International Journal of Electronic Engineering and Computer Science
Vol. 1, No. 1, 2016, pp. 6-13
http://www.aiscience.org/journal/ijeecs

Synthesis of Efficient Algorithms of DST for Types I, IV via Cyclic Convolutions

Ihor Prots'ko*

Department of Civil Security, Lviv State University of Life Safety, Lviv, Ukraine

Abstract

The method of synthesis the efficient algorithms for types I, IV of discrete sine transform (DST) the sequences of arbitrary number of points via cyclic convolutions is considered. The hashing arrays with the simplified arguments of the basis function of sine for synthesis for the efficient algorithm of the arbitrary-number transform lengths is analyzed. The hashing arrays in the process of synthesis algorithm define partitioning of the basis into cyclic submatrices. The examples of the synthesis of the algorithms for I, IV types of DST using proposed method are considered. The hashing arrays, used in the algorithms for the synthesis technique, are more versatile and generally better in terms of indexing mapping in comparison with the existing algorithms.

Keywords

Discrete Sine Transform, Types of DST, Fast Algorithm, Hashing Array, Synthesis, Cyclic Convolution

Received: May 23, 2016 / Accepted: June 4, 2016 / Published online: June 28, 2016

1. Introduction

The discrete sine transform has found a wide range of applications in signal processing such as image processing, adaptive digital filtering and interpolation. Ever since the introduction of the first version of the DST in 1976, eight types of DST [1-3] have been developed. The DST transforms of types I–IV form a group of so-called even discrete sine transform (EDST) and next four odd DST, indicating whether they are even or odd transforms.

The direct and inverse computation of EDST can be presented in the following form:

$$(\text{DST I}_N)^{-1} = (\text{DST I}_N)^T = (\text{DST I}_N); \quad (1)$$

$$(\text{DST II}_N)^{-1} = (\text{DST II}_N)^T = (\text{DST III}_N); \quad (2)$$

$$(\text{DST III}_N)^{-1} = (\text{DST III}_N)^T = (\text{DST II}_N); \quad (3)$$

$$(\text{DST IV}_N)^{-1} = (\text{DST IV}_N)^T = (\text{DST IV}_N), \quad (4)$$

where DST I and DST IV are identical with their inverse transform (1,4). The DST includes for I, IV types, defined by:

for DST I

$$X_{N-1}^{s1}(k) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-2} x(n) \sin\left[\frac{(k+1)(n+1)\pi}{N}\right], \quad (5)$$

for DST IV

$$X_N^{s4}(k) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x(n) \sin\left[\frac{(2k+1)(2n+1)\pi}{4N}\right], \quad (6)$$

where $\alpha(n)=1/\sqrt{2}$, if n=N-1; otherwise $\alpha(n)=1$; $k=0,1,\ldots,N-1$.

These transforms are very well studied and a number of efficient implementation techniques have been developed [4, 5]. There are two categories for efficient computing of the fast DST, indirect and direct implementations. In case of the indirect computation, DST is decomposed into other fast algorithms with smaller scales, such as FFT or through discrete Hartley transform (DHT), discrete cosine transform (DCT) [5]. The Rader type [6] approach of efficient algorithms gives the possibility to compute DST through

* Corresponding author
E-mail address: protsko@polynet.lviv.ua

cyclic convolutions.

Convolutions play a significant role in digital signal processing due to their nature of easy implementation. Moreover, the convolution-based algorithms are found to be efficient for read-only-memory (ROM) - based and adder-based very large scale integration (VLSI) implementation. Several attempts have been made for efficient implementation of prime-length DST and DCT in systolic hardware through convolution formulation [7–10] due to its remarkable advantages over the others, particularly for efficient input/output and data transfer operations. 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 article represents the synthesis of efficient algorithms for types I, IV of DST, what have the identical direct and inverse transform, computation based on the cyclic convolution of arbitrary-number transform lengths. This approach to the synthesis of algorithms is more general and efficient than the algorithms mentioned earlier and is applicable to sequences with arbitrary number of points. Section 1 of this paper presents analysis and simplification the arguments of functions the basis for types I, IV of DST for synthesis of efficient algorithm. In Section 2 the performance of the proposed general technique on examples of the DST I for size N=9, and DST IV for size N=12 are analyzed. In Section 3 the results are discussed and Section 4 presents conclusions.

2. Definition of Simplified Arguments of Basis Function of DST for Types I, IV

Most papers use a transition from discrete transform to compute cyclic convolutions mapping for prime size by Raiders [6], or split composite size on prime factors by Agarwal and Cooley [11], or the combination of these approaches.

Not much work has been dedicated to the development of efficient implementation of generalized techniques for computation of discrete transforms of Fourier class using cyclic convolutions. Paper [12] presents a DCT algorithm that converts the DCT computation into cyclic convolutions. It shows that by using multiplicative groups of integers, one can identify and arrange the computation as convolutions. The index sets can be extended to find a suitable group and the functions that can be used to compute the DCT as a convolution over a larger group.

The operation on integer set (1,2…N-1) of the algebraic system <N-1,*> corresponds to equivalent the arguments of functions the basis matrix of discrete sine transform. In case the size of transform N is prime, algebraic system <N-1,*> is of Abelian group. Besides, the algebraic system <N-1,*> with prime N presents cyclic group, and table of operation is a Hankel circular matrix. Elements of cyclic group are equal to natural power of generate element α ∈ G. Generate element α of cyclic group is a primitive root, such that

$$\alpha_i^{\,n} \bmod N = 1,\; T = \max \varphi(N)\,, \qquad (7)$$

where φ(N) - Euler's function and α is not the only one. Primitive element will be α^{N-1} also. Therefore, all elements of cyclic group can be determined by the powers of primitive element.

Let us analyze the structure of the matrix basis (5, 6) for the types I, IV of DST for integer arguments, where components $c_{k,n}$ are respectively:

for DST I

$$c_{k,n} = (k+1)(n+1)\pi/N, \quad (k,n=0,1,\dots,N\text{-}2); \qquad (8)$$

for DST IV

$$c_{k,n} = (2k+1)(2n+1)\pi/4N, \; (k,n=0,1,\dots,N\text{-}1). \qquad (9)$$

Basis periodic (2π), symmetric (π) and asymmetric (π/2) for the types I, IV of EDST, are presented respectively in Table 1.

Table 1. Properties basis for I, IV types of DST.

Type	Periodic T	Asymmetric	Symmetric
DST I	2N	N	N/2
DST IV	8N	4N	2N

Matrix arguments C_a the types I, IV of DST for property of periodic is equal respectively

$$C^{I}_{a}(k,n) = [(k+1)(n+1) \bmod T\,], \qquad (10)$$

$$C^{IV}_{a}(k,n) = [(2k+1)(2n+1) \bmod T\,], \qquad (11)$$

where (k,n=0,1,…,T-1).

Based on substitutions of rows of data matrix (10, 11) the hashing arrays $P(n)$ are formed, that define block cyclic structures of basis matrix.

Accordance properties (Table 1) of simplified matrix elements $c'_{k,n}$ of the arguments is determined by consistent performances

$$c_{k,n} = T - [(c_{k,n}) \bmod T\,],\ \text{if}\ [(c_{k,n}) \bmod T\,] > T/2; \qquad (12)$$

$$c'_{k,n} = T/2\text{-}\{T\text{-}[(c_{k,n}) \bmod 2N]\},\ \text{if}\ \{T\text{-}[(c_{k,n}) \bmod T]\}>T/4, \qquad (13)$$

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

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

$$Ss\,[k,n] = \begin{cases} +1, & if \quad 0 < c_{k,n} < T/2 \\ 0, & if \quad c_{k,n} = T/2 \\ -1, & if \quad T/2 < c_{k,n} < T \end{cases} . \qquad (14)$$

Availability of the arguments $c'_{k,n}$ on the interval (0, *T/4*) or (0, *T/2*) with opposite signs reduce the number of computations in the block cyclic structures of basis matrix.

The connection of the discrete sine transforms to a cyclic group is used in order to explain the structure of the basis into cyclic submatrices**.** Let us consider the group table of operation or parts of them (Table 2), corresponding the table of operation of multiplication modulo (10, 11).

Table 2. Table of operation of a cyclic group.

*	**0**	**1**	**2**	**…**	***N*-1**
0	$c_{(0,0)}$	$c_{(0,1)}$	$c_{(0,2)}$	…	$c_{(0,(N-1))}$
1	$c_{(1,0)}$	$c_{(1,1)}$	$c_{(1,2)}$	…	$c_{(1,(N-1))}$
2	$c_{(2,0)}$	$c_{(2,1)}$	$c_{(2,2)}$	…	$c_{(2,(N-1))}$
3	$c_{(3,0)}$	$c_{(3,1)}$	$c_{(3,2)}$	…	$c_{(3,(N-1))}$
…			…		…
N-1	$c_{(N-1),0}$	$c_{(N-1),1}$	$c_{(N-1),2}$	…	$c_{(N-1),(N-1)}$

Let us analyze the matrix of arguments of degree (*N*)×(*N*) as substitution of ψ_i for each *i* row (column) to first row (column) of matrix, where *N* is prime. Summation of substitutions $\{\psi_1,\psi_2, \psi_3, \psi_4, \psi_5,\ldots,\psi_x\}$ form the cyclic group. The quantity of generating and non-primitive elements is the same for substitutions for algebraic operation of arguments of multiplication by modulo *N* (*= (*n* x *k*) mod *N*). Based on the substitutions of rows/columns from data matrix (10, 11), hashing arrays *P*(*n*) are formed, and correspondence of the cyclic decomposition of substitution. The hashing array, according to the value of size *N* and type of transform is equal respectively

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

$$(n_{21}, n_{22}, n_{23,\ldots}, n_{2L2})\ldots(n_{kL1}, n_{kL2}, \ldots, n_{kLk}), \qquad (15)$$

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:

$$n = (L_1+L_2+\ldots+L_k). \qquad (16)$$

The k number of subarrays in *P*(*n*) is determined by the value of N size (simple, power of simple, composite) of transform and types of EDST. Forming hashing array briefly defines the block cyclic structures of Hankel submatrices in the basis matrix [13].

In case the composite value $N=N_1N_2\ldots N_i$ of size, the elements $c_{i,j}$ = ((*i* x *j*) mod N_i) of the cyclic group match the elements $c_{\alpha i,\alpha j}$ = ((α *i* x α *j*) mod *N*) of Hankel circular submatrices, where the first row is one of subarrays in *P*(*n*), where α= $N_1, N_2, \ldots, N_i$.

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 (indicate short +,-,0). The simplified hashing arrays are:

$$P'(n) = P'(n_1)P'(n_2)\ldots P'(n_k), \qquad (17)$$

$$Ss(n) = Ss(n_1)\; Ss(n_2)\ldots Ss(n_k) \qquad (18)$$

for a given N size and of DST type determined by a simplified matrix C'_a, with elements (12, 13) and signs *Ss*(*n*) are defined by (14).

Hashing array *P*(*n*) of transform defines the specific character of the structure of EDST basis matrix reduced to cyclic submatrices. The specific parameters *P*(*n*) that characterize accordingly modified basis matrix are:

k - the number of subarrays in hashing array;

($L_1, L_2, \ldots, L_k$) - the size of subarrays;

n_{i1} - first element of each subarray, *i* = 1 (1) *k*.

The examples of hashing arrays for types I, IV of DST are presented in Table 3.

Table 3. The examples of the hashing array for I, IV types of DST.

Type	***P*(*n*), *P'*(*n*), *Ss*(*n*)**
DST I	(1,3,9,13,11,5)(2,6,4,12,8,10)(7)(0),
N=7	(1,3,2,1,3,2)(2,1,3,2,1,3)(0),
k=4	(+,+,-,-,-,+)(+,+,+,-,-,-)(0);
DST IV	(1,3,9,27,25,19)(5,15,45,23,13,39)(7,21)(11,33,43,17,51,41)(29, 31,37,55,53,47)(35,49),
N=7	(1,3,9,1,3,9)(5,13,11,5,13,11)(7,7)(11,5,13,11,5,13)(1,3,9,1,3,9)(7,7),
k=6	(+,+,+,+,+,+)(+,+,-,+,+,-)(+,+)(+,-,-,+,-,-)(-,-,-,-,-,-)(-,-);
DST I	(1,3,9,11)(15,13,7,5)(2,6)(10,14)(4,12)(8)(16),
N=8	(1,3,1,3)(1,3,1,3)(2,2)(2,2)(4,4)(0)(0),
k=7	(+,+,-,-)(+,-,-,+)(+,+)(-,-)(+,-)(0)(0);
DST IV	(1,3,9,27,17,51,25,11,33,35,41,59,49,19,57,43)(5,15,45,7,21,63, 61,55,37,47,13,39,53,31,29,23),
N=8	(1,3,9,5,15,13,7,11,1,3,9,5,15,13,7,11)(5,15,13,7,11,1,3,9,5,15,1 3,7,11,1,3,9),
k=2	(+,+,+,+,+,-,+,+,-,-,-,-,-,+,-,-)(+,+,-,+,+,-,-,-,-,-,+,-,-,+,+,+);
DST I	(1,3,9,7)(5,15)(11,13,19,17)(2,6,18,14)(4,12,16, 8) (18)(0),
N=10	(1,3,1,3)(5,5)(1,3,1,3)(2,4,2,4)(4,2,4,2)(0),
k=6	(+,+,+,+)(+,-)(-,-,-,-)(+,+,-,-)(+,-,-,+)(0);
DST IV	(1,3,9,27)(5,15,45,55)(7,21,63,29)(11,33,19,57)(13,39,37,31)(17,5 1,73,59)(23,69,47,61)(25,75,65,35)(41,43,49,67)(53,79,77,71),
N=10	(1,3,9,13)(5,15,5,15)(7,19,17,11)(11,7,19,17)(13,1,3,9)(17,11,7, 19)(17,11,7,19)(15,5,15,5)(1,3,9,13)(13,1,3,9),
k=10	(+,+,+,+)(+,+,-,-)(+,+,-,+)(+,+,+,-)(+,+,+,+) (+,-,-,-) (+,-,-,-)(+,-,- ,+)(-,-,-,-)(-,-,-,-);

Then the process of synthesis analyzes the structure of basis matrix that defines the specific character of computational algorithm. The finding of identical and quasi-identical submatrices (with the same index, but opposite sign) is based on the values of parameters of hashing array *P*(*n*) and hashing array *P'*(*n*), supplemented array $Ss(n)$ of sine signs. Searching for identity cyclic submatrices among value elements of basis matrix may be done in advance, but a large search of all elements requires significant amount of memory to store and associated time costs. Only the first elements of submatrices in the analysis the basis structure of EDST can be identified more effectively [14]. The analyses are conducted on repeatability of cyclic structures and compared for coordinates of the first element ($c'_{1,k}$) of cyclic submatrices, which belong to the elements of simplified hashing arrays $P'(n)$. The analyses of reiteration the cyclic submatrices in the matrix structure reduces the number of cyclic convolutions in the computational algorithm of EDST. In case of the identical submatrices placed along the vertical of basis matrix, one cyclic convolution is computed. In case of the identical submatrices placed along the horizontal of basis matrix, one cyclic convolution with combined $x(n)$ of input data is computed.

Consider the examples using generalized scheme for synthesis of algorithm EDST that have specifics for types and concrete sizes of transform.

3. Algorithms of DST for I, IV Types Using Cyclic Convolutions in Examples

Distribution of cyclic submatrices in basis matrix structures for characteristics of hashing array $P(n)$ determine the complexity of the algorithm for efficient computation of EDST types.

Many implementations in our examples for $P'(n)$, $Ss(n)$ (Table 3) have sequences with reiterative identical groups of elements of cyclic convolution. In the following case, $h(n)=(h_1,h_2,\dots,h_m,\ h_1,h_2,\dots,h_m)$ or if identical group of elements has inverse sign $h(n) = (h_1,h_2,\dots,h_m, -h_1,-h_2,\dots,-h_m)$, $n=2m$, the cyclic convolution has reduces computational complexity by $n/m=2m/m=2$ times.

3.1. Specific Algorithm for DST I of Size *N*=9

Consider the generalized scheme for synthesis of algorithm and computation of DST I with an example for N=9. The hashing arrays $P(n_i)$ and $P'(n_i)$, $Ss(n_i)$ are:

a) (1,5,7,17,13,11)(3,15)(9)(2,10,14,16,8,4)(6,12),

(1,4,2,1,4,2)(3,3)(0)(2,1,4,2,1,4)(3,3),

(+,+,+,-,-,-)(+,-)(0)(+,-,-,-,+,+)(+,-);

b) (1,7,13)(3)(5,17,11) (9)(15)(2,14,8)(4,10,16)(6)(12),

(1,2,4)(3)(4,1,2)(0)(3)(2,4,1)(4,1,2)(3)(3),

(+,+,-)(+)(+,-,-)(0)(-)(+,-,+)(+,-,-)(+)(-);

Versions a), and b) of the hashing array $P(n)$ determines the same complexity of the algorithm for efficient computation. Consider example a) for synthesis of algorithm and computation of DST I of size N=9. The initial matrix $C^I_a(k,n)$ defines for the form (10): $C^I_a(k,n)= [(k+1)(n+1) \bmod 15]$.

Using first row and fifth row the arguments of the basis matrix for the substitution, which describe for the hashing array of the form:

$$P(17)= (1,5,7,17,13,11)(10,14,16,8,4,2)(3,15)(6,12)(9).$$

Using (12-14) accordance the hashing array $P(17)$, we are obtained the simplified hashing array $P'(17)$ with the appropriation the sign arrays $Ss(17)$ of the form:

$$P'(17) = (1,4,2,1,4,2)(1,4,2,1,4,2)(3,3)(3,3)(0),$$

$$Ss(17) = (+,+,+,-,-,-)(-,-,-,+,+,+)(+,-)(+,-)(0).$$

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

(i, j) 1 2 3 4 5 6 7 8 9 10 11 12 13 14 1516

(1,5,7,17,13,11)(10,14,16,8,4,2)(3,15)(6,12)

(1,4,2, 1, 4, 2) (1, 4, 2, 1,4,2) (3,3) (3,3),

(+,+,+, -, -, -) (-, -, -, +,+,+) (+,-) (+,-).

Coordinates of the first elements of submatrices are determined by (i + Li), (j + Li), starting with i = 1, j = 1. The values 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 x n_j) mod 2N, in the case of a value greater than N using simplified expression (12,13).

Table 4. Table of coordinates and first elements with signs of submatrices DST I, N=9.

(i+L_i, j+L_i) – $s_{i,j}c_{i,j}$; (sign and value)							
						(1,13)- +3;	(1,15)- +3;
(1,1)- +1;			(1,7)- -1;			(3,13)- +3;	(3,15)- +3;
						(5,13)- +3;	(5,15)- +3;
						(7,13)- -3;	(7,15)- +3;
(7,1)- -1;			(7,7)- -1;			(9,13)- -3;	(9,15)- +3;
						(11,13)- -3;	(11,15)- +3;
(13,1)-+3;	(13,3)-+3;	(13,5)-+3;	(13,7)- -3;	(13,9)- -3;	(13,11)- -3;	(13,13)- 0;	
(15,1)-+3;	(15,3)-+3;	(15,5)-+3;	(15,7)-+3;	(15,9)-+3;	(15,11)- +3;		

Table 4 summarizes the basis matrix of arguments of dimension (17x17), where the number of horizontal and vertical elements is indicated.

Full matrices (17x17) of simplified arguments and signs of DST I, N=9 are presented in Tables5, and 6. These matrices are not computed, except for the first simplified element with the sign of submatrices for the definition of identity cyclic submatrices (Table 4).

Table 5. Matrix of simplified arguments of DST I, N=9.

i\j	1	5	7	17	13	11	10	14	16	8	4	2	3	15	6	12	9
1	1	4	2	1	4	2	1	4	2	1	4	2	3	3	3	3	9
5	4	2	1	4	2	1	4	2	1	4	2	1	3	3	3	3	9
7	2	1	4	2	4	4	2	1	4	2	4	4	3	3	3	3	9
17	1	4	2	4	2	2	1	4	2	4	2	2	3	3	3	3	9
13	4	2	4	2	1	1	4	2	4	2	1	1	3	3	3	3	9
11	2	4	2	1	4	4	2	4	2	1	4	4	3	3	3	3	9
10	1	4	2	1	4	2	1	4	2	1	4	2	3	3	3	3	0
14	4	2	1	4	2	1	4	2	1	4	2	1	3	3	3	3	0
16	2	1	4	2	4	4	2	1	4	2	4	4	3	3	3	3	0
8	1	4	2	4	2	2	1	4	2	4	2	2	3	3	3	3	0
4	4	2	4	2	1	1	4	2	4	2	1	1	3	3	3	3	0
2	2	4	2	1	4	4	2	4	2	1	4	4	3	3	3	3	0
3	3	3	3	3	3	3	3	3	3	3	3	3	9	9	0	0	9
15	3	3	3	3	3	3	3	3	3	3	3	3	9	9	0	0	9
6	3	3	3	3	3	3	3	3	3	3	3	3	9	9	0	0	0
12	3	3	3	3	3	3	3	3	3	3	3	3	9	9	0	0	0
9	9	9	9	9	9	9	0	0	0	0	0	0	9	9	0	0	9

Table 6. Matrix of signs of DST I, N=9.

1	5	7	17	13	11	10	14	16	8	4	2	3	15	6	12	9
+	+	+	-	-	-	-	-	-	+	+	+	+	-	+	-	0
+	+	-	-	-	+	-	-	+	+	+	-	-	+	-	+	0
+	-	-	-	+	+	-	+	+	+	-	-	+	-	+	-	0
-	-	-	+	+	+	+	+	+	-	-	-	-	+	-	+	0
-	-	+	+	+	-	+	+	-	-	-	+	+	-	+	-	0
-	+	+	+	-	-	+	-	-	-	+	+	-	+	-	+	0
-	-	-	+	+	+	-	-	-	+	+	+	-	+	+	-	0
-	-	+	+	+	-	-	-	+	+	+	-	+	-	-	+	0
-	+	+	+	-	-	-	+	+	+	-	-	-	+	+	-	0
+	+	+	-	-	-	+	+	+	-	-	-	+	-	-	+	0
+	+	-	-	-	+	+	+	-	-	-	+	-	+	+	-	0
+	-	-	-	+	+	+	-	-	-	+	+	+	-	-	+	0
+	-	+	-	+	-	-	+	-	+	-	+	0	0	0	0	0
-	+	-	+	-	+	+	-	+	-	+	-	0	0	0	0	0
+	-	+	-	+	-	-	+	-	+	-	+	0	0	0	0	0
-	+	-	+	-	+	-	+	-	+	-	+	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Hashing array

$P(n)$→ (1,5,7,17,13,11)(10,14,16,8,4,2)(3,15)(6,12) specifies the order of elements of input data of the discrete sine transform using cyclic convolutions. In accordance with $P(n)$, matrix-column has such an order of input data:

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

In the result of the analyses (horizontal/vertical) of reiteration the cyclic submatrices in the matrix structure, the process of synthesis of efficient algorithms of DST I determines the order of combination of input data *x(n)*. Performance of element-wise additions of input data will be used for identity cyclic submatrices placed horizontally:

{*x*(1), *x*(5), *x*(7), -*x*(1), -*x*(5), -*x*(7)} ± {-*x*(8), -*x*(4), -*x*(2), *x*(8),*x*(4), *x*(2)} - identity for coordinates (1,1), (1,7) "-" and (7,1), (7,7) "+";

{*x*(3), -*x*(3)} ± {*x*(6),-*x*(6)} - identity for coordinates (1,13), (1,15) "+" and (7,13), (7,15) "-";

2[(*x*(1) + *x*(8)) - (*x*(5) + *x*(4)) + (*x*(7) + *x*(2))] - identity for coordinates (13,1), (13,3),… (13,11);

2[(*x*(1) - *x*(8)) - (*x*(5) - *x*(4)) + (*x*(7) - *x*(2))] - identity for coordinates (15,1), (15,3),… (15,11).

Computation of cyclic convolution is performed once for combined input data for identity submatrices selected for analysis vertically. The number of cyclic convolutions DST I of size N=9 is two3-point convolution and four of one-point.

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*(5), *X*(7), *X*(8), *X*(4), *X*(2), *X*(3), *X*(6).

The resulting structure for DST I of size *N*=9 consists of such components (Fig.1): ±/U – element-wise addition/subtraction units, n-point CCU – cyclic convolution units.

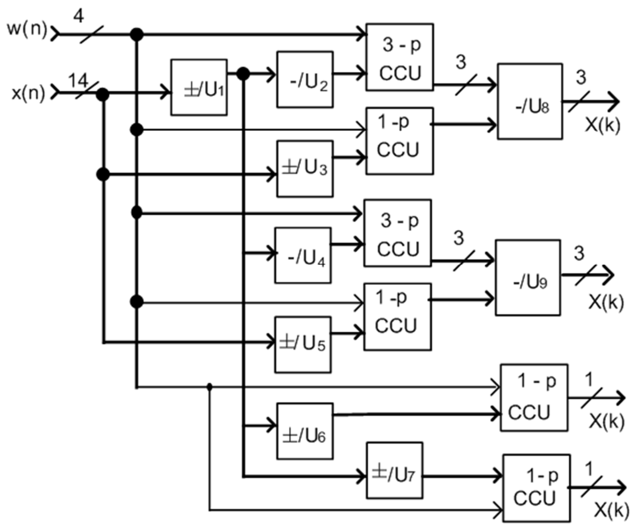

Figure 1. The structure for DST I of size *N*=9.

The computation of DST I of size N=9 for another b) hashing array needs two 3-point convolution and four of one-point cyclic convolutions also.

3.2. Specific Algorithm for DST IV of Size N=12

Consider the peculiarities features of the generalized scheme for synthesis of algorithm and computation of DST IV for size *N*=12. The hashing array *P*(*n*) and *P*'(*n*), *Ss*(*n*) are:

a)

(1,5,25,29,49,53,73,77)(3,15,75,87,51,63,27,39)(7,35,79,11,55,83,31,59)(9,45,33,69,57,93,81,21)(13,65,37,89,61,17,85,41)(19,95,91,71,67,47,43,23),

(1,5,23,19,1,5,23,19)(3,15,21,9,3,15,21,9)(7,13,17,11,7,13,17,11)(9,3,15,21,9,3,15,21)(13,17,11,7,13,17,11,7)(19,1,5,23,19,1,5,23),

(+,+,+,+,-,-,-,-)(+,+,-,-,-,-,+,+)(+,+,-,+,-,-,+,-)(+,+,+,-,-,-,-,+)(+,-,+,-,-,+,-,+)(+,-,-,-,-,+,+,+);

b)

(1,7,49,55)(3,21,51,69)(5,35,53,83)(9,63,57,15)(11,77,59,29)(13,91,61,43)(17,23,65,71)(19,37,67,85)(25,79,73,31)(27,93,75,45)(33,39,81,87)(41,95,89,47),

(1,7,1,7)(3,21,3,21)(5,13,5,13)(9,15,9,15)(11,19,11,19)(13,5,13,5)(17,23,17,23)(19,11,19,11)(23,17,23,17)(21,3,21,3)(15,9,15,9)(7,1,7,1),

(+,+,-,-)(+,+,-,-)(+,+,-,-)(+,-,-,+)(+,-,-,+)(+,-,-,+)(+,+,-,-)(+,+,-,-)(+,-,-,+)(+,-,-,+)(+,+,-,-)(+,-,-,+);

Consider the a) example for synthesis of algorithm and computation of DST I of size *N*=9.

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

P(24)=(1,5,25,29,49,53,73,77)(15,75,87,51,63,27,39,3)(79,11,55,83,31,59,7,35).

Owing to odd values of hashing array maps to the corresponding *P*(24) hashing array indices, elements $c_{i,j}=(n_{i,j}-1)/2$, and k takes all the values of natural set:

P(24)=(0,2,12,14,24,26,36,38)(**7**,37,43,25,31,13,19,1)(39,**5**,27,41,15,29,**3**,17).

Simplified hashing array and sign array have the form:

P'(24)=(1,5,23,19,1,5,23,19)(15,21,9,3,15,21,9,3)(17,11,7,13,17,11,7,13),

Ss(24)=(+,+,+,+,-,-,-,-)(+,-,-,-,-,+,+,+)(-,+,-,-,+,-,+,+).

According 12 rows of hashing array *P*(24) for DST IV horizontal are selected, which allows computing the 12 output data through cyclic convolutions. The matrix of simplified arguments with signs the first elements of submatrices of DST IV basis transform N=12 is presented in Table 7.

Table 7. Table of coordinates and first elements with signs of submatrices DST IV, *N*=12.

($i+L_i$, $j+L_i$) $-s_{i,j}c_{i,j}$; (sign and value)		
(1,1)- +1;	(1,9)- +15;	(1,17)- -17;
(9,1)- +15;	(9,9)- +15;	(9,17)- +15;
(17,1)- -17;	(17,9)- +15;	(17,17)- -1;

Hashing array of *P*(24) specifies the order of input data *x*(*n*), 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*=12 is only six 8-point cyclic convolutions, which have sequences with reiterative identical groups of elements, or six 4-point cyclic convolutions.

The horizontal 12 rows of hashing array *P*(*n*) are selected, allowing to compute output data values via cyclic convolutions. Output data of transform as a combination of the results of cyclic convolutions are scaled by two and determined for: *X*(0), *X*(2), *X*(11), *X*(9), -*X*(7), *X*(10), *X*(4), *X*(1), *X*(8), -*X*(5), *X*(3), *X*(6). Two output values must be taken with the opposite sign according to computation algorithm.

4. Conclusions and Results

The proposed method, which uses hashing array, is more general in comparison to referred [7-9]. That is the paper [7]uses an efficient restructuring of the computation of the prime *N*-length of DST II into two circular correlations, and then with additional multiplications the results of the correlations. Paper [8] describes computing an *N*-point prime-length DST II through two pairs of [(*N*-1)/4]- point cyclic convolutions, where [(*N*-1)/4] is an odd number. In paper [9] the implementation of prime *N*- length DST II has been transformed into two [(*N*-1)/2] - point cyclic convolutions with the same kernel. That is the referred [7-9] base on the Raiders approach that uses primitive roots and Chinese remainder theorem.

The effectiveness of the obtained independent computing cyclic convolutions uses the fast convolution algorithms [15]. Moreover, the basis matrix structures contain the submatrices that can be identical and quasi-identical, placed horizontally and vertically. That reduces the number of computations of cyclic convolution, because for identity and quasi-identity cyclic submatrices placed horizontally, we perform the computation of single cyclic convolution (first row submatrix and corresponding element-wise additions of input data) and use results only of single cyclic convolutions for all identity submatrices placed vertically.

The Table 8 presents the number *i* of *p*-point cyclic convolution for considered examples.

Table 8. The number *i* of *p*-point cyclic convolution for examples.

EDST \Convolution \ *p* - point	8-	4-	3-	2-	1-
DST I, N=8			2	2	4
DST I, N=9					4
DST IV, N=8	1				
DST IV, N=12		6			

There are no existing general approaches to each types of DST for sequences of arbitrary number of points using cyclic convolutions. Hence, the computational comparison will include the results from our approach and the results obtained from the traditional approach [16, 17], when the transform dimension is exponential of two. Number of arithmetic operations for our algorithms of EDST using cyclic convolutions and the traditional approach [16, 17] are presented in Table 9, where *a* - the number of additions and *m* is the number of multiplications.

Table 9. Number of arithmetic operations, N=8.

Type DST,N=8	Proposed method	Traditional approach
DST I	*m*=6, *a*=22	*m*=6, *a*=19 [16]
DST IV	*m*=14*a*=46	*m*=20,*a*=38 [17]

The proposed method uses cyclic convolutions with minimal numbers of multiplication [11] of sizes 2^n -point.

The synthesis of algorithms is applicable for efficient computation for types I, IV of DST with some specifics for each type. The synthesis of the algorithms for the symmetry transforms of DST I, DST IV use hashing array *P*(*n*), the transforms of DST II/DST III use hashing arrays for column *Pc*(*n*) and row *Pr*(*n*). Each type of transform for the hashing array specifies the order of elements, the number of repeats of input data, the signs of the output values in the process of synthesis of efficient algorithms.

The synthesis of algorithms and efficient computation of arbitrary number of transform length for types I, IV 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 I, IV of DST 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. The main contributions of our paper are as follows:

- 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 EDST length into parallel circular structures has been proposed;

- analysis of the level of simplified hashing array *P*'(*n*) with supplement of respective subarray of *Ss*(*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)*, *Ss(n)* and analysis of the structure of basis matrix uses integer arithmetic and is not elaborate.

The method for the conversion of the EDST into convolution structures makes use of the mature model in programming and hardware implementation. 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] Jain, A. K. (1976). A fast Karhunen-Loeve transform for a class of random processes. *IEEE Trans. Commun.*, vol. COM-24, pp. 1023-1029.

[2] Jain, A. K. (1979). A sinusoidal family of unitary transforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, pp. 356–365.

[3] Wang, Z. (1982). A fast algorithm for the discrete sine transform implemented by the fast cosine transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing,*vol. 30(5), pp. 814–815.

[4] Rose, K., Heiman, A., and Dinstein, I. (1990). DCT/DST alternate-transform of image coding. *IEEE Transactions on Communications*, vol. 38(1), pp. 94–101.

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

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

[7] Chiper, D. F. Swamy, M. N. S., Ahmad, M. O. (2013). A unified VLSI algorithm for a high performance systolic array implementation of type IV DCT/DST. *International Symposium Signals, Circuits and Systems (ISSCS),* pp.1- 4

[8] Meher, P. K. and Swamy, M. N. S. (2007). New systolic algorithm and array architecture for prime-length discrete sine transform. *IEEE Transactions on Circuits and Systems II*, vol. 54(3), pp. 262–266.

[9] Murty, M. N. (2013). Realization of Prime-Length Discrete Sine Transform Using Cyclic Convolution. *International Journal of Engineering Science and Technology (IJEST)* vol. 5(03), pp. 583-589.

[10] Murty, M. N. (2015) Algorithm for realization of Type-I Discrete Sine Transform. *Ultra Scientist*, vol. 27(2) A, pp. 164-168.

[11] Agarwal, R. C, and Cooley, J. W. (1977). New algorithms for digital convolution. *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 25(5), pp. 392-409.

[12] Wagh, M. D., Ganesh, H. (1980). A new algorithm for the discrete cosine transform of arbitrary number of points. *IEEE Trans. on Computers*, C-29 (4), pp. 269-277.

[13] Prots'ko I., Patent 96540 Ukraine, G06F 17/16 (2006.01), H03M 7/30 (2006.01).

[14] Prots'ko, I., Rikmas, R. (2011). Analysis cyclic submatrices in structure of basis discrete harmonic transform. *Proceeding of the VIIth International Conference* (MEMSTECH' 2011), pp.64-66.

[15] Breitzman, A. F. (2004). Automatic Derivation and Implementation of Fast Convolution Algorithms. *Journal of High Performance Computing and Applications*, special issue on ``Computer Algebra and Signal Processing", vol. 37(2), pp. 157-186.

[16] Britanak, V., Yip, P. C., Rao, K. R. (2007). *Discrete Cosine and Sine Transforms: General Properties, Fast Algorithms and Integer Approximations*, Academic Press.

[17] Egner, S. and Püschel M. (2001). Automatic Generation of Fast Discrete Signal Transforms. *IEEE Transactions on Signal Processing*, vol. 49(9), pp. 1992-2002.