The Algorithm and Structures for Efficient Computation of Type II/III DCT/ DST/ DHT Using Cyclic Convolutions

Ihor Prots'ko
Lviv Polytechnic National University, Lviv, Ukraine
Email: ioprotsko@gmail.com

***Abstract*—The general approach to the synthesis of algorithm for efficient computation of type II/III DCT/DST/DHT transforms using cyclic convolutions is considered. The technique is based on a hashing array, which is formed on the basis of simplified arguments of the basis transform. The synthesis of algorithm owing to hashing arrays defines partitioning of the basis into submatrices which can identify and arrange the computation as cyclic convolutions. The example of synthesis of the algorithms and the common computation structures for type II/III DCT/DST/ DHT for the sizes of powers of two are presented.**

***Index Terms*—discrete cosine transform (DCT), discrete sine transform (DST), discrete hartley transform (DHT), fast algorithm, cyclic convolution**

I. Introduction

Discrete transforms and convolutions are main operations and key tools in digital signal processing. The real discrete transforms, including the discrete Hartley transforms (DHT) with its types, and the discrete trigonometric transforms (DTT) with its types (common name discrete harmonic transform), which present signals in frequency domain, are especially widely used [1]. For example, the most common variant of discrete cosine transform is the type-II DCT, its inverse the type-III DCT, is used in JPEG, H.26x image compression, MJPEG video compression, and MPEG family video compression. There, the two-dimensional DCT-II of 8x8 blocks is computed, the results are quantized and entropy coded.

The successful use of transforms relies on the existence of the so-called fast transforms. One of the first applications of fast Fourier transform (FFT) algorithm was to implement convolution faster (theorem of convolution) than the usual direct method. The discrete convolutions obtained importance in various aspects of time-domain, especially in Finite impulse response (FIR) digital filters. The paradox of inverse connection between frequency and time domains lies in the fast computation of discrete transforms that can use convolutions as efficient transform technique.

The technique, first used by Rader for obtaining a prime length DFT [2], identifies cyclic structures within the transform matrix. Using low complexity of convolution algorithms in cyclic structures of basis matrix leads us to efficient computation of transforms [3], [4]. Efficient algorithmic schemes for the conversion of the discrete harmonic transforms into cyclic correlation or convolution structures are now available and have been found to be very efficient for hardware implementation using VLSI technology [5]-[7]. Consequently, cyclic convolution and circular correlation structures provide high computing speed, low computational complexity, and low I/O bandwidth.

The different strategy of cyclic or skew-cyclic structures identification within the transform matrix is investigated in papers [8]-[10]. The algorithm of the DCT/IDCT conversion with any length N by using two N-length linear convolutions or two cyclic convolutions form, such that one can easily implement with technologies that are well suited for doing convolutions, is presented in paper [8]. The paper [9] shows that when the length of a p prime is such that $(p\text{-}1)/2$ is odd, the DCT can be computed as two cyclic convolutions, each of length $(p\text{-}1)/2$. The paper [10] proposes to decompose the computation of the N point DCT into two matrix-vector multiplications, where each matrix is of size $(M-1)\times(M-1)$ and $M = N/2$. Each of the decomposed matrix-vector products is then converted into a pair of $[(M-1)/2]$ point circular convolution-like operations for reduced-complexity of concurrent systolization.

Not much work has been dedicated to development of efficient implementation of generalized techniques for computation of discrete harmonic transforms. Paper [11] presents a DCT algorithm that converts the DCT computation into cyclic convolutions. They show 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.

These techniques have the raised complexity or demand of concrete sizes of transform and type of harmonic transform. Therefore, the approaches and means of discrete transforms reformulation into cyclic convolutions or circular correlations need further development.

Manuscript received November 2, 2013; revised May 2, 2014.

doi: 10.12720/ijsps.2.2.119-127

In this paper, the approach to the synthesis of efficient algorithm based on the cyclic convolutions is proposed. This approach to the synthesis of algorithms is more general and efficient for arbitrary number of points than the algorithms mentioned earlier.

Information technologies widely use DCT, DST, DHT type II, represented respectively by the form:

for DCT^{II}

$$X_N^{c2}(k)=\sqrt{\frac{2}{N}}\sum_{n=0}^{N-1}\alpha(n)x(n)\cos[\frac{k(2n+1)\pi}{2N}],\ k=0,\ 1,\ ...,\ N-1 \quad (1)$$

where α(n)=1/√2, if n=0; otherwise α(n)=1,

for DST^{II}

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

where α(n)=1/√2, if n=N-1; otherwise α(n)=1,

for DHT^{II}

$$X_N^{h2}(k)=\sum_{n=0}^{N-1}x(n)\,\mathrm{cas}\,[\frac{(k(2n+1)\pi}{N}],\ k=0,\ 1,\ ...,\ N-1 \quad (3)$$

The common computation structures of DCT/DST/DHT type II and inverse transforms for the sizes of integer power of two with point of view of the general approach of circular formulation are considered. The rest of the paper is organized as follows. Section 2 of this paper presents analysis and simplified arguments of the discrete harmonic transforms basis. In Section 3, the synthesis of algorithm for computation of discrete harmonic transforms is defined. Section 4, presents the examples to demonstrate common computation structures of DCT/DST/DHT type II/III for the size N=8. Section 5 offers some concluding results of computations of DCT/DST/DHT type II/III for the sizes of integer power of two, and Section 6 presents conclusions.

II. Analysis and Simplified Arguments of the Discrete Harmonic Transforms Basis

Discrete harmonic transform reflects the input of a linear combination of weighted basis functions. There are 8 types of discrete cosine transform (DCT), 8 types of discrete sine transform (DST) [12], called discrete trigonometric transforms, and four types of generalized discrete Hartley transform (DHT) [13]. Wide applied the computation of DCT^{II}, DST^{II}, DHT^{II} and DCT^{III}, DST^{III}, DHT^{III} types using cyclic convolutions need close analysis and further development.

The matrix form of the discrete harmonic transform is defined by:

$$X = W * x \quad (4)$$

where $W(k,n)$ is a basic square matrix; $x(N)$, and $X(N)$ - matrix columns of input and output data; N- size of transform.

The basic square matrix contains harmonic function and can be presented in the form:

W(k,n)=[cos($c_{k,n}$)], discrete cosine transform (DCT);

W(k,n)=[sin($c_{k,n}$)], discrete sine transform (DCT);

W(k,n)=[cas($c_{k,n}$)] = [(cos($c_{k,n}$)+sin($c_{k,n}$)], discrete Hartley transform (DHT).

The analyses of the structured basis matrix for the main types of harmonic transforms for arguments with components $c_{k,n}$ are executed respectively, especially for two/thirt types:

for discrete cosine transform,

for DCT ^{II}

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

for DCT ^{III}

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

for discrete sine transform,

for DST^{II}

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

for DST^{III}

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

for discrete Hartley transform,

for DHT^{II}

$$c_{k,n}=k(2n+1)\pi/N,(k,n=0,1,...N-1) \quad (9)$$

for DHT^{III}

$$c_{k,n}=(2k+1)n\pi/N,(k,n=0,1,...N-1) \quad (10)$$

The periodic (2π), symmetric (π) and asymmetric (π/2) basis functions for each type of transforms are presented respectively in Table I.

TABLE I. Properties of Basis for Discrete Harmonic Transform Types.

Properties Types	Periodic T	Asymmetric	Symmetric
DCT^{II}, DCT^{III}	4N	2N	N
DST^{II}, DST^{III}	4N	2N	N
DHT^{II}, DHT^{III}	2N	N	N/4

Matrix of arguments $H_a(k,n)$ for each of the discrete harmonic transform types for periodic property is respectively equal to

$$H_a(k,n)=[(c_{k,n})\ mod\ (T)]=[h_a(k,n)] \quad (11)$$

where T - period of basis function, k,n=0, 1, ...N-1.

The algebraic system <N-1,*> with operation on set (1, 2…N-1) corresponds to equivalent basis matrix of discrete harmonic 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 matrix of arguments $H_a(k,n)$ as 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, and α is not the only one. Primitive element will also be α^{N-1} also. Therefore, all elements of cyclic group can be determined by the powers of primitive element. Non-primitive elements of cyclic group generate a part of set, and the other part of set is formed by multiplication of two elements of generated set by modulo N.

Let us analyze Hankel matrix of arguments of degree ($N\times N$) as a substitution of π_i for each row (column) $a_i, i\in\{1, 2, \ldots, x\}$ to first row (column) of matrix, where N is prime. Summation of substitutions $\{\pi_1, \pi_2, \pi_3, \pi_4, \pi_5 \ldots \pi_x\}$ form cyclic group. The quantity of generating and non-primitive elements is the same for substitutions and algebraic operation (*=(n x k) mod N) of arguments of multiplication by modulo N. Based on the substitutions of rows/columns from data matrix (11), hashing arrays *P(n)* are formed, and correspond of the cyclic decomposition of substitution. Forming hashing array briefly defines block cyclic structures of basis matrix [14], [15].

Accordance of properties of discrete harmonic transform of simplified matrix elements of the arguments is determined by the consistent performances:

for DCT $^{II-III}$, DST $^{II-III}$

$$h_{k,n}=T-[h_a(k,n)],\ if\ [h_a(k,n)]>T/2 \quad (12)$$

$$\underline{h}_{k,n}=T/2-\{T-[h_a(k,n)]\},\ if\ \{T-[h_a(k,n)]\}>T/4 \quad (13)$$

otherwise, $\underline{h}_{k,n}=c_{k,n}$;

for DHT^{II-III} with even N

$$\underline{h}_{k,n}=[h_a(k,n)]-T/2,\ if\ [h_a(k,n)]>T/2 \quad (14)$$

for DHT^{II-III} with N multiple 4

$$\underline{h}_{k,n}=\{T/4-[h_a(k,n)-T/2]\},\ if\ T/8<\{h_a(k,n)-T/2\}<T/4 \quad (15)$$

$$\underline{h}_{k,n}=T/4+\{T/2-[h_a(k,n)-T/2]\},\ if\ 3T/8<\{h_a(k,n)-T/2\}<T/2 \quad (16)$$

otherwise, $\underline{h}_{k,n}=h_{k,n}$.

Simplified matrix of arguments is complemented with matrix S $[k,n]$ of cosine, sine, casine signs, (17-19) defined by the inequalities:

for DCT^{II-III} matrix S_c of cosine signs

$$S_c[k,n]=\begin{cases}+1, & if \quad 3T/4<h_a(k,n)<T/4\\ 0, & if \quad h_a(k,n)=T/4, 3T/4\\ -1, & if \quad T/4<h_a(k,n)<3T/4\end{cases} \quad (17)$$

for DST^{II-III} matrix S_s of sine signs

$$S_s[k,n]=\begin{cases}+1, & if \quad 0<h_a(k,n)<T/2\\ 0, & if \quad h_a(k,n)=0,\ T/2\\ -1, & if \quad T/2<h_a(k,n)<T\end{cases} \quad (18)$$

for DHT^{II-III} matrix S_h of casine signs

$$S_h[k,n]=\begin{cases}+1, & if \quad 7T/8<h_a(k,n)<3T/8\\ 0, & if \quad h_a(k,n)=3T/8, 7T/8\\ -1, & if \quad 3T/8<h_a(k,n)<7T/8\end{cases} \quad (19)$$

where k,n=0, 1, ...N-1.

Therefore, expression (11) defines elements of matrix and forms a hashing array *P(n)*. Then, using expressions (12-16), one can define the elements of simplified hashing arrays $P'(n)$ and elements of signs arrays *S(n)* (17-19), which take part in the synthesis of algorithm for efficient computation of discrete harmonic transforms using cyclic convolutions.

III. Synthesis of the Algorithm for Computation of Discrete Harmonic Transforms

The research work is aimed at further development of generalized means of synthesis and computation of discrete harmonic transforms, which include DCT^{II}, DST^{II}, DHT^{II} types, on the basis of cyclic convolutions. The convolution structures play an important role in discrete harmonic transforms due to its regularity and simplicity during its software and hardware implementation.

The structural parts of means of synthesis of algorithms and computations of harmonic transforms consist of SU, PU - main components (Fig. 1). These are synthesis unit (SU), processor unit (PU) and unit (W) of computation of the coefficients of harmonic function. Input x(n) and output X(k) are real data sequences. The integer number N given on input of SU and W specifies arbitrary transform size.

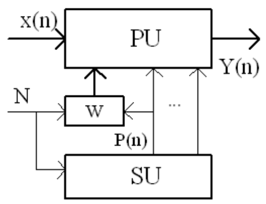

Figure 1. Structural parts of means of the synthesis and computation of discrete harmonic transforms.

The SU defines a hashing array, according to the value of size N and type of harmonic transform:

$$P(n)=P(n_1)P(n_2)\ldots P(n_k)=$$
$$=(n_{11}, n_{12}, n_{13}, \ldots, n_{1L1})(n_{21}, n_{22}, n_{23},\ldots, n_{2L2})\ldots(n_{k1}, n_{k2}, \ldots, n_{kLk}),\ n=(L_1+L_2+\ldots+L_k) \quad (20)$$

where k – number of subarrays, n – size of hashing array, n_{ij} – element of a hashing subarray, Li – number of elements or length of subarray $P(n_i)$. The $P(n)$ determines the structure of the basis matrix and also the order of elements of input data x(n) for computation of discrete harmonic transform.

The hashing arrays can efficiently be represented on smaller values of elements of subarrays $P'(n)$ (15-19), complemented with according subarrays of signs *S(n)* (20-22) on the basis of the property of the symmetry of basis of harmonic transforms. The submatrices of signs *S(n)* consist of values of elements equal to +1, -1, 0 (indicate short +, -, 0). The simplified hashing arrays are:

$$P'(n)=P'(n_1)P'(n_2)\ldots P'(n_k) \quad (21)$$

$$S(n)=S(n_1)S(n_2)\ldots S(n_k) \quad (22)$$

Then SU analyzes the structure of basis matrix that defines the specifics of computational algorithm. The SU conducts analyses of repeatability of cyclic structures and compares coordinates of first element ($n_{1,k}$) of cyclic submatrices. Analyses of reiteration the cyclic submatrices uses the matrix structure (Table II).

TABLE II. TABLE OF COORDINATES OF THE FIRST ELEMENTS OF SUBMATRICES AND THEIR VALUES.

$(i+L_i, j+L_i) - s_{i,j}\ \underline{c}_{i,j}$;			
$(0,0) - s_{i,j}\ \underline{c}_{i,j}$;		$(0,0+L_1) - s_{i,j}\ \underline{c}_{i,j}$;	…
$(0+L_1,1) - s_{i,j}\ \underline{c}_{i,j}$;		$(0+L_1,0+L_1) - s_{i,j}\ \underline{c}_{i,j}$;	…
$(0+L_1+L_2,1) - s_{i,j}\ \underline{c}_{i,j}$;		$(0+L_1+L_2,0+L_1) - s_{i,j}\ \underline{c}_{i,j}$;	
$(0+L_1+L_{2+\dots}+L_k,0) - s_{i,j}\ \underline{c}_{i,j}$;	$(0+L_1+L_2+_{\dots}+L_k,0+L_k) - s_{i,j}\ \underline{c}_{i,j}$;	$(0+L_1+L_2+_{\dots}+L_k,0+2L_k) - s_{i,j}\ \underline{c}_{i,j}$;	…

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. This reduces the number of cyclic convolutions in the computational algorithm of discrete harmonic transform.

To summarize, the algorithm of synthesis can list the steps involved in performing:

Step 1 – determination if the amount N belongs to a subset of integers {2, 4, 8, 16, 32, 64, 128, 256, ...};

Step 2 – definition of cyclic decomposition of substitutions P(n) using rows of argument the basic matrix;

Step 3 – simplification of cyclic decomposition of substituting P'(n) based on the symmetry of the basic functions;

Step 4 – complementation of signs S(n) of simplified cyclic decomposition of substituting P'(n);

Step 5 – analysis of the structure of basic matrix of transforms that used P(n), P'(n), S(n), what describes the block-matrix structure of the discrete harmonic transform of arbitrary size.

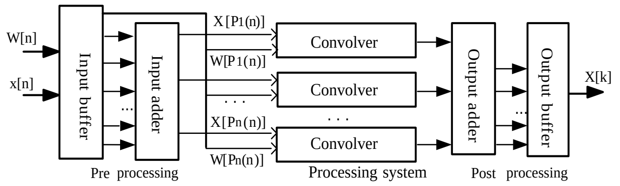

Figure 2. The sequential computational stages of PU.

The processor unit (PU) executes computation of discrete harmonic transforms during three (Fig. 2) sequential stages:

Pre-processing – union of input data according to identical horizontal submatrices;

Processing – computation of cyclic convolutions using efficient algorithms of fast cyclic convolution;

Post processing – union of the intermediate results of cyclic convolutions to output X(k) data.

The first and the last execution stages of PU are simple additions, and a middle stage computes fast cyclic convolutions with multiplications. The derivation of the levels of synthesis and processing is very general, and yields a wide variety of implementations of frequency and time domain techniques. Therefore, consider the examples using generalized scheme for synthesis of algorithm and computation of $DCT^{II}/DST^{II}/DHT^{II}$ for sizes of N=8.

IV. EXAMPLES OF COMPUTATION OF $DCT^{II}/DST^{II}/DHT^{II}$ USING CYCLIC CONVOLUTIONS

A. *The Structure of Direct Computation of $DCT^{II}/DST^{II}/DHT^{II}$ for Size N=8 Using Cyclic Convolutions*

The characteristics of hashing array $P(n)$ determine the complexity of the algorithm for efficient computation of $DCT^{II}/DST^{II}/DHT^{II}$. The specifics for each of transform types are analyzed in [16], [17]. A difference of values (5-10) in rows and columns requires transition from hashing array $P(n)$ to the appropriate hashing array indices of the rows $Pr(n)$ and columns $Pc(n)$. In result the cyclic decomposition of substitutions using rows/columns of argument the basic matrix (11) of the transforms of size N=8 the hashing arrays are:

for DCT^{II}

Pc(16)=(0,1,4,13,8,9,12,5) (15,14,11,2,7,6,3,10);

Pr(16)=(**1**,**3**, 9, **5**, 15,13,**7**,11)(**2**,**6**,14,10) (**4**,12)(8)(**0**);

for DST^{II}

Pc(16)=(0,1,4,13,8,9,12,5) (15,14,11,2,7,6,3,10);

Pr(16)=(**0**,**2**,8,26,16,18,**24**,**10**)(**1**,**5**,17,21)(**3**,11)(**7**,23);

for DHT^{II}

Pc(8)=(1,13,9,5)(3,7,11,15);

Pr(14)=(**1**,13,9,**5**)(**3**,**7**,11,15)(**2**,10)(**6**,14)(**4**)(**0**).

The bold font of the elements Pr(n) indicates output data.

Simplified hashing array
S(n)) defines expressions (12-19),
following form:

for DCT^{II}

Pc'(8)=(1,3,7,5,1,3,7,5),

Sc(8)=(+, +, -, +, -, -, +, -);

Pr'(16)=(1,3,7,5,1,3,7,5) (2,6,2,6)(4,4)(8)(0),

Sc(16)=(+, +, -, +, -, -, +, -) (+, +, -, -) (+, -) (0) (+1);

for DST^{II}

Pc'(8)=(1,3,7,5,1,3,7,5),

Ss(8)=(+, +, +, -, -, -, -, +);

Pr'(16)=(1,3,7,5,1,3,7,5) (2,6,2,6)(4,4)(8,8);

Ss(16)=(+, +, +, -, -, -, -, +)(+, +, -, -)(+, +)(+1,-1);

for DHT^{II}

Pc'(8)=(1,5,1,5)(1,5,1,5);

Sh(8)=(+, -, -, +)(+, -, -, +);

P'r(14)=(1,5,1,5)(1,5,1,5) (2,2) (6,6) (4)(0);

Sh(14)=(+, -, -, +)(+, -, -, +)(+, -)(0,0) (+1) (+1);

Hashing array $P(n)$ of transform defines specific structure of basis matrix reduced to cyclic submatrices. Identity cyclic submatrices are defined via analysis of submatrices distribution in the structure of the basis matrix for Table III and Table IV. Coordinates of the first elements of submatrices are determined by (i+Li), (j+Li), where Li is the size of hashing subarrays, which are chosen for condition of membership value of the first elements of the submatrices in the matrix structure to the element of hashing subarrays (21).

TABLE III. TABLE OF COORDINATES AND FIRST ELEMENTS OF SUBMATRICES DHT^{II}, N=8.

$(i+L_i, j+L_i)$ - $s_{ij}n_{ij}$ – sign with value of first element			
(0,0) – +1;		(0,4) – +1;	
(4,0) – +1;		(4,4) – -1;	
(8,0) – +2;	(8,2) – +2;	(8,4) – +6;	(8,8) – +6;
(10,0) – +6	(10,2) – +6;	(10,4) – +2;	(10,6) – +2;
(12,0) – +4;		(12,4) – -4;	
(13,0) – +0;			

The analysis of the structure of basis matrix (Table III) defines two of 4-point cyclic convolutions with identical group of elements and two of 2-point cyclic convolution with identical group of elements. The remaining two output data are determined through element-wise addition/subtraction.

TABLE IV. TABLE OF COORDINATES AND FIRST ELEMENTS OF SUBMATRICES DCT^{II}, DST^{II}, N=8.

$(i+L_i, j+L_i)$ - $s_{ij}n_{ij}$ – sign with value of first element	
(0,0) – +1; DCT^{II} (0,0) – +1; DST^{II}	
(8,0) – +2; DCT^{II} (8,0) – +2; DST^{II}	(8,4) – +2; DCT^{II} (8,4) – +2; DST^{II}
(12,0) – +4; DCT^{II} (12,0) – +4; DST^{II}	
(13,0) – +0; DCT^{II} (13,0) – +8; DST^{II}	

TABLE V. 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
11:	4	4	4	4	4	4	4	4
7:	8	8	8	8	8	8	8	8

The analysis of the structure of basis matrix (Table IV) defines 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 of two output data. The remaining two output data are determined through one point products. Basis matrix arguments resulted in a form of cyclic submatrices without signs can be reproduced with hashing arrays Pr(n), Pc(n). The matrix of simplified argument without signs of basis transform N=8 is presented in Table V, which corresponds to the generalized Table IV.

Computation of cyclic convolution is performed for combined input data for identity and quasi identity submatrices selected for analysis horizontally and vertically. The resulting structure for $DCT^{II}/DST^{II}/DHT^{II}$ of size N=8 consists of such components (Fig. 3): BRC – buffer register of coefficients, BRD – buffer register of input data, $\pm/U_i$ – element-wise addition/subtraction unit, n-point CCU - cyclic convolution unit, OBRD - buffer register of output data.

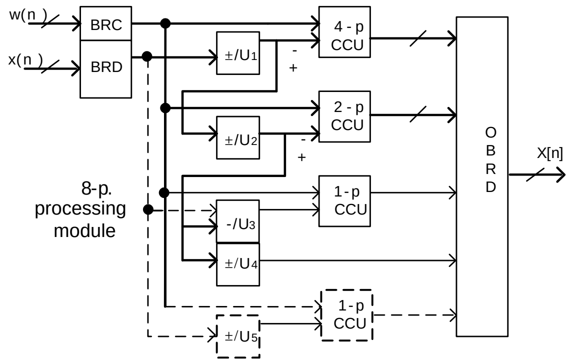

Figure 3. Structure of processing module $DCT^{II}/DST^{II}/DHT^{II}$ of size N=8.

The numbers of cyclic convolutions are the 4-point cyclic convolution and 2-point cyclic convolution, the remaining output data are determined through one point products (components denote edotted line - using for computation of DHT^{II} only). The values of sequence w(n) of coefficients of BRC specifies P'c(n) in trigonometric function for $\varphi_i=2\pi n_i/(T/2)$. Hashing array of Pc(n) specifies the order of input data BRD. Output data of transform in a result of computation are saved in OBRD.

For example, consider the execution of the computation of DCT^{II} in processing module.

The order of sequence of coefficients:

w(n)={cos (φ), cos (3φ), cos(7φ), cos(5φ), cos(2φ), cos(6φ), cos(4 φ)}

where φ=π/16.

The order of sequence of input data:

x(n)=2{x(0),x(1),x(4),x(2),x(7),x(6),x(3),x(5)},

which are combined in corresponding element-wise addition/subtraction by consistent performances in the $\pm/U_1$ unit:

x(0)+x(7), x(1)+x(6), x(4)+x(3), x(2)+x(5);
x(0)-x(7), x(1)-x(6), x(4)-x(3), x(2)-x(5);

in the $\pm/U_2$ unit:

[(x(0)+x(7)]-[x(4)+x(3)],[(x(1)+x(6)]-[(x(2)+x(5)],
[(x(0)+x(7))+(x(4)+x(3))], [(x(1)+x(6))+(x(2)+x(5))];
in the $\pm/U_3$ unit:
[(x(0)+x(7))+(x(4)+x(3))]-[(x(1)+x(6))+(x(2)+x(5))];
in the $\pm/U_4$ unit:
[(x(0)+x(7))+(x(4)+x(3))]+[(x(1)+x(6))+(x(2)+x(5))].

The cyclic submatrices have sequences with reiterative identical groups of elements for cyclic convolution in the form $W(n)=(w_1, w_2, \ldots, w_m, -w_1, -w_2, \ldots, -w_m)$. Computation (Table IV) of the 8-point cyclic convolution is performed as 4-point cyclic convolution and 4-point cyclic convolution as 2-point cyclic convolution with identical sequences by the following formula:

$$\begin{pmatrix} W(m) & \pm W(m) \\ \pm W(m) & W(m) \end{pmatrix} \begin{pmatrix} X_0 \\ X_1 \end{pmatrix} = \begin{pmatrix} W(m) \otimes (X_0 \pm X_1) \\ \pm W(m) \otimes (X_0 \pm X_1) \end{pmatrix} = \begin{pmatrix} Y_0 \\ Y_1 \end{pmatrix} \quad (23)$$

where $\otimes$ - cyclic convolution.

B. *The Structure of Inverse Computation of $DCT^{II}/DST^{II}/DHT^{II}$ for size N=8 Using Cyclic Convolutions*

The following relations hold for inverse $DCT^{II}/DST^{II}/DHT^{II}$ matrices are obtained as their transpose:

$$(DCT^{II})^{-1}=(DCT^{II})^{T}=DCT^{III}$$

$$(DST^{II})^{-1}=(DST^{II})^{T}=DST^{III}$$

$$(DHT^{II})^{-1}=(DHT^{II})^{T}=DHT^{III} \quad (24)$$

Therefore, DCT^{II} and DCT^{III}, DST^{II} and DST^{III}, DHT^{II} and DHT^{III} are inverses of each other. Information technologies widely use DCT^{III}, DST^{III}, DHT^{III} types, represented respectively by the form:

for DCT^{III}

$$X_N^{c3}(k)=\sqrt{\frac{2}{N}}\sum_{n=0}^{N-1}\alpha(n)\,x(n)\cos[\frac{(2k+1)n\pi}{2N}],\ k=0,\ 1,\ \ldots,\ N-1 \quad (25)$$

where $\alpha(n)=1/\sqrt{2}$, if n=0; otherwise $\alpha(n)=1$,

for DST^{III}

$$X_N^{s3}(k)=\sqrt{\frac{2}{N}}\sum_{n=0}^{N-1}\alpha(n)x(n)\sin[\frac{(2k+1)(n+1)\pi}{2N}],\ k=0,\ 1,\ \ldots,\ N-1 \quad (26)$$

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

for DHT^{III}

$$X_N^{h3}(k)=\sum_{n=0}^{N-1}x(n)\,\mathrm{cas}\,[\frac{(2k+1)n\pi}{N}],\ k=0,\ 1,\ \ldots,\ N-1 \quad (27)$$

In result of the definition (20) of cyclic decomposition of substitutions P(n) using rows/columns of argument, the basic matrix of the transforms of size *N*=8 hashing arrays are:

for DCT^{III}
Pc(16)=(1,3, 9, 5, 15,13,7,11)(2,6,14,10) (4,12)(8)(0)
Pr(8)=(1,3,9,27,17,19,25,11)→(0,1,4,13,8,9,12,5)
for DST^{III}
Pc(16)=(0,2,8,26,16,18,24,10)(1,5,17,21)(3,11)(7)(15)
Pr(8)=(1,3,9,27,17,19,25,11)→(0,1,4,13,8,9,12,5)
for DHT^{III}
Pr(8)=(1,13,9,5)(3,7,11,15)
Pc(15)=(**1**,13,9,**5**)(**3**,**7**,11,15)(**2**,10)(**6**,14)(**4**,**12**)(**0**).

Simplified hashing array (12-19) defines P'(n) complement of signs S(n), which have the following form:

for DCT^{III}
Pc'(16)=(1,3,7,5,1,3,7,5) (2,6,2,6)(4,4)(8)(0),
Sc(16)=(+, +, -, +, -, -, +, -) (+, +, -, -) (+, -)(0)(+1);
Pr'(n)=(1,3,7,5,1,3,7,5),
Sc(16)=(+, +, -, +, -, -, +, -);
for DST^{III}
Pc'(n)=(1,3,7,5,1,3,7,5)(2,6,2,6)(4,4)(8)(16),
Ss(n)=(+, +, +, -, -, -, -, +)(+, +, -, -) (+, +) (+1)(0).
Pr(n)=(1,3,7,5,1,3,7,5),
Ss(n)=(+, +, +, -, -, -, -, +);
for DHT^{III}
P'c(8)=(1,5,1,5)(1,5,1,5) (2,2) (6,6)(4,4)(0);
Sh(n)=(+, -, -, +)(+, -, -, +)(+, -)(0,0)(+1,-1)(+1);
Pr'(8)=(1,5,1,5)(1,5,1,5);
Sh(n)=(+, -, -, +)(+, -, -, +);

TABLE VI. TABLE OF COORDINATES AND FIRST ELEMENTS OF SUBMATRICES DHT^{III}, *N*=8.

(i+L$_i$, j+L$_i$) - $s_{ij}n_{ij}$ – sign with value of first element				
(0,0) – +1;	(0,4) – +1;	(0,8) – +2;	(0,10) – 6;	(0,12) – +0;
		(2,8) – +2;	(2,10)– 6;	
(4,0) – +1;	(4,4) – -1;	(4,8) – 6;	(4,10) – +2;	(4,12) – -0;
		(6,8) – 6;	(4,12) – +2;	

TABLE VII. TABLE OF COORDINATES AND FIRST ELEMENTS OF SUBMATRICES DCT^{III}, DST^{III}, *N*=8.

(i+L$_i$, j+L$_i$) - $s_{ij}n_{ij}$ – sign with value of first element				
(0,0) – +1;	(0,8) – +2;	(0,12) – +4;	(0,14) – +8;	(0,15) – 0;
		(2,12) – +4;		
	(4,8) – +2;	(4,12) – +4;		
		(6,12) – +4;		

TABLE VIII. TABLE OF VALUES OF SIMPLIFIED ELEMENTS WITHOUT SIGNS OF MATRIX DCT^{II}, DST^{III}, *N*=8.

1	3	7	5	1	3	7	5	2	6	2	6	4	4	8	0
3	7	5	1	3	7	5	1	6	2	6	2	4	4	8	0
7	5	1	3	7	5	1	3	2	6	2	6	4	4	8	0
5	1	3	7	5	1	3	7	6	2	6	2	4	4	8	0
1	3	7	5	1	3	7	5	2	6	2	6	4	4	8	0
3	7	5	1	3	7	5	1	6	2	6	2	4	4	8	0
7	5	1	3	7	5	1	3	2	6	2	6	4	4	8	0
5	1	3	7	5	1	3	7	6	2	6	2	4	4	8	0

The analysis and finding of identical and quasi identical submatrices (with the same index, but opposite signs) in the structure of basis matrix is based on the values of parameters of hashing array P(n) and hashing array P'(n), supplemented array of signs. For Table VI and Table VII only the first elements of Hankel submatrices are identified in the analysis of the structure of the basis in coordinates for placement submatrices.

The half of matrix of simplified arguments without signs of basis DCT^{II}, DST^{III} transform for N=8 is presented in Table VIII, which corresponds to the generalized Table VII and its transpose of matrix for Table V.

Consider the example for synthesis of algorithm and computation of DST^{III} N=8 also. For the contrary, with respect to DST^{II}, basis matrix of arguments DST^{III} with elements for (11) contains the 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 the values (2n+1) of quantity 16 and covers the entire period equal to 4N.

Hashing array DST^{III} for column is more extended and equals to DST^{II}:

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)(0)'

Ss(*n*)=(+,+,+,-,-,-,-,+)(-,-,-,+,+,+,+,-)(+,+,-,-)(-,-,+,+) (+,+)(-,-)(+,-)(0)(0).

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

Pc(n)=(1,3,9,27,17,19,25,11)(2,6,18,22)(4,12)(8,24)(16) (32)→(0,2,8,26,16,18,24,10)(1,5,17,21)(3,11)(7)(15);

Pc'(*n*)=(1,3,7,5,1,3,7,5)(2,6,2,6)(4,4)(8)(16),

Ss(*n*)=(+,+,+,-,-,-,-,+)(+,+,-,-)(+,+)(+)(0).

Hashing array for rows:

Pr(*n*)=(1,3,9,27,17,19,25,11)→(0,1,4,13,8,9,12,5),

The values of rows require transition $(2n+1)\rightarrow n$ from hashing array to the appropriate hashing array indexes of the rows *Pr*(*n*).

Since basis matrix of DST^{III} is a transposed basis to DST^{II}, rows and columns of the basis are rearranged. However, this approach, due to different indexing rows and columns, is a bit specific and perform element-wise additions of input data.

The resulting structure of processing module for inverse $DCTI^{II}/DST^{III}/DHT^{III}$ of size N=8 consists of such components (Fig. 4): BRC– buffer register of coefficients, BRD – buffer register of input data, $\pm/U_i$ – element-wise addition/subtraction unit, n-point CCU - cyclic convolution unit, "-" -inverse signs of the results of convolution, ∑ - output adder, OBRD - buffer register of output data.

Combining the results of cyclic convolutions in Output adder is performed on the basis of coordinates of the first elements of submatrices horizontally (Table VI, VII). Since basis matrix of DCT^{III} / DST^{III} / DHT^{III} is a transposed basis to DCT^{II} / DST^{II} / DHT^{II}, and efficient computing can be synthesised by supplementing the Output adder to Structure of processing module (Fig. 4).

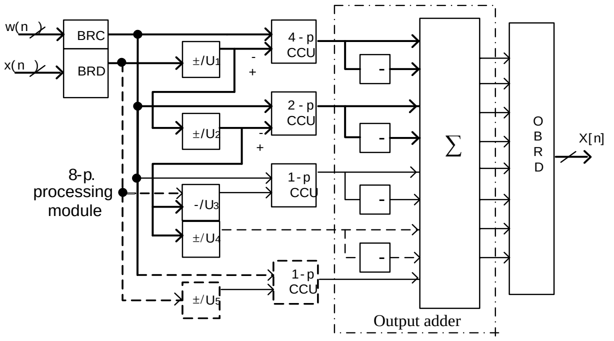

Figure 4. Structure of processing module $DCT^{III}/DST^{III}/DHT^{III}$ of size N=8.

V. Results and Discussion

The general approach is flexibility in the process of synthesis of corresponded algorithm for efficient computation of discrete harmonic transforms using cyclic convolutions. The distribution of cyclic submatrices in basis matrix structures and characteristics of hashing array P(n) determines the computational complexity of the algorithms. The main advantage of approach is availability of more than one hashing array P(n) for each type and size of transform, precisely:

The hashing arrays have the following form, for example, for DCT^{II}, DST^{II} of size N=16:

a) $P(32)=P_1(1)\ P_2(2^4)\ P_3(2^3)\ P_4(2^2)\ P_5(2^1)\ P_6(2^0)$,

b) $P(32)=P_1(1)\ P_2(2^3)\ P_3(2^3)\ P_4(2^2)\ P_5(2^2)\ P_6(2^1)\ P_7(2^1)\ P_8(2^0)\ P_9(2^0)\ P_{10}(2^0)$,

c) $P(32)=P_1(1)P_2(2^2)\ P_3(2^2)\ P_4(2^2)\ P_5(2^2)P_6(2^1)\ P_7(2^1)\ P_8(2^1)\ P_9(2^1)\ P_{10}(2^0)P_{11}(2^0)\ P_{12}(2^0)\ P_{13}(2^0)\ P_{14}(2^0)\ P_{15}(2^0)\ P_{16}(2^0)$,

The hashing arrays have a variety set of elements in subarray, for example, for DHT^{II} of size N=8:

P(15)= (0 (1,13,9,5)(3,7,11,15)(2,10)(6,14)(4)(12),
P'(15)= (1,5,1,5)(1,5,1,5) (2,2) (6,6) (4)(0); or
P(15)= (1,3,9,11) (5,15,13,7) (2,6) (10,14) (4,12) (8),
P'(15)= (1,1,1,1) (5,5,5,5) (2,6) (2,6) (0,0) (0).

That further select hashing array P(n) of DHT^{II} for structure (Fig. 3) of processing module $DCT^{II}/DST^{II}/DHT^{II}$ of size N=8. However, the approach has a bit specific of performance element-wise additions of input data for cyclic convolutions.

In general, computational expenses of proposed technique on the basis of cyclic convolutions on the stage of performance can be presented in the following form:

$$C = C_I^+ + \sum_i Cp_i^{+,*} + C_{III}^+ \qquad (28)$$

where C^+_I – addition/subtraction on the stage of unions of identical and quasi identical cyclic submatrices placed horizontally; $Cp_i^{+,*}$– arithmetic operations for computation of *p*-point cyclic convolution, *i*- number of cyclic convolution; C^{+}_{III} – addition/subtraction on the stage of unions of the results of cyclic convolutions and some input data. The matrix structure defines optimal serial-parallel combination of the results of cyclic convolutions in the last stage of synthesis. Computation of cyclic convolutions includes all multiplications of algorithm for the discrete harmonic transforms. This approach of efficient computation uses availability of the fast convolution algorithms [18]. Moreover, the submatrices in basis matrix structures can be identical and quasi-identical, placed horizontally and vertically. These reduce 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 number of arithmetic operations on the basis of this approach is largely dependent on the choice of fast cyclic convolution algorithm (with a minimum number of multiplications or balance in the operations of addition, and so on). The comparison the number of arithmetic operations from our approach using cyclic convolutions (with minimal numbers of multiplication) and the results obtained from the traditional approach [13], [19], [20] of existing algorithms for the size N=8 of transforms are presented in Table IX, where m - number of multiplication, a - number of addition/subtraction.

TABLE IX. THE NUMBER OF ARITHMETIC OPERATIONS FOR EXAMPLES

Transform	Proposed method	Traditional approach
DHT^{II}, DHT^{III}	N=8, m=6, a=26	N=8, m=4, a=18, [13]
DCT^{II}, DCT^{III}	N=8, m=8, a=33	N=8, m=12, a=29 [19]
DST^{II}, DST^{III}	N=8, m=8, a=37	N=8, m=9, a=24, [20]

Determination of the number of operations for each type and size of transform requires specific analysis of the kernel resulting structure and optimization on stages of combining input data and results of convolutions. Similar to the resulting structure of processing module for direct and inverse $DCTI^{II}/DST^{II}/DHT^{II}$ of size N=8, the computational structure for sizes of transforms $N=2^n$ on basis general approach of efficient computation of discrete harmonic transform can be developed.

TABLE X. THE SIZES OF CYCLIC CONVOLUTIONS OF DCT^{II}/DST^{II}, $N=2^N$

N	4	8	16	32	64	128	...	2^n
k	4	5	6	7	8	9	...	N+2
L_1 L_2 … L_k	2 1	4 2 1	8 4 2 1	16 8 4 2 1	32 16 8 ... 2 1	64 32 16 ... 2 1	...	2^{n-1} 2^{n-2} … 2^2 2 1

In case the computation DCT^{II}/DST^{II} for size of transform $N=2^n$(n=2, 3, …), using for the synthesis of the hashing array in the form a) $P(2^{n+1})=P_1(2^0)\ P_2(2^n)\ P_3(2^{n-1})\ P_4(2^{n-2})\ldots P_{k-1}(2^1)\ P_k(2^0)$, consists the set of the sizes of cyclic convolutions, which is presented in Table X, where k - number of subarrays in hashing array, Li - sizes of cyclic convolutions.

The further researches of proposed approach would be sent to the structures of the other sizes of discrete harmonic transforms. The further researches of proposed approach would be sent to the structures of the other sizes of discrete harmonic transforms.

VI. CONCLUSIONS

The proposed general approach of efficient computation of discrete harmonic transforms of sequences of arbitrary number of points using cyclic convolutions is suitable for DCT, DST, DHT type II/III transforms. The main characteristics of algorithm that specifies the types of transform 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, the proposed algorithm of computation DCT, DST, DHT type II/III using cyclic convolutions posses such advantages:

- 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 length into parallel circular structures;
- 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 are used in analysis of the structure of basic matrix of transforms;
- The synthesis of algorithms, including determination of P(n), P’(n), S(n) with variety set of elements in subarray and analysis of the structure of basis matrix uses integer arithmetic.

The general approach for the conversion of the discrete harmonic transform into convolution structures is now available and has been found to be very efficient for hardware implementation using VLSI technology. Indeed the circular formulation creates possibility to obtain modular structures, what consisting of the simple and regular elements.

REFERENCES

[1] N. L. Tsitsas, “On block matrices associated with discrete trigonometric transforms and their use in theory of wave propagation,” *Journal of Computational Mathematics*, vol. 28, pp. 1-15, 2010.

[2] C. M. Rader, “Discrete fourier transforms when the number of data samples is prime,” *Proc. IEEE*, vol. 56, pp. 104-105, 1968.

[3] M. Heideman, C. Burrus, and H. Johnson, “Prime factor FFT algorithms for real-valued series,” in *Proc. Intl. Conf. on Acoustics, Speech and Signal Processing ICASSP*, 1984, pp. 492-495.

[4] D. Pak-Kong Lun and Wan-Chi Siu, “On prime factor mapping for the discrete Hartley transform,” *IEEE Transactions on Signal Processing*, vol. 40, no. 6, pp. 1399-1411, Jun. 1992.

[5] T. S. Chang, J. I. Guo, and C. W. Jen, “Hardware-Efficient DFT designs with cyclic convolution and subexpression sharing,” *IEEE Trans. Circuits Syst. II: Analog Digital Signal Process.*, vol. 47, no. 9, pp. 886-892, Sep. 2000.

[6] D. F. Chiper, M. N. S. Swamy, M. O. Ahmad, and T. Stouraitis, “Systolic algorithms and a memory-based design approach for a unified architecture for the computation of DCT/DST/IDCT/IDST,” *IEEE Trans. Circuits Syst-I: Regular Papers*, vol. 52, no. 6, pp. 1125-1137, Jun. 2005.

[7] C. Cheng and K. K. Parhi, “Hardware efficient fast DCT based on novel cyclic convolution structures,” *Transactions on Signal Processing*, vol. 54, no. 11, pp. 4419-4434, 2006.

[8] Y. H. Chan and W. C. Siu, “Generalized approach for the realization of discrete cosine transform using cyclic convolutions,” in *Proc. Intl. Conf. on Acoustics, Speech and Signal Processing ICASSP'93*, 1993, pp. 277-280.

[9] V. Muddhasani and M. D. Wagh, “Bilinear algorithms for discrete cosine transforms of prime lengths,” *Signal Processing*, vol. 86, no. 9, pp. 2393-2406, 2006.

[10] P. K. Meher, “Systolic designs for DCT using a low complexity concurrent convolutional formulation,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 16, no. 9, pp. 1041-1050, Sep. 2006.

[11] M. D. Wagh and H. Ganesh, “A new algorithm for the discrete cosine transform of arbitrary number of points,” *IEEE Trans. on Computers*, vol. C-29, no. 4, pp. 269-277, 1980.

[12] Z. Wang and B. Hunt, “The discrete W-transform,” *Appl. Math. Comput.*, vol. 16, pp. 19-48, 1985.

[13] N. C. Hou, H. I. Chang, and O. Ersoy, “Generalized discrete Hartley transform,” *IEEE Trans. Signal Process.*, vol. 40, pp. 2931-2940, 1992.

[14] I. Prots’ko, “Analysis of algebraic system of arguments for prime size DHT,” in *Proc. XI International Conference TCSET’2012*, Feb. 2012, pp. 428.

[15] I. Prots’ko, “The method of formulation the discrete garmonic components of digital signal to cyclic convolutions,” Ukraine Patent 96540, Nov. 10, 2011.

[16] I. Prots’ko, “Algorithm of efficient computation of DCT I-IV using cyclic convolutions,” *Intl. J. of Circuits, Systems and Signal Processing*, vol. 1, no. 7, pp. 1-9, 2013.

[17] I. Prots’ko, “Algorithm of efficient computation of generalized discrete Hartley transform based on cyclic convolutions,” *IET Signal Processing*, 2013.

[18] A. F. Breitzman, “Automatic derivation and implementation of fast convolution algorithms,” Ph. D. thesis, Drexel University, Jan. 2003.

[19] M. Vetterli and H. Nussbaumer, “Simple FFT and DCT algorithms with reduced number of operations,” *Signal Processing*, vol. 6, pp. 267-278, 1984.

[20] Priyanka Jain and Anamika Jain, “Regressive structures for computation of DST-II and its inverse,” *Intl. Scholarly Research Network, ISRN Electronics*, pp. 1-4, 2012.

Ihor Prots’ko was born in Sambir, Ukraine, in 1958. He received the engineer degree and Ph.D. degree from Lviv Polytechnic University in 1981and 1999, where he is currently Assoc. Prof. CAD Department. His current research interests include fast algorithms of discrete transforms and special-purpose digital means.