Algorithm of efficient computation DST^{I-IV} using cyclic convolutions

IHOR PROTS'KO, VASYL TESLYUK
CAD Department
Lviv Polytechnic University
S.Bandery Str.,12, Lviv,79046,
UKRAINE
protsko@polynet.lviv.ua, vtesliuk@polynet.lviv.ua

Abstract: - The general method for efficient computation of discrete sine transform (DST) of sequences of arbitrary number of points using cyclic convolutions is considered. Forming hashing arrays on the basis of simplified arguments of basis sine transform for synthesis of efficient algorithm is analyzed. The hashing arrays in algorithm define partitioning of the basis into shift cyclic submatrices**.** The examples for size 8 of four types of DST $^{I-IV}$ using proposed method are analyzed. The hashing arrays, used in the algorithms of synthesis technique, are more versatile and generally better in terms of indexing mapping in comparison with the existing algorithms.

Key words: - discrete sine transform, types of DST, algorithm, hashing array, synthesis, cyclic convolution.

1 Introduction

The discrete Fourier transform (DFT) is widely used in many applications. In parallel with the development of computing means, the applications for efficient computation of DFT classes spread. Many studies on FFT (Fast Fourier Transform) emphasized the prospects of further application of only real discrete transforms of harmonic basis. In 1974 a discrete cosine transform (DCT) [1] and in 1976 a discrete sine transform (DST) [2,3] were proposed, a real basis which reproduces functionality in space or time dependence, similar to the DFT. Cosine and sine discrete transform and DFT interrelate strict mathematical forms that allow finding an effective way to compute one transform through another [4].

DST usage is widespread for several reasons. Firstly, basic DST functions are well approximated to a big number of stationary stochastic processes, which allows describing the signal with a given accuracy of minimal number of components. Secondly, DST contains a number of specific properties and due to this gives good results when processing weakly correlated signal of transform, which leads to consideration of significant signal energy. DST is used in many applications, especially in the processing of digital audio and video signals [5]. Further intensive development of information technologies puts higher demands on performance, functionality and specific opportunities for algorithmic, software and hardware real discrete transforms.

Efficient computing of one and two dimensional DST has been investigated for more than three decades. A significant number of publications is devoted to the efficient computation of DST [6]. Multivariate effective computing algorithms are divided into: algorithms for size of radix two, split radix, mixed radix, odd size, and prime factors composite transform size.

For the synthesis of efficient algorithms of DST, the following approaches are used:
1) direct matrix factorization of DST;
2) indirect calculation by FFT or through discrete Hartley transform, DCT;
3) algorithms based on the theory of complexity.

Generalized method based on polynomial transformations are used for synthesis of efficient algorithms. It was shown that four types of DST have a group symmetry (properties of the theory of groups and their representations) and for each of them fast algorithm is derived purely algebraically [7]. There are efficient computations of DST, which are embodied in the form of specific algorithms [8]. Summarized and systematized works on the study of fast DST and final step in this direction is the theory of the synthesis of fast algorithms [9]. The theory in [7,9] does not consider "Rader type" [10] algorithms.

That approach of efficient algorithms is the possibility to compute DST through the cyclic convolutions. The convolution-based algorithms are found to be efficient for very large scale integration (VLSI) implementation. Cyclic convolutions can be realized through very simple hardware structures. Because the usage of cyclic convolution or circular correlation structures provides high computing speed, low computational complexity, and low I/O bandwidth, it reduces the hardware cost. A lot of the papers [11-13] appeared about efficient hardware architectures connected with the computation through cyclic convolution. The proposed design uses an efficient restructuring of the computation of the DST for prime-number transform length into two circular correlations, having similar structures and only one half of the length of the original transform.

The main idea of the article is the representation of generalized algorithm of DST computation via cyclic convolution of arbitrary-number transform lengths. The proposed method, which uses hashing array, is more general in comparison to the referred Raiders approach that uses primitive roots and Chinese remainder theorem.

The paper is structured as follows. Section 2 describes types of discrete sine transform (subsection 2.1) and simplified arguments of basis DST^{I-IV} (subsection 2.2) for general algorithm for DST using cyclic convolutions (subsection 2.3). In Section 3, the performance of the proposed examples of DST^{I-IV} types for size N=8 are analyzed (subsection 3.1-3.4) and then conclusions are presented. The examples are provided for different DST types to show suitability and efficiency of the proposed method for DST^{I-IV}.

2 Efficient computation of the DST^{I-IV} using cyclic convolutions

Computation types DST and IDST (direct and inverse) are one of the most long-term procedures in information technology, such as analyzing and processing sizable data. This procedure requires the greatest degree of improvements that will speed up works of software and hardware. Efficient computation of DST using cyclic convolutions is important and needs development.

2.1 Types of discrete sine transform

Discrete sine transform reflects the input data to linear combination of weighted basis functions. There are 8 types of discrete sine transform discussed in [14]. This transform is further improved by the DFT for real input data. DSTs operate on finite, discrete sequences. The axis of symmetry of continuous discrete sine transform can be on the sample (even number) or between two samples, which corresponds to a shift in the half interval sampling. This allows different options of transform under the boundary conditions for real input data. Continuation of the input data can be extended: whole sample symmetrically (WS), whole sample asymmetrically (WA), half interval sampling symmetrically (HS) and half interval sampling asymmetrically (HA). There are only two axes of symmetry for a limited sequence and, consequently, a possible set of alternatives ε-type extension that corresponds to 8 types of DST (Table 1).

Table 1: Set options for ε-type expansion

ε	WAWA	HAHA	WAWS	HAHS
DST	DST^{I}	DST^{II}	DST^{III}	DST^{IV}

ε	WAHA	HAWA	WAHS	HAWS
DST	DST^{V}	DST^{VI}	DST^{VII}	DST^{VIII}

The relationship of direct and inverse computation of four types of DST can be presented in the following form:

$(DST^{I}_{N})^{-1} = (DST^{I}_{N})^{T} = (DST^{I}_{N})$;
$(DST^{II}_{N})^{-1} = (DST^{II}_{N})^{T} = (DST^{III}_{N})$;
$(DST^{III}_{N})^{-1} = (DST^{III}_{N})^{T} = (DST^{II}_{N})$;
$(DST^{IV}_{N})^{-1} = (DST^{IV}_{N})^{T} = (DST^{IV}_{N})$.

DST^{I}, DST^{IV} specify symmetrical forward and backward transform, and transform of DST^{II} and DST^{III} type specify transfer in one second.

Consider the efficient computation of DST^{I-IV} using cyclic convolutions that will speed up the work of software and hardware for many applications.

2.2 Define simplified arguments of basis DST^{I-IV}

Information technologies widely use DST^{I-IV} types, which can be represented respectively by the following formula:

for DST^{I}

$$X^{s1}_{N-1}(k) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-2} x(n) \sin[\frac{(k+1)(n+1)\pi}{N}], \quad k = 0,1,...,N-2 \qquad (1)$$

for DST^{II}

$$X^{s2}_{N}(k) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} \alpha(n) x(n) \sin[\frac{(k+1)(2n+1)\pi}{2N}], \quad k = 0,1,...,N-1 \qquad (2)$$

for DST ^{III}

$$X_N^{s3}(k)=\sqrt{\frac{2}{N}}\sum_{n=0}^{N-1}\alpha(n)x(n)\sin[\frac{(2k+1)(n+1)\pi}{2N}],\quad k=0,1,...,N-1 \qquad (3)$$

for DST ^{IV}

$$X_N^{s4}(k)=\sqrt{\frac{2}{N}}\sum_{n=0}^{N-1}x(n)\sin[\frac{(2k+1)(2n+1)\pi}{4N}],\quad k=0,1,...,N-1 \qquad (4)$$

where $\alpha(n)=1/\sqrt{2}$, if n=N-1; otherwise $\alpha(n)=0$.
Analyze the structure of the matrix basis for the types of DST for arguments, where components $c_{k,n}$ are respectively:

for DST ^{I}

$$c_{k,n}=(k+1)(n+1)\pi/N,\ (k,n=0,1,\ldots N-2); \qquad (5)$$

for DST ^{II}

$$c_{k,n}=(k+1)(2n+1)\pi/2N,\ (k,n=0,1,\ldots N-1); \qquad (6)$$

for DST ^{III}

$$c_{k,n}=(2k+1)(n+1)\pi/2N,\ (k,n=0,1,\ldots N-1); \qquad (7)$$

for DST ^{IV}

$$c_{k,n}=(2k+1)(2n+1)\pi/4N,\ (k,n=0,1,\ldots N-1). \qquad (8)$$

Basis periodic (2π), symmetric (π) and asymmetric ($\pi/2$) for each type of DST, are presented respectively in Table 2.

Table 2: Properties basis for types of DST

Type	periodic	asymmetric	symmetric
DST ^{I}	2N	N	N/2
DST ^{II}	4N	2N	N
DST ^{III}	4N	2N	N
DST ^{IV}	8N	4N	2N

Matrix arguments C_a each type DST for property of periodic is equal respectively

$$C^{I}_{a}(k,n) = [(k+1)(n+1) \bmod (2N)], \qquad (9)$$
$$C^{II}_{a}(k,n) = [(k+1)(2n+1) \bmod (4N)], \qquad (10)$$
$$C^{III}_{a}(k,n) = [(2k+1)(n+1) \bmod (4N)], \qquad (11)$$
$$C^{IV}_{a}(k,n) = [(2k+1)(2n+1) \bmod (8N)]. \qquad (12)$$

Based on substitutions of rows of data matrix (9-12) the hashing arrays P(n) are formed that define block cyclic structures of basis matrix. Accordance properties of simplified matrix elements of the arguments is determined by consistent performances

for DST ^{I}

$$c_{k,n} = 2N-[(c_{k,n}) \bmod 2N],\quad \text{if } [(c_{k,n}) \bmod 2N]>N; \qquad (13)$$
$$\underline{c}_{k,n} = N-\{2N-[(c_{k,n}) \bmod 2N]\},\quad \text{if } \{2N-[(c_{k,n}) \bmod 2N]\}>N/2, \qquad (14)$$

otherwise $\underline{c}_{k,n} = c_{k,n}$.

for DST ^{II} and for DST ^{III}

$$c_{k,n} = 4N-[(c_{k,n}) \bmod 4N],\quad \text{if } [(c_{k,n}) \bmod 4N]>2N; \qquad (15)$$
$$\underline{c}_{k,n} = 2N-\{4N-[(c_{k,n}) \bmod 4N]\},\quad \text{if } \{4N-[(c_{k,n}) \bmod 4N]\}>N, \qquad (16)$$

otherwise $\underline{c}_{k,n} = c_{k,n}$.

for DST ^{IV}

$$c_{k,n} = 8N-[(c_{k,n}) \bmod 8N],\quad \text{if } [(c_{k,n}) \bmod 8N]>4N; \qquad (17)$$
$$\underline{c}_{k,n} = 4N-\{8N-[(c_{k,n}) \bmod 8N]\},\quad \text{if } \{4N-[(c_{k,n}) \bmod 8N]\}>2N, \qquad (18)$$

otherwise $\underline{c}_{k,n} = c_{k,n}$.

Simplified matrix arguments complement matrices Ss of sine signs, defined by the inequalities

for DST^{I}

$$Ss[k,n]=\begin{cases}+1, & if \quad 0<c_{k,n}<N\\ 0, & if \quad c_{k,n}=N,\\ -1, & if \quad N<c_{k,n}<2N\end{cases}, \qquad (19)$$

for DST ^{II} and for DST ^{III}

$$Ss[k,n]=\begin{cases}+1, & if \quad 0<c_{k,n}<2N\\ 0, & if \quad c_{k,n}=2N\\ -1, & if \quad 2N<c_{k,n}<4N\end{cases}, \qquad (20)$$

for DST ^{IV}

$$Ss[k,n]=\begin{cases}+1, & if \quad 0<c_{k,n}<4N\\ 0, & if \quad c_{k,n}=4N\\ -1, & if \quad 4N<c_{k,n}<8N\end{cases}. \qquad (21)$$

Therefore, expressions (13,15,17) define elements and form the hashing array P(n) and then with P(n) expressions (14,16,18) define elements of simplified hashing arrays P'(n) and matrix signs Ss (19,20,21) taking part in the efficient computation of DST $^{I\text{-}IV}$.

2.3 Synthesis of efficient algorithms of DST$^{I\text{-}IV}$ on basis of cyclic convolutions

One approach of efficient algorithms gives the possibility to compute DST through the cyclic convolutions. Most papers use a transition from discrete transform to compute cyclic convolutions mapping for simple size by Raiders [10] or split composite size on prime factors by Agarwal and Cooley [15] or the combination of these approaches. Using the methods of computation for each type of DST based on cyclic convolutions has different specifics and is analyzed in this paper.

The proposed approach for efficient computation of discrete sine transforms is based on decomposition of basis matrix in block-cyclic structures [16,17]. As a result, block-cyclic structure of basis matrix can be set as a hashing array

$$P(n)=P(n_1)\,P(n_2)\,\ldots\,P(n_k)=(n_{11},\, n_{12},\, n_{13,\,\ldots,}\, n_{1L1})$$

$$(n_{21}, n_{22}, n_{23,\ldots}, n_{2L2}) \ldots (n_{kL1}, n_{kL2}, \ldots, n_{kLk}), \qquad (22)$$

where k – the number of subarrays, n_{ij} - element of a subarray; L_i - the number of elements in the subarray $P(n_i)$; n - size of the total array for P(n), which is determined by:

$$n = (L_1+L_2+\ldots+L_k)\ . \qquad (23)$$

The k number of subarrays in P(n) is determined by the value of N size (simple, easy power, composite) of transform and types of DST. Hashing array P(n) specifies the order of the input data for performance of the discrete transform.

The properties of symmetry and periodicity of DST basis lead to lower values representation of elements of subarrays $P'(n_i)$ with supplement respective subarrays of sine signs $Ss(n_i)$. Submatrices $Ss(n_i)$ contain elements that can be equal to +1, -1, 0.

Hashing array P(n) of transform defines specific the structure of DST basis matrix reduced to cyclic submatrices. So are the specific parameters P(n) that characterize accordingly modified basis matrix:
- the number of subarrays in hashing array k;
- the size of subarrays (L_1, L_2, ..., L_k);
- first element of each subarray n_{i1}, i = 1 (1) k.

The next step in the synthesis algorithm for computation of DST is to determine cyclic submatrices identity. That finding of identical and quasi identical submatrices (with the same index, but opposite signs) is based on the values of parameters of hashing array P(n) and hashing array P'(n), supplemented array Ss(n) of sine signs. The form of hashing arrays P'(n), Ss(n) are

$$P'(n)=P'(n_1)P'(n_2)\ldots P'(n_k),$$
$$Ss(n)= Ss(n_1)\ Ss(n_2)\ldots Ss(n_k) \qquad (24)$$

for a given N size and of DST type determined by a simplified matrix, respectively (14,16,18) with elements $\underline{C}_a$, and signs Ss(n) are defined by (19,20,21).

To find identity cyclic submatrices among value elements of basis matrix may in advance, but a large search of all elements requires significant amount of memory to store and associated time costs. More effectively identify only the first elements of submatrices in the analysis of the structure on the basis of coordinates for placement submatrices. That is calculated of the first elements of submatrices using the coordinates of the row and column values and is analyzed for value of the first elements.

In compliance with coordinates (i,j) of the elements, hashing array $P(n_i)$ and $P'(n_i)$ are:

(i\ j)	1 2 3 4 5 6 7 8 9 … n
$P(n_i)$	$(n_{11}, n_{12,\ldots}, n_{1L1}, n_{21}, n_{22,\ldots}, n_{2L2}, \ldots, n_{kL1}, \ldots, n_{kLk})$
$P'(n_i)$	$(n'_{11}, n'_{12,\ldots}, n'_{1L1}, n'_{21}, n'_{22,\ldots}, n'_{2L2}, \ldots, n'_{kL1}, \ldots, n'_{kLk})$.

where integers n_{ij} , $0 < n_{ij} < T$; $0 \le n'_{ij} \le n_{ij}$, T – period of DST basis.

Coordinates of the first elements of submatrices are determined by (i+Li), (j+Li), where Li - is the size of hashing subarrays, that are chosen for condition of membership value of the first elements of the submatrices in the matrix structure to the element of hashing subarrays. The first elements are calculated by relevant coordinates (i, j) and match the element P'(n) of hashing arrays defined by expressions (14,16,18) in accordance with the type of DST transform. Example of general matrix structure for coordinates is presented in Table 3.

Table 3: Table of coordinates and first elements of submatrix

(i+L$_i$, j+L$_i$) – $s_{i,j}$ $\underline{c}_{i,j}$; **(**sign and value)		
(1,1) – $s_{i,j}$ $\underline{c}_{i,j}$;		…
(1+ L_1,1) – $s_{i,j}$ $\underline{c}_{i,j}$;		
(1+ L_1+ L_2,1) – $s_{i,j}$ $\underline{c}_{i,j}$;		…
...		
(1+$L_{1\ldots}$+L_k,1) – $s_{i,j}$ $\underline{c}_{i,j}$;	(1+$L_{1\ldots}$+L_k, 1+L_k) – $s_{i,j}$ $\underline{c}_{i,j}$;	…

Definition of identity and quasi identity of cyclic submatrices is performed by selecting the coordinates equal $\underline{c}_{i,j}$ first elements of identical submatrices horizontally. Combining element-wise addition values of input data correspond to the coordinates of first elements of identical submatrices. These values will be used to compute cyclic convolutions. Computation of single cyclic convolutions is performed when analyzing identity submatrices vertically in the matrix structure. For the non-identity submatrices of cyclic convolution for specified coordinates is conducted in the case of analyzing process of the whole matrix structure.

Combining the results of convolutions is performed horizontally at the base of according coordinates of the first elements of submatrices. The resulting output data of transform corresponds to rows with the order according to the elements of hashing array P(n).

3 Algorithms DST^{I-IV} using cyclic convolutions in examples

Distribution of cyclic submatrices in basis matrix structures and characteristics of hashing array P(n) determine the complexity of the algorithm for efficient computation of DST^{I-IV} types. This approach of efficient computation of DST uses availability of the fast convolution algorithms [18].

Many implementations and our examples have sequences with reiterative identical groups of elements of cyclic convolution. Consider some cases of identical sequences. In the following case, $h(n) = (h_1,h_2,\ldots,h_m,\ h_1,h_2,\ldots,h_m)$ or if identical group of elements has inverse sign $h(n) = (h_1,h_2,\ldots,h_m,\ -h_1,-h_2,\ldots,-h_m)$, n=2m, the cyclic convolution is equal (25)

$$\begin{pmatrix} h(m) \pm h(m) \\ \pm h(m)\, h(m) \end{pmatrix} \otimes \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \begin{pmatrix} h(m) \otimes (x_0 \pm x_1) \\ \pm h(m) \otimes (x_0 - x_1) \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \end{pmatrix}, \quad (25)$$

where $h(m)=(h_1,\ h_2,\ldots,\ h_m)$, $x_0=(x_1,\ x_2,\ldots,\ x_m)$, $x_1=(x_{m+1},\ x_{m+2},\ldots,\ x_{2m})$, $\otimes$-operation of cyclic convolution.

Therefore, n/m – number of reiterations of identical group of elements of the sequence of cyclic convolution for size n, which reduces computational complexities by n/m times.

3.1 Specific algorithm for DST^I of size N=8

Consider the example of a generalized scheme for synthesis of algorithm and computation of DST^I of size N=8. The hashing array $P(n_i)$ and $P'(n_i)$ are:

$P(15) = P(n_1)P(n_2)P(n_3)P(n_4)P(n_5)P(n_6) =$
= (1,3,9,11) (15,13,7,5)(2,6)(10,14)(4,12)(8)(16).

P'(15) = (1,3,1,3)(1,3,1,3)(2,2)(2,2)(4,4)(0);
Ss(15) = (+,+,-,-)(+,-,-,+)(+,+)(-,-)(+,-)(0)(0);

Definition of identity cyclic submatrices is performed by selecting the coordinates of the first elements identical submatrices without signs in the basis matrix. In correspondence with coordinates (i,j) in Table 1, hashing array elements $P(n_i)$ and $P'(n_i)$ are:

(i, j) 1 2 3 4 5 6 7 8 9 10 11 12 13 14 1516
P(15)(**1**,**3**,9,11)(15,13,**7**,**5**)(**2**,**6**)(10,14)(**4**,12)(8)(16)
P'(15)(1,3,1,3) (1,3,1,3) (2,2) (2,2) (4,4) (0)(0);
Ss(15) (+,+,-,-) (-,-,+,+) (+,+) (-,-) (+,-) (0)(0).

Coordinates of the first elements of submatrices are determined by (i + Li), (j + Li), starting with i = 1, j = 1. The value of the first elements of submatrices are calculated by matching the coordinates (i, j) and the elements of P(n) hashing array using formula $(n_i \times n_j) \bmod N$, in the case of a value greater than N using simplified expression (13,14).

Table 4: Table of coordinates and first elements of matrix DST^I, N=8

<table>
<tr><th colspan="6">($i+L_i$, $j+L_i$) - $s_{ij}n_{ij}$</th></tr>
<tr><td rowspan="2">(1,1) – +1;</td><td rowspan="2">(1,5) – -1;</td><td>(1,9) – +2;</td><td>(1,11) – -2;</td><td rowspan="2">(1,13) – +4;</td><td rowspan="4">(1,15) – 0;</td></tr>
<tr><td>(3,9) – +2;</td><td>(3,11) – -2;</td></tr>
<tr><td rowspan="2">(5,1) – -1;</td><td rowspan="2">(5,5) – +1;</td><td>(5,9) – -2;</td><td>(5,11) – +2;</td><td rowspan="2">(5,13) – -4;</td></tr>
<tr><td>(7,9) – -2;</td><td>(7,11) – +2;</td></tr>
<tr><td>(9,1) – +2;</td><td>(9,5) – -2;</td><td colspan="2">(9,9) – +4;</td><td colspan="2">(9,15) – 0;</td></tr>
<tr><td>(11,1) – +4;</td><td>(11,5) – -4;</td><td colspan="4">(11,9) – 0;</td></tr>
</table>

Table 4 summarizes the basis matrix of arguments of dimension (11x15), where the number of horizontal and vertical elements is indicated. According to hashing array P(15) for DST^I, N=8, 11 horizontal rows are selected, allowing seven to compute the output values via convolution with identical group. With hashing arrays P'(15) can be reproduced the basis matrix, whose arguments resulted in a form of cyclic submatrices without signs. Part of matrix simplifies arguments without signs of sine basis transform of type I for N=8 presented in Table 5, which corresponds to the generalized Table 4.

Table 5: Part table of values of simplified elements without signs of matrix DST^I, N=8

k \ n	1:	3:	9:	11:	15:	13:	7:	5:	…
1:	1	3	1	3	1	3	1	3	…
3:	3	1	3	1	3	1	3	1	…
9:	1	3	1	3	1	3	1	3	…
11:	3	1	3	1	3	1	3	1	…
15:	1	3	1	3	1	3	1	3	…
13:	3	1	3	1	3	1	3	1	…
7:	1	3	1	3	1	3	1	3	…
5:	3	1	3	1	3	1	3	1	…
2:	2	2	2	2	2	2	2	2	…

6:	2	2	2	2	2	2	2	2	…
4:	4	4	4	4	4	4	4	4	…

Hashing array
P(n)→(1,3,9,11,15,13,7,5,2,6,10,14,4,12,8,16)
specifies the order of elements of input data of the discrete sine transform using cyclic convolutions. Initial order of input data is:

1 2 3 4 5 6 7 8
x(1), x(2), x(3), x(4), x(5), x(6), x(7), x(8),
9 10 11 12 13 14 15 16
-x(8), -x(7), -x(6), -x(5), -x(4), -x(3), -x(2), -x(1).

In accordance with P(n), matrix-column has such an order of input data:

x(1), x(3), -x(8), -x(6), -x(2), -x(4), x(7), x(5),
x(2), x(6), -x(7), -x(3), x(4), -x(5), x(8), -x(1).

Performance element-wise additions of input data will be used for identity and quasi identity cyclic submatrices placed horizontally. Computation of cyclic convolution is performed once for combined input data for identity and quasi identity submatrices selected for analysis vertically. The number of cyclic convolutions DST^I of size N=8 is a 4-point convolution with identical sequences and two of one points.

Combining the results of convolutions is performed horizontally at the base coordinates according to the first elements of submatrices. Output data of transform in a result of computation are scaled by two in the following order: X(1), X(3), X(7), X(5), X(2), X(6), X(4).

3.2 Specific algorithm for DST^{II} of size N=8

Consider the example of a generalized scheme for synthesis of algorithm and computation of DST^{II} for size N=8 with hashing array which is extended:

P(n)=(1,3,9,27,17,19,25,11)(31,29,23,5,15,13,7,21)(2,6,18,22)(30,26,14,10)(4,12)(28,20)(8,24)(16)(32) .

Basis matrix of arguments DST^{II} with elements for (10) contains the values (2k+1) of the elements in the first row quantity – 16, which cover the entire period equal to 4N, and (n+1) of the elements in the first column – 31, which cover the entire period equal to 4N.

A difference of values in rows and columns requires transition from hashing array P(n) to the appropriate hashing array indices of the rows Pr(n) and columns Pc(n):

P(n)=(1,3,9,27,17,19,25.11)(2,6,18,22)(4,12)(8,24);

Pr(n)=(0,2,8,26,16,18,24,10)(1,5,17,21)(3,11)(7,23;
Pr'(n) = (1,3,7,5,1,3,7,5) (2,6,2,6)(4,4)(8,8),
Ss(n) = (+,+, +, -, -, -, -, +)(+,+,-,-)(+,+)(+,-);

Pc(n) = (0,1,4,13,8,9,12,5) (15,14,11,2,7,6,3,10);
Pc'(n) = (1,3,7,5,1,3,7,5) (1,3,7,5,1,3,7,5),
Ss(n)= (+,+, +, -, -, -, -, +)(+, +, +, -, -, -, -, +).

According to hashing array Pr(n) for DST^{II} horizontal 14 rows are selected that allow computing 8 output data through cyclic convolutions. The convolution of two elements (the same) is replaced by convolution with a single point, respectively, for lines 3,7. Table 6 summarizes the basis matrix of arguments (14x16), where the numbers of values of horizontal and vertical are indicated.

Table 6: Table of coordinates and first elements of submatrix DST^{II}, N=8

<table>
<tr><th colspan="4">($i+L_i$, $j+L_i$) - $s_{ij}n_{ij}$</th></tr>
<tr><td colspan="2">(1,1) – +1;</td><td colspan="2">(1,9) – -1;</td></tr>
<tr><td>(9,1) – +2;</td><td>(9,4) – +2;</td><td>(9,9) – -2;</td><td>(9,13) – -2;</td></tr>
<tr><td colspan="2">(13,1) – +4;</td><td colspan="2">(13,9) – +4;</td></tr>
<tr><td colspan="4">(14,1) – +8;</td></tr>
</table>

With hashing arrays Pr(16), Pc(16) arguments of the basis matrix can be reproduced, which result a form of cyclic submatrices without signs. Part of matrix of simplified arguments without signs of sine basis transform N=8 is presented in Table 7, which corresponds to the generalized Table 6.

Table 7: Part table of values of simplified elements without signs of matrix DST^{II}, N=8

k \ n	0:	1:	4:	13:	8:	9:	12:	5:	…
0:	1	3	7	5	1	3	7	5	…
2:	3	7	5	1	3	7	5	1	…
8:	7	5	1	3	7	5	1	3	…
26:	5	1	3	7	5	1	3	7	…
16:	1	3	7	5	1	3	7	5	…
18:	3	7	5	1	3	7	5	1	…
24:	7	5	1	3	7	5	1	3	…
10:	5	1	3	7	5	1	3	7	…

1:	2	6	2	6	2	6	2	6	…
5:	6	2	6	2	6	2	6	2	…
17:	2	6	6	2	2	6	6	6	…
21:	6	2	6	2	6	2	6	2	…
3:	4	4	4	4	4	4	4	4	…
7:	8	8	8	8	8	8	8	8	…

Hashing array of Pc(n) specify the order of (0, 1, 4, 13, 8, 9.12, 5) (15, 14, 11, 2, 7, 6, 3, 10) input data when we conduct the discrete sine transform using cyclic convolutions.

Performance of element-wise additions of input data will be used for analysis of identity and quasi identity cyclic submatrices placed horizontally. Computation of cyclic convolution is performed once for combined input data for identity and quasi identity submatrices selected for analysis vertically. The number of cyclic symmetric convolutions of DST^{II} of size N=8 are the 8-point cyclic convolution with identical sequences, which resulted in four defined output data, and 4-point cyclic convolution with identical sequences, which resulted two output data. The remaining two output data are determined through one point products.

Combining the results of cyclic convolutions is performed at the base accordance coordinates of the first elements of submatrices horizontally. Output data of transform in a result of computation is scaled by two and defined for: X(0), X(2), -X(4), -X(6), X(1), X(5), X(3), X(7). Output data X(4), -X(6) must be taken with the opposite sign under P(n) of this algorithm.

Since DST^{III} is a transposed basis to DST^{II}, rearranging rows and columns of the basis, can be a synthesis for DST^{II} to efficiently compute and reverse transformation in general. However, this approach due to different indexing rows and columns is to some extent specific.

3.3 Specific algorithm for DST^{III} of size N=8

Consider the example for synthesis of algorithm and computation of DST^{III} N=8. In the contrary with DST^{II}, basis matrix of DST^{III} arguments with elements for (11) contains values (k+1) of the elements in the first row of quantity - 31 and covers the entire period equal to 4N. The first column contains values (2n+1) of quantity - 16 and covers the entire period equal to 4N. Hashing array for rows:

Pr(n) = (1,3,9,27,17,19,25,11) ,

rows define all 8 output data, where four output values for 27→5, 25→7, 19→13, 17→15 require the inversion of the sign of the result. The values of rows require transition from hashing array to the appropriate hashing array indexes (n-1)/2 of rows Pr(n):

(1,3,9,27,17,19,25,11) → (0,1,4,13,8,9,12,5).

Hashing array for column is more extended:
Pc(n)=(1,3,9,27,17,19,25,11)(31,29,23,5,15,13,7,2)(2,6,18,22)(30,26,14,10)(4,12)(28,20)(8,24)(16)(32) .

Permutation of number 32 columns is defined by the first horizontal row and the corresponding row in the matrix [(2k +1)(n +1)] mod 4N.
Simplified hashing array for column permutation has the form:

Pc'(n) = (1,3,7,5,1,3,7,5) (1,3,7,5,1,3,7,5) (2,6,2,6) (2,6,2,6) (4,4)(4,4) (8,8)(16)(32),
Ss(n) = (+,+,+,-,-,-,-,+) (-,-,-,+,+,+,+,-) (+,+,-,-) (-,-,+,+) (+,+)(-,-)(+,-)(0)(0).

Performance of element-wise subtractions of input data of simplified hashing array has the form:

Pc'(n) =(1,3,7,5,1,3,7,5)(2,6,2,6)(4,4)(8,8)(16)(32),
Ss(n) = (+,+,+,-,-,-,-,+) (+,+,-,-) (+,+) (+,-)(0)(0).

Determining identity cyclic submatrices required for efficient computation of DST^{III} for size N=8, is defined by analysis of Table 8 submatrices distribution in the structure of the basis matrix form.

Table 8: Table of coordinates and first elements of matrix DST^{III}, N=8

<table>
<tr><th colspan="5">($i+L_i$, $j+L_i$) - $s_{ij}n_{ij}$</th></tr>
<tr><td rowspan="4">(1,1) – +1;</td><td rowspan="2">(1,9) – +2;</td><td>(1,13) – +4;</td><td rowspan="4">(1,15) – +8;</td><td rowspan="4">(1,16) – 0;</td></tr>
<tr><td>(5,13) – +4;</td></tr>
<tr><td rowspan="2">(5,9) – +2;</td><td>(9,13) – +4;</td></tr>
<tr><td>(13,13) – +4;</td></tr>
</table>

The following table 8 summarizes basis matrix of arguments dimension (8x16), where the number of values of horizontal and vertical are indicated.

Hashing array of Pc(n) specifies the order of the input data in the computation of the discrete sine transform through cyclic convolutions. Continued input data to 4N, according algorithm, after performance of element-wise subtractions of input data matrix-column contains:

x(1)+x(2), x(3)+x(4), x(8)+x(7), -x(6)-x(5), -x(1)-x(2), -x(3)-x(4), -x(8)-x(7), x(6)+x(5), x(2)+x(3), x(6)+x(7), -x(2)-x(3), -x(6)-x(7), x(4)+x(5), x(5)+x(4), x(8)+x(8), x(1) .

Part of matrix of simplified arguments without signs of sine basis transform N=8 is presented in Table 9, which corresponds to the generalized Table 8.

Table 9: Part table of values of simplified elements without signs of matrix DST^{III}, N=8

k \ n	0:	2:	8:	26:	16:	18:	24:	10:	…
0:	1	3	7	5	1	3	7	5	…
1:	3	7	5	1	3	7	5	1	…
4:	7	5	1	3	7	5	1	3	…
13:	5	1	3	7	5	1	3	7	…
8:	1	3	7	5	1	3	7	5	…
9:	3	7	5	1	3	7	5	1	…
12:	7	5	1	3	7	5	1	3	…
5:	5	1	3	7	5	1	3	7	…

Difference indices in rows and columns need to be changed from 8 values hashing array to the according hashing array of indices (2k+1)→k for rows Pr(n):

(1,3,9,27,17,19,25,11) → (0,1,4,13,8,9,12,5).

The corresponding hashing array indices (n+1)→n for columns Pc(n):

(1,3,9,27,17,19,25,11)(2,6,18,22)(4,12)(8)(16) → (0,2,8,26,16,18,24,10)(1,5,17,21)(3,11)(7)(15).

Performance of element-wise additions of input data will be used for identity and quasi identity cyclic submatrices placed horizontally. Computation of cyclic convolution is performed once for combined input data for identity and quasi identity submatrices selected for analysis vertically. The number of cyclic convolutions DST^{III} of size N=8 are the 8-point cyclic convolution with identical sequences and two 4-point cyclic convolution with identical sequences. The remaining values are determined by one point products for vertically placed quasi identity submatrices.

Combining the results of cyclic convolutions is performed horizontally at the base coordinates of the first elements of submatrices. Output data of transform as a result of computation are scaled by four and determined for: X(0), X(1), X(4), -X(2), -X(7), -X(6), -X(3), X(5). Four output values -X(2),-X(7),-X(6),-X(3) must be taken with the opposite sign according to Pr(n) = (1,3,9,27,17, 19,25,11), where 27→5, 25→7, 19→13, 17→15, which correspond to a set of (2k+1) for k = 0(1)7. Therefore, horizontal 8 rows of respective hashing array Pr(n) are selected, allowing to compute output values via cyclic convolutions.

Function of arguments for DST^{II} and DST^{III} has different forms of rows and columns. Consequently a separate hashing array for row and column has to be applied, according to this approach. Hashing arrays for direct DST^{II} and hashing arrays for inverse $IDST^{II} = DST^{III}$ or vice versa differ in number of indices for Pr(n) and Pc(n) and corresponding structure of basis matrix with cyclic submatrices.

3.4 Specific algorithm for DST^{IV} of size N=8

Consider the example of synthesis of algorithm and computation of DST^{IV} with the values of the arguments [(2k+1)(2n+1)] for size N = 8. Basis matrix DST^{IV} for size N=8 describes the hashing array of arguments with elements smaller period equal 8N in the form:

P(32)=(1,3,9,27,17,51,25,11,33,35,41,59,49,19,57, 43)(63,61,55,37,47,13,39,53,31,29,23,5,15,45,7,21) .

Due to redundancy in the selected period 8N of hashing array, it is enough to present:

P(16)=(1,3,9,27,17,51,25,11,33,35,41,59,49,19,57, 43).

Owing to odd values of hashing array maps to the corresponding Pk(16) hashing array indices, elements $k_{i,j}=(n_{i,j}-1)/2$, and k takes all the values of natural set:

$P_k(16)$=(0,1,4,13,8,25,12,5,6,17,20,29,24,9,28,21).

Simplified hashing array and sign array have the form:

P'(n)=(1,3,9,5,15,13,7,11, 1,3,9,5,15,13,7,11);
Ss(n)= (+, +, +, +, +, -, +, +, -, -, -, -, -, +, -, -).

According 8 rows of hashing array $P_k(n)$ for DST^{IV} horizontal are selected that allow computing the 8 output data through cyclic convolutions. Part of matrix of simplified arguments with signs of DST^{IV} basis transform N=8 is presented in Table 10.

Table 10: Part table of values of simplified elements and signs of matrix DST^{IV}, N=8

k \ n	0:	1:	4:	13:	8:	25:	12:	5:	…
0:	1	3	9	5	15	13	7	11	…
1:	3	9	5	15	13	7	11	1	…
4:	9	5	15	13	7	11	1	3	…
13:	5	15	13	7	11	1	3	9	…
8:	15	13	7	11	1	3	9	5	…
25:	13	7	11	1	3	9	5	15	…
12:	7	11	1	3	9	5	15	13	…
5:	11	1	3	9	5	15	13	7	…
									…
0:	+	+	+	+	+	-	+	+	…
1:	+	+	+	+	-	+	+	-	…
4:	+	+	+	-	+	+	-	-	…
13:	+	+	-	+	+	-	-	-	…
8:	+	-	+	+	-	-	-	-	…
25:	-	+	+	-	-	-	-	-	…
12:	+	+	-	-	-	-	-	+	…
5:	+	-	-	-	-	-	+	-	…

Hashing array of P(32) specifies the order of input data when we conduct the discrete sine transform using cyclic convolutions. Performance of element-wise subtractions of input data will be used for convolution with cyclic submatrices placed horizontally. The number of cyclic convolutions DST^{IV} of size N=8 is only the 16-point cyclic convolution with identical sequences.

Output data of transform as a result of computation are scaled by two and determined for: X(0), X(1), X(4), X(2), X(7), -X(6), X(3), X(5). Four output values X(6) must be taken with the opposite sign according to $P_k(n)$=(0, 1, 4, 13, 8, 25, 12, 5), where 13→2, 8→7, 25→-6, 12→3. Therefore, horizontal 8 rows of hashing array Pk(n) are selected,, allowing to compute output data values via cyclic convolutions.

Specific algorithm DST^{IV} of size N=5 defines structure of basis matrix. The hashing array describes basic matrix DST^{IV} of arguments with elements of smaller period equal to 8N, and has the form:

P(20)=(1,3,9,27)(21,23,29,7)(19,17,11,33)
(39,37,31,13) (5,15) (25,35).

Due to redundancy in the selected period 8N of hashing array, it is enough to present:

P(10) = (1,3,9,27) (21,23,29,7) (5) (25).

Owing to odd values of hashing array maps to the corresponding Pk(10) hashing array indices, elements $k_{i,j}=(n_{i,j}-1)/2$, and k takes all the values of natural set:

$P_k(10)$ = (0,1,4,13) (10,11,14,3) (2) (12).

Simplified hashing array and sign array have the form:

P'(10) = (1,3,9,7) (1,3,9,7) (5) (5);
Ss(10) = (+, +, +, -) (-, -, -, +) (+) (-).

According 5 rows of hashing array $P_k(n)$ for DST^{IV} horizontal are selected that allow computing the 5 output data through cyclic convolutions (Table 11).

Table 11: Table of values of simplified elements and signs of matrix DST^{IV}, N=8

k\n	0	1	4	13	10	11	14	3	2	12
0:	1	3	9	7	1	3	9	7	5	5
1:	3	9	7	1	3	9	7	1	5	5
4:	9	7	1	3	9	7	1	7	5	5
13:	7	1	3	9	7	1	3	9	5	5
2:	5	5	5	5	5	5	5	5	5	5
0:	+	+	+	-	-	-	-	+	+	-
1:	+	+	-	+	-	-	+	-	+	-
4:	+	-	+	+	-	+	-	-	+	-
13:	-	+	+	+	+	-	-	-	+	-
2:	+	+	+	+	-	-	-	-	-	+

Specific sign array corresponds to the number of cyclic convolutions of DST^{IV} for size N=5 a 4-point convolution or 8-point with identical sequences, which resulted in four defined output data values and one output value determined by single-point product.

Combining the results of convolutions is performed horizontally through appropriate coordinates, starting with the first element of subarrays. Output data of DST^{IV} are scaled by two and 5 values are determined: X(0), X(1), X(4), -X(3), X(2) for proposed algorithm.

4 Conclusions

The general method for synthesis of algorithms and efficient computation of arbitrary number of transform length for each of the four types of DST can be performed on the basis of response of hashing arrays and then using fast cyclic convolution algorithms. The main characteristics of algorithm that specifies the types of DST^{I-IV} are: function of basis arguments; initial dimension of basis matrix; sequences of input data; sequence of output data; convolution with identical sequences; version of hashing arrays; axes of symmetry for size of transform. As a result, let us assess the advantages of the algorithm of computation of DST based on cyclic convolutions:

- general method of using hashing array P(n), which corresponds to the cyclic decomposition of substitution of rows/columns from basis matrix of arguments, to arrive at an efficient conversion of the basis of an arbitrary DST length into parallel circular structures has been proposed;
- analysis of the level of simplified hashing array P'(n) with supplement of respective subarray of S(n) signs reduces the amount of computation of cyclic convolutions;
- an efficient scheme for the definition of identity and quasi identity cyclic submatrices from the basis matrix structure has been proposed;
- the synthesis of algorithms, including determination of P(n), P'(n), S(n) and analysis of the structure of basis matrix uses integer arithmetic and is not elaborate.

The general method for the conversion of the DST into convolution structures are now available and have been found to be very efficient for hardware implementation using VLSI technology. Separate computations of cyclic convolutions, which are structured according to this approach to basis matrix, and the combinations of results make proposed technique topical for concurrent programming and for implementation in parallel systems.

References:

[1] N. Ahmed, T. Natarajan, and K. R. Rao, Discrete cosine transform, *IEEE Transactions on Computations*, Vol. C-23, 1974, pp. 90-93.

[2] A.K. Jain, "A fast Karhunen-Loeve transform for a class of random processes," *IEEE Trans. Commun.*, vol. COM-24, 1976, pp. 1023-1029.

[3] A. K. Jain, A sinusoidal family of unitary transforms, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-1, October 1979, pp. 356-365.

[4] Z. Wang, A fast algorithm for the discrete sine transform implemented by the fast cosine transform, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 30, No. 5, October 1982, pp. 814–815.

[5] K. Rose, A. Heiman, and I. Dinstein, DCT/DST alternate-transform of image coding, *IEEE Transactions on Communications*, Vol. 38, No. 1, Jan.1990, pp. 94–101.

[6] V. Britanak, P. Yip, and K. R. Rao, *Discrete Cosine and Sine Transforms*. New York, NY: Academic Press, 2007.

[7] M. P¨uschel and J. M. F. Moura, Algebraic signal processing theory: cooley-Tukey type algorithms for DCTs and DSTs, *IEEE Transactions on Signal Processing*, vol. 56, no. 4, 2008, pp. 1502–1521.

[8] P.S. Kumar, K.M.M. Rabhu, A set of new fast algorithms for DCTs and DSTs. –International journal of Electronics, vol.76, No. 4, 1994, pp.553-568.

[9] S. Egner and M. Pueschel, Automatic generation of fast discrete signal transforms, *IEEE Transactions on Signal Processing*, Vol. 49, No. 9, September 2001, pp. 1992–2002.

[10] C. M. Rader, Discrete Fourier transform when the number of data samples is prime, *Proceeding of IEEE*, 56, 1968, pp. 1107-1108.

[11] D. F. Chiper, M. N. S. Swamy, M. O. Ahmad, and T. Stouraitis, A systolic array architecture for the discrete sine transform, *IEEE Transactions on Signal Processing*, Vol. 50, No. 9, Sep. 2002, pp. 2347–2354.

[12] P. K. Meher and M. N. S. Swamy, New systolic algorithm and array architecture for prime-length discrete sine transform, *IEEE Transactions on Circuits and Systems II*, vol. 54, no. 3, 2007, pp. 262–266.

[13] M. N. Murty Realization of Prime-Length Discrete Sine Transform Using Cyclic Convolution, *International Journal of Engineering Science and Technology (IJEST)* Vol. 5 No.03, 2013, pp.583-589.

[14] Z. Wang and B. Hunt, The discrete W-transform, *Applied Mathematics and Computation*, Vol. 16, 1985, pp. 19-48.

[15] Agarwal R. C, and J. W. Cooley, New algorithms for digital convolution. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 25, No. 5, October 1977, pp. 392 - 409.

[16] I. Prots'ko, Analysis algebraic system of arguments for prime size DHT. *Proceeding of the XI International Conference TCSET'2012*, Lviv-Slavske, february 2012, –p.428.

[17] I. Prots'ko, *Patent 96540*, Ukraine, G06F 17/16 (2006.01), H03M 7/30 (2006.01).
[18] Anthony F. Breitzman, Automatic Derivation and Implementation of Fast Convolution Algorithms. *A Thesis for the degree of Doctor of Philosophy*, Drexel University, January 2003