Discontinuities are represented by energy in the high-frequency bins in the FFT results. These high-frequency artifacts are usually undesirable when using a transform for compression. But at the same time, DST is applied in the better portable graphics coder to save the high-frequency content of video sequences.

Thus, the paper [15] is devoted to the intra-frame coding of video sequences by High Efficiency Video Coding (HEVC). The term intra-frame coding means that different compression techniques are applied relative to information from the current frame only, and not based on any other frame in the video sequence. HEVC is widely used in video coding as an advanced image compression standard more efficient than JPEG 2000. HEVC exploits a significant number of coding tools, in particular, DCT and DST are both applied in HEVC. Then evaluating the amount of transform operations performed by the encoder in a realistic application is extremely important to select optimal settings in rate-distortion sense.

The DST can outperform the discrete Fourier transform in speech and voice applications, for example, in speech signal enhancement [16–18]. In [16] to estimate the spectra of speech signals for deep-learning models, the DST, DCT, and other well-known orthogonal transforms were applied rather

*e-mail: polyakova@op.edu.ua
**e-mail: anna.witenberg@pbs.edu.pl
***e-mail: atariov@wi.zut.edu.pl

than the traditional DFT. The widely used deep-learning architectures were tested, specifically, convolutional neural network and fully connected neural network. To evaluate the speech enhancement performance, several speech quality and intelligibility measures were estimated on the signals from the NOIZEUS database. The obtained results demonstrated that DST is better suited than DFT for speech enhancement with considered deep-learning models at signal-noise-ratio of 5, 10, and 15 dB [16].

The DST is also applied for voice register recognition [19]. Human voices are divided based on sound timbre, produced high and low tones, resonant space sensation, sound source, etc. In [19] the types of female voice were recognized, specifically, the chest voice, head voice, falsetto, and vocal fry. The true detection rate in percentage was estimated for recognition based on the DST, DCT, and DFT. The DST-based true detection rate ranged from 63% to 79%, while the same measure for the DCT-based and DFT-based recognition was around 5-10% lower. Therefore the DST-based methods is advisable to use in applications where the high frequency content of signals or images is significant.

Until now eight types of DST are known [20]. Although DST is employed less frequently than DCT, the most exploited transform is DST-II [21,22]. The other types of DST are much less applied. However, the discrete orthogonal transforms allow the acceleration of their implementation by developing so-called fast algorithms [2]. The number of arithmetic operations required for the direct computation of the DST-IV is of order N^2, where N is the length of the input sequence. That is why fast algorithms for DST-IV have been developed to reduce the implementation cost and computational complexity [21–24].

It is necessary to note that the case of large lengths of input data sequences is mostly considered in the papers relating to designing fast DST-IV algorithms [4, 22–26]. However, the short-length DST-IV algorithms are of special interest, since they can be considered as typical modules in synthesizing more complex algorithms [2,23]. Once constructed, the short-length DST-IV algorithms can be successfully applied in various projects to unify the process of developing the final product. For example, in real-time applications such as video conferencing or voice control systems applying the fast DST-IV algorithms for small-size block processing can reduce overall system complexity and delay, allowing the detailed analysis of audio signals [27,28].

Then the designing the fast DST-IV algorithms for short-length input sequences is a relevant problem. Further to select the approach for developing such fast DST-IV algorithms the related papers is analyzed.

1.1. State-of-art of the Problem

Among the various mathematical methods used to obtain fast algorithms for DST, the dominant approaches are the polynomial arithmetic approach [22, 23] and matrix factorization [4, 20, 21, 24]. In addition, new fast DST algorithms for direct very large scale integration (VLSI) have been developed in [25,26].

Based on the first approach the large class of fast general radix algorithms was introduced in [22]. The proposed algorithms were developed by exploiting a polynomial algebra for DST and well-known Cooley-Tukey fast Fourier transform instead of the manipulating the entries of transform matrices. In [23] a new decomposition DST-IV algorithm is described. The result was higher computational efficiency and simpler hardware implementation compared to existing algorithms. The efficiency of the introduced algorithm is verified by a real-time audio decoding application. The second approach is based on a deep analysis of the structures of discrete orthogonal transform matrices. As a result, the individual features of the arrangement of identical entries are identified [29]. Then, if necessary, the matrix structures are altered for subsequent use of certain matrix structures that lead to the suitable factorization of these matrices. In the end, the fast, efficient, and recursive DST algorithms were developed by applying, for example, the factorization on sparse, scaled orthogonal, rotational, and rotational-reflection matrices [4,20].

So, in [4] the authors have obtained the recursive radix-2 DST algorithms with the lowest multiplication complexity. The proposed algorithms are executed via factorization of DST matrices into a product of diagonal, bidiagonal, sparse, and scaled orthogonal matrices. As a result, the lowest multiplication complexity is achieved among DST-IV algorithms described in the literature. The relationship between DST-II and DST-IV matrices was derived using diagonal and bidiagonal matrices. The algorithms based on the proposed DST-II and DST-III factorizations were implemented within an image encryption scheme with double random phase encoding.

In [20] the factorizations of DST I-IV matrices are proposed. The resulting matrices are scaled orthogonal, sparse, butterfly, rotational, and rotational-reflection. Then the connection between the obtained factorizations and signal flow graph building blocks was established. As a result, the signal flow graphs were constructed for the DST I-IV algorithms if the length of the input sequence is equal to 8 or 16. These DST algorithms significantly improve the speed of calculations and have low arithmetic complexity.

In [24] the fast DST-IV algorithms were presented. Thus a lower count of real multiplications and additions than previously published algorithms was achieved without the numerical accuracy sacrificing. To develop the proposed algorithms the DCT-IV was considered as a special case of a discrete Fourier transform of length $8N$ with certain symmetries. Then the recent split-radix fast Fourier transform algorithm was applied. The improved algorithms for DST-IV follow immediately from the obtained DCT-IV algorithms.

The analysis of known fast DST-IV algorithms made it possible to define their shortcomings and formulate unsolved parts of the general problem of reducing computational complexity.

1.2. The Main Contributions of the Paper

As a result of the analysis of papers devoted to synthesizing the fast DST-IV algorithms, the following should be noted [5]. The aforementioned algorithms are focused on the input sequences with length N being the power of two or three. The mathe-

matical notations used by the authors to describe the proposed fast algorithms are often quite complicated. As a result, the understanding of the essence of the suggested solutions may be difficult for practitioners when a wide range of problems are solved. The data flow graphs are not always constructed for proposed fast algorithms which makes their implementation difficult. To avoid these disadvantages, in [5] the structural approach is proposed to develop fast algorithms for the matrix-vector product via matrix factorization. It is based on a deep analysis of the structures of base transform matrices, identifying individual features of the arrangement of identical entries. Then, if necessary, the matrix structures are changed for subsequent use of certain matrix identities that lead to the suitable factorization of these matrices. This way, the number of multiplication and addition operations necessary for the calculation of matrix-vector product may be reduced. The structural approach does not limit the length of the original data sequence, for example, to a power of two or three [5, 21, 30, 31]. At least for odd N DST-IV, the coefficients matrices are structured, then developing the fast DST-IV algorithms based on a structural approach to matrix factorization is relevant.

In addition, the case of large lengths of input data sequences is considered in papers concerning the efficient implementation of DSTs [4, 20–26]. However, short-length DST-IV algorithms are of particular interest since these transforms are applied as typical modules in synthesizing more complex algorithms [2]. In addition, fast algorithms for short-length sequences can be successfully used directly, for example, in the processing of signals with low sampling rates [32, 33]. Therefore, this research aims to develop the reduced-complexity DST-IV algorithms based on a structural approach for input sequences of length $N = 2,3,4,5,6,7,8,9$. These values of N allow for further construction of the radix-type algorithms with a large range of radix values.

2. MATERIAL AND METHODS

2.1. Preliminary Remarks

DST-IV can be expressed as follows [1, 4]:

$$y_k = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x_n sin \frac{\pi(2n+1)(2k+1)}{4N}, \tag{1}$$

where: $k = 0,1,\ldots,N-1$, y_k is the output sequence after the DST-IV; x_n is the input sequence; N is the number of signal samples.

DST-IV can be represented in matrix notation by the expression:

$$\mathbf{Y}_{N\times 1} = \mathbf{C}_N \mathbf{X}_{N\times 1}, \tag{2}$$

where

$$\mathbf{X}_{N\times 1} = [x_0, x_1, \ldots, x_{N-1}]^T,$$
$$\mathbf{Y}_{N\times 1} = [y_0, y_1, \ldots, y_{N-1}]^T,$$
$$c_{kl} = \sqrt{\frac{2}{N}} sin \frac{\pi(2l+1)(2k+1)}{4N}, \quad k,l = 0,1,\ldots,N-1.$$

In this paper, we use the following notations and signs:

$\mathbf{I}_N$ is an order N identity matrix;
$\mathbf{H}_2$ is a 2×2 Hadamard matrix;
$\mathbf{1}_{N\times M}$ is a $N\times M$ matrix of ones;
$\otimes$ is the Kronecker product of two matrices;
$\oplus$ is the direct sum of two matrices.

An empty cell in a matrix means it contains zero.We designated the values of the sines obtained as a result of the transformations as $s_m^{(N)}$.

As a graphical illustration of the space-time structures of the developed algorithms, we will use data flow graphs oriented from right to left. We will use regular straight lines to denote data transfer operations and dashed lines to denote data transfer operations with simultaneous sign changes. Nodes from which the lines diverge mean data transfer, and nodes where the lines connect symbolize summations. Circles denote multiplications by sine values obtained as a result of arithmetic operations.

DST-IV in matrix-vector representation is expressed as follows:

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-1} \end{bmatrix} = \begin{bmatrix} c_{0,0} & c_{0,1} & \cdots & c_{0,N-1} \\ c_{1,0} & c_{1,1} & \cdots & c_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ c_{N-1,0} & c_{N-1,1} & \cdots & c_{N-1,N-1} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{bmatrix}. \tag{3}$$

2.2. Algorithm for 2-point DST-IV

To obtain the algorithm for two-point DST-IV the expression (2) is represented as follows:

$$\mathbf{Y}_{2\times 1} = \mathbf{C}_2 \mathbf{X}_{2\times 1} \tag{4}$$

where:

$$\mathbf{X}_{2\times 1} = [x_0, x_1]^T, \quad \mathbf{Y}_{2\times 1} = [y_0, y_1]^T, \quad \mathbf{C}_2 = \begin{bmatrix} a_2 & b_2 \\ b_2 & -a_2 \end{bmatrix},$$

$$a_2 = \sin\left(\frac{\pi}{8}\right) \approx 0.3827, \quad b_2 = \sin\left(\frac{3\pi}{8}\right) \approx 0.9239.$$

Based on the structural properties of matrix $\mathbf{C}_2$ [5, 21], the expression for DST-IV for $N = 2$ can be presented as follows:

$$\mathbf{Y}_{2\times 1} = \mathbf{W}_{2\times 3} \mathbf{D}_3 \mathbf{W}_{3\times 2} \mathbf{X}_{2\times 1}, \tag{5}$$

where:

$$\mathbf{D}_3 = \mathrm{diag}\left(s_0^{(2)}, s_1^{(2)}, s_2^{(2)}\right),$$

$$s_0^{(2)} = a_2 - b_2, \quad s_1^{(2)} = -(a_2 + b_2), \quad s_2^{(2)} = b_2,$$

$$\mathbf{W}_{3\times 2} = \begin{bmatrix} 1 & \\ & 1 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{W}_{2\times 3} = \begin{bmatrix} 1 & & 1 \\ & 1 & 1 \end{bmatrix}.$$

A data flow graph of the synthesized algorithm for the two-point DST-IV is shown in the Figure 1. If this algorithm is applied, then the number of multiplication operations may be reduced from 4 to 3, although the number of addition operations is increased from 2 to 3.

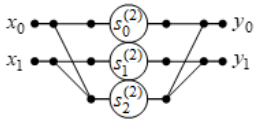

Fig. 1. The data flow graph of the proposed algorithm for the computation of two-point DST-IV

2.3. Algorithm for 3-point DST-IV

Next, we develop the algorithm for three-point DST-IV which is expressed as

$$\mathbf{Y}_{3\times1} = \mathbf{C}_3\mathbf{X}_{3\times1} \tag{6}$$

where:

$$\mathbf{X}_{3\times1} = [x_0, x_1, x_2]^T, \quad \mathbf{Y}_{3\times1} = [y_0, y_1, y_2]^T,$$

$$\mathbf{C}_3 = \begin{bmatrix} a_3 & b_3 & c_3 \\ b_3 & b_3 & -b_3 \\ c_3 & -b_3 & a_3 \end{bmatrix},$$

$$a_3 = \sqrt{\frac{2}{3}}\sin\left(\frac{\pi}{12}\right) \approx 0.2113,\ b_3 = \sqrt{\frac{2}{3}}\sin\left(\frac{3\pi}{12}\right) \approx 0.5774,$$

$$c_3 = \sqrt{\frac{2}{3}}\sin\left(\frac{5\pi}{12}\right) \approx 0.7887.$$

We can decompose the matrix $\mathbf{C}_3$ into two components:

$$\mathbf{C}_3 = \mathbf{C}_3^{(a)} + \mathbf{C}_3^{(b)} \tag{7}$$

where

$$\mathbf{C}_3^{(a)} = \begin{bmatrix} & b_3 & \\ b_3 & b_3 & -b_3 \\ & -b_3 & \end{bmatrix}, \quad \mathbf{C}_3^{(b)} = \begin{bmatrix} a_3 & 0 & c_3 \\ 0 & 0 & 0 \\ c_3 & 0 & a_3 \end{bmatrix}.$$

Matrix $\mathbf{C}_3^{(a)}$ has the same entries except on the sign in the second column and second row, which allows for reducing the number of operations without the need for further transformations. After eliminating the rows and columns containing only zero entries in the last matrix, we obtain

$$\mathbf{C}_2^{(a)} = \begin{bmatrix} a_3 & c_3 \\ c_3 & a_3 \end{bmatrix}.$$

Based on the properties of structural matrices [5, 21], the computational procedure for the three-point DST-IV is represented by expression

$$\mathbf{Y}_{3\times1} = \mathbf{W}_{3\times4}\mathbf{W}_4\mathbf{D}_4\mathbf{W}_{4\times3}\mathbf{W}_3\mathbf{X}_{3\times1}, \tag{8}$$

where:

$$\mathbf{D}_4 = \operatorname{diag}\left(s_0^{(3)}, s_1^{(3)}, s_2^{(3)}, s_3^{(3)}\right),$$

$$s_0^{(3)} = \frac{a_3 + c_3}{2}, \quad s_1^{(3)} = \frac{a_3 - c_3}{2}, \quad s_2^{(3)} = s_2^{(3)} = b_3,$$

$$\mathbf{W}_{3\times4} = \begin{bmatrix} 1 & & 1 & \\ & & 1 & 1 \\ & 1 & -1 & \end{bmatrix}, \quad \mathbf{W}_3 = \begin{bmatrix} 1 & & 1 \\ & 1 & \\ 1 & & -1 \end{bmatrix},$$

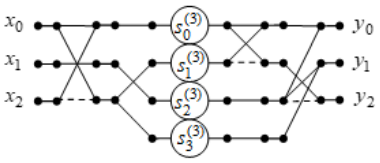

Fig. 2. The data flow graph of the proposed algorithm for the computation of three-point DST-IV

$$\mathbf{W}_{4\times3} = \begin{bmatrix} 1 & & \\ & & 1 \\ & 1 & \\ & & 1 \end{bmatrix}, \quad \mathbf{W}_4 = \mathbf{H}_2 \otimes \mathbf{I}_2,$$

A data flow graph of the proposed algorithm for the three-point DST-IV is shown on Figure 2. The number of multiplications may be reduced from 9 to 4, the number of additions is increased from 6 to 7, however.

2.4. Algorithm for 4-point DST-IV

To design the algorithm for four-point DST-IV we can rewrite (2) as

$$\mathbf{Y}_{4\times1} = \mathbf{C}_4\mathbf{X}_{4\times1}, \tag{9}$$

$$\mathbf{X}_{4\times1} = [x_0, x_1, x_2, x_3]^T, \quad \mathbf{Y}_{4\times1} = [y_0, y_1, y_2, y_3]^T,$$

$$\mathbf{C}_4 = \begin{bmatrix} a_4 & b_4 & c_4 & d_4 \\ b_4 & d_4 & a_4 & -c_4 \\ c_4 & a_4 & -d_4 & b_4 \\ d_4 & -c_4 & b_4 & -a_4 \end{bmatrix},$$

$$a_4 = \sqrt{\frac{1}{2}}\sin\left(\frac{\pi}{16}\right) \approx 0.1379,\ b_4 = \sqrt{\frac{1}{2}}\sin\left(\frac{3\pi}{16}\right) \approx 0.3928,$$

$$c_4 = \sqrt{\frac{1}{2}}\sin\left(\frac{5\pi}{16}\right) \approx 0.5879,\ d_4 = \sqrt{\frac{1}{2}}\sin\left(\frac{7\pi}{16}\right) \approx 0.6935.$$

To change the order of columns and rows of $\mathbf{C}_4$ the permutations

$$\pi_4^{(0)} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 2 & 1 & 4 \end{pmatrix}, \quad \pi_4^{(1)} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 3 & 2 \end{pmatrix}.$$

are defined. After permutation the columns of $\mathbf{C}_4$ according to $\pi_4^{(0)}$ and permutation the rows of $\mathbf{C}_4$ according to $\pi_4^{(1)}$ we obtain the matrix

$$\mathbf{C}_4^{(a)} = \begin{bmatrix} c_4 & b_4 & a_4 & d_4 \\ b_4 & -c_4 & d_4 & -a_4 \\ -d_4 & a_4 & c_4 & b_4 \\ a_4 & d_4 & b_4 & -c_4 \end{bmatrix}$$

with permutation matrices

$$\mathbf{P}_4^{(0)} = \begin{bmatrix} & & 1 & \\ & 1 & & \\ 1 & & & \\ & & & 1 \end{bmatrix}, \quad \mathbf{P}_4^{(1)} = \begin{bmatrix} 1 & & & \\ & & & 1 \\ & & 1 & \\ & 1 & & \end{bmatrix}.$$

The matrix $\mathbf{C}_4^{(a)}$ matches the matrix pattern

$$\mathbf{C}_4^{(a)} = \begin{bmatrix} \mathbf{A}_2 & \mathbf{B}_2 \\ \mathbf{C}_2^{(a)} & \mathbf{A}_2 \end{bmatrix},$$

where

$$\mathbf{A}_2 = \begin{bmatrix} c_4 & b_4 \\ b_4 & -c_4 \end{bmatrix}, \mathbf{B}_2 = \begin{bmatrix} a_4 & d_4 \\ d_4 & -a_4 \end{bmatrix}, \mathbf{C}_2^{(a)} = \begin{bmatrix} -d_4 & a_4 \\ a_4 & d_4 \end{bmatrix}.$$

Hence the matrix $\mathbf{C}_4^{(a)}$ can be represented as [5, 30]

$$\begin{aligned} \mathbf{C}_4^{(a)} &= (\mathbf{T}_{2\times3}^{(3)} \otimes \mathbf{I}_2) \\ &\cdot \left[(\mathbf{C}_2^{(a)} - \mathbf{A}_2) \oplus (\mathbf{B}_2 - \mathbf{A}_2) \oplus \mathbf{A}_2 \right] (\mathbf{T}_{3\times2}^{(3)} \otimes \mathbf{I}_2), \end{aligned} \tag{10}$$

where:

$$\mathbf{T}_{2\times3}^{(3)} = \begin{bmatrix} & 1 & 1 \\ 1 & & 1 \end{bmatrix}, \quad \mathbf{T}_{3\times2}^{(3)} = \begin{bmatrix} 1 & \\ & 1 \\ 1 & 1 \end{bmatrix}.$$

Considering the structure of the resulting matrices

$$\mathbf{C}_2^{(a)} - \mathbf{A}_2 = \begin{bmatrix} -d_4 - c_4 & a_4 - b_4 \\ a_4 - b_4 & d_4 + c_4 \end{bmatrix},$$

$$\mathbf{B}_2 - \mathbf{A}_2 = \begin{bmatrix} a_4 - c_4 & d_4 - b_4 \\ d_4 - b_4 & -a_4 + c_4 \end{bmatrix},$$

and $\mathbf{A}_2$, we note that these matrices match the patterns

$$\begin{bmatrix} a & b \\ b & -a \end{bmatrix}, \quad \begin{bmatrix} c & d \\ d & -c \end{bmatrix}, \quad \begin{bmatrix} e & f \\ f & -e \end{bmatrix},$$

respectively.

Here $a = -d_4 - c_4$; $b = a_4 - b_4$; $c = a_4 - c_4$; $d = d_4 - b_4$; $e = c_4$ and $f = b_4$. Then [5, 6]:

$$\begin{aligned} \mathbf{C}_2^{(a)} - \mathbf{A}_2 &= \mathbf{T}_{2\times3}^{(4)} \operatorname{diag}(s_0^{(4)}, s_1^{(4)}, s_2^{(4)}) \mathbf{T}_{3\times2}^{(3)}, \\ \mathbf{B}_2 - \mathbf{A}_2 &= \mathbf{T}_{2\times3}^{(4)} \operatorname{diag}(s_3^{(4)}, s_4^{(4)}, s_5^{(4)}) \mathbf{T}_{3\times2}^{(3)}, \\ \mathbf{A}_2 &= \mathbf{T}_{2\times3}^{(4)} \operatorname{diag}(s_6^{(4)}, s_7^{(4)}, s_8^{(4)}) \mathbf{T}_{3\times2}^{(3)}, \end{aligned} \tag{11}$$

where:

$$\begin{aligned} s_0^{(4)} &= -d_4 - c_4 - a_4 + b_4; \quad s_1^{(4)} = d_4 + c_4 - a_4 + b_4; \\ s_2^{(4)} &= a_4 - b_4; \quad s_3^{(4)} = a_4 - c_4 - d_4 + b_4; \\ s_4^{(4)} &= -a_4 + c_4 - d_4 + b_4; \quad s_5^{(4)} = d_4 - b_4; \\ s_6^{(4)} &= c_4 - b_4; \quad s_7^{(4)} = -c_4 - b_4; \quad s_8^{(4)} = b_4 \end{aligned}$$

and

$$\mathbf{T}_{2\times3}^{(4)} = \begin{bmatrix} 1 & & 1 \\ & 1 & 1 \end{bmatrix}.$$

Based on properties of structural matrices [5, 29], the computational procedure for the four-point DST-IV is represented by formula

$$\mathbf{Y}_{4\times1} = \mathbf{P}_4^{(1)} \mathbf{W}_{4\times6} \mathbf{W}_{6\times9} \mathbf{D}_9 \mathbf{W}_{9\times6} \mathbf{W}_{6\times4} \mathbf{P}_4^{(0)} \mathbf{X}_{4\times1}, \tag{12}$$

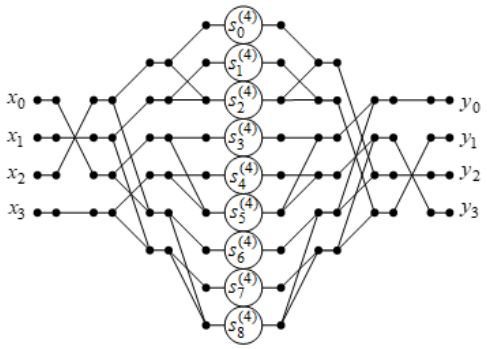

Fig. 3. The data flow graph of the proposed algorithm for the computation of four-point DST-IV

where:

$$\begin{aligned} \mathbf{D}_9 &= \operatorname{diag}(s_0^{(4)}, s_1^{(4)}, s_2^{(4)}, s_3^{(4)}, s_4^{(4)}, s_5^{(4)}, s_6^{(4)}, s_7^{(4)}, s_8^{(4)}), \\ \mathbf{W}_{4\times6} &= \mathbf{T}_{2\times3}^{(3)} \otimes \mathbf{I}_2, \quad \mathbf{W}_{6\times9} = \mathbf{T}_{2\times3}^{(4)} \oplus \mathbf{T}_{2\times3}^{(4)} \oplus \mathbf{T}_{2\times3}^{(4)}, \\ \mathbf{W}_{6\times4} &= \mathbf{T}_{3\times2}^{(3)} \otimes \mathbf{I}_2, \quad \mathbf{W}_{9\times6} = \mathbf{T}_{3\times2}^{(4)} \oplus \mathbf{T}_{3\times2}^{(4)} \oplus \mathbf{T}_{3\times2}^{(4)}. \end{aligned}$$

A data flow graph of the proposed four-point DST-IV algorithm is presented on Figure 3. In particularly, the number of multiplications may be reduced from 16 to 9, although the number of additions is increased from 12 to 15.

2.5. Algorithm for 5-point DST-IV

Let's obtain the algorithm for five-point DST-IV which is expressed as follows:

$$\mathbf{Y}_{5\times1} = \mathbf{C}_5 \mathbf{X}_{5\times1}, \tag{13}$$

$$\mathbf{X}_{5\times1} = [x_0, x_1, x_2, x_3, x_4]^T, \quad \mathbf{Y}_{5\times1} = [y_0, y_1, y_2, y_3, y_4]^T,$$

$$\mathbf{C}_5 = \begin{bmatrix} a_5 & b_5 & c_5 & d_5 & e_5 \\ b_5 & e_5 & c_5 & -a_5 & -d_5 \\ c_5 & c_5 & -c_5 & -c_5 & -c_5 \\ d_5 & -a_5 & -c_5 & e_5 & -b_5 \\ e_5 & -d_5 & c_5 & -b_5 & a_5 \end{bmatrix},$$

$$a_5 = \sqrt{\frac{2}{5}} \sin\left(\frac{\pi}{20}\right) \approx 0.0989, \; b_5 = \sqrt{\frac{2}{5}} \sin\left(\frac{3\pi}{20}\right) \approx 0.2871,$$

$$c_5 = \sqrt{\frac{2}{5}} \sin\left(\frac{\pi}{4}\right) \approx 0.4472, \quad d_5 = \sqrt{\frac{2}{5}} \sin\left(\frac{7\pi}{20}\right) \approx 0.5636,$$

$$e_5 = \sqrt{\frac{2}{5}} \sin\left(\frac{9\pi}{20}\right) \approx 0.6247.$$

To change the order of columns and rows of $\mathbf{C}_5$ the permutations $\pi_5^{(0)}$ and $\pi_5^{(1)}$ are defined as

$$\pi_5^{(0)} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 5 & 4 \end{pmatrix}, \quad \pi_5^{(1)} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 1 & 3 & 4 & 5 \end{pmatrix}.$$

The columns of $\mathbf{C}_5$ are permutated according to $\pi_5^{(0)}$ and the rows of $\mathbf{C}_5$ are permutated according to $\pi_5^{(1)}$. After permutations and altering the sign in fourth row and the fourth column,

the obtained matrix $\mathbf{C}_5^{(a)}$ is decomposed into two components

$$\mathbf{C}_5^{(a)} = \mathbf{C}_5^{(b)} + \mathbf{C}_5^{(c)}, \tag{14}$$

where

$$\mathbf{C}_5^{(b)} = \begin{bmatrix} & & c_5 & & \\ & & c_5 & & \\ c_5 & c_5 & -c_5 & -c_5 & -c_5 \\ & & c_5 & & \\ & & c_5 & & \end{bmatrix},$$

$$\mathbf{C}_5^{(c)} = \begin{bmatrix} b_5 & e_5 & d_5 & -a_5 \\ a_5 & b_5 & -e_5 & d_5 \\ & & & \\ -d_5 & a_5 & -b_5 & -e_5 \\ e_5 & -d_5 & -a_5 & -b_5 \end{bmatrix}.$$

Matrix $\mathbf{C}_5^{(b)}$ has the same entries except on the sign in the third column and third row, which allows decreasing the number of operations without the need for further transformations. After eliminating the rows and columns containing only zero entries in matrix $\mathbf{C}_5^{(c)}$, we obtain matrix $\mathbf{C}_4^{(d)}$:

$$\mathbf{C}_4^{(d)} = \begin{bmatrix} b_5 & e_5 & d_5 & -a_5 \\ a_5 & b_5 & -e_5 & d_5 \\ -d_5 & a_5 & -b_5 & -e_5 \\ e_5 & -d_5 & -a_5 & -b_5 \end{bmatrix}.$$

The matrix $\mathbf{C}_4^{(d)}$ matches the matrix pattern

$$\mathbf{C}_4^{(d)} = \begin{bmatrix} \mathbf{A}_2^{(0)} & \mathbf{B}_2^{(0)} \\ -\mathbf{B}_2^{(0)} & -\mathbf{A}_2^{(0)} \end{bmatrix},$$

where

$$\mathbf{A}_2^{(0)} = \begin{bmatrix} b_5 & e_5 \\ a_5 & b_5 \end{bmatrix}, \quad \mathbf{B}_2^{(0)} = \begin{bmatrix} d_5 & -a_5 \\ -e_5 & d_5 \end{bmatrix}.$$

Hence the matrix $\mathbf{C}_4^{(d)}$ can be represented as [5, 30]

$$\mathbf{C}_4^{(d)} = (\bar{\mathbf{I}}_2 \otimes \mathbf{I}_2)(\mathbf{H}_2 \otimes \mathbf{I}_2) \cdot \frac{1}{2}\left[(\mathbf{A}_2^{(0)} + \mathbf{B}_2^{(0)}) \oplus (\mathbf{A}_2^{(0)} - \mathbf{B}_2^{(0)})\right](\mathbf{H}_2 \otimes \mathbf{I}_2), \tag{15}$$

where

$$\bar{\mathbf{I}}_2 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

Considering the structure of the resulting matrices $\mathbf{A}_2^{(0)} + \mathbf{B}_2^{(0)}$ and $\mathbf{A}_2^{(0)} - \mathbf{B}_2^{(0)}$ we note that these matrices match the patterns

$$\begin{bmatrix} a & b \\ d & -a \end{bmatrix}, \quad \begin{bmatrix} c & d \\ d & c \end{bmatrix},$$

respectively. For that it is necessary to alter the sigh in the second row of $\mathbf{A}_2^{(0)} + \mathbf{B}_2^{(0)}$ matrix. Here $a = b_5 + d_5$; $b = e_5 - a_5$; $c = b_5 - d_5$; $d = e_5 + a_5$. Then [5, 6]

$$\frac{1}{2}\left(\mathbf{A}_2^{(0)} + \mathbf{B}_2^{(0)}\right) = \bar{\mathbf{I}}_2 \mathbf{T}_{2\times 3}^{(4)} \operatorname{diag}\left(\frac{a-b}{2}, \frac{-a-b}{2}, \frac{b}{2}\right) \mathbf{T}_{3\times 2}^{(3)},$$

$$\frac{1}{2}\left(\mathbf{A}_2^{(0)} - \mathbf{B}_2^{(0)}\right) = \mathbf{H}_2 \operatorname{diag}\left(\frac{c+d}{4}, \frac{c-d}{4}\right) \mathbf{H}_2, \tag{16}$$

where $\mathbf{T}_{2\times 3}^{(4)}$ is defined as in (11), $\mathbf{T}_{3\times 2}^{(3)}$ is defined as in (10). Consequently,

$$\mathbf{C}_4^{(d)} = (\bar{\mathbf{I}}_2 \otimes \mathbf{I}_2)(\mathbf{H}_2 \otimes \mathbf{I}_2)\left((\bar{\mathbf{I}}_2 \mathbf{T}_{2\times 3}^{(4)}) \oplus \mathbf{H}_2\right) \cdot \operatorname{diag}(s_0^{(5)}, s_1^{(5)}, s_2^{(5)}, s_5^{(5)}, s_6^{(5)})(\mathbf{T}_{3\times 2}^{(3)} \oplus \mathbf{H}_2)(\mathbf{H}_2 \otimes \mathbf{I}_2), \tag{17}$$

where

$$s_0^{(5)} = \frac{b_5 + d_5 - e_5 + a_5}{2}; \quad s_1^{(5)} = \frac{-b_5 - d_5 - e_5 + a_5}{2};$$
$$s_2^{(5)} = \frac{e_5 - a_5}{2}; \quad s_3^{(5)} = s_4^{(5)} = c_5;$$
$$s_5^{(5)} = \frac{b_5 - d_5 + e_5 + a_5}{4}; \quad s_6^{(5)} = \frac{b_5 - d_5 - e_5 - a_5}{4}.$$

Taken in account properties of structural matrices [5, 21], the computational procedure for the five-point DST-IV is represented by expression

$$\mathbf{Y}_{5\times 1} = \mathbf{P}_5^{(1)} \mathbf{W}_{5\times 6} \mathbf{W}_6 \mathbf{W}_{6\times 7} \mathbf{D}_7 \mathbf{W}_{7\times 6} \mathbf{W}_6 \mathbf{W}_{6\times 5} \mathbf{P}_5^{(0)} \mathbf{X}_{5\times 1}, \tag{18}$$

where:

$$\mathbf{W}_{6\times 7} = \bar{\mathbf{I}}_2 \mathbf{T}_{2\times 3}^{(4)} \oplus \mathbf{I}_2 \oplus \mathbf{H}_2,$$
$$\mathbf{W}_{7\times 6} = \mathbf{T}_{3\times 2}^{(3)} \oplus \mathbf{I}_2 \oplus \mathbf{H}_2,$$
$$\mathbf{D}_7 = \operatorname{diag}(s_0^{(5)}, s_1^{(5)}, s_2^{(5)}, s_3^{(5)}, s_4^{(5)}, s_5^{(5)}, s_6^{(5)}),$$

$$\mathbf{P}_5^{(1)} = \begin{bmatrix} & 1 & & & \\ 1 & & & & \\ & & 1 & & \\ & & & -1 & \\ & & & & 1 \end{bmatrix}, \quad \mathbf{P}_5^{(0)} = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & & -1 \\ & & & 1 & \end{bmatrix},$$

$$\mathbf{W}_{6\times 5} = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ 1 & 1 & & -1 & -1 \\ & & & 1 & \\ & & & & 1 \end{bmatrix},$$

$$\mathbf{W}_6 = \begin{bmatrix} 1 & & & & 1 & \\ & 1 & & & & 1 \\ & & 1 & & & \\ & & & 1 & & \\ 1 & & & & -1 & \\ & 1 & & & & -1 \end{bmatrix},$$

$$\mathbf{W}_{5\times 6} = \begin{bmatrix} 1 & & 1 & & & \\ & 1 & 1 & & & \\ & & -1 & 1 & & \\ & & 1 & & -1 & \\ & & 1 & & & -1 \end{bmatrix}.$$

A data flow graph of the proposed algorithm of the five-point DST-IV is presented in Figure 4. In particularly, the number of multiplications may be reduced from 25 to 7, but the number of additions is increased from 20 to 23.

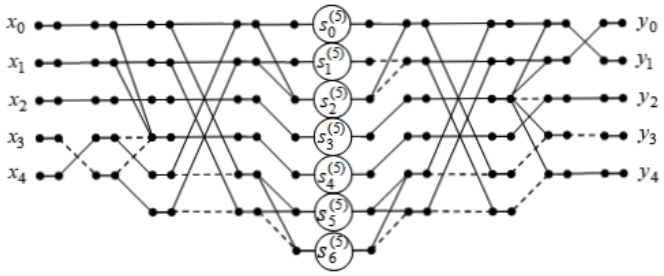

Fig. 4. The data flow graph of the proposed algorithm for the computation of five-point DST-IV

2.6. Algorithm for 6-point DST-IV

Now we propose the algorithm for six-point DST-IV. The six-point DST-IV is expressed as follows:

$$\mathbf{Y}_{6\times1} = \mathbf{C}_6\mathbf{X}_{6\times1}, \tag{19}$$

where:

$$\mathbf{X}_{6\times1} = [x_0, x_1, x_2, x_3, x_4, x_5]^T,$$
$$\mathbf{Y}_{6\times1} = [y_0, y_1, y_2, y_3, y_4, y_5]^T,$$

$$\mathbf{C}_6 = \begin{bmatrix} a_6 & b_6 & c_6 & d_6 & e_6 & f_6 \\ b_6 & e_6 & e_6 & b_6 & -b_6 & -e_6 \\ c_6 & e_6 & -a_6 & -f_6 & -b_6 & d_6 \\ d_6 & b_6 & -f_6 & a_6 & e_6 & -c_6 \\ e_6 & -b_6 & -b_6 & e_6 & -e_6 & b_6 \\ f_6 & -e_6 & d_6 & -c_6 & b_6 & -a_6 \end{bmatrix},$$

$$a_6 = \sqrt{\frac{1}{3}}\sin\left(\frac{\pi}{24}\right) \approx 0.0754, \quad b_6 = \sqrt{\frac{1}{3}}\sin\left(\frac{\pi}{8}\right) \approx 0.2209,$$
$$c_6 = \sqrt{\frac{1}{3}}\sin\left(\frac{5\pi}{24}\right) \approx 0.3515, \; d_6 = \sqrt{\frac{1}{3}}\sin\left(\frac{7\pi}{24}\right) \approx 0.4580,$$
$$e_6 = \sqrt{\frac{1}{3}}\sin\left(\frac{9\pi}{24}\right) \approx 0.5334, \; f_6 = \sqrt{\frac{1}{3}}\sin\left(\frac{11\pi}{24}\right) \approx 0.5724.$$

The columns and rows of $\mathbf{C}_6$ are permutated according to π_6 which is defined in the following form

$$\pi_6 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 6 & 2 & 3 & 5 & 4 & 1 \end{pmatrix}.$$

In addition, the sign is altered in third row and the third column. Then the matrix $\mathbf{C}_6^{(a)}$ is obtained. The matrix $\mathbf{C}_6^{(a)}$ matches the matrix pattern

$$\mathbf{C}_6^{(a)} = \begin{bmatrix} \mathbf{A}_3 & \mathbf{B}_3 \\ \mathbf{B}_3 & -\mathbf{A}_3 \end{bmatrix},$$

where

$$\mathbf{A}_3 = \begin{bmatrix} -a_6 & -e_6 & -d_6 \\ -e_6 & e_6 & -e_6 \\ -d_6 & -e_6 & -a_6 \end{bmatrix}, \quad \mathbf{B}_3 = \begin{bmatrix} -c_6 & b_6 & f_6 \\ b_6 & -b_6 & b_6 \\ f_6 & b_6 & -c_6 \end{bmatrix}.$$

Hence the matrix $\mathbf{C}_6^{(a)}$ can be represented as [5, 29]

$$\mathbf{C}_6^{(a)} = (\mathbf{T}_{2\times3}^{(4} \otimes \mathbf{I}_3) \cdot [(\mathbf{A}_3 - \mathbf{B}_3) \oplus (-\mathbf{A}_3 - \mathbf{B}_3) \oplus \mathbf{B}_3] (\mathbf{T}_{3\times2}^{(3)} \otimes \mathbf{I}_3) \tag{20}$$

Then we alter the sign in second column of each resulted matrices $\mathbf{A}_3 - \mathbf{B}_3$, $-\mathbf{A}_3 - \mathbf{B}_3$, $\mathbf{B}_3$, and represent the resulting matrices

$$(\mathbf{A}_3 - \mathbf{B}_3)^{(0)} = \begin{bmatrix} -a_6 + c_6 & -e_6 - b_6 & d_6 + f_6 \\ -e_6 - b_6 & e_6 + b_6 & e_6 + b_6 \\ -d_6 - f_6 & -e_6 - b_6 & a_6 - c_6 \end{bmatrix},$$

$$(-\mathbf{A}_3 - \mathbf{B}_3)^{(0)} = \begin{bmatrix} a_6 + c_6 & e_6 - b_6 & -d_6 + f_6 \\ e_6 - b_6 & -e_6 + b_6 & -e_6 + b_6 \\ d_6 - f_6 & e_6 - b_6 & -a_6 - c_6 \end{bmatrix},$$

$$\mathbf{B}_3^{(0)} = \begin{bmatrix} -c_6 & b_6 & -f_6 \\ b_6 & -b_6 & -b_6 \\ f_6 & b_6 & c_6 \end{bmatrix}$$

as

$$(\mathbf{A}_3 - \mathbf{B}_3)^{(0)} = \mathbf{C}_3^{(c)} + \mathbf{C}_3^{(d)},$$
$$(-\mathbf{A}_3 - \mathbf{B}_3)^{(0)} = \mathbf{C}_3^{(e)} + \mathbf{C}_3^{(f)}, \tag{21}$$
$$\mathbf{B}_3^{(0)} = \mathbf{B}_3^{(a)} + \mathbf{B}_3^{(b)},$$

where

$$\mathbf{C}_3^{(c)} = \begin{bmatrix} & -e_6 - b_6 & \\ -e_6 - b_6 & e_6 + b_6 & e_6 + b_6 \\ & -e_6 - b_6 & \end{bmatrix},$$

$$\mathbf{C}_3^{(d)} = \begin{bmatrix} -a_6 + c_6 & -d_6 + f_6 \\ & \\ -d_6 - f_6 & a_6 - c_6 \end{bmatrix}$$

$$\mathbf{C}_3^{(e)} = \begin{bmatrix} & e_6 - b_6 & \\ e_6 - b_6 & -e_6 + b_6 & -e_6 + b_6 \\ & e_6 - b_6 & \end{bmatrix},$$

$$\mathbf{C}_3^{(f)} = \begin{bmatrix} a_6 + c_6 & -d_6 + f_6 \\ & \\ d_6 - f_6 & -a_6 - c_6 \end{bmatrix},$$

$$\mathbf{B}_3^{(a)} = \begin{bmatrix} & b_6 & \\ b_6 & -b_6 & -b_6 \\ & b_6 & \end{bmatrix}, \mathbf{B}_3^{(b)} = \begin{bmatrix} -c_6 & -f_6 \\ & \\ f_6 & c_6 \end{bmatrix}.$$

Further we eliminate the zeros from the matrices $\mathbf{C}_3^{(d)}$, $\mathbf{C}_3^{(f)}$, $\mathbf{B}_3^{(b)}$ and obtain the matrices

$$\mathbf{C}_2^{(g)} = \begin{bmatrix} -a_6 + c_6 & d_6 + f_6 \\ -d_6 - f_6 & a_6 - c_6 \end{bmatrix}, \mathbf{C}_2^{(h)} = \begin{bmatrix} a_6 + c_6 & -d_6 + f_6 \\ d_6 - f_6 & -a_6 - c_6 \end{bmatrix},$$
$$\mathbf{B}_2^{(c)} = \begin{bmatrix} -c_6 & -f_6 \\ f_6 & c_6 \end{bmatrix}.$$

Then we note that the matrices match the pattern

$$\begin{bmatrix} a & b \\ -b & -a \end{bmatrix}.$$

Here $a=-a_6+c_6$; $b=d_6+f_6$ for $\mathbf{C}_2^{(g)}$; $a=a_6+c_6$; $b=-d_6+f_6$ for $\mathbf{C}_2^{(h)}$; $a=-c_6$; $b=-f_6$ for $\mathbf{B}_2^{(c)}$; then [5, 6]

$$\begin{aligned}\mathbf{C}_2^{(g)} &= \bar{\mathbf{I}}_2\,\mathbf{H}_2\mathrm{diag}\left(s_0^{(6)},s_1^{(6)}\right)\mathbf{H}_2,\\ \mathbf{C}_2^{(h)} &= \bar{\mathbf{I}}_2\,\mathbf{H}_2\mathrm{diag}\left(s_4^{(6)},s_5^{(6)}\right)\mathbf{H}_2,\\ \mathbf{B}_2^{(c)} &= \bar{\mathbf{I}}_2\,\mathbf{H}_2\mathrm{diag}\left(s_8^{(6)},s_9^{(6)}\right)\mathbf{H}_2,\end{aligned} \tag{22}$$

where

$$\begin{aligned}s_0^{(6)} &= \frac{-a_6+c_6+d_6+f_6}{2}; s_1^{(6)} = \frac{-a_6+c_6-d_6-f_6}{2};\\ s_4^{(6)} &= \frac{a_6+c_6-d_6+f_6}{2}; s_5^{(6)} = \frac{a_6+c_6+d_6-f_6}{2};\\ s_8^{(6)} &= \frac{-c_6-f_6}{2}; s_9^{(6)} = \frac{-c_6+f_6}{2}.\end{aligned}$$

Taken in account properties of structural matrices [5, 21], the computational procedure for the six-point DST-IV is represented by expression

$$\mathbf{Y}_{6\times1} = \mathbf{P}_6\mathbf{W}_{6\times9}\mathbf{W}_{9\times12}\mathbf{W}_{12}\mathbf{D}_{12}\mathbf{W}_{12\times9}\mathbf{W}_9\mathbf{W}_{9\times6}\mathbf{P}_6\mathbf{X}_{6\times1}, \tag{23}$$

where

$$\begin{aligned}\mathbf{W}_{6\times9} &= \mathbf{T}_{2\times3}^{(4)}\otimes\mathbf{I}_3; \quad \mathbf{W}_{9\times6} = \mathbf{T}_{3\times2}^{(3)}\otimes\mathbf{I}_3;\\ \mathbf{W}_{9\times12} &= \mathbf{W}_{3\times4}\oplus\mathbf{W}_{3\times4}\oplus\mathbf{W}_{3\times4};\\ \mathbf{W}_{12} &= \bar{\mathbf{I}}_2\,\mathbf{H}_2\oplus\mathbf{I}_2\oplus\bar{\mathbf{I}}_2\,\mathbf{H}_2\oplus\mathbf{I}_2\oplus\bar{\mathbf{I}}_2\,\mathbf{H}_2\oplus\mathbf{I}_2;\\ \mathbf{W}_{12\times9} &= \mathbf{W}_{4\times3}\oplus\mathbf{W}_{4\times3}\oplus\mathbf{W}_{4\times3};\\ \mathbf{W}_9 &= \mathbf{W}_3\,\bar{\mathbf{I}}_3\oplus\mathbf{W}_3\,\bar{\mathbf{I}}_3\oplus\mathbf{W}_3\,\bar{\mathbf{I}}_3;\end{aligned}$$

$$\mathbf{W}_{3\times4} = \begin{bmatrix}1 & & 1\\ & -1 & 1\\ 1 & 1 & \end{bmatrix}; \mathbf{W}_3 = \begin{bmatrix}1 & & 1\\ & 1 & \\ 1 & & -1\end{bmatrix};$$

$$\mathbf{W}_{4\times3} = \begin{bmatrix}1 & & \\ & & 1\\ & 1 & \\ & & 1\end{bmatrix}; \bar{\mathbf{I}}_3 = \begin{bmatrix}1 & & \\ & 1 & \\ & & -1\end{bmatrix};$$

$$\mathbf{P}_6 = \begin{bmatrix} & & & & & 1\\ & 1 & & & & \\ & & -1 & & & \\ & & & 1 & & \\ & & & & 1 & \\ 1 & & & & & \end{bmatrix};$$

$$\begin{aligned}\mathbf{D}_{12} = \mathrm{diag}\Big(&s_0^{(6)},s_1^{(6)},s_2^{(6)},s_3^{(6)},s_4^{(6)},s_5^{(6)},\\ &s_6^{(6)},s_7^{(6)},s_8^{(6)},s_9^{(6)},s_{10}^{(6)},s_{11}^{(6)}\Big);\end{aligned}$$

$$\begin{aligned}s_2^{(6)} &= -e_6-b_6; \quad s_3^{(6)} = -e_6-b_6; \quad s_6^{(6)} = e_6-b_6;\\ s_7^{(6)} &= e_6-b_6; \quad s_{10}^{(6)} = b_6; \quad s_{11}^{(6)} = b_6.\end{aligned}$$

A data flow graph of the proposed algorithm of the five-point DST-IV is presented on Figure 5. In particularly, the number of multiplication operations may be reduced from 36 to 12, but the number of addition operations is the same.

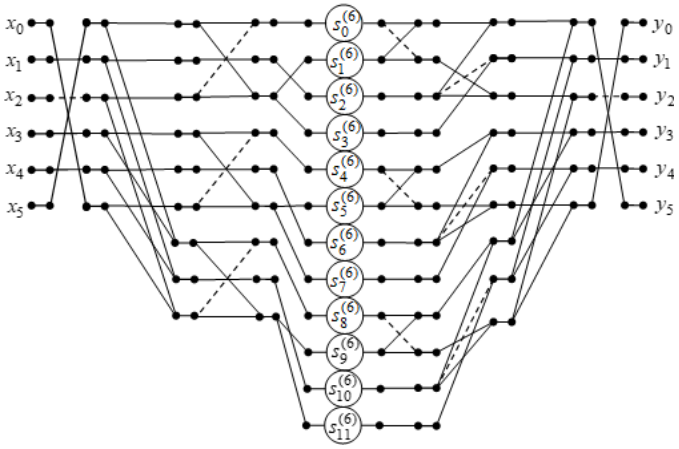

Fig. 5. The data flow graph of the proposed algorithm for the computation of six-point DST-IV

2.7. Algorithm for 7-point DST-IV

To elaborate the algorithm for seven-point DST-IV the formula of this transform is expressed as follows:

$$\mathbf{Y}_{7\times1} = \mathbf{C}_7\mathbf{X}_{7\times1}, \tag{24}$$

where:

$$\begin{aligned}\mathbf{X}_{7\times1} &= [x_0,x_1,x_2,x_3,x_4,x_5,x_6]^T,\\ \mathbf{Y}_{7\times1} &= [y_0,y_1,y_2,y_3,y_4,y_5,y_6]^T,\end{aligned}$$

$$\mathbf{C}_7 = \begin{bmatrix}a_7 & b_7 & c_7 & d_7 & e_7 & f_7 & g_7\\ b_7 & e_7 & g_7 & d_7 & a_7 & -c_7 & -f_7\\ c_7 & g_7 & b_7 & -d_7 & -f_7 & -a_7 & e_7\\ d_7 & d_7 & -d_7 & -d_7 & d_7 & d_7 & -d_7\\ e_7 & a_7 & -f_7 & d_7 & b_7 & -g_7 & c_7\\ f_7 & -c_7 & -a_7 & d_7 & -g_7 & e_7 & -b_7\\ g_7 & -f_7 & e_7 & -d_7 & c_7 & -b_7 & a_7\end{bmatrix},$$

$$a_7 = \sqrt{\frac{2}{7}}\sin\left(\frac{\pi}{28}\right)\approx0.0598,\ b_7 = \sqrt{\frac{2}{7}}\sin\left(\frac{3\pi}{28}\right)\approx0.1765,$$

$$c_7 = \sqrt{\frac{2}{7}}\sin\left(\frac{5\pi}{28}\right)\approx0.2844,\ d_7 = \sqrt{\frac{2}{7}}\sin\left(\frac{\pi}{4}\right)\approx0.3780,$$

$$e_7 = \sqrt{\frac{2}{7}}\sin\left(\frac{9\pi}{28}\right)\approx0.4526,\ f_7 = \sqrt{\frac{2}{7}}\sin\left(\frac{11\pi}{28}\right)\approx0.5045,$$

$$g_7 = \sqrt{\frac{2}{7}}\sin\left(\frac{13\pi}{28}\right)\approx0.5312.$$

To change the order of columns and rows, we define the permutation π_7 in the following form

$$\pi_7 = \begin{pmatrix}1 & 2 & 3 & 4 & 5 & 6 & 7\\ 3 & 2 & 1 & 4 & 5 & 6 & 7\end{pmatrix}.$$

Let permute columns and rows of $\mathbf{C}_7$ according to π_7. After permutation and altering the sign in fifth row and fifth column the resulted matrix $\mathbf{C}_7^{(a)}$ is decomposed into two components:

$$\mathbf{C}_7^{(a)} = \mathbf{C}_7^{(b)} + \mathbf{C}_7^{(c)}, \tag{25}$$

where

$$\mathbf{C}_7^{(b)} = \begin{bmatrix} & & & -d_7 & & & \\ & & & d_7 & & & \\ & & & d_7 & & & \\ -d_7 & d_7 & d_7 & -d_7 & d_7 & -d_7 & -d_7 \\ & & & d_7 & & & \\ & & & -d_7 & & & \\ & & & -d_7 & & & \end{bmatrix},$$

$$\mathbf{C}_7^{(c)} = \begin{bmatrix} b_7 & g_7 & c_7 & -f_7 & a_7 & e_7 \\ g_7 & e_7 & b_7 & a_7 & c_7 & -f_7 \\ c_7 & b_7 & a_7 & e_7 & -f_7 & g_7 \\ & & & & & \\ -f_7 & a_7 & e_7 & b_7 & g_7 & c_7 \\ a_7 & c_7 & -f_7 & g_7 & e_7 & b_7 \\ e_7 & -f_7 & g_7 & c_7 & b_7 & a_7 \end{bmatrix}.$$

Matrix $\mathbf{C}_7^{(b)}$ has the same entries regardless of sign in the third column and third row, which allows us to reduce the number of operations without the need for further transformations. After eliminating the rows and columns containing only zero entries in matrix $\mathbf{C}_7^{(c)}$, we obtain matrix $\mathbf{C}_6^{(d)}$

$$\mathbf{C}_6^{(d)} = \begin{bmatrix} b_7 & g_7 & c_7 & -f_7 & a_7 & e_7 \\ g_7 & e_7 & b_7 & a_7 & c_7 & -f_7 \\ c_7 & b_7 & a_7 & e_7 & -f_7 & g_7 \\ -f_7 & a_7 & e_7 & b_7 & g_7 & c_7 \\ a_7 & c_7 & -f_7 & g_7 & e_7 & b_7 \\ e_7 & -f_7 & g_7 & c_7 & b_7 & a_7 \end{bmatrix}.$$

The obtained matrix acquires the structure

$$\mathbf{C}_6^{(d)} = \begin{bmatrix} \mathbf{A}_3^{(a)} & \mathbf{B}_3^{(a)} \\ \mathbf{B}_3^{(a)} & \mathbf{A}_3^{(a)} \end{bmatrix}$$

with

$$\mathbf{A}_3^{(a)} = \begin{bmatrix} b_7 & g_7 & c_7 \\ g_7 & e_7 & b_7 \\ c_7 & b_7 & a_7 \end{bmatrix}, \quad \mathbf{B}_3^{(a)} \begin{bmatrix} -f_7 & a_7 & e_7 \\ a_7 & c_7 & -f_7 \\ e_7 & -f_7 & g_7 \end{bmatrix}.$$

Then [5, 32]

$$\begin{aligned} \mathbf{C}_6^{(d)} =& (\mathbf{H}_2 \otimes \mathbf{I}_3) \\ & \frac{1}{2}\left[(\mathbf{A}_3^{(a)} + \mathbf{B}_3^{(a)}) \oplus (\mathbf{A}_3^{(a)} - \mathbf{B}_3^{(a)})\right](\mathbf{H}_2 \otimes \mathbf{I}_3). \end{aligned} \tag{26}$$

Let's represent the submatrices:

$$\mathbf{A}_3^{(a)} + \mathbf{B}_3^{(a)} = \begin{bmatrix} b_7 - f_7 & a_7 + g_7 & c_7 + e_7 \\ a_7 + g_7 & c_7 + e_7 & b_7 - b_7 \\ c_7 + e_7 & b_7 - f_7 & a_7 + g_7 \end{bmatrix},$$

$$\mathbf{A}_3^{(a)} - \mathbf{B}_3^{(a)} = \begin{bmatrix} b_7 + f_7 & g_7 - a_7 & c_7 - e_7 \\ g_7 - a_7 & e_7 - c_7 & b_7 + f_7 \\ c_7 - e_7 & b_7 + f_7 & a_7 - g_7 \end{bmatrix}$$

of quasi-diagonal matrix $(\mathbf{A}_3^{(a)} + \mathbf{B}_3^{(a)}) \oplus (\mathbf{A}_3^{(a)} - \mathbf{B}_3^{(a)})$ as circular convolution matrices [32]. For that we rearranged the first and second columns in both matrices and then altered the sign in first row and first column of $\mathbf{A}_3^{(a)} - \mathbf{B}_3^{(a)}$ matrix. As a result the matrices $\mathbf{C}_3^{(k)}$ and $\mathbf{C}_3^{(l)}$ were respectively obtained:

$$\mathbf{C}_3^{(k)} = \begin{bmatrix} b_7 - f_7 & c_7 + e_7 & a_7 + g_7 \\ a_7 + g_7 & b_7 - f_7 & c_7 + e_7 \\ c_7 + e_7 & a_7 + g_7 & b_7 - f_7 \end{bmatrix},$$

$$\mathbf{C}_3^{(l)} = \begin{bmatrix} b_7 + f_7 & e_7 - c_7 & a_7 - g_7 \\ a_7 - g_7 & b_7 + f_7 & e_7 - c_7 \\ e_7 - c_7 & a_7 - g_7 & b_7 + f_7 \end{bmatrix}.$$

Further the matrices $\mathbf{C}_3^{(k)}$ and $\mathbf{C}_3^{(l)}$ were factorized in accordance with the expressions for calculating the entries of a circular convolution matrix $\mathbf{H}_3$,

$$\mathbf{H}_3 = \begin{bmatrix} h_0 & h_2 & h_1 \\ h_1 & h_0 & h_2 \\ h_2 & h_1 & h_0 \end{bmatrix}$$

for $N = 3$ [32]:

$$\mathbf{H}_3 = \mathbf{T}_3^{(1)} \mathbf{T}_{3\times 4} \operatorname{diag}(s_0, s_1, s_2, s_3) \mathbf{T}_{4\times 3} \mathbf{T}_3^{(0)}, \tag{27}$$

where

$$s_0 = \frac{h_0 + h_1 + h_2}{3}, \quad s_3 = \frac{h_0 + h_1 - 2h_1}{3},$$
$$s_2 = h_1 - h_2, \quad s_1 = h_0 - h_2,$$

$$\mathbf{T}_3^{(1)} = \begin{bmatrix} 1 & 1 & \\ 1 & -1 & 1 \\ 1 & & 1 \end{bmatrix}, \quad \mathbf{T}_{3\times 4} = \begin{bmatrix} 1 & & & \\ & 1 & & -1 \\ & & 1 & -1 \end{bmatrix},$$

$$\mathbf{T}_{4\times 3} = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \\ & 1 & 1 \end{bmatrix}, \quad \mathbf{T}_3^{(0)} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & & -1 \\ & 1 & -1 \end{bmatrix}.$$

As a result we obtain in (27) $h_0 = b_7 - h_7$, $h_1 = a_7 + g_7$, $h_2 = c_7 + e_7$ for $\mathbf{C}_3^{(k)}$ matrix and $h_0 = b_7 + f_7$, $h_1 = a_7 - g_7$, $h_2 = e_7 - c_7$ for $\mathbf{C}_3^{(l)}$ matrix.

Then

$$\begin{aligned} \mathbf{C}_6^{(d)} =& (\mathbf{H}_2 \otimes \mathbf{I}_3)\left(\mathbf{T}_3^{(1)} \oplus \mathbf{P}_3^{(a)} \mathbf{T}_3^{(1)}\right)(\mathbf{T}_{3\times 4} \oplus \mathbf{T}_{3\times 4}) \\ & \times \operatorname{diag}\left(s_0^{(7)}, s_1^{(7)}, s_2^{(7)}, s_3^{(7)}, s_6^{(7)}, s_7^{(7)}, s_8^{(7)}, s_9^{(7)}\right) \\ & \times (\mathbf{T}_{4\times 3} \oplus \mathbf{T}_{4\times 3})\left(\mathbf{T}_3^{(0)} \oplus \mathbf{T}_3^{(0)}\right)\mathbf{P}_6^{(a)}(\mathbf{H}_2 \otimes \mathbf{I}_3), \end{aligned} \tag{28}$$

where

$$s_0^{(7)} = \frac{b_7 - f_7 + g_7 + a_7 + c_7 + e_7}{6};$$

$$s_1^{(7)} = \frac{b_7 - f_7 - e_7 - c_7}{2};$$
$$s_2^{(7)} = \frac{g_7 + a_7 - c_7 - e_7}{2};$$
$$s_3^{(7)} = \frac{b_7 - f_7 + g_7 + a_7 - 2e_7 - 2c_7}{6};$$
$$s_6^{(7)} = \frac{b_7 + f_7 + a_7 - g_7 + e_7 - c_7}{6};$$
$$s_7^{(7)} = \frac{b_7 + f_7 - e_7 + c_7}{2};$$
$$s_8^{(7)} = \frac{a_7 - g_7 - e_7 + c_7}{2};$$
$$s_9^{(7)} = \frac{b_7 + f_7 + a_7 - g_7 - 2e_7 + 2c_7}{6};$$

$$\mathbf{P}_3^{(a)} = \begin{bmatrix} -1 & & \\ & 1 & \\ & & 1 \end{bmatrix}, \mathbf{P}_6^{(a)} = \begin{bmatrix} 1 & & & & & & \\ & & 1 & & & & \\ & 1 & & & & & \\ & & & -1 & & & \\ & & & & & 1 & \\ & & & & 1 & & \end{bmatrix}.$$

Based on properties of structural matrices [5, 32], the computational procedure for the seven-point DST-IV is represented by the expression:

$$\begin{aligned} \mathbf{Y}_{7\times1} = &\mathbf{P}_7\mathbf{W}_{7\times8}\mathbf{W}_8^{(0)}\mathbf{W}_8^{(1)}\mathbf{W}_{8\times10}\mathbf{D}_{10} \\ &\cdot\mathbf{W}_{10\times8}\mathbf{W}_8^{(2)}\mathbf{P}_8^{(a)}\mathbf{W}_8^{(0)}\mathbf{W}_{8\times7}\mathbf{P}_7\mathbf{X}_{7\times1}, \end{aligned} \tag{29}$$

where

$$\mathbf{D}_{10} = \operatorname{diag}\left(s_0^{(7)}, s_1^{(7)}, s_2^{(7)}, s_3^{(7)}, s_4^{(7)}, s_5^{(7)}, s_6^{(7)}, s_7^{(7)}, s_8^{(7)}, s_9^{(7)}\right);$$

$$s_4^{(7)} = d_7; \quad s_5^{(7)} = d_7;$$

$$\mathbf{W}_{8\times10} = \mathbf{T}_{3\times4} \oplus \mathbf{I}_2 \oplus \mathbf{T}_{3\times4}; \ \mathbf{W}_{10\times8} = \mathbf{T}_{4\times3} \oplus \mathbf{I}_2 \oplus \mathbf{T}_{4\times3};$$
$$\mathbf{W}_8^{(1)} = \mathbf{T}_3^{(1)} \oplus \mathbf{I}_2 \oplus \left(\mathbf{P}_3^{(a)}\mathbf{T}_3^{(1)}\right); \mathbf{W}_8^{(2)} = \mathbf{T}_3^{(0)} \oplus \mathbf{I}_2 \oplus \mathbf{T}_3^{(0)};$$

$$\mathbf{W}_8^{(0)} = \begin{bmatrix} 1 & & & & 1 & & & \\ & 1 & & & & 1 & & \\ & & 1 & & & & & 1 \\ & & & 1 & & & & \\ & & & & 1 & & & \\ 1 & & & & -1 & & & \\ & 1 & & & & -1 & & \\ & & 1 & & & & & -1 \end{bmatrix}$$

$$\mathbf{W}_{7\times8} = \begin{bmatrix} 1 & & & 1 & & & & \\ & 1 & & 1 & & & & \\ & & 1 & -1 & & & & \\ & & & -1 & 1 & & & \\ & & & 1 & & 1 & & \\ & & & 1 & & & 1 & \\ & & & -1 & & & & 1 \end{bmatrix}$$

$$\mathbf{W}_{8\times7} = \begin{bmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 1 & & & & \\ & & & 1 & & & \\ -1 & 1 & 1 & & 1 & -1 & -1 \\ & & & & 1 & & \\ & & & & & 1 & \\ & & & & & & 1 \end{bmatrix}$$

$$\mathbf{P}_8^{(a)} = \begin{bmatrix} 1 & & & & & & & \\ & & 1 & & & & & \\ & 1 & & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & -1 & & \\ & & & & & & & 1 \\ & & & & & & 1 & \end{bmatrix}$$

$$\mathbf{P}_7 = \begin{bmatrix} & & 1 & & & & \\ & 1 & & & & & \\ 1 & & & & & & \\ & & & 1 & & & \\ & & & & 1 & & \\ & & & & & -1 & \\ & & & & & & 1 \end{bmatrix}$$

Figure 6 shows a data flow graph of the synthesized algorithm for the seven-point DST-IV. As can be seen, we are able to reduce the number of multiplication operations from 49 to 10, although the number of addition operations is increased from 42 to 45.

2.8. Algorithm for 8-point DST-IV

Let's design the algorithm for eight-point DST-IV. The eight-point DST-IV is expressed as follows:

$$\mathbf{Y}_{8\times1} = \mathbf{C}_8\mathbf{X}_{8\times1}, \tag{30}$$

where:

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

$$\mathbf{C}_8 = \begin{bmatrix} a_8 & b_8 & c_8 & d_8 & e_8 & f_8 & g_8 & h_8 \\ b_8 & e_8 & h_8 & f_8 & c_8 & -a_8 & -d_8 & -g_8 \\ c_8 & h_8 & d_8 & -b_8 & -g_8 & -e_8 & a_8 & f_8 \\ d_8 & f_8 & -b_8 & -h_8 & -a_8 & g_8 & c_8 & -e_8 \\ e_8 & c_8 & -g_8 & -a_8 & h_8 & -b_8 & -f_8 & d_8 \\ f_8 & -a_8 & -e_8 & g_8 & -b_8 & -d_8 & h_8 & -c_8 \\ g_8 & -d_8 & a_8 & c_8 & -f_8 & h_8 & -e_8 & b_8 \\ h_8 & -g_8 & f_8 & -e_8 & d_8 & -c_8 & b_8 & -a_8 \end{bmatrix},$$

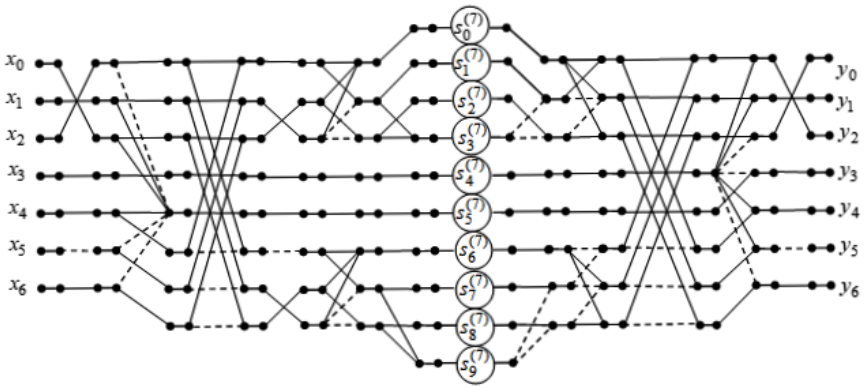

Fig. 6. The data flow graph of the proposed algorithm for the computation of seven-point DST-IV

with

$$a_8 = \sqrt{\frac{1}{4}}\sin\left(\frac{\pi}{32}\right) \approx 0.0490,\ b_8 = \sqrt{\frac{1}{4}}\sin\left(\frac{3\pi}{32}\right) \approx 0.1451,$$
$$c_8 = \sqrt{\frac{1}{4}}\sin\left(\frac{5\pi}{32}\right) \approx 0.2357,\ d_8 = \sqrt{\frac{1}{4}}\sin\left(\frac{7\pi}{32}\right) \approx 0.3172,$$
$$e_8 = \sqrt{\frac{1}{4}}\sin\left(\frac{9\pi}{32}\right) \approx 0.3865,\ f_8 = \sqrt{\frac{1}{4}}\sin\left(\frac{11\pi}{32}\right) \approx 0.4410,$$
$$g_8 = \sqrt{\frac{1}{4}}\sin\left(\frac{13\pi}{32}\right) \approx 0.4785,\ h_8 = \sqrt{\frac{1}{4}}\sin\left(\frac{15\pi}{32}\right) \approx 0.4976.$$

Let us define the permutation

$$\pi_8 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 4 & 2 & 3 & 1 & 5 & 7 & 6 & 8 \end{pmatrix}$$

to change the order of columns and rows. As a result of the permutations and altering the sign in second and third row and the second and third column, the matrix $\mathbf{C}_8^{(a)}$ is obtained. The matrix $\mathbf{C}_8^{(a)}$ matches the matrix pattern

$$\mathbf{C}_8^{(a)} = \begin{bmatrix} \mathbf{A}_4^{(a)} & \mathbf{B}_4^{(a)} \\ \mathbf{B}_4^{(a)} & -\mathbf{A}_4^{(a)} \end{bmatrix},$$

where

$$\mathbf{A}_4^{(a)} = \begin{bmatrix} -h_8 & f_8 & b_8 & -d_8 \\ f_8 & e_8 & -h_8 & -b_8 \\ b_8 & -h_8 & d_8 & c_8 \\ -d_8 & -b_8 & c_8 & a_8 \end{bmatrix},$$
$$\mathbf{B}_4^{(a)} = \begin{bmatrix} -a_8 & c_8 & g_8 & -e_8 \\ c_8 & -d_8 & -a_8 & -g_8 \\ g_8 & -a_8 & e_8 & -f_8 \\ -e_8 & -g_8 & -f_8 & -h_8 \end{bmatrix}.$$

Hence the matrix $\mathbf{C}_8^{(a)}$ can be represented as [5, 30]:

$$\mathbf{C}_8^{(a)} = (\mathbf{T}_{2\times3}^{(4)} \otimes \mathbf{I}_4) \cdot \left[(\mathbf{A}_4^{(a)} - \mathbf{B}_4^{(a)}) \oplus (-\mathbf{A}_4^{(a)} - \mathbf{B}_4^{(a)}) \oplus \mathbf{B}_4^{(a)}\right] (\mathbf{T}_{3\times2}^{(3)} \otimes \mathbf{I}_4) \quad (31)$$

where $\mathbf{T}_{2\times3}^{(4)}$, $\mathbf{T}_{3\times2}^{(3)}$ are defined as in (11).

Considering the structures of the resulting matrices $\mathbf{B}_4^{(a)}$,

$$\mathbf{A}_4^{(a)} - \mathbf{B}_4^{(a)} = \begin{bmatrix} -h_8+a_8 & f_8-c_8 & b_8-g_8 & -d_8+e_8 \\ f_8-c_8 & e_8+d_8 & -h_8+a_8 & -b_8+g_8 \\ b_8-g_8 & -h_8+a_8 & d_8-e_8 & f_8+c_8 \\ -d_8+e_8 & -b_8+g_8 & f_8+c_8 & h_8+a_a \end{bmatrix},$$
$$-\mathbf{A}_4^{(a)} - \mathbf{B}_4^{(a)} = \begin{bmatrix} h_8+a_8 & -f_8-c_8 & -b_8-g_8 & d_8+e_8 \\ -f_8-c_8 & -e_8+d_8 & h_8+a_8 & b_8+g_8 \\ -b_8-g_8 & h_8+a_8 & -d_8-e_8 & -f_8+c_8 \\ d_8+e_8 & b_8+g_8 & -f_8+c_8 & -h_8+a_a \end{bmatrix},$$

we note that these matrices matches the pattern

$$\begin{bmatrix} \mathbf{A}_2^{(b)} & \mathbf{B}_2^{(b)} \\ \mathbf{C}_2^{(b)} & \mathbf{A}_2^{(b)} \end{bmatrix}$$

after permutation of the columns and rows according to

$$\pi_9 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 2 & 1 \end{pmatrix}, \quad \pi_{10} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 4 & 3 \end{pmatrix},$$

respectively. Also a sign in the fourth row and third column was altered.

Then [5, 32]

$$\mathbf{A}_4^{(a)} - \mathbf{B}_4^{(a)} = (\mathbf{T}_{2\times3}^{(3)} \otimes \mathbf{I}_2) \cdot \left[(\mathbf{C}_2^{(b)} - \mathbf{A}_2^{(b)}) \oplus (\mathbf{B}_2^{(b)} - \mathbf{A}_2^{(b)}) \oplus \mathbf{A}_2^{(b)}\right] (\mathbf{T}_{3\times2}^{(3)} \otimes \mathbf{I}_2), \quad (32)$$

where $\mathbf{T}_{2\times3}^{(3)}$, $\mathbf{T}_{3\times2}^{(3)}$ are defined as in (10),

$$\mathbf{A}_2^{(b)} = \begin{bmatrix} b_8-g_8 & e_8-d_8 \\ a_8-h_8 & g_8-b_8 \end{bmatrix}, \quad \mathbf{B}_2^{(b)} = \begin{bmatrix} c_8-f_8 & a_8-h_8 \\ e_8-d_8 & f_8-c_8 \end{bmatrix},$$
$$\mathbf{C}_2^{(b)} = \begin{bmatrix} f_8+c_8 & a_8+h_8 \\ e_8-d_8 & -f_8-c_8 \end{bmatrix}.$$

The matrices $-\mathbf{A}_4^{(a)} - \mathbf{B}_4^{(a)}$ and $\mathbf{B}_4^{(a)}$ are decomposed simi-

larly to $\mathbf{A}_4^{(a)} - \mathbf{B}_4^{(a)}$. Then

$$\mathbf{A}_2^{(c)} = \begin{bmatrix} -b_8 - g_8 & e_8 + d_8 \\ a_8 + h_8 & g_8 + b_8 \end{bmatrix}, \mathbf{B}_2^{(c)} = \begin{bmatrix} c_8 + f_8 & a_8 + h_8 \\ -e_8 + d_8 & -f_8 - c_8 \end{bmatrix},$$
$$\mathbf{C}_2^{(c)} = \begin{bmatrix} f_8 - c_8 & -a_8 + h_8 \\ e_8 + d_8 & -f_8 + c_8 \end{bmatrix}.$$

and

$$\mathbf{A}_2^{(d)} = \begin{bmatrix} g_8 & -e_8 \\ -a_8 & -g_8 \end{bmatrix}, \quad \mathbf{B}_2^{(d)} = \begin{bmatrix} -c_8 & -a_8 \\ d_8 & c_8 \end{bmatrix},$$
$$\mathbf{C}_2^{(d)} = \begin{bmatrix} -f_8 & -h_8 \\ -e_8 & f_8 \end{bmatrix}$$

are used instead of $\mathbf{A}_2^{(b)}$, $\mathbf{B}_2^{(b)}$, $\mathbf{C}_2^{(b)}$.

Further, the matrices $\mathbf{C}_2^{(b)} - \mathbf{A}_2^{(b)}$, $\mathbf{B}_2^{(b)} - \mathbf{A}_2^{(b)}$, $\mathbf{A}_2^{(b)}$, $\mathbf{C}_2^{(c)} - \mathbf{A}_2^{(c)}$, $\mathbf{B}_2^{(c)} - \mathbf{A}_2^{(c)}$, $\mathbf{A}_2^{(c)}$, $\mathbf{C}_2^{(d)} - \mathbf{A}_2^{(d)}$, $\mathbf{B}_2^{(d)} - \mathbf{A}_2^{(d)}$, $\mathbf{A}_2^{(d)}$ matches the pattern $\left[\begin{smallmatrix} a & b \\ c & -a \end{smallmatrix}\right]$. So the matrix $\mathbf{A}_2^{(b)}$ is represented as

$$\mathbf{A}_2^{(b)} = \mathbf{T}_{2\times 3}^{(3)} \operatorname{diag}\left((c-a),(b+a)\right) \mathbf{T}_{3\times 2}^{(4)}, \tag{33}$$

wnere $\mathbf{T}_{2\times 3}^{(3)}$ is defined as in (10),

$$\mathbf{T}_{3\times 2}^{(4)} = \begin{bmatrix} 1 & \\ & 1 \\ -1 & 1 \end{bmatrix},$$

$a = b_8 - g_8$, $b = e_8 - d_8$, $c = a_8 - h_8$. The matrices $\mathbf{C}_2^{(b)} - \mathbf{A}_2^{(b)}$, $\mathbf{B}_2^{(b)} - \mathbf{A}_2^{(b)}$, $\mathbf{C}_2^{(c)} - \mathbf{A}_2^{(c)}$, $\mathbf{B}_2^{(c)} - \mathbf{A}_2^{(c)}$, $\mathbf{A}_2^{(c)}$, $\mathbf{C}_2^{(d)} - \mathbf{A}_2^{(d)}$, $\mathbf{B}_2^{(d)} - \mathbf{A}_2^{(d)}$, $\mathbf{A}_2^{(d)}$ are decomposed similarly to $\mathbf{A}_2^{(b)}$.

Combining the decompositions (31), (32), (33), we obtain the following matrix factorization:

$$\begin{aligned} \mathbf{Y}_{8\times 1} =& \mathbf{P}_8^{(1)} \mathbf{W}_{8\times 12} \mathbf{P}_{12}^{(1)} \mathbf{W}_{12\times 18} \mathbf{W}_{18\times 27} \mathbf{D}_{27} \\ & \cdot \mathbf{W}_{27\times 18} \mathbf{W}_{18\times 12} \mathbf{P}_{12}^{(0)} \mathbf{W}_{12\times 8} \mathbf{P}_8^{(0)} \mathbf{X}_{8\times 1}, \end{aligned} \tag{34}$$

where

$$\mathbf{D}_{27} = \operatorname{diag}\left(s_0^{(8)}, s_0^{(8)}, \ldots, s_{26}^{(8)}\right);$$

$$s_0^{(8)} = e_8 - d_8 + h_8 - a_8 - f_8 - c_8 + b_8 - g_8;$$
$$s_1^{(8)} = f_8 + c_8 - b_8 + g_8 + h_8 + a_8 + d_8 - e_8;$$
$$s_2^{(8)} = -f_8 - c_8 + b_8 - g_8;$$
$$s_3^{(8)} = -e_8 - d_8 + h_8 - a_8 - c_8 + f_8 + b_8 - g_8;$$
$$s_4^{(8)} = c_8 - f_8 - b_8 + g_8 - h_8 + a_8 + d_8 - e_8;$$
$$s_5^{(8)} = -c_8 + f_8 + b_8 - g_8; \quad s_6^{(8)} = -h_8 + a_8 - b_8 + g_8;$$
$$s_7^{(8)} = b_8 - g_8 - d_8 + e_8; \quad s_8^{(8)} = -b_8 + g_8;$$
$$s_9^{(8)} = d_8 + e_8 - h_8 - a_8 + c_8 - f_8 - b_8 - g_8;$$
$$s_{10}^{(8)} = -a_8 + h_8 - d_8 - e_8 - c_8 + f_8 + b_8 + g_8;$$
$$s_{11}^{(8)} = c_8 - f_8 - b_8 - g_8;$$
$$s_{12}^{(8)} = e_8 - d_8 - h_8 - a_8 - c_8 - f_8 - b_8 - g_8;$$
$$s_{13}^{(8)} = a_8 + h_8 - d_8 - e_8 + c_8 + f_8 + b_8 + g_8;$$
$$s_{14}^{(8)} = -c_8 - f_8 - b_8 - g_8; \quad s_{15}^{(8)} = h_8 + a_8 + b_8 + g_8;$$
$$s_{16}^{(8)} = -b_8 - g_8 + d_8 + e_8; \quad s_{17}^{(8)} = b_8 + g_8;$$
$$s_{18}^{(8)} = -e_8 + a_8 + f_8 + g_8; \quad s_{19}^{(8)} = -f_8 - g_8 - h_8 + e_8;$$
$$s_{20}^{(8)} = f_8 + g_8; \quad s_{21}^{(8)} = d_8 + a_8 + c_8 + g_8;$$
$$s_{22}^{(8)} = -g_8 - c_8 + e_8 - a_8; \quad s_{23}^{(8)} = c_8 + g_8;$$
$$s_{24}^{(8)} = -a_8 - g_8; \quad s_{25}^{(8)} = g_8 - e_8; \quad s_{26}^{(8)} = -g_8;$$

$$\mathbf{W}_{8\times 12} = \mathbf{T}_{2\times 3}^{(4)} \otimes \mathbf{I}_4; \quad \mathbf{W}_{12\times 8} = \mathbf{T}_{3\times 2}^{(3)} \otimes \mathbf{I}_4;$$
$$\mathbf{W}_{12\times 18} = \left(\mathbf{T}_{2\times 3}^{(3)} \otimes \mathbf{I}_2\right) \oplus \left(\mathbf{T}_{2\times 3}^{(3)} \otimes \mathbf{I}_2\right) \oplus \left(\mathbf{T}_{2\times 3}^{(3)} \otimes \mathbf{I}_2\right);$$
$$\mathbf{W}_{18\times 12} = \left(\mathbf{T}_{3\times 2}^{(3)} \otimes \mathbf{I}_2\right) \oplus \left(\mathbf{T}_{3\times 2}^{(3)} \otimes \mathbf{I}_2\right) \oplus \left(\mathbf{T}_{3\times 2}^{(3)} \otimes \mathbf{I}_2\right);$$
$$\mathbf{P}_{12}^{(0)} = \mathbf{P}_4^{(a)} \oplus \mathbf{P}_4^{(a)} \oplus \mathbf{P}_4^{(a)};$$
$$\mathbf{P}_{12}^{(1)} = \mathbf{P}_4^{(b)} \oplus \mathbf{P}_4^{(b)} \oplus \mathbf{P}_4^{(b)};$$

$$\begin{aligned} \mathbf{W}_{27\times 18} =& \mathbf{T}_{3\times 2}^{(4)} \oplus \mathbf{T}_{3\times 2}^{(4)} \oplus \mathbf{T}_{3\times 2}^{(4)} \oplus \mathbf{T}_{3\times 2}^{(4)} \oplus \mathbf{T}_{3\times 2}^{(4)} \oplus \\ & \oplus \mathbf{T}_{3\times 2}^{(4)} \oplus \mathbf{T}_{3\times 2}^{(4)} \oplus \mathbf{T}_{3\times 2}^{(4)} \oplus \mathbf{T}_{3\times 2}^{(4)}; \\ \mathbf{W}_{18\times 27} =& \mathbf{T}_{2\times 3}^{(3)} \oplus \mathbf{T}_{2\times 3}^{(3)} \oplus \mathbf{T}_{2\times 3}^{(3)} \oplus \mathbf{T}_{2\times 3}^{(3)} \oplus \mathbf{T}_{2\times 3}^{(3)} \oplus \\ & \oplus \mathbf{T}_{2\times 3}^{(3)} \oplus \mathbf{T}_{2\times 3}^{(3)} \oplus \mathbf{T}_{2\times 3}^{(3)} \oplus \mathbf{T}_{2\times 3}^{(3)}; \end{aligned}$$

$$\mathbf{P}_4^{(a)} = \begin{bmatrix} & 1 & \\ & & 1 \\ -1 & & \\ 1 & & \end{bmatrix}; \mathbf{P}_4^{(b)} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & & -1 \\ & & 1 & \end{bmatrix};$$

$$\mathbf{P}_8^{(0)} = \begin{bmatrix} & & 1 & & & & & \\ & 1 & & & & & & \\ & & -1 & & & & & \\ -1 & & & & & & & \\ & & & & 1 & & & \\ & & & & & & 1 & \\ & & & & & 1 & & \\ & & & & & & & 1 \end{bmatrix};$$

$$\mathbf{P}_8^{(1)} = \begin{bmatrix} & & & -1 & & & & \\ & 1 & & & & & & \\ & & -1 & & & & & \\ 1 & & & & & & & \\ & & & & 1 & & & \\ & & & & & & 1 & \\ & & & & & 1 & & \\ & & & & & & & 1 \end{bmatrix}.$$

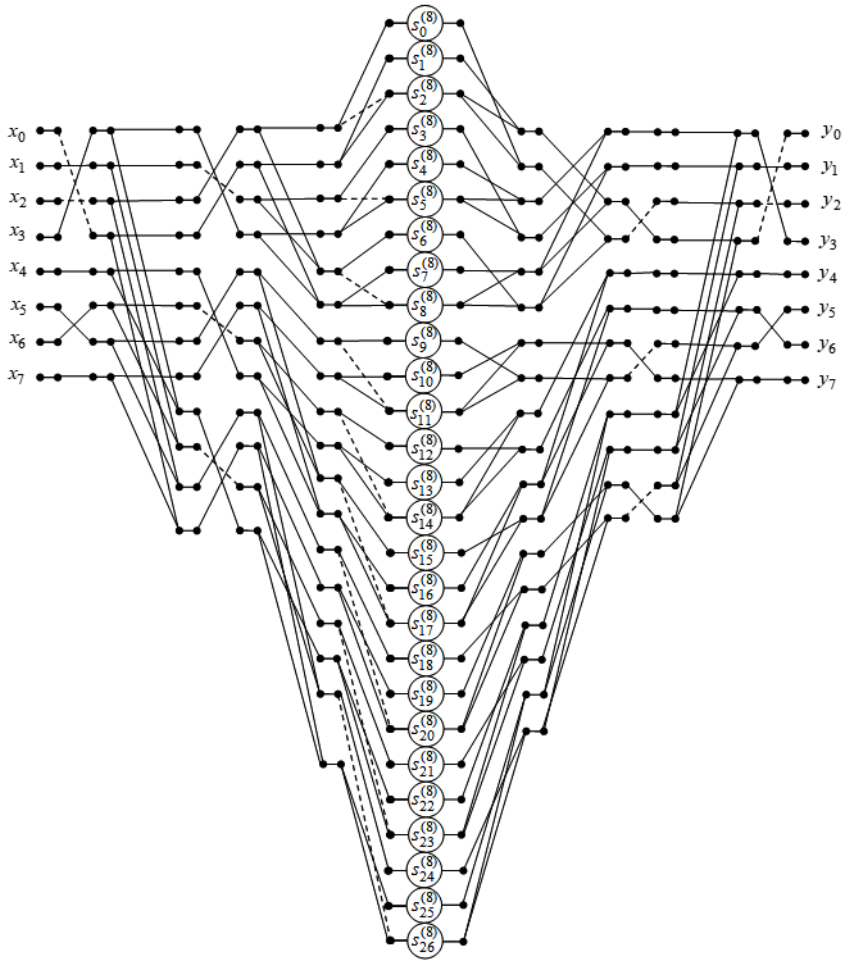

Fig. 7. The data flow graph of the proposed algorithm for the computation of eight-point DST-IV

The data flow graph of the designed algorithm for the eight-point DST-IV is presented on Figure 7. As can be seen, we are able to reduce the number of multiplication operations from 81 to 27, although the number of addition operations is increased from 56 to 57.

2.9. Algorithm for 9-point DST-IV

To develop the algorithm for nine-point DST-IV the formula of this transform is expressed as follows:

$$\mathbf{Y}_{9\times 1} = \mathbf{C}_9\mathbf{X}_{9\times 1}, \tag{35}$$

where:

$$\mathbf{C}_9 = \begin{bmatrix} a_9 & b_9 & c_9 & d_9 & e_9 & f_9 & g_9 & h_9 & q_9 \\ b_9 & e_9 & h_9 & h_9 & e_9 & b_9 & -b_9 & -e_9 & -h_9 \\ c_9 & h_9 & f_9 & a_9 & -e_9 & -q_9 & -d_9 & b_9 & g_9 \\ d_9 & h_9 & a_9 & -g_9 & -e_9 & c_9 & q_9 & b_9 & -f_9 \\ e_9 & e_9 & -e_9 & -e_9 & e_9 & e_9 & -e_9 & -e_9 & e_9 \\ f_9 & b_9 & -q_9 & c_9 & e_9 & -g_9 & -a_9 & h_9 & -d_9 \\ g_9 & -b_9 & -d_9 & q_9 & -e_9 & -a_9 & f_9 & -h_9 & c_9 \\ h_9 & -e_9 & b_9 & b_9 & -e_9 & h_9 & -h_9 & e_9 & -b_9 \\ q_9 & -h_9 & g_9 & -f_9 & e_9 & -d_9 & c_9 & -b_9 & a_9 \end{bmatrix},$$

$$\mathbf{X}_{9\times 1} = [x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8]^T,$$

$$\mathbf{Y}_{9\times 1} = [y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8]^T,$$

$$a_9 = \sqrt{\frac{2}{9}}\sin\left(\frac{\pi}{36}\right) \approx 0.0411,\ b_9 = \sqrt{\frac{2}{9}}\sin\left(\frac{\pi}{12}\right) \approx 0.1220,$$

$$c_9 = \sqrt{\frac{2}{9}}\sin\left(\frac{5\pi}{36}\right) \approx 0.1992,\ d_9 = \sqrt{\frac{2}{9}}\sin\left(\frac{7\pi}{36}\right) \approx 0.2704,$$

$$e_9 = \sqrt{\frac{2}{9}}\sin\left(\frac{\pi}{4}\right) \approx 0.3333,\ f_9 = \sqrt{\frac{2}{9}}\sin\left(\frac{11\pi}{36}\right) \approx 0.3862,$$

$$g_9 = \sqrt{\frac{2}{9}}\sin\left(\frac{13\pi}{36}\right) \approx 0.4272,\ h_9 = \sqrt{\frac{2}{9}}\sin\left(\frac{5\pi}{12}\right) \approx 0.4553,$$

$$q_9 = \sqrt{\frac{2}{9}}\sin\left(\frac{17\pi}{36}\right) \approx 0.4696.$$

To change the order of columns and rows, we define the permutation π_{12} in the following form

$$\pi_{12} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 2 & 3 & 4 & 5 & 9 & 8 & 7 & 6 \end{pmatrix}.$$

Let permute columns and rows of $\mathbf{C}_9$ according to π_{12}. After permutation and altering the sign in second and ninth rows, seventh and ninth columns the obtained matrix $\mathbf{C}_9^{(a)}$ is decom-

posed into two components:

$$\mathbf{C}_9^{(a)} = \mathbf{C}_9^{(b)} + \mathbf{C}_9^{(c)}, \tag{36}$$

where

$$\mathbf{C}_9^{(b)} = \begin{bmatrix} & & & & e_9 & & & & \\ & & & & -e_9 & & & & \\ & & & & -e_9 & & & & \\ & & & & -e_9 & & & & \\ e_9 & e_9 & -e_9 & -e_9 & e_9 & e_9 & e_9 & -e_9 & -e_9 \\ & & & & e_9 & & & & \\ & & & & -e_9 & & & & \\ & & & & -e_9 & & & & \\ & & & & -e_9 & & & & \end{bmatrix},$$

$$\mathbf{C}_9^{(c)} = \begin{bmatrix} a_9 & b_9 & c_9 & d_9 & q_9 & -h_9 & g_9 & -f_9 \\ -b_9 & -e_9 & -h_9 & -h_9 & h_9 & -e_9 & b_9 & b_9 \\ c_9 & h_9 & f_9 & a_9 & g_9 & -b_9 & -d_9 & q_9 \\ d_9 & h_9 & a_9 & -g_9 & -f_9 & -b_9 & q_9 & -c_9 \\ q_9 & -h_9 & g_9 & -f_9 & a_9 & b_9 & c_9 & d_9 \\ h_9 & -e_9 & b_9 & b_9 & -b_9 & -e_9 & -h_9 & -h_9 \\ g_9 & -b_9 & -d_9 & q_9 & h_9 & -h_9 & e_9 & -b_9 \\ -f_9 & -b_9 & q_9 & -c_9 & -d_9 & c_9 & -b_9 & a_9 \end{bmatrix}.$$

Matrix $\mathbf{C}_9^{(b)}$ has the same entries except on the sign in the fifth column and fifth row, which allows us to reduce the number of operations without the need for further transformations. After eliminating the rows and columns containing only zero entries in matrix $\mathbf{C}_9^{(c)}$, we obtain matrix $\mathbf{C}_8^{(d)}$. The obtained matrix acquires the structure

$$\mathbf{C}_8^{(d)} = \begin{bmatrix} \mathbf{A}_4^{(b)} & \mathbf{B}_4^{(b)} \\ \mathbf{B}_4^{(b)} & \mathbf{A}_4^{(b)} \end{bmatrix},$$

$$\mathbf{A}_4^{(b)} = \begin{bmatrix} a_9 & b_9 & c_9 & d_9 \\ -b_9 & -e_9 & -h_9 & -h_9 \\ c_9 & h_9 & f_9 & a_9 \\ d_9 & h_9 & a_9 & -g_9 \end{bmatrix},$$

$$\mathbf{B}_4^{(b)} = \begin{bmatrix} q_9 & -h_9 & g_9 & -f_9 \\ h_9 & -e_9 & b_9 & b_9 \\ g_9 & -b_9 & -d_9 & q_9 \\ -f_9 & -b_9 & q_9 & -c_9 \end{bmatrix}.$$

Then [5, 32]

$$\mathbf{C}_8^{(d)} = (\mathbf{H}_2 \otimes \mathbf{I}_3) \cdot \frac{1}{2}\left[(\mathbf{A}_4^{(b)} + \mathbf{B}_4^{(b)}) \oplus (\mathbf{A}_4^{(b)} - \mathbf{B}_4^{(b)})\right](\mathbf{H}_2 \otimes \mathbf{I}_3). \tag{37}$$

Let's represent the submatrices

$$\mathbf{A}_4^{(b)} + \mathbf{B}_4^{(b)} = \begin{bmatrix} a_9+q_9 & b_9-h_9 & c_9+g_9 & d_9-f_9 \\ -b_9+h_9 & -2e_9 & -h_9+b_9 & -h_9+b_9 \\ c_9+g_9 & h_9-b_9 & f_9-d_9 & a_9+q_9 \\ d_9-f_9 & h_9-b_9 & a_9+q_9 & -g_9-c_9 \end{bmatrix},$$

$$\mathbf{A}_4^{(b)} - \mathbf{B}_4^{(b)} = \begin{bmatrix} a_9-q_9 & b_9+h_9 & c_9-g_9 & d_9+f_9 \\ -b_9-h_9 & 0 & -h_9-b_9 & -h_9-b_9 \\ c_9-g_9 & h_9+b_9 & f_9+d_9 & a_9-q_9 \\ d_9+f_9 & h_9+b_9 & a_9-q_9 & -g_9+c_9 \end{bmatrix},$$

of quasi-diagonal matrix $(\mathbf{A}_4^{(b)} + \mathbf{B}_4^{(b)}) \oplus (\mathbf{A}_4^{(b)} - \mathbf{B}_4^{(b)})$ as circular convolution matrices [32]. We permutated the columns and rows in both matrices with

$$\pi_{13} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix}, \quad \pi_{14} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 3 & 4 \end{pmatrix},$$

respectively. Then the sign in second row and second column of the transformed $\mathbf{A}_4^{(b)} + \mathbf{B}_4^{(b)}$ matrix was altered. As a result the matrices $\mathbf{C}_4^{(k)}$ and $\mathbf{C}_4^{(l)}$ were respectively obtained:

$$\mathbf{C}_4^{(k)} = \begin{bmatrix} -2e_9 & -b_9+h_9 & b_9-h_9 & b_9-h_9 \\ b_9-h_9 & a_9+q_9 & -d_9+f_9 & -c_9-g_9 \\ -b_9+h_9 & -c_9-g_9 & a_9+q_9 & -d_9+f_9 \\ -b_9+h_9 & -d_9+f_9 & -c_9-g_9 & a_9+q_9 \end{bmatrix},$$

$$\mathbf{C}_4^{(l)} = \begin{bmatrix} 0 & -b_9-h_9 & -b_9-h_9 & -b_9-h_9 \\ h_9+b_9 & a_9-q_9 & d_9+f_9 & c_9-g_9 \\ h_9+b_9 & c_9-g_9 & a_9-q_9 & d_9+f_9 \\ h_9+b_9 & d_9+f_9 & c_9-g_9 & a_9-q_9 \end{bmatrix}.$$

Then the matrices $\mathbf{C}_4^{(k)}$, $\mathbf{C}_4^{(l)}$ are decomposed into two components:

$$\mathbf{C}_4^{(k)} = \mathbf{C}_4^{(m)} + \mathbf{C}_4^{(n)}; \quad \mathbf{C}_4^{(l)} = \mathbf{C}_4^{(p)} + \mathbf{C}_4^{(r)}; \tag{38}$$

where

$$\mathbf{C}_4^{(m)} = \begin{bmatrix} -2e_9 & -b_9+h_9 & b_9-h_9 & b_9-h_9 \\ b_9-h_9 & & & \\ -b_9+h_9 & & & \\ -b_9+h_9 & & & \end{bmatrix},$$

$$\mathbf{C}_4^{(n)} = \begin{bmatrix} & a_9+q_9 & -d_9+f_9 & -c_9-g_9 \\ & -c_9-g_9 & a_9+q_9 & -d_9+f_9 \\ & -d_9+f_9 & -c_9-g_9 & a_9+q_9 \end{bmatrix},$$

$$\mathbf{C}_4^{(p)} = \begin{bmatrix} & -b_9-h_9 & -b_9-h_9 & -b_9-h_9 \\ h_9+b_9 & & & \\ h_9+b_9 & & & \\ h_9+b_9 & & & \end{bmatrix};$$

$$\mathbf{C}_4^{(r)} = \begin{bmatrix} & a_9-q_9 & d_9+f_9 & c_9-g_9 \\ & c_9-g_9 & a_9-q_9 & d_9+f_9 \\ & d_9+f_9 & c_9-g_9 & a_9-q_9 \end{bmatrix}.$$

Matrices $\mathbf{C}_4^{(m)}$, $\mathbf{C}_4^{(p)}$ have the same entries except on the sign in the first column and first row, and except on element in first row and first column. This allows reducing the number of operations without the need for further transformations. After

eliminating the rows and columns containing only zero entries in matrices $\mathbf{C}_4^{(n)}$, $\mathbf{C}_4^{(r)}$, we obtain matrices $\mathbf{C}_3^{(n)}$, $\mathbf{C}_3^{(r)}$:

$$\mathbf{C}_3^{(n)} = \begin{bmatrix} a_9+q_9 & -d_9+f_9 & -c_9-g_9 \\ -c_9-g_9 & a_9+q_9 & -d_9+f_9 \\ -d_9+f_9 & -c_9-g_9 & a_9+q_9 \end{bmatrix};$$

$$\mathbf{C}_3^{(r)} = \begin{bmatrix} a_9-q_9 & d_9+f_9 & c_9-g_9 \\ c_9-g_9 & a_9-q_9 & d_9+f_9 \\ d_9+f_9 & c_9-g_9 & a_9-q_9 \end{bmatrix}.$$

Further the matrices $\mathbf{C}_3^{(n)}$, $\mathbf{C}_3^{(r)}$ were factorized in accordance with the expressions for calculating the entries of a circular convolution matrix $\mathbf{H}_3$

$$\mathbf{H}_3 = \begin{bmatrix} h_0 & h_2 & h_1 \\ h_1 & h_0 & h_2 \\ h_2 & h_1 & h_0 \end{bmatrix}$$

for $N = 3$ (expression (27)) [32].

As a result we obtain in (27) $h_0 = a_9 + q_9$, $h_1 = -c_9 - g_9$, $h_2 = -d_9 + f_9$ for $\mathbf{C}_3^{(n)}$ matrix and $h_0 = a_9 - q_9$, $h_1 = c_9 - g_9$, $h_2 = d_9 + f_9$ for $\mathbf{C}_3^{(r)}$ matrix.

Next, we define

$$s_0^{(9)} = -e_9; \quad s_1^{(9)} = \frac{a_9+q_9-d_9+f_9-c_9-g_9}{6};$$

$$s_2^{(9)} = \frac{a_9+q_9+d_9-f_9}{2}; \quad s_3^{(9)} = \frac{-c_9-g_9+d_9-f_9}{2};$$

$$s_4^{(9)} = \frac{a_9+q_9-c_9-g_9+2d_9-2f_9}{6}; \quad s_5^{(9)} = \frac{b_9-h_9}{2};$$

$$s_6^{(9)} = \frac{b_9-h_9}{2}; \quad s_7^{(9)} = e_9; \quad s_8^{(9)} = e_9;$$

$$s_9^{(9)} = \frac{a_9-q_9+d_9+f_9+c_9-g_9}{6};$$

$$s_{10}^{(9)} = \frac{a_9-q_9-d_9-f_9}{2}; \quad s_{11}^{(9)} = \frac{c_9-g_9-d_9-f_9}{2};$$

$$s_{12}^{(9)} = \frac{a_9-q_9+c_9-g_9-2d_9-2f_9}{6}$$

$$s_{13}^{(9)} = \frac{b_9+h_9}{2}; \quad s_{14}^{(9)} = \frac{b_9+h_9}{2}.$$

Based on properties of structural matrices [5, 32], the computational procedure for the nine-point DST-IV is represented by the expression

$$\begin{aligned}\mathbf{Y}_{9\times1} =& \mathbf{P}_9^{(1)}\mathbf{W}_{9\times10}\mathbf{P}_{10}^{(1)}\mathbf{W}_{10\times13}\mathbf{W}_{13}^{(1)}\mathbf{W}_{13\times15}\mathbf{D}_{15}\mathbf{W}_{15\times13} \\ &\cdot \mathbf{W}_{13}^{(0)}\mathbf{W}_{13\times17}\mathbf{W}_{17\times10}\mathbf{P}_{10}^{(0)}\mathbf{W}_{10}\mathbf{W}_{10\times9}\mathbf{P}_9^{(0)}\mathbf{X}_{9\times1};\end{aligned} \tag{39}$$

where

$$\mathbf{D}_{15} = \operatorname{diag}\left(s_0^{(9)}, s_0^{(9)}, \ldots, s_{14}^{(9)}\right);$$

$$\mathbf{W}_{13\times15} = 1 \oplus \mathbf{T}_{3\times4} \oplus \mathbf{I}_4 \oplus \mathbf{T}_{3\times4} \oplus \mathbf{I}_2;$$

$$\mathbf{W}_{15\times13} = 1 \oplus \mathbf{T}_{4\times3} \oplus \mathbf{I}_4 \oplus \mathbf{T}_{4\times3} \oplus \mathbf{I}_2;$$

$$\mathbf{W}_{13}^{(1)} = 1 \oplus \mathbf{T}_3^{(1)} \oplus \mathbf{I}_4 \oplus \mathbf{T}_3^{(1)} \oplus \mathbf{I}_2;$$

$$\mathbf{W}_{13}^{(0)} = 1 \oplus \mathbf{T}_3^{(0)} \oplus \mathbf{I}_4 \oplus \mathbf{T}_3^{(0)} \oplus \mathbf{I}_2;$$

$$\mathbf{W}_{13\times17} = \mathbf{I}_4 \oplus \mathbf{P}_{2\times4}^{(a)} \oplus \mathbf{I}_5 \oplus \mathbf{P}_{2\times4}^{(b)};$$

$$\mathbf{W}_{10\times13} = \mathbf{P}_{4\times6}^{(b)} \oplus \mathbf{I}_2 \oplus \mathbf{P}_{4\times5}^{(a)};$$

$$\mathbf{W}_{17\times10} = (\mathbf{1}_{2\times1} \otimes \mathbf{I}_4) \oplus \mathbf{I}_2 \oplus \mathbf{P}_{7\times4};$$

$$\mathbf{P}_{2\times4}^{(a)} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & & \end{bmatrix}; \quad \mathbf{P}_{2\times4}^{(b)} = \begin{bmatrix} -1 & -1 & -1 \\ 1 & & \end{bmatrix};$$

$$\mathbf{P}_{4\times5}^{(a)} = \begin{bmatrix} & & 1 & \\ 1 & & & 1 \\ & 1 & & 1 \\ & & 1 & 1 \end{bmatrix};$$

$$\mathbf{P}_{4\times6}^{(b)} = \begin{bmatrix} 1 & & & 1 & & \\ & 1 & & & & -1 \\ & & 1 & & & -1 \\ & & & 1 & & -1 \end{bmatrix};$$

$$\mathbf{P}_{7\times4} = \begin{bmatrix} & 1 & & \\ & & 1 & \\ & & & 1 \\ 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix};$$

$$\mathbf{P}_{10}^{(0)} = \begin{bmatrix} & 1 & & & & & & & & \\ -1 & & & & & & & & & \\ & & 1 & & & & & & & \\ & & & 1 & & & & & & \\ & & & & 1 & & & & & \\ & & & & & 1 & & & & \\ & & & & & & 1 & & & \\ & & & & & & & 1 & & \\ & & & & & & & & 1 & \end{bmatrix};$$

$$\mathbf{P}_{10}^{(1)} = \begin{bmatrix} -1 & & & & & & & & & \\ 1 & & & & & & & & & \\ & & 1 & & & & & & & \\ & & & 1 & & & & & & \\ & & & & 1 & & & & & \\ & & & & & 1 & & & & \\ & & & & & & 1 & & & \\ & & & & & & & 1 & & \\ & & & & & & & & 1 & \end{bmatrix};$$

$$\mathbf{P}_9^{(0)} = \begin{bmatrix} 1 & & & & & & & & \\ & 1 & & & & & & & \\ & & 1 & & & & & & \\ & & & 1 & & & & & \\ & & & & 1 & & & & \\ & & & & & & & & 1 \\ & & & & & & & -1 & \\ & & & & & & 1 & & \\ & & & & & -1 & & & \end{bmatrix};$$

$$\mathbf{P}_9^{(1)} = \begin{bmatrix} 1 & & & & & & & & \\ & -1 & & & & & & & \\ & & 1 & & & & & & \\ & & & 1 & & & & & \\ & & & & 1 & & & & \\ & & & & & & & -1 & \\ & & & & & & 1 & & \\ & & & & & 1 & & & \\ & & & & & & & & \end{bmatrix};$$

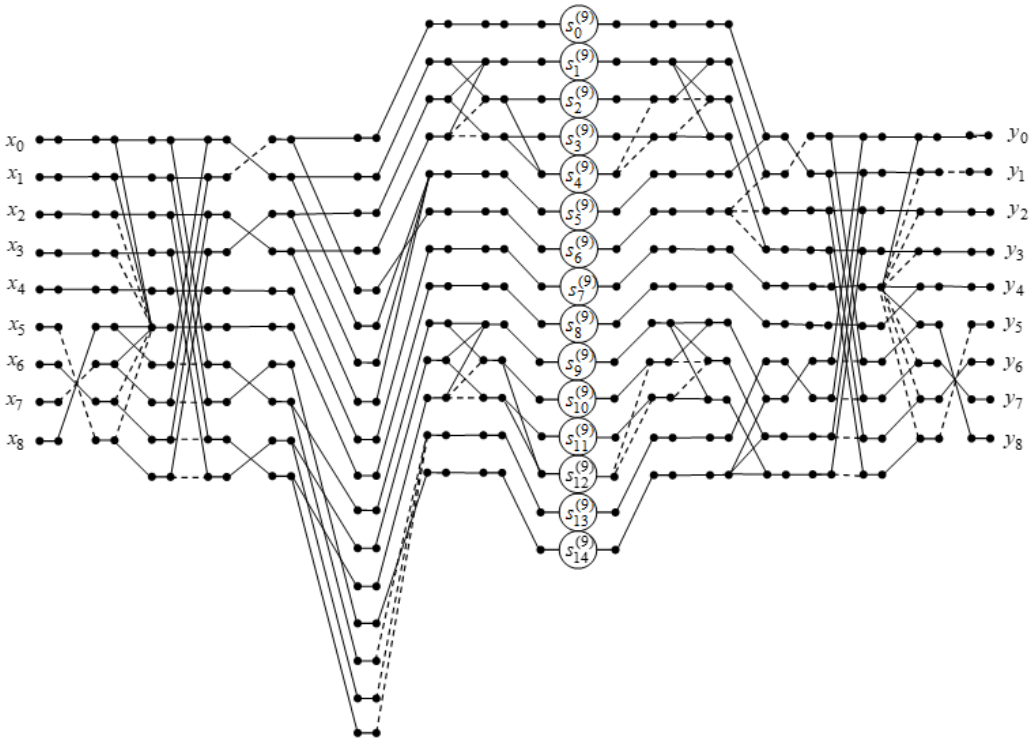

Fig. 8. The data flow graph of the proposed algorithm for the computation of nine-point DST-IV

$$\mathbf{W}_{9\times 10} = \begin{bmatrix} 1 & & & & 1 & & & & \\ & 1 & & & -1 & & & & \\ & & 1 & & -1 & & & & \\ & & & 1 & -1 & & & & \\ & & & & 1 & 1 & & & \\ & & & & 1 & & 1 & & \\ & & & & -1 & & & 1 & \\ & & & & -1 & & & & 1 \\ & & & & -1 & & & & \end{bmatrix};$$

$$\mathbf{W}_{10\times 9} = \begin{bmatrix} 1 & & & & & & & & \\ & 1 & & & & & & & \\ & & 1 & & & & & & \\ & & & 1 & & & & & \\ & & & & 1 & & & & \\ 1 & 1 & -1 & -1 & & 1 & 1 & -1 & -1 \\ & & & & & 1 & & 1 & \\ & & & & & & 1 & & \\ & & & & & & & 1 & \\ & & & & & & & & 1 \end{bmatrix};$$

$$\mathbf{W}_{10} = \begin{bmatrix} 1 & & & & & 1 & & & & \\ & 1 & & & & & 1 & & & \\ & & 1 & & & & & 1 & & \\ & & & 1 & & & & & & 1 \\ & & & & 1 & & & & & \\ 1 & & & & & -1 & & & & \\ & 1 & & & & & -1 & & & \\ & & 1 & & & & & -1 & & \\ & & & 1 & & & & & -1 & \end{bmatrix}.$$

Figure 8 shows a data flow graph of the synthesized algorithm for the nine-point DST-IV.

3. RESULTS

The correctness of the developed algorithms was tested using the MATLAB R2023b environment. For this purpose, the matrices of coefficients of the DST-IV were calculated using expression (1) for $N = 2,3,4,5,6,7,8,9$. Then factorizations of DST-IV matrices were calculated using the expressions (5), (8), (12), (18), (23), (29), (34), (39). The coincidence of the results for the same N indicated the correctness of the developed algorithms. And, similarly, it was tested the correctness of the data flow graphs for the proposed algorithms. The 30 sequences of random normally distributed numbers were generated for each considered value of length N. Each sequence was transformed according to the data flow graph and also using the direct matrix-vector product. The coincidence of the results obtained for each N indicated the correctness of the synthesized data flow graphs for the developed algorithms.

Further, the computational complexity of the proposed algorithms was evaluated. The obtained results are shown in Table 1. In parentheses, the percentage difference in the number of operations is indicated, plus means that the number of operations has risen compared to the direct method, and minus means that it has reduced. Thus, the number of multiplications for $N = 3,4,5,6,7,8,9$ was reduced by an average of 63%. At the same time, the number of additions was increased by an average of 8%.

Table 1. The number of additions and multiplications of the direct method against the proposed algorithms

N	Direct method		Proposed algorithms	
	Adds.	Mults.	Adds.	Mults.
2	2	4	3 (+50%)	3 (-25%)
3	6	9	7 (+17%)	4 (-55%)
4	12	16	15 (+25%)	9 (-44%)
5	20	25	23 (+15%)	7 (-60%)
6	30	36	30 (0%)	12 (-64%)
7	42	49	45 (+7%)	10 (-78%)
8	56	64	57 (+2%)	27 (-58%)
9	72	81	65 (-10%)	15 (-81%)

4. DISCUSSION OF COMPUTATIONAL COMPLEXITY

The number of multiplications and additions for the algorithms known from the literature is shown in Table 2. Analyzing the obtained results, we note that the number of multiplications was reduced relative to a completely recursive algorithm [20] for $N = 4$ and $N = 8$. But at the same time, the number of additions was increased. The completely recursive algorithm was developed exploiting the same approach as the proposed algorithms, specifically, the matrix factorization approach. The algorithms designed with the polynomial arithmetic approach are implemented with fewer multiplications than the proposed algorithms. However, for $N = 9$ the proposed algorithm reduced the number of multiplications by 25% compared to the general radix algorithm developed with the polynomial arithmetic approach [22]. At the same time the number of additions for $N = 9$ is increased by 41%. The reduction in multiplications is significantly contributed to speeding up the signal processing since the multiplications are more expensive to use than additions. As a result, the amount of resources used in the signal processor is significantly reduced allowing for easier operation in real time.

The proposed DST-IV fast algorithms do not limit the length of the input data sequence to powers of two or three. The data flow graphs constructed for the proposed algorithms reveal their modular space-time structure suitable for VLSI implementation.

5. THE EXAMPLE OF THE PROPOSED ALGORITHM APPLYING FOR THE SPEECH SIGNAL DENOISING

The orthogonal transforms including DST are widely used in speech denoising [16–18, 32, 33]. This practical example can be explored for understanding the advantages of the proposed algorithms. We base on the simple but effective scheme of the speech signal denoising inscribed in [32] but use the DST-IV instead of the DCT-II as in the original source. Then the stages of this technique are as follows.

1. We assume that the speech signal contains additive white Gaussian noise with zero mean and a previously known or accurately estimated variance σ^2 .
2. The initial speech signal is windowed on non-overlapping or partly-overlapping frames of N pixels.
3. The orthogonal transform is performed for each of these frames; in particular, we apply the proposed fast DST-IV algorithms depending on the N. These algorithms were implemented using the simulated model based on the data flow graphs. After the DST-IV the first coefficient in each frame is related to the mean of the signal frame and further has not been processed.
4. The resulting signal is thresholded using the following expression [32]:

$$D_{\text{thr}}(k) = \begin{cases} D(k), & \text{if } |D(k)| > T; \\ D^3(k)/T^2 & \text{if } |D(k)| \le T; \end{cases}$$

   where k is a number of DST-IV coefficient in the frame, $T = \beta\sigma$ is a threshold value, β is a coefficient which is selected by tuning approximately in the range from 2 to 5.
5. After the described thresholding the inverse DST-IV is applied. The fast algorithms for this transform can be easily obtained keeping in mind that the inverse transform matrix is the transposed matrix of the direct DST-IV.

The accuracy of the DCT-based filtering for speech signal denoising problem was evaluated by the improvement in the signal-to-noise ratio (SNR) in dB [32]:

$$\text{ISNR} = 10\log_{10}\frac{\sigma^2}{\text{MSE}} = \text{SNR}_{\text{out}} - \text{SNR}_{\text{in}}$$

where ISNR is the improvement in the SNR, an MSE is a mean squared error between the initial signal and the denoised signal, SNR_{in} is the SNR value before signal denoising, SNR_{out} is the SNR value after signal denoising.

To test the performance of the DST-based filtering, in this paper the so-called Harvard phrases are used. These are the recordings of English male voice utters which are taken from a set of speech signals formed at McGill University in Montreal, Canada [34]. The above dataset is often applied to research speech signal processing techniques [32]. The duration of each signal is approximately two seconds at the sampling rate is 48 kHz which is also used as the standard rate together with 44.1 kHz for high-quality audio recording.

During the research, a set of audio files was selected. The white Gaussian noise was added to the speech signal from each file. The noise level was chosen to obtain the noised signal with an SNR from 0 to 10 dB because the DST-IV is better to use at high noise levels as we observed. The initial speech signal is windowed on non-overlapping frames of N pixels where N is ranged from 3 to 9. The research was performed using an Intel Core i5-7400 processor, 3 GHz CPU, 16 GB memory, Windows 10 operating system, 64 bit. The provided experiment has shown that the fast DST-IV algorithms allow for reducing the speech signal denoising time by 62–78% as compared to direct matrix-vector product application. The researched speech signals contained about 90,000 samples. We also resampled these signals to obtain a sampling rate of about 8 kHz and to consider the low-quality audio recordings. The resampled signals consist of about 15,000 samples. In this case, the speech signal denoising time is decreased by 53–76%. As for the accuracy of speech signal denoising then applying DST-IV has given similar results as compared with DCT-II in the

Table 2. The number of operations for the algorithms known from the literature

Algorithm	Reference, publication year	N=3		N=4		N=8		N=9	
		Mults.	Adds.	Mults.	Adds.	Mults.	Adds.	Mults.	Adds.
Completely recursive radix-2 DIT algorithm	[4], 2021	-	-	8	12	20	36	-	-
Completely recursive algorithm	[20], 2018	-	-	10	10	30	30	-	-
Improved algorithm	[24], 2008	-	-	10	10	20	34	-	-
General radix algorithm	[22], 2008	3	6	8	12	20	36	20	38
Proposed algorithm	-	4	7	9	15	27	57	15	65

statistical error limits. In particular, for $N = 5$ the accuracy of DCT-based speech signal denoising is exceeded on 0.11-0.14 dB the DST-based denoising. And this estimation was not dependent on the signal sampling rate. For $N = 5$ the speech signal denoising time is decreased by 65-67% (0.51 sec against 0.16 sec on average) if the sampling rate is equal 8 kHz, and by 69-71% (3.12 sec against 0.95 sec on average) if the sampling rate is equal 48 kHz. For $N = 7$ the speech signal denoising time is decreased by 66-68% (0.40 sec against 0.13 sec on average) if the sampling rate is equal 8 kHz, and by 70-72% (2.47 sec against 0.70 sec on average) if the sampling rate is equal 48 kHz. For $N = 7$ the accuracy of DCT-based speech signal denoising is exceeded on 0.09-0.18 dB the DST-based denoising. Therefore this practical example of applying the proposed algorithms to speech signal denoising shows that the denoising accuracy can be slightly decreased while the processing time is significantly reduced.

6. CONCLUSIONS

This paper presents type-IV DST algorithms with reduced multiplicative complexity. Experimental research of developed algorithms compared their computational complexity with the direct calculation of matrix-vector products. The resulting factorizations of DST-IV matrices reduce the number of multiplications by an average of 63% but increase the number of additions by an average of 8% in the range of signal lengths from 3 to 9. The practical example of the applying the proposed algorithms to denoise the speech signals is provided.

If you analyze any of the signal flow graphs shown, you will see that they contain pre-additions, post-additions, and several multiplications in the middle like Winograd's fast Fourier transform algorithms. Unfortunately, for even-length input sequences, the reduction in the number of multiplications is not as significant as for odd-length sequences. This is somewhat surprising since for some other well-known discrete trigonometric transforms (DFT, discrete Hartley transform) the opposite situation holds. However, the fact remains a fact.

The advantage of the algorithms described in the paper compared to known algorithms (see, for example, [4,20]) is that the critical path in the signal flow graph of any of the presented algorithms contains only one multiplication. If there is more than one multiplication in the critical path of the algorithm, then this will create additional problems for the implementation of computations. As a result of multiplying two n-bit operands, a $2n$-bit product is obtained. The need for repeated multiplication requires more manipulations with the operands and therefore requires more time and effort than when dealing with only a single multiplication. In fixed-point devices, this fact can cause overflow-underflow handling. If we want to preserve the accuracy, then double access to the memory is required both when writing and when reading. Using floating-point arithmetic in this case also creates additional problems related to exponent alignment, mantissa addition, etc.

The solutions offered here are meant to be an addition to the collection of fast discrete trigonometric transform algorithms that numerous researchers have been honing for many years. The future development of the research might be the implementation of the proposed algorithms to combine the discrete transforms in hybrid forms [16]. Here, we introduce novel algorithms, but we make no claims about their optimality. This is all we have managed to obtain so far, and we want to share at least this. If someone can show better results, we will be very pleased.

REFERENCES

[1] V. Britanak, P. Yip and K. Rao, *Discrete cosine and sine transforms: general properties, fast algorithms and integer approximations*, 1st ed.; Academic Press Inc., Elsevier Science: Amsterdam, Holland, 2007. [Online]. Available: https://doi.org/10.1016/B978-0-12-373624-6.X5000-0

[2] G. Bi, and Y. Zeng, *Transforms and fast algorithms for signal analysis and representations*, 1st ed.; Birkhäuser: Boston, United States, 2004. [Online]. Available: http://doi.org/10.1007/978-0-8176-8220-0

[3] D.F. Chiper, and L.-T. Cotorobai, "A new approach for a unified architecture for type IV DCT/DST with an efficient incorporation of obfuscation technique". *Electronics* 2021, 10, 1656. [Online]. Available: http://doi.org/10.3390/electronics12214471

[4] S.M. Perera, and L. Lingsch, "Sparse matrix-based low-complexity, recursive, radix-2 algorithms for discrete sine transforms", *IEEE Access* 2021, 99, 1–1. [Online]. Available: http://doi.org/10.1109/ACCESS.2021.3120051

[5] A. Cariow, "Strategies for the synthesis of fast algorithms for the computation of the matrix-vector product", *Journal of Signal Processing Theory and Applications* 2014,

3(1), 1–19. [Online]. Available: http://doi.org/10.7726/jspta.2014.1001

[6] A. Cariow, M. Makowska, and P. Strzelec, "Small-size FDCT/IDCT algorithms with reduced multiplicative complexity", *Radioelectronics and Communications Systems* 2019, 62(11), 559–576. [Online]. Available: http://doi.org/10.3103/S0735272719110025

[7] A. Ajmera, M. Divecha, S.S. Ghosh, I. Raval, and R. Chaturvedi, "Video steganography: using scrambling-AES encryption and DCT, DST steganography", In the *Proceedings of IEEE Pune Section International Conference (PuneCon)*, Pune, India, 18-20 December 2019. [Online]. Available: https://doi.org/10.1109/PuneCon46936.2019.9105666

[8] V.P.S. Naidu, M. Divya, and P. Mahalakshmi, "Multimodal medical image fusion using multi-resolution discrete sine transform", *Control and Data Fusion e-Journal* 2017, 1(2), 13–26

[9] S. Garg, R. Yadav, and M. Kumar, "Discrete sine transform interpolation-based design of 2-D FIR fractional delay digital filter", In the *Proceedings of 2nd International Conference on Computational Electronics for Wireless Communications (ICCWC)*, Mangalore, India, 9-10 June 2022. [Online]. Available: https://doi.org/10.1007/978-981-19-6661-3_38

[10] X. Zhou, C. Wang, Z. Zhang, and Q. Fu, "Interpolation filter design based on all-phase DST and its application to image de-mosaicking", *Information* 2018, 9(9), 206. [Online]. Available: https://doi.org/10.3390/info9090206

[11] L.O. Hnativ, "Discrete cosine-sine transform type VII and fast integer transforms for intra-prediction images and video coding", (In Ukrainian). *Cybernetics and Systems Analysis* 2022, 57 (5), 175–185. [Online]. Available: https://doi.org/10.1007/s10559-021-00408-z

[12] T.A. Chowdary, and P. Nalluri, "DST-VII based multiple transform selection algorithms for versatile video coding", In the *Proceedings of 3rd International Conference on Artificial Intelligence and Signal Processing (AISP)*, Vijayawada, India, 18-20 March 2023. [Online]. Available: https://doi.org/10.1109/AISP57993.2023.10134908

[13] F.S. Al-Kamali, A.F. Al-Junaid, and M.Y.H. Al-Shamri, "New image transmission schemes for DST-based MC-CDMA system" *Arabian Journal for Science and Engineering* 2021, 46(1), 1465–1479. [Online]. Available: https://doi.org/10.1007/s13369-020-05173-3

[14] S. Malini, and R.S. Moni, "Use of discrete sine transform for a novel image denoising technique", *International Journal of Image Processing* 2014, 8(4), 204–213

[15] M. Masera, L. Re Fiorentin, E. Masala, G. Masera, and M. Martina, "Analysis of HEVC transform throughput requirements for hardware implementations", *Signal Processing: Image Communication*, 2017, 57, 173-182. [Online]. Available: https://doi.org/10.1016/j.image.2017.06.001

[16] W.A. Jassim, and N. Harte, "Comparison of discrete transforms for deep-neural-networks-based speech enhancement", *IET Signal Processing* 2022, 16(4), 438–448. [Online]. Available: https://doi.org/10.1049/sil2.12109

[17] S.R. Park, and J. Lee, "A fully convolutional neural network for speech enhancement", In the *Proceedings of the 18th Annual Conference of the International Speech Communication Association (INTERSPEECH 2017)*, Stockholm, Sweden, 20-24 August 2017. [Online]. Available: https://doi.org/10.21437/Interspeech.2017-1465

[18] Y. Hu,Y. Liu, S. Lv, M. Xing, S. Znang, Y. Fu, J. Wu, B. Zhang, and L. Xie, "DCCRN: deep complex convolution recurrent network for phase-aware speech enhancement", In the *Proceedings of the 21st Annual Conference of the International Speech Communication Association (INTERSPEECH 2020)*, Shanghai, China, 25-29 October 2020. Online]. Available: https://doi.org/10.21437/Interspeech.2020-2537

[19] M. Yustisar, P. Sihombing, and S. Efendi, "Discrete sine transform analysis, discrete cosine transform and discrete Fourier transform for introduction to voice register", *International Journal of Research and Review*, 2020, 7(2), 155–162. [Online]. Available: https://www.ijrrjournal.com/IJRR_Vol.7_Issue.2_Feb2020/IJRR0024.pdf

[20] S.M. Perera, "Signal flow graph approach to efficient and forward stable DST algorithms", *Linear Algebra and its Applications* 2018, 542, 360–390. [Online]. Available: https://doi.org/10.1016/j.laa.2017.05.050

[21] K. Bielak, A. Cariow, and M. Raciborski, "The development of fast DST-II algorithms for short-length input sequences", *Electronics* 2024, 13(12), 2301. [Online]. Available: https://doi.org/10.3390/electronics13122301

[22] M. Püschel, and J.M.F. Moura, "Algebraic signal processing theory: Cooley-Tukey type algorithms for DCTs and DSTs", *IEEE Transactions on Signal Processing* 2008, 56(4), 1502–1521. [Online]. Available: https://doi.org/10.1109/TSP.2007.907919

[23] H. Li, P. Li, Y. Wang, Q. Wang, and L. Gao, "A new decomposition algorithm of DCT-IV/DST-IV for realizing fast IMDCT computation", *IEEE Signal Processing Letters* 2009, 16(9), 735–738. [Online]. Available: https://doi.org/10.1109/LSP.2009.2022789

[24] X. Shao, and S.G. Johnson, "Type-IV DCT, DST and MDCT algorithms with reduced number of arithmetic operations", *Signal Processing* 2008, 88(6), 1313–1326. [Online]. Available: https://doi.org/10.1016/j.sigpro.2007.11.024

[25] D.F. Chiper, and L.-T. Cotorobai, "An improved algorithm for an efficient VLSI implementation of type IV DST using short quasi-band correlation structure", In the *Proceedings of 14th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, Ploiesti, Romania, 30 June - 1 July 2022. [Online]. Available: https://doi.org/10.

3390/electronics12010243

[26] D.F. Chiper, and A. Cracan, “An area-efficient unified VLSI architecture for type IV DCT/DST having an efficient hardware security with low overheads”, *Electronics* 2023, 12(21), 4471. [Online]. Available: https://doi.org/10.3390/electronics12214471

[27] M. Ponomarenko, O. Miroshnichenko, V. Lukin, and K. Egiazarian, “Blind estimation of noise level based on pixels values prediction”, In the *Proceedings of IS&T International Symposium on Electronic Imaging (EI 2022)*, online, 17-26 January 2022. [Online]. Available: https://doi.org/10.2352/EI.2022.34.14.COIMG-152

[28] V. Lukin, S. Krivenko, and V. Oliinyk, “Blind estimation of noise variance for 1D signal denoising”. *Telecommunications and Radio Engineering* 2020, 79(7), 567–581. [Online]. Available: https://doi.org/10.1615/TelecomRadEng.v79.i7.30

[29] M. Polyakova, A. Witenberg, and A. Cariow, ”The design of fast type-V discrete cosine transform algorithms for short-length input sequences”, *Electronics* 2024, 13(21), 4165. [Online]. Available: https://doi.org/10.3390/electronics13214165

[30] A. Cariow, and Ł. Lesiecki, ”Small-size algorithms for type-IV discrete cosine transform with reduced multiplicative complexity”, *Radioelectronics and Communications Systems* 2020, 63(9), 465–487. [Online]. Available: https://doi.org/10.3103/S0735272720090022

[31] M. Raciborski, A. Cariow, J. Bandach, "The development of fast DST-I algorithms for short-length input sequences", *Electronics*. 2024, 13, 5056. [Online]. Available: https://doi.org/10.3390/electronics13245056

[32] P. Brysin, and V. Lukin, ”DCT-based denoising of speech signals”, *Herald of Khmelnytskyi National University*. 2024, 4(339), 301–309. [Online]. Available: https://doi.org/10.31891/2307-5732-2024-339-4-48

[33] Luo, Y; Mesgarani, N. ”Conv-TasNet: surpassing ideal time–frequency magnitude masking for speech separation”, *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 2019, 27(8), 1256–1266

[34] TSP speech database. [Online]. Available: https://www.mmsp.ece.mcgill.ca/Documents/Data/TSP-Speech-Database/TSP-Speech-Database.pdf