ISSN 0146-4116, Automatic Control and Computer Sciences, 2012, Vol. 46, No. 5, pp. 207–213. © Allerton Press, Inc., 2012.
Original Russian Text © M.I. Vashkevich, A.A. Petrovsky, 2012, published in Avtomatika i Vychislitel'naya Tekhnika, 2012, No. 5, pp. 36–45.

An Algebraic Method for Synthesizing Fast Algorithms of Discrete Cosine Transform of Arbitrary Size

M. I. Vashkevich and A. A. Petrovsky

Belarusian State University of Informatics and Radioelectronics,
ul. P. Brovki 6, Minsk, 220013 Belarus
e-mail: vashkevich@bsuir.by, palex@bsuir.by

Received July 3, 2012; in final form July 31, 2012

Abstract—An algebraic method for synthesizing fast algorithms of the discrete cosine transform (DCT) of arbitrary size is proposed. The method is based on the polynomial algebra $\mathbb{F}[x]/p(x)$ associated with the DCT. The fast DCT algorithm comes as a result of the step-by-step decomposition of this algebra. In turn, the decomposition requires step-by-step factorization of the polynomial $p(x)$. This problem is solved using Galois's theory, which allows finding all the subfields of the splitting field of the polynomial $p(x)$ where $p(x)$ can be factorized.

Keywords: fast algorithm, discrete cosine transform, Galois's theory

DOI: 10.3103/S0146411612050082

1. INTRODUCTION

The discrete cosine transform (DCT) is critical to many practical applications of digital signal processing such as image/video compression, pattern recognition, speech encoding, medical signal recording (EEG, ECG), etc. [1]. Being widespread, DCT evokes great interest in constructing efficient algorithms to compute it [2]. Since the data blocks are of dimension 4×4, 8×8, or 16×16 in popular DCT application areas such as image and video encoding, the majority of the existing fast algorithms (FA) for computing DCT can be applied when the size of the transform is $n = 2^k$ [3]. Along with this, there is great interest in constructing FA for DCT in other applications when the size of the transform is other than powers of two.

The fast development of computational platforms such as field-programmable gate arrays (FPGA) is accompanied by creating software systems that automatically generate processor structures for DCT [4, 5]. Such systems allow finding, in a sense, the optimal structure of the processor for DCT of a given size for a given computing platform. The automatic search problem of the processor's structure for DCT is solved as an optimization problem over the space of alternative DCT computation algorithms. Thus, the quality of the solution significantly depends on how many alternative fast DCT algorithms exist for the particular size of the transform. In this work, looking to expand the capabilities of such systems, we propose a method that helps obtain fast algorithms for DCT of arbitrary size. In a number of cases, it yields several alternative fast DCT algorithms for one size of the transform.

The proposed method is based on the approach that uses the polynomial algebra $\mathbb{F}[x]/p(x)$ associated with the DCT [6]. Note that the mathematical apparatus of abstract algebra was shown to be efficient in synthesizing fast algorithms for discrete transforms as early as in the Soviet period [7–9].

The polynomial algebra is a vector space

$$\mathcal{A} = \mathbb{F}[x]/p(x). \tag{1}$$

The algebra consists of the set of polynomials with their coefficients belonging to the field and their power being less than $n = \deg(p(x))$. The operation "/" in (1) means that, after the results of summation and multiplication of the elements in the polynomial algebra are reduced modulo the polynomial $p(x)$:

$$r_1(x) = [v_1(x) + v_2(x)] \mod p(x),$$
$$r_2(x) = [v_1(x) \times v_2(x)] \mod p(x),$$

where

$$v_1(x),\ v_2(x),\ r_1(x),\ r_2(x) \in \mathbb{F}[x]/p(x).$$

In [6], the synthesis of fast DCT algorithms is shown to result from the step-by-step decomposition of the polynomial algebra (1), which requires the step-by-step factorization of the polynomial $p(x)$.

Choosing the field of constants $\mathbb{F}$ in (1) is an important part of synthesizing fast DCT algorithms. For instance, in [10], the field of complex numbers $\mathbb{C}$ was chosen as $\mathbb{F}$. This is convenient since any polynomial can be expressed as the product of linear polynomials in $\mathbb{C}$ [11]. Nevertheless, chosen $\mathbb{C}$ as the base field does not answer the question of how we should perform the step-by-step factorization of $p(x)$ required to synthesize fast algorithms. To solve this problem, we introduce the splitting field of the polynomial $p(x)$. To perform sequential factorization of $p(x)$, we need to find all the subfields of the splitting field of the polynomial $p(x)$, which is done using Galois's theory. In each subfield, the polynomial $p(x)$ has unique factorization that can be used to synthesize FA. Since the coefficients of the Chebyshev polynomial included in the definition of the DCT-related algebra are integer, we choose the field of rational numbers $\mathbb{Q}$ as the initial field. When we perform the step-by-step factorization of $p(x)$, the field $\mathbb{Q}$ is gradually added with numbers not included in $\mathbb{Q}$. The field at the last step is the splitting field of the polynomial $p(x)$.

Since we know polynomial algebras for all eight types of discrete cosine and sine transforms [10], the proposed method can be applied to synthesize fast algorithms for any transform from this class. As a practical application of the method, we give an example of two fast algorithms of the 7-point DCT of the fourth type.

2. USING POLYNOMIAL ALGEBRA TO SYNTHESIZE FAST ALGORITHMS

In this section, we consider theoretical foundations of the synthesis of fast DCT algorithms based on the concept of polynomial algebra.

Applying the Chinese remainder theorem (CRT), we can factorize polynomial algebra (1) into the direct sum of one-dimensional subalgebras [10]

$$\mathcal{F}_p: \mathbb{F}[x]/p(x) \to \bigoplus_{0 \le k < n} \mathbb{F}[x]/(x - \alpha_k), \tag{2}$$

given that $\alpha = (\alpha_0, \ldots, \alpha_{n-1})$ are pairwise different zeroes of $p(x)$ and $\alpha_k \in \mathbb{F}$. If we manage to choose the basis $b = (p_0(x), \ldots, p_{n-1}(x))$ in the algebra $\mathbb{F}[x]/p(x)$ and set the unit basis $(x^0) = (1)$ in each subalgebra $\mathbb{F}[x]/(x - \alpha_k)$, the mapping $\mathcal{F}_p$ can be written in the matrix form as

$$\mathcal{F}_p = \mathcal{P}_{b,\alpha} = [p_\ell(\alpha_k)]_{k,\ell}, \tag{3}$$

where $0 \le k \le n$, $0 \le \ell < n$. $\mathcal{P}_{b,\alpha}$ is called the polynomial transform. If different bases β_k are chosen in each subalgebra $\mathbb{F}[x]/(x - \alpha_k)$, the resulting polynomial transform is called scaled:

$$\mathcal{F}_p = \operatorname{diag}(1/\beta_0, \ldots, 1/\beta_{n-1}) \cdot \mathcal{P}_{b,\alpha}. \tag{4}$$

It is well known that the discrete transforms widely used in digital signal processing such as the discrete Fourier transform and DCT can be represented as multiplication of the input vector $\mathbf{x}$ by the matrix

$$\mathbf{y} = \mathcal{P}_{b,\alpha}\mathbf{x}.$$

The fast algorithm of the transform can be written as factorization of the matrix $\mathcal{P}_{b,\alpha}$ into the product of sparse structured matrices. This approach reflects the structure of the fast algorithm and simplifies the process of obtaining its various variants.

As we mentioned above, the fast algorithm is obtained by factorization of the matrix of the transform $\mathcal{P}_{b,\alpha} = B_m \cdot B_{m-1} \cdot \ldots \cdot B_1$, where B_k is a sparse matrix. If we take into account that $\mathcal{P}_{b,\alpha}$ is actually the matrix of the basis change, we will need to find the sequence of bases $b \xrightarrow{B_1} b_1 \xrightarrow{B_2} \ldots \xrightarrow{B_m} \alpha.$ to synthesize FA. In terms of transform (2), the fast algorithm is obtained if $\mathbb{F}[x]/p(x)$ is decomposed into the sum of one-dimensional subalgebras in several steps [10].

One of the ways to perform a step-by-step decomposition of $\mathbb{F}[x]/p(x)$ is to use the factorization $p(x) = q(x)r(x)$. If $\deg(q) = k$ and $\deg(r) = m$, we have

$$\mathcal{F}_p: \mathbb{F}[x]/p(x) \to \mathbb{F}[x]/q(x) \oplus \mathbb{F}[x]/r(x) \tag{5}$$

$$\to \left(\bigoplus_{0\le i<k} \mathbb{F}[x]/(x-\beta_i) \right) \oplus \left(\bigoplus_{0\le j<m} \mathbb{F}[x]/(x-\gamma_i) \right) \quad (6)$$

$$\to \bigoplus_{0\le i<n} \mathbb{F}[x]/(x-\alpha_i), \quad (7)$$

where β_i and γ_j are zeroes of the polynomials $q(x)$ and $r(x)$, respectively, and $\oplus$ is the operation of the direct sum of the algebras. If we choose the basis c in the subalgebra $\mathbb{F}[x]/q(x)$ and the basis d in $\mathbb{F}[x]/r(x)$, step-by-step decomposition (5)–(7) can be written as the product of the matrices

$$\mathcal{P}_{b,\alpha} = P\left(_{c,\beta} \oplus \mathcal{P}_{d,\gamma}\right) B, \quad (8)$$

where $A \oplus B = \begin{bmatrix} A & \\ & B \end{bmatrix}$ stands for the direct sum of the matrices. The matrix B maps the basis b to the concatenation of the bases (c, d) and corresponds to expression (5). The direct sum of the matrices $\mathcal{P}_{c,\beta}$ and $\mathcal{P}_{d,\gamma}$ corresponds to transform (6), which decomposes $\mathbb{F}[x]/q(x)$ and $\mathbb{F}[x]/r(x)$ using the CRT. At stage (7), the one-dimensional algebras are permuted. In (8), this step is associated with the permutation matrix P, which performs the mapping $(\beta,\gamma) \mapsto \alpha$. If B is a sparse matrix, (8) is a fast algorithm since the rest of the multiplier matrices are sparse by definition. We apply the above described procedure in what follows to synthesize fast DCT algorithms.

3. ALGEBRAIC METHOD OF SYNTHESIZING FAST DCT ALGORITHMS

3.1. Method Description

The proposed algebraic method for synthesizing fast DCT algorithms includes the following steps.

Step 1. Find the polynomial algebra $\mathbb{Q}[x]/p(x)$ (and the basis in it) that corresponds to the given type of DCT.

Step 2. Obtain all subfields $\mathbb{L}_i$ of the splitting field $\mathbb{E}$ of the polynomial $p(x)$ using the Galois theory.

Step 3. Factorize the polynomial $p(x)$ step-by-step to match the tower of nested subfields $\mathbb{Q} = \mathbb{L}_0 \subset \mathbb{L}_1 \subset \cdots \subset \mathbb{L}_r = \mathbb{E}$.

Step 4. Use the factorization obtained at step 3 and expressions (5)–(8) to synthesize the FA for the DCT.

Polynomial algebras that correspond to all 8 types of discrete cosine and sine transforms can be found in [6, 10]. We give information on the Galois theory needed to understand the method in the Appendix (at the end of this work).

3.2. Synthesizing FA for DCT-4 of size 7

To elaborate on the proposed method, we consider the problem of synthesizing FA for the fourth-type DCT. The conventional methods of synthesizing FA for DCT are generally applicable when the size of the transform is a composite number [10]. To show the advantage of the proposed method, we synthesize FA for DCT-4 with its size being the prime number 7.

Step 1. We consider the polynomial algebra associated with DCT-4 [10]:

$$\mathcal{A} = \mathbb{Q}[x]/2T_n(x), \quad b = (V_0(x),\ldots,V_{n-1}(x)), \quad (9)$$

where $T_k(x)$ and $V_k(x)$ are the Chebyshev polynomials of the first and third types and of the k-th order, respectively ($x = \cos\psi$):

$$T_n(x) = \cos(n\psi), \quad V_n(x) = \cos\left(n+\frac{1}{2}\right)\psi \Big/ \cos\left(\frac{1}{2}\psi\right).$$

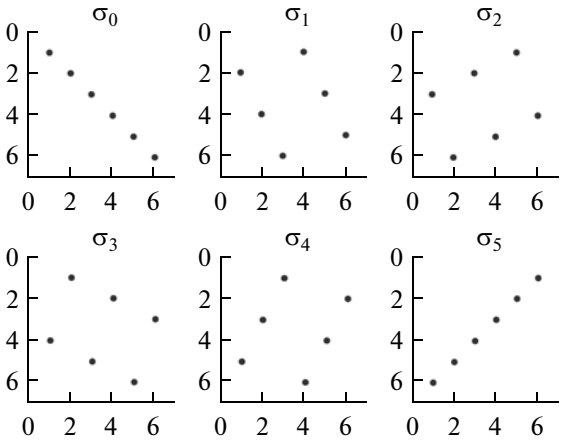

Fig. 1. The Galois groups Gal(P_6) are represented by the permutation matrices $\{\sigma_0, \ldots, \sigma_5\}$. The circles stand for the positions of the unities in the matrix.

Since the roots of $2T_n(x)$ are $\alpha_k = \cos(k+\frac{1}{2})\frac{\pi}{n}$, $0 \le k < n$, then, by (3), the polynomial transform for (9) is given as

$$\mathcal{P}_{b,\alpha} = [V_\ell(\alpha_k)]_{0\le k,\,\ell<n} = \left[\frac{\cos\left(k+\frac{1}{2}\right)\left(\ell+\frac{1}{2}\right)\frac{\pi}{n}}{\cos\left(k+\frac{1}{2}\right)\frac{\pi}{2n}}\right]_{0\le k,\,\ell<n}. \quad (10)$$

If we multiply (10) from the left by the scaling diagonal matrix

$$D_n^{(C4)} = \mathrm{diag}_{0\le k<n}\left(\cos\left(k+\frac{1}{2}\right)\frac{\pi}{2n}\right), \quad (11)$$

we have the DCT-4 matrix:

$$\text{DCT-4}_n = D_n^{(C4)}\mathcal{P}_{b,\alpha} = \left[\cos\left(k+\frac{1}{2}\right)\left(\ell+\frac{1}{2}\right)\frac{\pi}{n}\right]_{0\le k,\,\ell<n}. \quad (12)$$

Thus, expressions (10)–(12) show that DCT-4 is a scaled polynomial transform of type (4).

Step 2. The polynomial algebra associated with the 7-point DCT-4 has the form

$$\mathcal{A} = \mathbb{Q}[x]/2T_7(x), \quad b = (V_0(x),\ldots,V_6(x)). \quad (13)$$

The roots of the polynomial $2T_7(x)$ are $\alpha_k = \cos\frac{\pi(1+2k)}{14}$, $k = 0, 1, \ldots, 6$. Over the field $\mathbb{Q}$, this polynomial has the factorization

$$2T_7(x) = 2xP_6(x), \quad (14)$$

since the root $\alpha_3 = 0$. We use this factorization to obtain the FA.

The polynomial $P_6(x)$ is irreducible over the field $\mathbb{Q}$. Its splitting field is $\mathbb{Q}_{\cos(\frac{\pi}{14})}$, since all the roots of $P_6(x)$ can be expressed via the primitive element $\theta = \cos\left(\frac{\pi}{14}\right)$ as

$$\alpha_k = T_{1+2k}(\theta),\ k = 0,1,2,4,5,6. \quad (15)$$

By the above described FA synthesis method, we need to find the subfields of the field $\mathbb{Q}_{\cos(\frac{\pi}{14})}$, where $P_6(x)$ can be factorized. To do this, we define the Galois group Gal (P_6). We use the method described in [11] to find all the automorphisms of the field $\mathbb{Q}_{\cos(\frac{\pi}{14})}$ that form the group Gal (P_6). We introduce the following notations: $\theta_0 = \alpha_0$, $\theta_1 = \alpha_1$, $\theta_2 = \alpha_2$, $\theta_3 = \alpha_4$, $\theta_4 = \alpha_5$, and $\theta_5 = \alpha_6$. Since the elements $\theta_0, \ldots, \theta_5$ are numbers conjugate with θ (and can be expressed via θ, see (15)), the substitutions

$$\theta \mapsto \theta_k, \quad (16)$$

$k = 0, \ldots, 5$ exhaust the entire Galois group Gal (P_6). Thus, each element of the Galois group maps the system of roots of $P_6(x)$ into itself. Substitutions (16) are handy to be represented by the permutation matrices $\{\sigma_0, \ldots, \sigma_5\}$, which are sparse (Fig. 1).

Thus, Gal (P_6) is found to be a cyclic group of the sixth order ($\mathbb{Z}_6$). One knows that $\mathbb{Z}_6$ has two subgroups viz. the cyclic subgroup of order 2: $\mathbb{Z}_2 = \{\sigma_0, \sigma_5\}$, and the cyclic subgroup of order 3: $\mathbb{Z}_3 = \{\sigma_0, \sigma_1, \sigma_3\}$. Figure 2a shows the group Gal (P_6) as a lattice of subgroups. By the Galois correspondence, there should exist a similar grid of subfields of the field $\mathbb{Q}_{\cos(\frac{\pi}{14})}$ (Fig. 2b).

A question arises of how one can find the subfield $\mathbb{F}$ that corresponds to some subgroup H of the Galois group. In [11], one can find the following solution. If the subgroup $H = \{\sigma_0, \ldots, \sigma_{h-1}\}$ is given, one needs to form the product

$$(x - \sigma_0\theta)(x - \sigma_1\theta)\ldots(x - \sigma_{h-1}\theta); \quad (17)$$

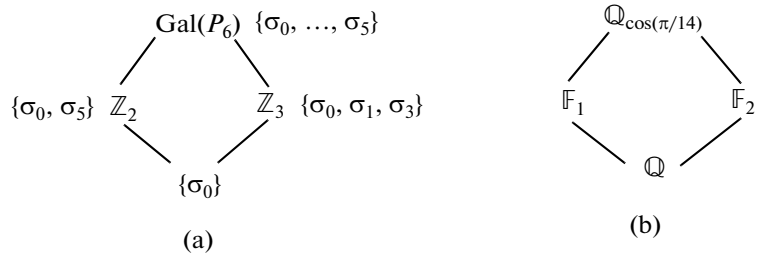

Fig. 2. (a) Subgroups of the Galois group Gal(P_6) and (b) subfields of the decomposition field for $P_6(x)$.

the coefficients of this polynomial, by the fundamental theorem of Galois theory, should belong to the field $\mathbb{F}$ and even generate it.

We use this way to find the subfields $\mathbb{F}_1$ and $\mathbb{F}_2$ (Fig. 2b). We consider the polynomial

$$(x-\sigma_0\theta)(x-\sigma_5\theta) = x^2 - \frac{1}{2}\cos\left(\frac{6\pi}{7}\right), \tag{18}$$

with its coefficients belonging to the field $\mathbb{Q}_{\cos(\frac{\pi}{7})}$. This is the sought field $\mathbb{F}_1$. The field $\mathbb{F}_2$ can be found similarly if we consider the polynomial

$$\begin{aligned}(x-\sigma_0\theta)(x-\sigma_1\theta)(x-\sigma_3\theta) = x^3 + \left(\cos\frac{\pi}{14} + \cos\frac{3\pi}{14} + \cos\frac{9\pi}{14}\right)x^2 \\ -\frac{1}{4}\left(\cos\frac{\pi}{14} + \cos\frac{3\pi}{14} + \cos\frac{9\pi}{14}\right),\end{aligned} \tag{19}$$

with its coefficients belonging to the field $\mathbb{Q}_{\cos\frac{\pi}{14}+\cos\frac{3\pi}{14}+\cos\frac{9\pi}{14}} \cong \mathbb{Q}_{\sqrt{7}}$, which corresponds to field $\mathbb{F}_2$.

Step 3. Using the two towers of fields in Fig. 2b, we can obtain two different ways of the step-by-step factorization of the polynomial $P_6(x)$:

$$2T_7(x) = \underbrace{2x\,P_6(x)}_{\mathbb{Q}} = \underbrace{2x\left(2T_2 - 2\cos\frac{6\pi}{7}\right)\left(2T_2 - 2\cos\frac{4\pi}{7}\right)\left(2T_2 - 2\cos\frac{2\pi}{7}\right)}_{\mathbb{Q}_{\cos\left(\frac{\pi}{7}\right)}} = \underbrace{2\prod_{i=0}^{6}(x-\alpha_i)}_{\mathbb{Q}_{\cos\left(\frac{\pi}{14}\right)}}, \tag{20}$$

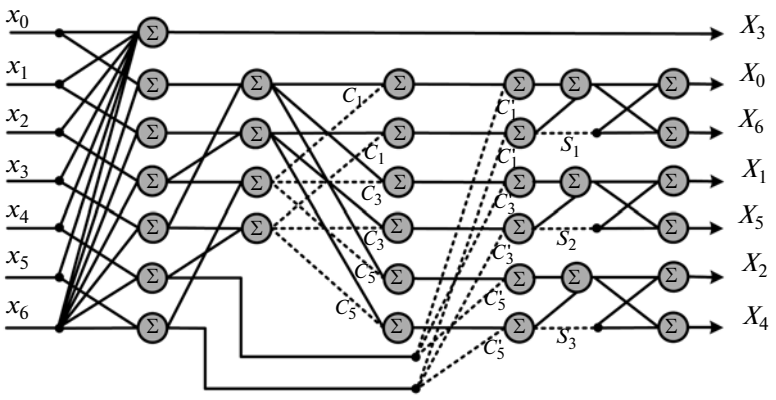

Fig. 3. Flow graph of the fast algorithm for DCT-4 of size 7, $c_k = 2\cos\frac{k\pi}{7}$, $c'_k = c_k^2 - 1$, and $s_k = 2\cos\frac{k\pi}{14}$.

$$2T_7(x) = \underbrace{2x\,P_6(x)}_{\mathbb{Q}} = \underbrace{2x(2T_3 - 2\sqrt{7}T_2 + 6T_1 - \sqrt{7})(2T_3 + 2\sqrt{7}T_2 + 6T_1 + \sqrt{7})}_{\mathbb{Q}\,\sqrt{7}} = \underbrace{2\prod_{i=0}^{6}(x - \alpha_i)}_{\mathbb{Q}\,\cos\left(\frac{\pi}{14}\right)}. \quad (21)$$

Step 4. Using factorizations (20), (21) and the FA synthesis method described in subsection 3.2, we obtained two FA for the 7-point DCT-4. For the details of how the FA can be obtained using the known factorization of the polynomial, see [10, 12].

Figure 3 shows a flow graph of the FA for the 7-point DCT-4 obtained using factorization (20). The signals taken with the negative sign in the summation blocks are marked red. The dotted line stands for the internal signals that are multiplied by the constant (which is given beside it). The summator near the input x_0 summarizes 7 summands and therefore requires 6 two-input summators to be implemented.

Figure 3 does not show the scaling operation for the output data that corresponds to the output data multiplied by the diagonal matrix $D_7^{(C4)}$. If we do not take the scaling multipliers into account, the FA for the 7-point DCT-4 (Fig. 3) requires 15 multiplications and 37 summations.

The fast algorithm obtained using factorization (21) is represented as a product of the sparse matrices

$$\mathrm{DCT4}_7 = D_7^{(C4)} \times P \times B_5 \times B_4 \times B_3 \times B_2 \times B_1, \quad (22)$$

where

$$B_1 = \begin{bmatrix} 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & & & & & & 1 \\ & 1 & & & & & -1 \\ & & 1 & & & & -1 \\ & & & 1 & & & 1 \\ & & & & 1 & & 1 \\ & & & & & 1 & -1 \end{bmatrix}, \; B_2 = \begin{bmatrix} 1 & & & & & & \\ & 1 & & -3 & 4 & -6 & \\ & & 1 & -3 & 6 & -8 & \\ & & & 1 & -1 & 4 & -4 \\ & & & & 1 & -2 & 2 \\ & & & & 1 & -2 & 3 \\ & & & & 1 & -1 & 2 \end{bmatrix}, \; B_3 = \begin{bmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 1 & & & & \\ & & & 1 & & & \\ & & & & a & & \\ & & & & & a & \\ & & & & & & a \end{bmatrix},$$

$$B_4 = \begin{bmatrix} 1 & & & & & & \\ & 1 & & 1 & & & \\ & & 1 & & 1 & & \\ & & & 1 & & 1 & \\ & 1 & & -1 & & & \\ & & 1 & & -1 & & \\ & & & 1 & & -1 & \end{bmatrix}, \; B_5 = \begin{bmatrix} 1 & & & & & & \\ & 1 & b_1 & d_1 & & & \\ & 1 & b_3 & d_3 & & & \\ & 1 & b_9 & d_9 & & & \\ & & & & 1 & b_5 & d_5 \\ & & & & 1 & b_{11} & d_{11} \\ & & & & 1 & b_{13} & d_{13} \end{bmatrix}, \; P = \begin{bmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & & 1 & & & \\ 1 & & & & & & \\ & & 1 & & & & \\ & & & & 1 & & \\ & & & & & & 1 \end{bmatrix},$$

where $a = \sqrt{7}$, $b_k = 2\cos\dfrac{k\pi}{14} - 1$, and $d_k = 8\cos^2\dfrac{k\pi}{14} - 4\cos\dfrac{k\pi}{14} - 1$. If we do not take into account multiplications by the matrix $D_7^{(C4)}$, 15 multiplications and 49 summations are required to perform algorithm (22). Thus, the second variant of the algorithm for the 7-point DCT-4 requires 12 summations more than the first one.

4. CONCLUSIONS

We proposed the algebraic FA synthesis method for DCT of arbitrary size. The method employs the DCT represented as the decomposition matrix of the particular polynomial algebra $\mathbb{F}[x]/p(x)$. The fast algorithm comes as a result of the step-by-step decomposition of this algebra. In turn, the decomposition is closely connected with the factorization of the polynomial $p(x)$. To solve this problem, we use the mathematical apparatus of the Galois theory that helps find all the subfields of the splitting field of the polynomial $p(x)$. In these subfields, the polynomial $p(x)$ has factorization used to obtain the fast algorithm. As a practical application, we obtained two variants of the FA for the 7-point DCT-4 that require the same number of multiplications (15) yet differ by the number of summations (37 and 49, respectively). The proposed method can be used in systems of automatic synthesis of processor structures for DCT [13] to expand the space of alternative fast DCT algorithms.

APPENDIX

The Galois Theory: A Brief Review

The set of all the polynomials of x with the coefficients from the field $\mathbb{F}$ is denoted as $\mathbb{F}[x]$. Suppose $p(x) \in \mathbb{F}[x]$, the splitting field $\mathbb{E}$ of the polynomial $p(x)$ is the minimum extension of the field $\mathbb{F}$ that includes all the roots of $p(x)$. For instance, $\mathbb{Q}_{\sqrt{2}}$ is the splitting field of the polynomial $p(x) = x^2 - 2$. $\mathbb{Q}_{\sqrt{2}}$ is formed by adjunction the number $\sqrt{2}$ to $\mathbb{Q}$. The one-to-one mapping of the field $\mathbb{E}$ onto it self is called an automorphism. The $\mathbb{F}$ automorphism of the field $\mathbb{E}$ is the automorphism φ such that $\varphi(x) = x, \ \forall x \in \mathbb{F}$. The group of $\mathbb{F}$ automorphisms of the field $\mathbb{E}$ is called the Galois group of the polynomial $p(x)$ and is denoted by Gal (p) or $\mathrm{Gal}(\mathbb{E}/\mathbb{F})$. As an example, we define the function $\varphi: \mathbb{Q}_{\sqrt{2}} \to \mathbb{Q}_{\sqrt{2}}$ such that

$$\varphi(a + b\sqrt{2}) = a - b\sqrt{2}.$$

Then, φ is the $\mathbb{Q}$ automorphism of the field $\mathbb{Q}_{\sqrt{2}}$. The group of automorphisms $\mathrm{Gal}(p) = \{\mathrm{id}, \varphi\}$ is a cyclic group of the second order, where *id* is the $\mathbb{Q}$ automorphism leaves all the elements of the field $\mathbb{Q}_{\sqrt{2}}$ in their places. Obviously, $\varphi \cdot \varphi = \mathrm{id}$.

An important result of the Galois theory is that it connects the structure of the subfields of the splitting field of the polynomial $p(x)$ and the structure of the subgroups of the Galois group $\mathrm{Gal}(\mathbb{E}/\mathbb{F})$. Each subgroup H of the group $\mathrm{Gal}(\mathbb{E}/\mathbb{F})$ is matched to the subfield $\mathbb{L} \subset \mathbb{E}$ consisting of the elements of $\mathbb{E}$ that are stationary to automorphisms from H. Similarly, each subfield $\mathbb{L} \subset \mathbb{E}$ has its respective subgroup H of the Galois group that leaves the elements of $\mathbb{L}$ in their places. As a result, one needs only to study all subgroups of the group $\mathrm{Gal}(\mathbb{E}/\mathbb{F})$ to study all the subfields of the field $\mathbb{E}$. Each tower (chain of nested fields)

$$\mathbb{F} = \mathbb{L}_0 \subset \mathbb{L}_1 \subset \cdots \subset \mathbb{L}_r = \mathbb{E} \tag{23}$$

is matched to the normal series of nested (in the reverse order) groups

$$\mathrm{Gal}(\mathbb{E}/\mathbb{F}) = \mathrm{G}_0 \supset \mathrm{G}_1 \supset \cdots \supset \mathrm{G}_r = \{1\},$$

and vice versa (the Galois correspondence).

Thus, the Galois correspondence helps find all the subfields of the decomposition field of the polynomial $p(x)$. Each subfield $p(x)$ can be factorized uniquely into the product of polynomials irreducible over this subfield. Using (23), we can perform the step-by-step factorization of $p(x)$ needed to synthesize FA for DCT.

REFERENCES

1. Ifeachor, E. and Jervis, B., *Digital Signal Processing. A Practical Approach*, Pearson, 2002.
2. Britanak, V., Yip, P., and Rao, K., *Discrete Cosine and Sine Transforms: General Properties, Fast Algorithms and Integer Approximations*, London: Academic, 2007.
3. Miano, J., *Compressed Image File Formats,* Reading, MA: Addison Wesley, 1999.
4. Püschel, M., Moura, J.M.F., Johnson, J., et al., SPIRAL: Code Generation for DSP Transforms, *Proc. IEEE,* Special Issue: *Program Generation, Optimization, and Adaptation*: 2005, vol. 93, no. 2, pp. 232–275
5. Bartholoma, R., Greiner, T., Kesel, F., and Rosenstiel, W., A Systematic Approach for Synthesizing VLSI Architecture of Lifting-Based Filter Bank and Transforms, *IEEE Trans. Circuits Syst. I: Regular Papers,* 2008, vol. 55, pp. 1939–1952.
6. Püschel, M. and Moura, J.M.F., The Algebraic Approach to the Discrete Cosine and Sine Transform, *SIAM J. Comp.*, 2003, vol. 32, pp. 1280–1316.
7. Vairadyan, A.S., Pchelintsev, I.P., and Chelyshev, M.M., Algorithms for Computation of Digital Convolutions, *Zarubezh. Radioelektron.*, 1982, no. 3, pp. 3–34.
8. Labunets, V.G., *Algebraicheskaya teoriya signalov i sistem: tsifrovaya obrabotka signalov* (Algebraic Theory of Signals and Systems: Digital Signal Processing), Krasnoyarsk: Krasnoyarsk. Univ., 1984.
9. Krot, A.M., *Diskretnye modeli dinamicheskikh sistem na osnove polinomial'noi algebry* (Polynomial Algebra-Based Discrete Models of Dynamical Systems), Minsk: Nauka Tekhnika, 1990.
10. Püschel, M. and Moura, J.M.F., Algebraic Signal Processing Theory: Coole-Tukey Type Algorithms for DCTs and DSTs, *IEEE Trans. Signal Proc.*, 2008, vol. 54, no. 4, pp. 1502–1521.
11. van der Waerden, B.L., *Algebra*, London: Springer-Verlag, 2003.
12. Vashkevich, M.I. and Petrovsky, A.A., Use of Polynomial Algebras and Galois Theory for Synthesis of Fast Algorithms of Discrete Cosine Transformations, *Tsifrovaya Obrabotka Signalov*, 2011, no. 3, pp. 2–10.
13. Petrovsky, Al.A., Stankevich, A.V., and Petrovsky, A.A., *Bystroe proektirovanie sistem mul'timedia ot prototipa* (Fast Design of Multimedia Systems from Prototype), Minsk: Bestprint, 2011.